

Fachhochschule Düsseldorf

Fachbereich: Maschinenbau und Verfahrenstechnik

Diplomarbeit

Ralf Eggert

Matrikel-Nr.:288244

Objektorientierte Programmierung eines 2-Kanal
Frequenzanalysators unter DASyLab und LabVIEW

Institut für Strömungsmaschinen IFS

Prof. Dr. Ing. F. Kameier
Dipl. Ing. C. Haukap

Düsseldorf, im Juni 2001

Vorwort

Diese Diplomarbeit entstand an der Fachhochschule Düsseldorf am Institut für Strömungsmaschinen.

Ich möchte mich an dieser Stelle bei meinem sehr geehrten Prof. Dr. Ing. F. Kameier für die Anregung zu dieser Diplomarbeit bedanken.

Besonders möchte ich mich für die Betreuung, die mir während der gesamten Arbeit gewährt wurde, bei Prof. F. Kameier und Dipl. Ing. C. Haukap sowie allen Mitarbeitern des Instituts für Strömungsmaschinen bedanken.

Ebenso möchte ich mich bei allen Verwandten und Freunden bedanken, die mich bei dieser Arbeit unterstützt haben.

Düsseldorf, im Juni 2001

Vorwort

1. Einleitung

2. Grundlagen

2.1 Analyseverfahren mittels der Frequenzanalyse

2.1.1 Fourierspektrum

2.1.2 Amplitudenspektrum

2.1.3 Gesamtpegel

2.1.4 Phasenspektrum

2.1.5 Kreuzleistungsspektrum

2.1.6 Kohärenz

3. Umsetzung mit DASyLab

3.1 Erfassen der Messdaten

3.2 Datenfenster

3.3 Fouriertransformation der Signale

3.4 Diagramm des Amplitudenspektrums

3.5 Diagramm des Leistungsspektrums

3.6 Das Phasenspektrum

3.7 Kohärenz

3.8 Der Gesamtpegel

3.8.1 Der Gesamtpegel im Zeitbereich

3.8.2 Der Gesamtpegel im Frequenzbereich

4. Umsetzung mit LabVIEW

5. Versuchsaufbau und Versuchsdurchführung

5.1 Messungen des Frequenzganges einer Soundkarte

5.2 Überprüfung auf Phasengenauigkeit

5.2.1 Überprüfung der Soundkarte auf Phasengenauigkeit

5.2.2 Überprüfung der Messkarte auf Phasengenauigkeit

6. Vergleich der Messergebnisse

6.1 Resultate der Messungen mit Soundkarten

6.2 Resultat der Messungen mit der Messkarte

7. Zusammenfassung

8. Literaturverzeichnis

9. Anhänge

1. Einleitung

Die Kosten für einen PC mit einer Messkarte und der dazugehörigen Software liegen deutlich unter denen für professionelle Messgeräte. Insbesondere durch die Entwicklung des Multimediabereiches sind Soundkarten, die mittlerweile zu den Standardkomponenten eines PC's gehören, sehr leistungsfähig geworden. Um den Anforderungen des Hifi bzw. Multimediabereiches zu genügen, erfüllen einige der Soundkarten höchste Ansprüche, was auch ihren Einsatz als Messkarte in Betracht zieht.

Die immer weiter steigende Leistungsfähigkeit von Personal-Computern, sowie die Verfügbarkeit von leistungsfähiger und bedienungsfreundlicher Software machen den Einsatz in wissenschaftlichen und messtechnischen Bereichen zunehmend interessanter. Die preisgünstigen Komponenten ermöglichen den Aufbau von Messapparaturen, die durch die Anpassung der Programme sehr variabel sein können, zu einer schnellen und finanziell überschaubaren Angelegenheit.

Die Software für solche Messeinrichtungen ist oft für objektorientierte Programmierung ausgelegt, was ein schnelles kombinieren von Modulen und Virtuellen-Instrumenten ermöglicht. Umstellungen eines Messaufbaus können so schnell und einfach realisiert werden können.

Die vorliegende Untersuchung, ob und wie weit sich eine Soundkarte für messtechnische Anwendungen eignet, soll neben der Umsetzung der Frequenzanalyse mittels DASyLab und LabVIEW Bestandteil dieser Diplomarbeit sein. Insbesondere wird die Amplituden- und Phasentreue verschiedener Soundkarten im Vergleich zu professionellen A/D Wandlerkarten untersucht.

Im Rahmen der Anwendung der objektorientierten Programmierung wird die Benutzerfreundlichkeit und die Dokumentation verschiedener Berechnungen wie Mittelung und Spektralanalyse bewertet.

Das Programm MATLAB wurde im Rahmen der Messungen zur Konvertierung, Überprüfung und Ausgabe (Plots erstellen) ebenfalls verwendet.

2. Grundlagen

Vor knapp 200 Jahren erkannte der Mathematiker, Naturwissenschaftler und Berater Napoleons Joan Baptist FOURIER (1768-1830), dass jedes Signal (Schwingungen bzw. Wellen) aus einheitlichen „Elementen“ zusammengesetzt werden kann, auf mathematischen Weg. Bei diesen „Elementen“ handelt es sich um Sinusschwingungen mit unterschiedlichen Frequenzen und Phasenlagen. Wie z.B. ein Sägezahnsignal annähernd aus Sinusschwingungen zusammengesetzt wird zeigt Abbildung 2.1

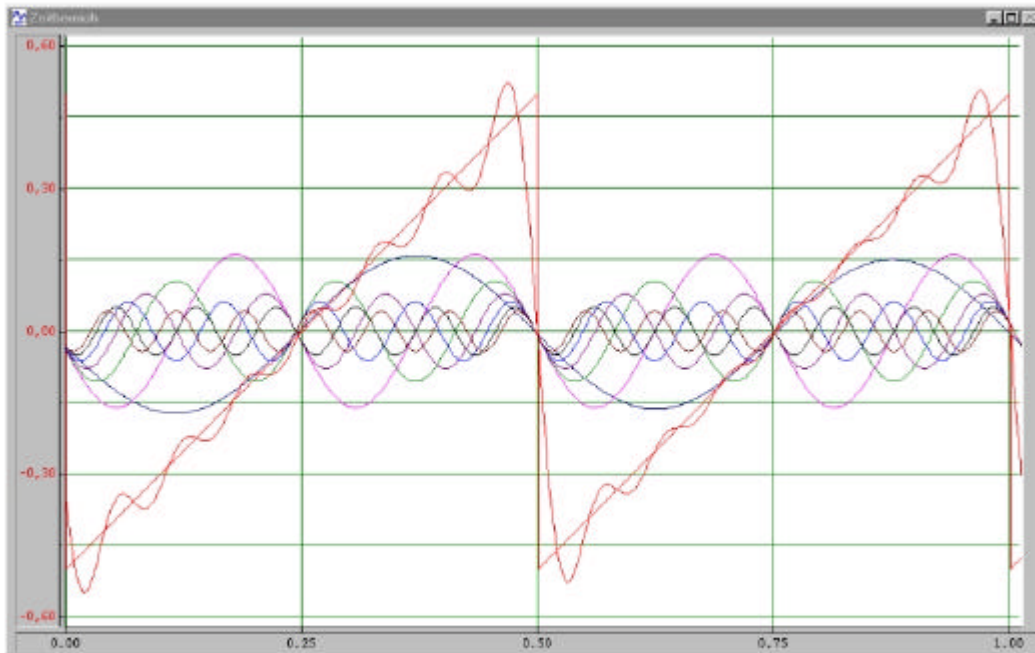


Abb. 2.1: In Sinusschwingungen zerlegtes Sägezahnsignal, vgl. Karrenberg, Die Welt der Signale-Prozesse-Systeme, 2000

Die rote wellige Linie neben dem Sägezahnsignal zeigt die Resultierende (Summenkurve) aus allen hier gezeigten Sinus-Signalen. Die allgemeine Formel für eine solche Fourierreihe lautet:

$$s(t) = b_0 + \sum_{k=1}^{\infty} a_k * \sin(2 * \mathbf{p} * f_0 * k * t) + \sum_{k=1}^{\infty} b_k * \cos(2 * \mathbf{p} * f_0 * k * t)$$

$s(t)$: periodisches Zeitsignal

f_0 : Grundfrequenz

b_0 : Grundanteil oder Offset einer Schwingung, also eine Konstante die keine Schwingung darstellt.

vgl. <http://www.phonetik.uni-muenchen.de/AP/APFourierVert.html>

Der Gleichanteil wird nach folgender Formel berechnet:

$$b_0 = \frac{1}{2 * \mathbf{p}} \int_0^{2 * \mathbf{p}} s(2 * \mathbf{p} * f_0 * t) \cdot dt$$

$(2 * \mathbf{p} * f_0 * t)$ ist das zu berechnende Zeitsignal.

a_k und b_k Diese Faktoren beschreiben den Schwingungsanteil der Harmonischen, also dem Vielfachen der Grundfrequenz (Faktor k).

Die Faktoren für die Harmonischen werden mit folgenden Formeln aus dem Zeitsignal $s(t)$ berechnet:

$$a_k = \frac{1}{2 \cdot p} \int_0^{2 \cdot p} s(2 \cdot p \cdot f_0 \cdot t) \cdot \sin(2 \cdot p \cdot k \cdot f_0 \cdot t) \cdot dt$$

$$b_k = \frac{1}{2 \cdot p} \int_0^{2 \cdot p} s(2 \cdot p \cdot f_0 \cdot t) \cdot \cos(2 \cdot p \cdot k \cdot f_0 \cdot t) \cdot dt .$$

Der Sachverhalt, dass jedes Signal aus Sinusschwingungen besteht, lässt auch den umgekehrten Weg zu. So kann man nicht nur Signale aus Sinusschwingungen modulieren, sondern auch ein Signal in seine Sinusschwingungen zerlegen. Diese Zerlegung in ein lineares Spektrum erfolgt durch die Zergliederung einer Zeitfunktion in eine Grundfrequenz und eine Serie von Harmonischen (= das Vielfache der Grundfrequenz.).

Die Frequenzkomponenten werden als Real- und Imaginärwertepaar oder als Betrags- und Phaseninformation (Polardarstellung) wiedergegeben. Ein Signal wird also vom Zeit- in den Frequenzbereich transferiert (Fourier Transformation). Siehe Abbildung 2.2

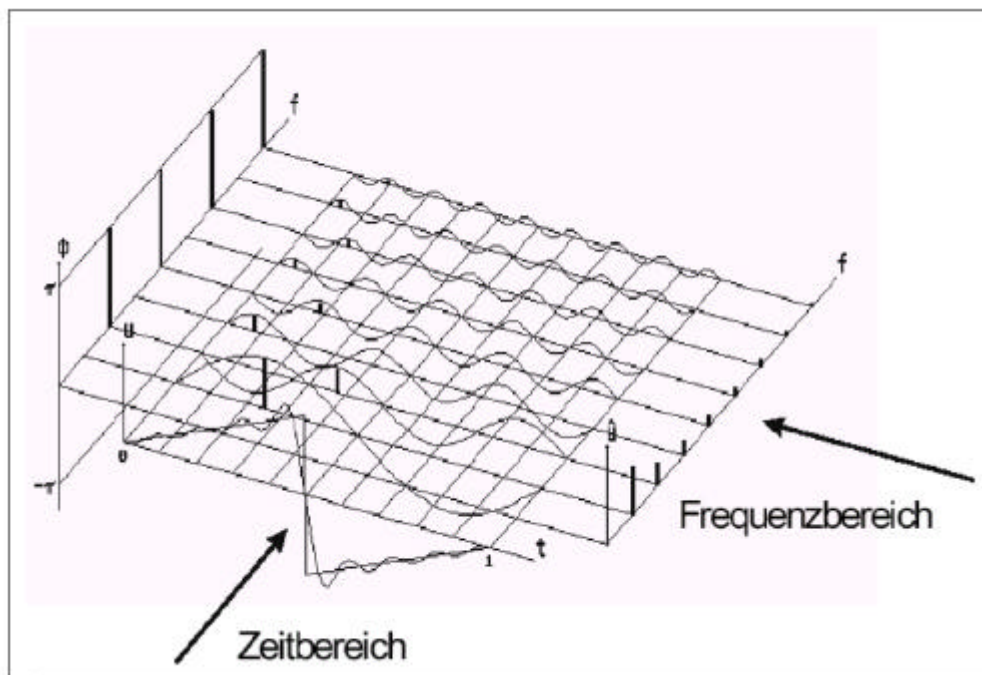


Abb. 2.2 Der Zeitbereich und der Frequenzbereich, vgl. Karrenberg, Die Welt der Signale-Prozesse-Systeme, 2000

Ein transformiertes Signal kann so auf seine Bestandteile (Schwingungen) untersucht werden. Die Fouriertransformation ist damit eine Grundlage, die es ermöglicht, viele Analysen von Signalen durchzuführen. So kann man z.B. feststellen, wie energiereich ein Signal (Schallintensität) ist und aus welcher Richtung es empfangen wird.

Die meiste Software, die im messtechnischen Bereich eingesetzt wird, bietet für eine Fouriertransformation eine Routine an, die FFT (Fast Fourier Transformation) heißt. Um eine FFT an einem Signal durchzuführen, müssen jedoch mehrere Dinge beachtet werden. Die Grundsätzlichen Merkmale, die Schwingungen betreffend, werden hier noch einmal kurz aufgeführt.

Die Frequenz, f einer Schwingung ist der Kehrwert der Periodendauer, T . $f = \frac{1}{T}$

Beispiel: $T=0,25s \Rightarrow f = \frac{1}{0,25s} \quad f = 4Hz$

Die Daten für eine FFT werden während der Aufzeichnung zu einem Block zusammengefasst. Ein Block wird auch als Fenster bezeichnet. Die Blöcke, die für die FFT eingelesen werden, können sich auch überschneiden, d.h., dass die Werte eines Blocks auch in dem darauffolgenden Block verwendet werden können. Man bezeichnet dieses als Überlappung oder auch „overlapping“. Eine Blocklänge von 1024 besagt, dass die Fouriertransformation über 1024 eingelesene Werte (Samples) durchgeführt werden soll. Die Daten werden mit einer Abtastfrequenz (auch als Abtastrate oder Samplerate bekannt) eingelesen, die an die Frequenz des einzulesenden Signals angepasst werden muss. Das Abtasttheorem, nach Shannon/Nyquist, besagt, dass ein Signal mindestens mit dem doppelten der Signalfrequenz abgetastet werden muss. Vgl. E.Oran Brigham FFT (Einführung in die Nachrichtentechnik) 1982.

$$f_{Signal_MAX} < \frac{f_{Sample}}{2}$$

In der Praxis werden auch Werte von $f_{Signal_Max} < f_{Sample}/2,2$ (CD-Norm) (vgl. Martin Mayer, Signalverarbeitung, 1998) oder $f_{Signal_Max} < f_{Sample}/2,56$ bei HP- und B&K Frequenzanalysatoren, verwendet. Der Wert 2,56 ist ein praktischer Erfahrungswert, der bei diesen, zuletzt genannten, Geräten zum Einsatz kommt. Die auch Aliasingfrequenz genannte Grenzfrequenz (Nyquistfrequenz) ist die Frequenz, bei der ein Signal noch frequenzgenau abgetastet werden kann. Wird ein Signal mit Frequenzanteilen, die oberhalb der Nyquistfrequenz liegen, abgetastet, so werden diese in Folge von Unterabtastung nicht richtig erkannt und vom Rechner falsch interpretiert. Aus der Blocklänge und der Abtastfrequenz ergibt sich noch die Frequenzauflösung, Δf_{FFT} , der FFT. Die Frequenzauflösung ist der Kehrwert von Δt_{FFT} .

$$\Delta f_{FFT} = \frac{1}{\Delta t_{FFT}}$$

Die Zeit Δt_{FFT} ist die, die benötigt wird, um einen „Block“ zu füllen. Sie ist abhängig von der Abtastfrequenz. Die Frequenzauflösung gibt an, wie fein die Auflösung der Frequenzanalyse ist. Die Frequenzauflösung Δf_{FFT} darf nicht mit der Abtastfrequenz f_{Sample} verwechselt werden!

Über die Abtastfrequenz ergibt sich auch die Bandbreite. Sie ist die Spanne bis zu der Frequenz, bei der die Signale noch richtig abgetastet werden.

$$\text{Bandbreite} = \frac{\text{Samplefrequenz}}{2.0}$$

Die Bandbreite ist in Linien eingeteilt. Die Anzahl der Linien ergibt sich aus der Bandbreite des Systems und der Frequenzauflösung Δf_{FFT} . Es ergeben sich bei einer Bandbreite von 1600Hz und einer Auflösung von 1 Hz =1600 Linien (Beim HP Analysator 1601 Linien, da die Null-Linie mitgezählt wird!). Die Daten des Signals werden zu einem Block zusammengefasst, da sie so von der FFT als periodisches Signal interpretiert werden können. Auch wenn das Signal nur in der Zeit der Abtastung scheinbar periodisch war, so wird es bei der FFT verarbeitet, als würde es unentwegt periodisch weiter laufen. Ist jedoch die Blocklänge so ausgefallen, dass sich zwischen der Amplitude des Signals am Blockanfang und der Amplitude am Blockende eine Differenz ergibt, oder dass die Funktion an dieser Stelle eine Unstetigkeit aufweist, so spricht man hier von einem Leakage Effekt. Tritt dieser Effekt auf, so wird die FFT das Signal nicht richtig interpretieren und ein falsches Frequenzspektrum ausgeben. Um dieses zu verhindern, bedient man sich der Fensterfunktionen, die dafür sorgen, dass die Werte am Blockanfang und Blockende gleich sind. Allerdings werden durch die Fensterfunktion die Ergebnisse hinsichtlich des Energieanteils auch verfälscht, was durch Korrekturwerte wieder behoben werden muss. Die Korrekturwerte sind von den jeweiligen Fensterfunktionen abhängig. Die folgende Abbildung 2.3 zeigt den Verlauf der Flat Top Fensterfunktion, wie sie von DASYS Lab verwendet wird.

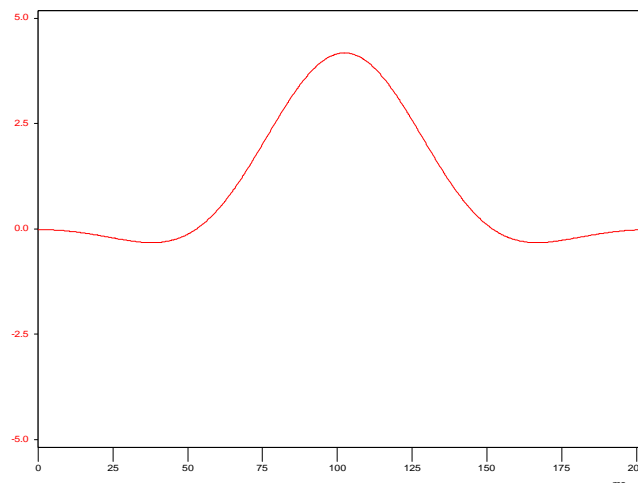


Abb. 2.3 Verlauf der Flat Top Funktion erstellt mit DASYS Lab

Diese Tabelle gibt einen Überblick über die Korrekturwerte für die Fensterfunktionen:

Fensterfunktion	Mögliche Parameter	Korrekturwert e
Bartlet (Dreieck)	Nein	1,328143
Hamming	Nein	1,362826
Hanning	Nein	1,5
cos hoch x	Werte für x	1,233702 (für x=1)
Blackman	Nein	1,726757
Blackman-Harris	Nein	2,004353
Tukey	Ja	1,482575 (für Param.=0)
Poisson	Ja	1,078642 (für Param.=1)
Hanning-Poisson	Ja	1,169676 (für Param.=1)
Cauchy	Ja	1,039319 (für Param.=1)
Gauss	Ja	1,017797 (für Param.=1)
Flat Top	Nein	3,428555

Tabelle 2.1

Die Daten dieser Tabelle wurden mit einem DASYLab Programm erzeugt, siehe „Korrekturwertermittlung.DSB“, welches den Effektivwert des ungewichteten Signals mit dem Effektivwert des gewichteten Signals vergleicht (siehe Anhang). Die Formel nach der e berechnet wurde ist folgende:

$$e = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n f(i)^2}}{\sqrt{\frac{1}{n} \sum_{i=1}^n f(x)^2}}$$

Worin $f(x)$ der jeweils ungewichtete Wert und $f(i)$ der jeweils mit einer Fensterfunktion gewichtete Wert ist.

Hanning und Flat Top sind die geläufigsten Fenster, wobei das Flat Top Fenster eine höhere Amplitudengenauigkeit hat und damit besser zum Kalibrieren geeignet ist, das Hanning Fenster hingegen ermöglicht eine höhere Frequenzgenauigkeit.

2.1 Analyseverfahren mittels der Frequenzanalyse

2.1.1 Fourierspektrum:

Die folgenden Kapitel zeigen die Analysemöglichkeiten die mittels FFT durchgeführt werden können.

Zum ersten gibt es das Fourierspektrum, es besteht aus den negativen und positiven Frequenzanteilen. Die FFT wandelt das Signal mit einem Algorithmus von der Zeitebene in die Frequenzebene um.

$$f_x(t) \rightarrow S_x(\omega)$$

Das Ergebnis der Fouriertransformation ist gemäß folgender Formel komplex, d.h., es besteht aus realen und imaginären Werten. Die Formel für den kontinuierlichen Verlauf der FFT lautet.

$$S_x(\omega) = \int_{-\infty}^{\infty} f(t) \cdot e^{i\omega t} dt$$

Die Formel für die diskrete FFT ergibt sich aus der für den kontinuierlichen Verlauf.

$$S_x(\omega) = \sum_{n=0}^{N-1} f_n(t) \cdot e^{i\omega n t} \quad \text{oder} \quad S_x(f) = \sum_{n=0}^{N-1} f_n(t) \cdot e^{i2\pi f n / N}$$

Der Parameter N steht für die Anzahl der Werte, die im Block zusammengefasst sind (Blocksize) und das $f_n(t)$ steht für die jeweilige Frequenz im Frequenzspektrum.

Folgende Abbildung zeigt ein Fourierspektrum, welches sich über den gesamten positiven und negativen Frequenzbereich erstreckt.

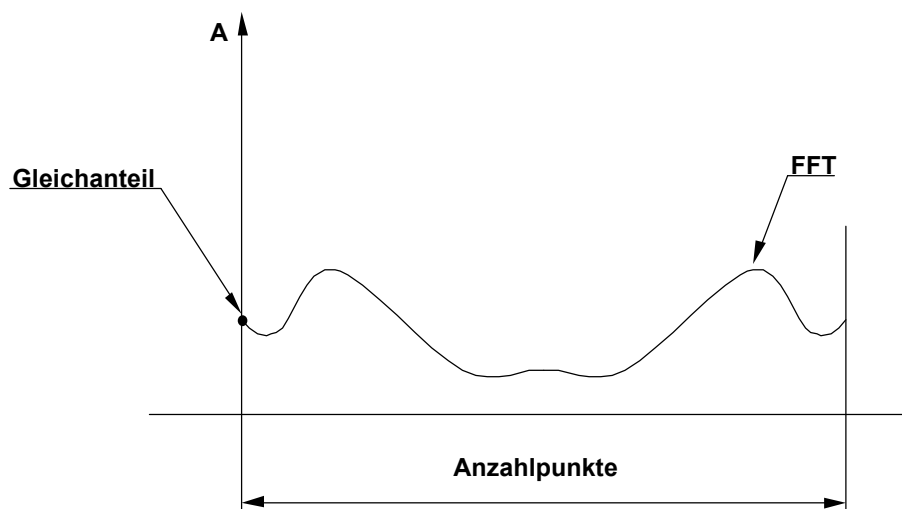


Abb. 2.4 Fourierspektrum

vgl. Younes Chlyeh, Numerische Simulation und experimentelle Überprüfung von Wanddruckschwankungsverläufen, 2000

Werden die negativen und positiven Frequenzen sortiert, so ergibt sich bei reellen Signalen ein an der Nyquistfrequenz spiegelnder Frequenzgang. Siehe Abbildung 2.5. An einem Frequenzanalysator werden daher immer nur die positiven Frequenzanteile dargestellt. Die Amplituden der gespiegelten Anteile, ohne die 0-Hz Linie, werden mit dem Faktor 2 multipliziert.

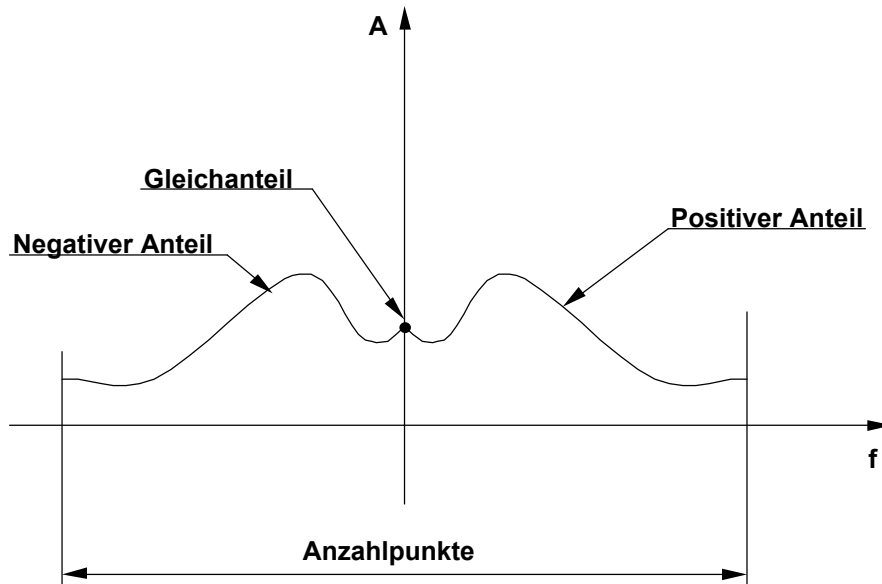


Abb. 2.5 Symmetrisches Fourierspektrum

vgl. Younes Chlyeh, Numerische Simulation und experimentelle Überprüfung von Wanddruckschwankungsverläufen, 2000

Wird der Wert des Frequenzspektrums S_x mit dem konjugiert komplexen Wert \hat{S}_x multipliziert und durch die Anzahl der Abtastungen der FFT dividiert, so erhält man das Leistungsspektrum. Vgl. Rahman Jamal u. Herbert Pichlik, LabVIEW Programmiersprache der vierten Generation, 1997.

$$P_x = \frac{S_x * \hat{S}_x}{n_{FFT}} \quad [V^2]$$

2.1.2 Amplitudenspektrum:

Zieht man aus diesem Leistungsspektrum die Quadratwurzel, so erhält man das Amplitudenspektrum.

$$G_x = \sqrt{\frac{S_x * \hat{S}_x}{n_{FFT}}} \quad [V]$$

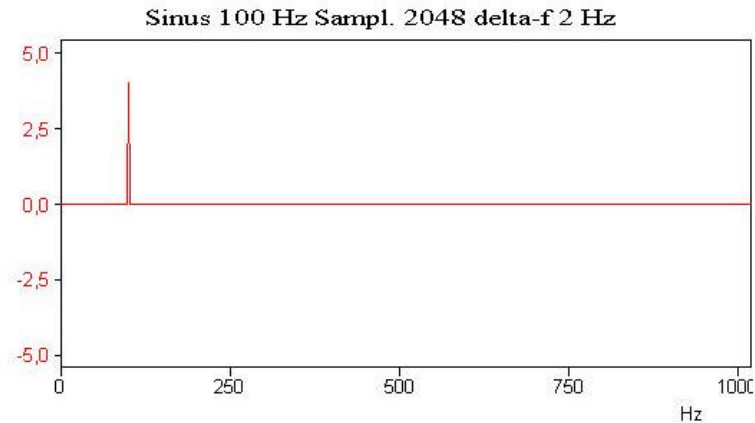


Abb. 2.6 Amplitudenspektrum eines Sinussignals mit 100 Hz und einer Amplitude von 4 V

Soll ein gemittelttes Amplitudenspektrum erzeugt werden, so kommt folgende Formel zum

Einsatz:

$$\overline{G_x} = \sqrt{\frac{1}{Z} \sum_{n=1}^Z G_{xn}^2} \quad [V].$$

Die unter DasyLAB berechneten Amplituden des Spektrums entsprechen den sogenannten Spitzenwerten (peak). Soll nun der rms-Wert (rms steht für root mean square und beschreibt den Effektivwert) für jede einzelne Amplitude gebildet werden, so muss dieser nach folgender

Formel berechnet werden:

$$G_{x(rms)} = \frac{G_x}{\sqrt{2}} \quad [V_{rms}].$$

2.1.3 Der Gesamtpegel:

Nach dem Theorem von Parseval hat ein Signal über einen bestimmten Zeitraum, z.B. einen Block einer FFT, im Zeitbereich sowie im Frequenzbereich den gleichen Energiegehalt.

Der Gesamtpegel oder Effektivwert ist ein Einzahlenwert einer Schwankungsgröße, der aus den Amplituden aller Frequenzkomponenten berechnet wird. Für den Zeitbereich wird der Effektivwert über das quadratische Mittel gebildet:

$$GP(t) = \sqrt{\frac{1}{N} \sum_{i=1}^N A(t)_i^2}.$$

Die Formel für den Gesamtpegel im Frequenzbereich ist folgendermaßen definiert:

$$GP = \sqrt{\frac{1}{e} \sum_{i=1}^N A_i^2}.$$

Sie besagt, dass die Summe aus allen quadrierten Amplituden der Frequenzkomponenten über die Anzahl der Stützstellen der FFT gebildet wird und dieser Wert durch den Korrekturwert der Fensterfunktion dividiert wird. Danach muss noch die Quadratwurzel aus dem sich ergebenden Wert gezogen werden.

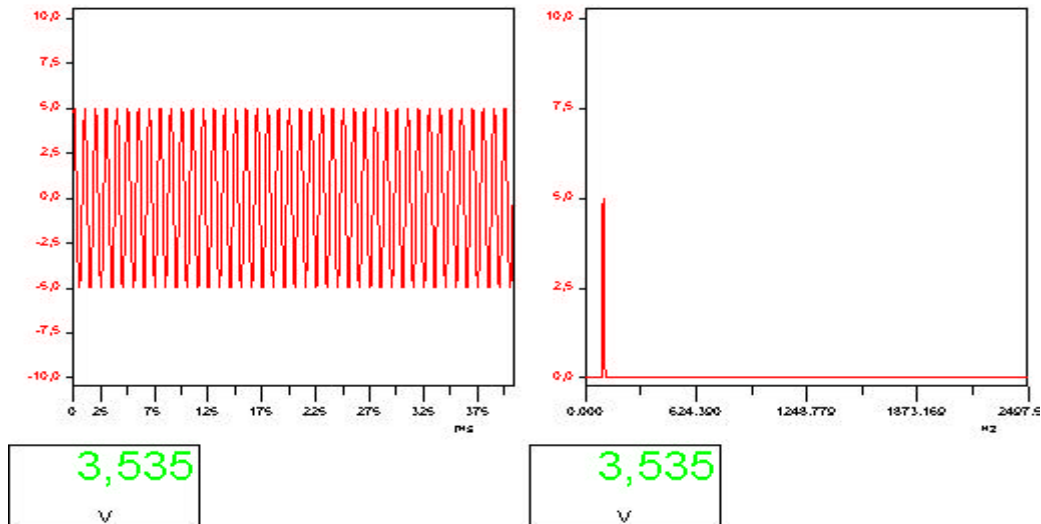


Abb. 2.7 Effektivwert im Zeit- und Gesamtpegel im Frequenzbereich erzeugt mit DASyLab, vgl. Programm: Gesamtpegel.DSB

Diese Grafik zeigt ein Signal mit 100 Hz und einer Amplitude von 5 Volt im Zeitbereich (links) und im Frequenzbereich (rechts) mit den jeweiligen Gesamtpegeln darunter.

2.1.4 Phasenspektrum:

Aus den reellen und komplexen Anteilen des Fourierspektrums lässt sich die Phase der jeweiligen Sinusschwingung berechnen, indem man den Imaginärteil durch den Realteil dividiert und anschließend den Arcustangens aus diesem Wert bildet. Mit dieser Berechnung erhält man das Phasenspektrum.

$$j = \arctan \frac{\text{Im}(S_x)}{\text{Re}(S_x)}$$

2.1.5 Kreuzleistungsspektrum:

Das Kreuzleistungsspektrum, auch Cross Power Spectrum (CPS) genannt, wird auf folgende Weise aus zwei Signalen gebildet. Man nehme an, dass ein Signal x und ein Signal y fouriertransformiert wurde.

$$CPS_{xy} = \frac{S_x * \hat{S}_y}{nfft}$$

$nfft$ ist die Anzahl der Stützstellen des Blocks (Blocksize).

Wobei \hat{S}_y der konjugiert komplexe Wert von S_y ist. Um den zeitlichen Mittelwert des Kreuzleistungsspektrums zu erhalten, muss das arithmetische Mittel gebildet werden. Durch genügend häufige Mittelungen werden die Signalanteile herausgefiltert, die nur in einem der beiden Signale vorhanden sind und es werden die Signalanteile hervorgehoben, die in beiden Signalen vorhanden sind.

$$\overline{CPS} = \frac{1}{n} \sum_{i=1}^n CPS_i$$

Aus dem Kreuzleistungsspektrum lassen sich die Magnitude (Amplitude) und die Phase berechnen.

$$Mag(CPS) = \sqrt{\text{Re}^2 + \text{Im}^2}$$

$$Phase(CPS)\mathbf{j} = \arctan \frac{\text{Im}(CPS)}{\text{Re}(CPS)}$$

Mit dem zeitlich gemittelten Kreuzleistungsspektrum lassen sich ebenso die gemittelte Amplitude und Phase berechnen.

2.1.6 Kohärenz

Die Kohärenzfunktion gibt einen statistischen Zusammenhang der beiden Signale im Frequenzbereich an. Die Kohärenzfunktion \mathbf{g}^2 ist reell und kann nur Werte zwischen 0 und 1 annehmen. Die Kohärenz wird mit folgender Formel berechnet.

$$\mathbf{g}^2 = \frac{\overline{S_x \hat{S}_y} * \overline{S_y \hat{S}_x}}{\overline{S_x \hat{S}_x} * \overline{S_y \hat{S}_y}} = \frac{\overline{CPS_{xy}} * \overline{CPS_{yx}}}{\overline{APS_x} * \overline{APS_y}}$$

Wichtig ist, dass mit gemittelten Kreuzleistungsspektren bzw. Leistungsspektren gerechnet wird, da sich ansonsten immer der Wert 1 für die Kohärenz ergibt, da sich die einzelnen Komponenten der Leistungsspektren und der Kreuzleistungsspektren herauskürzen würden.

3. Umsetzung mit DASyLab

DASyLab ermöglicht die Realisierung von Messinstrumenten als Virtuelle-Geräte. Die Instrumente werden aus Modulen zusammengesetzt, die von DASyLab vorgegeben werden. Die Module werden in einem Schaltbild zusammengefügt und für die Bedienung können ein oder mehrere Layouts mit Verknüpfungen zu den Bedien- und Ansichtselementen gebildet werden. Die folgende Umsetzung eines Zweikanal-Frequenzanalysators zeigt, welche Module zum Einsatz kommen und was bei deren Einsatz an Parametern und Einstellungen zu beachten ist.

3.1 Erfassen der Messdaten

Um die Messdaten zu erfassen, kann z.B. das Modul „A/D Wandler“ (Analogeingang) verwendet werden, über das die Soundkarte angesprochen wird oder, wie in diesem Programm vorhanden, der Analogeingang des NI-DAQ Treibers, mit dem eine Messkarte angesprochen werden kann. Die NI-DAQ Treiber werden von der Firma National Instruments zu deren Messkarten mitgeliefert. Die Messkarte, die hier zum Einsatz kommt ist eine AT-MIO-16E-10 der Firma National Instruments. Bei dieser Karte handelt es sich um eine 16 Kanal Messkarte mit einer max. Summenabtastrate von 100 kHz. Diese Karte kann bis zu 16 Kanäle im single-ended, oder 8 Kanäle im differential Mode, mit einer Abtasttiefe von 12 bit, erfassen. Diese Messkarte verfügt über keinen Antialiasingfilter.



Abb. 3.1 Modul für den Analogeingang der Messkarte

Im Menü für den Analogeingang lassen sich Kanal und Eingangsspannung auswählen. Bei dem Analogeingang für die Soundkarte ist zudem eine Skalierfunktion für die Kalibrierung des Eingangs vorhanden.

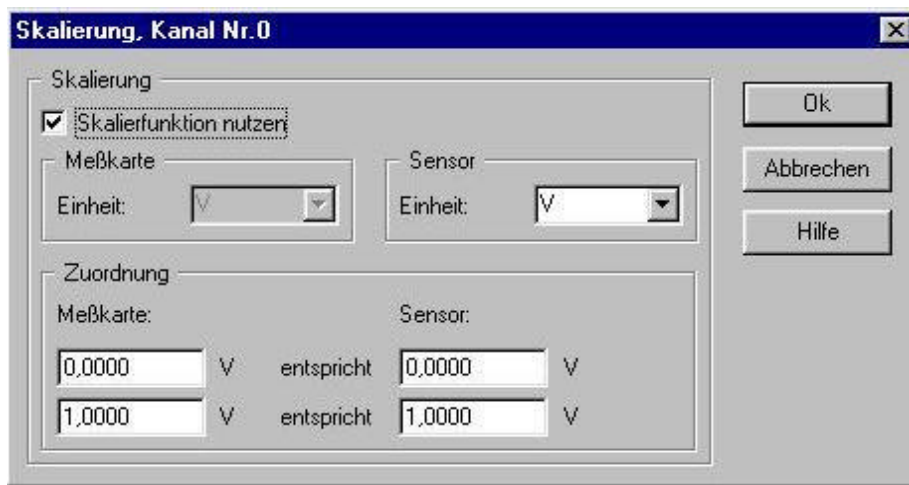


Abb.3.2 Skalierfunktion des Soundkarten-Eingang

3.2 Datenfenster

Als nächstes werden die Signale kanalweise mit einer Fensterfunktion gewichtet. Bei diesen Fensterfunktionen werden am häufigsten, wie unter Punkt 2 beschrieben, das Hanning- und das Flat Top Fenster verwendet.

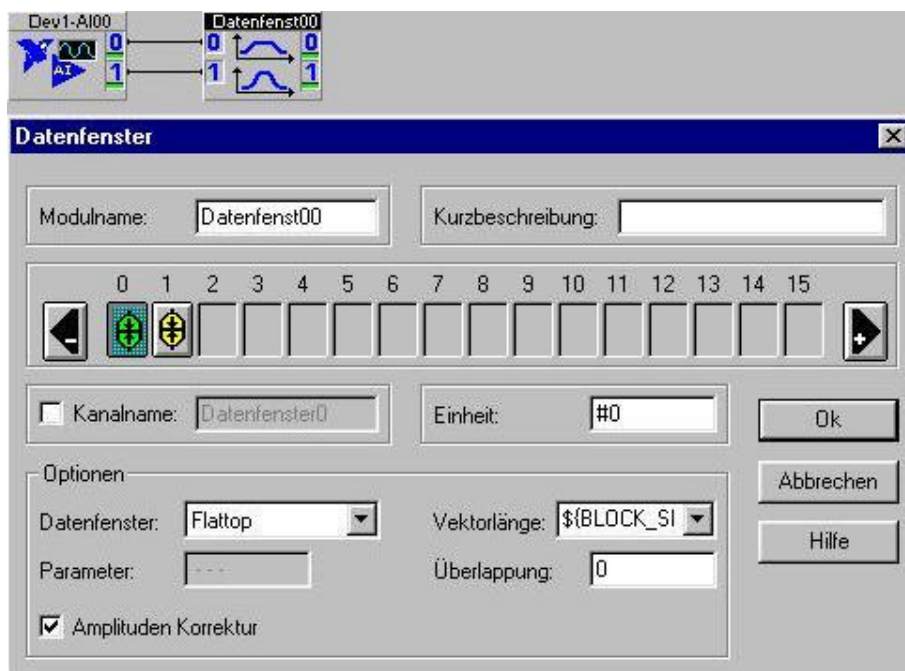


Abb. 3.3 Fensterfunktion

Das Menü für die Fensterfunktion läßt neben der Auswahl der Fensterfunktion noch die Angabe der Länge des Datenbereichs, über den die Fensterfunktion angewendet werden soll, zu. Des weiteren können hier ggf. noch die Länge einer Überlappung und, wenn notwendig, die Parameter für eine Fensterfunktion angegeben werden. Im Menü für das Datenfenster ist noch die Option der Amplituden Korrektur vorhanden.

Wie sich diese auf ein Signal auswirkt, zeigt folgende Abbildung.

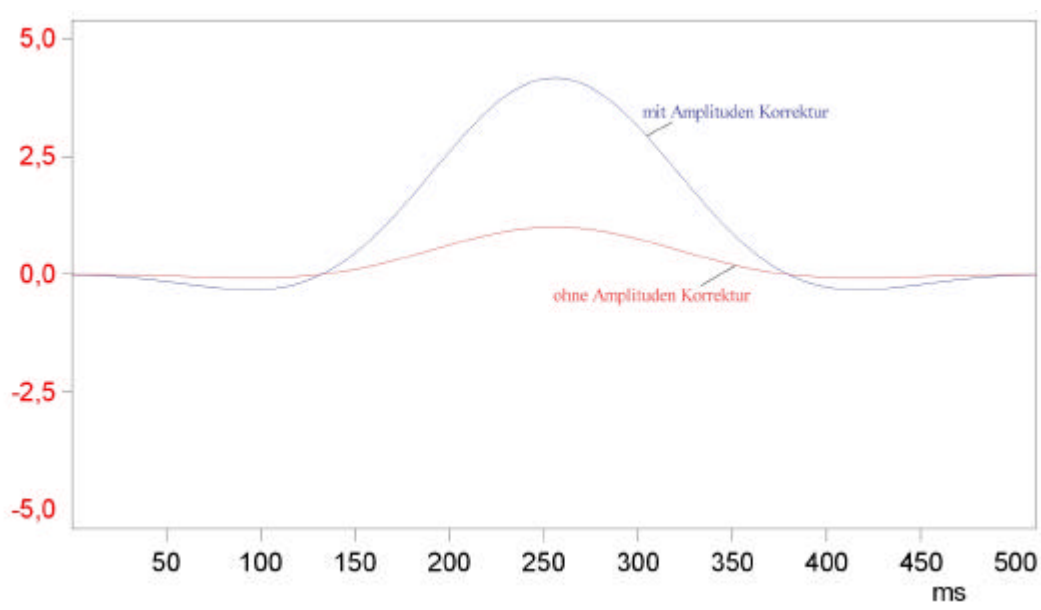


Abb.3.4 Der blaue Verlauf ist mit, der rote ohne Amplituden Korrektur

Eine Konstante wurde hier mit einem Fenster ohne Amplituden Korrektur gewichtet und eine weitere mit einem Fenster mit Amplituden Korrektur. Es wurde nicht dokumentiert was durch die Amplitudenkorrektur bewirkt wird. Durch Versuche zeigte sich, dass durch die Amplitudenkorrektur die Normierung des Fensters durchgeführt wird, (das Häkchen muss für die Programmierung eines Frequenzanalysators gesetzt sein, damit das Fenster nicht normiert ist !).

3.3 Fouriertransformation der Signale

Anschließend wird die Fouriertransformation durchgeführt. Ein Auswahlmeneu der Fouriertransformation bietet folgende Möglichkeiten.



Abb.3.5 Auswahl der Fouriertransformationen

Zunächst kommt die „Reelle FFT eines reellen Signals mit Bewertung“ zum Einsatz, denn ein Signal, welches z.B. mit einem Mikrophon aufgezeichnet wurde, ist ein reelles Signal. Das Menü für die FFT bietet neben der reinen Fourier Analyse noch das Amplituden-Spektrum, das Leistungs-Spektrum, das Leistungsdichte-Spektrum und das Phasenspektrum. Zudem ist noch eine dB Bewertung des Signals möglich. Der Bezugswert kann in das dafür vorgesehene Feld eingetragen werden. Auf die im Menü aufgeführte Möglichkeit der Filterung wird hier nicht weiter eingegangen.

Da zunächst das Amplituden-Spektrum erzeugt werden soll, wird die entsprechende Auswahl im Menü getroffen. Siehe Abbildung 3.6 .

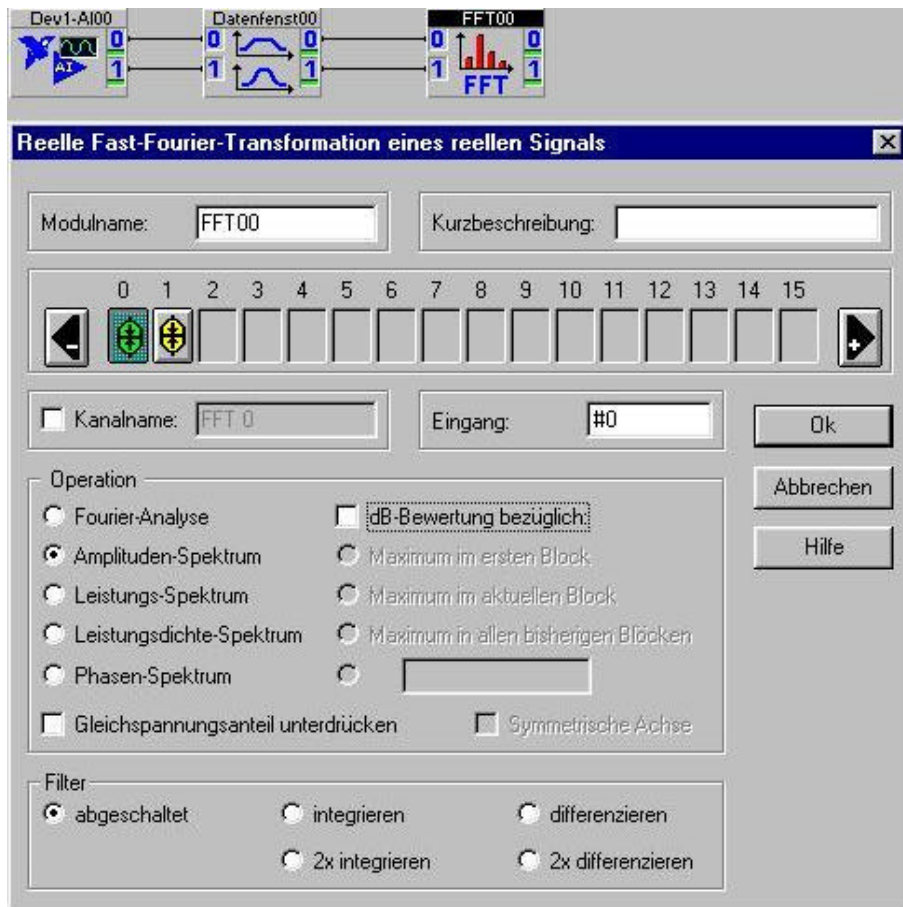


Abb. 3.6 Auswahlmenü der Fouriertransformation

3.4 Diagramm des Amplitudenspektrums

Das Amplituden-Spektrum wird mit einem Modul „X/t Grafik“ angezeigt. Im Menü der X/t Grafik können Skalierwerte und Anzeigebereiche (Zoom Bereiche) angegeben werden.



Abb. 3.7 Menü des X/t Grafikfensters

Die folgende Grafik zeigt das Amplituden-Spektrum einer Sinus-Schwingung mit 100 Hz und einer Amplitude von 2 V, die Amplitude im Amplitudenspektrum hat eine lineare Einheit. Das Signal wurde bei einer Blocksize von 1024 mit 2048 Hz abgetastet. Der Wert für Δ_f betrug damit 2 Hz.

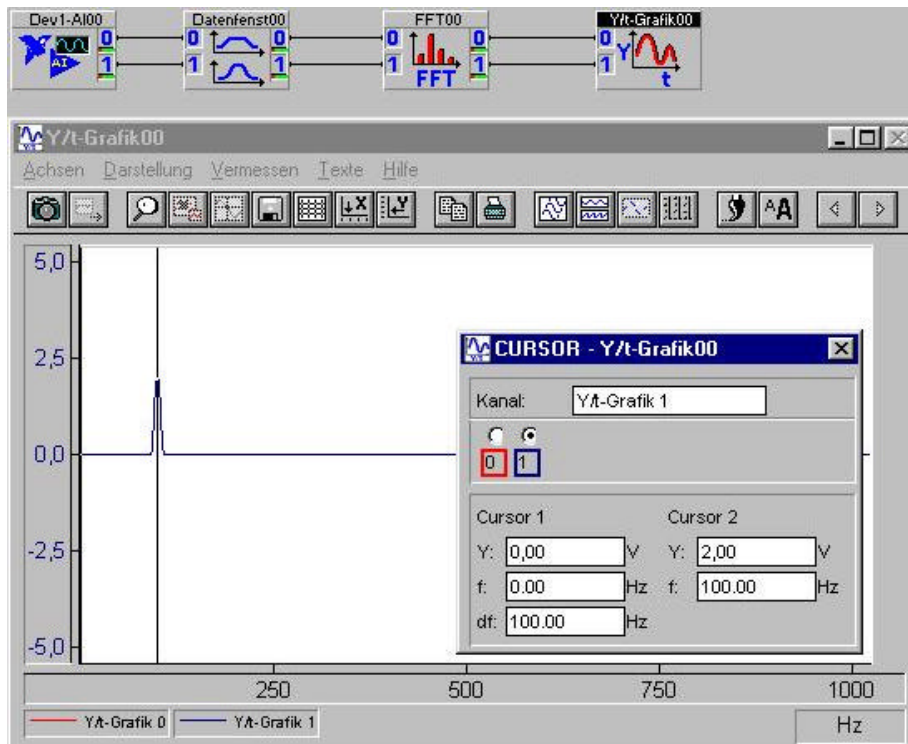


Abb. 3.8 Amplitudenspektrum: Sinus: 100Hz Blocksize: 1024 f_s : 2048Hz Δ_f : 2Hz

3.5 Diagramm des Leistungsspektrums

Um das einseitige Leistungsspektrum zu erhalten muss man das Amplitudenspektrum quadrieren, dadurch ergibt sich im Leistungsspektrum eine quadratische Einheit für die Amplitude. DASyLab bietet hier auch, über das Modul „Reelle FFT eines reellen Signals“, die Möglichkeit, ein Leistungsspektrum auszugeben. Jedoch wird dieses durch Quadrieren der positiven Werte des zweiseitigen Fourierspektrums gebildet. Dadurch sind die Werte in diesem Leistungsspektrum um den Faktor 4 kleiner, als im über das Amplitudenspektrum erzeugte, Leistungsspektrum.

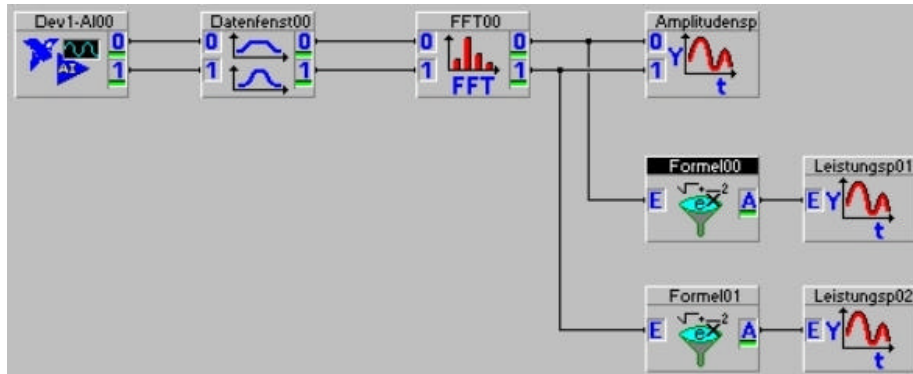


Abb 3.9 Schaltbild für das Leistungsspektrum

Mittels des Moduls „Formelinterpret“ (Formel 00 & Formel 01) werden die Amplitudenspektren je Kanal quadriert, (Achtung: sqr = quadrieren sqrt = Wurzel).

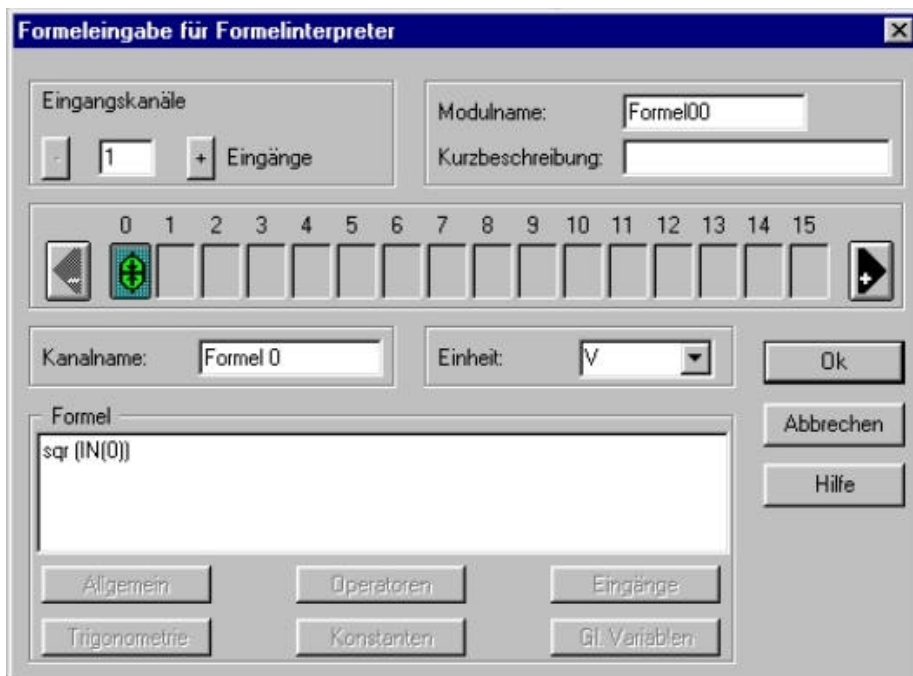


Abb. 3.10 Formelinterpret für das Leistungsspektrum

Die folgende Grafik zeigt die beiden Leistungsspektren. Als Signal wurde wieder ein Sinus mit 100 Hz generiert. Die Amplitude betrug 2 V. Die Blocksize betrug 1024 bei einer Abtast-rate von 2048 und einem $\Delta f=2\text{Hz}$.

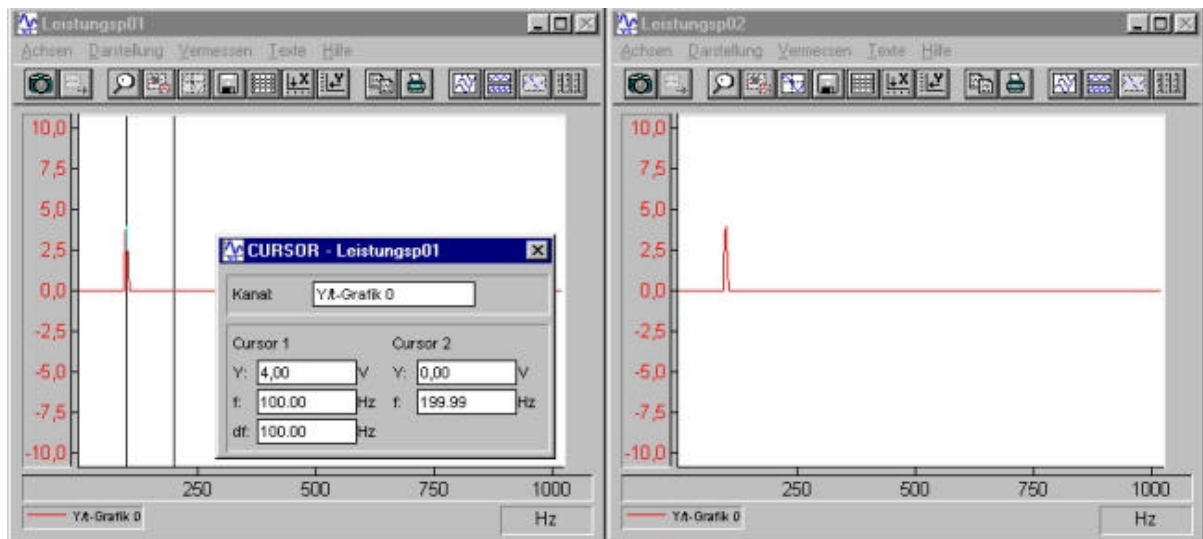


Abb. 3.11 Leistungsspektrum: Sinus: 100Hz Blocksize: 1024 f_s : 2048Hz Δf : 2Hz

3.6 Das Phasenspektrum

Ein Phasenspektrum kann die Lage eines Sinussignals zu einem Referenz-Zeitpunkt (Startpunkt oder Triggersignal) angeben, d.h., die Lage zweier Signale zueinander, die verglichen werden sollen. Die Lage der Phase kann in Grad (360° bilden einen Vollkreis) oder in Radiant ($2 \text{ Rad} = 2 \cdot \pi$ bilden einen Vollkreis) angegeben werden. Das Phasenspektrum könnte aus dem Real- und Imaginärteilen des Kreuzspektrums (FFT Modul) berechnet werden, jedoch bietet DASYLab mit dem Sondermodul Übertragungsfunktion einen komfortableren Weg, der die in der Regel immer notwendige Mittelung gleich mit einschließt. Dieses Sondermodul kann folgende Funktionen vorweisen:

1. Übertragungsfunktion mit den Funktionen
 - 1.1 $H_{xy}(f) = F_y(f)/F_x(f)$
 - 1.2 $H_{xy}(f) = F_{xy}(f)/F_{xx}(f)$
 - 1.3 $H_{xy}(f) = F_{yy}(f)/F_{xy}(f)$
2. Kohärenzfunktion
3. $F_{xx}(f)$ Leistungsspektrum des X-Signals
4. $F_{yy}(f)$ Leistungsspektrum des Y-Signals
5. $F_{xy}(f)$ Kreuzleistungsspektrum

Die Übertragungsfunktionen 1.2 und 1.3 erwiesen sich als besser geeignet, da sie weniger Störungen im Phasenverlauf zeigten als die Übertragungsfunktion 1.1. Woher diese Störungen herrühren, konnte hier nicht geklärt werden. Im Übertragungsfunktion-Modul ist zudem die Art der Mittelungen einstellbar, die für die Messung verwendet werden soll. Dieses ist wiederum ein großer Vorteil gegenüber dem Zusammenschalten einzelner Module.

Die Abbildung 3.12 zeigt das Menü des Übertragungsfunktion Moduls. Hier sei noch darauf hingewiesen, dass neben der Ausgabe als Phase und Amplitude auch eine Ausgabe von Realteil und Imaginärteil möglich ist.

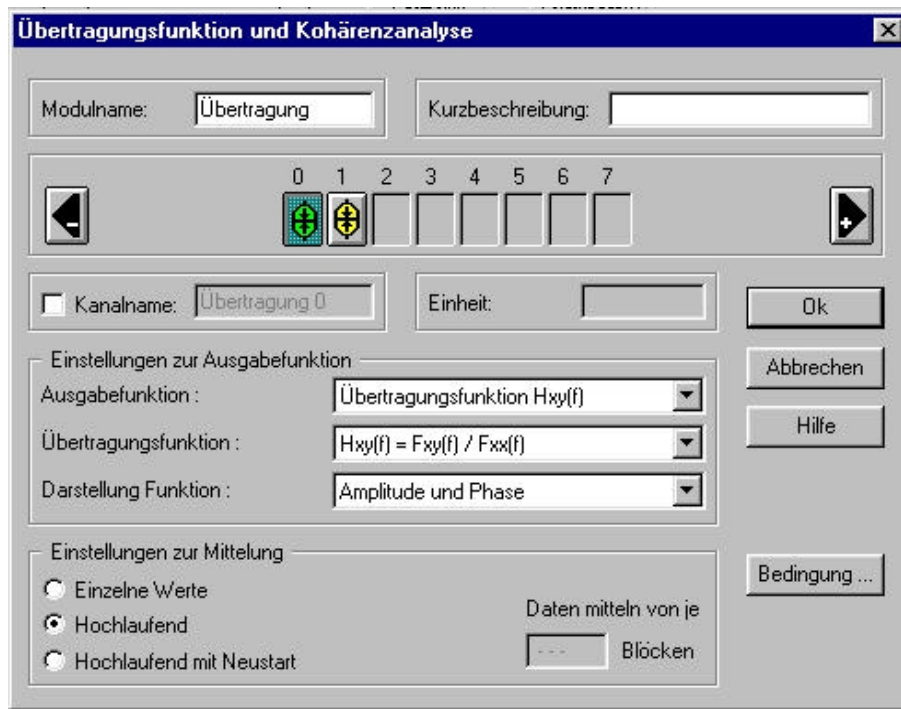


Abb. 3.12 Menü des Übertragungsfunktionen Moduls

Die Darstellung des Amplitudenspektrums oder des Phasenspektrums geschieht im Diagramm als zweiseitiges Spektrum, d.h., es werden ab der Nyquistfrequenz die Amplituden- und Phasenwerte für die negativen Frequenzen angegeben. Die Phase kann auch über das Kreuzleistungsspektrum (5.) des Übertragungsfunktion-Modul bestimmt werden. Auch hier geschieht die Ausgabe wieder für den positiven und negativen Frequenzbereich, was bei der Betrachtung von reellen Signalen nicht sinnvoll ist.



Abb. 3.13 Ausgabe von Phase und Amplitude über das Kreuzleistungsspektrum

Natürlich kann man, wie bereits erwähnt, die Phase auch über das Modul „FFT“ → „Kreuzspektrum zweier reeller Signale“ bestimmen, indem man die Phase wie unter 2.1.5 berechnet.

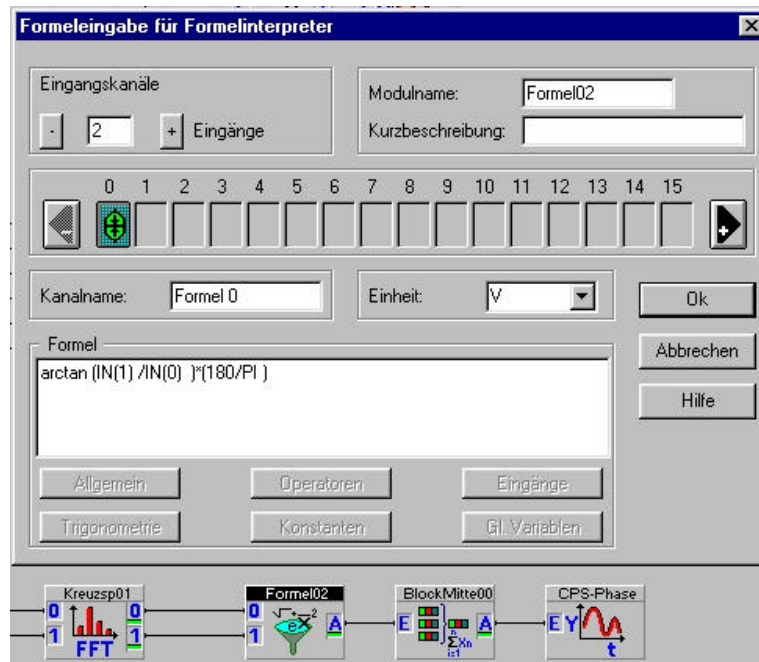


Abb. 3.14 Berechnung der Phase

Jedoch muss hier berücksichtigt werden, dass die Phase nur für Winkel von $\pm 90^\circ$ richtig berechnet wird. Für Phasen über $\pm 90^\circ$ wäre ein größerer Programmieraufwand nötig, der die Anteile des komplexen Signals in den entsprechenden Quadranten im Zeigerdiagramm berücksichtigt.

3.7 Kohärenz

Die Kohärenzfunktion (2.) gibt eine Aussage über den quantitativen Zusammenhang der beiden Signale, wie unter 2.1.5 beschrieben. Wie das Phasenspektrum wird auch die Kohärenz in einem zweiseitigen Spektrum ausgegeben. Ein idealer Kohärenzverlauf hat über den gesamten Frequenzbereich den Wert 1.

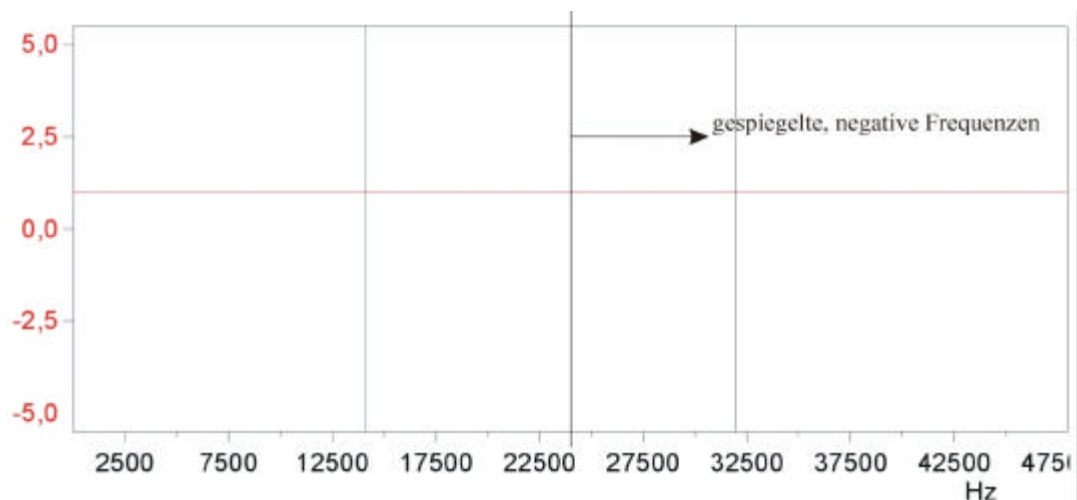


Abb.3.15 idealer Verlauf der Kohärenz

Das Schaltbild sieht nun folgendermaßen aus.

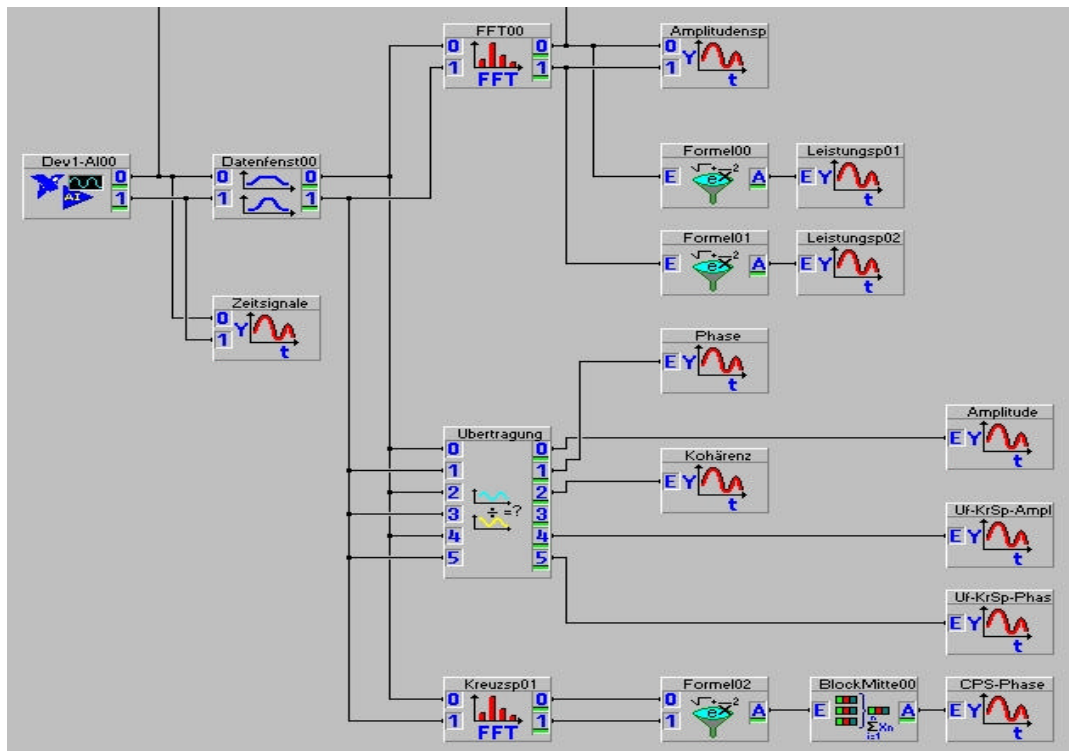


Abb. 3.16 Schaltbild

3.8 Der Gesamtpegel

3.8.1 Der Effektivwert im Zeitbereich

Der Gesamtpegel dient dazu, wie unter 2.1.3 beschrieben, den Energiegehalt eines Signals zu bestimmen. Für den Zeitbereich spricht man besser von Effektivwert, er kann mit dem Modul: „Mittlung“ berechnet werden. Abbildung 3.17 zeigt das Menüfenster dieses Moduls.



Abb. 3.17 Menüfenster des Moduls: Mittlung

Als Operation wird das Quadratische Mittel benötigt. Um über die gesamte Blocklänge einen gemittelten Wert zu erhalten, muss in das Feld für die Anzahl der zu mittelnden Werte die Blocklänge (Blocksize) eingetragen werden. Hier kann auch die Globale Variable für die Blocksize eingetragen werden. Darüber hinaus kann die Blocklänge am Ausgang auf 1 gesetzt werden. Siehe auch Anhang S.61 und Programm: „Mittelung.DSB“. Die Ausgabe kann in einer Liste erfolgen oder, wie hier, in einem Digitalinstrument.

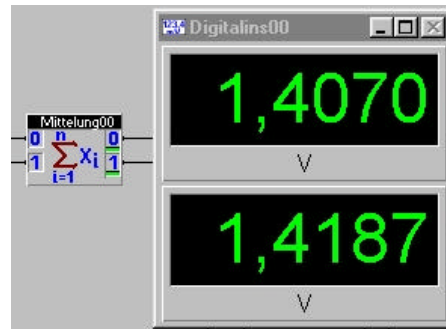


Abb. 3.18 Digitalinstrument für den Gesamtpegel

Es kann aber auch die Option „Gleitend“ oder „Hochlaufend“ verwendet werden. Allerdings geben diese Verfahren am Ausgang immer die Anzahl an Werten aus, die der Blocklänge entspricht, von denen jeweils nur der zuletzt ausgegebene Wert im Digitalinstrument erscheint. Dieser (letzter) Wert entspricht nicht immer dem Effektivwert. Je nach Anwendung können auch die Module „Statistische Werte“ oder „Blockmittelung/Peak Hold“ verwendet werden um gemittelte Werte zu erhalten. Hierbei muss aber wiederum auf die Art der Mittelung und die Form der Ausgabe geachtet werden.

3.8.2 Der Gesamtpegel im Frequenzbereich

Für die Berechnung des Gesamtpegels im Frequenzbereich sind mehrere Module notwendig, mit denen die Formel, wie unter 2.1.3 beschrieben, umgesetzt werden kann. Zuerst müssen die Amplituden des Spektrums quadriert werden, dazu können die für die Leistungsspektren bereits vorhandenen „Formelinterpretier“ verwendet werden. Anschließend werden die Werte summiert, dieses geschieht mit dem Modul „Integration“.



Abb. 3.19 Menü für die Ableitung/Integration

Dieses Modul (Integration) wird verwendet, weil das Modul „Arithmetik“ mit der Operation: „Aufsummieren“, kontinuierlich aufsummiert. Dieses ist damit nicht in der Lage nur über die jeweilige Blocklänge zu summieren.

Die Integration beinhaltet, gemäß Formel und Beispielprogramm im Anhang S.61 hier eine Aufsummierung über jeweils einen Block. Dieses Modul bildet aus der Summe der Amplituden, multipliziert mit dem Abstand Δf der Samples, eine Fläche. Um aus dieser Fläche nur die Summe aller Amplituden zu erhalten ist es notwendig, diese wiederum durch Δf zu dividieren.

Da $\Delta t = \frac{1}{\Delta f}$ ist, kann hier auch mit Δt multipliziert werden, um das Integral durch Δf zu dividieren.

Das Δt kann normalerweise durch Division der Block_Size durch die Abtastfrequenz ermittelt werden, jedoch arbeitet DASYLab bei der Globalen Variable Sample_Rate mit der Zeit, die zwischen den einzelnen Sample Vorgängen vergeht. Um Δt zu berechnen, muss folgende

Formel verwendet werden:
$$\Delta t = \frac{Block_Size * Sample_Rate}{1000}$$

Diese Formel fließt so in die Berechnung für den Gesamtpegel ein. Als nächstes werden die Summen im Formelinterpreter durch den Korrekturwert der Fensterfunktion dividiert, daraufhin wird die Wurzel gezogen und anschließend wird dieser Wert durch die Wurzel aus 2 dividiert, um den Effektivwert zu erhalten.

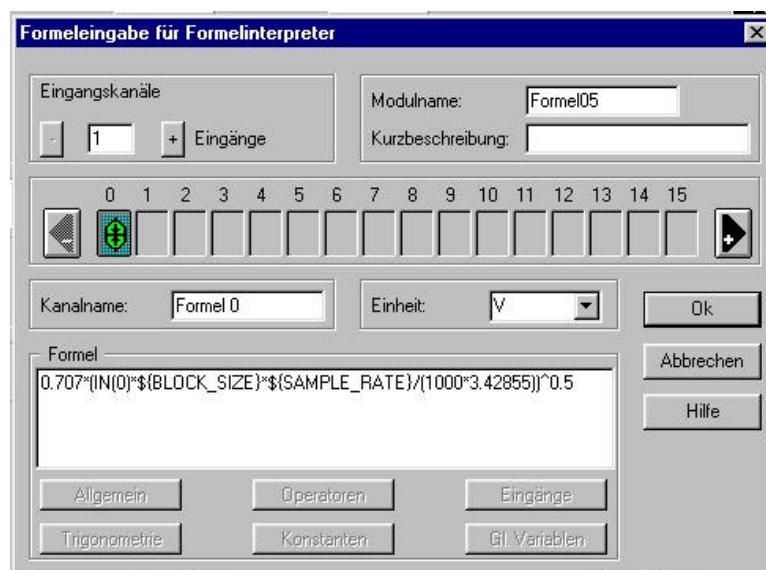


Abb. 3.20 Formel für den Gesamtpegel im Frequenzbereich

Nach dem Formelinterpreter folgt noch ein Modul zum Mitteln der Werte, hier das Modul „Blockmittelung“ und ein „Digitalinstrument“ zur Ausgabe.

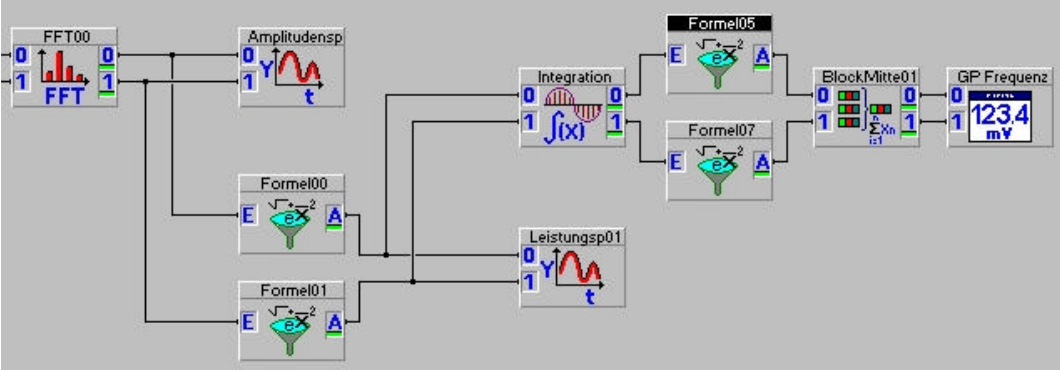


Abb. 3.21 Schaltbild für den Gesamtpegel im Frequenzbereich

4. Umsetzung mit LabVIEW

Wie DASyLab verfügt auch LabVIEW über ein Schaltbild und ein Layout (Bedienung und Anzeige), die aber unter LabVIEW Diagramm und Frontpanel genannt werden. LabVIEW bietet wie DASyLab die Möglichkeit, ein Programm aus Modulen zusammen zu setzen. Diese Module heißen unter LabVIEW Vi, was für Virtuelle Instrumente steht. Ein solches Vi kann wiederum auch aus Vi's bestehen, man spricht dann von SubVi's. Es ist ein großer Vorteil von LabVIEW, dass jedes Vi bis in seine unterste Struktur eingesehen und nachvollzogen werden kann. Zudem können SubVi aus Vi's heraus kopiert werden und in anderen Diagrammen eingefügt werden. Die gesamten Funktionen bzw. Vi's die von LabVIEW zur Verfügung gestellt werden, sind in der Funktionenpalette aufgeführt. Abbildung 4.1 zeigt diese Funktionenpalette.

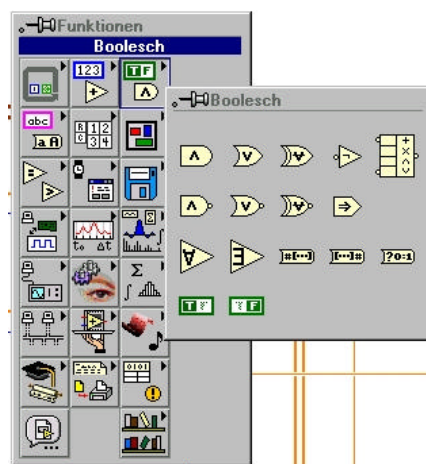


Abb. 4.1 Funktionenpalette

Diese Funktionen können je nach Bedarf in das Diagramm eingefügt werden. Diese Funktionen bzw. Vi's werden mit dem Verbindungswerkzeug untereinander verbunden.



Abb. 4.2 Werkzeugpalette

Beim Verbinden muss darauf geachtet werden, dass die richtigen Anschlusspunkte miteinander verbunden werden. Unter LabVIEW gibt es eine Vielzahl verschiedener Signalleitungen, die dazu führen, dass schnell eine inkompatible Verbindung entsteht. Diese führt dann zu Fehlermeldungen oder dazu, dass das Vi nicht ausgeführt werden kann. Neben dem Verbindungswerkzeug bietet die Werkzeugpalette noch folgende Werkzeuge, von denen hier nur die wichtigsten genannt werden.

Der Finger dient als Bedienelement, der Pfeil wird zur Auswahl und Positionierung verwendet und das A steht für ein Textbearbeitungswerkzeug. Die Hand kann zum Verschieben des Fensters verwendet werden und mit der Pipette kann eine Farbe übernommen werden.

Ein Frequenzanalysator kann wie folgt mit LabVIEW aufgebaut werden: Zuerst werden Vi's benötigt, mit denen die Daten eingelesen werden. Dazu wird das Vi „AI Multi PT“ verwendet. Es bietet die Möglichkeit ein Signal zu erfassen, dazu können Abtastrate, Blockgröße, Eingangsbereich, Eingangskanal sowie das Gerät (Messkarte) eingestellt werden. Abbildung 4.3 zeigt dieses Vi im Diagramm und seinen Einstellmöglichkeiten im Frontpanel.

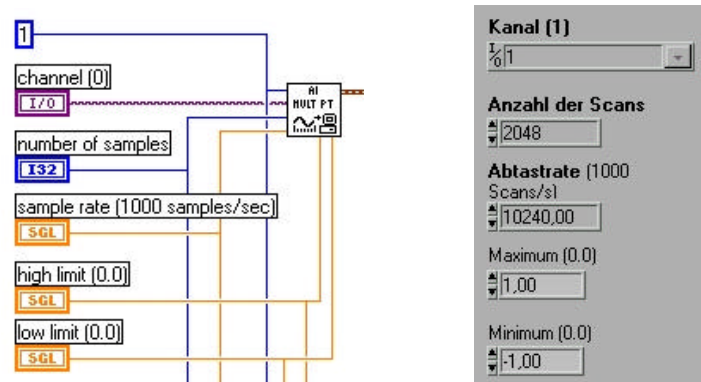


Abb. 4.3 Analogeingang AI Multi PT

Um diese Signale im Zeitbereich anzuzeigen, wird ein Anzeigeelement erzeugt.

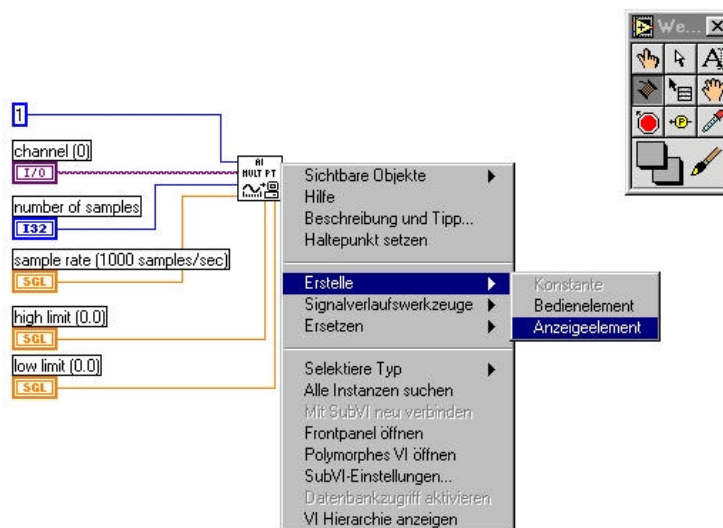


Abb. 4.4 Erstellen des Zeitfensters

Das dargestellte Menü öffnet sich durch einen Klick auf die rechte Maustaste und nach der Auswahl des Anzeigeelements erscheint dieses auf dem Frontpanel.

Folgende Abbildung 4.5 zeigt das Frontpanel vor und nach der Umwandlung des Anzeigeelementes.

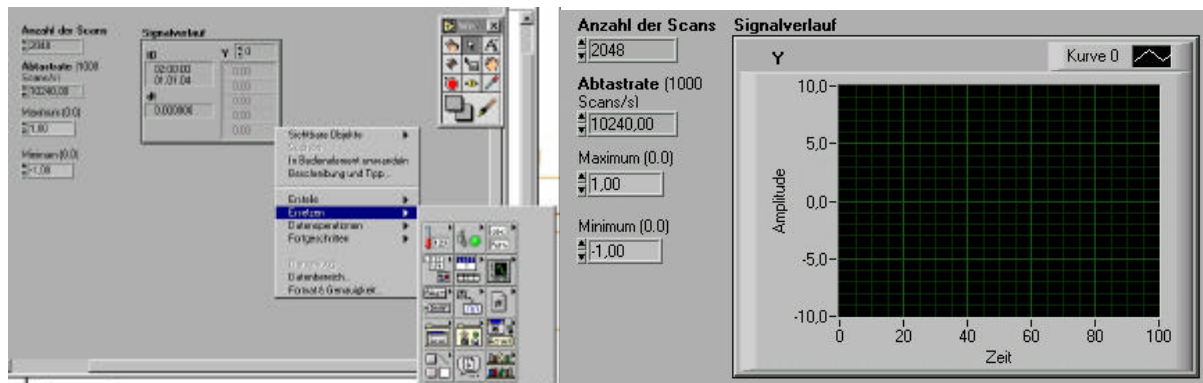


Abb. 4.5 Ändern des Anzeigeelementes

Für die Anzeige wurde der Kurvengraph ausgewählt, da er den Signalverlauf gut darstellen kann. Als nächstes ist es notwendig, die Signale mit einem Fenster zu wichten. Zuvor müssen die Signale von den noch gebündelten Daten wie t_0 und Δt getrennt werden, was mit dem Vi: „Signalverlaufskomponenten lesen“ bewerkstelligt wird. Für die Fensterfunktion wird das Vi „Skaliertes Zeitbereichsfenster“ verwendet. Abbildung 4.6 zeigt den entsprechenden Diagramm-Ausschnitt.

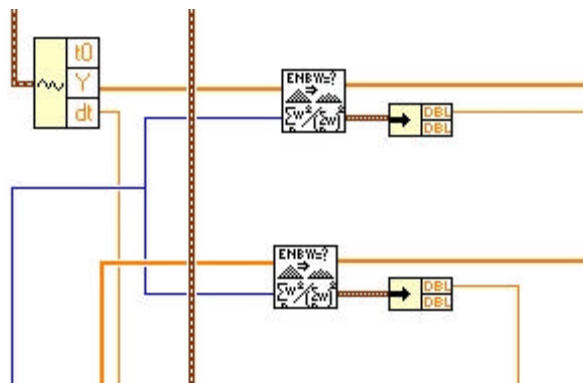


Abb. 4.6 Diagramm der Fensterfunktion

Das Vi „Skaliertes Zeitbereichsfenster“ hat über ein Bedienelement die üblichen Fensterfunktionen zur Auswahl und gibt neben den gewichteten Signalen auch die Fensterkonstanten aus, aus denen der Wert für ϵ separiert wird (rechts in Abb. 4.6 zu sehen). Das untere der beiden Vi's ist für den zweiten Kanal bestimmt, da ein Zweikanal Frequenzanalysator programmiert werden soll. Als nächster Schritt wird das Amplitudenspektrum gebildet, was mit dem Vi „Amplituden- und Phasenspektrum“ gut umzusetzen ist. Es errechnet Betrag und Phase des einseitigen, skalierten Amplitudenspektrums eines echten Zeitbereichsignals. Neben den Signaldaten benötigt es noch die Daten für Δt , die vor der Fensterfunktion separiert worden sind. Das Vi gibt die Daten für das Amplitudenspektrum sowie die Frequenzauflösung Δf aus. Beide müssen zusammengeführt werden, um dann ausgegeben zu werden.

Abbildung 4.7 zeigt das Amplitudenspektrum im Diagramm.

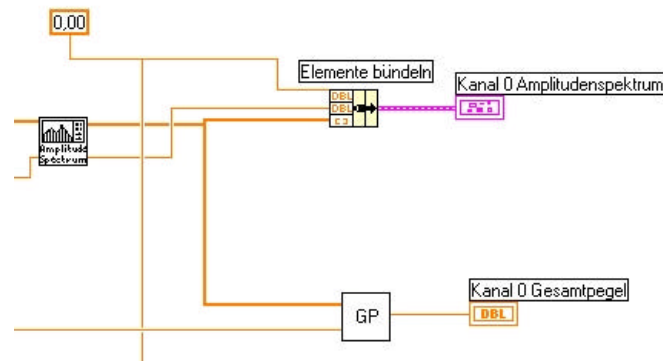


Abb. 4.7 Diagramm für das Amplitudenspektrum

Aus dem Amplitudenspektrum wird nun der Gesamtpegel berechnet. Für die Berechnung des Gesamtpegels wurde ein Vi geschaffen, dessen Signalverlauf gut zu verstehen ist, wenn man ihn mit Abschnitt 2.1.3 vergleicht.

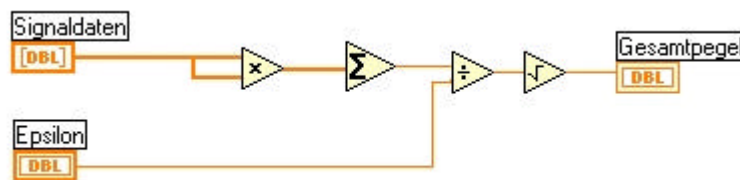


Abb. 4.8 Diagramm des Gesamtpegels

Das einseitige Leistungsspektrum, auch Auto Power Spectrum (APS) genannt, wird als nächstes erstellt. Dafür steht unter LabVIEW das Vi „Einseitiges Leistungsspektrum“ zur Verfügung. Es erstellt aus den Zeitdaten das Leistungsspektrum in RMS-Volt zum Quadrat (V_{rms}^2). Neben den gewichteten Zeitdaten benötigt dieses Vi auch die Werte für Δt , ebenso wie das Amplitudenspektrum. Dieses Vi gibt ein einseitiges Leistungsspektrum und die Frequenzauflösung aus, welche wiederum gebündelt werden müssen, um ausgegeben zu werden.

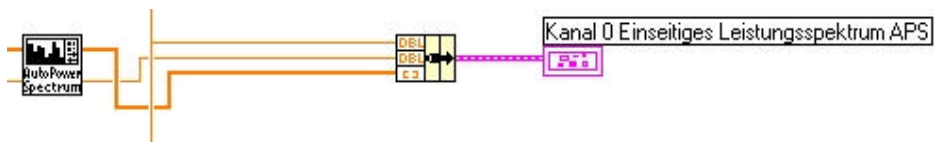


Abb. 4.9 Diagramm des Leistungsspektrums (APS)

Bei all diesen Vorgängen muss natürlich jeweils die Anzeige oder das Eingabeelement im Frontpanel erstellt, ggf. umgewandelt und positioniert werden, wie dies beim Zeitsignal (siehe oben) geschehen ist. Natürlich müssen diese Spektren zweimal erzeugt werden, um den Zweikanal Frequenzanalysator zu vervollständigen.

Für die Übertragungsfunktion hält LabVIEW das Vi „Frequenzantwortfunktion (Betrag-Phase)“ parat. Dieses kann aus den beiden Zeitsignalen X und Y (Signal X ist das Erreger-signal und Signal Y ist die Systemantwort) den Betrag (Amplitude) und die Phase der Übertragungsfunktion berechnen. Zudem gibt es die Kohärenz aus. Die Daten der Signale müssen zuvor nicht mit einem Fenster gewichtet werden, da auch diese Funktion im Vi implementiert ist. Ebenso ist eine Mittelwertbildung vorhanden, deren Parameter über das Frontpanel eingegeben werden.

Auch kann man bei der Ausgabe des Betrages die Ausgabe in dB veranlassen, oder die Phase wahlweise in Rad oder Grad ausgeben. Die Phase kann außerdem entzerrt werden, was Unstetigkeiten mit einem Absolutwert größer π ausgeschaltet. Wie die Bedienelemente im Frontpanel aussehen, zeigt Abbildung 4.10.

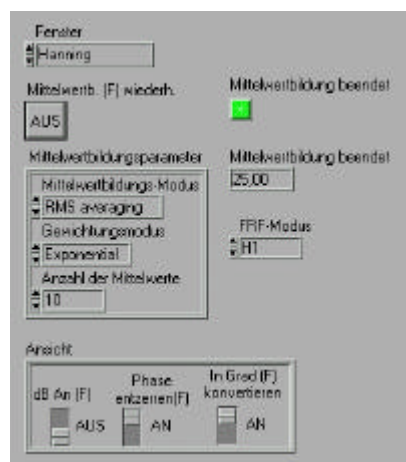


Abb. 4.10 Bedienelemente der Übertragungsfunktion

Die Übertragungsfunktion kann im Modus H1, H2 oder H3 durchgeführt werden (FRF-Modus), leider ist der Vi Beschreibung nicht zu entnehmen, welche Funktionen sich dahinter verbergen. Vermutlich werden hier die gleichen Funktionen zu Grunde gelegt, wie sie in DASyLab zur Verfügung stehen (siehe 3.6 Das Phasenspektrum auf S.20). Der Diagrammausschnitt der Übertragungsfunktion wird in Abbildung 4.11 gezeigt.

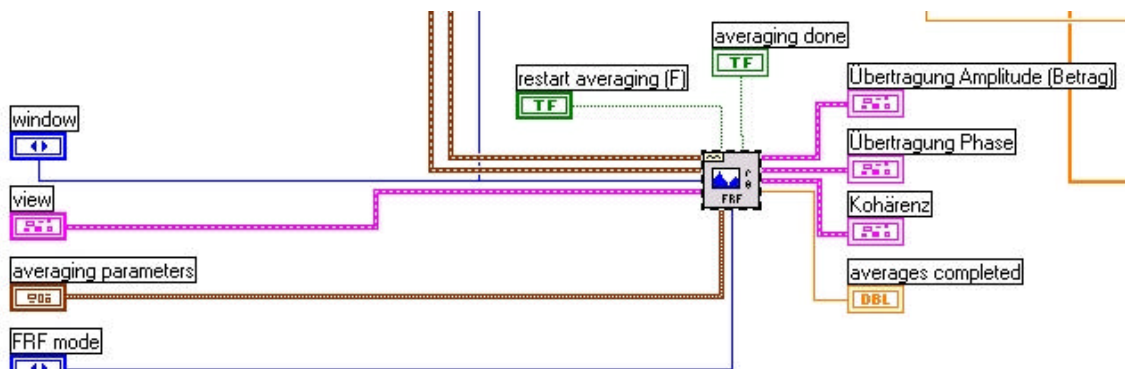


Abb. 4.11 Diagramm der Übertragungsfunktion

Das gesamte Frontpanel des Zweikanal Frequenzanalysators sieht folgendermaßen aus.

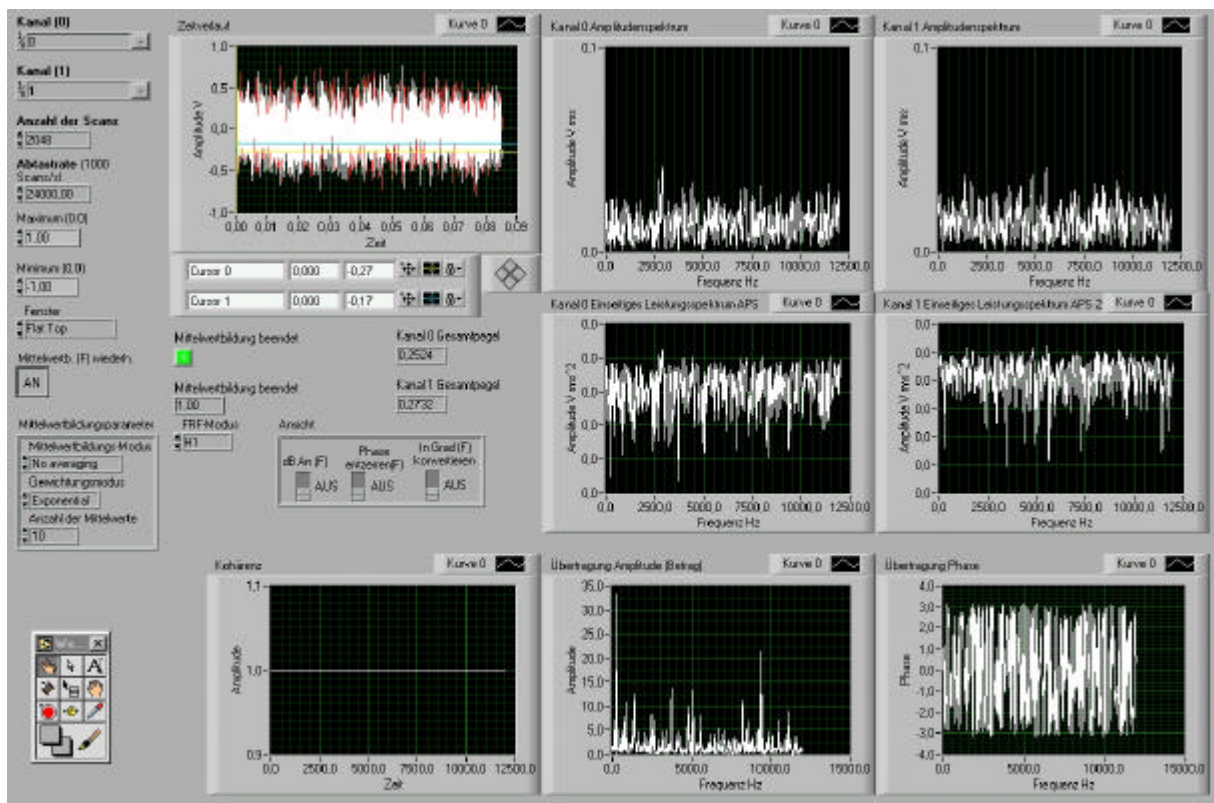


Abb. 4.12 Frontpanel des Zweikanal Frequenzanalysators

Das dazugehörige Schaltbild zeigt diese Abbildung.

Achtung! Bei diesem Schaltbild wird eine Mittelung nur über die Funktionen des Vi's „Übertragungsfunktion“ durchgeführt.

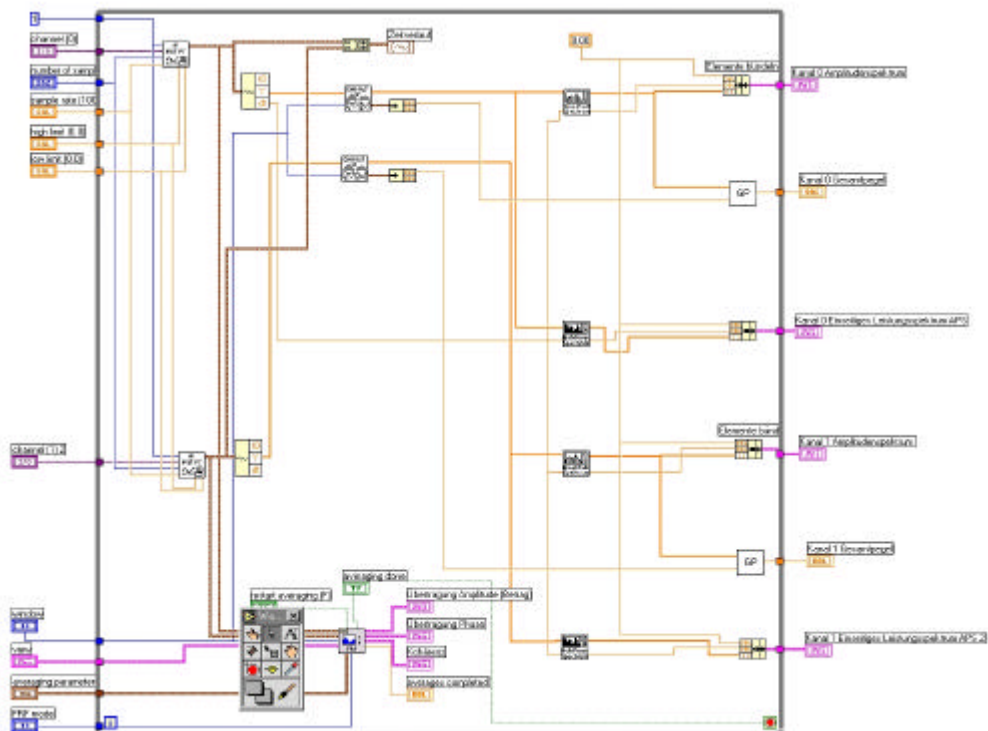


Abb. 4.13 Schaltplan des Zweikanal Frequenzanalysators

5. Versuchsaufbau und Versuchsdurchführung

5.1 Messungen des Frequenzganges einer Soundkarte

Zunächst wurde eine Art Übertragungsfunktion der Soundkarte zum Referenzmessgerät HP Dynamik Signal Analyzer 35665A ermittelt. Es wurde, bei einer Blocksize von 2048, mit 32768 Hz abgetastet. Daraus ergab sich ein $\Delta f=16\text{Hz}$. Der Frequenzgang ist bis $f_{\text{sample}}/2,56=12,8\text{ kHz}$, anhand von Amplitudenspektren, bewertet worden. Es wurden zunächst Leistungsspektren aufgezeichnet, welche jeweils 1000 mal gemittelt wurden. Als Quelle für die Messdaten wurde die Signalquelle des HP Analysators verwendet. Die Daten, welche mit der Soundkarte aufgezeichnet wurden, sind mit einem unter DASyLab erstellten Programm „test_uebertragungsverhalten.DSB“ aufgezeichnet worden.

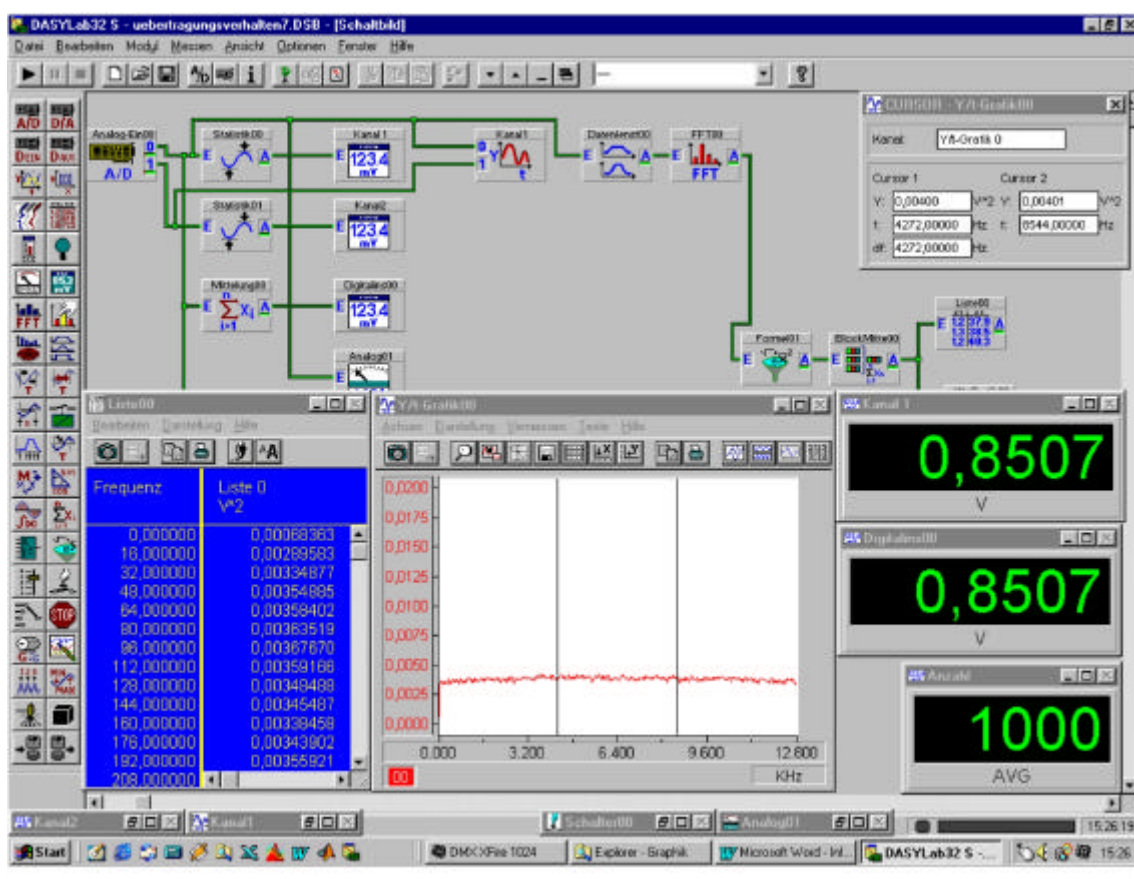


Abb. 5.1.1 Programm zur Überprüfung der Soundkarte

Programmablauf:

Die Daten wurden über die Soundkarte (A/D Wandler) eingelesen, mit einer Fensterfunktion (Datenfenster) gewichtet und anschließend eine FFT durchgeführt. Bei der FFT wurde ein Amplitudenspektrum erzeugt, welches im Formelinterpreter quadriert wurde, um ein Leistungsspektrum zu erhalten. Dieses wurde daraufhin gemittelt, um das Rauschsignal zu glätten. Die Daten wurden zum einen als Grafik ausgegeben und zum anderen in einer Liste abgespeichert. Diese Liste wurde dann exportiert und in einem Format abgespeichert, welches von MATLAB erkannt wird, um die Leistungsspektren zusammenzuführen und auszuploten.

Das Programm zum plotten der Amplitudenspektren sah folgendermaßen aus:

```
clear all
fid = fopen('UebDas11.txt'); %oeffnen der Textdatei (Text mit Zeilen-
umbruch)
a = fscanf(fid,'%g %g',[2 inf]); %zweireihig abgespeichert .
a = a';
fclose(fid)
load UebHPA06.mat;
c1=(c1/2); %halbierung der y-Werte da bedingt durch den konventierungs-
prozess die Werte zwei mal größer wurden
c1=sqrt(c1);%Wurzel aus dem Leistungsspektrum des HP Analysators
c2x=a(:,1);
c2=a(:,2);
c2=sqrt(c2);%Wurzel aus dem Leistungsspektrum der Soundkarte
plot(c1x,c1,c2x,c2);
xlabel('Hz')
ylabel('Vrms')
```

Die *.txt Dateien wurden über die Soundkarte eingelesen, die Dateien mit der Endung *.DAT sind mit dem HP 35665A erstellt worden.

Die erste Messung zeigt die Amplitudenspektren des HP Analysators und der Soundkarte mit einer Flat Top Fensterfunktion jedoch ohne Kalibrierung des Soundkarten Eingangs. Die hier getestete Soundkarte war die TerraTec DMX X-fire 1024 mit einer max. Abtastfrequenz von 48 kHz. Der Messbereich des Soundkarten Eingangs wurde hier auf +/- 5 V eingestellt.

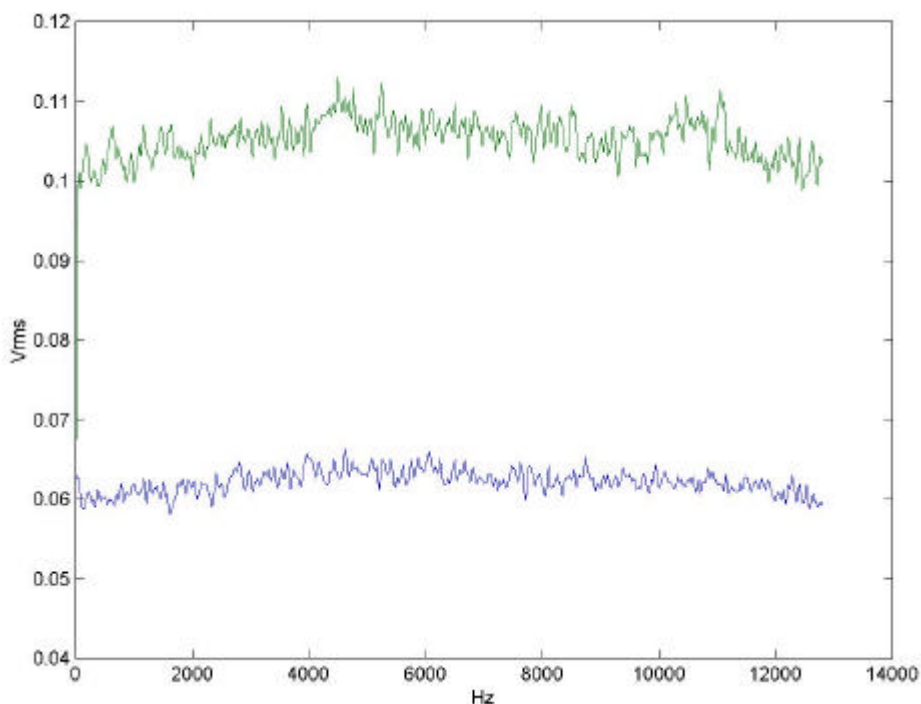


Abb. 5.1.2 Frequenzgänge $f_{\text{Sample}} = 32768\text{Hz}$, $\text{Blocksize} = 2048$, $\Delta f = 16\text{Hz}$,
 $\text{Mittelungen} = 1000$, $\text{Source} = 1V_{\text{rms}}$

Der grüne Verlauf zeigt die Werte der Soundkarte, der blaue zeigt die Werte des HP Analysators. Es zeigte sich, dass die mit der Soundkarte gemessenen Werte um ca.75% höher waren als die des HP Analysators.

Das folgende Diagramm, Abbildung 5.1.3, zeigt die Abweichungen der beiden Frequenzverläufe zueinander in dB.

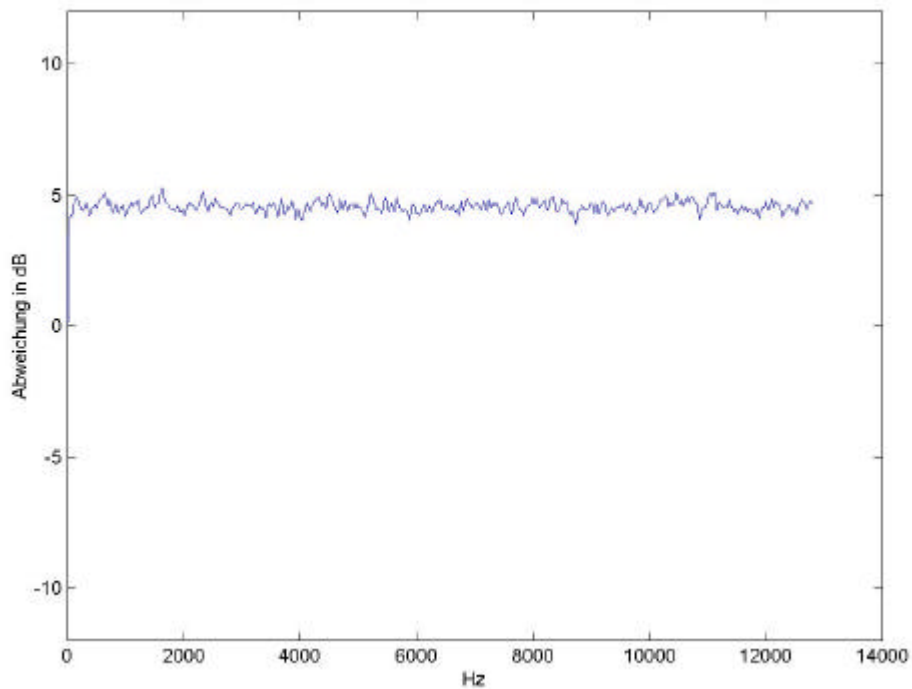


Abb. 5.1.3 Abweichung der Soundkarte in dB

Bei der nächsten Messung wurde der Messbereich der Soundkarte auf $\pm 1V$ eingestellt, wobei eine Übersteuerung zu erwarten war.

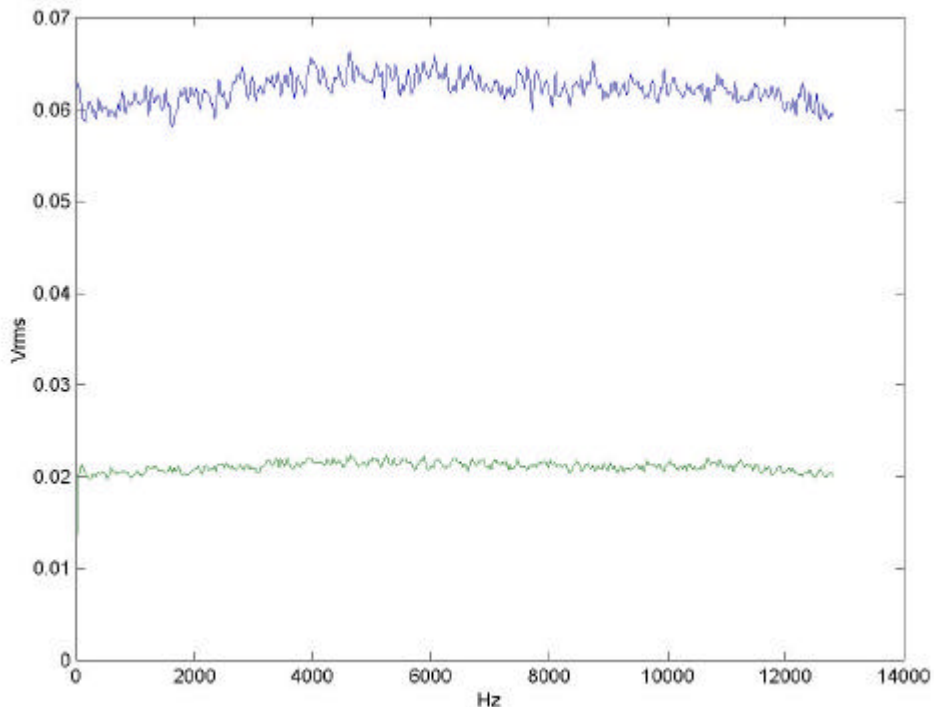


Abb. 5.1.4 Frequenzgänge $f_{Sample} = 32768Hz$, $Blocksize = 2048$, $\Delta f = 16Hz$,
Mittelungen = 1000, $Source = 1V_{rms}$

Der grüne Verlauf zeigt wieder die Werte der Soundkarte, der blaue zeigt die Werte des HP Analysators. Die so ermittelten Werte waren jedoch um ca. 66% kleiner als die des HP Analysators.

Die Abweichungen in dB zeigt das folgende Diagramm Abb. 5.1.5.

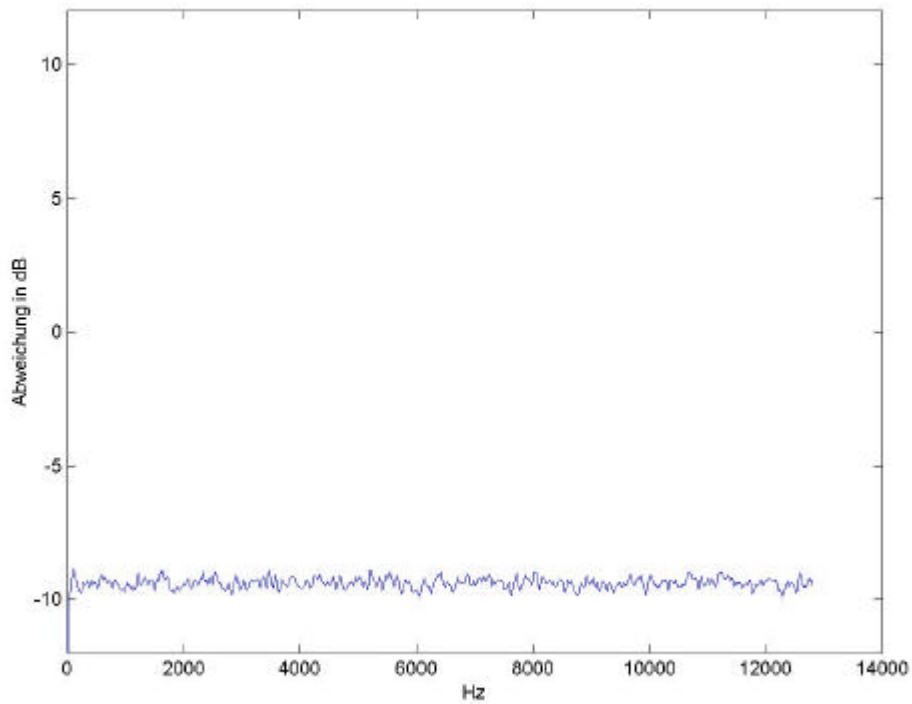


Abb. 5.1.5 Abweichung der Soundkarte in dB

Bei der folgenden Messung wurde ein Messbereich von $\pm 2V$ gewählt.

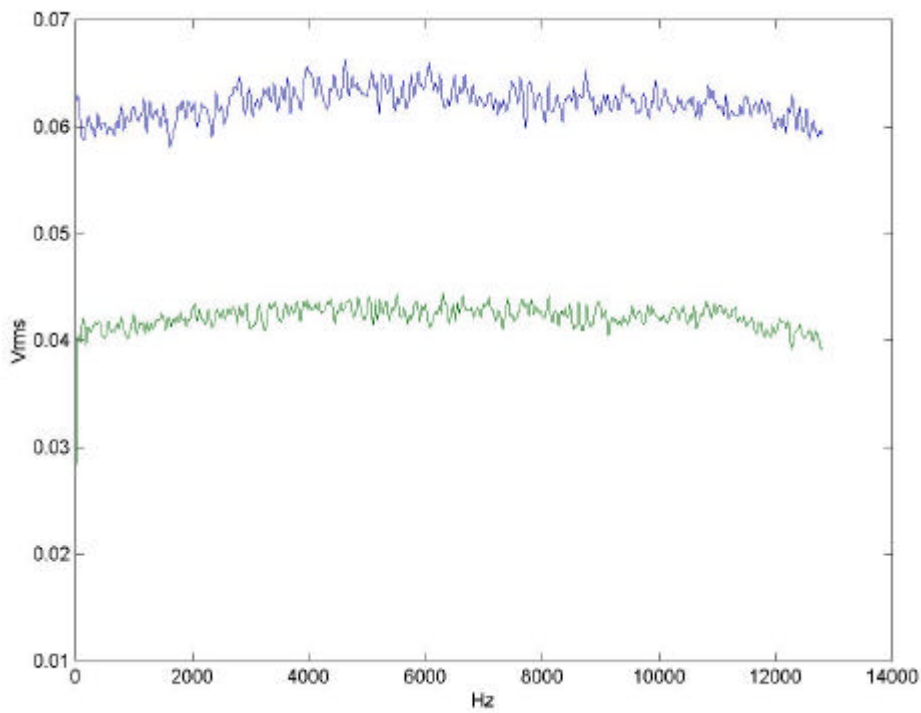


Abb. 5.1.6 Frequenzgänge $f_{\text{Sample}} = 32768\text{Hz}$, $\text{Blocksize} = 2048$, $\Delta f = 16\text{Hz}$,
 $\text{Mittelungen} = 1000$, $\text{Source} = 1V_{\text{rms}}$

Die vorherige Grafik zeigt, dass die gemessenen Werte hier den Werten des HP Analysators wesentlich näher kamen. Die Abweichungen betragen hier ca. 50%. Das folgende Diagramm zeigt wieder die Abweichungen in dB.

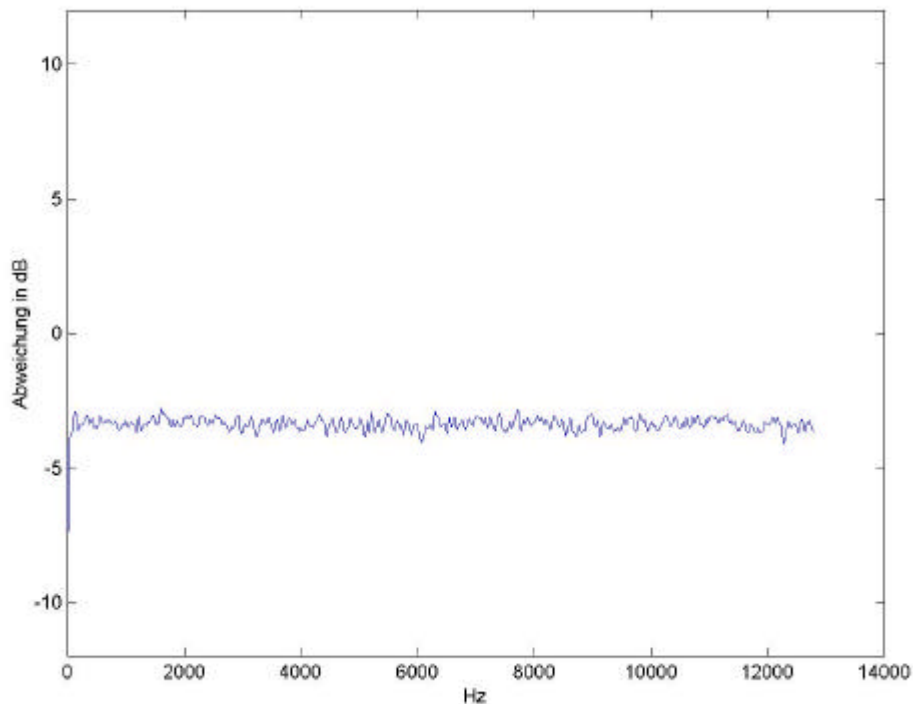


Abb. 5.1.7 Abweichung der Soundkarte in dB

Für eine bessere Anpassung musste nun eine Kalibrierung durchgeführt werden. Die Kalibrierung, unter DASyLab, wurde mittels der Kanal Skalierung für den Eingang der Soundkarte (Analogeingang) durchgeführt.

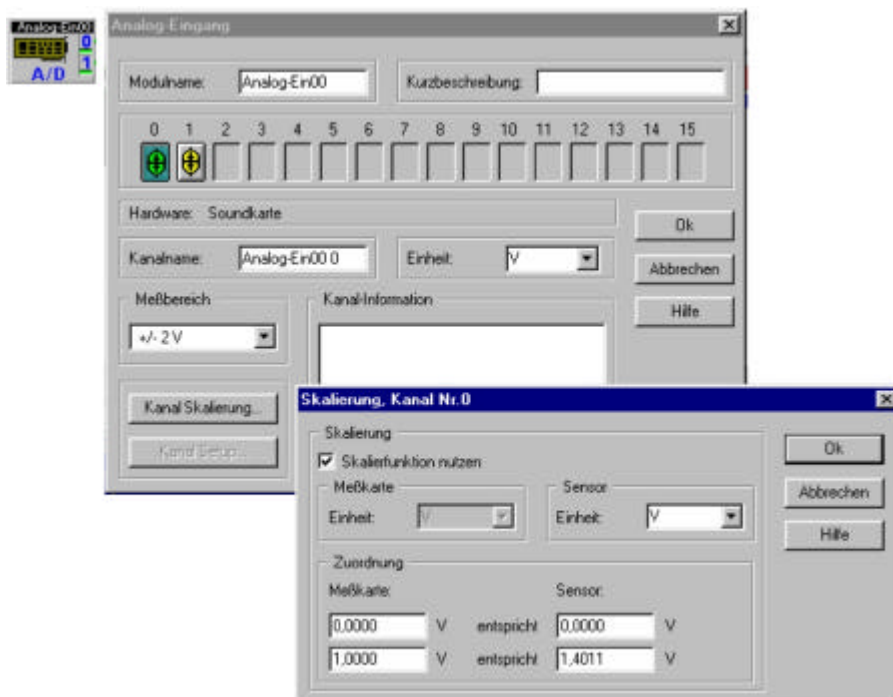


Abb. 5.1.8 Skalierfunktion des Analog-Eingangs

Als Anhaltspunkt für die Kalibrierung dienten hier die Band Power Funktion des HP- Analysators und der unter DASYSLab ermittelte Effektivwert. Über das Verhältnis von Sollwert zu Istwert ($V_{\text{soll}}/V_{\text{ist}}$), der vorherigen Messung, wurde ein Kalibrierwert von 1:1,4010 eingestellt.

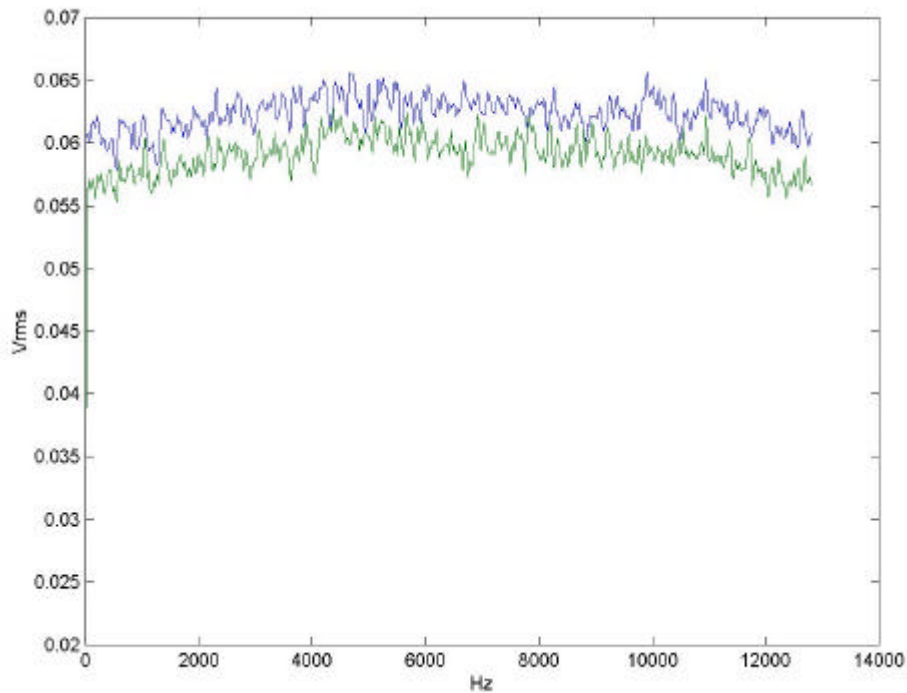


Abb. 5.1.9 Frequenzgänge $f_{\text{Sample}} = 32768\text{Hz}$, $\text{Blocksize} = 2048$, $\Delta f = 16\text{Hz}$,
 Mittelungen = 1000, Source = $1V_{\text{rms}}$

Die Messung ergab, dass die Werte der Soundkarte immer noch um ca. 10% kleiner waren als die des HP Analysators.

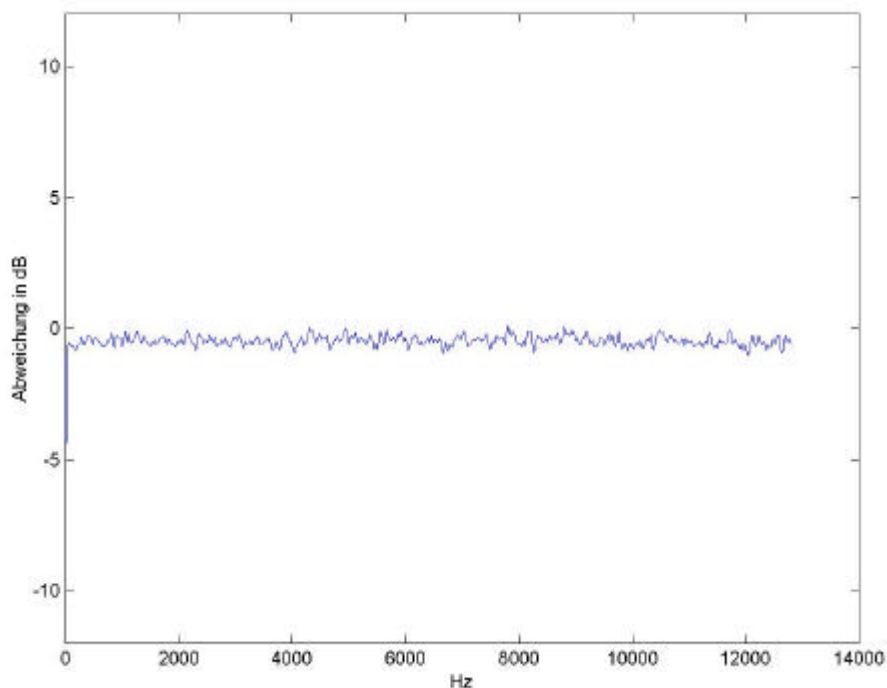


Abb. 5.1.10 Abweichung der Soundkarte in dB

Durch erneute Berechnung wurde ein Skalierwert von 1:1,48 gefunden, bei dem die Abweichungen fast nicht mehr erkennbar waren.

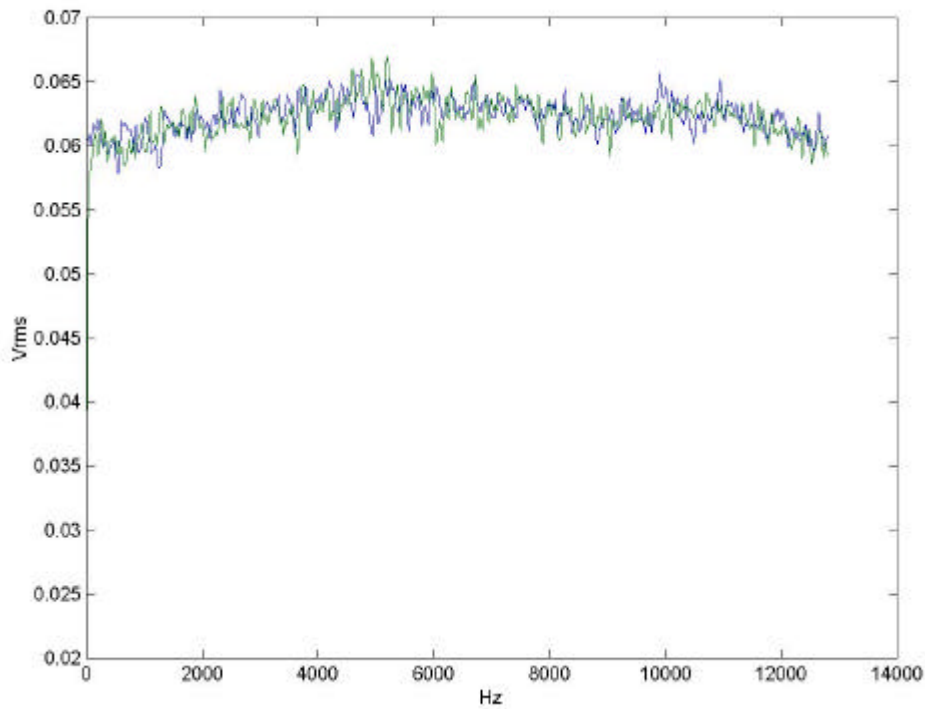


Abb. 5.1.11 Frequenzgänge $f_{\text{Sample}} = 32768\text{Hz}$, $\text{Blocksize} = 2048$, $\Delta f = 16\text{Hz}$,
 $\text{Mittelungen} = 1000$, $\text{Source} = 1V_{\text{rms}}$

Die beiden Verläufe, grün = Soundkarte und blau = HP Analyser, der Signale lagen hier nahezu übereinander. Es zeigte sich hier, dass die Signale bis zu einer Frequenz von 12,8 kHz recht gut übereinstimmen. Die beiden Frequenzverläufe weichen im Schnitt um ca $\pm 0,3$ dB voneinander ab.

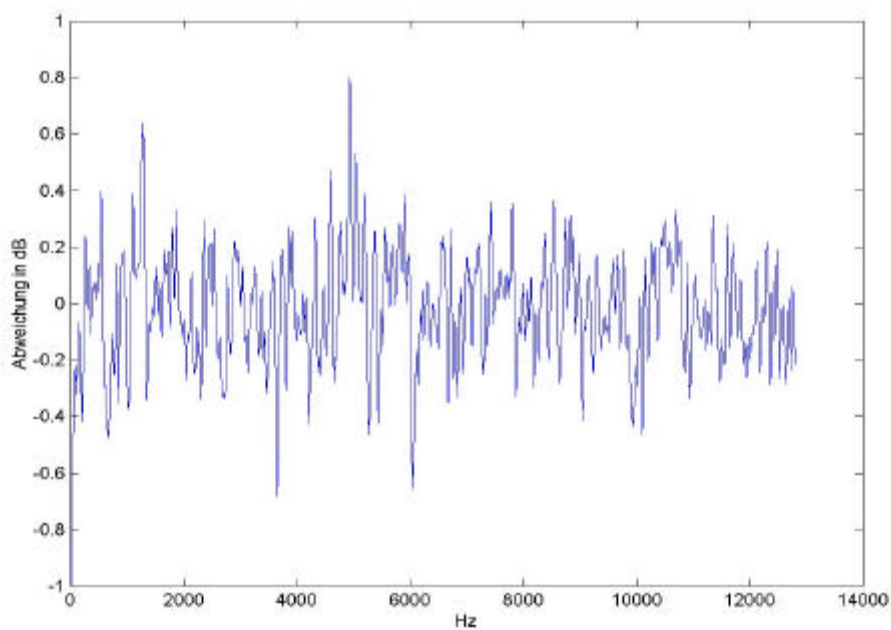


Abb. 5.1.12 Abweichung der Soundkarte in dB

Auch bei einer Spannung von $1,135 V_{rms}$, praktisch die maximale Amplitude dessen, was die Quelle des HP Analysators leisten kann, war der Verlauf der Signale fast synchron.

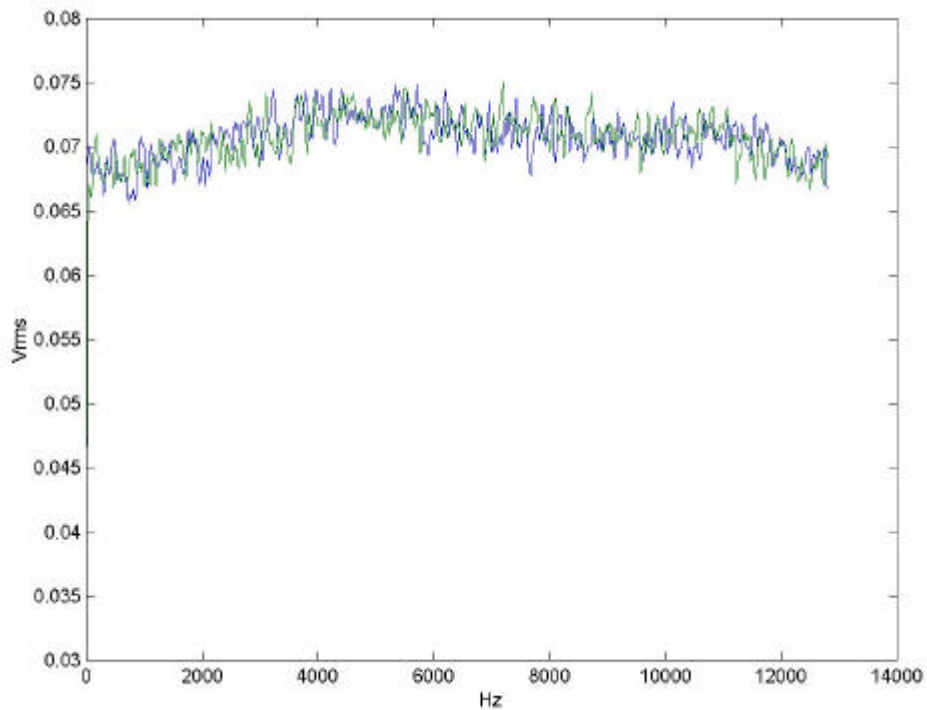


Abb. 5.1.13 Frequenzgänge $f_{Sample} = 32768\text{Hz}$, $Blocksize = 2048$, $\Delta f = 16\text{Hz}$,
 $Mittelungen = 1000$, $Source = 1,135V_{rms}$

Ebenso wichen die beiden Frequenzverläufe im Schnitt um maximal $\pm 0,3 \text{ dB}$ voneinander ab.

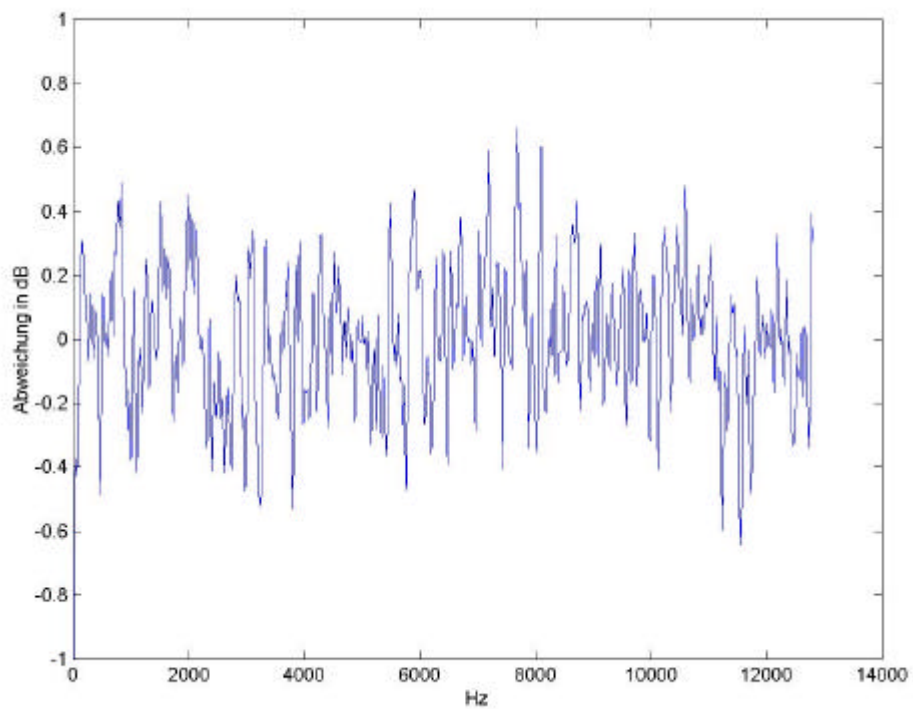


Abb. 5.1.14 Abweichung der Soundkarte in dB

Auch die folgende Messung mit einer Quellenspannung von $0,5 V_{rms}$ zeigte eine gute Übereinstimmung.

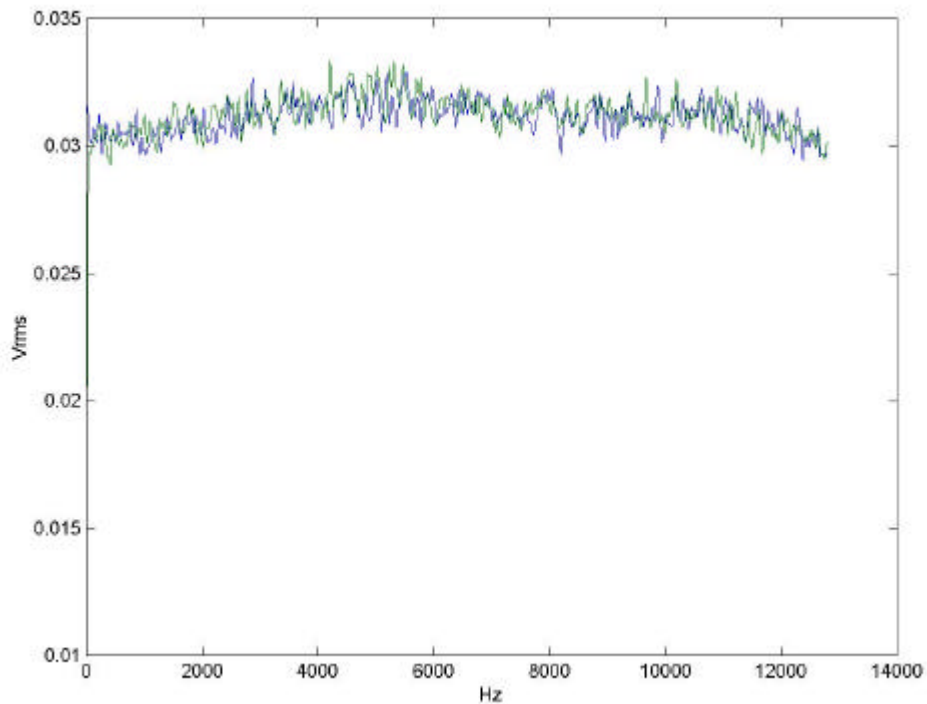


Abb. 5.1.15 Frequenzgänge $f_{Sample} = 32768\text{Hz}$, $Blocksize = 2048$, $\Delta f = 16\text{Hz}$,
 $Mittelungen = 1000$, $Source = 0,5V_{rms}$

Die beiden Frequenzverläufe wichen im Schnitt auch hier um nicht mehr als $\pm 0,3 \text{ dB}$ voneinander ab.

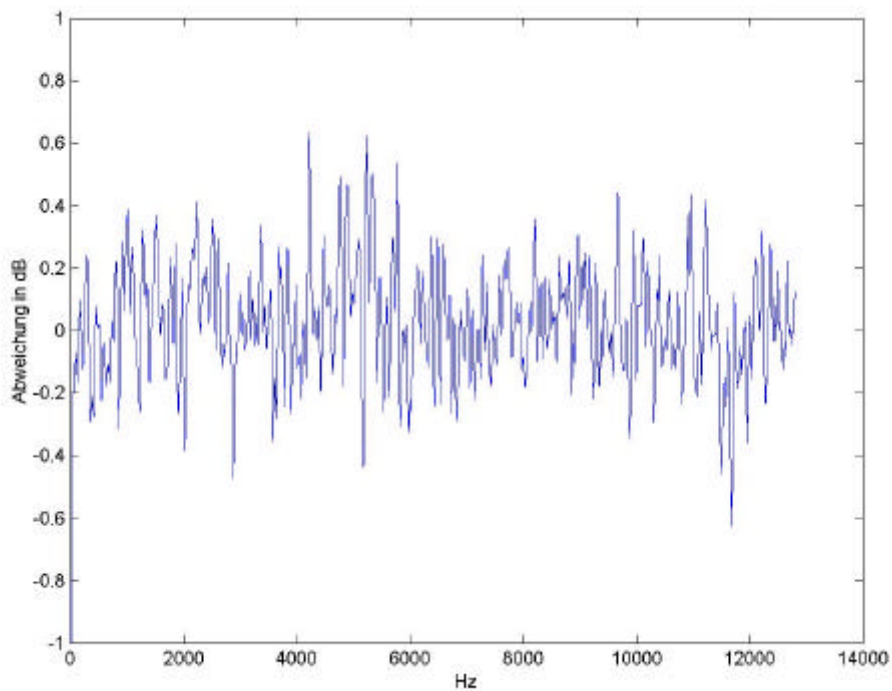


Abb. 5.1.16 Abweichung der Soundkarte in dB

Ebenso verhielt es sich mit den Frequenzverläufen bei einer Quellenspannung von $0,2 V_{rms}$.

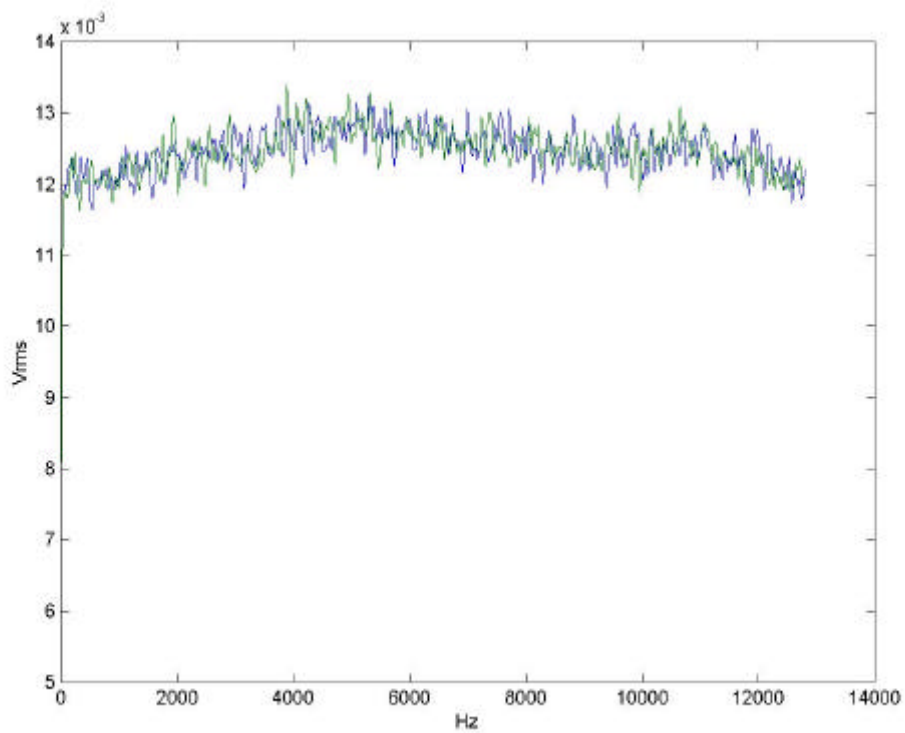


Abb. 5.1.17 Frequenzgänge $f_{Sample} = 32768\text{Hz}$, $Blocksize = 2048$, $\Delta f = 16\text{Hz}$,
 $Mittelungen = 1000$, $Source = 0,2V_{rms}$

Wieder zeigte sich, dass die beiden Frequenzverläufe im Schnitt nicht mehr als $\pm 0,3$ dB voneinander abweichen.

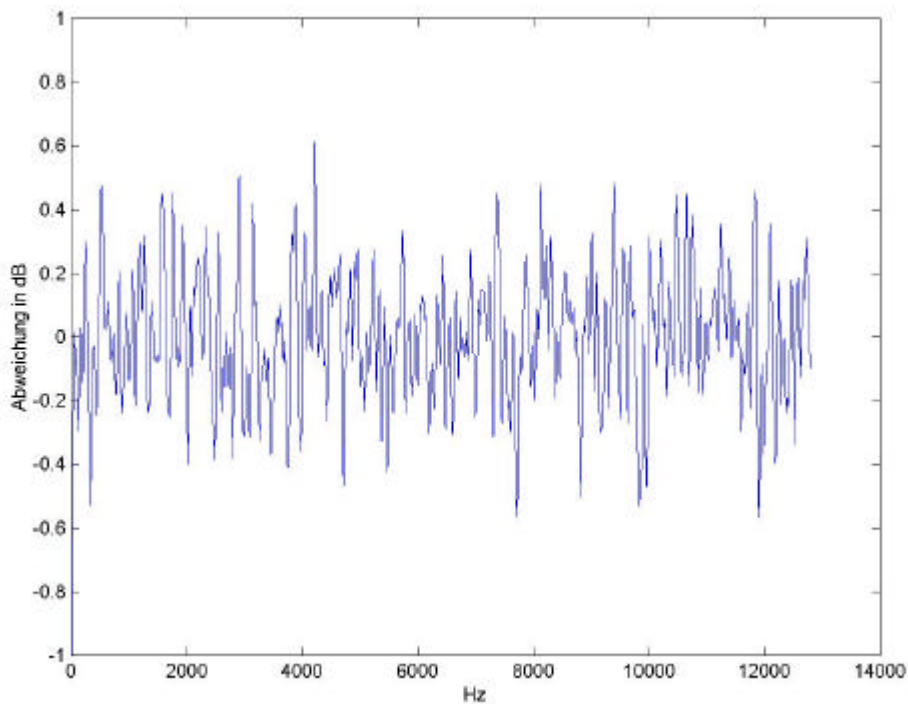


Abb. 5.1.18 Abweichung der Soundkarte in dB

Der aufgezeichnete Frequenzgang einer SB 128 PCI Soundkarte (die zur Standardausstattung der PC's an der FH-Düsseldorf gehört) weicht hingegen vom annähernd linearen Frequenzgang des HP Analysators ab. Die Quellenspannung betrug hier $1 V_{rms}$ und der Skalierfaktor war nach wie vor auf 1,48 eingestellt.

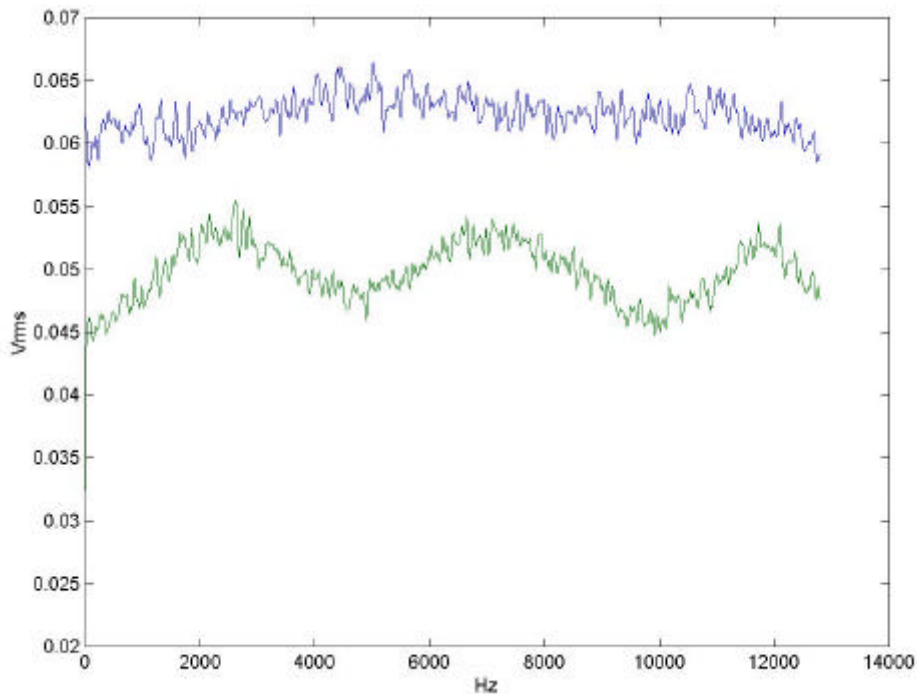


Abb. 5.1.19 Frequenzgänge $f_{Sample} = 32768Hz$, $Blocksize = 2048$, $\Delta f = 16Hz$,
Mittelungen = 1000, $Source = 1V_{rms}$

Folgendes Diagramm zeigt recht große Abweichungen von bis zu -3 dB.

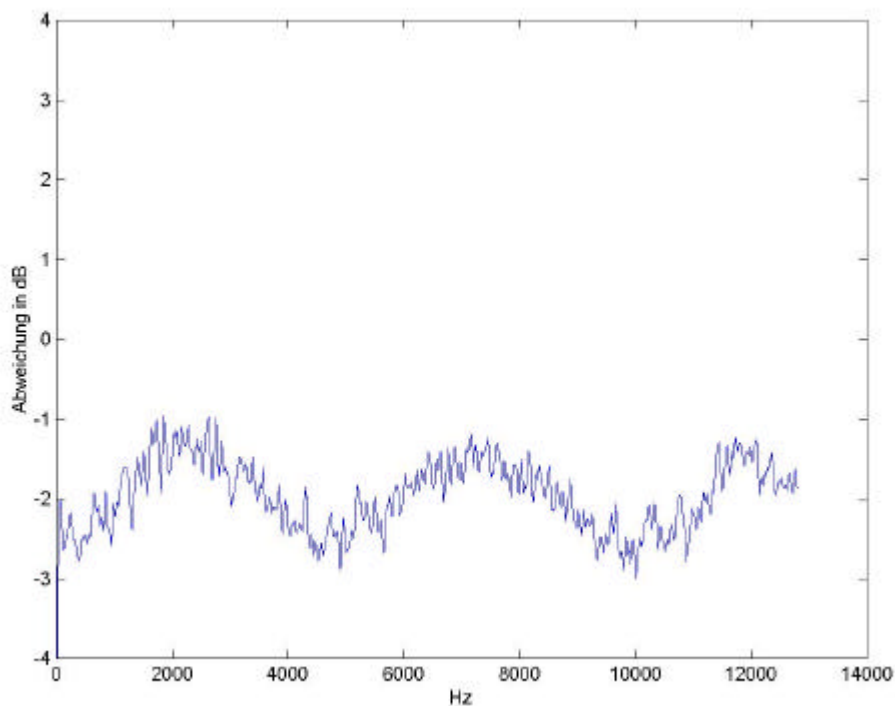


Abb. 5.1.20 Abweichung der Soundkarte in dB

Bei einem Skalierfaktor von 1:2 ergaben sich folgende Frequenzverläufe.

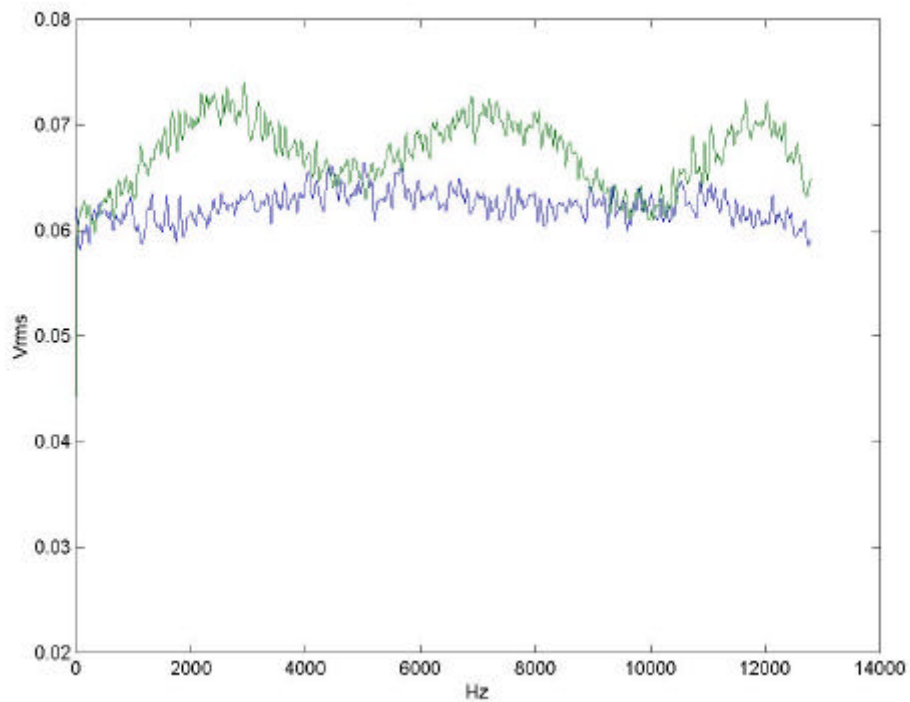


Abb. 5.1.21 Frequenzgänge $f_{\text{Sample}} = 32768\text{Hz}$, $\text{Blocksize} = 2048$, $\Delta f = 16\text{Hz}$,
 $\text{Mittelungen} = 1000$, $\text{Source} = 1V_{\text{rms}}$

Es traten hier noch Abweichungen von $-0,3\text{ dB}$ bis zu $1,5\text{ dB}$ auf.

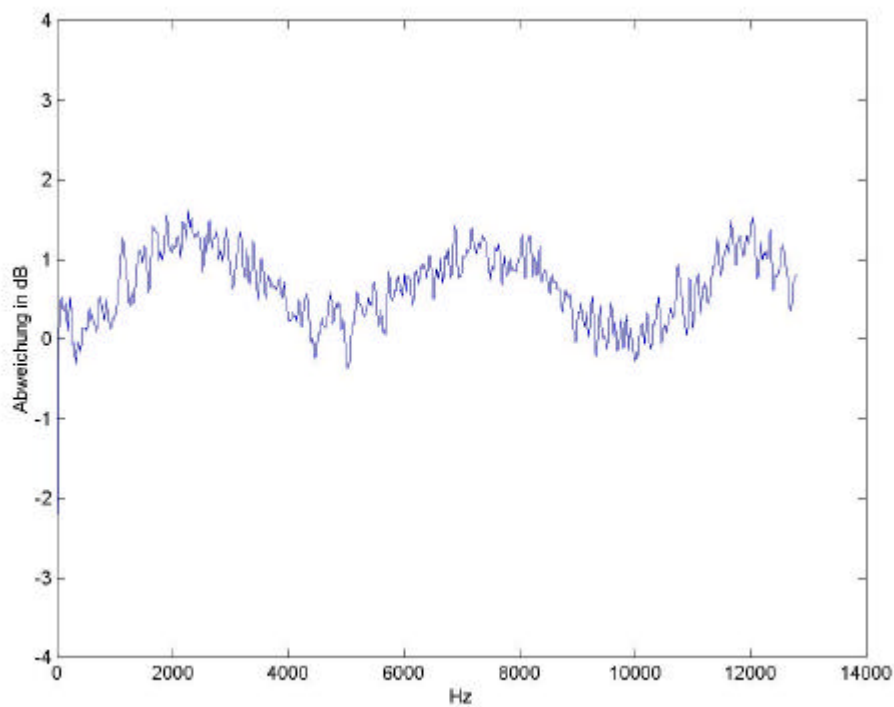


Abb. 5.1.22 Abweichung der Soundkarte in dB

Bei einem Skalierfaktor von 1:1,8 ergaben diese sich diese Frequenzverläufe für die SB 128 PCI Soundkarte (= grün) und den HP Analysator (= blau).

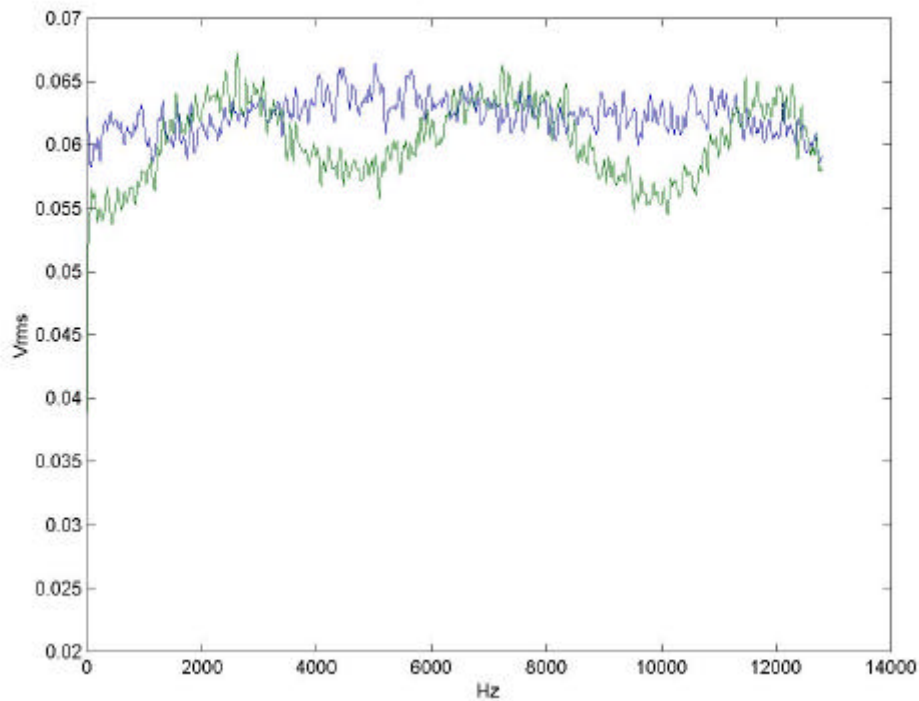


Abb. 5.1.23 Frequenzgänge $f_{Sample} = 32768\text{Hz}$, $Blocksize = 2048$, $\Delta f = 16\text{Hz}$,
 $Mittelungen = 1000$, $Source = 1V_{rms}$

Hierbei zeigten sich Abweichungen von ca. $-1,3\text{ dB}$ bis $+0,5\text{ dB}$.

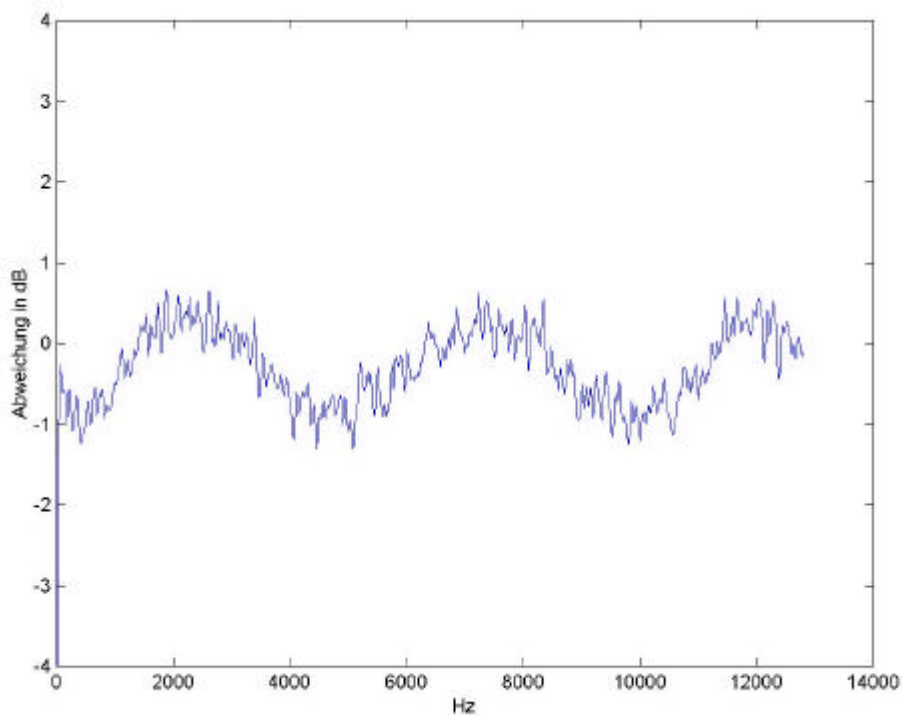


Abb. 5.1.24 Abweichung der Soundkarte in dB

Diese Messungen zeigten, dass die getestete Soundkarte vom Typ DMX Xfire 1024 der Firma TerraTec ein sehr gutes lineares Übertragungsverhalten hat. Die im Gegenzug getestete Soundkarte der Fa. Creative Labs, Typ Soundblaster 128 PCI, zeigte eine Welligkeit, d.h. eine zyklische Abweichung vom linearen Frequenzverlauf. Es sollte aber dabei erwähnt werden, dass diese SB 128 PCI als Standard Soundkarte gesehen werden muss, die den normalen Anforderungen (Ausgabe von Tönen bzw. Musik) für einen PC genügt.

Es ist noch zu erwähnen, dass in das Programm „test_übertragungsverhalten.DSB“ eine Effektivwertbildung eingefügt wurde. Dieses war zwar nicht notwendig, weil dieser Faktor mit in den Kalibrierwert eingeht. Es ist für die Überschaubarkeit des Programms aber von Vorteil.

Hier sollte noch erwähnt werden, dass die Audioeigenschaften des PC's folgendermaßen eingestellt waren:

Eingang: Line

Aufnahmeregler des Line Eingang im Aufnahmeregler bzw. Audiomixer der DMX Xfire 1024: Unterste Stellung.

Der HP Analysator 35665A wurde mittels eines Adapters von 2xChinch-Stecker (Mono) auf 1x 3,5 mm Klinke-Stecker (Stereo) angeschlossen. Zu den Chinch-Stecker kamen noch Adapter von BNC auf Chinch-Kupplung, um die Verbindungen zu den Ausgängen des HP Analysator 35665A herzustellen.

5.2 Überprüfung auf Phasengenauigkeit

5.2.1 Überprüfung der Soundkarte auf Phasengenauigkeit

Um über die Qualität einer mehrkanaligen Messkarte oder Soundkarte etwas auszusagen, muss diese auf ihre Phasengenauigkeit überprüft werden. Überprüft wird hier eine TerraTec DMX Xfire1024 mit einer maximalen Abtastfrequenz von 48 kHz je Kanal. Bei dieser handelt es sich um eine Soundkarte des etwas gehobenen Niveaus, die im Handel für 100,- bis 130,- DM vertrieben wird. Für die Überprüfung wird ein Phasengang erzeugt, indem ein Signal aus *einer* Quelle auf beide Kanäle der Soundkarte gegeben wird. Im Idealfall zeigt sich im Phasenverlauf des Übertragungsmoduls ein ebener Verlauf entlang der 0° Linie. Der Wert der Amplitude der Übertragungsfunktion gibt an, inwiefern die Amplituden der Signale übereinstimmen. Der Wert der Amplitude sollte sich bei 1 bewegen. Weiterhin wird noch die Kohärenz der beiden Signale überprüft, deren Wert möglichst bei 1 liegen sollte, um die Phase überhaupt interpretieren zu dürfen. Bei der folgenden Messung wurde mit einem HP 35665A Frequenzanalysator ein weißes Rauschen bis zu einer Frequenz von 25,6 kHz mit einer Amplitude von 1 V_{pk} erzeugt. Die Signale wurden jeweils mit einer Abtastfrequenz von 48 kHz abgetastet. Die Blocksize betrug 2048 Werte und es wurde mit einer Flat Top Fensterfunktion gewichtet. Es wurden 200 Mittelungen durchgeführt.

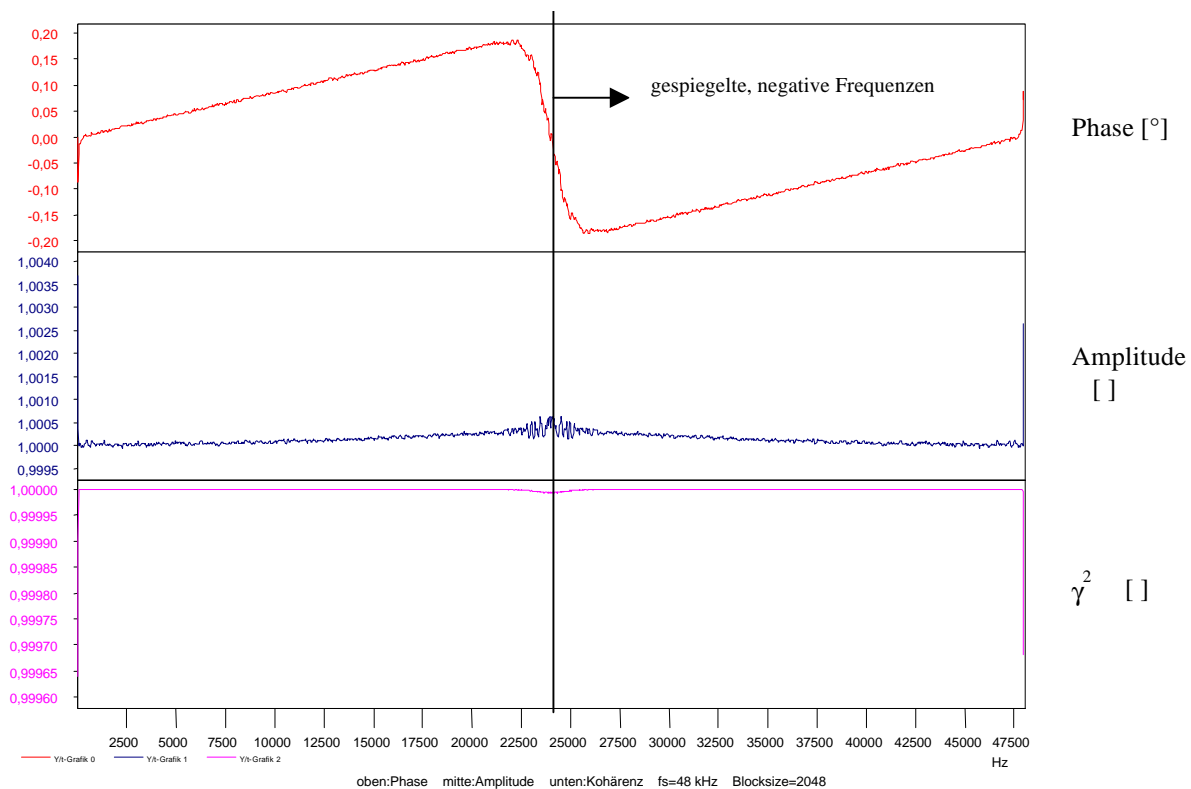


Abb. 5.2.1.1 Phase, Amplitude und Kohärenz der Soundkarte: DMX Xfire 1024
f_s: 48 kHz Blocksize: 2048 Mittelungen: 200

Dieses Diagramm gibt die Phase, Amplitude und Kohärenz auch für die negativen Frequenzen an. Es wäre jedoch sinnvoll, die Verläufe nur bis zur Nyquistfrequenz darzustellen. Es muss auch beachtet werden, dass in der Nähe der Nyquistfrequenz Fehler durch Aliasingeffekte auftreten.

Es zeigte sich, dass eine maximale Phase von $0,18^\circ$ bei etwa 22,3 kHz auftrat, was einen sehr guten Wert darstellt. Auch für die Amplitude und die Kohärenz ergaben sich sehr gute Werte. Der Verlauf der Phase verhält sich bis zu einer Frequenz von 22,5 kHz nahezu linear, was bei Messungen von Vorteil ist, da keine aufwendigen Kalibrierfunktionen nötig sind.

Bei der nächsten Messung wurde nur der erste Kanal mit einem Signal belegt. Es wurde wieder mit einer Abtastfrequenz von 48 kHz und einer Blocksize von 2048 gemessen. Die Anzahl der Mittelungen betrug auch hier 200.

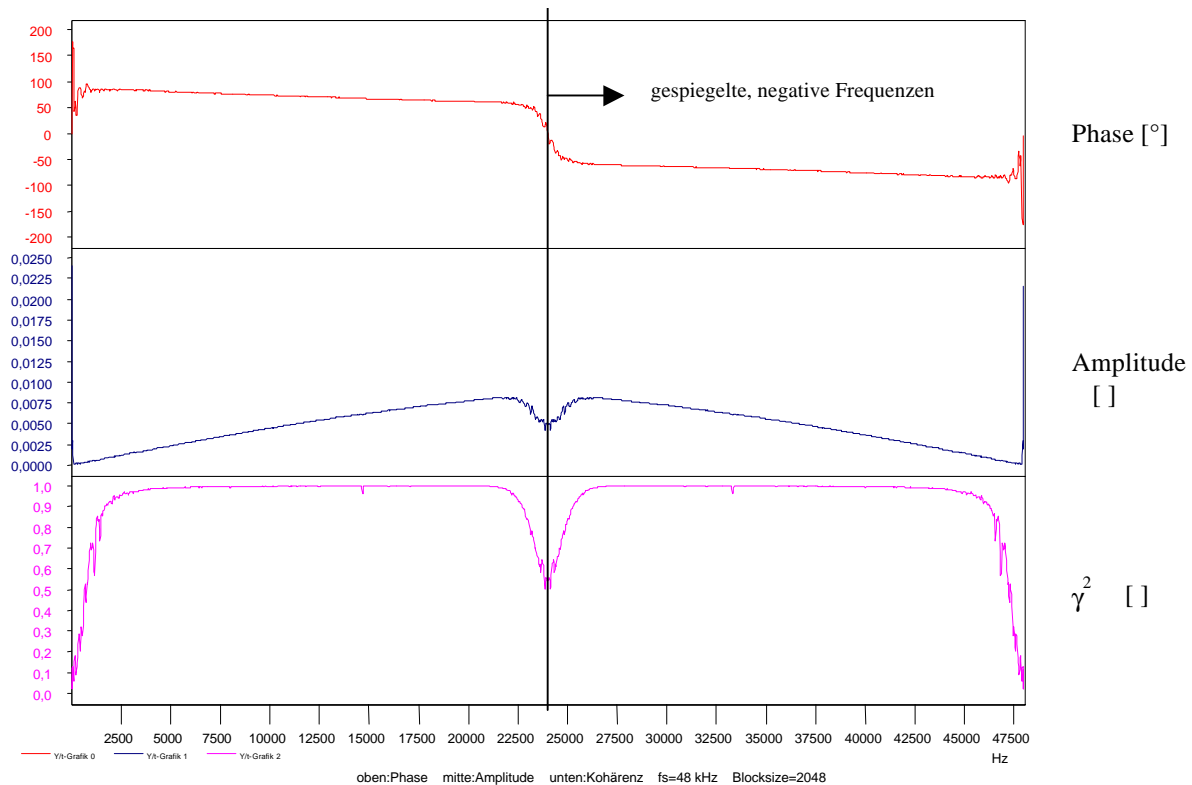


Abb. 5.2.1.2 Phase, Amplitude und Kohärenz der Soundkarte: DMX Xfire 1024
 f_s : 48 kHz Blocksize: 2048 Mittelungen: 200

Diese Diagramme zeigen wie stark die Phase abweicht, wenn Signal 1 und Signal 2 nicht übereinstimmen. Besonders in den niedrigen Frequenzen ist die Phasenlage vollkommen unbestimmt. Auch die Amplitude zeigt nur recht geringe Werte von maximal 0,00819 bei einer Frequenz von 22007,8 Hz. Es ist verwunderlich, dass überhaupt ein Wert für die Amplitude ausgegeben wird, da kein Signal am 2. Kanal anlag. Ebenso verhielt es sich mit der Kohärenz, die immerhin Werte über 0,99 im Frequenzbereich von 5 kHz bis 21,5 kHz annahm. Um zu überprüfen, ob kein Fehler von DASYLab herrührt, wurden je eine Wav-Datei mit einem Kanal und mit beiden Kanälen aufgezeichnet und anschließend mit MATLAB ausgewertet.

Das folgende Listing zeigt das MATLAB Programm, mit dem die unter DASYLab erzielten Ergebnisse verglichen wurden. Diese MATLAB-Routinen sind zuvor mit dem System PAK der Fa. Müller BBM verglichen worden, um die Routine auf Richtigkeit zu überprüfen..

```
clear all
[y,Fs,bits] = wavread('48khz-stereo16b-rauschen-eine');
nfft=4096;
window=hanning(nfft);
[B1,f,t]=specgram(y(:,1),nfft,Fs>window,nfft/2);
[B2,f,t]=specgram(y(:,2),nfft,Fs>window,nfft/2);
cps12=B1.*conj(B2);
[m,n]=size(cps12);
cps12_mit=mean(cps12(:,1:n)');
phase12=180*angle(cps12_mit)/pi;
B1B1=mean((B1.*conj(B1))');
B2B2=mean((B2.*conj(B2))');
coh=cps12_mit.*conj(cps12_mit)./(B1B1.*B2B2);
fmax=5000;
subplot(3,1,1),plot(f,20*log10(B1B1),f,20*log10(B2B2)),axis ([0 fmax 0
20]),title('Amplitudenspektren');
subplot(3,1,2),plot(f,phase12),axis ([0 fmax -180 180]),title('Phase');
subplot(3,1,3),plot(f,coh),axis ([0 fmax 0 1.1]),title('Kohärenz');
```

Die folgenden Grafiken sind mit oben stehendem Programm erzeugt worden.

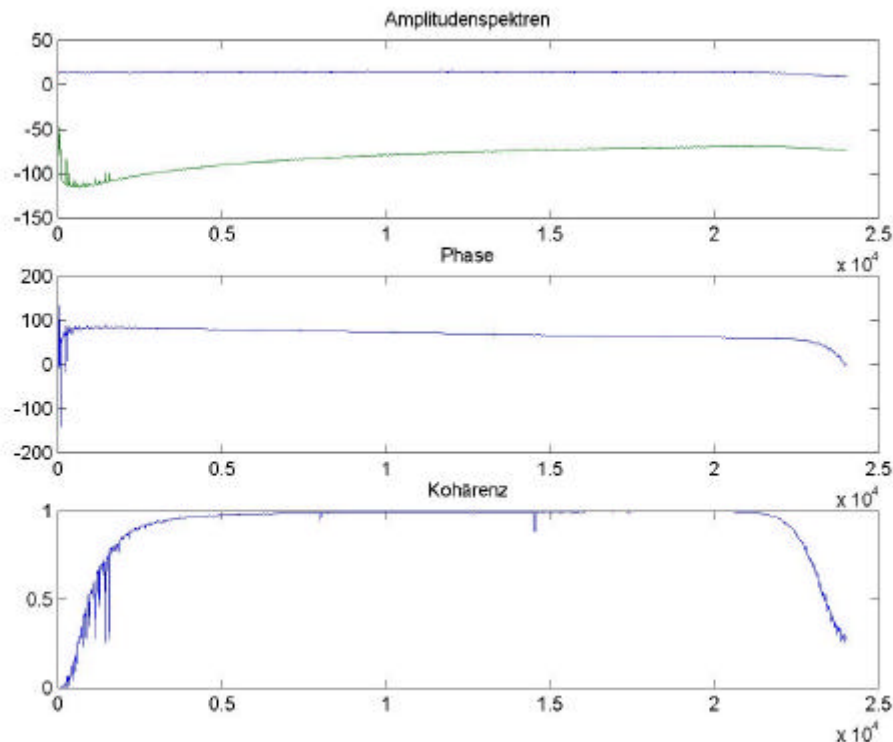


Abb. 5.2.1.3 Phase, Amplitude und Kohärenz der Soundkarte erzeugt mit MATLAB

Bei diesen Diagrammen muss beachtet werden, dass diese nur einseitige Frequenzspektren ohne die negativen Frequenzen darstellen. Hier ist zu sehen, dass die Resultate der Analyse mit MATLAB mit den Resultaten von DASYLab übereinstimmen.

Es zeigt sich auch hier eine Kohärenz von fast 1. Vermutlich liegt hier ein Übersprechen des Kanals 1 auf den zweiten offenen Kanal vor. Um dieses zu überprüfen wurde ein Sinus mit 1000 Hz und einer Amplitude von $1V_{pk}$ auf den 1. Kanal gelegt und die Messung wiederholt.

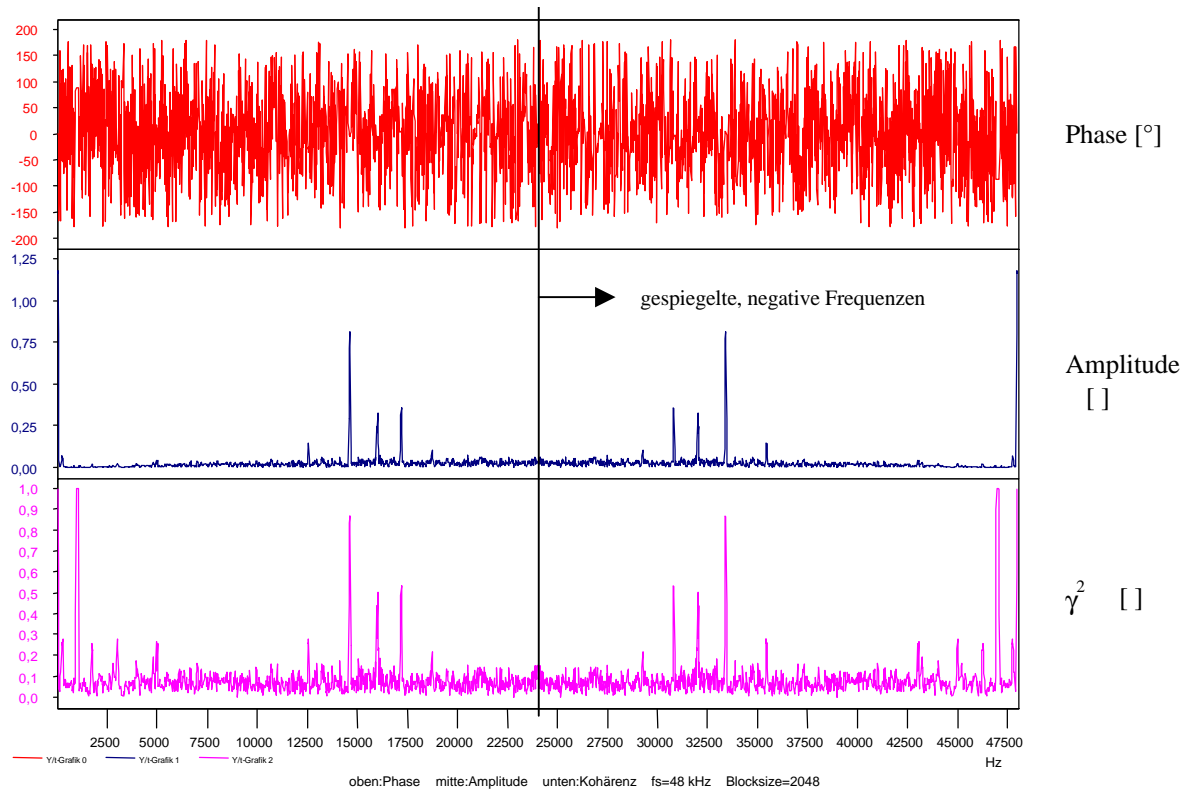


Abb. 5.2.1.4 Phase, Amplitude und Kohärenz der Soundkarte
 f_s : 48 kHz Blocksize: 2048 Mittelungen: 200

Hier ist gut zu erkennen, dass die Phase keine Aussagekraft mehr hat und auch die Amplitude in sehr kleinen Bereichen liegt. Ausnahmen liegen bei etwa 14,5 kHz, 16 kHz und 17,2 kHz vor, deren Herkunft hier unklar bleibt. Es ist vor allem zu erkennen, dass die Kohärenz im Bereich von 1 kHz den Wert 1 annimmt, was auf ein Signal auf Kanal 2 hinweist.

Den Amplitudenspektren ist zu entnehmen, dass sich auf dem 2. Kanal bei etwa 1 kHz ein Signal von ca. 0,0005 V_{pk} befindet. Dies entspricht -66 dB.

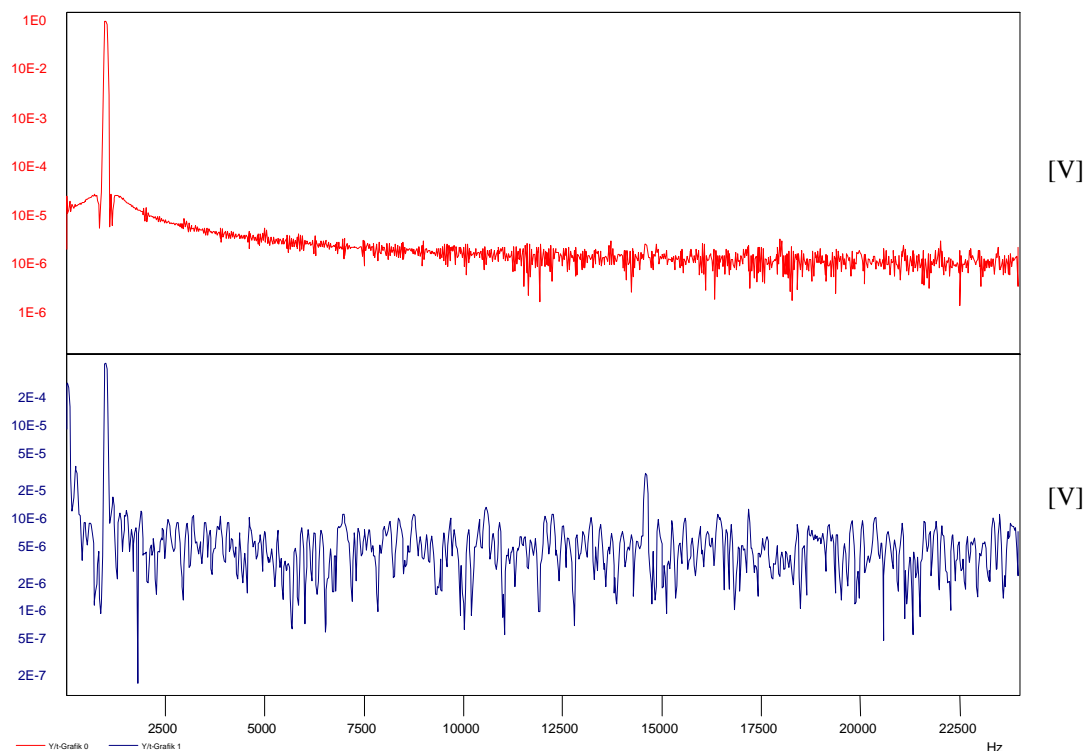
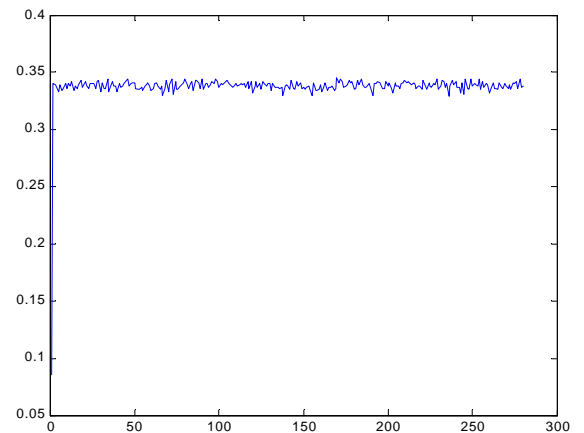


Abb. 5.2.1.5 Amplitudenspektren der Soundkarte
 f_s : 48 kHz Blocksize: 2048 Mittelungen: 200

Ein Übersprechen auf den Kanal 2 ist also vorhanden, was auch durch eine Messung der Kreuzkorrelationskoeffizienten, mit den zuvor aufgezeichneten Wav-Dateien, im Zeitbereich nachgewiesen werden kann.

Kreuzkorrelationskoeffizienten des einseitigen Rauschsignals:

```
clear all
[y,Fs,bits] = wavread('48khz-stereo16b-rauschen-eine');
nfft=8192;
[m,n]=size(y);
NN=round(m/nfft);
%Daten blockweise aufteilen
for i=1:NN;
%sig_1(:,i)=y(((1-1)*nfft+1):(1*nfft),1);
%sig_2(:,i)=y(((2-1)*nfft+1):(2*nfft),2);
sig_1=y(((i-1)*nfft+1):(i*nfft),1);
sig_2=y(((i-1)*nfft+1):(i*nfft),2);
sig_1_mit=sum(sig_1)/nfft;
sig_2_mit=sum(sig_2)/nfft;
zaehler=sum((sig_1-sig_1_mit).*(sig_2-sig_2_mit));
nenner=((sum((sig_1-sig_1_mit).^2))*(sum((sig_2-sig_2_mit).^2)))^0.5;
correlation(i)=zaehler/nenner;
end
plot (correlation)
```

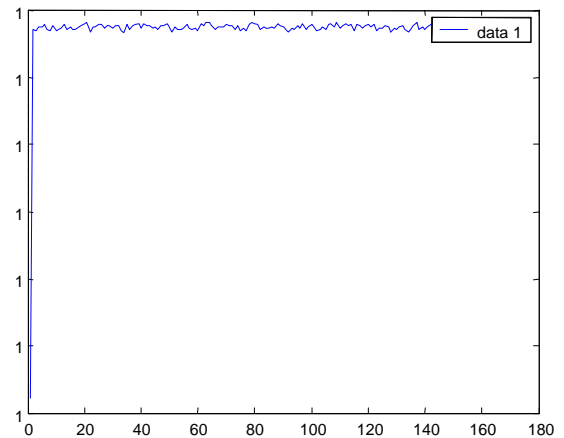


Kreuzkorrelationskoeffizienten des beidseitigen Rauschsignals

```

clear all
[y,Fs,bits] = wavread('48khz-stereo16b-rauschen-beid');
nfft=8192;
[m,n]=size(y);
NN=round(m/nfft);
%Daten blockweise aufteilen
for i=1:NN;
%sig_1(:,i)=y(((1-1)*nfft+1):(1*nfft),1);
%sig_2(:,i)=y(((2-1)*nfft+1):(2*nfft),2);
sig_1=y(((i-1)*nfft+1):(i*nfft),1);
sig_2=y(((i-1)*nfft+1):(i*nfft),2);
sig_1_mit=sum(sig_1)/nfft;
sig_2_mit=sum(sig_2)/nfft;
zaehler=sum((sig_1-sig_1_mit).*(sig_2-sig_2_mit));
nenner=((sum((sig_1-sig_1_mit).^2))*(sum((sig_2-sig_2_mit).^2)))^0.5;
correlation(i)=zaehler/nenner;
end
plot (correlation)

```



Die Kreuzkorrelationskoeffizienten sind ebenfalls ein Maß dafür, inwiefern zwei Signale übereinstimmen und können Werte von -1 bis 1 erreichen. Der Wert 1 steht für eine sehr gute Übereinstimmung, der Wert 0 steht für eine schlechte bzw. keine Übereinstimmung und der Wert -1 steht für eine gute Antikorrelation, d.h. das eine Signal steigt in dem Verhältnis in dem das Andere fällt. Nach dieser Formel werden die Kreuzkorrelationskoeffizienten berechnet.

$$Corr(x, y) = \frac{\sum x_{(i)} - x_x * y_{(i)} - y_y}{\sqrt{\sum (x_{(i)} - x_x)^2 * \sum (y_{(i)} - y_y)^2}}$$

$x_{(i)}$ = Wert an der Position i Kanal 1

$y_{(i)}$ = Wert an der Position i Kanal 2

$x_x = \sum x_i / \text{Blockgröße Kanal 1}$

$y_y = \sum y_i / \text{Blockgröße Kanal 2}$

Sicherlich ist es von Nachteil, wenn ein solches Übersprechen bei dieser Soundkarte vorkommt, jedoch kann die Karte zu Messungen verwendet werden, wenn neben der Kohärenz die Phase und die Amplitude beachtet werden.

5.2.2 Überprüfung der Messkarte auf Phasengenauigkeit

Es wurde hier eine Karte der Firma National Instruments vom Typ AT-MIO-16E-10 getestet. Bei dieser Karte handelt es sich um eine 16 Kanal Messkarte mit einer max. Summenabtastrate von 100 kHz und einer Auflösung von 12 Bit. Diese Karte kann bis zu 16 Kanäle im single-ended, oder 8 Kanäle im differential Mode erfassen.

Hier wurden die Messungen wie mit der Soundkarte wiederholt. Als Signal wurde ein weißes Rausch bis 25,6 kHz mit einem Level von 1 V_{pk} generiert und auf die ersten Beiden der 16 (8) Kanäle gelegt. Die Abtastrate betrug wieder 48 kHz, wobei dieser Wert durch die Software auf 47,9616 kHz geändert wurde und die Blocksize betrug ebenfalls 2048.

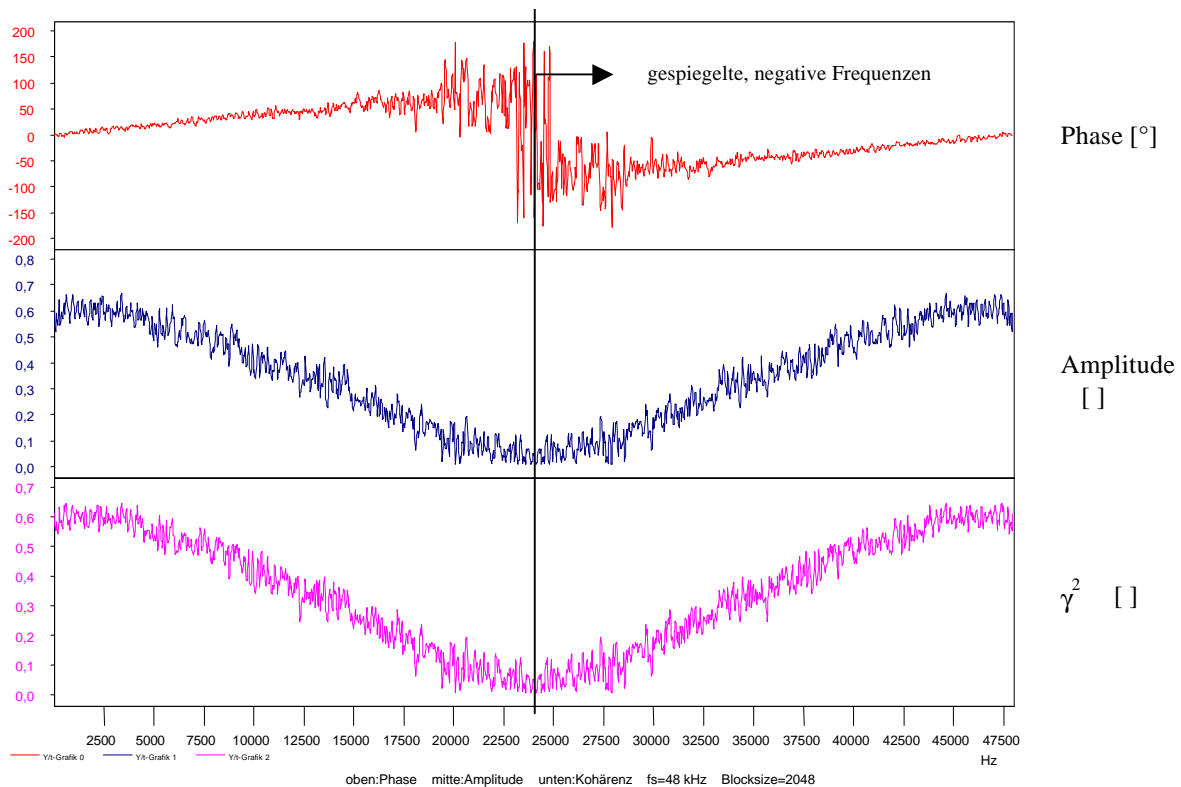


Abb. 5.2.2.1 Phase, Amplitude und Kohärenz der Messkarte
f_s: 48 kHz Blocksize: 2048 Mittelungen: 200

Diese Messung ergab, dass die Phase bis zu einer Frequenz von ca. 16,5 kHz recht linear verlief, jedoch in diesem Bereich bereits Werte bis zu 90° annahm. Die Amplituden- und Kohärenzwerte nahmen mit steigender Frequenz auch recht schlechte Werte ein. Mit zunehmender „Unruhe“ in der Phase wurden die Werte für Amplitude und Kohärenz wesentlich kleiner, da scheinbar keine Übereinstimmung mehr vorlag.

Als nächstes wurde eine Messung durchgeführt, bei der nur am ersten Kanal ein Signal anlag. Die Messbedingungen blieben die selben wie zuvor.

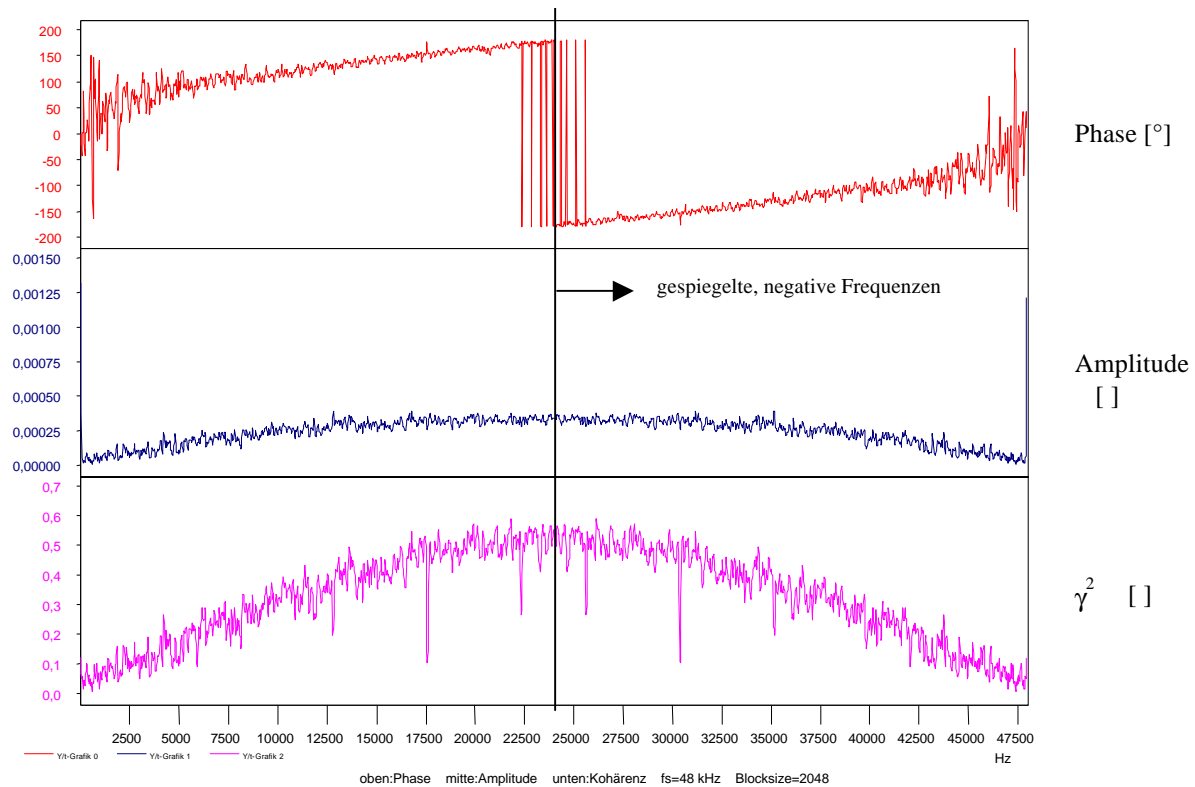


Abb. 5.2.2.2 Phase, Amplitude und Kohärenz der Messkarte
 f_s : 48 kHz Blocksize: 2048 Mittelungen: 200

Es zeigten sich wesentlich größere Werte für die Phase, die besonders in den niedrigen Frequenzen sehr unbestimmt waren. Bereits bei einer Frequenz von 6,5 kHz betrug die Phase durchweg über 90° , was sich bis zur Frequenz von ca. 22kHz auf 171° steigerte. Anhand der Amplitude konnte erkannt werden, dass nur ein sehr kleines Signal auf dem zweiten Kanal vorhanden sein konnte. Ein sehr kleines, eventuell durch Übersprechen entstandenes, Signal könnte auch die Erklärung für die recht hohen Kohärenzwerte sein, die sich bis auf 0,52 bei einer Frequenz von 22 kHz steigerten. Das Amplitudenspektrum zeigt hier Werte im Bereich von $0,00001 V_{pk}$ über den gesamten Frequenzbereich.

Um zu testen ob ein Übersprechen stattgefunden hat, wurde wieder ein Sinus von 1 kHz mit einer Amplitude von 1 V_{pk} generiert und über den ersten Kanal gemessen.

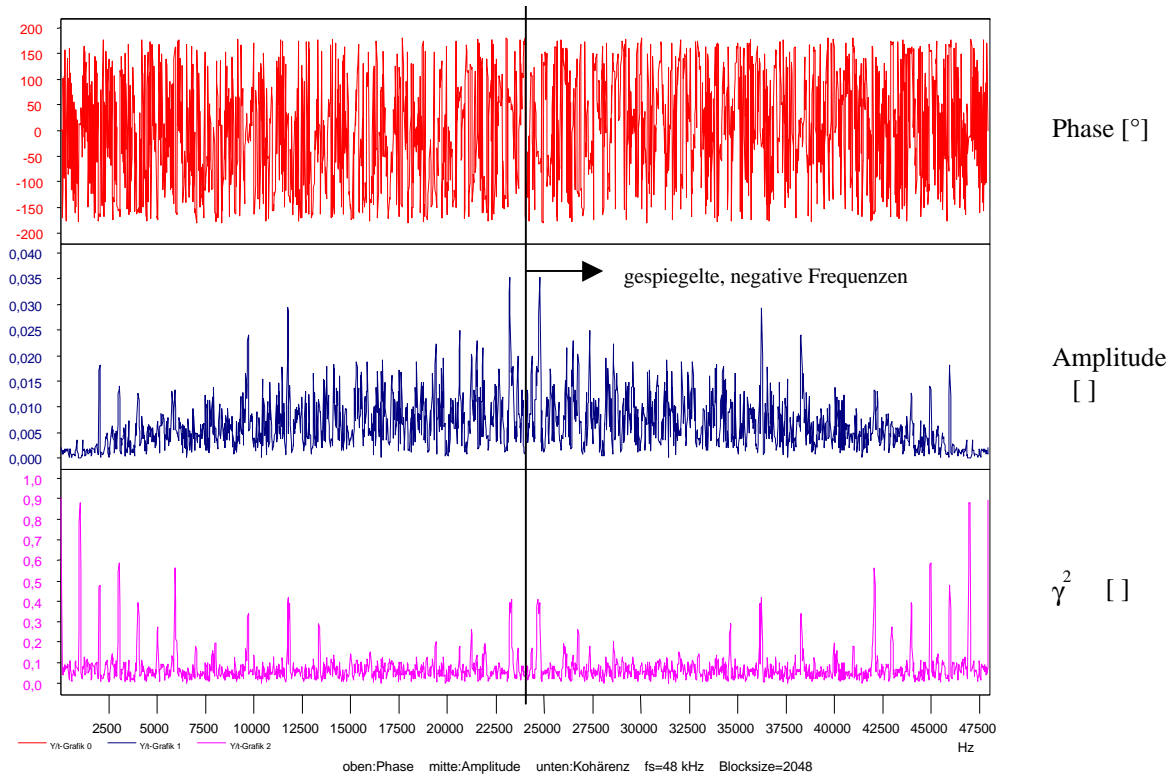


Abb. 5.2.2.3 Phase, Amplitude und Kohärenz der Messkarte
f_s: 48 kHz Blocksize: 2048 Mittelungen: 200

Die Messung ergab, dass sich weder im Amplitudenspektrum bei 1 kHz noch bei der Amplitude der Übertragungsfunktion ein nennenswerter Wert ergab. Jedoch stellte sich bei der Kohärenz ein Wert von 0,87 ein. Ein Übersprechen scheint es bei der Messkarte AT-MIO-16E-10 nur im geringsten Masse zu geben, woher die Kohärenz kommen könnte. Leider zeigt diese Karte eine nicht so gute Phasengenauigkeit, was ihre Verwendung für Übertragungsfunktionen oder Phasenmessungen einschränkt.

6. Vergleich der Messergebnisse

Im Rahmen der Diplomarbeit wurden neben der Programmierung der Frequenzanalysatoren unter DASyLab und LabVIEW Soundkarten auf ihre Verwendbarkeit als Messkarten untersucht.

6.1 Resultate der Messungen mit Soundkarten

Die Messungen zeigten, dass die getestete Soundkarte vom Typ DMX Xfire 1024 der Firma TerraTec ein sehr gutes lineares Übertragungsverhalten zeigt. Die zweite getestete Soundkarte der Firma Creative Labs, Typ SB 128 PCI, zeigte eine Welligkeit, deren Ursache leider nicht festgestellt werden konnte. Bei den anschließenden Untersuchungen der Phasentreue wurde die SB 128 PCI nicht weiter getestet, da diese Welligkeit die Messergebnisse verfälscht hätte. Die Untersuchung auf Phasentreue zeigte, dass bei der Soundkarte DMX Xfire 1024 eine maximale Phase von $0,18^\circ$ bei etwa 22,3 kHz auftrat, was einen sehr guten Wert darstellt. Auch für die Amplitude und die Kohärenz ergaben sich sehr gute Werte.

Leider zeigte sich bei der Soundkarte DMX Xfire 1024 mit einem Abstand von 66 dB ein leichtes Übersprechen auf den zweiten Kanal.

6.2 Resultat der Messungen mit der Messkarte

Die Überprüfung der Messkarte AT-MIO-16E-10 der Firma National Instruments zeigte leider eine nicht so gute Phasentreue, was durch die asynchron Abtastung der Kanäle bedingt ist. Ein Übersprechen war hier nicht messbar, jedoch zeigte sich bei den Messungen eine Kohärenz, die nicht erklärbar war.

7. Zusammenfassung

Die vorliegende Diplomarbeit sollte zeigen, wie mit den Programmen DASyLab und LabVIEW ein Frequenzanalysator erstellt werden kann. Darüber hinaus wurden zwei Soundkarten und eine professionelle Messkarte hinsichtlich ihrer Verwendbarkeit für einen Zweikanal Frequenzanalysator untersucht. Es wurden eine Soundkarte der Firma Creative Labs vom Typ Soundblaster 128 PCI, eine Soundkarte der Firma TerraTec Typ DMX Xfire 1024 und eine Messkarte der Firma National Instruments mit der Bezeichnung AT-MIO-16E-10 untersucht.

Die Größenordnung für die Anschaffung der Karten beliefen sich für die Soundblaster 128 PCI bei 45,- DM, die der DMX Xfire 1024 bei 120,- DM und bei der Messkarte AT-MIO-16E-10 bei 2500,- DM.

Es zeigte sich bei der Untersuchung der Soundkarten, dass Karten einer gewissen Qualität für Messungen hinsichtlich der Phase oder Übertragung recht brauchbar erscheinen. Die Soundkarte SB 128 PCI zeigte beim Übertragungsverhalten eine Welligkeit im Frequenzspektrum, was eine Verwendung als Messkarte ausschließt. Ein Übersprechen, mit einem Abstand von 66 dB bei der Soundkarte von TerraTec (DMX Xfire 1024) schränkt eine Verwendung als Messkarte hinsichtlich der Kohärenz zwar ein, macht eine Verwendung aber nicht unbedingt unmöglich. Soll z.B. bei einer Übertragungsfunktion die Phase bestimmt werden, so müssen auch die Amplitude und Kohärenz beachtet werden, um zu sehen, ob diese Signale korrelieren.

Der Verlauf der Phase verhält sich bei der DMX Xfire 1024 bis zu einer Frequenz von 22,5 kHz nahezu linear. Das ist bei Messungen von Vorteil, da keine aufwendigen Kalibrierfunktionen nötig sind. Ein Fortsetzen der Untersuchungen in diese Richtung ist notwendig.

Die Untersuchung der Messkarte AT-MIO-16E-10 zeigt, dass diese eine nicht so gute Phasentreue hat, was durch die asynchrone Abtastung der Kanäle bedingt sein kann. Bei dieser Messkarte sind die Kanäle wesentlich besser getrennt, jedoch war die Kohärenz bei Messungen sehr hoch, was nicht vorkommen sollten.

DASyLab und LabVIEW sind beides Programme, die durch modulare Strukturen das Programmieren von Instrumenten für Messsysteme am PC ermöglichen. DASyLab ermöglicht ohne viel Vorkenntnisse ein Programm zu erzeugen, da es in vielen Funktionen intuitiv bedient werden kann. Leider sind die meisten Module hinsichtlich ihrer Funktion nicht zu überprüfen, da sie wie eine Blackbox abgeschlossen sind. LabVIEW hingegen erlaubt, jedes Vi (Modul) in seiner Struktur bis in unterste Ebenen zu untersuchen. LabVIEW erfordert in seiner Bedienung mehr Vorkenntnisse, z.B. bezüglich der Variablentypen für die Verbindungen der Vi's.

Die Umsetzung des Frequenzanalysators mit DASyLab und LabVIEW zeigt, dass es bei beiden Programmen noch Verbesserungsansätze hinsichtlich Funktion und Visualisierung gibt. Auch die Dokumentation (Online-Dokumentation) hat hier und da Lücken.

Zusammenfassend lässt sich sagen, dass Soundkarten der mittleren bzw. oberen Preiskategorie als Messkarten in Zusammenarbeit mit einem Personalcomputer zu Messzwecken in Betracht kommen.

Durch die weite Verbreitung von PC's und deren Peripherie, Monitor als Ausgabegerät, Drucker, Plotter, Speichermedien, lässt sich ohne großen finanziellen und organisatorischen Aufwand für viele Anwendungsgebiete ein akzeptables Messsystem aufbauen. Besonderer Wert muss allerdings auf die Auswahl der eingesetzten Soundkarte gelegt werden.

8. Literaturverzeichnis

- [1.] H. J. Bartsch, Taschenbuch mathematischer Formeln, Fachbuchverlag, 1994, 16. Aufl.
- [2.] E.Oran Brigham, FFT (Einführung in die Nachrichtentechnik), Oldenbuorg Verlag, 1982.
- [3.] Younes Chlyeh, Numerische Simulation und experimentelle Überprüfung von Wanddruckschwankungsverläufen, 2000
- [4.] Rüdiger Hoffmann, Grundlagen der Frequenzanalyse, expert Verlag, 2001
- [5.] Frank Kameier / Dieter Reinartz, Strömungsakustik Skript, 2000
- [6.] Karrenberg, Die Welt der Signale-Prozesse-Systeme, ASAT-Verlag, 2000, 1.Aufl.
- [7.] Martin Mayer: Signalverarbeitung; Vieweg Verlag, 1998, 2. Aufl.
- [8.] Rahman Jamal u. Herbert Pichlik, LabVIEW, Programmiersprache der vierten Generation, Prentice Hall Verlag, 1997

9. Anhang

Anregungen zur Verbesserung von DASYSLab und LabVIEW

DASYSLab

1. Modul „Datenfenster“

Im Modul „Datenfenster“ muss die Vektorlänge eingegeben werden. Es wäre hilfreich, wenn hier die Globale Variable „BLOCK_SIZE“ als default Wert eingestellt wäre.

Des weiteren kann in diesem Modul die Option „Amplituden Korrektur“ aktiviert werden, der Hilfefunktion kann man entnehmen, dass diese das gewichtete Signal bezüglich des Energiegehalts korrigiert, jedoch wird nicht angegeben mit welchen Faktoren gerechnet wird. Diese Amplitudenkorrektur wird unter LabVIEW als Kohärente Verstärkung bezeichnet. Ohne die Amplitudenkorrektur wird das Spektrum in einer „Normierten Form“, wie sie auch in MatLab verwendet wird, ausgegeben. Damit könnte die Bezeichnung der Amplitudenkorrektur auch in „Normiert“ bzw. „Nicht Normiert“ geändert werden. Für eine korrekte Programmierung eines Frequenzanalysators muss das nicht normierte Fenster verwendet werden. Ebenso werden die zu den Fensterfunktionen gehörenden ε -Werte nicht angegeben, die für die Berechnung des Gesamtpegels benötigt werden.

2. Modul „FFT“

Das Leistungsspektrum wird einseitig ausgegeben und berechnet sich aus dem Quadrat des Fourier-Spektrums. Das Quadrat müsste nach meinem Verständnis aus einem Fourier-Spektrum gebildet werden, welches, durch Addition der negativen und positiven Frequenzen (mit Ausnahme des Gleichspannungsanteils), in ein Einseitiges transformiert wurde (vgl. Rahman Jamal/ Herbert Pichlik LabVIEW Programmiersprache der vierten Generation S.230). DASYSLab bildet das Leistungsspektrum offensichtlich nur aus den positiven Frequenzanteilen.

3. Modul „Übertragungsfunktion“

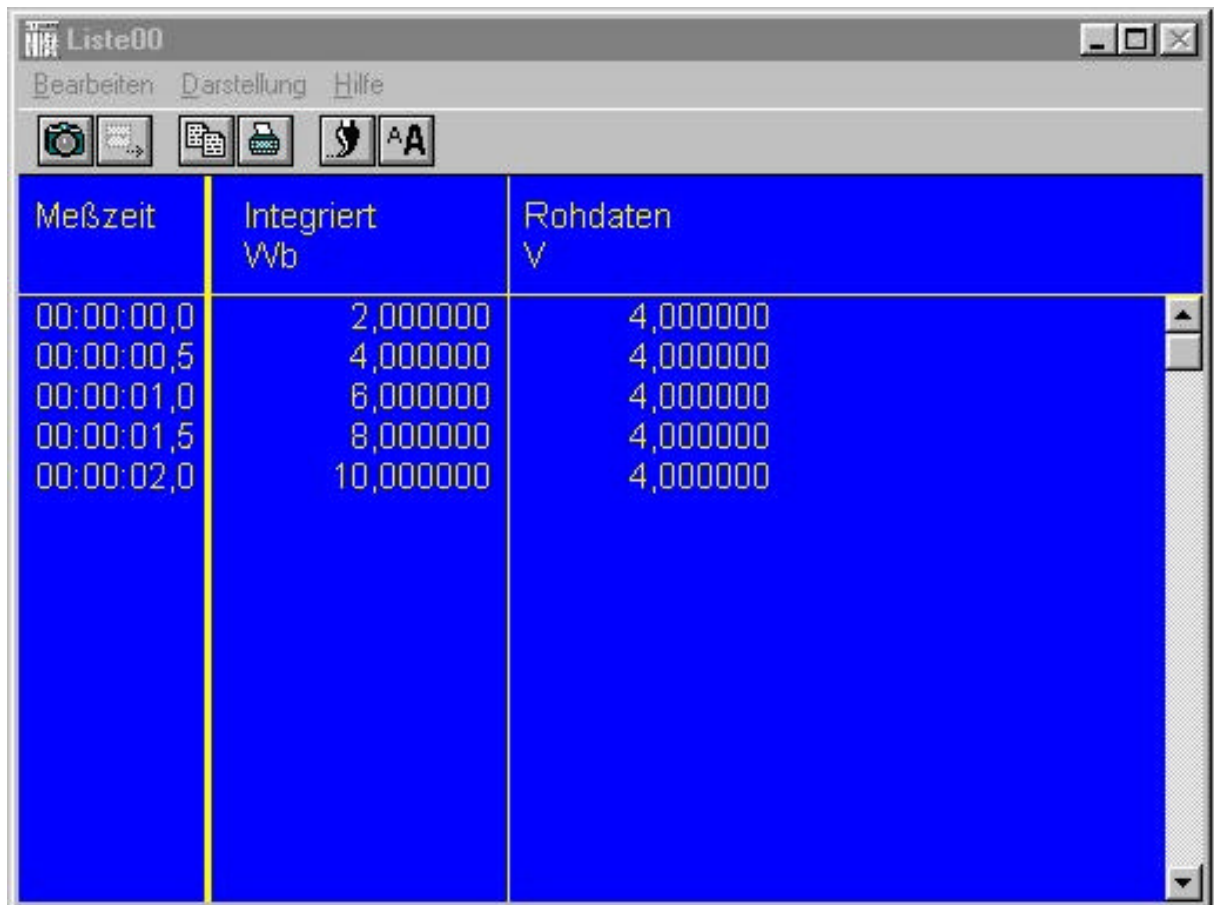
Im Modul „Übertragungsfunktion“ werden drei Verfahren für die Übertragungsfunktion bereitgestellt. In der Hilfefunktion wird nicht erläutert, welches der Verfahren welche Vorteile hat. Es zeigten sich bei Versuchen Störungen im Phasenverlauf, wenn die Funktion $F_{xy}(f) = F_y(f)/F_x(f)$ verwendet wurde, eine Erklärung hierfür ist nicht vorhanden.

4. Modul „Integration“

Das Modul „Integration“ arbeitet nach folgender Formel: $I = \int_a^b f(x) \cdot dx = \left(\sum_{i=1}^N f_i \right) dx$ wo-

bei dx für den Zeitabstand zwischen zwei aufeinanderfolgenden Messwerten steht. Ein genauerer Wert für das Integral würde sich nach der Trapezregel für das Integral $I = \left(\sum_{i=1}^{N-1} \frac{f_i + f_{i+1}}{2} \right) dx$ ergeben.

So ergibt z.B. die Integration einer Konstanten $\int_0^2 a \cdot dt$ über 2 Sekunden mit $a=4$ einen Wert von 10 Vs anstelle der zu erwartenden 8 Vs , vgl. folgende Abbildung A.4.1 .



Meßzeit	Integriert Wb	Rohdaten V
00:00:00,0	2,000000	4,000000
00:00:00,5	4,000000	4,000000
00:00:01,0	6,000000	4,000000
00:00:01,5	8,000000	4,000000
00:00:02,0	10,000000	4,000000

Abb.A.4.1 Daten der Überprüfung der DasyLAB Integration

Dieses Beispiel zeigt dass bereits zum Zeitpunkt 0 ein Wert von 2 V^2 ergibt, was nicht sein kann.

Siehe auch Programm: „Integral_Probe.DSB“.

5. Modul „Formelinterpreter“

Im Modul „Formelinterpreter“ muss an Stelle des Kommas (,) ein Punkt (.) eingegeben werden, bei der Eingabe der Abtaste (Messeinstellungen) oder im „Generatormodul“ jedoch das Komma. Ein einheitliches Dezimaltrennzeichen ist für die Bedienung notwendig.

6. Modul „Y/t Grafik“

Es wird bei der Anzeige mehrerer Signale in einem Fenster nur eine Skalierung angezeigt. Die Skalierung der Signale kann, z.B. bedingt durch Autoskalierung, voneinander abweichen. Eine Zuweisung mehrerer Signale zu einer Skalierung wäre sehr sinnvoll.

7. Modul „Generatormodul“

Die Amplituden können nur als peak-Wert angegeben werden. Eine Auswahl, die eine Ausgabe als rms-Wert ermöglicht, sollte integriert werden. Die Angabe, dass es sich um einen peak-Wert handelt, ist erforderlich

8. Modul „Mittelung“

Das Modul Mittelung gibt bei Blockweiser Mittelung am Ausgang die gleiche Anzahl an Werten aus wie am Eingang. Um einen über einen Block gemittelten Wert zu erhalten, muss die Anzahl der zu mittelnden Werte hoch gesetzt werden (z.B. $4 \rightarrow 512$). Siehe auch Programm: „Mittelung.DSB“. Bei blockweiser Mittelung sollte daher der Wert für die Anzahl der zu mittelnden Werte auf „Blocksize“ voreingestellt sein.

9. Modul „Arithmetik“

Die Funktion Aufsummieren funktioniert nur so, dass die Werte fortlaufend addiert werden und nicht blockweise. Sieh auch Programm „Arithmetik_Aufsummieren.DSB“.

10. Treiber für Soundkarten

Die Aussteuerung für die Soundkarte ist bei $\pm 5V$ empfindlicher als bei $\pm 1V$, vgl. Seite 34 bis 37.

11. Phase im Kreuzleistungsspektrum

Die Ausgabe des Kreuzleistungsspektrums geschieht nur als Real- und Imaginärteil, jedoch nicht als Amplitude und Phase. Aus Real- und Imaginärteil kann nach den Formeln unter 2.1.5 die Amplitude und die Phase berechnet werden. Die Phase kann so allerdings nur für Winkel von $\pm 90^\circ$ bestimmt werden. Eine Option zur Ausgabe in Amplitude und Phase ist hier notwendig. Vgl. Programm „Kreuzspektrum_Phase.DSB“.

LabVIEW

1. Onlinedokumentation

In der Onlinedokumentation werden die Funktionen der Übertragungsfunktion im Modul „Übertragungsfunktion“ nicht ausführlich beschrieben, es wird nur erwähnt, dass es drei verschiedene Verfahren gibt.

9.1 Anhang

Liste der auf der CD beigefügten Programme.

Programme der Diplomarbeit:

- Arithmetik_Aufsummieren.DSB
- Frequenzanalysator_generator_mittelung_gp.DSB
- Frequenzanalysator_messkarte.DSB
- Frequenzanalysator_messkarte_mittelung.DSB
- Frequenzanalysator_messkarte_mittelung_gp.DSB
- Frequenzanalysator_soundkarte.DSB
- Frequenzanalysator_soundkarte_mittelung.DSB
- Frequenzanalysator_soundkarte_mittelung_gp.DSB
- Gesamtpegel.DSB
- Integral_Probe.DSB
- Korrekturwertermittlung.DSB
- Kreuzspektrum_Phase.DSB
- LabView_Frequenzanalysator_Messkarte.vi
- Mittelungen.DSB
- Mittelungen_Zeit_Frequenz.DSB
- test_uebertragungsverhalten.DSB
- Uebertplott_ampl.m
- Uebertplott_db_differenz.m

9.2 Anhang

Speichern von Daten mit dem HP Dynamic Signal Analyzer 35665A

Um nach erfolgter Messung die Daten weiter zu bearbeiten, ist oft eine Übertragung zu einem PC erforderlich.

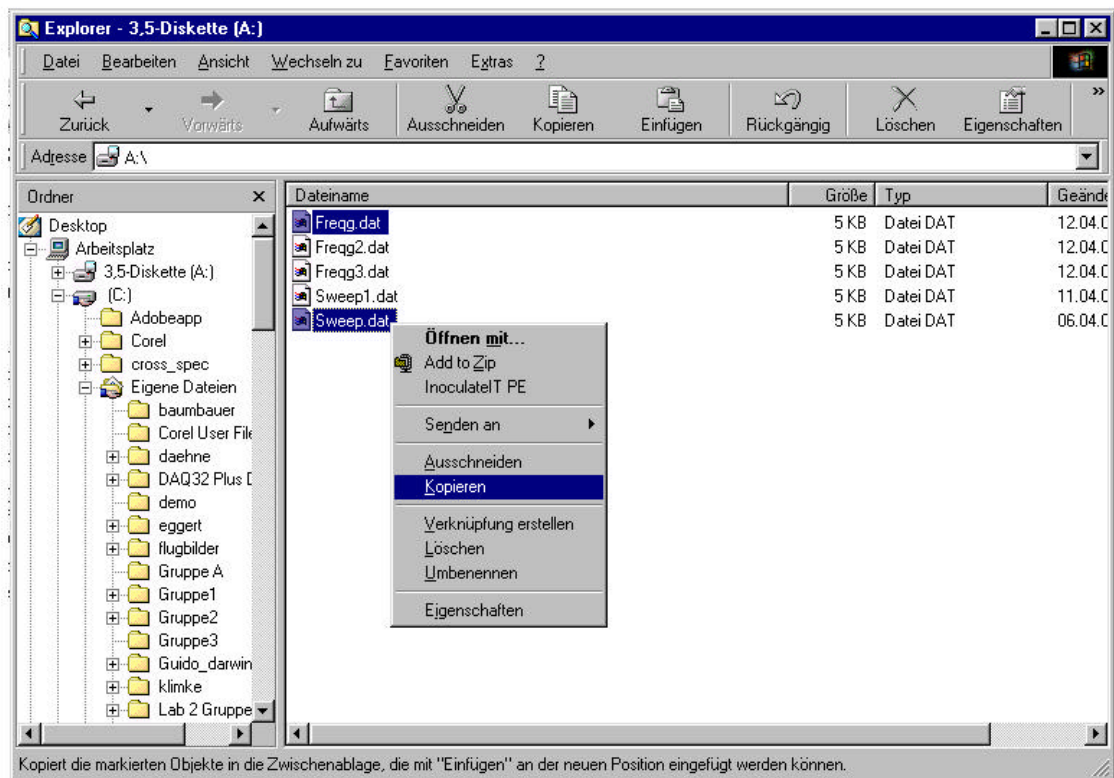
Dieses kann recht einfach mit einer Diskette im 3 1/2 Zoll Format geschehen.

Die Vorgehensweise ist folgende:

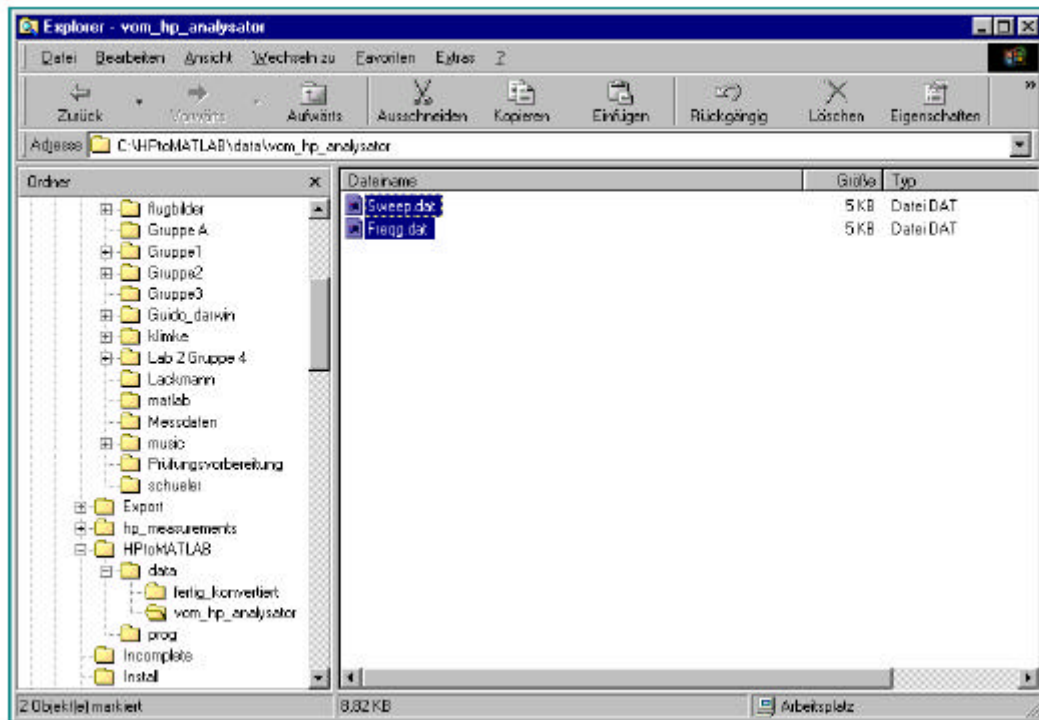
1. Einlegen einer formatierten 3 1/2 Zoll Diskette in das Laufwerk des Analysators.
2. Im Menü SYSTEM die Taste [Save/Recall] drücken.
3. Die Taste F1 für (SAVE DATA) drücken.
4. Anschließend z.B. die Taste F1 für (SAVE TRACE) drücken.
An dieser Stelle ist es auch möglich, die Funktionen F3 (SAVE CAPTURE), F4 (SAVE WATERFALL) oder F5 (CONTINUE SAVE) auszuwählen.
5. F9 (INTO FILE) zum Speichern in eine Datei drücken.
6. Den vorgeschlagenen *Dateinamen.DAT* nach Belieben verändern, die Eingabe erfolgt über die Bedienungstasten rechts am Gerät.
7. Die Taste F1 (ENTER) drücken. Die Daten werden nun auf der Diskette gespeichert.

Bearbeitung mit MATLAB

Die Daten sollten zunächst auf den PC in ein Verzeichnis ihrer Wahl überspielt werden. Z.B. Kopieren mit dem Explorer.



Die Daten müssen in ein Verzeichnis eingefügt werden, welches uns nachher als Quellenverzeichnis dient.



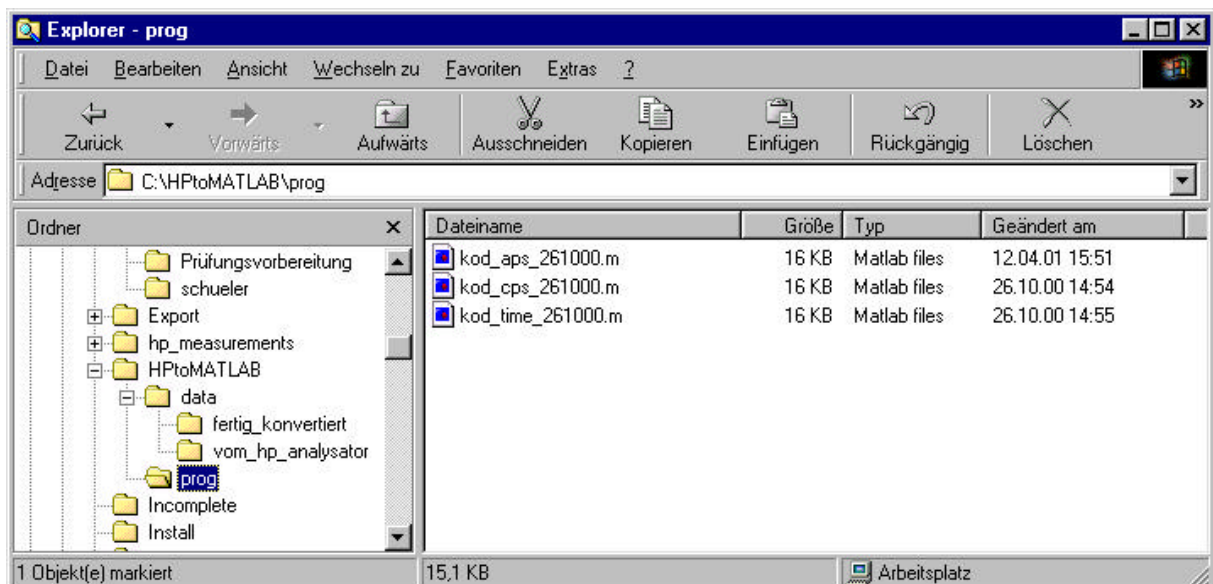
Weiterhin muss ein Verzeichnis angelegt werden, in dem die konvertierten Daten abgelegt werden können.

Damit MATLAB die Daten in ein passendes Format konvertiert kann, müssen die Dateien *sdprint.exe* und *sdftoml.exe* im Verzeichnis C:\Windows vorhanden sein.

Zum konvertieren wurden Programme von Thorsten Frey, A. Kleinfeld und Prof. Kameier, geschrieben.

Diese Programme sollten in einem Ordner, z.B.: C:\HPtoMATLAB\prog, vorliegen. Ist das nicht der Fall, so sollte man sich an Herrn Kameier oder einen seiner Mitarbeiter wenden.

Diese Programme unterscheiden sich im wesentlichen in den Skalierung für die jeweiligen Anwendungen.



Soll z.B. ein Autopowerspectrum APS konvertiert werden, so muss man die Programmvorlage *kod_aps_261000.m* verwenden.

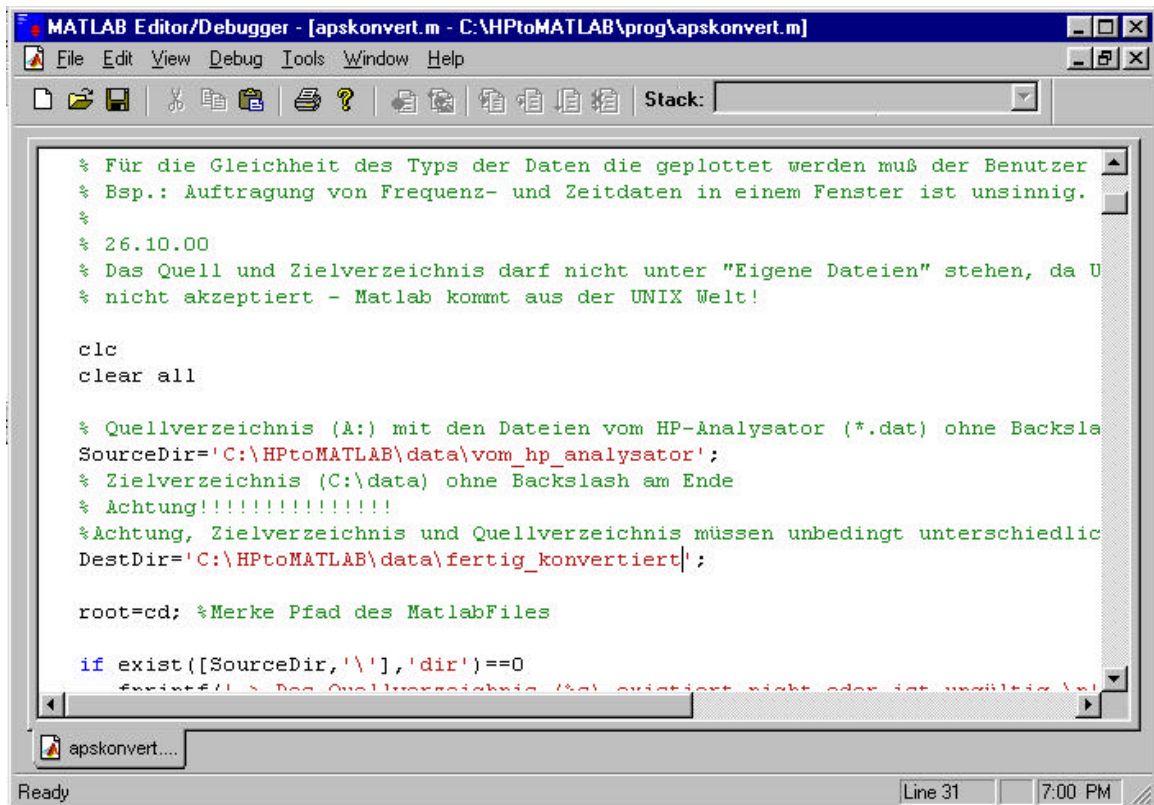
Durch einen Doppelklick auf diese Datei öffnet sich der Editor, und nun muss man diese folgendermaßen anpassen:

In Zeile 27 muss die „SourceDir=“ Angabe auf das Verzeichnis verweisen in dem die „Quelldateien.Dat“ stehen.

In Zeile 31 muss die „DestDir=“ Angabe auf das Verzeichnis verweisen in dem die „Zieldateien.mat“ stehen sollen.

Diese veränderte Datei kann man nun unter einem anderen Namen abspeichern!

Anschließend den EDITOR schließen.



```
MATLAB Editor/Debugger - [apskonvert.m - C:\HPtoMATLAB\prog\apskonvert.m]
File Edit View Debug Tools Window Help
Stack:
% Für die Gleichheit des Typs der Daten die geplottet werden muß der Benutzer
% Bsp.: Auftragung von Frequenz- und Zeitdaten in einem Fenster ist unsinnig.
%
% 26.10.00
% Das Quell und Zielverzeichnis darf nicht unter "Eigene Dateien" stehen, da U
% nicht akzeptiert - Matlab kommt aus der UNIX Welt!

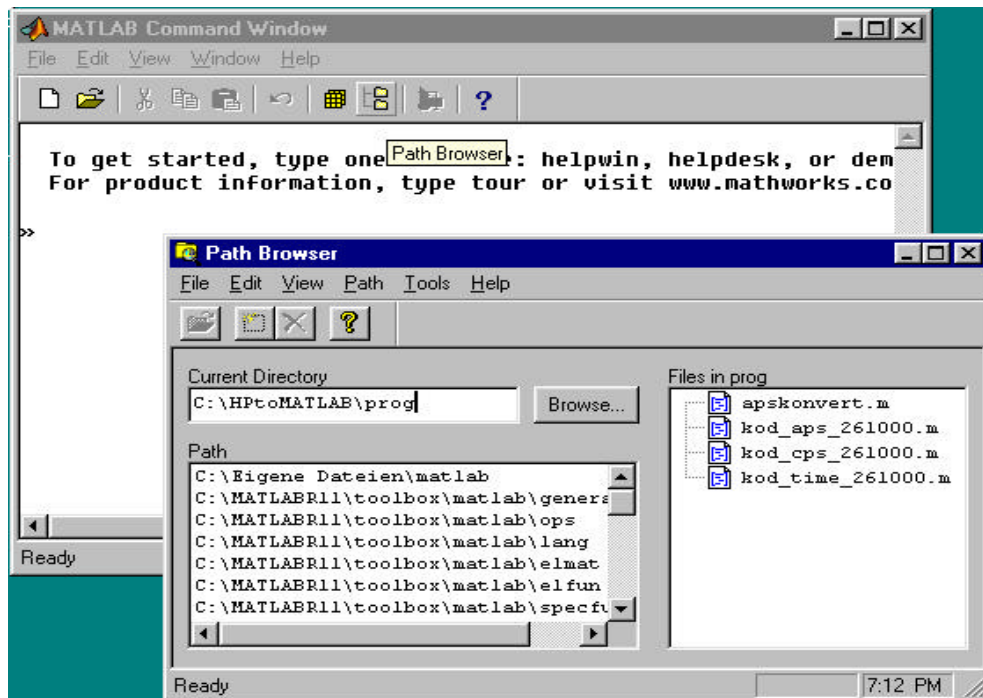
clc
clear all

% Quellverzeichnis (A:) mit den Dateien vom HP-Analysator (*.dat) ohne Backsla
SourceDir='C:\HPtoMATLAB\data\vom_hp_analysator';
% Zielverzeichnis (C:\data) ohne Backslash am Ende
% Achtung!!!!!!!!!!!!!!!!!!!!
%Achtung, Zielverzeichnis und Quellverzeichnis müssen unbedingt unterschiedlic
DestDir='C:\HPtoMATLAB\data\fertig_konvertiert';

root=cd; %Merke Pfad des MatlabFiles

if exist([SourceDir,'\'],'dir')==0
    fprintf('! Das Quellverzeichnis (A) existiert nicht oder ist unvollständig!\n')
```

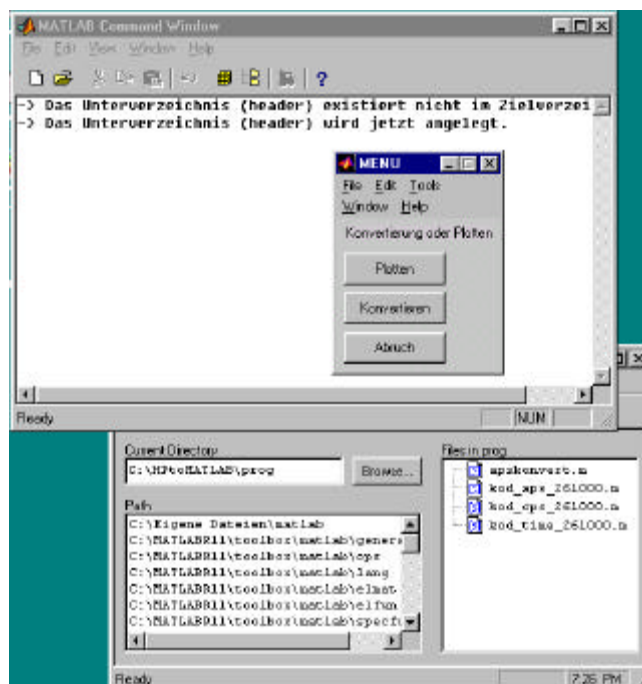
Um mit der eigentlichen Konvertierung zu beginnen, muss MATLAB gestartet werden. In MATLAB muss über das Menü „Path Browser“ der Pfad zu dem Verzeichnis angegeben werden, in dem das Programm (apskonvert.m) liegt, siehe Bild.



Der Aufruf des Programms im MATLAB „Command Window“ startet dieses (hier z.B.: apskonvert im Eingabefenster eingeben).

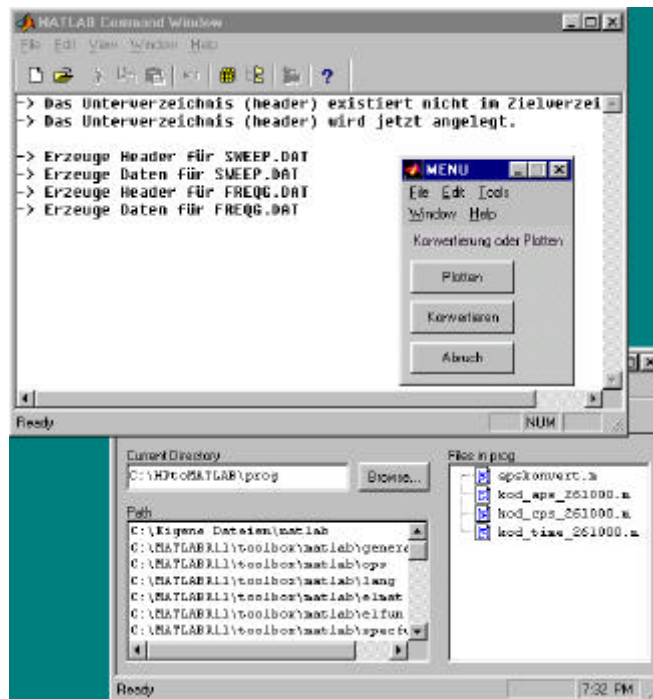
Man startet dieses Skript auch über die Funktionen: File → Run Script → Browse... → *klick auf apskonvert.m* → [öffnen].

Es erscheinen u.U. die ersten Meldungen und es öffnet sich ein weiteres Fenster „MENU“.



Die Auswahl von [Konvertieren] unter „MENU“ startet den Konvertiervorgang, wobei es nötig ist, die Enter Taste mehrmals zu drücken.

Achtung: Jeder Neustart der Konvertierung löscht alle bisher konvertierten Daten.

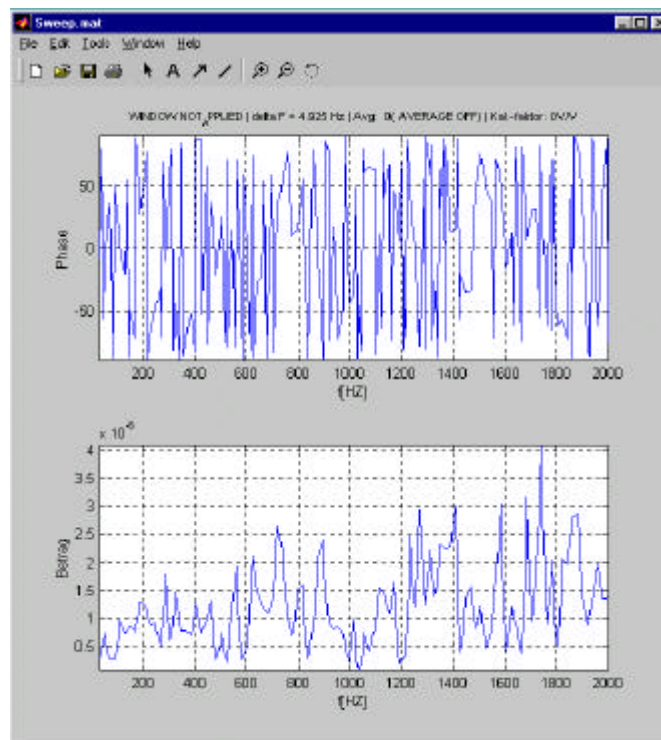


Als nächstes kann man nun im Fenster MENU die Funktion Plotten aufrufen.

Es erscheint ein Fenster in dem man die gewünschte Datei, die geplottet werden soll, auswählen kann.



Über *Öffnen* wird der Plott (hier Sweep.mat) nun ausgeführt.



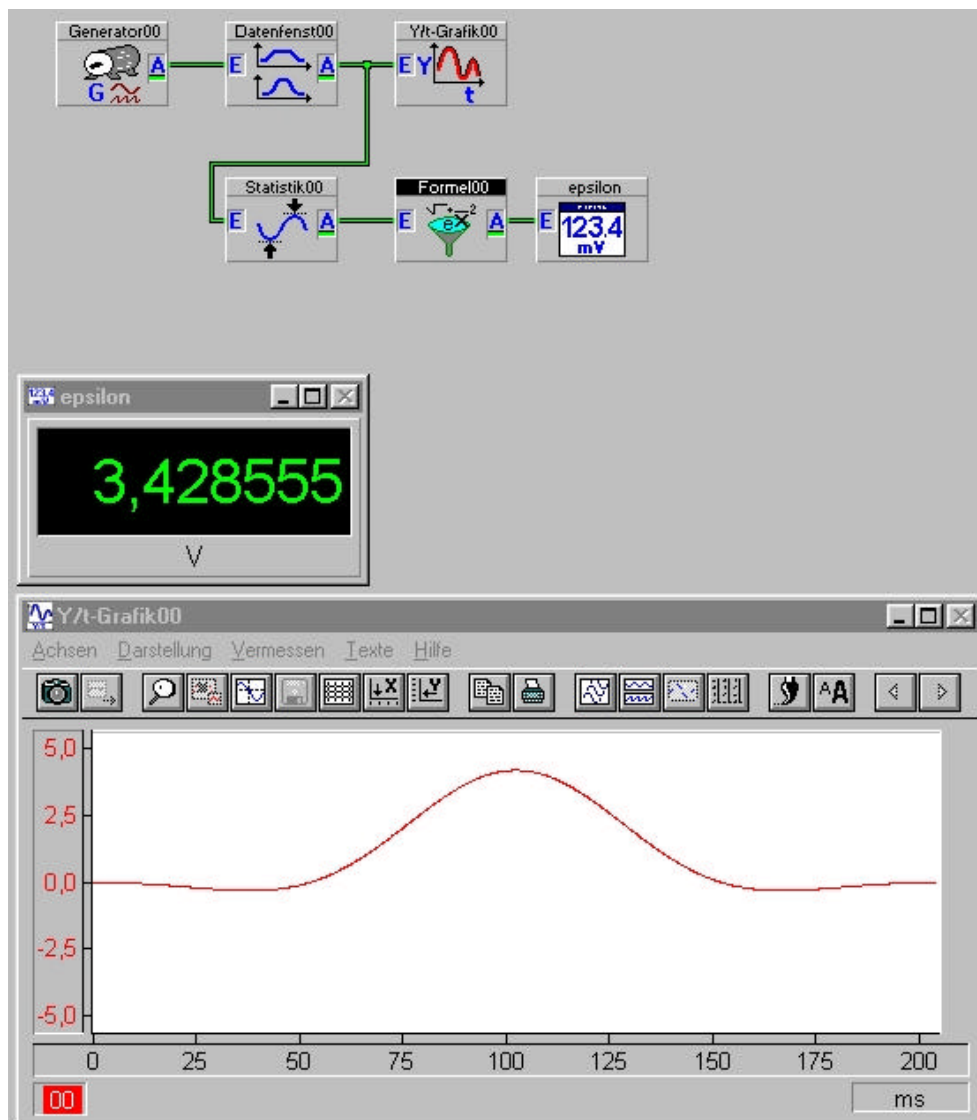
Wechselt man nun in das MATLAB „Command Window“ und drückt die RETURN Taste so erscheint erneut das MENU und man hat die Auswahl ob man einen [NEU]en Plott erstellen möchte, oder ob man dem vorhandenen Plott einen weiteren Kurvenverlauf [Hinzufügen] möchte. Mit [ENDE] wird das Konvertierprogramm beendet.

9.3 Anhang

Ermittlung des Korrekturwertes ϵ der Fensterfunktionen von DASYLab

Um die Korrekturwerte der einzelnen Fensterfunktionen, insbesondere derer die nicht einheitlich festgelegt sind, wurde unter DASYLab ein Programm erzeugt, Korrekturwertermittlung.DSB, in dem die Effektivwerte der ungewichteten Signale mit den Effektivwerten der gewichteten Signale verglichen worden. Als Signalquelle diente der programmeigene Signalgenerator. Die Signale wurden mit den verschiedenen Fensterfunktionen gewichtet und anschließend mit dem Statistik Modul der Effektivwert über die Blocklänge gebildet. Daraufhin wurden im Formelinterpreter die Werte der gewichteten Signale mit denen der ungewichteten Signale dividiert und zuletzt quadriert (Betrachtung der Leistung), um den Korrekturwert ϵ zu erhalten. Die Formel für ϵ lautet somit:

$$\epsilon = \frac{\sqrt{\frac{1}{n} \sum_{i=1}^n f(i)^2}}{\sqrt{\frac{1}{n} \sum_{i=1}^n f(x)^2}}$$



Die Ergebnisse sind in der Tabelle 2.1 auf Seite 9 zusammengefasst.