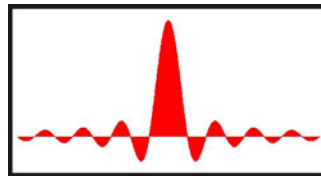


Digitale Filter

Grundlagen und Programmierung



Prof. Dipl. Ing. H.W. Neuschwander

Stand: 30.09.2010



**Fachhochschule
Kaiserslautern**
University of Applied Sciences

**Angewandte
Ingenieurwissenschaften**
Kaiserslautern

Einleitung

Dieses Manuskript dient der Einführung in die Theorie und Programmierung digitaler Filter. Um die bei der Berechnung digitaler Filter erforderlichen mathematischen Berechnungen durchführen zu können und um die Ergebnisse auch grafisch darstellen zu können benötigen wir spezielle Software. In der Regel verwendet man das recht teure Programmpaket *Matlab* von der Fa. "The Mathworks".

Es gibt allerdings eine kostenlose Alternative und zwar das Programm *SCILAB*. Funktionalität und Syntax von SCILAB sind dabei zu einem großen Teil identisch mit Matlab. Im Labor arbeiten wir ausschließlich mit SCILAB. Sie finden das komplette Programmpaket (ca.75 MB) unter ...



Für die Durchführung der Übungen und für die Klausurvorbereitung sollten Sie sich die Software auf Ihren eigenen PC herunterladen und installieren und sich natürlich in die grundlegende Bedienung etwas einarbeiten. Die Installation ist problemlos. Nach dem Programmstart erscheint das in Abbildung 1 dargestellte Fenster (hier mit der Version 5.2.2 vom August 2010).

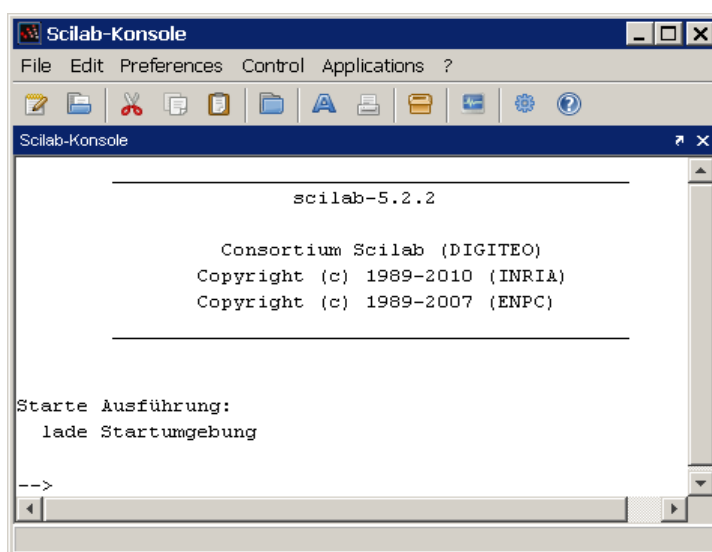


Abbildung 1: Startmenü von SCILAB Version 5.2.2

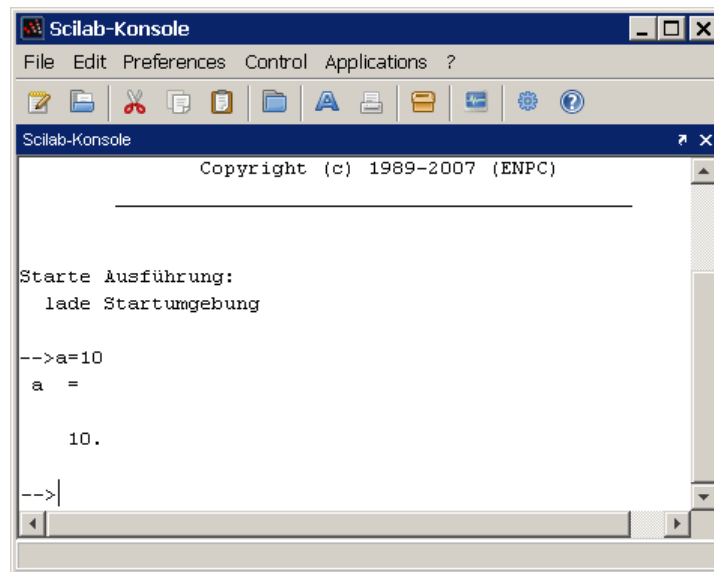


Abbildung 2: Eingabe von Kommandos in SCILAB Version 5.2.2

Nach dem Programmstart können wir bereits direkt Kommandos eingeben (rechts neben dem Pfeil in Abbildung 2). Die Eingabe von „a=10“ liefert z.B. das in der obigen Abbildung gezeigte Ergebnis.

Die im Manuskript vorhandenen Beispiele, deren Listings über die Seite ...

[http://www.fh-kl.de/~hans.neuschwander/
download_skripte/download_difil.html](http://www.fh-kl.de/~hans.neuschwander/download_skripte/download_difil.html)

... heruntergeladen werden können, sind gekennzeichnet mit dem Zeichen



Alle Downloads von dieser Seite sind ohne Passwort möglich (inkl. diesem Manuskript).

Inhaltsverzeichnis

1. Allgemeines zu digitalen Filtern	1
2. Wiederholung der wichtigsten Grundlagen der digitalen Signalverarbeitung	7
2.1. Einleitung	7
2.2. Prinzip eines digitalen Signalverarbeitungssystems.....	8
2.2.1. Digitalisierung eines analogen Signales	11
2.2.2. Elementarfolgen	14
2.2.3. Die z-Transformation	15
2.2.4. Die Differenzgleichung eines zeitdiskreten Systems.....	17
2.2.5. Die Übertragungsfunktion eines zeitdiskreten Systems	18
2.2.6. Antwort eines zeitdiskreten Systems nach Anregung mit der Impulsfolge (Diracfolge).....	20
3. FIR-Filter	25
3.1. Einleitung	25
3.2. Herleitung des Frequenz- und Phasenganges von FIR-Filtern.....	26
3.3. Berechnung der Filterkoeffizienten von FIR-Filtern	32
3.3.1. Einleitung.....	32
3.3.2. Berechnung der Filterkoeffizienten mit der Fenstermethode.....	32
3.4. Einführungsbeispiel.....	40
3.5. FIR-Filter-Simulation mit LTSpice	44
4. IIR-Filter	47
4.1. Einleitung	47
4.2. Die bilineare Transformation	47
5. Entwurf rekursiver Filter (IIR-Filter) mit SCILAB	53
5.1. Einleitung	53
5.2. Filtereinführungsbeispiel.....	53
5.3. Beispielberechnungen.....	56

6. Entwurf nichtrekursiver Filter (FIR-Filter) mit SCILAB	68
6.1. Einleitung.....	68
6.2. Tiefpass-Filterbeispiel.....	70
7. Grafische Darstellung von Filteralgorithmen	73
7.1. Einleitung.....	73
8. Beispiele einfacher (symmetrischer) FIR-Filter	76
8.1. Einleitung.....	76
8.2. Tiefpass/Interpolator 1. Ordnung.....	76
8.3. Hochpass 1. Ordnung.....	81
8.4. Tiefpass 2. Ordnung.....	85
8.5. Hochpass 2. Ordnung.....	89
8.6. Bandsperre 2. Ordnung.....	93
Literaturverzeichnis	97
Stichwortverzeichnis	99

1. Allgemeines zu digitalen Filtern

Digitale Filter werden heute in vielen Bereichen der Signalverarbeitung eingesetzt. Die ersten Anwender benutzten diese Filter auf Großrechnern zur Erzeugung von Filtern, welche analogtechnisch nicht realisierbar waren (zu tiefe Filtergrenzfrequenzen oder zu hohe Filterordnung). Ein besonderer Vorteil von Digitalfiltern ist z.B. der auf Wunsch mögliche lineare Phasengang bei FIR-Filtern oder die fast beliebig je nach Rechenleistung des verwendeten Signalprozessors vergrößerbare Filterordnung.

Digitale Filter können als

- Tiefpass
- Hochpass
- Bandpass
- Bandsperre
- Allpass

realisiert werden. All dies ist mit einer z-Übertragungsfunktion wie z.B. in Gleichung 1 mit verschiedenen Filterkoeffizienten A_i und B_i möglich.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2}}{1 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2}}$$

Gleichung 1: Beispiel Z-Übertragungsfunktion eines zeitdiskreten Systems 2. Ordnung

Mit speziellen mathematischen Verfahren kann man auch Filter mit fast beliebigem Frequenzgang realisieren (Frequenzabtastfilter).

Die Filterkoeffizienten in Gleichung 1 sind in der Regel konstante Werte und werden im Zähler „A“ und im Nenner „B“ genannt. Diese Bezeichnungen sind in der Literatur durchaus nicht überall gleich. Manche Autoren nennen die Filterkoeffizienten im Zähler „B“ und im Nenner „A“ oder auch ganz anders.



Achtung

Bei adaptiven Filtern werden die Filterkoeffizienten über der Zeit mittels spezieller Algorithmen verändert!

Die Übertragungsfunktion $H(z)$ ist als Produkt zweier Übertragungsfunktionen $H_z(z)$ (Zählerpolynom) und $H_n(z)$ (1/Nennerpolynom) darstellbar.

$$H(z) = H_z(z) \cdot H_n(z)$$

Gleichung 2: Übertragungsfunktion in Produktform

$$\text{mit } H_z(z) = A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2} \text{ und } H_n(z) = \frac{1}{1 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2}}$$

FIR-Filter

Ein Filter mit einer Übertragungsfunktion **ohne** Nennerpolynom wird *nichtrekursives* Filter (FIR-Filter) genannt.

FIR steht dabei für „**finite impulse response**“ (endliche Impulsantwort).

Der gegenwärtige Wert der Ausgangsfolge $\{y(n)\}$ wird nur aus den aktuellen und vergangenen Werten der Eingangsfolge $\{x(n)\}$ berechnet.

Ein Impulseingangssignal wird immer mit einem Ausgangssignal beantwortet, das wieder bis auf Null abklingt. Es gibt keine Rückkopplung vom Ausgang auf den Eingang, ein FIR-Filter kann also nie schwingen, es ist immer stabil.

IIR-Filter

Ein Filter mit einer Übertragungsfunktion **mit** Nennerpolynom wird *rekursives* Filter (IIR-Filter) genannt.

IIR bedeutet dabei „**infinite impulse response**“ (unendliche Impulsantwort).

Der gegenwärtige Wert der Ausgangsfolge $\{y(n)\}$ wird aus den aktuellen und vergangenen Werten der Eingangsfolge $\{x(n)\}$ sowie aus vergangenen Werten der Ausgangsfolge berechnet.

Da der Ausgang auf den Eingang rückgeführt ist, neigen solche Filter zum Schwingen, sie müssen also besonders sorgfältig dimensioniert werden. Dafür besitzen sie wesentlich bessere Filterwirkungen und steilere Flanken als die FIR-Filter.

FIR- und IIR-Filter werden in den nachfolgenden Kapiteln noch genauer vorgestellt.

In den nachfolgenden Ausführungen werden wir alle oben genannten Filtertypen außer dem Allpass behandeln. Bei all diesen Filtertypen interessiert in der Praxis sehr oft nur der Betrag des Amplitudenfrequenzganges. Dessen Verlauf über der Frequenz wird durch Angabe bestimmter Eckdaten vorgegeben. Dabei legt man ein Toleranzschema fest.

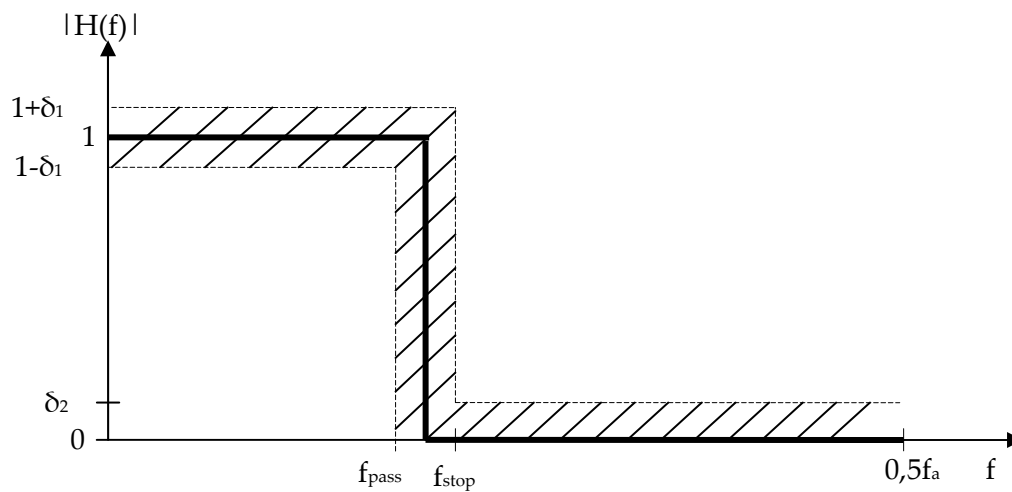


Abbildung 3: Amplitudengang des idealen Tiefpasses

In Abbildung 3 sehen wir als dicke Linie den Verlauf des Betrages der Ausgangsamplitude eines idealen Tiefpasses, dargestellt über der Eingangssignalfrequenz f . Bei der halben Abtastfrequenz $0,5f_a$ endet das Diagramm. Die Verstärkung im Bereich 0 bis zur Frequenz $(f_{\text{pass}} + f_{\text{stop}})/2$ ist 1.

Der schraffierte Bereich gibt einen Toleranzbereich an, innerhalb dessen der Verlauf bei einem zu berechnenden realen Tiefpass liegen soll.

Dieser Toleranzbereich wird vom Entwickler festgelegt. Anschließend wird der Frequenzgang des gewünschten realen Filters mittels diverser Approximationsverfahren in das schraffierte Band gelegt.

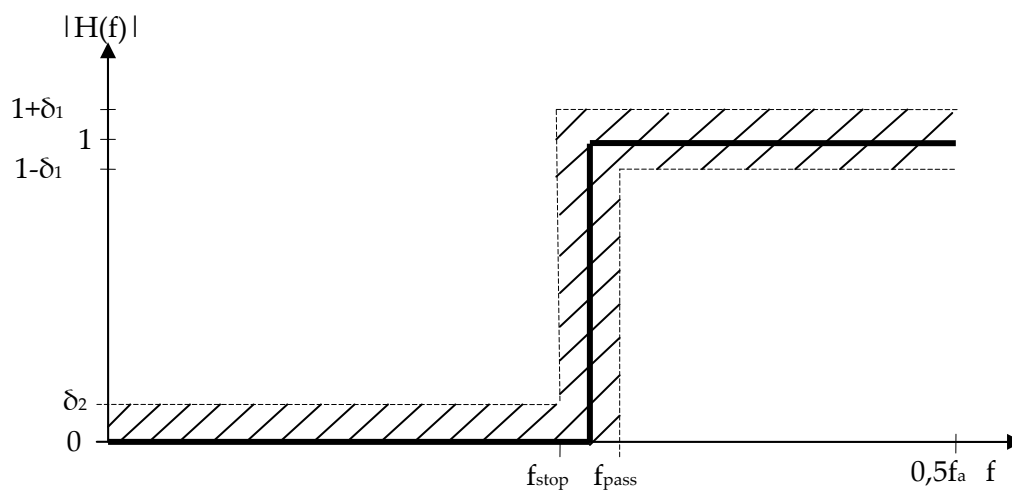


Abbildung 4: Amplitudengang des idealen Hochpasses

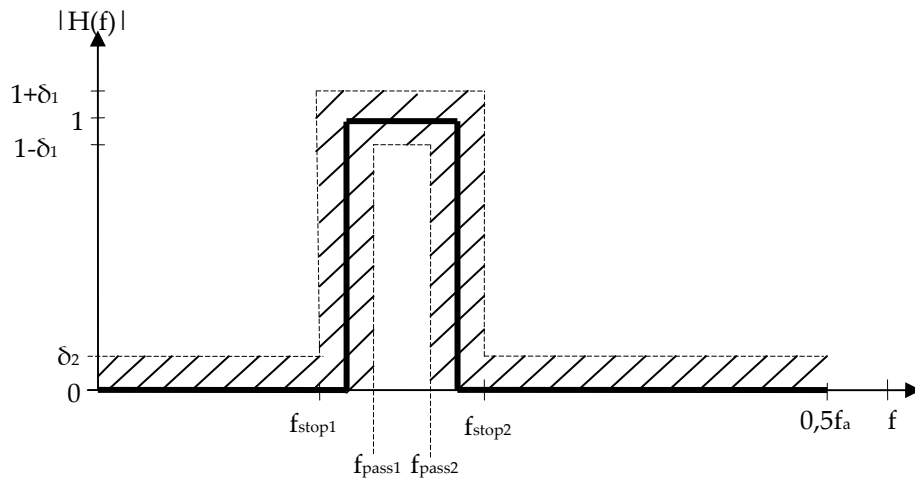


Abbildung 5: Amplitudengang des idealen Bandpasses

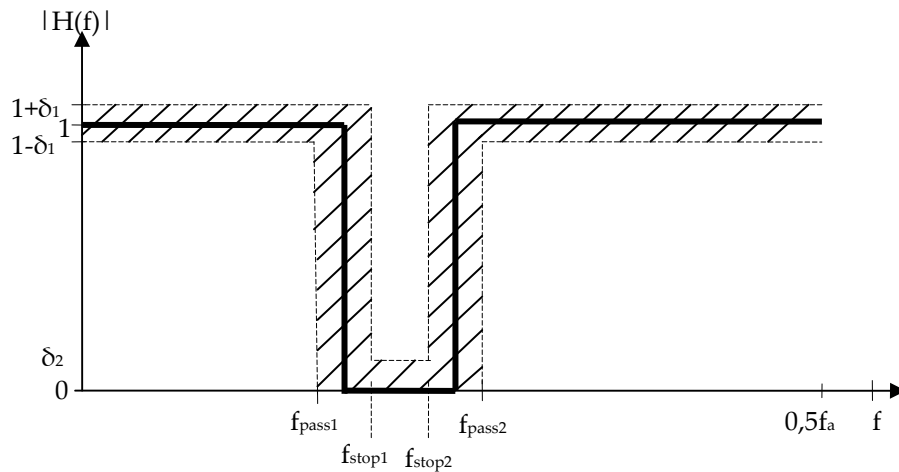


Abbildung 6: Amplitudengang der idealen Bandsperre



Achtung

Filter mit rechteckförmigen Amplitudengängen haben eine unendlich hohe Ordnung und sind somit praktisch nicht zu realisieren!

In der Praxis legt man darum den Toleranzbereich (in den obigen Bildern schraffiert) fest, in dem sich der Amplitudengang für ein bestimmtes Filter befinden darf.

Die maximal zulässigen Abweichungen δ_1 und δ_2 vom idealen Amplitudengang heißen „*Rippel*“ im Durchlassbereich und Sperrbereich. Die Bereiche, die durch das Festlegen der Frequenzgrenzen entstehen, nennt man *Durchlass-, Übergangs- und Sperrbereich* und die dazugehörenden Frequenzgrenzen nennt man *Durchlass- und Sperrfrequenzen*.

Beim Tiefpass erstreckt sich der Durchlassbereich von 0 bis f_{pass} , der Übergangsbereich von f_{pass} bis f_{stop} und der Sperrbereich von f_{stop} bis $0,5f_a$.

Oberhalb der Nyquistfrequenz $0,5f_a$ wiederholen sich diese Bereiche, weil der Amplitudengang eines zeitdiskreten Systems periodisch und symmetrisch ist!

Die Wahl der Toleranzgrenzen ist eine typische Ingenieuraufgabe und hängt in erster Linie von der Filteranwendung ab.

Anhand des Tiefpassfilters wollen wir uns in den folgenden vier Abbildungen die grundlegenden Approximationsarten des Amplitudenganges anschauen. Die Welligkeit im Durchlass- oder Sperrbereich ist in der Praxis durchaus gewollt, damit die Filterordnung klein bleibt. Bei elektroakustischen Anwendungen von Digitalfiltern stört es das menschliche Ohr z.B. nicht, wenn eine leichte Welligkeit im Durchlassbereich des Filters vorhanden ist!

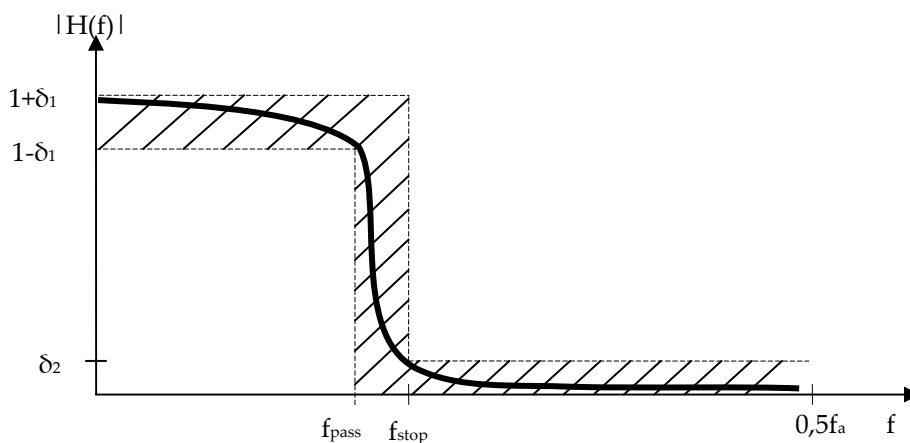


Abbildung 7: Amplitudengang eines realen Tiefpasses mit flachem Verlauf in Durchlass- und Sperrbereich

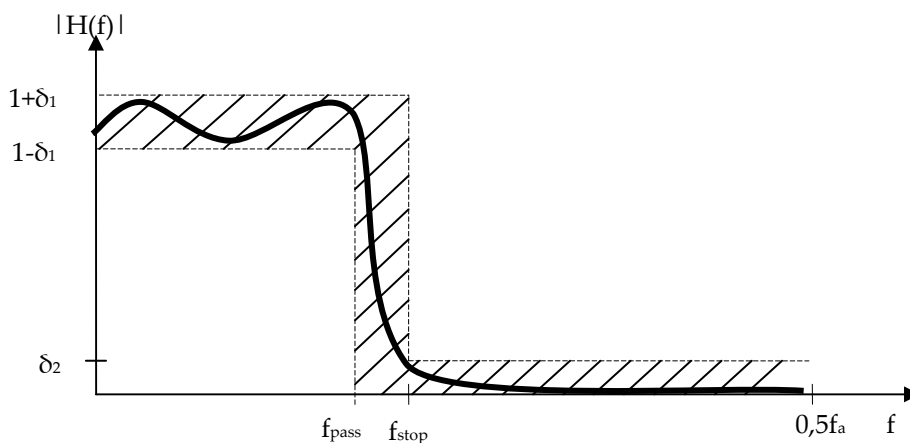


Abbildung 8: Amplitudengang eines realen Tiefpasses mit Welligkeit im Durchlassbereich

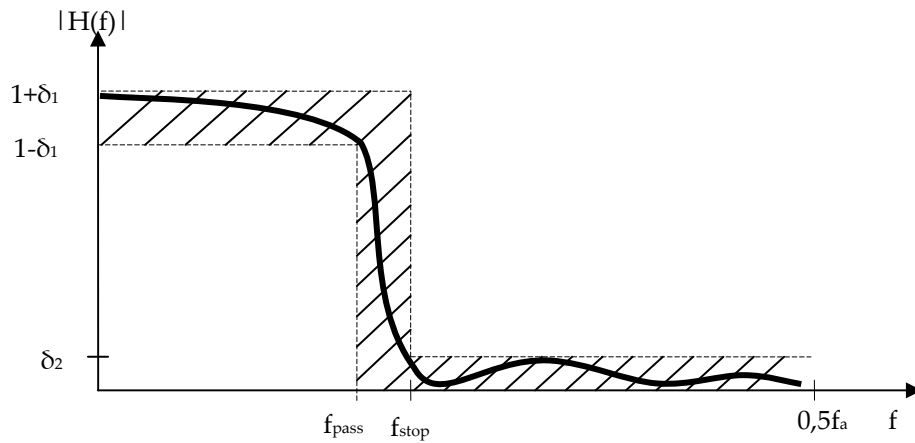


Abbildung 9: Amplitudengang eines realen Tiefpasses mit Welligkeit im Sperrbereich

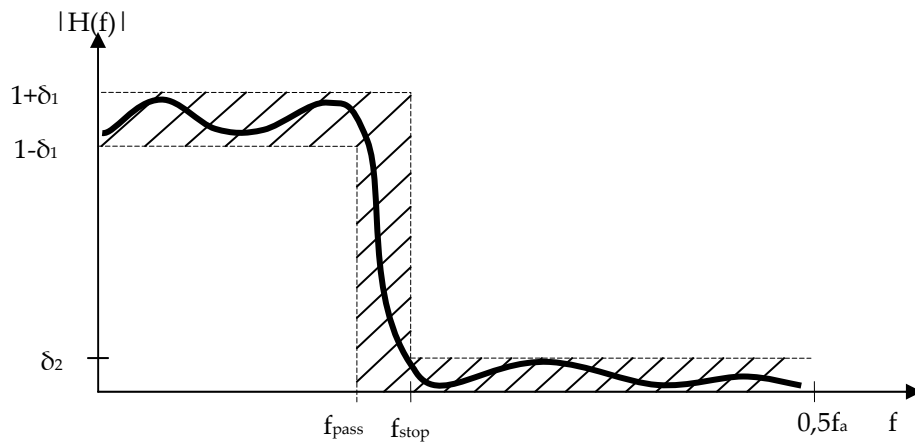


Abbildung 10: Amplitudengang eines realen Tiefpasses mit Welligkeit im Durchlass- und Sperrbereich

Häufig wird für die Filterberechnung die aktuelle Signalfrequenz auf die Abtastfrequenz bezogen. Wir erhalten die normierte Signalfrequenz f' aus Gleichung 3.

$$f' = \frac{f}{f_a} \quad \text{mit } f_a = \text{Abtastfrequenz}$$

Gleichung 3: Normierte Signalfrequenz

2. Wiederholung der wichtigsten Grundlagen der digitalen Signalverarbeitung

2.1. Einleitung

Unter der „digitalen Signalverarbeitung“ verstehen wir die digitale Verarbeitung analoger Signale. In der Regel erfolgt die Verarbeitung der Signale in Echtzeit auf einem Signalprozessor.

Definition
„Digitale
Signal-
verarbeitung“

Dies kann z.B. die Verarbeitung von Bilddaten oder Audiodaten sein. Wir werden alle Beispiele wegen der Einfachheit des Rechneraufbaues mit akustischen Signalen (Sprache, Musik) durchführen. Die dazu nötigen Signalprozessoren und Signalwandler (Analog-Digital- und Digital-Analogwandler) sind wesentlich preiswerter als im Videobereich. In vielen Gebieten der Musik und Elektroakustik werden heute Audiosignale mit Digitalrechnern verarbeitet. Beispiele sind elektronische Musikinstrumente, digitale Mischpulte, CD-Player usw. Dabei werden teilweise komplizierte mathematische Verfahren benötigt.

Ein Signalprozessor (DSP) ist ein Digitalrechner, welcher für die Verarbeitung digitalisierter Analogsignale optimiert ist. Jeder normale Mikroprozessor kann auch Signalverarbeitungsalgorithmen ausführen, aber viel langsamer und uneffektiver als ein spezieller DSP!



**Was bedeutet
DSP?**

Ein mit einem DSP verarbeitetes Signal (Musik, Video usw.) kann man in der Regel mittels der Fouriertransformation über einen bestimmten Zeitbereich in seine Bestandteile (reine Sinusschwingungen) zerlegen. Man redet dann von der Grundschwingung (der Sinus mit der tiefsten Frequenz) und den Oberschwingungen (Frequenz oft ein ganzzahliges Vielfaches der Grundschwingung). Die Amplituden der einzelnen Sinusschwingungen variieren über der Zeit und ergeben dadurch einen bestimmten Klangeindruck.

2.2. Prinzip eines digitalen Signalverarbeitungssystems

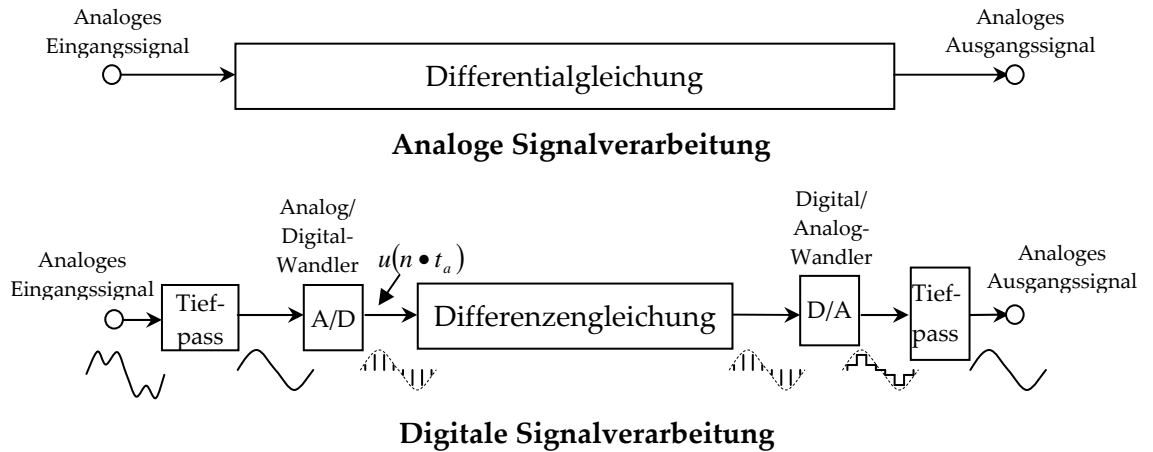


Abbildung 11: Vergleich analoge und digitale Signalverarbeitung

In Abbildung 11 sehen wir oben ein analoges und unten ein digitales System. Das digitale System erhält am Eingang ein analoges Audiosignal, welches in seiner Bandbreite noch nicht begrenzt ist. Nach dem Tiefpass sieht man an der angedeuteten Signalform unter dem Bild, dass Obertöne entfernt wurden. Das bandbreitenbegrenzte Signal wird einem A/D-Wandler zugeführt, welcher das analoge Signal zu äquidistanten Zeitpunkten abtastet (Abtastfrequenz oder Samplefrequenz f_a) und in eine digitale Zahlenform mit einer bestimmten Bitzahl (heute meist 24-Bit oder mehr) umwandelt. Diese Zahlen erscheinen als Signal $x(n)$ mit dem Abtasttakt am Eingang eines Signalprozessors. Dabei bedeutet „ $x(n)$ “ das Eingangssignal vom aktuellen Abtasttakt und „ $x(n-1)$ “ das Eingangssignal von einem Abtasttakt vorher. Der Signalprozessor arbeitet nun mit jedem Abtasttakt eine mathematische Formel ab, in welcher das aktuelle Eingangssignal $x(n)$, ältere Eingangssignale $x(n-1)$, $x(n-2)$... (sind zwischengespeichert) sowie ältere Ausgangssignale $y(n-1)$, $y(n-2)$... nach bestimmten Regeln verknüpft werden und ein aktuelles Ausgangssignal $y(n)$ erzeugen. Die Rechendauer dieses Algorithmus darf maximal $1/t_a$ sein. Ist die Rechendauer länger als $1/t_a$, würde der A/D-Wandler schon den nächsten Eingangswert zum DSP senden, während dieser noch den alten Ausgangswert berechnet. Dies darf nie sein!

Das Übertragungsverhalten eines analogen Systems wird durch Differentialgleichungen beschrieben. Das Übertragungsverhalten eines digitalen Systems wird durch Differenzengleichungen beschrieben.

Wie wir schon gesehen haben, wird bei der digitalen Signalverarbeitung das analoge Eingangssignal zuerst mittels eines Tiefpasses von unerwünschten hohen Frequenzanteilen befreit. Dies ist nötig, um das Abtasttheorem zu erfüllen.

Das Abtasttheorem von „SHANNON“ besagt:

Jedes analoge Signal kann durch Abtastung mit einer Frequenz *größer* als die doppelte im Signal vorkommende maximale Obertonfrequenz (Oberton = reines Sinussignal) digitalisiert werden und danach auf umgekehrtem Weg wieder in ein Analogsignal umgewandelt werden. *Es entstehen dabei keine Signalveränderungen!* Die Rekonstruktion des analogen Ausgangssignals erfolgt mit einem analogen Tiefpass.



**Abtasttheorem
von Shannon**

Beispiel:

Ein Audiosignal beansprucht normalerweise den Frequenzbereich von 20 Hz bis 20 kHz. Tastet man dieses Signal mit einem A/D-Wandler in Zeitabständen von 1/40,001 kHz ab, so lässt sich das analoge Signal aus den digitalisierten Werten *theoretisch* fehlerfrei rekonstruieren.

In der Praxis wählt man die Abtastfrequenz ein gutes Stück höher (z.B. 44,1kHz bei der Audio-CD) als die doppelte im Signal vorkommende höchste Spektralkomponente (hier 20kHz), weil dadurch der Entwurf des analogen Ein- und Ausgangstiefpasses einfacher wird. Bei modernen Systemen bedient man sich der Überabtastung. Man tastet mit einer sehr hohen Frequenz (Vielfaches der doppelten maximalen Eingangsfrequenz) ab. Dann kann der vorgeschaltete Tiefpass einen relativ flachen Übergang vom Durchlass- in den Sperrbereich haben.

Das Ausgangssignals des Analog-Digitalwandlers ist mathematisch gesehen eine Zahlenfolge $u(n \cdot t_a)$ über der Zeit. Dabei ist „u“ der Signalmomentanwert zum Zeitpunkt $(n \cdot t_a)$ und t_a ist das Abtastintervall.

Die Ausgangswerte von Analog-Digitalwandlern sind üblicherweise Zahlen im Zweierkomplementformat. Sie stellen positive- oder negative Signalmomentanwerte dar.

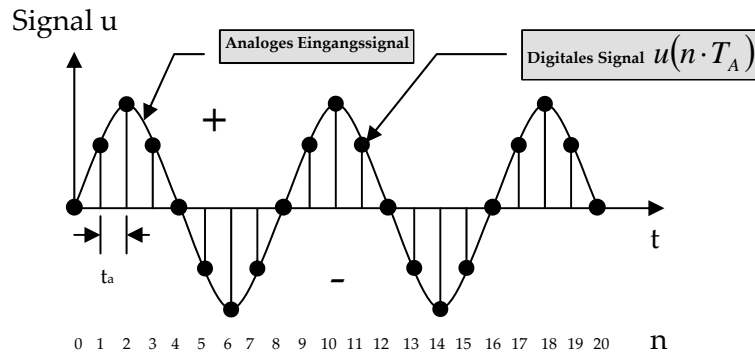


Abbildung 12: Signalverlauf eines Abtastsystems

In Abbildung 12 sehen wir, wie ein reines Sinussignal mit der achtfachen Frequenz des Sinussignals selbst digitalisiert wird. An den Stellen mit den runden Punkten entstehen die Zahlenwerte, welche jeweils als Eingangswerte $x(n) = u(n \cdot t_a)$ der Differenzgleichungen dienen. Zwischen den Abtastpunkten ist das Ausgangssignal des A/D-Wandlers undefiniert.

Beispiel:

Ein A/D-Wandler hat einen Eingangsbereich von +1,25V Spitze-Spitze und wandelt dieses Signal mit einer Abtastrate von 44,1kHz in 16-Bit Zahlen um. Mit einer 16-Bit-Zahl kann man 2^{16} -Werte = 65536 Werte darstellen. Der Signalnullwert wird üblicherweise dem Ausgangswert 0x0000 zugeordnet. Der positiv maximale Wert von +1,25V wird dann in die größte positive Zahl 0x7FFF (+32767₁₀) und der negativ maximale Wert von -1,25V wird dann in die größte negative Zahl 0x8000 (-32768₁₀) umgewandelt. Theoretisch ist auch jede andere Lage des Nullpunktes möglich.

0x1234 bedeutet „Hexadezimaldarstellung“ => hier Hexadezimalzahl 1234₁₆
Wir benutzen ab sofort für hexadezimale Zahlen den Prefix „0x“ wie in der Hochsprache „C“ üblich.

Die vom A/D-Wandler gelieferten Zahlenfolgen werden nun mittels geeigneter Differenzgleichungen bearbeitet und das Ergebnis mit einem Digital-Analogwandler und einem nachfolgenden Tiefpass wieder in ein Analogsignal zurückgewandelt.

2.2.1. Digitalisierung eines analogen Signales

Ein analoges Signal wird in ein digitales Signal umgewandelt mittels

- Zeitquantisierung
- Amplitudenquantisierung
- Kodierung

Die Zeitquantisierung wird mit einem Abtast-Halteglied (Sample-Hold-Stufe) durchgeführt. Dieses speichert eine Probe des analogen Eingangssignales zum Zeitpunkt $(n \cdot t_a)$ und führt es dem Analog-Digitalwandler zu. Der Analog-Digitalwandler wandelt das Signal in endlicher Zeit (Wandlungszeit) in eine Zahl mit endlicher Bitbreite um (Amplitudenquantisierung und Kodierung). Dies wird je nach Hersteller nach verschiedenen Verfahren gemacht. Wir wollen hier nicht auf die Wandlungsverfahren eingehen.

Das analoge Eingangssignal muss nach dem Abtasttheorem bandbreitenbegrenzt sein, d.h. es muss eine Signalgrenzfrequenz f_{gs} geben, oberhalb derer es keine Signalfrequenzkomponenten mehr gibt. Nur dann gelten die nachfolgend angegebenen mathematischen Berechnungen.

Für das Spektrum eines bandbegrenzten Signals (Fourier-Transformierte) muss daher gelten:

$$\underline{U}(f) = \int_{-\infty}^{+\infty} u(t) \exp(-j \cdot 2 \cdot \pi \cdot f \cdot t) dt$$

Gleichung 4: Spektrum eines Signals nach Fourier

Mit $\underline{U}(f) = 0$ für $|f| \geq f_{gs}$ f_{gs} = Signalgrenzfrequenz

Die Abtastfrequenz f_a ($f_a = \frac{1}{t_a}$) muss grösser als $2 \cdot f_{gs}$ sein $\Rightarrow f_a > 2 \cdot f_{gs}$

Sind diese Bedingungen erfüllt, so kann man das analoge Signal laut Gleichung 5 aus den Abtastwerten $u(n \cdot t_a)$ fehlerfrei rekonstruieren.

$$u(t) = \sum_{n=-\infty}^{n=\infty} u(n \cdot t_a) \cdot \frac{\sin\left(\pi \cdot \frac{t - n \cdot t_a}{t_a}\right)}{\left(\pi \cdot \frac{t - n \cdot t_a}{t_a}\right)} \text{ mit } t_a < \frac{1}{2 \cdot f_{gs}}$$

Gleichung 5: Interpolationsformel des Abtasttheorems

Das analoge Signal wird dabei als eine unendliche Summe von gewichteten ($u(n \cdot t_a)$) und um ganzzahlige Vielfache der Abtastperiode gegeneinander verschobenen *si-Funktionen* dargestellt.

$$g(t) = \frac{1}{t_a} \cdot \frac{\sin\left(\pi \cdot \frac{t}{t_a}\right)}{\left(\pi \cdot \frac{t}{t_a}\right)}$$

Gleichung 6: Impulsantwort $g(t)$ eines idealen Tiefpasses

si-Funktion

Die *si-Funktion* ist die Impulsantwortfunktion eines idealisierten Tiefpasses mit einem exakt rechteckigen Frequenzgang und der Grenzfrequenz $f_g = f_a/2$

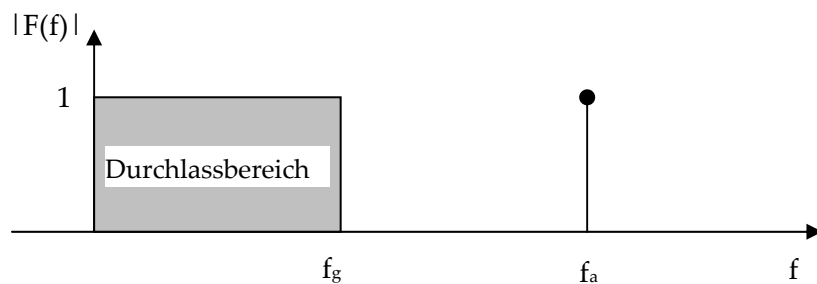


Abbildung 13: Frequenzgang eines idealen Tiefpasses (grau hinterlegt)

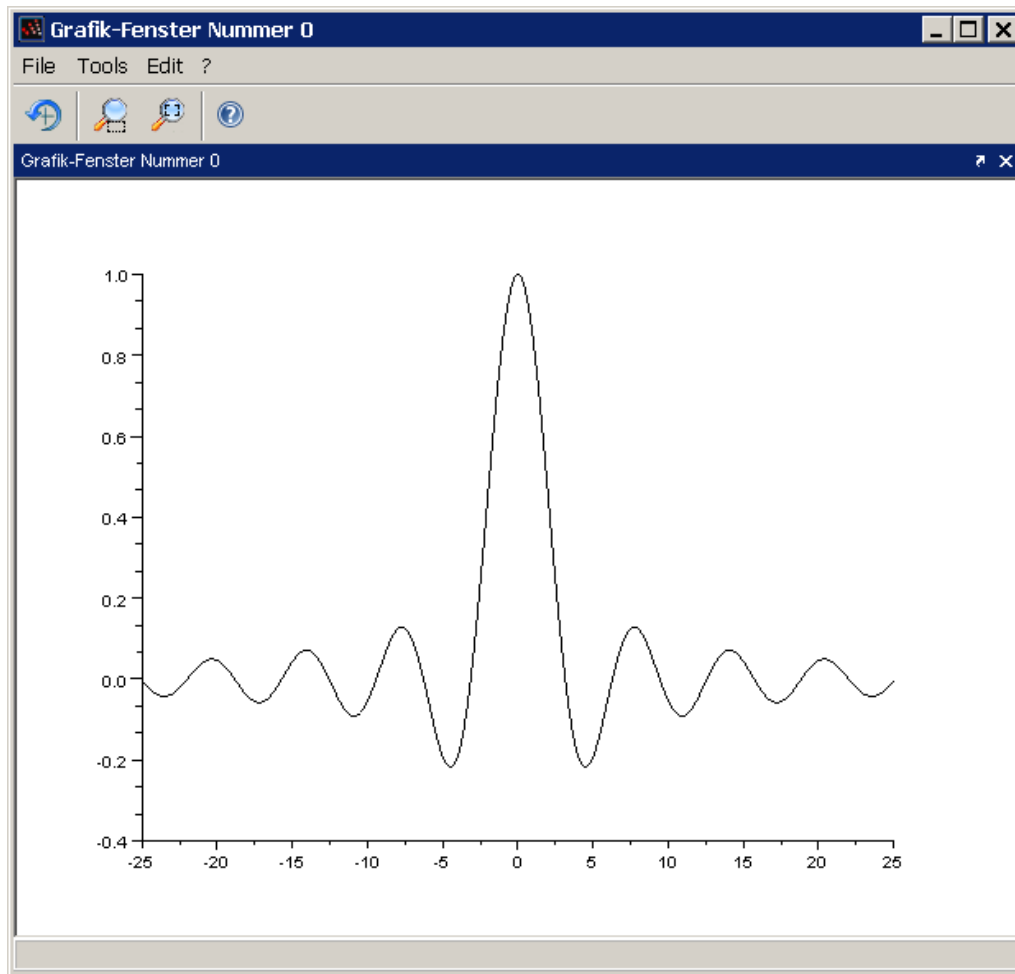


Abbildung 14: Verlauf der si-Funktion, berechnet mit SCILAB

In Abbildung 14 sehen den Verlauf der si-Funktion laut Gleichung 7 im Bereich ± 25 des Argumentes x . Die Berechnung erfolgte mit 10000 Stützstellen zwischen -25 und $+25$.

$$si(x) = \frac{\sin(x)}{(x)}$$

Gleichung 7: Gleichung der „si-Funktion“ $si(x)$

Dateiname: „sci.sce“



Einführung in SCILAB siehe Seite 53!

2.2.2. Elementarfolgen

In der digitalen Signalverarbeitung gibt es eine Reihe von standardisierten Zahlenfolgen, von denen nachfolgend nur die wichtigsten dargestellt sind. Diese sind die meistgebrauchten *Eingangsanregungssignale* eines Digitalsystems. Sie sind vergleichbar mit den Standard-Eingangssignalen „Eingangssprung“, „Eingangsimpuls“ usw. der analogen Signalverarbeitung.

2.2.2.1. Die Einheits-Sprungfolge

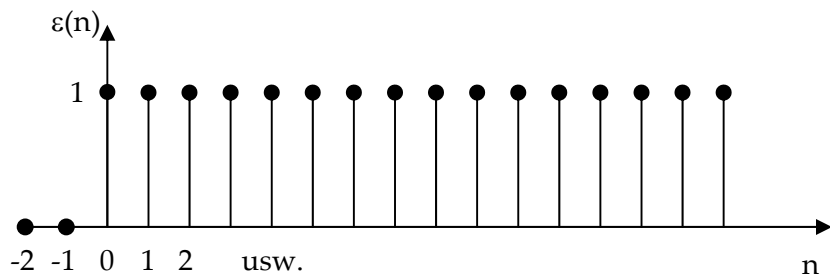


Abbildung 15: Die Einheits-Sprungfolge

Mit der Definition:

$$\varepsilon(n) = \begin{cases} 1 & \text{für } n \geq 0 \\ 0 & \text{sonst} \end{cases}$$

Das Signal existiert nur zu bestimmten Zeitpunkten ($n = 0, 1, 2, \dots$) mit bestimmten Werten („0“ oder „1“). Die Zeitpunkte haben den Abstand $1/f_a$.

2.2.2.2. Die Einheits-Impulsfolge

Sie wird auch Dirac-Folge genannt. Kennt man die Impulsantwort eines linearen zeitinvarianten Systems auf dieses Eingangssignal, so kann man die Antwort auf beliebige andere Eingangssignale aus der Impulsantwort durch die diskrete Faltung berechnen.

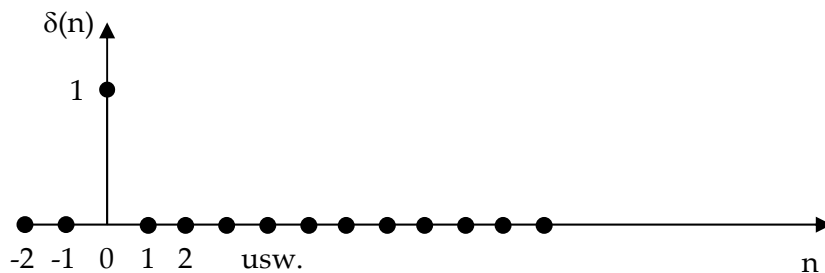


Abbildung 16: Die Einheits-Impulsfolge

Mit der Definition:
$$\delta(n) = \begin{cases} 1 & \text{für } n = 0 \\ 0 & \text{sonst} \end{cases}$$

2.2.3. Die z-Transformation

Bei analogen Systemen unterscheidet man Zeit- und Frequenzbereich. Im Zeitbereich arbeitet man mit Spannungs-Zeitfunktionen. Analoge Systeme werden üblicherweise durch Differentialgleichungen beschrieben. Sind die Systeme „linear und zeitinvariant“ (sogenannte LTI-Systeme), so erhält man lineare Differentialgleichungen mit konstanten Koeffizienten. Zur Berechnung des Ausgangssignals bei gegebenem Eingangssignal ist die Lösung der Differentialgleichung erforderlich, was direkt meist sehr schwierig oder gar nicht möglich ist! An dieser Stelle wird oft die Laplace-Transformation eingesetzt, mit deren Hilfe man die Differentialgleichung in eine leichter lösbare algebraische Gleichung im Bildbereich überführt, diese löst und anschließend mittels Korrespondenztabelle wieder in den Zeitbereich zurücktransformiert.

Geht man zu zeitdiskreten Signalen und Systemen über, so entsprechen den Zeitfunktionen die Zahlenfolgen, der Differentialgleichung die Differenzgleichung und der Laplace-Transformation die z-Transformation, welche eine Differenzgleichung in eine algebraische Gleichung überführt.

Bei der Definition der z-Transformation wird von einer idealen Abtastung eines analogen Signales ausgegangen.

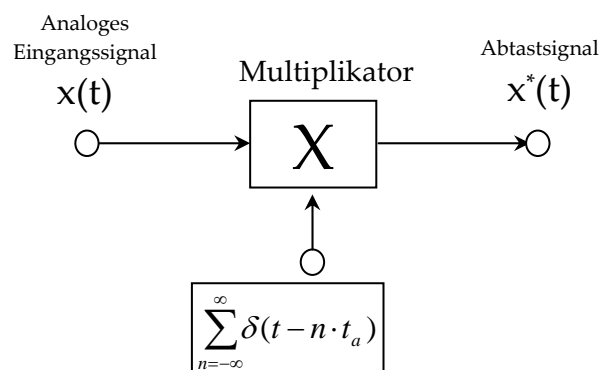


Abbildung 17: Ideale Abtastung eines Analogsignals mit Dirac-Puls

Der Multiplikator (Abtaster) multipliziert das analoge Eingangssignal mit unendlich vielen zeitlich verschobenen Diracpulsen. Dadurch entsteht ein Ausgangswert (Abtastsignal) nur an den Stellen, an welchen die Diracpulse eine „1“ haben.

Am Ausgang des Multiplizierers erhält man dann ein ideales Abtastsignal.

$$x^*(t) = x(t) \cdot \sum_{n=-\infty}^{\infty} \delta(t - n \cdot t_a)$$

Gleichung 8: Ideal abgetastetes Analogsignal

Da $x(t)$ durch die Multiplikation mit den Diracpulsen nur an den Stellen $x(n \cdot t_a)$ am Ausgang erscheint und der Ausgang ansonsten 0 ist, kann man $x(t)$ als $x(n \cdot T_A)$ auch in die Summation einbeziehen:

$$x^*(t) = \sum_{n=-\infty}^{\infty} x(n \cdot t_a) \cdot \delta(t - n \cdot t_a)$$

Gleichung 9: Ideales Abtastsignal

In Gleichung 9 sehen wir, dass ein ideal abgetastetes Signal als eine Folge von Diracpulsen (Höhe 1) mit verschiedenem Flächeninhalt (Breite $x(n \cdot t_a)$) darstellbar ist.

In diesem Abtastsignal steckt die Information in den Zahlen $x(n \cdot t_a)$. Man kann das Abtastsignal als Darstellung der Zahlenfolge $\{x(n \cdot t_a)\}$ als Zeitfunktion auffassen.

Das Abtastsignal wird nun einer Laplacetransformation unterzogen. Man erhält die Laplace-Transformierte

$$X^*(s) = L\{x^*(t)\} = \int_{-\infty}^{\infty} x^*(t) \cdot e^{-st} dt = \sum_{n=-\infty}^{\infty} x(n \cdot t_a) \cdot e^{-s \cdot t_a \cdot n}$$

Gleichung 10: Ausgangsgleichung der Z-Transformationsherleitung

Die Diracfolge erzeugt unter dem Integral nur an den Stellen „ $n \cdot t_a$ “ einen Wert ungleich 0. Dadurch kann man auch das Integral durch die Summe ersetzen. Mit der Substitution $e^{s \cdot t_a} = z$ wird daraus

$$L\{x^*(t)\} = \sum_{n=-\infty}^{\infty} x(n) \cdot z^{-n}$$

Gleichung 11: Funktion mit der Variablen „ z “

Damit wird einer Zahlenfolge $\{x(n)\}$ eine gebrochene rationale Funktion von z zugeordnet, die man die z -Transformierte $X(z)$ der Zahlenfolge $x(n)$ nennt. Die Berechnung dieser Funktion wird z -Transformation genannt. Sie leistet für Zahlenfolgen das, was die Laplace-Transformation für Zeitfunktionen erbringt. z-Transformation

Die Rücktransformation erfolgt wie bei der Laplace-Transformation mittels Korrespondenztabelle.

Man schreibt häufig $Z[\{x(n)\}] = X(z)$

Verbreitet ist auch die Verwendung des Zuordnungssymbols $x(n) \circ \longrightarrow \bullet X(z)$

2.2.4. Die Differenzgleichung eines zeitdiskreten Systems

Die Differenzgleichung ist eine Rechenvorschrift, bei welcher der gegenwärtige Wert der Ausgangsfolge $\{y(n)\}$ eines zeitdiskreten Systems aus dem gegenwärtigen Wert der Eingangsfolge $\{x(n)\}$ sowie aus vergangenen Werten von Eingangs- und Ausgangsfolge berechnet wird. Am wichtigsten sind die linearen Differenzgleichungen mit konstanten Koeffizienten.

In der Fachliteratur tauchen die Koeffizientennamen Ax oder Bx je nach Autor bei $x(n)$ oder $y(n)$ auf. Dies ist nicht einheitlich! Wir ordnen die A -Koeffizienten den $x(n)$ zu!



Achtung

Für ein System 2. Ordnung (Signalverzögerung um maximal 2 Abtasttakte T_a) gilt dann die Differenzgleichung

$$y(n) = A_0 \cdot x(n) + A_1 \cdot x(n - 1) + A_2 \cdot x(n - 2) - B_1 \cdot y(n - 1) - B_2 \cdot y(n - 2)$$

Gleichung 12: Differenzgleichung 2. Ordnung

Wie man sieht, wird der aktuelle Ausgangswert $y(n)$ aus dem aktuellen Eingangswert $x(n)$ und früheren Werten von $x(n) \Rightarrow x(n-i)$ und $y(n) \Rightarrow y(n-k)$ berechnet.

Bei einer allgemeinen Systemordnung von „N“ ergibt sich die Gleichung

$$y(n) = \sum_{i=0}^M A_i \cdot x(n - i) - \sum_{i=1}^N B_i \cdot y(n - i) \text{ mit } M \leq N$$

Gleichung 13: Differenzgleichung N. Ordnung

2.2.5. Die Übertragungsfunktion eines zeitdiskreten Systems

Man erhält die Übertragungsfunktion durch Anwendung der Z-Transformation auf die Differenzgleichung. Dabei benutzt man die in der Fachliteratur veröffentlichten Korrespondenztabelle.

Mit Hilfe der beiden Korrespondenzen (aus Fachliteratur entnommen)

$$x(n) \text{ --- } \bullet \text{ } X(z) \quad \text{und} \quad x(n-i) \text{ --- } \bullet \text{ } X(z) \cdot z^{-i}$$

wird z.B. aus Gleichung 12

$$Y(z) = A_0 \cdot X(z) + A_1 \cdot X(z) \cdot z^{-1} + A_2 \cdot X(z) \cdot z^{-2} - B_1 \cdot Y(z) \cdot z^{-1} - B_2 \cdot Y(z) \cdot z^{-2}$$

Gleichung 14: Algebraische Gleichung 2. Ordnung

Bildet man nun den Quotienten der z-Transformierten von Ausgangs- und Eingangsfolge, so ergibt sich die Übertragungsfunktion

$$H(z) = \frac{Y(z)}{X(z)} = \frac{A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2}}{1 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2}}$$

Gleichung 15: Übertragungsfunktion eines zeitdiskreten Systems

Allgemein gilt für die Übertragungsfunktion die Definition:

$$H(z) = \frac{Z[\{y(n)\}]}{Z[\{x(n)\}]} = \frac{Y(z)}{X(z)}$$

Gleichung 16: Definition der Übertragungsfunktion eines zeitdiskreten Systems

Die Systemantwort eines zeitdiskreten Systems kann man wie in Abbildung 18 dargestellt ermitteln.

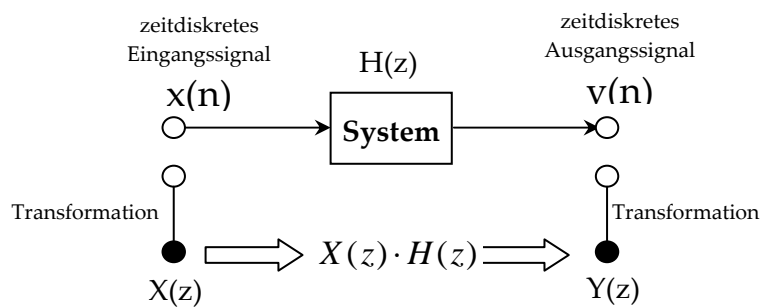


Abbildung 18: Bild zur Berechnung der Systemantwort

Daraus ergibt sich folgendes Schema zur Berechnung der Ausgangsfolge eines zeitdiskreten Systems:

1. Transformation der Eingangsfolge $x(n)$ in den z -Bereich
2. Berechnung der Z -Transformierten der Ausgangsfolge durch Multiplikation der Z -Transformierten der Eingangsfolge mit der Übertragungsfunktion $H(z)$
3. Rücktransformation der Z -Transformierten der Ausgangsfolge in den Folgenbereich $y(n)$

2.2.6. Antwort eines zeitdiskreten Systems nach Anregung mit der Impulsfolge (Diracfolge)

Im analogen System ist die Impulsantwortfunktion die Systemantwort auf Anregung mit einem Dirac-Impuls. Im zeitdiskreten System ist die Impulsantwortfolge die Systemantwort auf die Impulsfolge (Z-Transformierte von $\delta(n) = 1$) als Eingangsfolge.

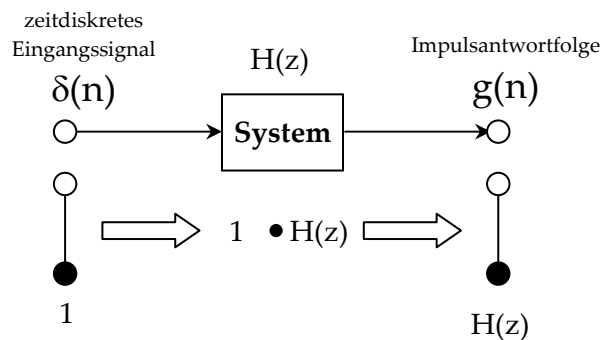


Abbildung 19: Bild zur Ermittlung der Impulsantwortfolge

Wie man aus Abbildung 19 sieht, ist die Impulsantwortfolge eine vollständige Beschreibung eines zeitdiskreten Systems, da diese ja nichts anderes als die Systemübertragungsfunktion $H(z)$ selbst ist!

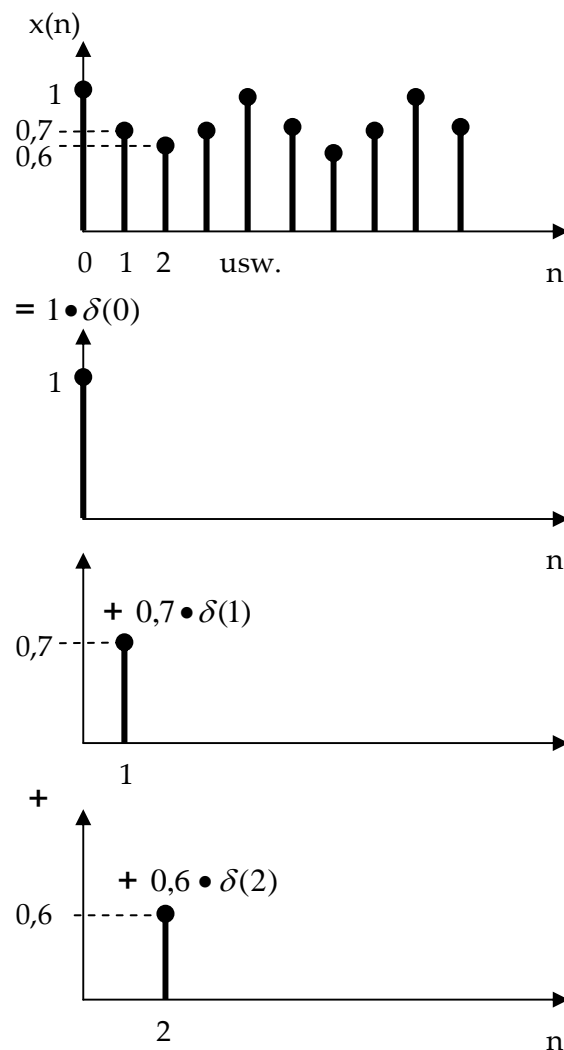
Daraus kann man schließen, dass bei bekannter Impulsantwortfolge eines Systems die Antwort auf beliebige Eingangssignalfolgen leicht berechenbar ist.

Eine beliebige Folge $\{x(n)\}$ kann wie in Abbildung 20 dargestellt durch eine Summe von gegeneinander verschobenen Dirac-Folgen dargestellt werden, die mit den Elementen der Folge gewichtet werden.

In mathematischer Formulierung erhält man dann

$$x(n) = \sum_{m=-\infty}^{\infty} x(m) \cdot \delta(n - m)$$

Gleichung 17: Gewichtete Diracfolge



und so weiter

Abbildung 20: Entstehung der Faltungssumme

Man nennt dies Faltungssumme. Sie entspricht der Faltungsoperation für kontinuierliche Funktionen. Durch die Faltungssumme zweier Folgen $u(n)$ und $v(n)$ wird eine neue Folge $w(n)$ entsprechend

$$w(n) = u(n) * v(n) = \sum_{m=-\infty}^{\infty} u(m) \cdot v(n-m)$$

Gleichung 18: Faltung zweier Folgen

gebildet. Der „*“ ist der Faltungsoperator. Die Faltungssumme ist kommutativ.

Es gilt:

$$u(n) * v(n) = v(n) * u(n)$$

Gleichung 19: Faltung ist kommutativ

Will man die Ausgangsfolge eines beliebigen Systems auf eine beliebige Eingangsfolge ermitteln, so genügt es, die Ausgangsfolge (Impulsantwortfolge = $g(n)$) dieses Systems auf die Impulsfolge als Eingang zu kennen.

Jede beliebige Eingangsfolge lässt sich als Summe von gegeneinander verschobenen mit den Elementen der Eingangsfolge gewichteten Dirac-Folgen darstellen. Deshalb kann die gesuchte Ausgangsfolge als Summe von mit den Elementen der Eingangsfolge gewichteten gegeneinander verschobenen Impulsantwortfolgen ($g(n)$) formuliert werden.

$$y(n) = \sum_{m=-\infty}^{\infty} x(m) \cdot g(n - m)$$

Gleichung 20: Antwortfolge eines Systems als Faltung mit der Impulsantwortfolge

$g(n-m)$ = eine Impulsantwortfolge an der Stelle „ $n-m$ “
 $x(m)$ = ein Wert der Eingangsfolge an der Stelle „ m “

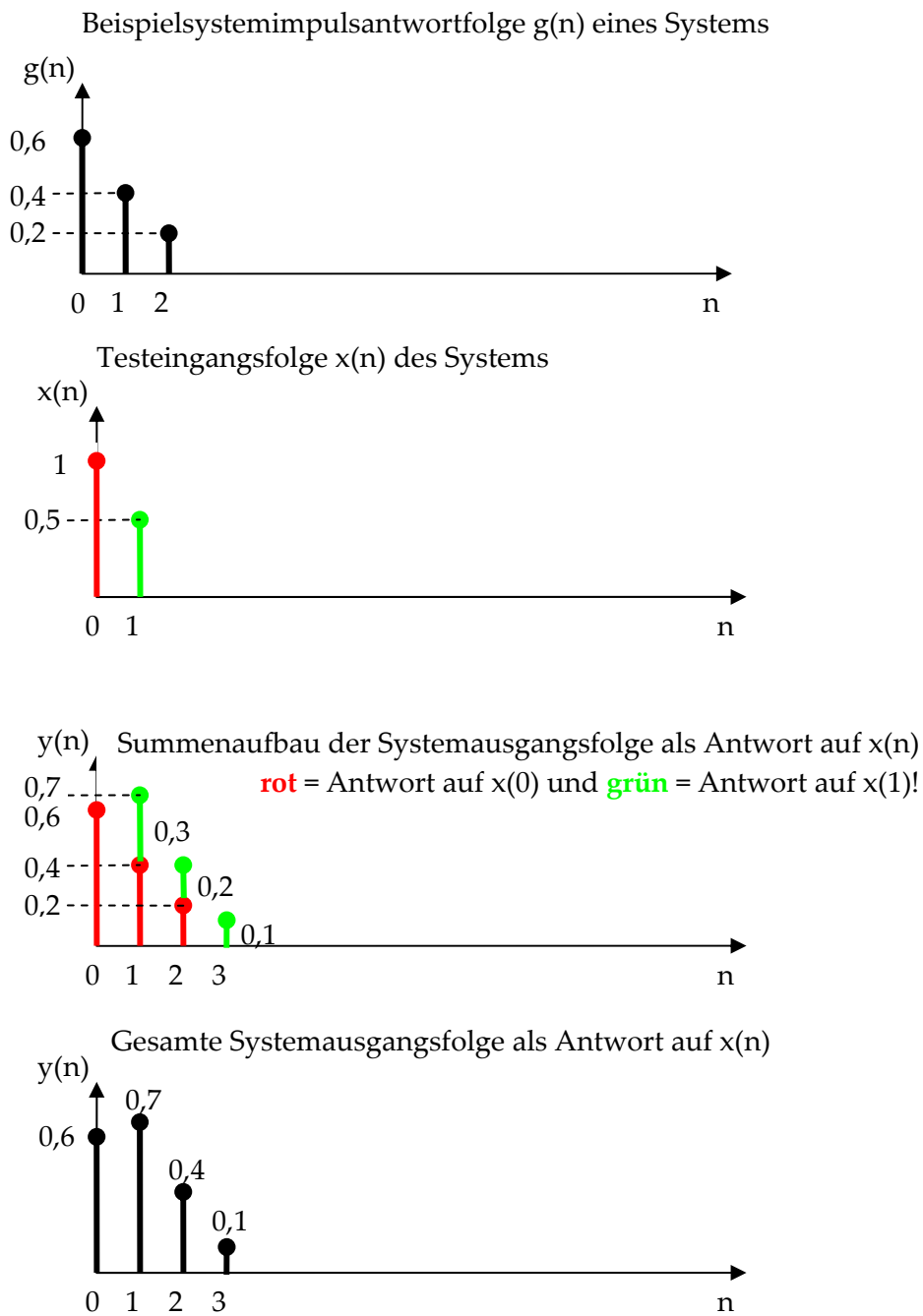


Abbildung 21: Beispiel der Berechnung einer Faltungssumme

In Abbildung 21 sehen wir, wie ein System mit der Beispielsystemimpulsantwortfunktion $g(n)$ auf ein Testeingangssignal $x(n)$ reagiert.

Es gilt:
$$y(n) = \sum_{m=0}^2 x(m) \bullet g(n-m)$$

Wir berechnen nun die einzelnen Impulsanteile $y(n)$ für $n = 0-3$ am Ausgang:

$$Y(0) = x(0) \cdot g(0) + x(1) \cdot g(-1) + x(2) \cdot g(-2) \quad \text{ab } x(2) \text{ ist } x(n) = 0, g(-?) \text{ immer } 0!$$

$$Y(0) = 1 \cdot 0,6 + 0,5 \cdot 0 = \mathbf{0,6} \quad \text{da } x(2) \text{ und } g(-1,-2) = 0$$

$$Y(1) = x(0) \cdot g(1) + x(1) \cdot g(0) \quad \text{da, } g(-?) \text{ immer } 0 \text{ ist!}$$

$$Y(1) = 1 \cdot 0,4 + 0,5 \cdot 0,6 = 0,4 + 0,3 = \mathbf{0,7} \quad \text{da, } g(-1) = 0$$

$$Y(2) = x(0) \cdot g(2) + x(1) \cdot g(1) \quad \text{ab } x(2) \text{ ist } x(n) = 0!$$

$$Y(2) = 1 \cdot 0,2 + 0,5 \cdot 0,4 = 0,2 + 0,2 = \mathbf{0,4} \quad \text{da, } g(-1) = 0$$

$$Y(3) = x(0) \cdot g(3) + x(1) \cdot g(2) \quad \text{da, } g(-?) \text{ immer } 0 \text{ ist!}$$

$$Y(3) = 1 \cdot 0 + 0,5 \cdot 0,2 = \mathbf{0,1} \quad \text{da, } g(3) = 0$$

3. FIR-Filter

3.1. Einleitung

FIR-Filter sind digitale Filter, zu denen es in der Analogtechnik keinen direkten Vergleich gibt! Sie haben wie der Name es sagt eine endliche Impulsantwort (**Finite Impulse Response-Filter**). Dieses Verhalten erreicht man dadurch, dass das Filterausgangssignal nur von zeitlich zurückliegenden Filtereingangswerten abhängt. Das Filter besitzt intern also keine Speicher für vergangene Filterausgangswerte => es sind keine Rückkopplungen vom Filterausgang in das Filter vorhanden.

Ein Impulseingangssignal wird immer mit einem Ausgangssignal beantwortet, das wieder bis auf Null abklingt. Ein FIR-Filter kann also nie schwingen, da es keine Rückkopplung gibt, ist es immer stabil.

FIR-Filter werden auch als „Nichtrekursive Filter“ bzw. als „Transversalfilter“ bezeichnet. Bei ihnen besteht $H(z)$ nur aus dem Zählerpolynom. Sie haben besondere Eigenschaften. Bei ihnen kann man Amplituden- und Phasengang getrennt voneinander vorgeben. Es ist auch ein linearer Phasengang erreichbar, was bei analogen Filtern unmöglich ist.

Die Impulsantwortfolge ist begrenzt und gleich der Folge der Filterkoeffizienten. Die Impulsantwortfolge eines FIR-Filters mit z.B. 10 Koeffizienten dauert dann eigentlich $9 \cdot t_a$, weil nur die 9 Verzögerungsglieder vorhanden sind. Die Koeffizienten werden A_0 bis A_9 genannt. A_0 ist $x(n)$ zugeordnet. A_1 ist $x(n-1)$ zugeordnet usw.

Das Ausgangssignal $y(n)$ taucht aber bei realen Digitalfiltern, welche mit einem Signalprozessor realisiert sind, erst einen Abtasttakt nach Einlesen von $x(n)$ auf. Somit dauert die Antwortfolge 10 Abtasttakte.



Die hohe Koeffizientenanzahl ist der große Nachteil eines FIR-Filters. Wenn man in der Audiotechnik steilflankige Filter erzeugen will, so muss man viele Filterkoeffizienten haben. *Dies bedeutet aber eine sehr lange Audiosignalverzögerung, was man beim Spiel auf Musikinstrumenten wie z.B. Elektrogitarren akustisch eventuell auch hört.*

Kurzdaten der nichtrekursiven Filter (FIR-Filter):

- Amplituden- und Phasengang getrennt voneinander vorgebar
- linearer Phasengang erreichbar
- Impulsantwortfolge ist begrenzt und besteht aus den Filterkoeffizienten
- keine Grenzyklen
- Bei symmetrischen Filterkoeffizienten ist ein linearer Phasengang vorhanden
- Bei symmetrischen Filterkoeffizienten ist die Gruppenlaufzeit konstant => wichtig für die verzerrungsfreie Übertragung von Impulsen
- Die Gleichspannungsverstärkung eines FIR-Filters ist gleich der Summe aller Filterkoeffizienten
- Die Einheitsimpulsantwortfolge eines FIR-Filters ist gleich der Folge der Filterkoeffizienten
- Multipliziert man alle Filterkoeffizienten mit einer Konstanten, so entspricht dies einer Änderung der Filtergrundverstärkung. Die Filtercharakteristik bleibt erhalten
- Die Verstärkung bei der halben Abtastfrequenz ist die mit +1 und -1 gewichtete Koeffizientensumme

3.2. Herleitung des Frequenz- und Phasenganges von FIR-Filtern

Die z-Übertragungsfunktion eines FIR-Filters N-ter Ordnung ist

$$H(z) = \sum_{k=0}^N A_k \cdot z^{-k} = A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2} + \dots + A_N \cdot z^{-N}$$

Gleichung 21: z- Übertragungsfunktion eines FIR-Filters

Es gilt auch:

$$s' = \frac{s}{f_a} = s \cdot t_a$$

Formel 1: Normierung der komplexen Frequenz in der s-Ebene

$$e^{s' \cdot t_a} = e^s = z$$

Formel 2: Beziehung der normierten komplexen Frequenz in der s-Ebene zur z-Ebene

Setzt man in Gleichung 21 die Eulersche Beziehung

$$z^{-1} = e^{-j2\pi f'} = \cos(2\pi f') - j\sin(2\pi f') \text{ mit } f' = \frac{f}{f_a}$$

Gleichung 22: Eulersche Beziehung in „z“ einsetzen

ein, folgt daraus der komplexe Frequenzgang

$$\underline{A}(f') = \sum_{k=0}^N A_k \cdot e^{-j2\pi k f'}$$

Gleichung 23: Komplexer Frequenzgang des FIR-Filters

Diese Beziehung kann vereinfacht werden, wenn die Koeffizienten A_k symmetrisch sind, nämlich:

$A_{N-k} = A_k$ gerade Symmetrie (spiegelsymmetrisch)
 $A_{N-k} = -A_k$ ungerade Symmetrie (punktsymmetrisch)

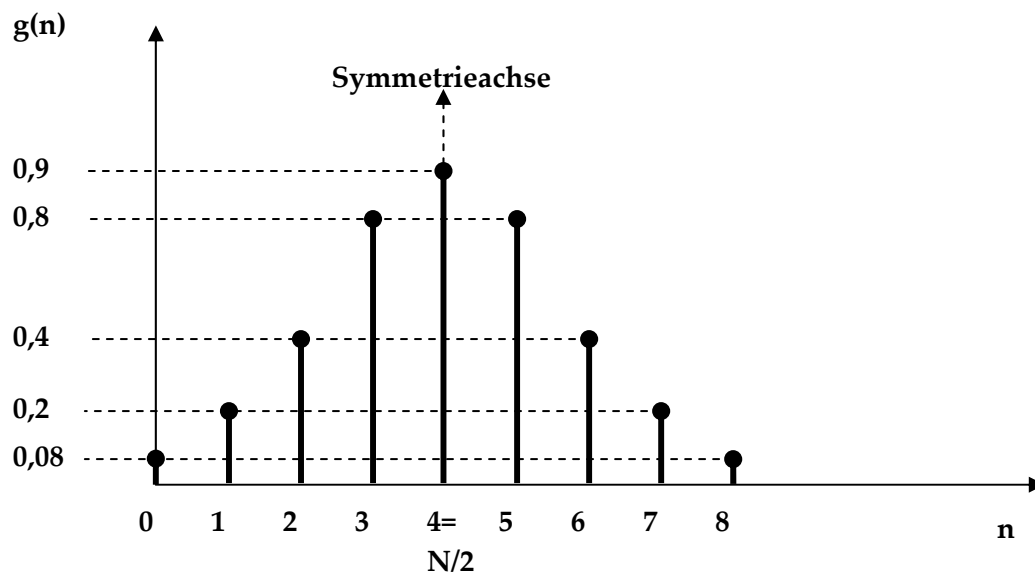


Abbildung 22: Impulsantwortfolge eines FIR-Filters mit Spiegel­symmetrie und N =gerade

In Abbildung 22 sehen wir die Impulsantwortfolge eines spiegelsymmetrischen FIR-Filters mit 9 Koeffizienten ($N=8$ =gerade). Die Werte $g(n)$ bei $n = 0$ bis 3 sind gleich den Werten $g(n)$ bei $n = 8$ bis 5. **Der Symmetriepunkt liegt genau auf dem mittleren Filterkoeffizienten (=4)!**

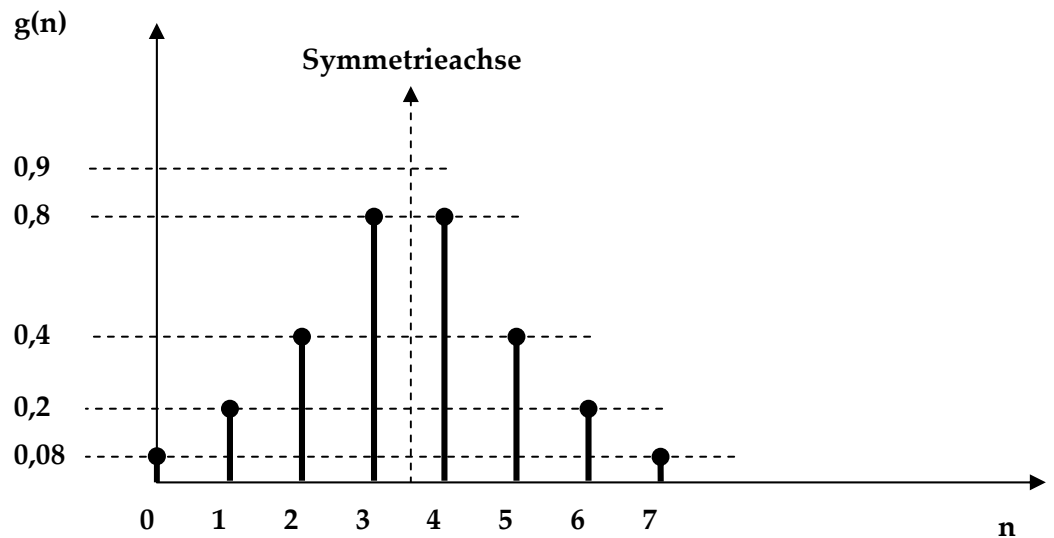


Abbildung 23: Impulsantwortfolge eines FIR-Filters mit Spiegelsymmetrie und $N=7$ =ungerade

In Abbildung 23 sehen wir die Impulsantwortfolge eines spiegelsymmetrischen FIR-Filters mit 8 Koeffizienten ($N=7$ =ungerade). Die Werte $g(n)$ bei $n = 0$ bis 3 sind gleich den Werten $g(n)$ bei $n = 7$ bis 4. **Der Symmetriepunkt liegt mittig zwischen den beiden mittleren Filterkoeffizienten (zwischen $n=3$ und $n=4$)!**

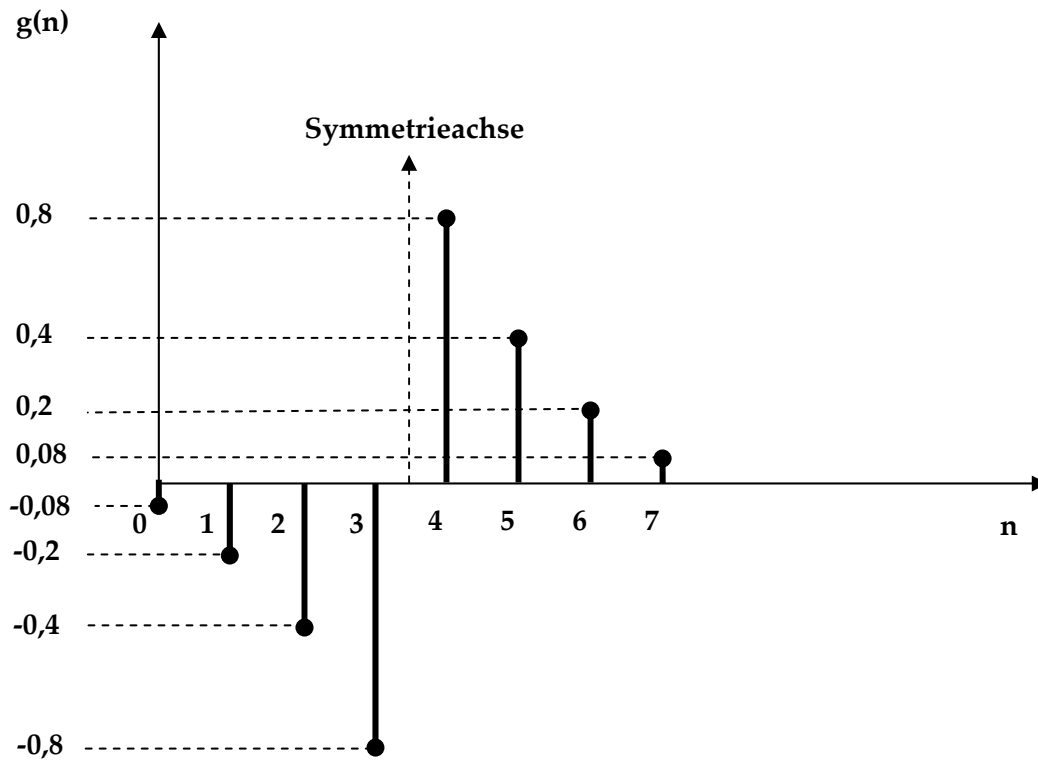


Abbildung 24: Impulsantwortfolge eines FIR-Filters mit Punktsymmetrie und $N=$ ungerade

In Abbildung 24 sehen wir die Impulsantwortfolge eines punktsymmetrischen FIR-Filters mit 8 Koeffizienten ($N=7=$ ungerade). Die Werte $g(n)$ bei $n = 0$ bis 3 sind gleich den negativen Werten $g(n)$ bei $n= 7$ bis 4. **Der Symmetriepunkt liegt mittig zwischen den beiden mittleren Filterkoeffizienten (zwischen $n=3$ und $n=4$)!**

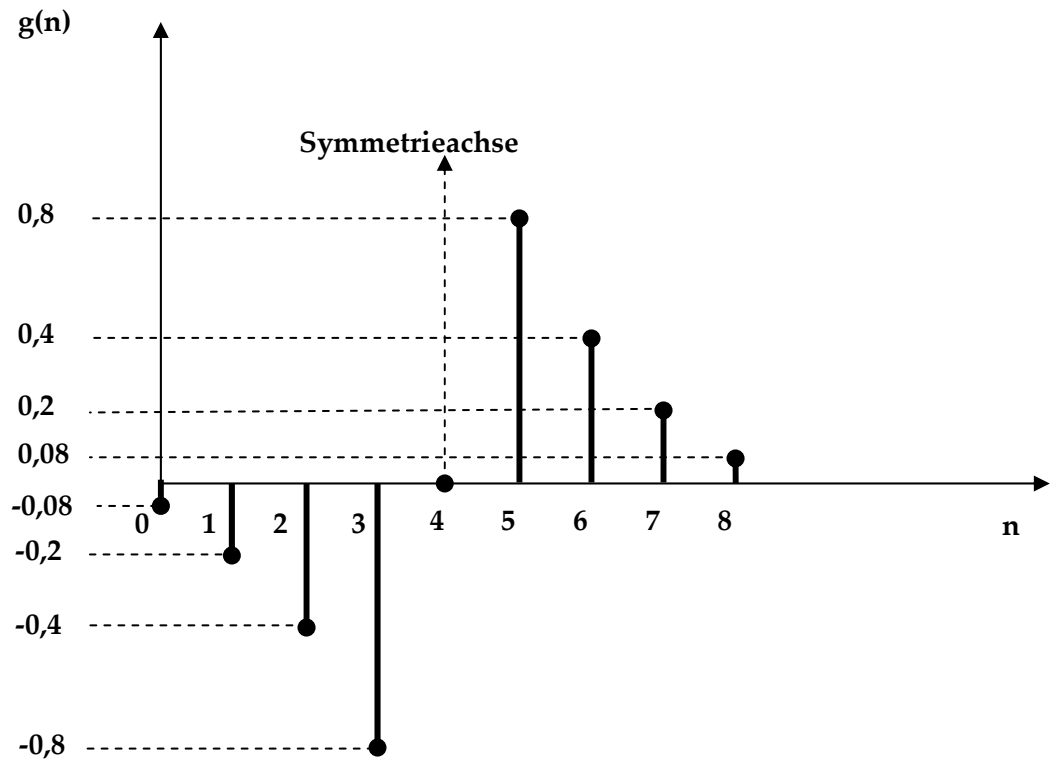


Abbildung 25: Impulsantwortfolge eines FIR-Filters mit Punktsymmetrie und N=gerade

In Abbildung 25 sehen wir die Impulsantwortfolge eines punktsymmetrischen FIR-Filters mit 9 Koeffizienten (N=8=gerade). Die Werte g(n) bei n = 0 bis 3 sind gleich den negativen Werten g(n) bei n= 8 bis 4. **Der Filterkoeffizient am Symmetriepunkt (n=4) muss 0 sein!**

Bei Symmetrie lassen sich in Gleichung 23 zwei Terme mit betragsmäßig gleichen Koeffizienten zusammenfassen und ein gemeinsamer Phasenfaktor lässt sich ausklammern.

Gleichung 23 vereinfacht sich dann unter folgenden Randbedingungen:

- $A_{N-k} = A_k$ gerade Symmetrie (spiegelsymmetrisch)
- $A_{N-k} = -A_k$ ungerade Symmetrie (punktsymmetrisch)

bei Spiegelsymmetrie der Koeffizienten nach längerer Umformung der Gleichung zu:

$$\underline{A}(f') = e^{-j \pi \cdot N \cdot f'} \cdot \sum_{k=0}^N A_k \cdot \cos(\pi \cdot (N - 2 \cdot k) \cdot f')$$

Gleichung 24: Komplexer Frequenzgang bei Spiegelsymmetrie

bei Punktsymmetrie der Koeffizienten zu:

$$\underline{A}(f') = e^{-j \cdot \pi \cdot N \cdot f'} \cdot \sum_{k=0}^N A_k \cdot \sin(\pi \cdot (N - 2 \cdot k) \cdot f')$$

Gleichung 25: Komplexer Frequenzgang bei Punktsymmetrie

Man erhält eine Darstellung nach Betrag und Phase

$$\underline{A}(f') = \begin{cases} B(f') \cdot e^{-j \cdot \pi \cdot N \cdot f'} & \text{für Spiegelsymmetrie} \\ B(f') \cdot j \cdot e^{-j \cdot \pi \cdot N \cdot f'} & \text{für Punktsymmetrie} \end{cases}$$

Gleichung 26: Komplexer Frequenzgang nach Betrag und Phase

Mit $B(f')$ = Betrag des Frequenzganges = Wert der Summe in Gleichung 24 und Gleichung 25. Die Phasenverschiebung folgt aus der e-Funktion vor der Summation in Gleichung 24 und Gleichung 25.

$$\varphi(f') = \begin{cases} -\pi \cdot N \cdot f' & \text{für Spiegelsymmetrie} \\ -\pi \cdot N \cdot f' + \frac{\pi}{2} & \text{für Punktsymmetrie} \end{cases}$$

Gleichung 27: Phasengang bei Spiegel- und Punktsymmetrie

Man erkennt in Gleichung 27 die „lineare Phase“ ($\varphi(f') = K \cdot f'$), die bei beliebigen symmetrischen Koeffizienten exakt erfüllt ist.

Phasengang linear

Die Gruppenlaufzeit eines Filters gibt an, wie lange (gerechnet in Abtastzeiten t_a) ein Impuls vom Filterein- zum -ausgang benötigt. Sie ergibt sich aus der negativen Ableitung des Phasenganges nach der Kreisfrequenz.

Gruppenlaufzeit

Wir berechnen hier nur die Ableitung für Spiegelsymmetrie, weil der konstante Summand $\frac{\pi}{2}$ bei Punktsymmetrie durch die Ableitung sowieso wegfällt!

$$t_{gr} = -\frac{d\varphi}{d\omega} = -\frac{d(-\pi \cdot N \cdot f')}{d\omega} = -\frac{d(-\pi \cdot N \cdot \frac{\omega \cdot T_a}{2 \cdot \pi})}{d\omega}$$

$$t_{gr} = -(-\pi \cdot N \cdot \frac{T_a}{2 \cdot \pi}) = \frac{1}{2} \cdot N \cdot T_a$$

Gleichung 28: Gruppenlaufzeit eines symmetrischen FIR-Filters

Die Gruppenlaufzeit ist demnach konstant und frequenzunabhängig. Laufzeitverzerrungen können also bei diesen Filtern nicht auftreten.

3.3. Berechnung der Filterkoeffizienten von FIR-Filtern

3.3.1. Einleitung

Zur Berechnung der Filterkoeffizienten sind zwei Verfahren gebräuchlich:

- die Fenstermethode und
- der Remez-Exchange-Algorithmus.

Der Remez-Exchange-Algorithmus basiert auf einer Approximation des gewünschten Frequenzganges mit Tschebyscheff-Funktionen und soll hier nicht weiter vertieft werden. Er bildet die Basis vieler Computerprogramme zur Filterberechnung, ist aber sehr kompliziert in der Herleitung.

3.3.2. Berechnung der Filterkoeffizienten mit der Fenstermethode

FIR-Filter besitzen eine besonders anschauliche Impulsantwort. Legt man am Eingang die Einheitsimpulsfolge (siehe Abbildung 26) an, so erscheint am Ausgang die Einheitsimpulsantwortfolge, welche nichts anderes als die Folge der Filterkoeffizienten ist.

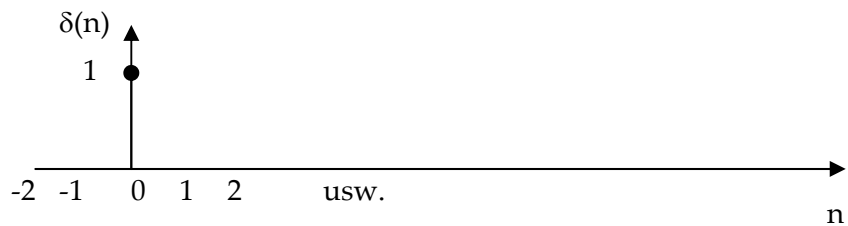


Abbildung 26: Die Einheits-Impulsfolge als Anregung eines FIR-Filters

Mit der Definition:

$$\delta(n) = \begin{cases} 1 & \text{für } n = 0 \\ 0 & \text{sonst} \end{cases}$$

In Abbildung 27 sehen wir als Beispiel einer Impulsantwortfolge die Antwortfolge des Tiefpasses 2. Ordnung aus Abbildung 75.

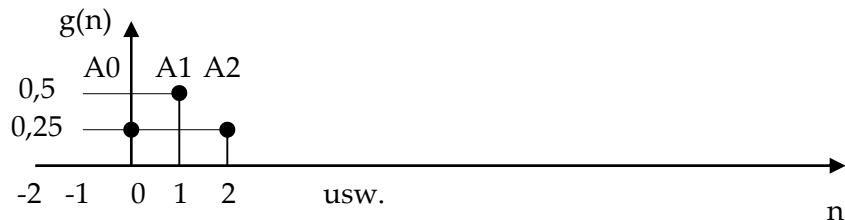


Abbildung 27: Die Einheitsimpulsantwortfolge als Ausgang eines FIR-Filters

Nun muss man wissen, dass die Impulsantwortfolge die inverse Fouriertransformierte des Frequenzganges ist.

Somit erhalten wir als Gleichung für die Filterkoeffizienten

$$\{y(k \cdot T_A)\} = A_0, A_1, A_2, \dots = \{A_k\} = \left\{ \int_{-\frac{f_A}{2}}^{+\frac{f_A}{2}} A_w(jf) \cdot \exp(j \cdot 2 \cdot \pi \cdot f \cdot k \cdot T_a) df \right\}$$

$$\{y(k \cdot T_A)\} = A_0, A_1, A_2, \dots = \{A_k\} = \left\{ \int_{-\frac{f_A}{2}}^{+\frac{f_A}{2}} A_w(jf) \cdot \exp(j \cdot 2 \cdot \pi \cdot k \cdot \frac{f}{f_a}) df \right\}$$

Gleichung 29: Bestimmung der Roh-Filterkoeffizienten mittels inverser Fouriertransformation

Zur Berechnung der Filterkoeffizienten „Ak“ gibt man den gewünschten Frequenzgang $A_w(jf)$ vor.

Sehr einfach berechenbar ist nun aber der Fall eines idealen Tiefpasses mit rechteckigem Frequenzgang.

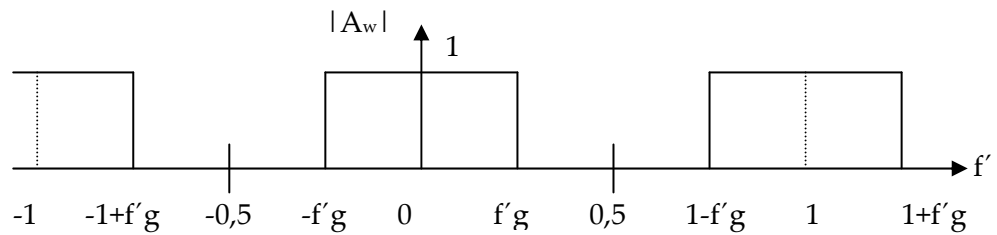


Abbildung 28: Normierter Wunschfrequenzgang eines idealen Tiefpasses mit periodischer Fortsetzung

Der Wunschfrequenzgang $A_w(jf)$ hat den Betrag 1 im Durchlassbereich und 0 im Sperrbereich. Wir wollen noch eine konstante Gruppenlaufzeit \Rightarrow „symmetrische Filterkoeffizienten“ fordern. Dadurch kann man $A_w(jf)$ wie in Gleichung 26 darstellen.

Gleichung 26 sagt nämlich:

$$\underline{A}(f') = \begin{cases} B(f') \cdot \exp(-j \cdot \pi \cdot N \cdot f') & \text{für Spiegelsymmetrie} \\ B(f') \cdot j \cdot \exp(-j \cdot \pi \cdot N \cdot f') & \text{für Punktsymmetrie} \end{cases}$$

Mit dem Amplitudenfrequenzgang $B(f') = 1$ im Durchlassbereich des Filters wird daraus:

$$A_w(jf) = \begin{cases} \exp(-j \cdot \pi \cdot f \cdot N \cdot T_A) & \text{für } -fg \leq f \leq fg \\ 0 & \text{sonst} \end{cases}$$

Gleichung 30: Wunschfrequenzgangvorgabe bei gerader Symmetrie

Setzt man Gleichung 30 in Gleichung 29 ein, so erhält man

$$A_{kr} = 2 \cdot fg \cdot \frac{\sin[(2 \cdot k - N) \cdot \pi \cdot f'g]}{[(2 \cdot k - N) \cdot \pi \cdot f'g]} = \text{für } k=0,1,2,\dots,N$$

Gleichung 31: Koeffizienten des idealen Tiefpasses

Die Filterrohkoeffizienten berechnen sich also im Prinzip mit der Funktion $\sin(x)/x$.

Diese Funktion wird auch „sinc“-Funktion genannt.



Dateiname: „sinc.sce“

Durch Simulation mit SCILAB erhalten wir das in Abbildung 29 dargestellte Ergebnis.

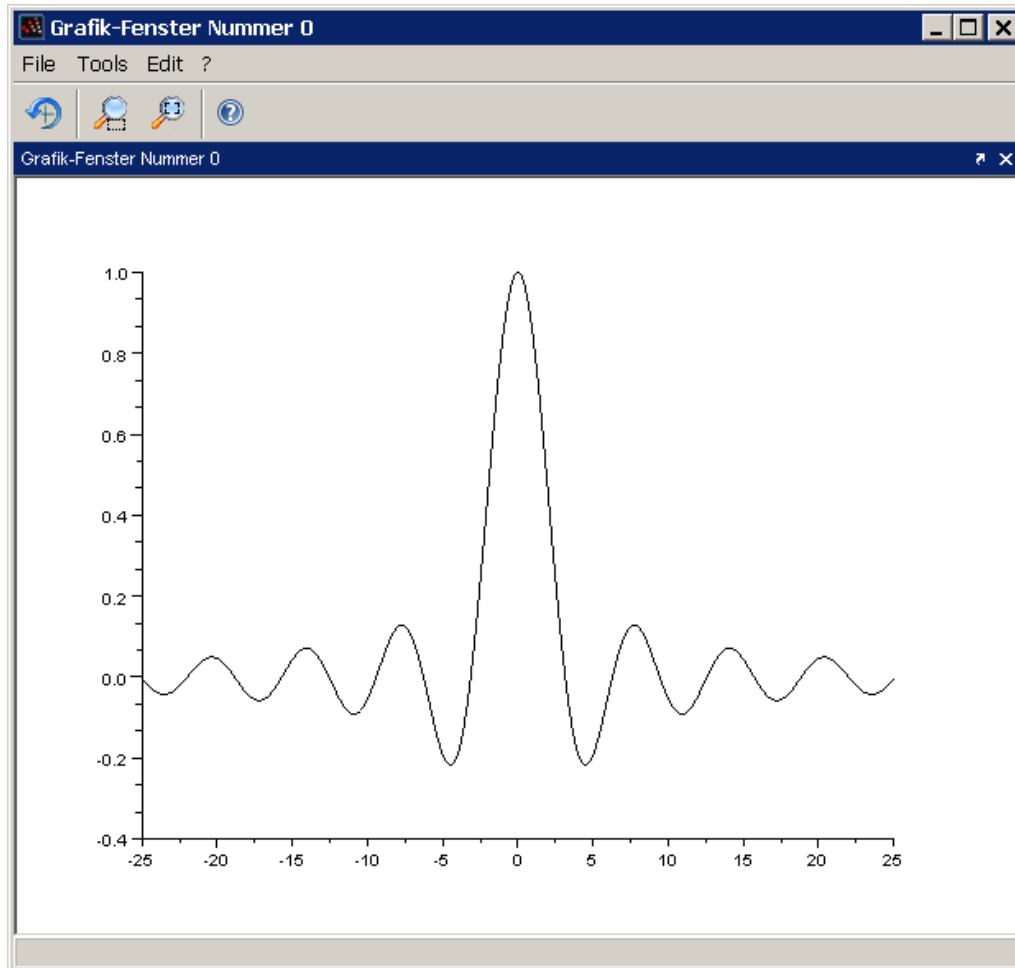


Abbildung 29: Simulation der „sinc“-Funktion mit SCILAB

Die mit Gleichung 31 berechneten Filterkoeffizienten sind nur Rohkoeffizienten (Index „ r “), da wir bei der Berechnung ja von einer **unendlichen Ordnungszahl** „ N “ ausgegangen sind.

Fensterfunktion

In der Praxis benutzt man nur eine endliche Anzahl „ N “ von Koeffizienten. Dadurch ergibt sich eine Abweichung des Frequenzganges vom gewünschten Verhalten.

In der Praxis multipliziert man die Formel für die Rohkoeffizienten (Gleichung 31) mit einer *Fensterfunktion* und begrenzt damit die Koeffizientenzahl auf einen endlichen Wert.

Es gibt verschiedene Fensterfunktionen, welche verschiedene Auswirkungen auf die dadurch berechneten Filter haben. Im Prinzip modifiziert eine

Fensterfunktion die endlichen Koeffizienten derart, dass ein bestimmter Frequenzgang entsteht.

Die einfachste Fensterfunktion ist das Rechteckfenster, welches in der Praxis nicht eingesetzt wird. Es verursacht zu starke Schwinger im Frequenzgang.

Als häufig benutzte Fensterfunktion leistet das **Hamming-Fenster** gute Dienste, weil es gute Ergebnisse bei kleinem Rechenaufwand erbringt.

$$W_k = 0,54 - 0,46 \cdot \cos\left(\frac{2 \cdot \pi \cdot k}{N}\right) \text{ für } k = 0, 1, 2, \dots, N$$

Gleichung 32: Formel des Hamming-Fensters

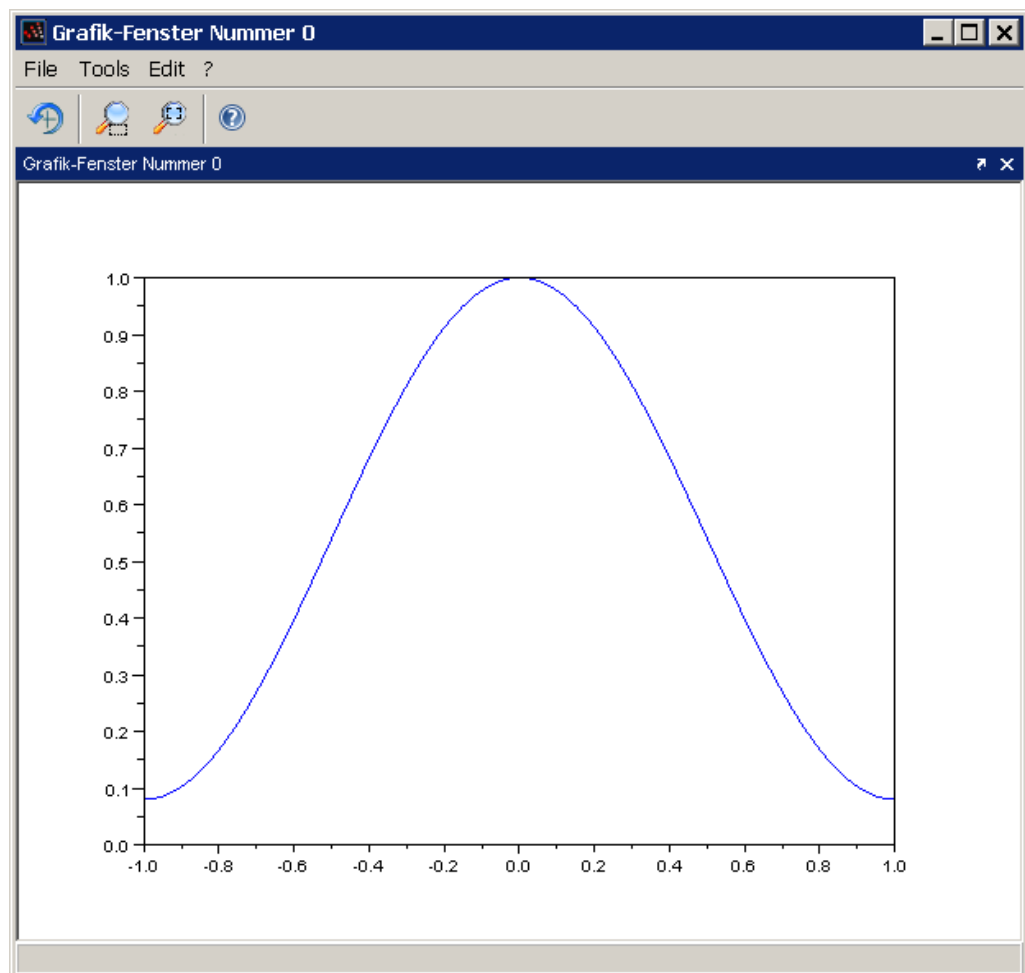


Abbildung 30: Hamming-Fenster im Bereich +1



Dateiname: „hamming.sce“

$W_k(0)$ und $W_k(N)$ werden dabei immer 0,08. In der Mitte ($k=N/2$) hat die

Fensterfunktion bei geradem N den Wert 1, weil $\cos\left(\frac{2 \cdot \pi \cdot \frac{N}{2}}{N}\right) = \cos(\pi) = -1$

ist und dadurch die Summe $0,54 + 0,46 = 1,0$ entsteht.

Wir wollen nun an einem Wertebeispiel eines Tiefpasses ($N=5$, $f'g = 0,25$) das Vorgehen erläutern. Dabei werden wir den Ausdruck $2 \cdot f'g$ in der Formel für die Rohkoeffizienten (Gleichung 31) erst mal nicht berücksichtigen, da er nur eine multiplikative Konstante ist!

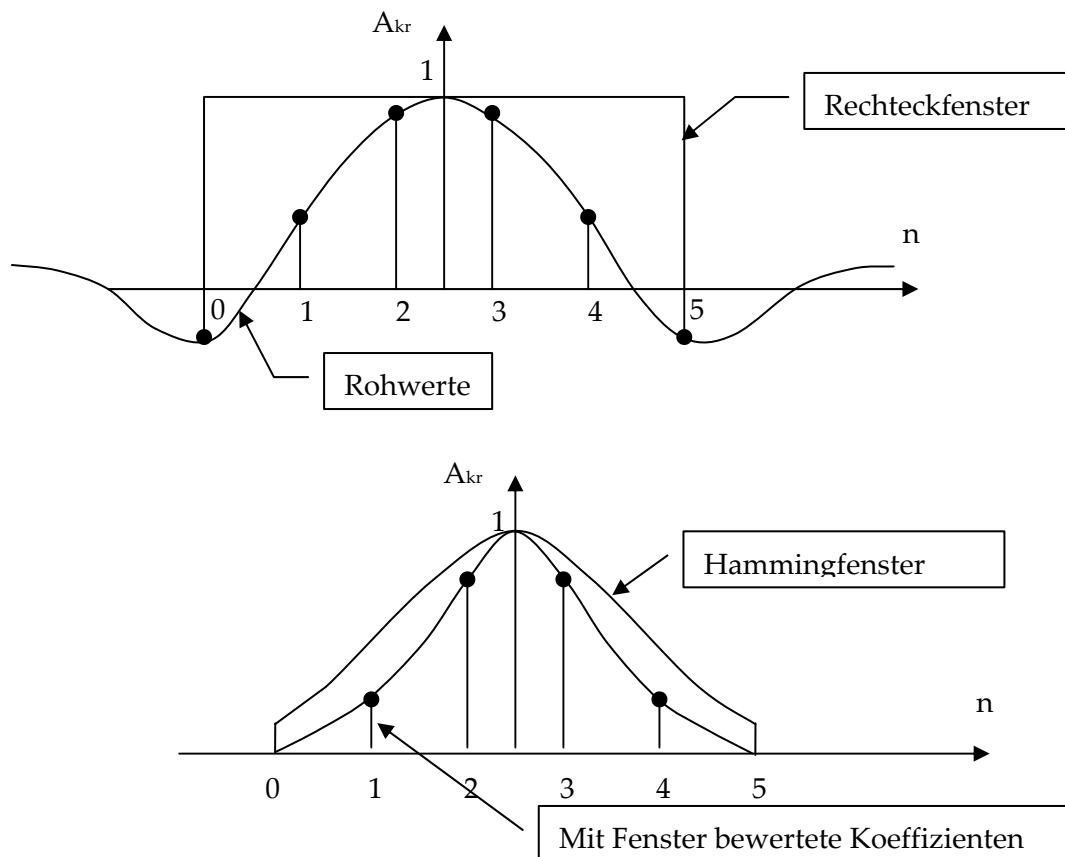


Abbildung 31: Schritte zur Berechnung der Koeffizienten eines FIR-Filters

In Abbildung 31 oben sehen wir die Rohkoeffizienten (0 bis 5) eines Tiefpasses mit der normierten Grenzfrequenz 0,25, welche sich aus der Berechnung mit Gleichung 31 unter Verzicht auf die multiplikative Konstante $2 \cdot f'g$ ergeben.

Mit SCILAB erhalten wir folgende Werte für die Rohkoeffizienten (Index „r“ = roh):

$$A0_r = A5_r = -0.0900316$$

$$A1_r = A4_r = 0.1500527$$

$$A2_r = A3_r = 0.4501582$$

Durch Bewertung mit dem Hammingfenster (H_x = Werte des Hammingfensters)

$$H0=H5 = 0.08; \quad H1 = H4 = 0.3978522 \quad ; H2 = H3 = 0.9121478$$

erhält man die Filterkoeffizienten (Index „rh“ = roh mit Hamming) zu:

$$A0_{rh} = A5_{rh} = A0_r * H0 = -0.0072025$$

$$A1_{rh} = A4_{rh} = A1_r * H1 = 0.0596988$$

$$A2_{rh} = A3_{rh} = A2_r * H2 = 0.4106108$$

In der Praxis dividiert man nach der Fensterung noch jeden Koeffizienten durch die Summe aller Koeffizienten, so dass die Gleichspannungsverstärkung 1 wird.

Damit ergeben sich die Filterkoeffizienten zu

$$A0 = A5 = -0.0077763$$

$$A1 = A4 = 0.0644546$$

$$A2 = A3 = 0.4433217$$



Dateiname: „FIR Rohkoeffizienten mit sinc diskret programmiert.sce“

Durch Simulation mit SCILAB erhalten wir das in Abbildung 32 dargestellte Ergebnis.

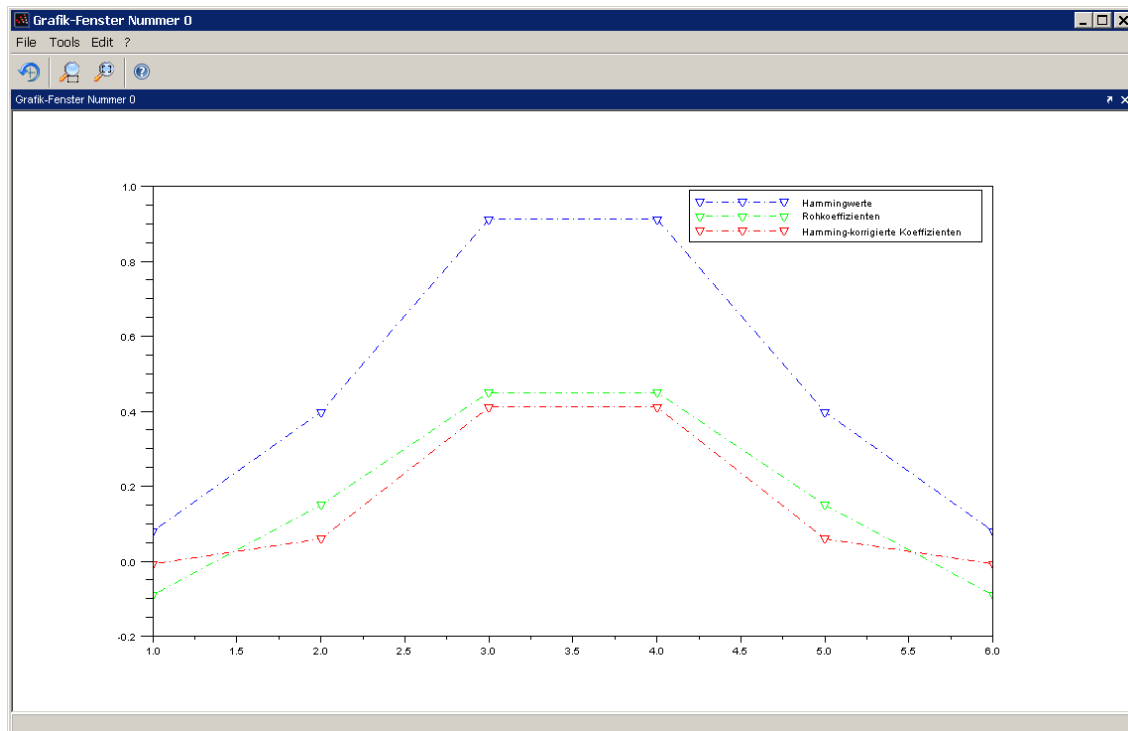


Abbildung 32: Ergebnis der FIR-Filterberechnung mit SCILAB

Da wir die Originalkoeffizienten mit dem Hammingfenster modifiziert haben, ändert sich die Grenzfrequenz des Tiefpasses. Man muss nun die Grenzfrequenz leicht modifizieren (hier auf 0,32) und den gesamten Designablauf wiederholen! Dies führt zwangsläufig zu einem iterativen Vorgang.

Am Ende der Iterationen erhält man folgende Werte für die Koeffizienten:

$$A_0 = A_5 = -0,00979$$

$$A_1 = A_4 = +0,00979$$

$$A_2 = A_3 = +0,5$$

In der Praxis realisiert man mit obigem Verfahren Tiefpässe mit ungeradem „N“, weil dadurch der Frequenzgang bei der halben Abtastfrequenz immer 0 ist. Außerdem erhält man eine gerade Anzahl von Filterkoeffizienten, was bei der Realisierung in integrierten Schaltungen nützlich ist.

Um zu einer möglichst einfachen Realisierung von FIR-Filter zu gelangen, kann man sich fragen:

Wie muss die Grenzfrequenz gewählt werden, damit viele Filterkoeffizienten 0 werden.

Fall 1: $f'_g = 1/2$

Hier verschwinden in gerader Ordnung (ungerade Koeffizientenzahl => N = gerade) alle Koeffizienten bis auf den mittleren, welcher den Wert 1 besitzt. Das resultierende Filter ist ein Allpass => kein Tiefpass wie gewünscht!

Fall 2: $f'_g = 1/4$

Diesen Filtertyp nennt man Halbbandfilter, weil die Grenzfrequenz die Hälfte des maximal möglichen Durchlassbereiches ist. Hier verschwindet in gerader Ordnung (ungerade Koeffizientenzahl => N = gerade) jeder zweite Koeffizient.

Halbbandfilter schaltet man in der Praxis in Serie, um dadurch eine fortlaufende Halbierung der Grenzfrequenz und damit fast beliebige Grenzfrequenzen zu erreichen.

3.4. Einführungsbeispiel

Wir wollen zur Einführung das einfachst mögliche FIR-Filter besprechen. Die Schaltung in Abbildung 33 zeigt ein FIR-Filter 1. Ordnung (N=1). Diese Art der Darstellung wie ein Blockschaltbild findet man in der Praxis sehr häufig. Seine Verstärkung für Gleichspannung ist 1 (Summe aller Filterkoeffizienten). Wir werden sehen, dass es ein Tiefpass. Bei der halben Abtastfrequenz ist die Verstärkung nämlich 0. Man kann ihn sehr gut wegen der gleichen Koeffizienten zur Mittelwertbildung einsetzen.

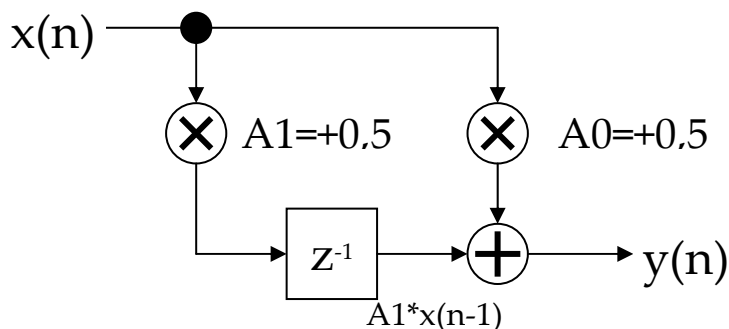


Abbildung 33: Tiefpass/Interpolator 1. Ordnung

Die Formel für die Ausgangssignalberechnung kann man direkt aus dem Blockschaltbild ablesen. Der Block mit z^{-1} stellt eine Verzögerung um einen Abtasttakt dar.

$$y(n) = A_0 \cdot x(n) + A_1 \cdot x(n-1)$$

Gleichung 33: Differenzgleichung des FIR-Beispielfilters

Die Anwendung der Z-Transformation mit Verschiebungssatz liefert:

$$y(z) = A_0 \cdot x(z) + A_1 \cdot x(z) \cdot z^{-1}$$

Gleichung 34: Berechnungsformel für y(z) des FIR-Beispielfilters

Ein Umstellen nach $\frac{y(z)}{x(z)}$ und Einsetzen der Koeffizientenwerte ergibt:

$$H(z) = \frac{y(z)}{x(z)} = 0,5 \cdot (1 + z^{-1})$$

Gleichung 35: z-Übertragungsfunktion des FIR-Beispielfilters

Wir wollen nun den Frequenzgang des Filters berechnen. Dazu ersetzen wir in Gleichung 35 den Ausdruck z^{-1} durch $e^{-s \cdot t_a}$ und wandeln die e-Funktion in die Summe aus Cosinus und Sinus um.

Es gilt nämlich:

$$e^{-jx} = \cos(x) - j \cdot \sin(x)$$

Gleichung 36: Trigonometrische Umrechnung der e-Funktion (1)

Wir ersetzen die komplexe Variable „s“ durch „j ω “ = „j · 2 · π · f“:

$$e^{-s \cdot t_a} = e^{-j \cdot 2 \cdot \pi \cdot f \cdot t_a} = e^{-j \cdot 2 \cdot \pi \cdot \frac{f}{f_a}} = e^{-j \cdot 2 \cdot \pi \cdot f'} = \cos(2 \cdot \pi \cdot f') - j \cdot \sin(2 \cdot \pi \cdot f')$$

Gleichung 37: Trigonometrische Umrechnung der e-Funktion (2)

Mit f' => auf Abtastfrequenz normierte Frequenz!

Es ergibt sich dann als normierter komplexer Frequenzgang des Filters:

$$\underline{H}(f') = 0,5 \cdot (1 + \cos(2 \cdot \pi \cdot f') - j \sin(2 \cdot \pi \cdot f'))$$

Gleichung 38: Komplexer Frequenzgang des FIR-Beispielfilters

Nun berechnen wir den Betrag des Frequenzgangs. Dieser ist die Wurzel aus der Summe von Realteil ins Quadrat und Imaginärteil ins Quadrat.

$$|\underline{H}(f')| = 0,5 \cdot \sqrt{(1 + \cos(2 \cdot \pi \cdot f'))^2 + (-\sin(2 \cdot \pi \cdot f'))^2}$$

Gleichung 39: Betragsfrequenzgang des FIR-Beispielfilters (1)

Quadrieren des Realteils unter der Wurzel liefert:

$$|\underline{H}(f')| = 0,5 \cdot \sqrt{1 + 2 \cdot \cos(2 \cdot \pi \cdot f') + \cos^2(2 \cdot \pi \cdot f') + \sin^2(2 \cdot \pi \cdot f')}$$

Gleichung 40: Betragsfrequenzgang des FIR-Beispielfilters (2)

$$„\sin^2(x) = 0,5 - 0,5 \cdot \cos(2x)“ \text{ und } „\cos^2(x) = 0,5 + 0,5 \cdot \cos(2x)“$$

Gleichung 41: Trigonometrische Umrechnung der sinus- und cosinus-Quadrat-Funktion

Die Quadrate der Sinus- und Cosinus-Funktionen werden nun entsprechend Gleichung 41 umgerechnet.

$$|\underline{H}(f')| = 0,5 \cdot \sqrt{1 + 2 \cdot \cos(2 \cdot \pi \cdot f') + (0,5 + 0,5 \cos(4 \cdot \pi \cdot f')) + (0,5 - 0,5 \cos(4 \cdot \pi \cdot f'))}$$

Gleichung 42: Betragsfrequenzgang des FIR-Beispielfilters (3)

Die Terme „ $0,5 \cos(4 \cdot \pi \cdot f')$ “ subtrahieren sich weg!

Somit ergibt sich:

$$\begin{aligned} |\underline{H}(f')| &= 0,5 \cdot \sqrt{2 + 2 \cdot \cos(2 \cdot \pi \cdot f')} \\ |\underline{H}(f')| &= 0,5 \cdot \sqrt{4} \cdot \sqrt{0,5 + 0,5 \cos(2 \cdot \pi \cdot f')} \end{aligned}$$

Gleichung 43: Betragsfrequenzgang des FIR-Beispielfilters (4)

Mit $\cos^2(x) = 0,5 + 0,5 \cdot \cos(2x)$ wird aus $0,5 + 0,5 \cos(2 \cdot \pi \cdot f') \Rightarrow \cos^2(\pi \cdot f')$

Nun kann man einfach die Wurzel ziehen.

$$|\underline{H}(f')| = 0,5 \cdot \sqrt{4} \cdot \sqrt{0,5 + 0,5 \cos(2 \cdot \pi \cdot f')} = 0,5 \cdot \sqrt{4} \cdot \sqrt{\cos^2(\pi \cdot f')} = 0,5 \cdot 2 \cdot \sqrt{\cos^2(\pi \cdot f')}$$

Gleichung 44: Betragsfrequenzgang des FIR-Beispielfilters (5)

$$|\underline{H}(f')| = \cos(\pi \cdot f')$$

Gleichung 45: Betrag des Frequenzgangs des FIR-Beispielfilters

In Gleichung 45 sehen wir also, dass der Betrag des Frequenzganges eine Cosinusfunktion ist. Da der Cosinus auch negativ werden kann, müssen wir oberhalb der halben Abtastfrequenz (wo „cos“ negativ wird) den Betrag der Cosinusfunktion benutzen!

Die Grenzfrequenz ergibt sich aus der Frequenz, bei welcher der Betrag des Frequenzganges $\frac{1}{\sqrt{2}}$ ist.

Durch beidseitiges Anwenden von \cos^{-1} (Arcuscosinus) auf Gleichung 45 erhalten wir:

$$|\underline{H}(f')| = \frac{1}{\sqrt{2}} = \cos(\pi \cdot f'_g) \Rightarrow \cos^{-1}\left(\frac{1}{\sqrt{2}}\right) = \cos^{-1}(\cos(\pi \cdot f'_g))$$

$$f'_g = \frac{\cos^{-1}\left(\frac{1}{\sqrt{2}}\right)}{\pi} = \frac{0,785398163}{3,141592654} = 0,25$$

Gleichung 46: Grenzfrequenz des FIR-Beispielfilters

Der Phasengang ergibt sich aus \tan^{-1} (Arcustangens) von (Imaginärteil/Realteil).

$$\varphi = \tan^{-1} \frac{-\sin(2 \cdot \pi \cdot f')}{1 + \cos(2 \cdot \pi \cdot f')}$$

Gleichung 47: Phasengang des FIR-Beispielfilters (1)

Mit $1 + \cos(2 \cdot \pi \cdot f') = 2(\cos(\pi \cdot f'))^2$ wie oben schon mal benutzt wird daraus:

$$\varphi = \tan^{-1} \frac{-\sin(2 \cdot \pi \cdot f')}{2(\cos(\pi \cdot f'))^2}$$

Gleichung 48: Phasengang des FIR-Beispielfilters (2)

Nun haben wir das Problem, dass im Zähler das Argument des Sinus doppelt so groß ist wie im Nenner beim Cosinus. Durch die trigonometrische Umformung des Zählers mit „ $\sin(2x) = 2 \sin(x) \cos(x)$ “ erhalten wir dann:

$$\varphi = \tan^{-1} \frac{-2 \cdot \sin(\pi \cdot f') \cdot \cos(\pi \cdot f')}{2(\cos(\pi \cdot f'))^2}$$

Gleichung 49: Phasengang des FIR-Beispielfilters (3)

und daraus durch Kürzen:

$$\varphi = \tan^{-1} \frac{-\sin(\pi \cdot f')}{\cos(\pi \cdot f')} = \tan^{-1}(-\tan(\pi \cdot f')) = -\pi \cdot f'$$

Gleichung 50: Phasengang des FIR-Beispielfilters

Die Gruppenlaufzeit eines Filters gibt an, wie lange (gerechnet in Abtastzeiten t_a) ein beliebig geformter Impuls vom Filterein- zum -ausgang benötigt. Sie ergibt sich aus der negativen Ableitung des Phasenganges nach der Kreisfrequenz.

$$t_{gr} = -\frac{d\varphi}{d\omega} = -\frac{d(-\pi \cdot f')}{d\omega} = -\frac{d(-\pi \cdot \frac{f}{fa})}{d\omega} = -\frac{d(-0,5 \cdot \frac{\omega}{fa})}{d\omega} = \frac{0,5}{fa} = 0,5 \cdot t_a$$

Gleichung 51: Gruppenlaufzeit des FIR-Beispielfilters

3.5. FIR-Filter-Simulation mit LTSpice

Nun wollen wir mittels der Software LTSpice dieses einfache digitale Filter in seinem Amplituden- und Phasengang simulieren. Diese Software ist dafür eigentlich nicht geeignet! Sie wurde ursprünglich zur Simulation analoger Schaltungen entwickelt. Sie ist aber aus der Vorlesung „Bauelemente und Schaltungstechnik II“ schon bekannt und in eingeschränktem Umfang einsetzbar.

Prinzip digitales Filter

Ein digitales FIR-Filter addiert prinzipiell zu bestimmten Zeitpunkten (Zeitabstand $1/fa$, fa = Abtastfrequenz) mit den Filterkoeffizienten gewichtete Eingangssignalwerte zum Ausgangssignal zusammen. Dies kann man auch in LTSpice mittels zeitverzögerter Analogsignale machen. Man muss dazu nur das Eingangssignal in seinem Zeitverlauf verschieben, was der Befehl „delay(Signal, Zeit)“ bewirkt. Er verzögert das Signal mit dem Namen „Signal“ um die Zeit „Zeit“.

Wenn wir also schreiben „ $V = A0 \cdot v(XN) + A1 \cdot \text{delay}(v(XN), TS)$ “, dann meinen wir, dass LTSpice das Steuersignal „V“ (großes „V“!) aus der Summe von „mit Koeffizient $A0$ gewichtetem Eingangssignal $v(XN)$ “ (Achtung:

kleingeschriebenes „v“ => Spannung am Signal „XN“) und „mit Koeffizient A1 gewichtetem um TS verzögertem Eingangssignal v(XN)“ bilden soll. Das Signal V steuert eine formelgesteuert Spannungsquelle. Deren Ausgangssignal YN verhält sich wie „V“. Das Eingangssignal der Schaltung kommt aus dem Knoten „XN“, welcher das Ausgangssignal der Quelle „V1“ ist.

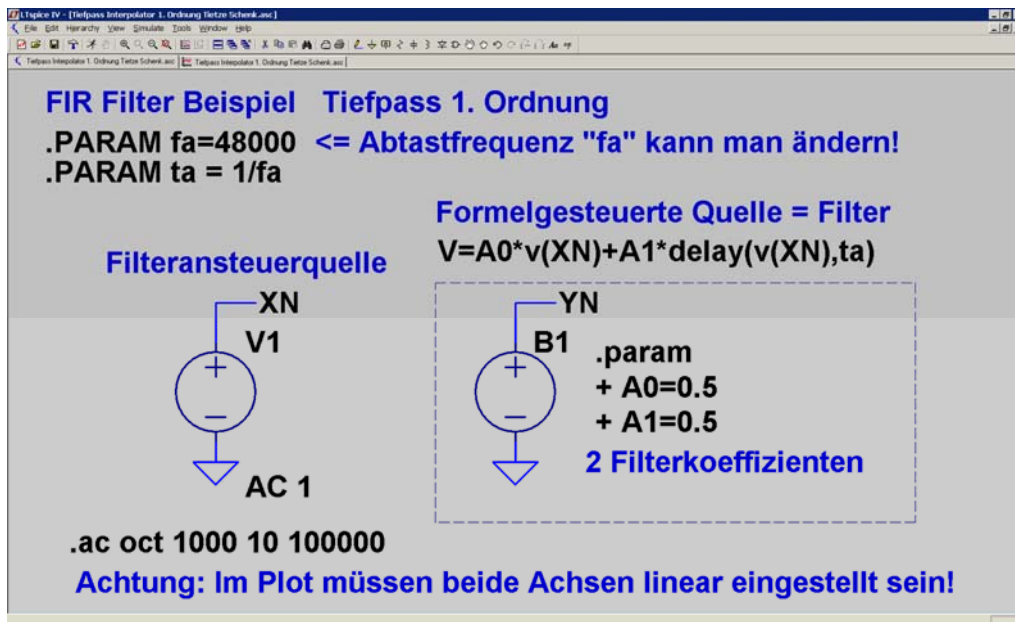


Abbildung 34: Schaltbild des Digitalfilters in der Analogsimulationssoftware LTSpice

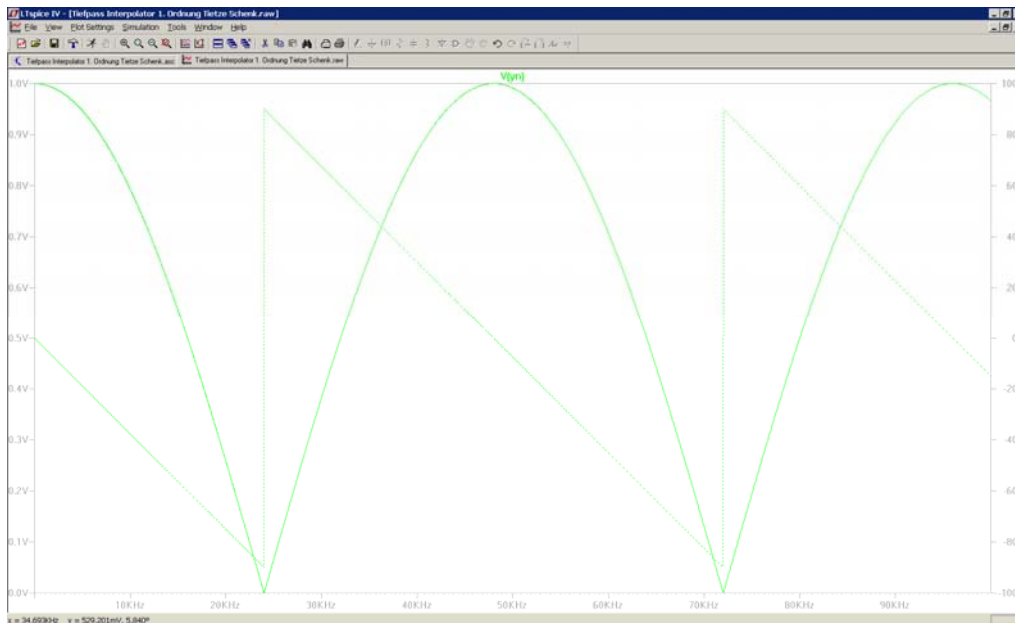


Abbildung 35: Amplituden- und Phasengang in linearer Darstellung des Digitalfilters

Man sieht sehr schön den cosinusförmigen Verlauf des Amplitudenganges und den negativ linearen Verlauf des Phasenganges. Beides wiederholt sich periodisch mit der Abtastfrequenz. Man darf darum nur Signale am Filtereingang einspeisen, welche das Abtasttheorem erfüllen.

Bei der halben Abtastfrequenz 24kHz (FA = 48000HZ) ist der Betrag des Amplitudenganges 0 => es kann also kein Hochpass sein !

Bei sehr tiefer Frequenz (z.B. 10Hz) ist der Betrag des Amplitudenganges 1 => es muss also ein Tiefpass sein !

4. IIR-Filter

4.1. Einleitung

IIR-Filter können als Tiefpass, Hochpass, Bandpass, Bandsperre und Allpass mit ähnlichen Eigenschaften realisiert werden, wie sie analoge Filter aufweisen. Zu beachten ist besonders die begrenzte Wertegenauigkeit (Auflösung in Bit) durch endliche Wortlängen der Signalprozessoren und die dadurch entstehenden Schwingungen und Grenzyklen.

Kurzdaten der rekursiven Filter (IIR-Filter):

- Amplituden- und Phasengang hängen voneinander ab
- normalerweise nichtlinearer Phasengang
- Impulsantwortfolge ist nicht begrenzt
- mögliche Grenzyklen

Die Filterkoeffizienten digitaler Filter kann man heute nach diversen mathematischen Verfahren berechnen. Dies erledigen in der Regel spezielle Computerprogramm wie SCILAB.

Zur Verdeutlichung der Vorgehensweise bei der Koeffizientenberechnung wird nachfolgend ein einfach verständliches Verfahren, die bilineare Transformation, erläutert. Mit diesem Verfahren kann man die Filterkoeffizienten des Digitalfilters aus den Koeffizienten eines bekannten Analogfilters errechnen. Analogfilter sind nämlich schon sehr lange bekannt und sehr gut erforscht. Auch gibt es in der Literatur sehr viele Tabellen mit Filterkoeffizienten von Analogfiltern.

4.2. Die bilineare Transformation

Ausgangspunkt ist die Übertragungsfunktion eines bekannten analogen Referenzfilters, welches die gleichen Eigenschaften wie das gesuchte Digitalfilter haben soll.

Wie schon bekannt, stellt die Gleichung

$$e^{s \cdot t_a} = z \quad \text{mit } t_a = \frac{1}{f_a}$$

Gleichung 52: Zuordnung der s- Ebene zur z-Ebene

die Beziehung der Variablen „z“ zur komplexen Frequenz „s“ dar.

Nun beziehen wir die komplexen Frequenzen der analogen s-Ebene auf die Abtastfrequenz f_a des digitalen Systems. Dies ist gängige Praxis in der digitalen Signalverarbeitung. Man erhält dadurch eine von der Abtastfrequenz unabhängige Darstellung.

So erhält man:

$$s' = \frac{s}{f_a} = s \cdot t_a$$

Gleichung 53: Normierung der Frequenz in der s-Ebene

$$e^{s \cdot t_a} = e^{s'} = z$$

Gleichung 54: Beziehung der normierten Frequenz in der s-Ebene zur z-Ebene

Will man nun eine Gleichung zur Bestimmung von „z“ haben, so muss die Gleichung 54 beidseitig logarithmiert werden.

$$s' = \ln(z)$$

Gleichung 55: Bestimmung von s'

Problem bei der Berechnung von s'

Wenn man nun dieses s' in die Übertragungsfunktion $F(s')$ des analogen Referenzfilters einsetzt, entsteht eine nichtrationale Funktion in „z“, welche man nicht in eine Differenzgleichung umformen kann.

Die Lösung dieses Problems ist wie so oft in der Mathematik eine Reihenentwicklung der Funktion $\ln(z)$ als Näherungslösung.

$$\ln(z) = 2 \cdot \left[\frac{z-1}{z+1} + \frac{1}{2} \cdot \left(\frac{z-1}{z+1} \right)^2 + \frac{1}{3} \cdot \left(\frac{z-1}{z+1} \right)^3 + \dots \right]$$

Gleichung 56: Darstellung von ln(z) als Reihe

Bricht man diese Reihenentwicklung nach dem ersten Glied ab, so ergibt sich:

$$s' = s \cdot t_a = \ln(z) \approx 2 \cdot \frac{z-1}{z+1} \quad (a)$$

$$s' = 2 \cdot \frac{z-1}{z+1} \quad (b)$$

Gleichung 57: Bestimmung der normierten komplexen Frequenzvariablen s'

In Zukunft wollen wir anstelle des Zeichens ungefähr („ \approx “) in Gleichung 57(a) das Gleichheitszeichen („ $=$ “) wie in Gleichung 57 (b) schreiben, weil diese Näherung bei Frequenzen weitab der Abtastfrequenz relativ genau ist. In der Literatur findet man oft auch die Darstellung laut Gleichung 58.

$$s = \frac{2}{t_a} \cdot \frac{z-1}{z+1}$$

Gleichung 58: Bestimmung der komplexen Frequenzvariablen s

Setzt man Gleichung 57(b) anstelle von s' in die Übertragungsfunktion $F(s')$ des analogen Referenzfilters ein, so ergibt sich eine realisierbare Funktion $H(z)$ eines zeitdiskreten Filters, welche das Frequenzverhalten des analogen Referenzfilters näherungsweise approximiert. Wegen des Abbruchs der Reihenentwicklung entsteht ein Fehler, welcher sich aber erst bei Signalfrequenzen in der Nähe der halben Abtastfrequenz f_A deutlicher bemerkbar macht.

Die bilineare Transformation ordnet alle Punkte einer mit f' bezifferten Frequenzachse des analogen Referenzfilters bestimmten Punkten der mit f_d' bezifferten Frequenzachse des digitalen (diskreten) Filters zu. Diese Zuordnung gewinnt man, indem man in der Substitutionsgleichung mit $s' = j \cdot 2 \cdot \pi \cdot f'$ und $z = e^{s' t_a} = e^{j \cdot 2 \cdot \pi \cdot f_d'}$ in beiden Bereichen auf die Frequenzen übergeht.

Aus Gleichung 57 (b) erhält man dann

$$s' = 2 \cdot \frac{z-1}{z+1} \Rightarrow j \cdot 2 \cdot \pi \cdot f' = 2 \cdot \frac{e^{j \cdot 2 \cdot \pi \cdot f_d'} - 1}{e^{j \cdot 2 \cdot \pi \cdot f_d'} + 1}$$

Gleichung 59: Bestimmung von f'

mit

f' = Frequenz des analogen Referenzfilters

f_d' = Frequenz des digitalen Wunschfilters

und nach einigen Umformungen

$$f' = \frac{1}{\pi} \cdot \tan(\pi \cdot f_d')$$

Gleichung 60: Formel für die Frequenz des analogen Referenzfilters f'

Wichtige Frequenzpunkte wie Bandmittenfrequenzen und Grenzfrequenzen sind also beim analogen Referenzfilter so festzulegen, dass sie nach der Transformation beim zeitdiskreten Filter an der richtigen Stelle liegen.

fd'	fd' (digital)	f' (analog)	Fehler[%]
0	0	0	0
1/10000	0,0001	0,0001	0
1/1000	0,001	0,01000003	0
1/100	0,01	0,010003291	0
1/16	0,0625	0,063315773	1,3
1/8	0,125	0,1318483	5,44
1/4	0,25	0,3183099	27,32

Tabelle 1: Wertebeispiele mit Frequenzfehler

Man sieht an den Werten in Tabelle 1 sehr schön, dass der Rechenfehler bei kleinen Frequenzen ($f < f_a/16$) unter 1% liegt.

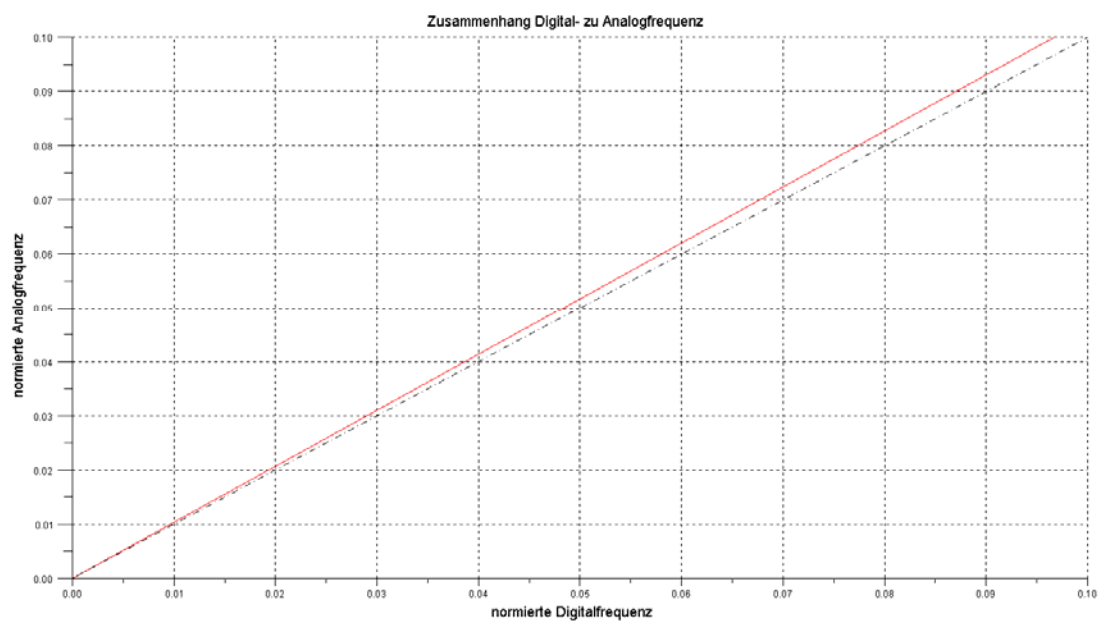


Abbildung 36: Zusammenhang Digital- zu Analogfilterfrequenz

In Abbildung 36 sehen wir das Ergebnis einer Simulation von Gleichung 60 in SCILAB. Rot ist Gleichung 60 dargestellt. Strichliert ist der gewünschte lineare Zusammenhang dargestellt.

Dateiname: „Zusammenhang Analog-Digitalfrequenz.sce“



Beispiel:

Gesucht sind die Filterkoeffizienten eines digitalen Tiefpasses mit der 3dB-Grenzfrequenz f_{dg} von 1kHz und der Abtastfrequenz f_a von 44,1kHz.

Er soll das analoge Referenzsystem laut Gleichung 61

$$F(s') = \frac{1}{1 + s' \cdot T'} \quad \text{mit } T' = \frac{1}{2 \cdot \pi \cdot f'_g}$$

Gleichung 61: Normierte Übertragungsfunktion eines analogen Tiefpasses 1. Ordnung

mit der 3dB-Grenzfrequenz f'_g approximieren.

Gleichung 57(b) als s' eingesetzt in Gleichung 61 ergibt die Übertragungsfunktion des gesuchten digitalen Systems.

$$H(z) = \frac{1}{1 + 2 \cdot T'} \cdot \frac{1 + z^{-1}}{1 + \frac{1 - 2 \cdot T'}{1 + 2 \cdot T'} \cdot z^{-1}}$$

Gleichung 62: Übertragungsfunktion des gesuchten digitalen Tiefpasses

Durch Koeffizientenvergleich mit der allgemeinen Übertragungsfunktion eines digitalen Systems 1. Ordnung

$$H(z) = \frac{A0 + A1 \cdot z^{-1}}{1 + B1 \cdot z^{-1}} = A0 \cdot \frac{1 + \frac{A1}{A0} \cdot z^{-1}}{1 + B1 \cdot z^{-1}}$$

Gleichung 63: Übertragungsfunktion eines Systems 1. Ordnung

ergeben sich die Koeffizienten des digitalen Filters zu:

$$A0 = \frac{1}{1 + 2T'} = \frac{\pi f'_g}{1 + \pi f'_g} ; \quad A1 = A0$$

$$B1 = \frac{1 - 2T'}{1 + 2T'} = \frac{\pi f'_g - 1}{\pi f'_g + 1}$$

Gleichung 64: Gleichungen der Koeffizienten des Digitalfilters

In Gleichung 64 wird f'_g (Grenzfrequenz des analogen Referenz Tiefpasses) gemäß Gleichung 60

$$f'_g = \frac{1}{\pi} \cdot \tan(\pi f'_{dg})$$

so gewählt, dass letztlich das Digitalfilter die gewünschte 3dB-Grenzfrequenz f'_{dg} hat. Man muss nun zuerst die Grenzfrequenz f'_g des analogen Referenzfilters aus der Grenzfrequenz des gesuchten Digitalfilters berechnen und diesen Wert danach in die Gleichung 64 für die Koeffizienten des Digitalfilters einsetzen.

$$f'_g = \frac{f_g}{f_a} = \frac{1}{\pi} \cdot \tan(\pi \cdot f'_{dg}) = \frac{1}{\pi} \cdot \tan\left(\frac{\pi \cdot f_{dg}}{f_a}\right) = \frac{1}{\pi} \cdot \tan\left(\frac{\pi \cdot 10^3}{44100}\right) = \frac{1}{\pi} \cdot 0,0713586$$

$$f'_g = 0,0227141$$

Mit $f'_g = \frac{f_g}{f_a}$ und daraus durch Umformung $f_g = f'_g \cdot f_a = 0,0227141 \cdot f_a$ wird $f_g = 1001,6951 \text{ Hz}$.

Diese Grenzfrequenz des analogen Referenztiefpasses weicht hier nur geringfügig (um 1,6951Hz) von der gewünschten Grenzfrequenz des digitalen Tiefpasses (1kHz) ab. Nähert man sich der halben Abtastfrequenz immer mehr, wird die Abweichung größer. Bei einer gewünschten Grenzfrequenz des Digitalfilters von z.B. 10kHz muss der analoge Referenztiepass eine Grenzfrequenz von 12,123kHz haben.

Mit unseren am Anfang gewählten Wertebeispielen von 1kHz für f_{dg} und 44,1kHz für f_a ergeben sich die Filterkoeffizienten zu:

$$A0 = A1 = \frac{\pi \cdot 0,0227141}{1 + \pi \cdot 0,0227141} = 0,0666057$$

$$B1 = \frac{\pi \cdot 0,0227141 - 1}{\pi \cdot 0,0227141 + 1} = -0,8667884$$

Also sieht die Übertragungsfunktion des gesuchten Digitalfilters wie folgt aus:

$$H(z) = \frac{0,0666057 + 0,0666057 \cdot z^{-1}}{1 - 0,8667884 \cdot z^{-1}}$$

Gleichung 65: Übertragungsfunktion des gesuchten Digitalsystems 1. Ordnung

In den beiden Folgekapiteln wollen wir die Berechnung der Filterkoeffizienten mittels SCLIB üben. Wir beginnen dabei mit dem obigen IIR-Filter.

5. Entwurf rekursiver Filter (IIR-Filter) mit SCILAB

5.1. Einleitung

SCILAB enthält für den Entwurf von Digitalfiltern mächtige Funktionen. Wir können aus Zeitgründen nur wenige Funktionen besprechen.

5.2. Filtereinführungsbeispiel

Es wäre sehr schön, wenn man zur Kontrolle den Verlauf der Amplitudenübertragungsfunktion des IIR-Digitalfilters als Bodediagramm darstellen könnte. Dies lässt sich sehr einfach mit SCILAB realisieren:

Im Kommandofenster von SCILAB geben wir die folgenden 4 Befehlszeilen, jeweils gefolgt von einem Druck der Taste „ENTER“ ein:

1. Die Übertragungsfunktion des gewünschten Filters mit Werten eingeben

```
hz=(0.0666057+0.0666057*%z^-1)/(1-0.8667884*%z^-1);
```

Durch Eingabe von „hz“ (gefolgt von einem Druck der Taste „ENTER“) kann man sich die Formel zur Kontrolle noch einmal anzeigen lassen. *SCILAB formt dabei die Übertragungsfunktion für die Anzeige so um, dass nur positive Exponenten von z auftauchen!*



Achtung

Wir erhalten dann folgende Anzeige:

```
hz =
  0.0666057 + 0.0666057z
  -----
    - 0.8667884 + z
```

Diese Form der Übertragungsfunktion mit positiven Exponenten von „z“ ergibt sich, wenn man die Übertragungsfunktion des Digitalfilters mit z/z multipliziert.

2. Umwandeln des zeitdiskreten Systems in ein zeitlineares System

```
lisys = syslin (1/44100, hz);
```

Die Abtastfrequenz ist 44100Hz. Im Prinzip wird jetzt von SCILAB der Amplitudengang berechnet, indem z^{-1} durch e^{-sta} ersetzt und mittels der Eulerschen Formel umgerechnet wird.

3. Ausgabe des Bodediagrammes zwischen 10Hz und 10kHz

```
bode(lisys,10,10000);
```

Also alle Eingaben als Block zusammengefasst:

```
hz=(0.0666057+0.0666057*%z^-1)/(1-0.8667884*%z^-1);  
lisys = syslin (1/44100, hz);  
bode(lisys,10,10000);
```

Listing 1: Befehle zum Darstellen eines Bodeplots aus der Z-Übertragungsfunktion

Nach manueller Eingabe der obigen Zeilen (oder einfach aus diesem Skript kopieren und in das SCILAB-Kommandofenster einfügen) öffnet sich ein neues grafisches Fenster mit dem Bodediagramm laut Abbildung 37.

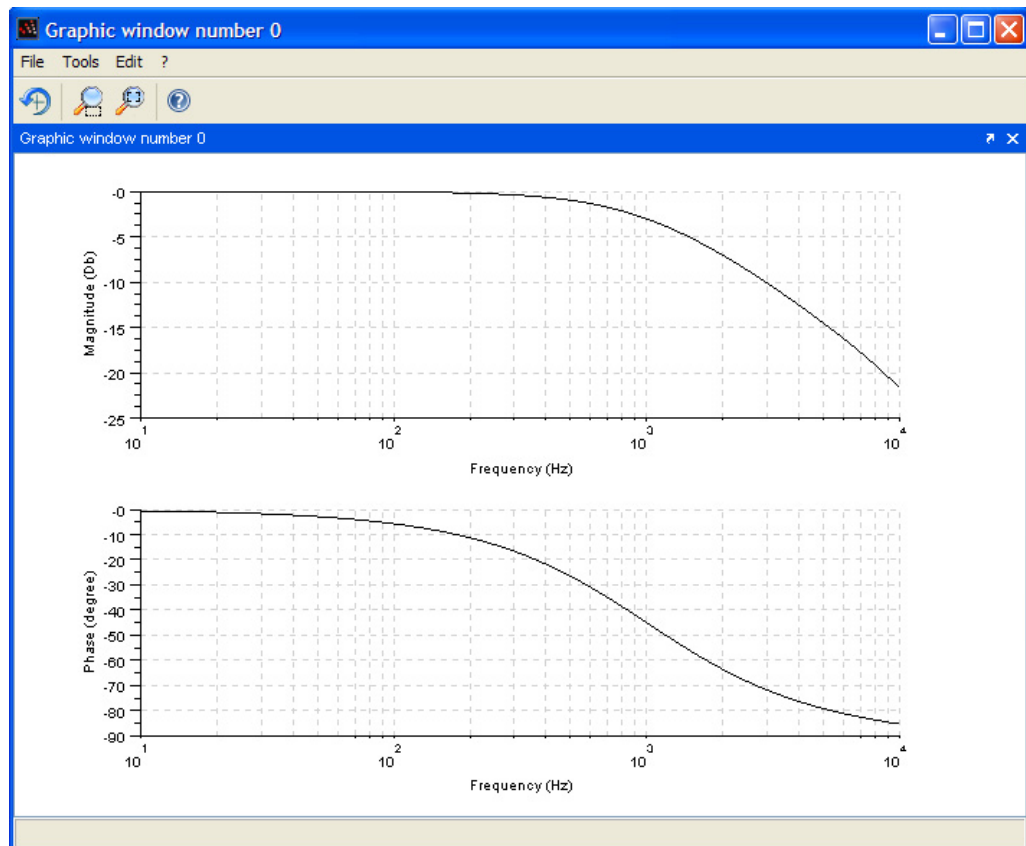
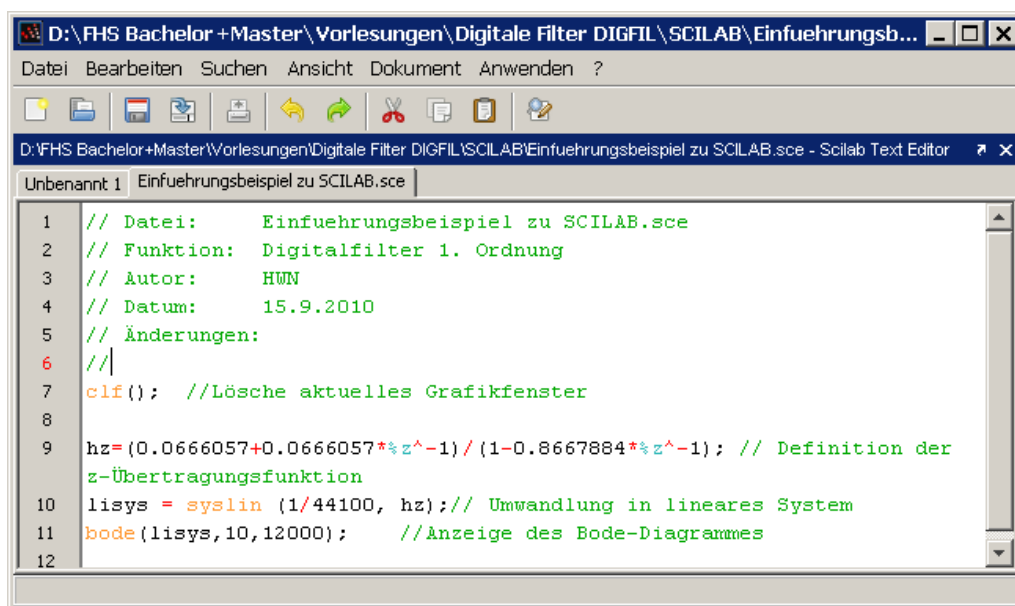


Abbildung 37: SCILAB-Plot von Amplituden- und Phasengang des gesuchten Digitalfilters

Man sieht sehr schön den Abfall der Amplitude auf -3dB bei der Grenzfrequenz 1kHz. Dort beträgt dann die Phasenverschiebung -45 Grad.

Man kann mit SCILAB auch Befehle aus einer Kommandodatei (Dateiname *.SCE) ausführen lassen. Dazu klickt man einfach im SCILAB-Fenster den Menüpunkt *<Applications->Editor>* an. Es öffnet sich ein Editorfenster (Abbildung 38), in welchem man die gewünschten Befehle eingibt. Kommentare kann man wie in der Programmiersprache C++ durch zwei Schrägstriche einleiten. Alle Befehle werden farbig verschieden hervorgehoben.



```
1 // Datei: Einfuehrungsbeispiel zu SCILAB.sce
2 // Funktion: Digitalfilter 1. Ordnung
3 // Autor: HWN
4 // Datum: 15.9.2010
5 // Änderungen:
6 //
7 clf(); //Lösche aktuelles Grafikfenster
8
9 hz=(0.0666057+0.0666057*%z^-1)/(1-0.8667884*%z^-1); // Definition der
z-Übertragungsfunktion
10 lsys = syslin (1/44100, hz); // Umwandlung in lineares System
11 bode(lsys,10,12000); //Anzeige des Bode-Diagrammes
12
```

Abbildung 38: Eingabe der SCILAB-Kommandos im Texteditor

Nach Eingabe aller Befehle kann man sie in einer SCE-Kommandodatei mit *<Datei – Speichern als ...>* speichern.

Zur Ausführung der Befehle wählt man im Editor *<Anwenden – In Scilab ausführen>*.

Dateiname: „Einfuehrungsbeispiel zu SCILAB.sce“



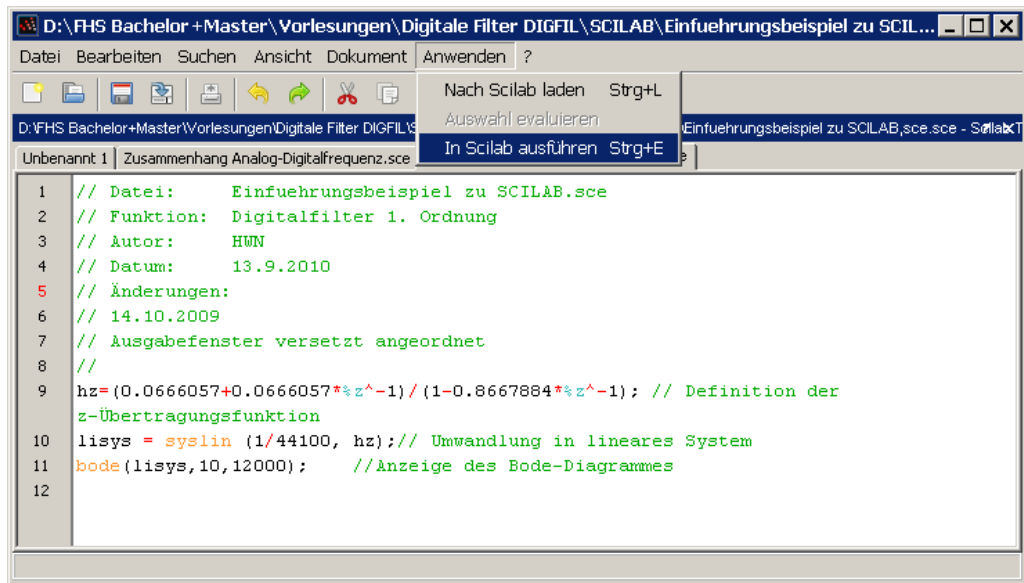


Abbildung 39: Ausführen der Befehle aus der Kommandodatei

5.3. Beispielberechnungen

Im vorigen Kapitel haben wir das Programm SCILAB schon zur Darstellung des Bodediagrammes eines IIR-Filters benutzt. Nun wollen wir uns ansehen, wie wir die Filterkoeffizienten von IIR-Filtern mit SCILAB berechnen können. Dies werden wir an einfachen Beispielen durchführen. Wir geben die Befehle zur Filterberechnung nicht direkt an der Kommandozeile von SCILAB ein, sondern starten den in SCILAB eingebauten Editor. Hier geben wir die Befehle ein und erzeugen dann eine SCE-Kommandodatei. Diese führen wir dann anschließend aus.

Beispiel 1:

Wir wollen einen Tschebyscheff-Tiefpass mit folgenden Kenndaten entwerfen:

- 3dB-Grenzfrequenz 440 Hz
- Welligkeit im Durchlassbereich 3dB
- Übergangsfrequenz in den Sperrbereich 800 Hz bei einer Dämpfung von 40dB
- Abtastfrequenz 48000 Hz

Die dazu nötigen Eingaben in SCILAB sehen wir in nachfolgendem Listing:



Dateiname: „IIR-Beispiel LP 440Hz.sce“

```

1 // Datei: IIR-Beispiel LP 440Hz.sce
2 // Autor: HWN
3 // Datum: 16.8.2010
4 // Entwurf eines Tschebyscheff Tiefpass Filters
5 // Dämpfung D[db] = 20*log(u) => D[db]/20 = D[db]*0,05 = log(u)
6 // Beidseitig "10 hoch" (10** in SCILAB) bilden ergibt: 10**(D[db]*0,05) = u = 0,707 bei -3dB
7 // deltap = 1.0 - 10.0**(-0.05*ap) = 0,0114
8 // deltas = 10.00**(-0.05*as) = 0,707
9 fa = 48000; // Abtastfrequenz in Hz
10 f1 = 440; // 3 db Grenzfrequenz in Hz => (Wert "ap" in Zeile 14)!
11 ap = 3; // Durchlassband Welligkeit in dB, "ap" muss immer kleiner als "as" sein !!!!!
12 f2 = 800; // Beginn Sperrbereich in Hz
13 as = 40; // Sperrbereich Dämpfung in dB bei f2
14 f3 = 450; // Nur bei Bandpass oder Bandsperre
15 f4 = 460; // Nur bei Bandpass oder Bandsperre
16 om=[f1*(2**pi)/fa,f2*(2**pi)/fa,f3*(2**pi)/fa,f4*(2**pi)/fa]; // Berechne relative normierte Frequenzen für SCILAB
17 deltap=(1.0-10.0**(-0.05*ap)); //Berechne Durchlassbereich für SCILAB
18 deltas=10.00**(-0.05*as); //Berechne Sperrbereich für SCILAB
19 ftype = 'lp'; // Filtertyp, möglich sind "'lp','hp','sb','bp'"
20 fopt = 'cheb1'; // Filteroptimierung, möglich sind "'butt','cheb1','cheb2','ellip'"
21 [cells, fact, zzeros, zpoles]=eqiir(ftype,fopt,om,deltap,deltas); //Berechne iir-matrix
22 hz=fact*poly(zzeros,'z')/poly(zpoles,'z') //Erzeuge iir-Gleichung
23 lisys = syslin(1/fa, hz); //Umrechnen in lineares System für bodeplot
24 bode(lisys,100,1000); //bodeplot

```

Abbildung 40: SCILAB-Kommandodatei für das 440Hz Tschebyscheff-Tiefpassfilter

Zur besseren Lesbarkeit sind die Zeilen 1-24 noch einmal separat aufgelistet:

```

1 // Datei: IIR-Beispiel LP 440Hz.sce
2 // Autor: HWN
3 // Datum: 16.8.2010
4 // Entwurf eines Tschebyscheff Tiefpass Filters
5 // Dämpfung D[db] = 20*log(u) => D[db]/20 = D[db]*0,05 = log(u)
6 // Beidseitig "10 hoch" (10** in SCILAB) bilden ergibt: 10**(D[db]*0,05) = u = 0,707 bei -3dB
7 // deltap = 1.0 - 10.0**(-0.05*ap) = 0,0114
8 // deltas = 10.00**(-0.05*as) = 0,707
9 fa = 48000; // Abtastfrequenz in Hz
10 f1 = 440; // 3 db Grenzfrequenz in Hz => (Wert "ap" in Zeile 11)!
11 ap = 3; // Durchlassband Welligkeit in dB, "ap" muss immer kleiner als "as" sein !!!!!
12 f2 = 800; // Beginn Sperrbereich in Hz
13 as = 40; // Sperrbereich Dämpfung in dB bei f2
14 f3 = 450; // Nur bei Bandpass oder Bandsperre
15 f4 = 460; // Nur bei Bandpass oder Bandsperre
16 om=[f1*(2**pi)/fa,f2*(2**pi)/fa,f3*(2**pi)/fa,f4*(2**pi)/fa]; //Berechne relative normierte
//Frequenzen für SCILAB
17 deltap=(1.0-10.0**(-0.05*ap)); //Berechne Durchlassbereich für SCILAB
18 deltas=10.00**(-0.05*as); //Berechne Sperrbereich für SCILAB
19 ftype = 'lp'; // Filtertyp, möglich sind "'lp','hp','sb','bp'"
20 fopt = 'cheb1'; // Filteroptimierung, möglich sind "'butt','cheb1','cheb2','ellip'"
21 [cells, fact, zzeros, zpoles]=eqiir(ftype,fopt,om,deltap,deltas); //Berechne iir-matrix
22 hz=fact*poly(zzeros,'z')/poly(zpoles,'z') //Erzeuge iir-Gleichung
23 lisys = syslin(1/fa, hz); //Umrechnen in lineares System für bodeplot
24 bode(lisys,100,1000); //bodeplot

```

Listing 2: SCILAB-Kommandodatei für das gesuchte Filter

Bedeutung der einzelnen Zeilen in der Kommandodatei laut Listing 2:

Zeilen 1 - 8: reiner Kommentar (beginnt mit //) => grün dargestellt

Zeile 9: Definition der Abtastfrequenz in Hz

Zeile 10: Definition der Grenzfrequenz 1 (Ende des Durchlassbereiches) in Hz. Hier hat die Dämpfung den Wert laut Zeile 11

Zeile 11: Definition der Welligkeit im Durchlassbereich in dB

- Zeile 12:** Definition der Grenzfrequenz 2 (Beginn des Sperrbereiches) in Hz. Hier hat die Dämpfung den Wert laut Zeile 13
- Zeile 13:** Definition der Dämpfung am Anfang des Sperrbereiches in dB
- Zeile 16:** Wert „om“ => Normalisierung der Grenzfrequenzen auf die Abtastfrequenz, die Abtastfrequenz entspricht in SCILAB immer 2π
- Zeile 17:** Wert „deltap“ => Umrechnung der Welligkeit von dB in lineare Darstellung
- Zeile 18:** Wert „deltas“ => Umrechnung der Stopbanddämpfung von dB in lineare Darstellung
- Zeile 19:** Definition des Filtertyps => hier Tiefpass => 'lp'
- Zeile 20:** Definition der Filterapproximation => hier Tschebyscheff 1 => 'cheb1' => Welligkeit nur im Durchlassbereich!
- Zeile 21:** Berechnung der Filterdaten

Übergabeparameter an die Filterberechnung:

om, deltap und deltas wurden bereits oberhalb erläutert und berechnet
 ftype => 'lp' definiert einen Tiefpass

möglich sind auch 'hp' = Hochpass, 'sb' = Bandsperre, 'bp' = Bandpass

fopt => 'cheb1' definiert die Approximation Tschebyscheff 1

möglich sind auch 'butt' = Butterworth, 'cheb2' = Tschebyscheff 2, 'ellip' = Elliptisch

Der Unterschied im Filtertyp 'cheb1' und 'cheb2' besteht in der Definition der Welligkeit im Durchlass- und Sperrbereich:

- Für 'cheb1' - Filter gilt: (1-ap) < Welligkeit < 1 im Durchlassbereich
- Für 'cheb2' - Filter gilt: 0 < Welligkeit < as im Sperrbereich
- Für elliptische Filter gilt: (1-ap) < Welligkeit < 1 im Durchlassbereich UND 0 < Welligkeit < as im Sperrbereich

Berechnete Werte:

- cells** = Darstellung der Übertragungsfunktion des gesuchten Filters als Kettenschaltung von Biquadfiltern (SOS = Second Order Sections)
- fact** = Verstärkungsfaktor des gesuchten Filters
- zzeros** = Nullstellen der Gesamtübertragungsfunktion des gesuchten Filters
- zpoles** = Pole der Gesamtübertragungsfunktion des gesuchten Filters

- Zeile 22:** Berechnung der Filterdatenübertragungsfunktion aus Polen (zpoles), Nullstellen (zzeros) und Verstärkungsfaktor (fact) des Filters
- Zeile 23:** Umrechnung der Übertragungsfunktion in ein lineares System zum Zeichnen des Bodediagrammes
- Zeile 24:** Zeichnen des Bodediagrammes zwischen den Grenzen 10 und 100Hz

Nach dem Start der Befehlsausführung im SCILAB-Editor erfolgt die Filterberechnung und der Bodeplot wird laut Abbildung 41 angezeigt.

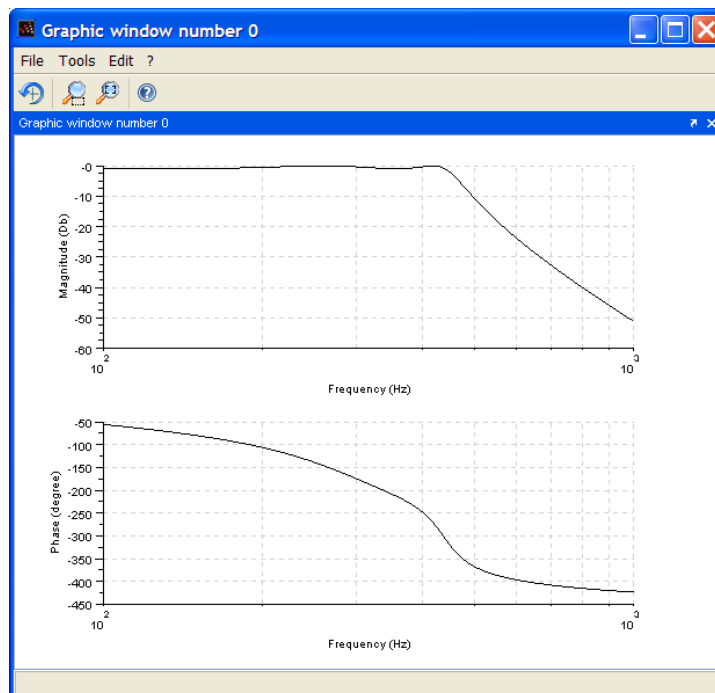


Abbildung 41: Bodediagramm des 440Hz Tschebyscheff-Tiefpasses

Die gesuchte Übertragungsfunktion kann man sich durch Eingabe des Namens der Übertragungsfunktion (hz) anzeigen lassen. Dies ist in Abbildung 42 dargestellt.

```

hz =
      2      3      4      5
  2.500D-09 + 1.250D-08z + 2.500D-08z + 2.500D-08z + 1.250D-08z + 2.500D-09z
  -----
      2      3      4      5
  - 0.9456673 + 4.7772133z - 9.6574534z + 9.7659297z - 4.9400223z + z
  
```

Abbildung 42: Übertragungsfunktion des 440Hz Tschebyscheff-Tiefpasses

Die positiven Potenzen von „z“ in Abbildung 42 kann man durch Division von Zähler und Nenner durch z^6 eliminieren. Dann kommt man auf die übliche Form mit z^{-n} . Wir sehen hier also ein Filter 5. Ordnung, welches aus 2 Biquadfiltern (Filter 2. Ordnung) und einem Filter 1. Ordnung aufgebaut wird.

Wenn man das Filter genau nach der Formel in Abbildung 42 realisiert, kommt es bei Festkomma-Signalprozessoren wegen der relativ ungenauen Zahlendarstellung zu Abweichungen vom gewünschten Übertragungsverhalten des Filters. Deshalb zerlegt man die Filterformel in eine Serienschaltung (Kettenschaltung) von Filterstufen 2. Ordnung. Diese ist besonders tolerant gegenüber ungenauer Zahlendarstellung im Festkommaformat.

Eine Darstellung als Kettenschaltung von Biquadfiltern (Filter 2. Ordnung) sehen wir in Abbildung 43. Dies wird durch Eingabe von „cells“ angezeigt.



Achtung

Bei der praktischen Realisierung des Filters mit einem Signalprozessor ist zu beachten, dass der Verstärkungsfaktor „fact“ (siehe Abbildung 45) noch als multiplikative Konstante bei der Kettenschaltung einzubauen ist!

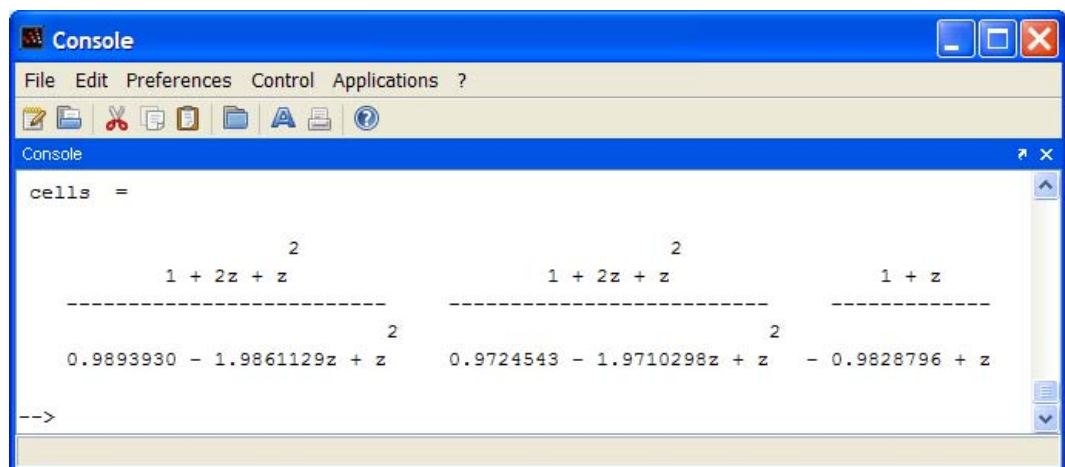


Abbildung 43: Biquad-Darstellung der Übertragungsfunktion des 440Hz -Tiefpasses

Wir erinnern uns an die z-Übertragungsfunktion eines Filters 2. Ordnung:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{A0 + A1 \cdot z^{-1} + A2 \cdot z^{-2}}{1 + B1 \cdot z^{-1} + B2 \cdot z^{-2}} \Rightarrow \text{Nun mit } \frac{z^2}{z^2} \text{ multiplizieren}$$

$$\Rightarrow \text{SCILAB - Form } \frac{A2 + A1 \cdot z + A0 \cdot z^2}{B2 + B1 \cdot z + z^2}$$

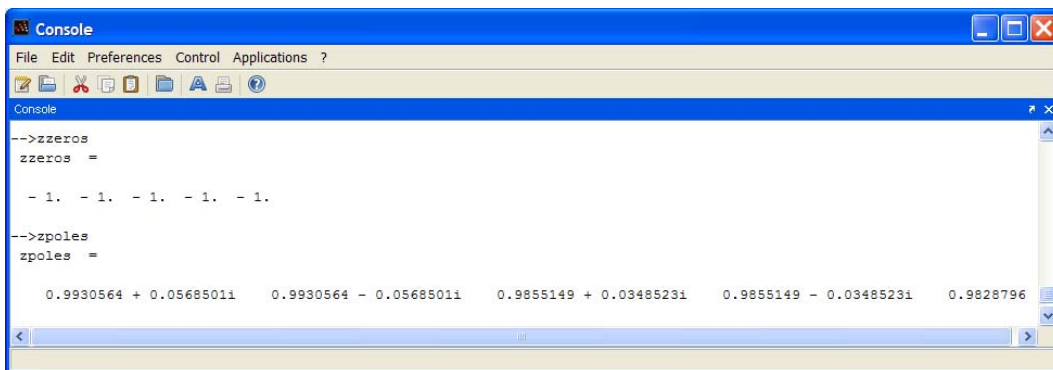
Die Qualität eines Filters kann man anhand der Filterkoeffizienten nicht abschätzen. Besser geeignet ist dazu die **Filtergüte Q**.

$$Q = \frac{\sqrt{(1+B2)^2 - B1^2}}{2 \cdot (1-B2)}$$

Gleichung 66: Polgüte eines Biquadfilter (Quelle: Literaturverzeichnis Nr. 2)

Sie gibt an, wie stark die Schwingneigung eines Filters ist. Eigenschwingungen des Filters können gedämpft (Filterausgangssignal schwingt nach endlicher Zeit aus) oder ungedämpft (Filter ist Oszillator) sein. Je höher die Güte ist, umso höher ist die Schwingneigung des Filters.

Eine Darstellung der Filterpole und Nullstellen sehen wir in Abbildung 44. Dies wird durch Eingabe von „zpoles“ und „zzeros“ angezeigt. .



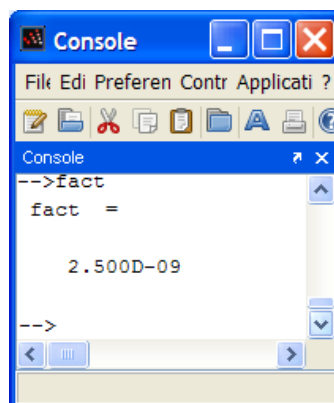
```

Console
File Edit Preferences Control Applications ?
Console
-->zzeros
zzeros =
- 1. - 1. - 1. - 1. - 1.
-->zpoles
zpoles =
0.9930564 + 0.0568501i 0.9930564 - 0.0568501i 0.9855149 + 0.0348523i 0.9855149 - 0.0348523i 0.9828796

```

Abbildung 44: Pole und Nullstellen der Übertragungsfunktion des 440 Hz Tiefpasses

Eine Darstellung der Filterverstärkung sehen wir Abbildung 45. Dies wird durch Eingabe von „fact“ angezeigt. .



```

Console
File Edit Preferences Control Applications ?
Console
-->fact
fact =
2.500D-09
-->

```

Abbildung 45: Filterverstärkung der Übertragungsfunktion des 440 Hz Tiefpasses

„fact“ muss bei der Realisierung als Kettenschaltung von Biquadfiltern als multiplikative Konstante in den Signalweg geschaltet werden. Da reale Werte von „fact“ sehr klein sind, teilt man „fact“ besser auf die einzelnen

Biquadfilter auf und rechnet den Wert direkt in die Zählerfilterkoeffizienten jedes einzelnen Biquadfilters ein. Durch diese Gleichverteilung auf alle Biquadfilter reduziert man auch die Gefahr der Übersteuerung einzelner Biquadfilter.

Beispiel:

Beim obigen Filterbeispiel mit 3 Biquad-Stufen ergibt sich z.B. ein Wert für „fact“ von $2.5 \cdot 10^{-9}$.

Bildet man die dritte Wurzel (wegen 3 Biquad-Stufen) aus diesem Wert, so kommt man auf den Wert 0.0001357208 für den Verstärkungsfaktor jeder einzelnen Biquad-Stufe.

Mit dieser Zahl wird nun in SCILAB jeder Zählerkoeffizient der Biquadfilter vor dem Speichern in die Koeffizientendatei für den Signalprozessor multipliziert. Somit muss man im DSP-Programm keine Multiplikation mit dem Filterverstärkungsfaktor durchführen.

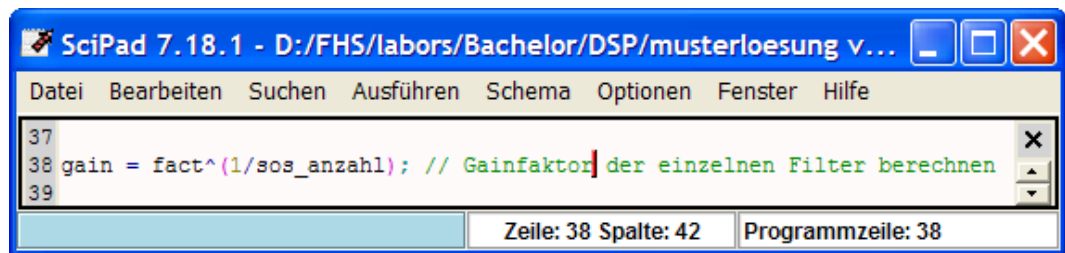


Abbildung 46: Berechnung der Einzelfilterverstärkung aus dem Wert „fact“

Wenn wir das Filter mit einem Signalprozessor realisieren wollen, müssen die Filterkoeffizienten in einer Datei stehen, welche in die Entwicklungssoftware des Signalprozessors eingebunden werden kann.

Im Labor ist dies „VISUAL DSP“ und die Filterkoeffizientendatei muss die Dateiendung „.dat“ besitzen.

In unserem Beispiel soll die Datei den Namen „IIR_LP_440HZ.dat“ besitzen.

Dazu sind die Zeilen ab Nummer 26 in der Datei nötig. Da die Datei zu groß für ein Bild ist, schauen wir sie uns in Teilen an.

```

26 // Nun die Filterkoeffizienten in Zaehler und Nenner der Biquad-Stufen auftrennen
27 nenner = coeff(denom(cells));
28 zaehler = coeff( numer(cells));
29
30 an = size(nenner); // Anzahl der Nennerkoeffizienten berechnen an(2) = Anzahl der Koeffizienten
31 az = size(zaehler); // Anzahl der Zählerkoeffizienten berechnen zn(2) = Anzahl der Koeffizienten
32 // nenner = b21, b22, b23,.. ,b2n, b11, b12, b13,.. ,b1n, b01, b02, b03,.. , b0n      bei "n" SOS Stufen
33 // zaehler = a21, a22, a23,.. ,a2n, a11, a12, a13,.. ,a1n, a01, a02, a03,.. , a0n      bei "n" SOS Stufen
34 // b0 sind immer 1!
35
36 sos_anzahl = an(2)/3; // Berechne Anzahl der Biquadfilter, je Biquad (SOS) gibt es 3 Koeffizienten für Zaehler + Nenner!
37
38 gain = fact^(1/sos_anzahl); // Gainfaktor der einzelnen Filter berechnen
39

```

Abbildung 47: Zeilen 26 bis 38 der Kommandodatei „IIR-Beispiel LP 440Hz.sce“

- Zeile 27-28:** Berechnung der Filterkoeffizienten von Zähler und Nenner der Biquadfilter (aus „cells“). Wir bauen den Gesamtfiter aus einer Serienschaltung mehrerer Filter 2. Ordnung (Biquadfilter, auch SOS = Second Order Sections genannt) auf.
- Zeile 30-36:** Berechnung der Anzahl der Biquadfilter. Diese ergibt sich aus dem Inhalt des zweiten Gliedes des Vektors „an“ (an(2) =Nenner 2. Glied) geteilt durch 3 (Anzahl der Nennerkoeffizienten).
- Zeile 38:** Berechnung des Verstärkungsfaktors jedes einzelnen Biquadfilters.

```

40 // Nun die Filterkoeffizienten in eine Datei für Visual DSP speichern
41 fd = mopen('d:/SCILAB/DIGFIL/IIR_LP_440HZ.DAT','wt'); // Zieldatei öffnen
42
43 // Reihenfolge der Koeffizienten in der Datei
44 // a01 = a0 erstes Biquad
45 // a11 = a1 erstes Biquad
46 // a21 = a2 erstes Biquad
47 // b11 = b1 erstes Biquad
48 //...
49 // a0n = a0 letztes Biquad
50 // a1n = a1 letztes Biquad
51 // a2n = a2 letztes Biquad
52 // b1n = b1 letztes Biquad
53 // b2n = b2 letztes Biquad

```

Abbildung 48: Zeilen 40 bis 53 der Kommandodatei „IIR-Beispiel LP 440Hz.sce“

- Zeile 41:** Hier wird die Datei, welche die Filterkoeffizienten enthalten soll, zum Schreiben geöffnet. Ist die Datei noch nicht vorhanden, wird sie neu erstellt.

Eine bereits vorhandene Datei wird überschrieben.

Zeile 43-53: Hier ist der Aufbau der Koeffizientendatei als Kommentarfeld dargestellt.

```

55 // Dateikommentarkopf schreiben
56 fprintf(fd,'%s\n',"// IIR-Filterentwurf ");
57 fprintf(fd,'%s\n',"=====");
58 fprintf(fd,'%s\n',"// HWN");
59 fprintf(fd,'%s\n',"// Datei: IIR-Beispiel LP 440Hz.sce");
60 fprintf(fd,'%s\n'," ",date());
61 fprintf(fd,'%s\n',"// Filtertyp = ",ftype);
62 fprintf(fd,'%d%s\n',"// Abtastfrequenz = ",fa," Hz");
63 fprintf(fd,'%d%s\n',"// 3dB-Grenzfrequenz 1 = ",f1," Hz");
64 fprintf(fd,'%d%s\n',"// Beginn Sperrbereich = ",f2," Hz");
65 fprintf(fd,'%f10\n',"// Gain Einzelfilter = ",gain);
66 fprintf(fd,'%d\n',"// Biquadanzahl = ",sos_anzahl);
67 fprintf(fd,'%s\n',"=====");
68
69 // Koeffizienten als mehrere Biquadfilter in richtiger Reihenfolge für DSP-Algorithmus in Datei speichern
70 // Anzahl der Koeffizienten steht in an(2) oder az(2) => sollten immer gleich sein => Anzahl der SOS * 3!
71 for i=1:sos_anzahl,
72 fprintf(fd,'%s\n',"=====");
73 fprintf(fd,'%d\n',"// Biquad = ",i);
74 fprintf(fd,'%s\n',"=====");
75 // Speichere a0
76 fprintf(fd,'%10.10f%s\n',zaehler(i+(sos_anzahl*2))*gain," //a0");
77 // Speichere a1
78 fprintf(fd,'%10.10f%s\n',zaehler(i+sos_anzahl)*gain," //a1");
79 // Speichere a2
80 fprintf(fd,'%10.10f%s\n',zaehler(i)*gain," //a2");
81 // Nun prüfe ich, ob das letzte Biquad nur 1. Ordnung hat => erkennbar an a0 = 0!
82 // Dann muss man die Vorzeichen von b1,2 des letzten Biquad umdrehen! Warum keine Ahnung!
83 if i == sos_anzahl then // Letztes Biquad?
84 if zaehler(i+(sos_anzahl*2)) == 0 then //a0 = 0?
85 // Speichere -b1 weil letztes Filter = 1. Ordnung!!!
86 fprintf(fd,'%10.10f%s\n',-nenner(i+sos_anzahl)," //-b1");
87 // Speichere -b2 weil letztes Filter = 1. Ordnung!!!
88 fprintf(fd,'%10.10f%s\n',-nenner(i)," //-b2");
89 else
90 // Speichere b1 bei Biquad mit Ordnung 2
91 fprintf(fd,'%10.10f%s\n',nenner(i+sos_anzahl)," //b1");
92 // Speichere b2 bei Biquad mit Ordnung 2
93 fprintf(fd,'%10.10f%s\n',nenner(i)," //b2");
94 end
95 else
96 // Speichere b1 bei Biquad mit Ordnung 2
97 fprintf(fd,'%10.10f%s\n',nenner(i+sos_anzahl)," //b1");
98 // Speichere b2 bei Biquad mit Ordnung 2
99 fprintf(fd,'%10.10f%s\n',nenner(i)," //b2");
100
101 end; // Ende if i == sos_anzahl
102
103 end; // Ende for
104 fclose(fd);

```

Abbildung 49: Zeilen 55 bis 104 der Kommandodatei „IIR-Beispiel LP 440Hz.sce“

Zeile 55-67: an den Anfang der Datei wird ein Kommentarkopf geschrieben

Zeile 71-103: in einer for-Schleife, welche so oft wie die Anzahl der Biquad-Stufen durchlaufen wird (sos_anzahl), werden die Filterkoeffizienten in die Datei geschrieben



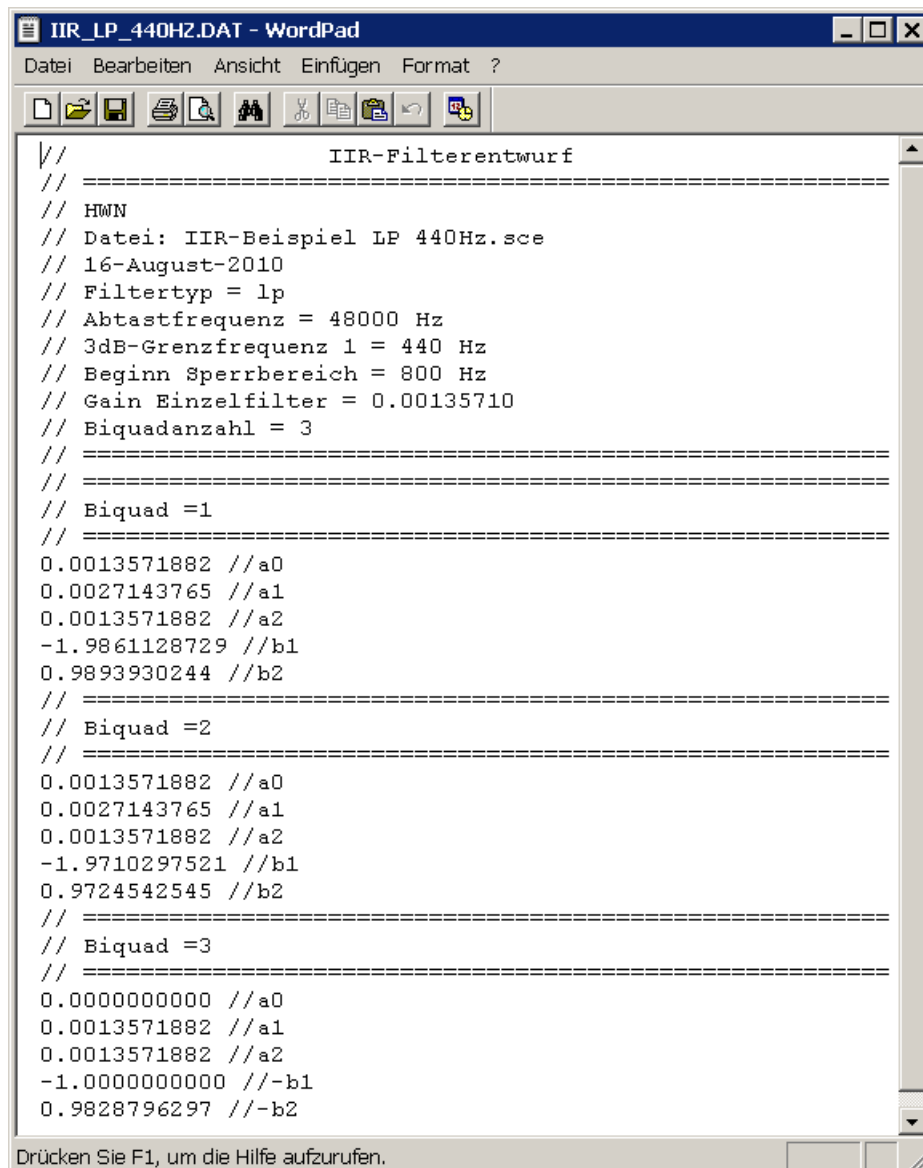
Achtung

Zu beachten ist, dass die Koeffizienten b1 und b2 der letzten Biquadstufe mit -1 multipliziert werden müssen, falls das letzte Biquadfilter nur 1. Ordnung hat (erkennbar an $a_0 = 0$). Warum dies so ist, ist dem Autor unbekannt. Es wurde

experimentell ermittelt! Im Kommentar hinter dem Koeffizienten ist dies in der erzeugten Koeffizientendatei durch „-b1“ und „-b2“ gekennzeichnet!

Zeile 104: Datei wird geschlossen

Die resultierende Koeffizientendatei können wir mit einem einfachen Texteditor laden (hier: WordPad), sie sieht dann folgendermaßen aus:



```
// IIR-Filterentwurf
// =====
// HWN
// Datei: IIR-Beispiel LP 440Hz.sce
// 16-August-2010
// Filtertyp = lp
// Abtastfrequenz = 48000 Hz
// 3dB-Grenzfrequenz 1 = 440 Hz
// Beginn Sperrbereich = 800 Hz
// Gain Einzelfilter = 0.00135710
// Biquadanzahl = 3
// =====
// Biquad =1
// =====
0.0013571882 //a0
0.0027143765 //a1
0.0013571882 //a2
-1.9861128729 //b1
0.9893930244 //b2
// =====
// Biquad =2
// =====
0.0013571882 //a0
0.0027143765 //a1
0.0013571882 //a2
-1.9710297521 //b1
0.9724542545 //b2
// =====
// Biquad =3
// =====
0.0000000000 //a0
0.0013571882 //a1
0.0013571882 //a2
-1.0000000000 //-b1
0.9828796297 //-b2
```

Abbildung 50: Inhalt der erzeugten Koeffizientendatei

Man erkennt an Biquad3, dass es nur ein Filter 1. Ordnung ist (erster Koeffizient $a_0 = 0$!

Beispiel 2:

Wir wollen einen Tschebyscheff-Bandpass mit folgenden Kenndaten entwerfen:

- 0,1 dB-Welligkeit im Durchlassbereich von 400 Hz bis 440 Hz
- Dämpfung von 3dB bei 360 Hz und bei 480 Hz
- Abtastfrequenz 48000 Hz

**Dateiname „IIR-Beispiel BP 440Hz.sce“**

Die dazu nötigen Eingaben in SCILAB sehen wir in Abbildung 51.

```

1 // Datei: IIR-Beispiel BP 440Hz.sce
2 // Autor: HWN
3 // Datum: 16.8.2010
4 // Entwurf eines Tschebyscheff Bandpass Filters
5 // Dämpfung D[db] = 20*log(u) => D[db]/20 = D[db]*0,05 = log(u)
6 // Beidseitig "10 hoch" (10** in SCILAB) bilden ergibt: 10**(D[db]*0,05) = u = 0,707 bei -3dB
7 // deltap = 1.0 - 10.0**(-0.05*ap) = 0,0114
8 // deltas = 10.00**(-0.05*as) = 0,707
9 fa = 48000; // Abtastfrequenz in Hz
10 f1 = 360; // Beginn Sperrbereich (Wert "ap" in Zeile 11)!
11 ap = 0.1; // Durchlassband Welligkeit in dB, "ap" muss immer kleiner als "as" sein !!!!!
12 f2 = 400; // Beginn Durchlassbereich in Hz
13 as = 3; // Sperrbereich Dämpfung in dB bei f1,f4
14 f3 = 440; // Ende Durchlassbereich in Hz
15 f4 = 480; // Beginn Sperrbereich
16 om=[f1*(2*pi)/fa,f2*(2*pi)/fa,f3*(2*pi)/fa,f4*(2*pi)/fa]; // Berechne relative normierte
// Frequenzen für SCILAB
17 deltap=(1.0-10.0**(-0.05*ap)); //Berechne Durchlassbereich für SCILAB
18 deltas=10.00**(-0.05*as); //Berechne Sperrbereich für SCILAB
19 ftype = 'bp'; // Filtertyp, möglich sind 'lp','hp','sb','bp'
20 foft = 'cheb1'; // Filteroptimierung, möglich sind 'butt','cheb1','cheb2','ellip'
21 [cells, fact, zzeros, zpoles]=eqiir(ftype,foft,om,deltap,deltas); //Berechne iir-matrix
22 hz=fact*poly(zzeros,'z')/poly(zpoles,'z') //Erzeuge iir-Gleichung
23 lsys = syslin(1/fa, hz); //Umrechnen in lineares System für bodeplot
24 bode(lsys,100,10000); //bodeplotbode(lsys,100,10000); //bodeplot
25

```

Abbildung 51: SCILAB-Kommandodatei für das gesuchte Filter

Nach Start der Filterberechnung im SCILAB-Editor erfolgt die Filterberechnung und der Bodeplot laut Abbildung 52 wird angezeigt.

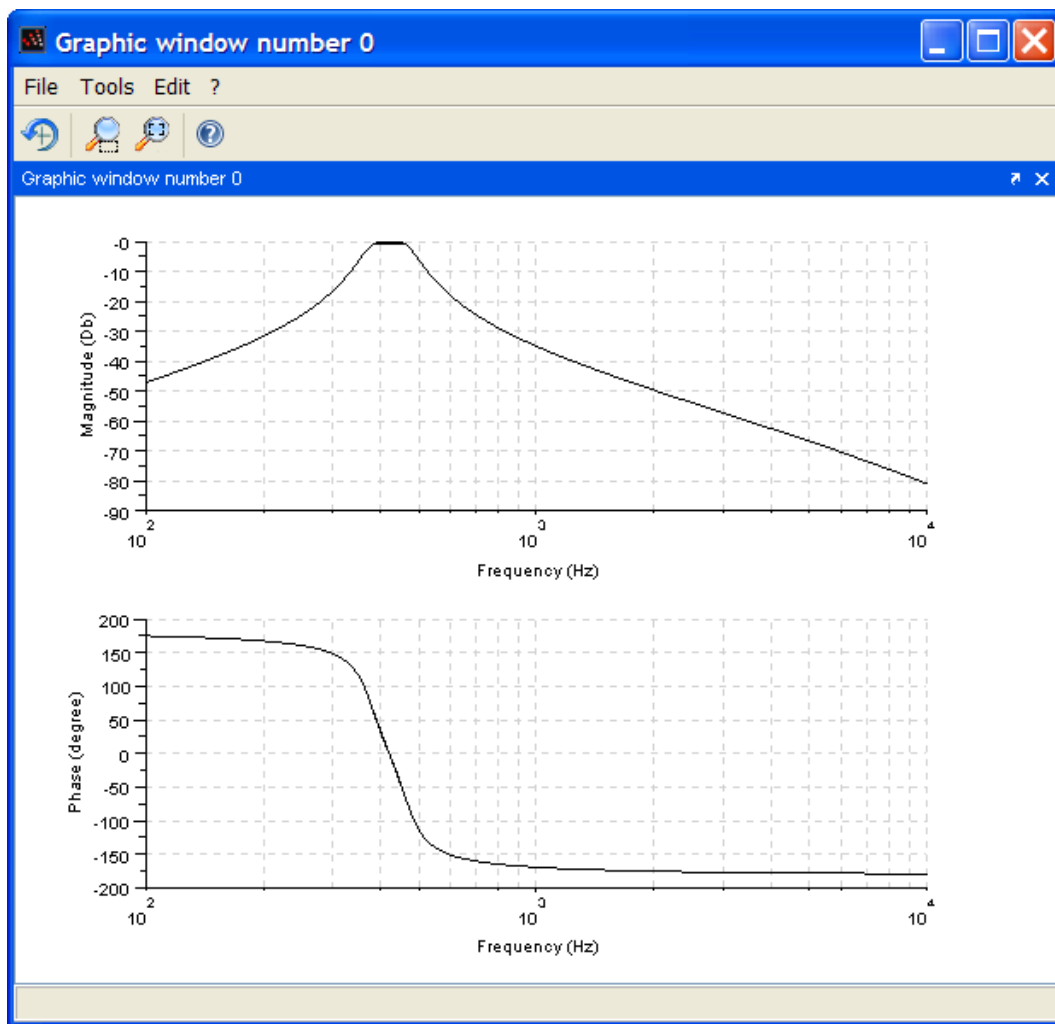


Abbildung 52: Bodediagramm des 440Hz-Bandpasses

Alle weiteren berechneten Werte („cells“ usw.) bitte selbst nachprüfen! Auch hier wird eine Koeffizientendatei erzeugt.

6. Entwurf nichtrekursiver Filter (FIR-Filter) mit SCILAB

6.1. Einleitung

SCILAB beinhaltet zur Berechnung von FIR-Filtern die folgenden Funktionen:

Funktionsname	Anwendung
<u>eqfir</u>	minimax approximation of FIR filter
<u>ffilt</u>	coefficients of FIR low-pass
<u>frmag</u>	magnitude of FIR and IIR filters
<u>fsfirlin</u>	design of FIR, linear phase filters, frequency sampling technique
<u>hilb</u>	FIR approximation to a Hilbert transform filter
<u>wfir</u>	linear-phase FIR filters
<u>remezb</u>	Minimax approximation of magnitude response

Tabelle 2: SCILAB-Funktionen mit Anwendung aus der Online-Hilfe

Alle genannten Funktionen können wir hier aus Platzgründen nicht beschreiben.

Wir benutzen im folgenden Beispiel die Funktion „wfir“, welche die Filterkoeffizienten nach der Fenstermethode berechnet.

Wfir
interaktiv
nutzen

Man kann die Funktion „wfir“ auch interaktiv aufrufen. Nach Start von SCILAB und Eingabe von „wfir“ wird man wie folgt begrüßt:

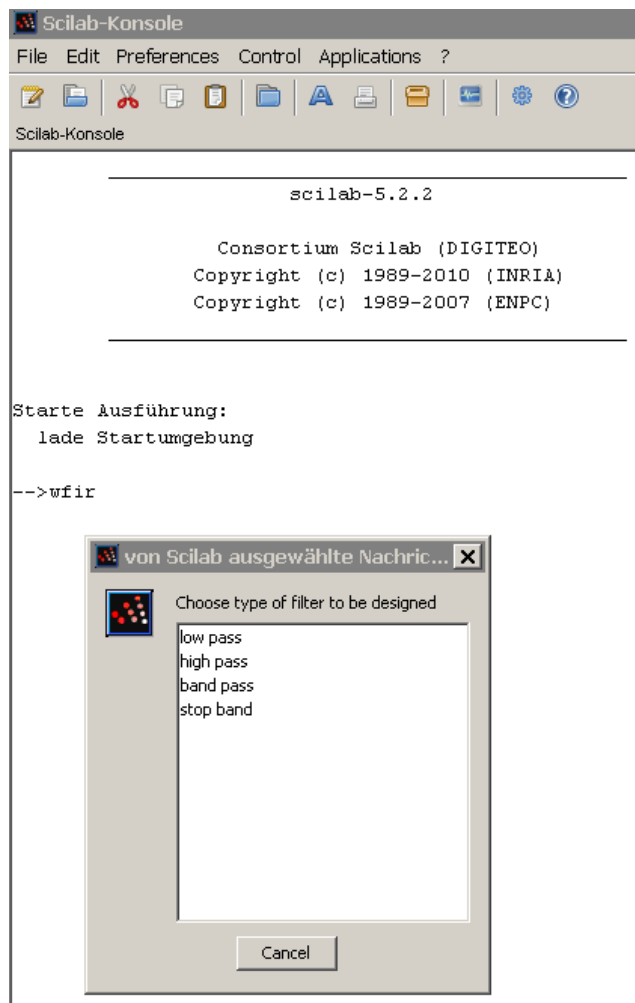


Abbildung 53: Erstes Menü nach interaktivem Aufruf von „wfir“

Nun wählt man mit der Maus durch Doppelklick auf den gewünschten Filtertyp den Filtertyp aus und dann geht es weiter mit neuen Menüs, bis man am Ende eine Liste mit den berechneten Filterkoeffizienten erhält.

Eingegebene Filterdaten:

lowpass
 cut-off frequency 0.1
 Filter length: 10
 Input window type: Hamming

```
-->wfir
ans =
    0.0017487    0.0138044    0.0585845    0.1321926    0.1912689    0.1912689    0.1321926    0.0585845    0.0138044    0.0017487
```

Abbildung 54: Von „wfir“ berechnete Filterkoeffizienten

Beispiel mit „ffilt“:

```

-->ffilt('lp',10,0.1)
ans =
    0.0218585    0.0735766    0.1273240    0.1716787    0.1967263    0.1967263    0.1716787    0.1273240    0.0735766    0.0218585

```

Dies ist das gleiche Ergebnis wie mit „wfir“, wenn man dort als Fenster „Rectangular“ eingibt.

6.2. Tiefpass-Filterbeispiel

Nun wollen wir als Beispiel einen FIR-Tiefpass mit der Grenzfrequenz von 700Hz berechnen. Die nötigen Kommandos für SCILAB sind in Abbildung 55 dargestellt. Hier schreiben wir die berechneten Filterkoeffizienten in eine Textdatei, welche wir direkt mit unserer Signalprozessor-Entwicklungssoftware „Visual DSP“ benutzen können.



Dateiname „FIR-Beispiel LP 700Hz.sce“

```

D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\FIR-Beispiel LP 700Hz\FIR-Beispiel LP 700Hz.sce - Scilab Text Editor
Datei Bearbeiten Suchen Ansicht Dokument Anwenden ?
D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\FIR-Beispiel LP 700Hz\FIR-Beispiel LP 700Hz.sce - Scilab Text Editor
Unbenannt 1 | IIR-Beispiel BP 440Hz.sce | FIR-Beispiel LP 700Hz.sce
1 // Datei: FIR-Beispiel LP 700Hz.sce
2 // HWN 16.10.2009
3 // Entwurf eines FIR Tiefpass-Filters
4 // Achtung: Funktioniert nicht mit beliebigen Filterordnungen!!
5 // Filterkoeffizienten werden in Datei "FIR_LP_700HZ.DAT" im Ordner "d:/FHS/SCILAB/FILES/DSSP/" gespeichert
6 // Achtung: Pfad für Koeffizientendatei nu beliebig, aber vorher "Options-Filenames-Full path" setzen im Editor!
7 //-----
8 fa=48000; // Abtastfrequenz in Hz
9 f1=700; // untere 3dB-Grenzfrequenz f1 = 700Hz
10 f2=900; // obere 3dB-Grenzfrequenz f2 = nur sinnvoll bei "bp" oder 'sb', beliebig bei Tief - oder Hochpass
11 ftype = 'lp'; // Filtertyp Tiefpass, möglich 'hp'= Hochpass,'bp'= Bandpass,'sb' = Bandsperre
12 forder = 249; // Filterordnung, muss ungerade sein für 'hp' oder 'sb'
13 cf1 = f1/fa; // Normierte Grenzfrequenz f1 berechnen. Anders als beim IIR-Filter ohne "*2*pi" !!!!!!!!
14 cf2 = f2/fa; // Normierte Grenzfrequenz f2 berechnen. Anders als beim IIR-Filter ohne "*2*pi" !!!!!!!!
15 cfreq = [cf1 cf2]; // Wert zur Übergabe an "wfir" formatieren
16 wtype = 'hm'; // Hamming-Window
17 fpar = [0 0]; // Nicht benutzt wegen 'Hamming' Window
18
19 [wft,wfm,fr]=wfir(ftype,forder,cfreq,wtype,fpar); // Aufruf der Filterberechnungsfunktion
20 // Rückgabe von wft = Filterkoeffizienten; wfm = Amplitudenwerte der Filterübertragungsfunktion bei "fr"-Frequenzen
21
22 z=poly(0,'z'); // Leeres Polynom mit "z" als Variable definieren
23
24 h=poly(wft,'z','c')/z**(forder-1); //FIR-z-Übertragungsfunktion erzeugen
25
26 hz=syslin(1/fa,h); // Übertragungsfunktion in zeitkontinuierliches System umwandeln für das Bodediagramm!
27
28 bode(hz,1,10000); // Zeichnen des Bodediagrammes
29

```

Abbildung 55: SCILAB-Kommandodatei Teil 1 für das gesuchte FIR-Filter

Zur Erklärung der Kommandos aus Abbildung 55 + Abbildung 56 finden Sie in den entsprechenden Zeilen ausführliche Kommentare.



Achtung

Wichtig ist es, dass man bei der Frequenznormierung im Gegensatz zum IIR-Filter den multiplikativen Faktor „2*%pi“ weglässt.

```

D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\FIR-Beispiel LP 700Hz\FIR-Beispiel LP 700Hz.sce - ...
Datei Bearbeiten Suchen Ansicht Dokument Anwenden ?
D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\FIR-Beispiel LP 700Hz\FIR-Beispiel LP 700Hz.sce - Scilab Text Editor
Unbenannt 1 | IIR-Beispiel BP 440Hz.sce | FIR-Beispiel LP 700Hz.sce
30 // Nun Filterkoeffizienten (stehen in Rückgabewert "wft" von wfir()) in Koeffizientendatei schreiben
31 fd = mopen('D:/SCILAB/DIGFIL/FIR_LP_700HZ.DAT','wt'); // Zieldatei öffnen
32
33 // Dateikommentarkopf schreiben
34 mfprintf(fd,'%s\n',"//      FIR-Filterentwurf      ");
35 mfprintf(fd,'%s\n',"// -----");
36 mfprintf(fd,'%s\n',"// HWN");
37 mfprintf(fd,'%s\n',"// Datei: FIR-Beispiel LP 700Hz.sce");
38 mfprintf(fd,'%s\n',"// ",date());
39 mfprintf(fd,'%s\n',"// Filtertyp = ",ftype);
40 mfprintf(fd,'%s\n',"// Abtastfrequenz = ",fa," Hz");
41 mfprintf(fd,'%s\n',"// 3dB-Grenzfrequenz 1 = ",f1," Hz");
42 mfprintf(fd,'%s\n',"// 3dB-Grenzfrequenz 2 = ",f2," Hz, nur bei Bandpass und Bandsperre
gültig!");
43 mfprintf(fd,'%s\n',"// Ordnung = ",forder);
44 mfprintf(fd,'%s\n',"// Fenstertyp = ",wtype);
45 mfprintf(fd,'%s\n',"// -----");
46
47 // Koeffizienten schreiben
48
49 for i=1:forder,
50 // mfprintf(fd,'hello %s %d.\n','world',i);
51 // mfprintf(fd,'%f\n',wft(:,i));
52 end;
53 mclose(fd);

```

Abbildung 56: SCILAB-Kommandodatei Teil 2 für das gesuchte FIR-Filter

In Abbildung 56 sehen wir den Programmteil, welcher die Koeffizientendatei erzeugt.

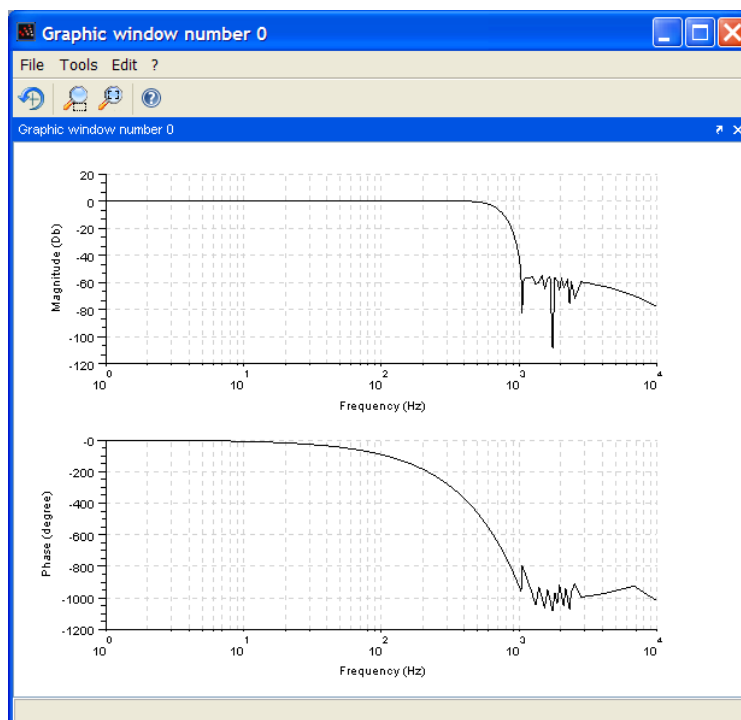
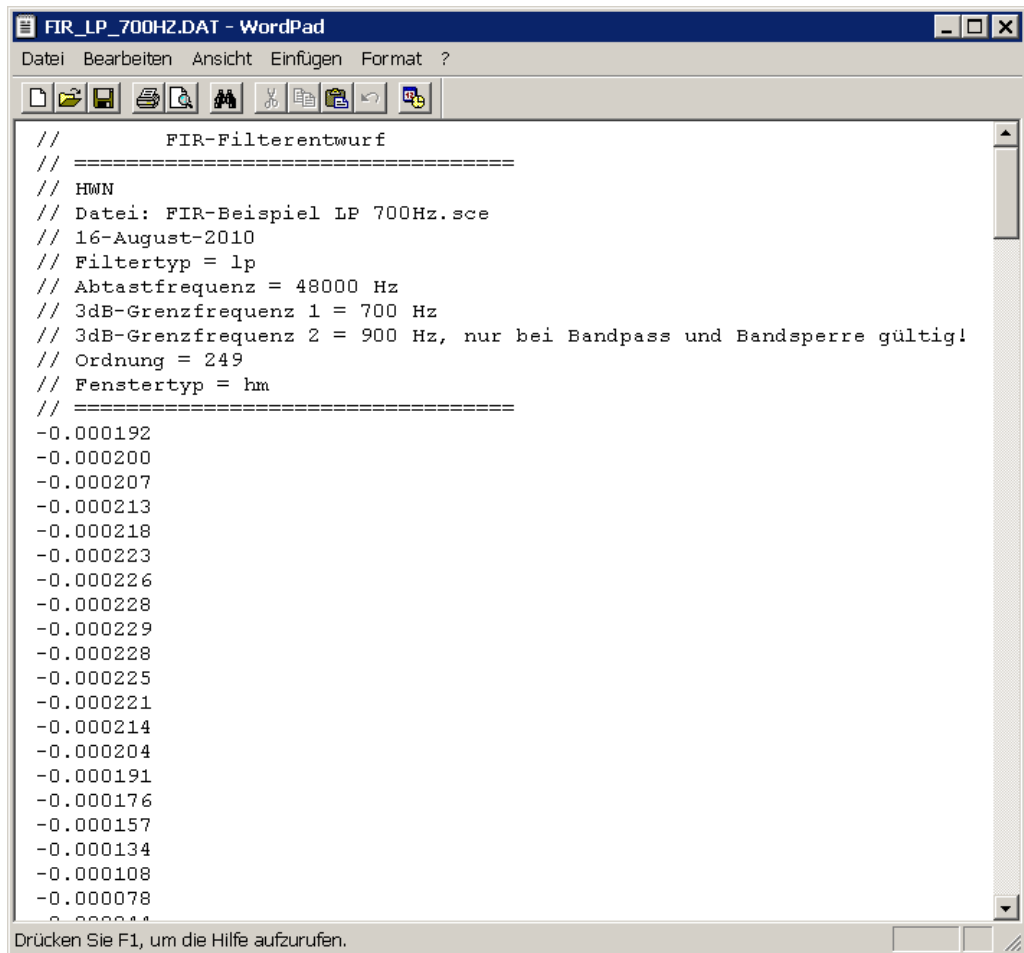


Abbildung 57: Bodediagramm des gesuchten FIR-Filters



```
//          FIR-Filterentwurf
// =====
// HWN
// Datei: FIR-Beispiel LP 700Hz.sce
// 16-August-2010
// Filtertyp = lp
// Abtastfrequenz = 48000 Hz
// 3dB-Grenzfrequenz 1 = 700 Hz
// 3dB-Grenzfrequenz 2 = 900 Hz, nur bei Bandpass und Bandsperre gültig!
// Ordnung = 249
// Fenster typ = hm
// =====
-0.000192
-0.000200
-0.000207
-0.000213
-0.000218
-0.000223
-0.000226
-0.000228
-0.000229
-0.000228
-0.000225
-0.000221
-0.000214
-0.000204
-0.000191
-0.000176
-0.000157
-0.000134
-0.000108
-0.000078
0.000048
```

Abbildung 58: Auszug aus der erzeugten Koeffizientendatei des gesuchten FIR-Filters

Die Koeffizientendatei, welche in Abbildung 58 dargestellt ist, kann direkt in die Programme üblicher Signalprozessorsoftware eingebunden werden. Wir machen dies im Labor für Signalprozessoren mit dem Signalprozessor ADSP-21262. Dies ist ein 32-Bit Fließkomma-DSP.

7. Grafische Darstellung von Filteralgorithmen

7.1. Einleitung

Digitale Filter werden beispielsweise entsprechend den vorgenannten Verfahren oder mittels anderer spezieller Rechnerprogramme entworfen. Die dadurch ermittelten Koeffizienten müssen nun in einen Algorithmus für einen Signalprozessor umgesetzt werden. Dabei zeichnet man den Algorithmus bei einfachen Filtern in einer Art Blockschaltbild wie in Abbildung 59 dargestellt. In dem gezeichneten Algorithmus kommen die Operationen „Multiplikation“, „Addition“ sowie Zeitverzögerung um einen Abtasttakt (Block z^{-1}) vor.

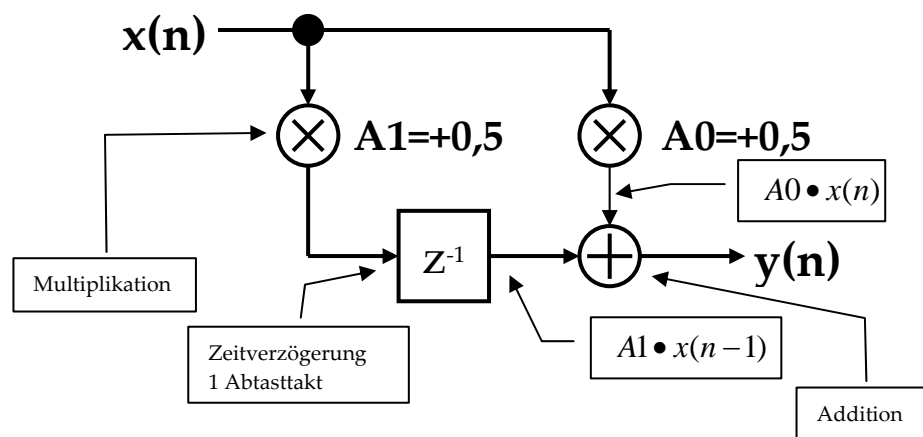


Abbildung 59: Blockschaltbild eines einfachen nichtrekursiven Filteralgorithmus

In diesem Bild ist $x(n)$ der Filtereingang zum aktuellen Zeitpunkt (in diesem Moment). Dieses Signal wird mit der Filterkonstanten A_0 (+0,5) multipliziert und zu dem mit dem Filterkoeffizienten A_1 (+0,5) multiplizierten Eingangssignal $x(n-1)$ addiert. $x(n)$ wird durch den Block mit z^{-1} um einen Abtasttakt verzögert \Rightarrow dies ergibt also $x(n-1)$. Als Summe entsteht das Ausgangssignal $y(n)$.

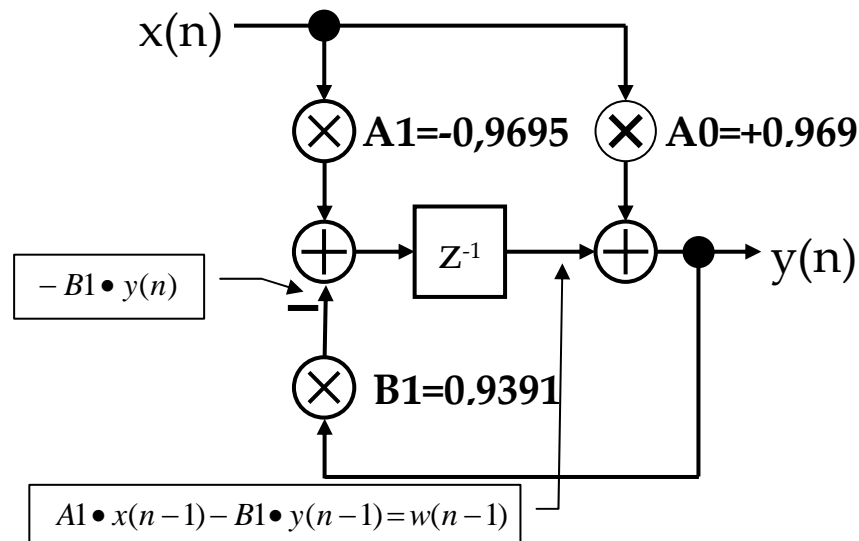


Abbildung 60: Blockschaltbild eines einfachen rekursiven Filteralgorithmus

In Abbildung 60 sehen wir einen IIR-Hochpass erster Ordnung mit folgenden Daten:

$$H(z) = \frac{0,9695 - 0,9695 \cdot z^{-1}}{1 - 0,9351 \cdot z^{-1}} = \frac{y(z)}{x(z)}$$

$$f'_g = 0,01$$

Die Eingabe der entsprechenden Filterdaten in SCILAB sehen wir in Abbildung 61.

```

1 // Datei: IIR-Hochpass 1_Ordnung.sce
2 // Frequenzgang Hochpass 1. Ordnung
3 // Eingabe der z-Übertragungsfunktion
4 // HWN
5 // 13.9.2010
6 fa = 44100; // Abtastfrequenz
7
8 // z-Übertragungsfunktion definieren
9 hz = (0.9695 - 0.9695*z^-1) / (1 - 0.9351*z^-1);
10
11 lisy = syslin (1/fa, hz); // Berechnung des linearen zeitkontinuierlichen Systems aus hz
12
13 // Anzeige Bodediagramm
14 bode(lisy, 10, 12000);
15
    
```

Abbildung 61: SCILAB-Kommandos des Hochpasses 1. Ordnung



[Dateiname „IIR-Hochpass 1_Ordnung.sce“](#)

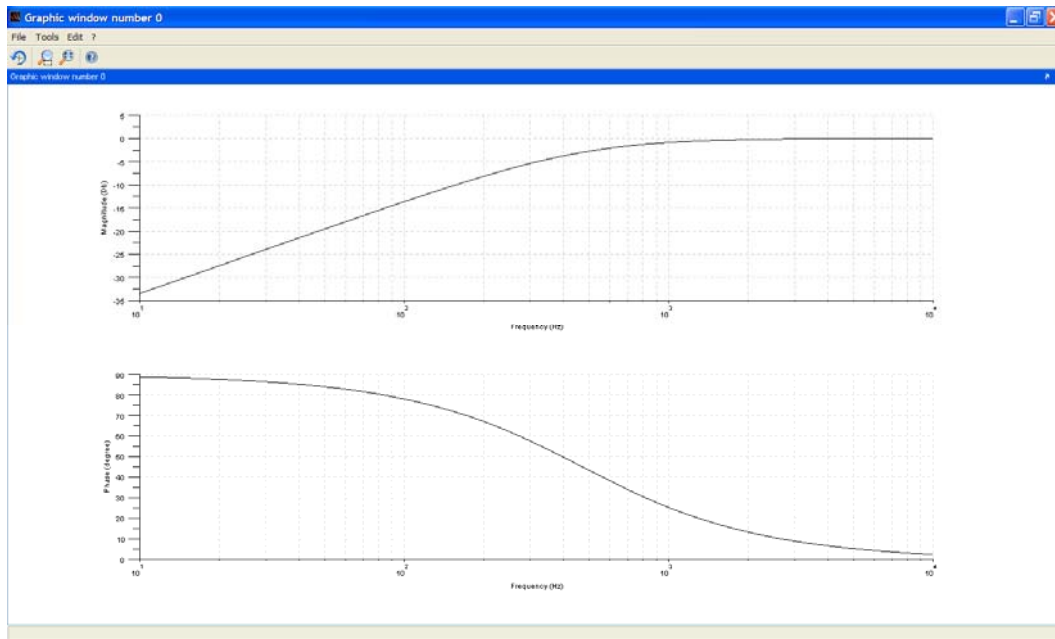


Abbildung 62: Bodediagramm des Hochpasses 1. Ordnung

Aus Abbildung 60 kann man direkt den Algorithmus des DSP ablesen, wenn man mittels inverser z-Transformation ($x(n) \cdot z^{-1} \rightarrow x(n-1)$) die Differenzgleichung bildet:

$$y(n) = A0 \cdot x(n) + A1 \cdot x(n-1) - B1 \cdot y(n-1)$$

Hier sehen wir eine Rückkopplung des Ausgangssignales $y(n) \Rightarrow$ Rekursion, welches dann über den Block mit z^{-1} um einen Abtasttakt verzögert am Ausgang selbst erscheint. Wir haben dadurch den Wert $y(n-1)$ erzeugt. Wählt man den Koeffizienten $B1$ falsch, kann das Filter an seinem Ausgang schwingen.

In dem Schaltbild in Abbildung 60 wird nur ein Verzögerer zum gemeinsamen Verzögern von Ein- und Ausgangssignal benutzt. Dies ist eine Besonderheit, welche man der Filterübertragungsfunktion $H(z)$ nicht direkt ansieht. Es wird hier nicht Aus- und Eingangssignal separat sondern in einer Summenform als Signal ($w(n-1)$) verzögert. Dies erreicht man durch geschickte mathematische Umformung der Übertragungsfunktion. Man spart sich dadurch einen Verzögerer, was einer Speicherplatzeinsparung im den Filteralgorithmus realisierenden Signalprozessorsystem entspricht.

**Nur ein
Verzögerer**

8. Beispiele einfacher (symmetrischer) FIR-Filter

8.1. Einleitung

Die folgenden Beispiele wurden dem Buch „Tietze-Schenk Halbleiter-Schaltungstechnik“ entnommen. Sie zeigen sehr anschaulich die verschiedenen Filtertypen.

8.2. Tiefpass/Interpolator 1. Ordnung

Die Schaltung in Abbildung 63 zeigt ein FIR-Filter 1. Ordnung (N=1). Seine Verstärkung für Gleichspannung ist 1 (Summe aller Filterkoeffizienten). Es ist ein Tiefpass. Bei der halben Abtastfrequenz ist die Verstärkung nämlich 0. Man kann ihn sehr gut wegen der gleichen Koeffizienten zur Mittelwertbildung einsetzen.

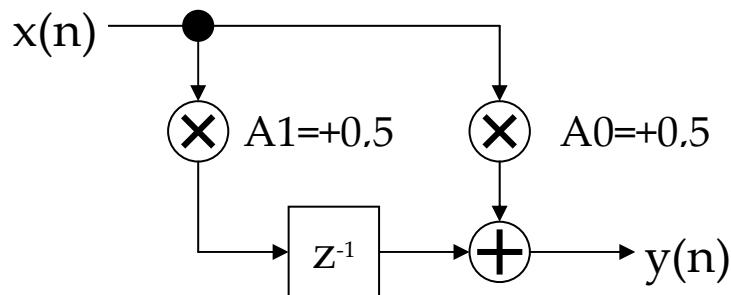


Abbildung 63: Tiefpass/Interpolator 1. Ordnung

$$H(z) = 0,5 \cdot (1 + z^{-1})$$

$$\underline{H}(f') = 0,5 \cdot (1 + \cos(2 \cdot \pi \cdot f') - j \sin(2 \cdot \pi \cdot f'))$$

$$|\underline{H}(f')| = |\cos(\pi \cdot f')|$$

$$f'_g = 0,25 \quad \text{weil } |\underline{H}(f'_g)| = |\cos(\pi \cdot f'_g)| = \frac{1}{\sqrt{2}}$$

$$\varphi = -\pi \cdot f'$$

$$t_{gr} = 0,5 \cdot T_A$$

Gleichung 67: Berechnung des Tiefpasses/Interpolators 1. Ordnung

Die 3dB-Grenzfrequenz ist z.B. bei einer Abtastfrequenz von 44100Hz (wie bei der Audio-CD) hier $0,25 \cdot 44100 = 11025\text{Hz}$. Dies sehen wir auch schön im Ergebnis der Simulation des Frequenzganges in Abbildung 66.

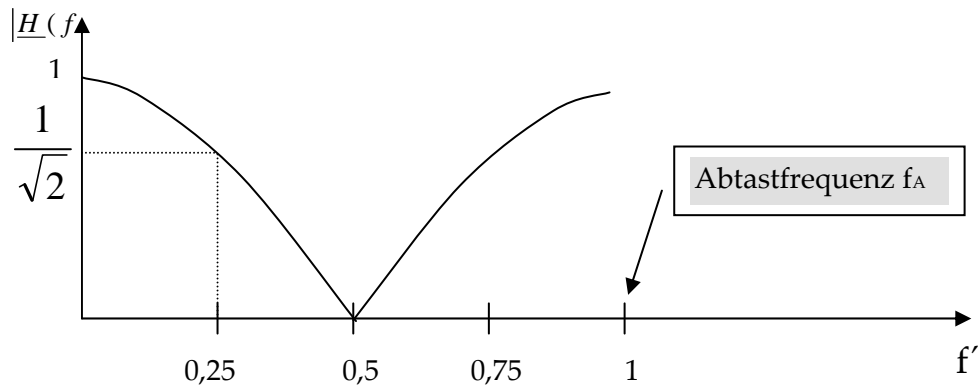


Abbildung 64: Frequenzgang des Tiefpasses 1. Ordnung aus Abbildung 63

Die Eingabe der Filterdaten in SCILAB sehen wir in Abbildung 65.

Dateiname „tiefpass-interpolator 1. Ordnung tietze-schenk.sce“



```

D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\Tiefpass Interpolator 1_Ordnung ti...
Datei Bearbeiten Suchen Ansicht Dokument Anwenden ?
D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\Tiefpass Interpolator 1_Ordnung tietze-schenk\tiefpass-interpolator 1_Ordnung tietze-schenk
Unbenannt1 tiefpass-interpolator 1_Ordnung tietze-schenk.sce
1 // Datei: tiefpass-interpolator 1_Ordnung tietze-schenk.sce
2 // Funktion: Frequenzgang Tiefpass 1. Ordnung
3 // Autor: HWN
4 // Datum: 12.8.2010
5 // Änderungen:
6 // 14.10.2009
7 // Ausgabefenster versetzt angeordnet
8 //
9 fa = 44100; // Definition der Abtastfrequenz
10 hz=(0.5 + 0.5*z^-1); // Definition der z-Übertragungsfunktion
11 lisis = syslin (1/fa, hz); // Umwandlung in lineares System
12
13
14 //Definition von 3 Grafik Ausgabefenstern
15 fenster0=scf(0); // Fenster 0 = für Bodeplot
16 fenster1=scf(1); // Fenster 1 = für Gruppenlaufzeit
17 fenster2=scf(2); // Fenster 2 = für linearen Frequenzgang
18
19
20 // Anzeige Bodediagramm in Fenster 0
21 fp = scf(fenster0); // Aktiviere Fenster 0 zur Ausgabe
22 scf(); // Grafikfenster auf Werkswerte
23 bode(lisis,10,12000);
24 fp.figure_position=[100,100]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
25
26
27 // Anzeige Gruppenlaufzeit in Fenster 1
28 [tg,fr]=group(100,hz); // Berechnung der Gruppenlaufzeit
29 fp = scf(fenster1); // Aktiviere Fenster 1 zur Ausgabe
30 plot(fr,tg); // Anzeige der Gruppenlaufzeit
31 // Beschriftung der Plotachsen
32 xtitle('Gruppenlaufzeit in Abtasttakten', 'Normierte Frequenz', 'Wert in
33 Abtasttakten');
34 fp.figure_position=[400,200]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
35
36 //Anzeige des Frequenzganges linear in Fenster 2
37 fp = scf(fenster2); // Aktiviere Fenster 2 zur Ausgabe
38 [hzm,fr]=frmag(hz,256);
39 plot(fr,hzm,'blue');
40 // Beschriftung der Plotachsen
41 xtitle('Frequenzgang linear', 'Normierte Frequenz', 'Betrag der Verstärkung');
42 fp.figure_position=[700,300]; // Ausgabefenster verschieben gegenüber vorherigem Fenster

```

Abbildung 65: SCILAB-Kommandos des Tiefpasses/Interpolator 1. Ordnung

Hier haben wir eine weitere Verbesserung der Anzeige durchgeführt. Wir erzeugen 3 Ausgabefenster, welche seitensversetzt übereinander dargestellt werden.

- Fenster 0 Anzeigedes Bodediagramms
- Fenster 1 Verlauf der Gruppenlaufzeit über der normierten Frequenz
- Fenster 2 Verlauf der Filterverstärkung über der normierten Frequenz

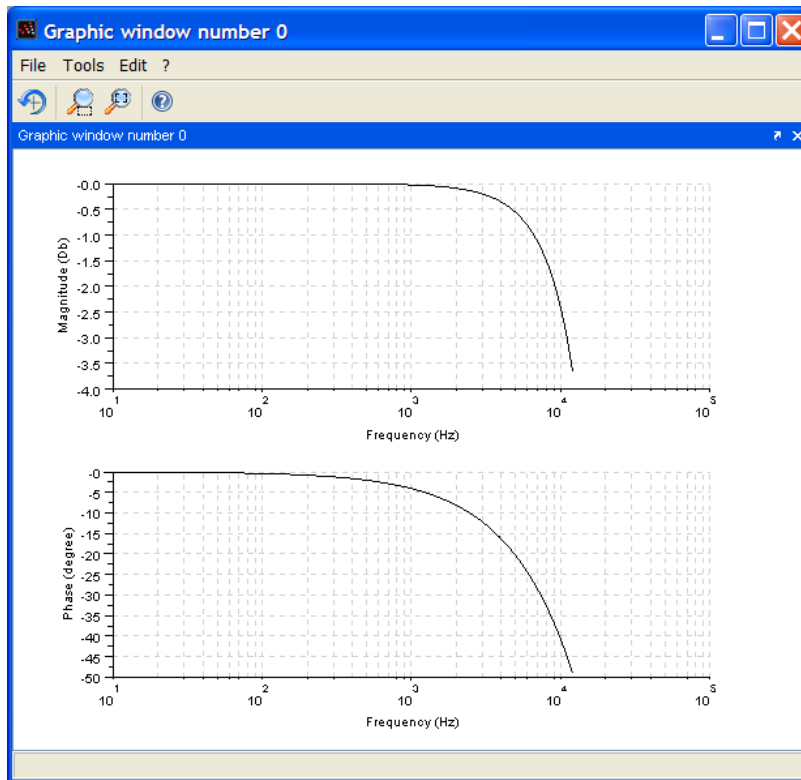


Abbildung 66: Bodediagramm des Tiefpasses/Interpolators 1. Ordnung

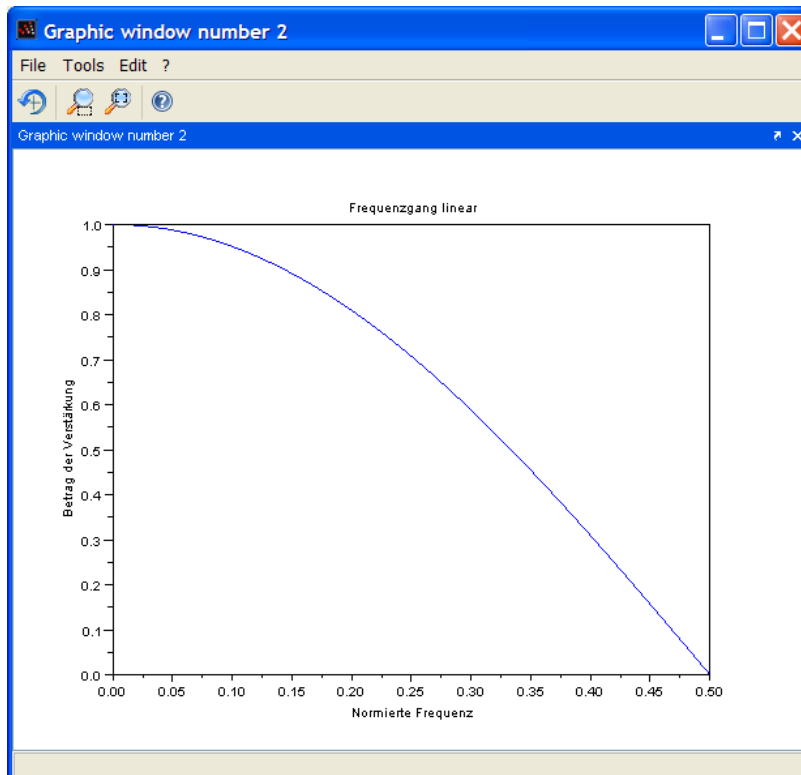


Abbildung 67: Frequenzgang des Tiefpasses/Interpolator 1. Ordnung (linear)

Beim linear dargestellten Frequenzgang wird das Ergebnis unserer Berechnung in Gleichung 67 bestätigt. Die Filterübertragungsfunktion ist cosinusförmig.

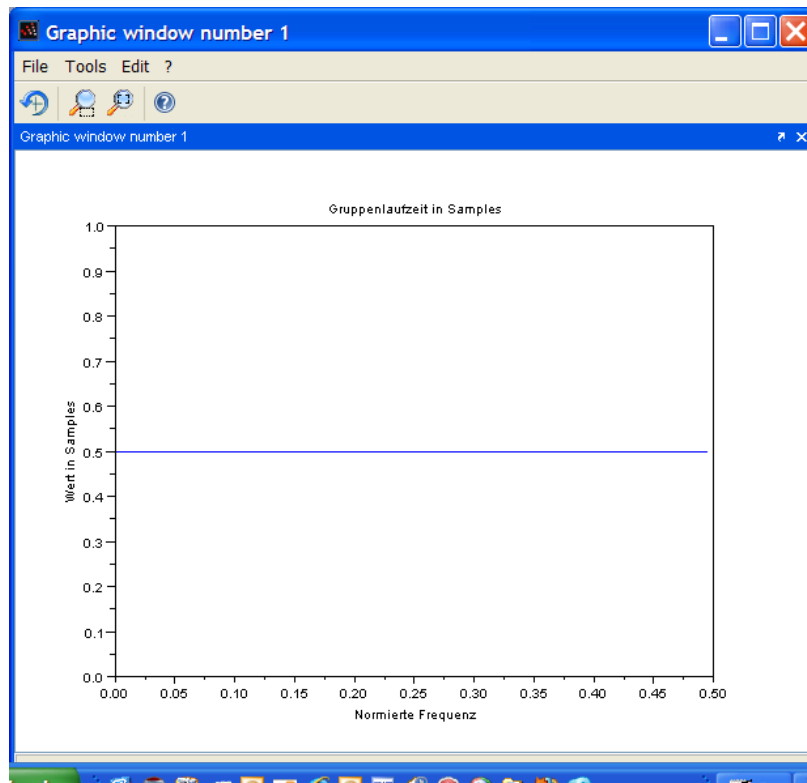


Abbildung 68: Gruppenlaufzeit des Tiefpasses/Interpolator 1. Ordnung

Bei der Gruppenlaufzeit wird ebenfalls das Ergebnis unserer Berechnung in Gleichung 67 bestätigt. Das Filter gibt also das Eingangssignal um 0,5 Samples (Mittelwert) verzögert am Ausgang aus.

8.3. Hochpass 1. Ordnung

Die Schaltung in Abbildung 69 zeigt ein FIR-Filter 1. Ordnung ($N=1$). Seine Verstärkung für Gleichspannung ist 0 (Summe aller Filterkoeffizienten). Somit ist es ein Hochpass. Bei der halben Abtastfrequenz ist die Verstärkung 1.

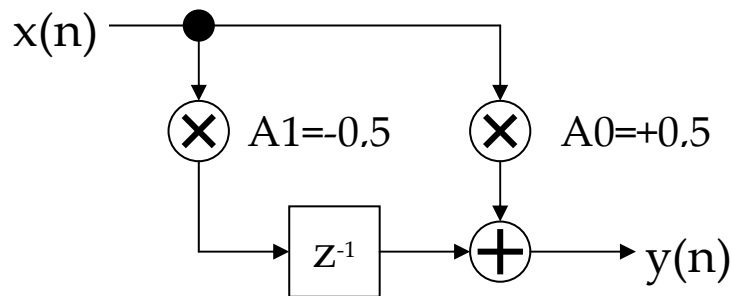


Abbildung 69: Hochpass 1. Ordnung

$$H(z) = 0,5 \cdot (1 - z^{-1})$$

$$\underline{H}(f') = 0,5 \cdot (1 - \cos(2 \cdot \pi \cdot f')) + j \sin(2 \cdot \pi \cdot f')$$

$$|\underline{H}(f')| = |\sin(\pi \cdot f')|$$

$$f'_g = 0,25 \quad \text{weil } |\underline{H}(f'_g)| = |\sin(\pi \cdot f'_g)| = \frac{1}{\sqrt{2}}$$

$$\varphi = \pi \cdot (0,5 - f')$$

$$t_{gr} = 0,5 \cdot T_A$$

Gleichung 68: Berechnung des Hochpasses 1. Ordnung

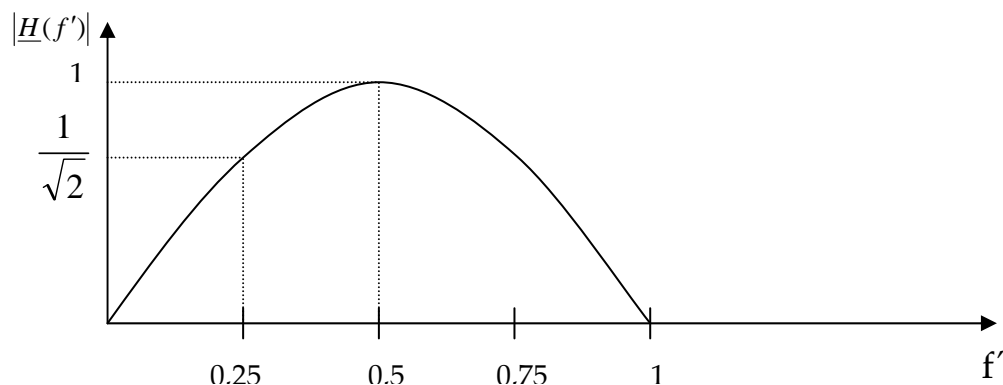


Abbildung 70: Frequenzgang des Hochpasses 1. Ordnung aus Abbildung 69

Die Eingabe der Filterdaten in SCILAB sehen wir in Abbildung 71.

Dateiname „hochpass 1 Ordnung tietze-schenk.sce“



```

1 // Datei: hochpass 1_Ordnung tietze-schenk.sce
2 // Frequenzgang Hochpass 1. Ordnung
3 // Eingabe der z-Übertragungsfunktion
4 // HWN
5 // 16.8.2010
6 fa = 44100; // Abtastfrequenz
7
8 // Achtung: SCILAB hat mit der folgenden Formel für die z-Übertragungsfunktion ein Problem!
9 // Bei der Berechnung der Filtergruppenlaufzeit entsteht eine Division durch 0!
10 // hz = 0.5 - 0.5*z^-1;
11 // Abhilfe: die Differenz in hz darf bei z^-1 = 1 nie 0 werden
12 // Also schreibt man einfach eine etwas ungenauere Formel (anstelle von 0.5 => 0.499999999)
13 hz = 0.5 - 0.499999999*z^-1;
14
15 lisys = syslin(1/fa, hz); // Berechnung des linearen zeitkontinuierlichen Systems aus hz
16 //Definition von 3 leeren Grafik Ausgabefenstern + direkt übereinander zeichnen,
17 ///// Fenster 0 = für Bodeplot, Fenster 1 = für Gruppenlaufzeit, Fenster 2 = für linearen Frequenzgang
18 fenster0=scf(0),fenster1=scf(1),fenster2=scf(2);
19
20 // Anzeige Bodediagramm in Fenster 0
21 fp = scf(fenster0),sdf(); // Aktiviere Fenster 0 zur Ausgabe, Grafikfenster auf Werkswerte
22 bode(lisys,10,12000);
23 fp.figure_position=[100,100]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
24
25 // Berechnung und Anzeige der Gruppenlaufzeit in Fenster 1
26 [tg,fr]=group(100,hz); // Berechnung der Gruppenlaufzeit an 10 Frequenzen
27 fp=scf(fenster1); // Aktiviere Fenster 1 zur Ausgabe
28 // Wertebereich skaliert auf 0 bis 0.5 für die normierte Frequenz
29 // und 0 bis 1 für die Gruppenlaufzeit
30 p=gca(); // Lade Handle auf Grafikfensterausgabe
31 p.thickness=3; // Linienstärke dicker für Grupp delayplot
32
33 plot2d(fr,tg,rect=[0,0,0.5,1],style = 5); // Gruppenlaufzeit mit dickem Strich in rot
34 p.thickness=1; // Linienstärke dünn für Gitterrasterplot
35 // Beschriftung der Plotachsen
36 xtitle('Gruppenlaufzeit in Abtasttakten', 'Normierte Frequenz', 'Wert in Abtasttakten');
37 xgrid(2); //Raster in blau wegen (2) einzeichnen
38 // Beschriftung der Plotachsen
39 xtitle('Gruppenlaufzeit in Abtasttakten', 'Normierte Frequenz', 'Wert in Abtasttakten');
40 fp.figure_position=[500,300]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
41
42 //Anzeige des Frequenzganges linear in Fenster 2
43 fp = scf(fenster2); // Aktiviere Fenster 2 zur Ausgabe
44 [hzm,fr]=fzmag(hz,256);
45 plot(fr,hzm,'blue');
46 // Beschriftung der Plotachsen
47 xtitle('Frequenzgang linear', 'Normierte Frequenz', 'Betrag der Verstärkung');
48 fp.figure_position=[900,500]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
    
```

Abbildung 71: SCILAB-Kommandos des Hochpasses 1. Ordnung

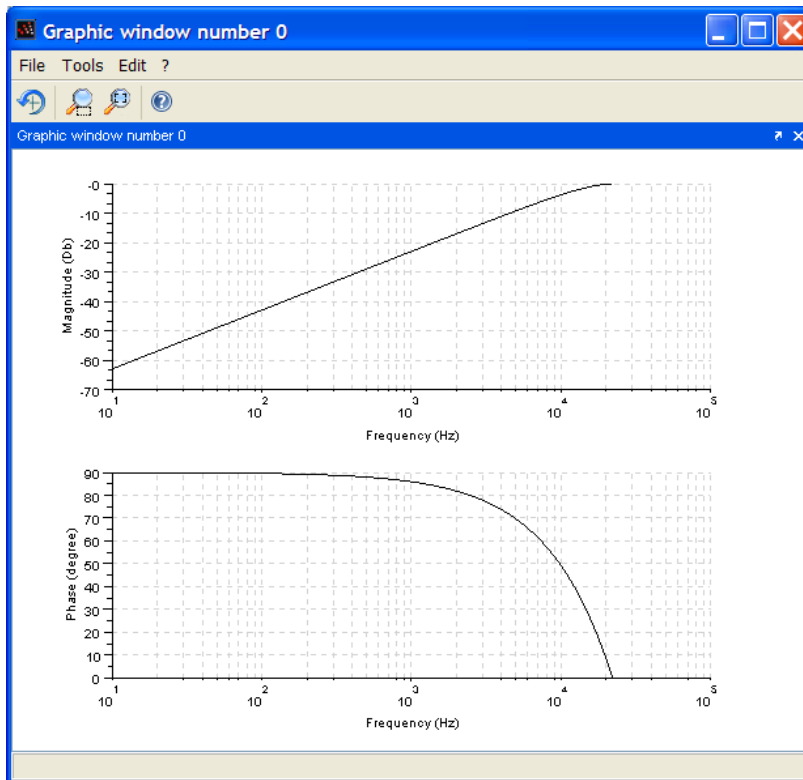


Abbildung 72: Bodediagramm des Hochpasses 1. Ordnung

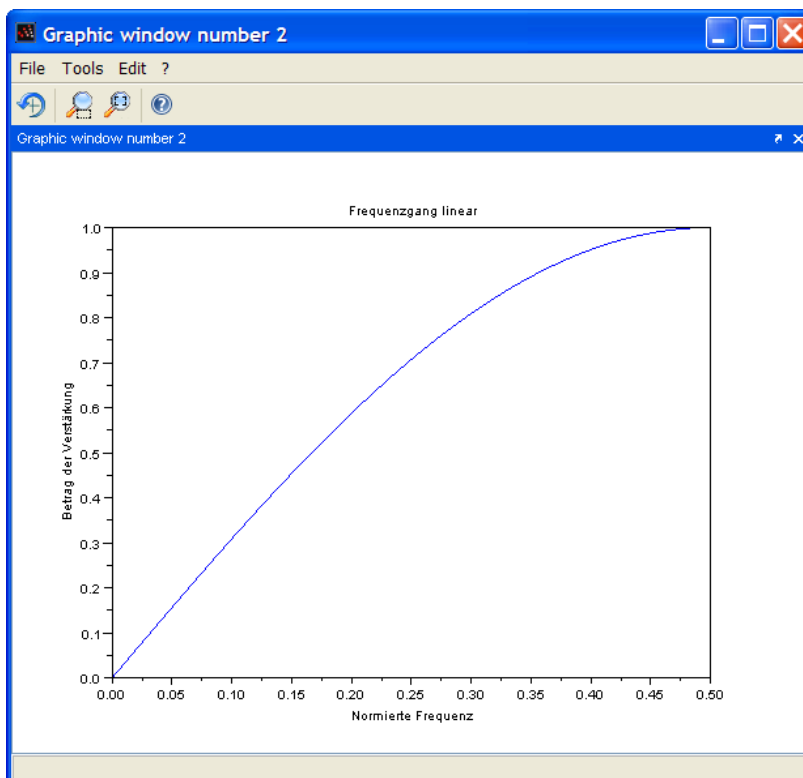


Abbildung 73: Frequenzgang des Hochpasses 1. Ordnung (linear)

Der lineare Frequenzgang entspricht unserer Berechnung in Gleichung 68.

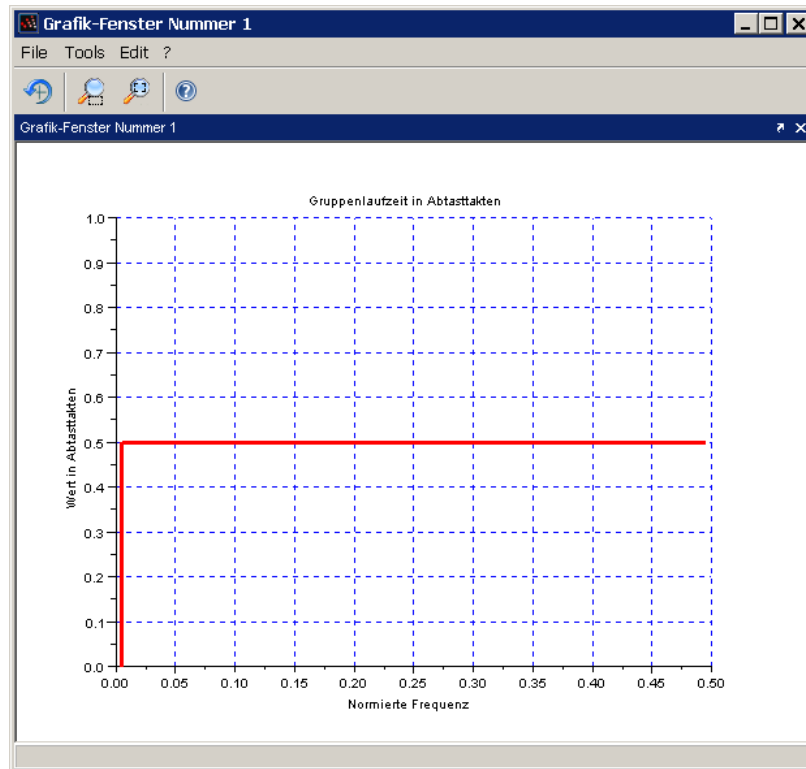


Abbildung 74: Gruppenlaufzeit des Hochpasses 1. Ordnung

Wie man in Abbildung 74 sieht, ist die Gruppenlaufzeit nicht konstant. Dies ist aber ein Fehler in der Berechnung in SCILAB. Wir mussten nämlich die z-Übertragungsfunktion modifizieren, weil es sonst eine Division durch 0 in SCILAB ergab (siehe Kommentare in Abbildung 71 ab Zeile 8).

8.4. Tiefpass 2. Ordnung

Die Schaltung in Abbildung 75 zeigt ein FIR-Filter 2. Ordnung ($N=2$). Seine Verstärkung für Gleichspannung ist 1 (Summe aller Filterkoeffizienten). Es ist ein Tiefpass. Bei der halben Abtastfrequenz ist die Verstärkung nämlich 0. Das Argument des „cos“ ist doppelt so groß wie beim Tiefpass 1. Ordnung aus Abbildung 63.

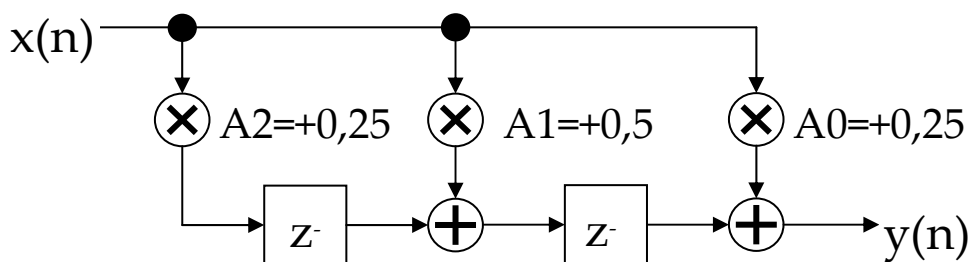


Abbildung 75: Tiefpass 2. Ordnung

$$H(z) = 0,25 + 0,5 \cdot z^{-1} + 0,25 \cdot z^{-2}$$

$$|H(f')| = 0,5 + 0,5 \cdot \cos(2 \cdot \pi \cdot f')$$

$$f'_g = \frac{1}{2 \cdot \pi} \arccos(\sqrt{2} - 1) = 0,182$$

$$\varphi = -2 \cdot \pi \cdot f'$$

$$t_{gr} = T_A$$

Gleichung 69: Berechnung des Tiefpasses 2. Ordnung

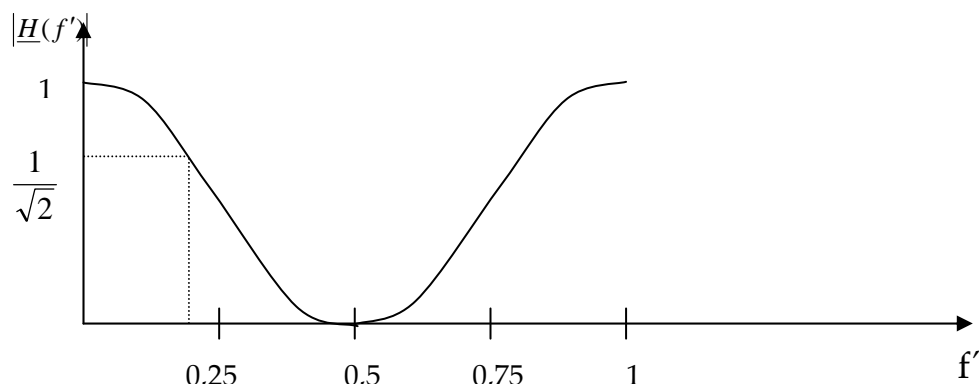


Abbildung 76: Frequenzgang des Tiefpasses 2. Ordnung aus Abbildung 75

Wir wollen hier nicht alle möglichen Diagramme mit SCILAB berechnen sondern nur beispielhaft vorgehen. Interessant ist immer der direkte Vergleich zwischen dem von Hand berechneten Frequenzgang (Abbildung 76) und dem Simulationsergebnis von SCILAB, welches wir z.B. in Abbildung 79 sehen.

Dateiname „tiefpass 2 Ordnung tietze-schenk.sce“



```
D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\tiefpass 2_Ordnung tietze-schenk\tiefp...
Datei Bearbeiten Suchen Ansicht Dokument Anwenden ?
D:\FHS Bachelor+Master\Vorlesungen\Digitale Filter DIGFIL\SCILAB\tiefpass 2_Ordnung tietze-schenk\tiefpass 2_Ordnung tietze-schenk.sce - Scilab Text Editor
Unbenannt 1 tiefpass 2_Ordnung tietze-schenk.sce
1 // Datei: tiefpass 2_Ordnung tietze-schenk.sce
2 // Funktion: Frequenzgang Tiefpass 1. Ordnung
3 // Autor: HWN
4 // Datum: 16.8.2010
5 fa = 44100; // Definition der Abtastfrequenz
6 hz=(0.25 + 0.5*z^-1 + 0.25*z^-2); // Definition der z-Übertragungsfunktion
7 lisy = syslin (1/fa, hz); // Umwandlung in lineares System
8
9 //Definition von 3 Grafik Ausgabefenstern
10 fenster0=scf(0); // Fenster 0 = für Bodeplot
11 fenster1=scf(1); // Fenster 1 = für Gruppenlaufzeit
12 fenster2=scf(2); // Fenster 2 = für linearen Frequenzgang
13
14 // Anzeige Bodediagramm in Fenster 0
15 fp=scf(fenster0); // Aktiviere Fenster 0 zur Ausgabe
16 bode(lisy,10,12000);
17 fp.figure_position=[100,100]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
18
19
20
21 // Anzeige Gruppenlaufzeit in Fenster 1
22 [tg,fr]=group(100,hz); // Berechnung der Gruppenlaufzeit
23 fp=scf(fenster1); // Aktiviere Fenster 1 zur Ausgabe
24 plot(fr,tg); // Anzeige der Gruppenlaufzeit
25 // Beschriftung der Plotachsen
26 xtitle('Gruppenlaufzeit in Abtasttakten', 'Normierte Frequenz', 'Wert in Abtasttakten');
27 fp.figure_position=[500,300]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
28
29
30 //Anzeige des Frequenzganges linear in Fenster 2
31 fp=scf(fenster2); // Aktiviere Fenster 2 zur Ausgabe
32 [hzm,fr]=frmag(hz,256);
33 plot(fr,hzm,'blue');
34 // Beschriftung der Plotachsen
35 xtitle('Frequenzgang linear', 'Normierte Frequenz', 'Betrag der Verstärkung');
36 fp.figure_position=[900,500]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
37
```

Abbildung 77: SCILAB-Kommandodatei des Tiefpasses 2. Ordnung

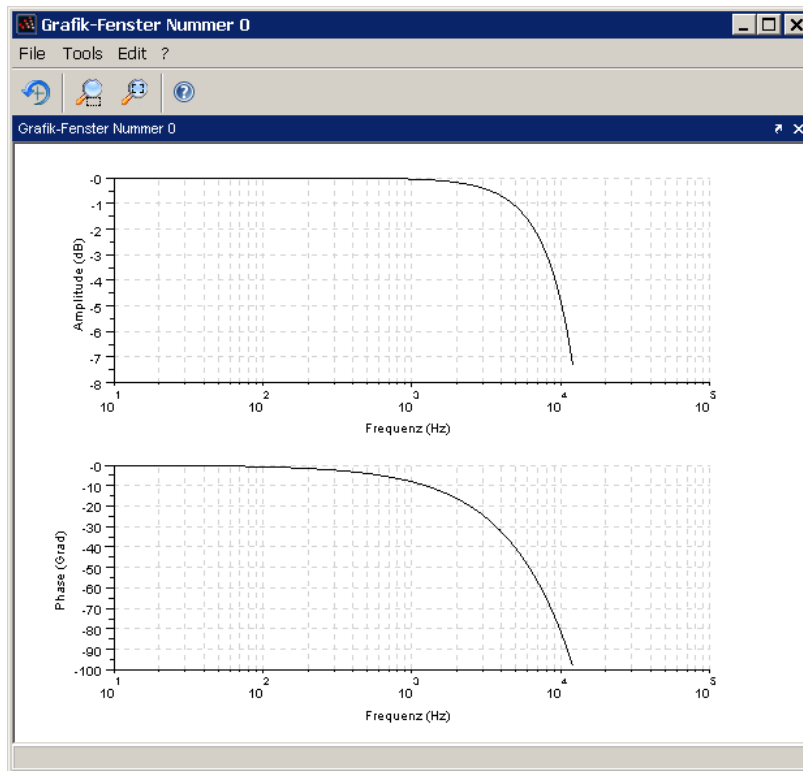


Abbildung 78: Bodediagramm des Tiefpasses 2. Ordnung

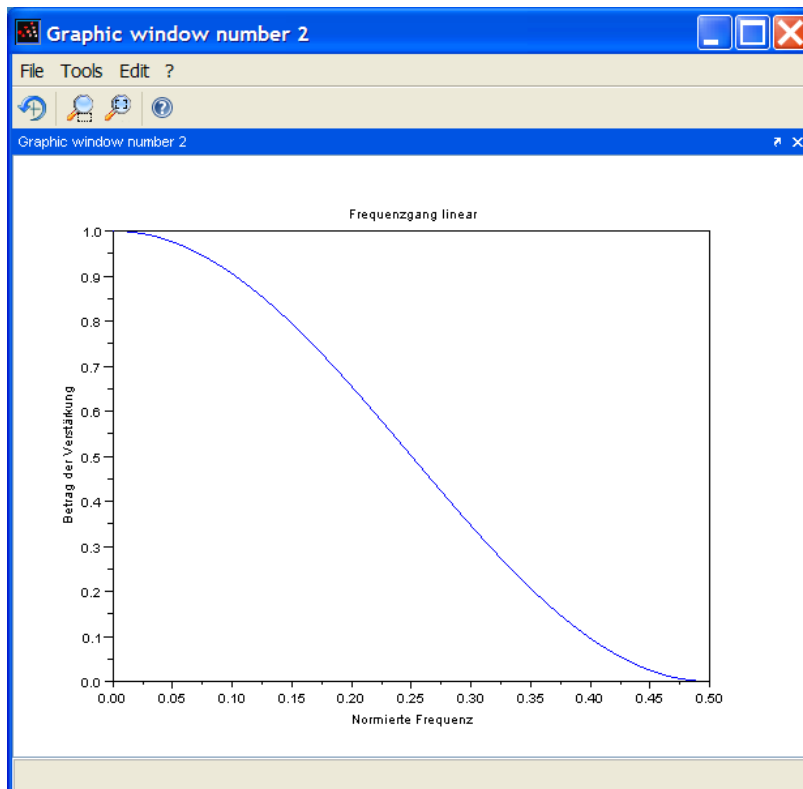


Abbildung 79: Mit SCILAB simulierter Frequenzgang des Tiefpasses 2. Ordnung (linear)

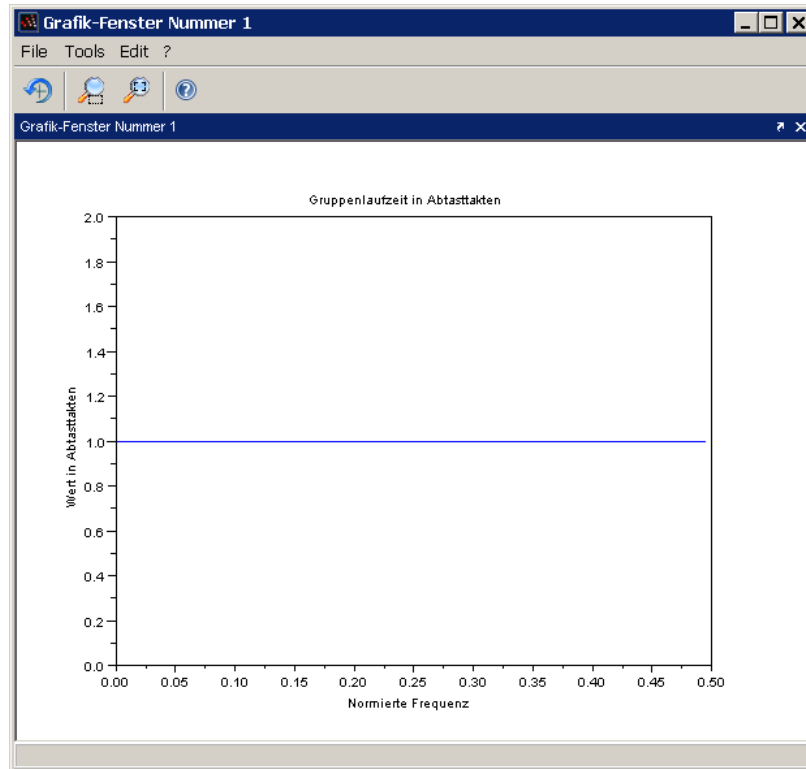


Abbildung 80: Gruppenlaufzeit des Tiefpasses 2. Ordnung

8.5. Hochpass 2. Ordnung

Die Schaltung in Abbildung 81 zeigt ein FIR-Filter 2. Ordnung ($N=2$). Seine Verstärkung für Gleichspannung ist 0 (Summe aller Filterkoeffizienten). Es ist darum ein Hochpass. Bei der halben Abtastfrequenz ist die Verstärkung nämlich 1. Das Argument des „cos“ ist doppelt so groß wie beim Hochpass 1. Ordnung aus Abbildung 69.

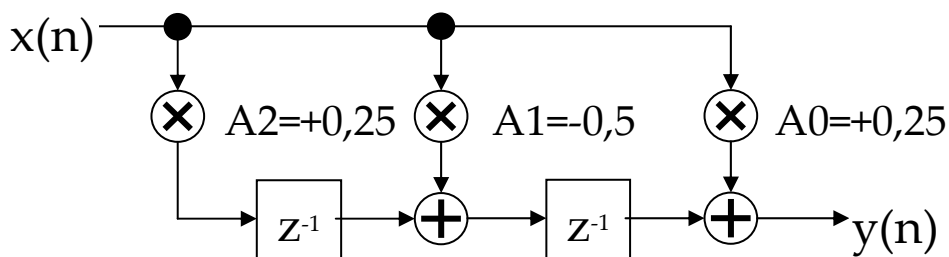


Abbildung 81: Hochpass 2. Ordnung

$$H(z) = 0,25 - 0,5 \cdot z^{-1} + 0,25 \cdot z^{-2}$$

$$|H(f')| = 0,5 - 0,5 \cdot \cos(2 \cdot \pi \cdot f')$$

$$f'_g = \frac{1}{2 \cdot \pi} \arccos(1 - \sqrt{2}) = 0,318$$

$$\varphi = -2 \cdot \pi \cdot f'$$

$$t_{gr} = T_A$$

Gleichung 70: Berechnung des Hochpasses 2. Ordnung

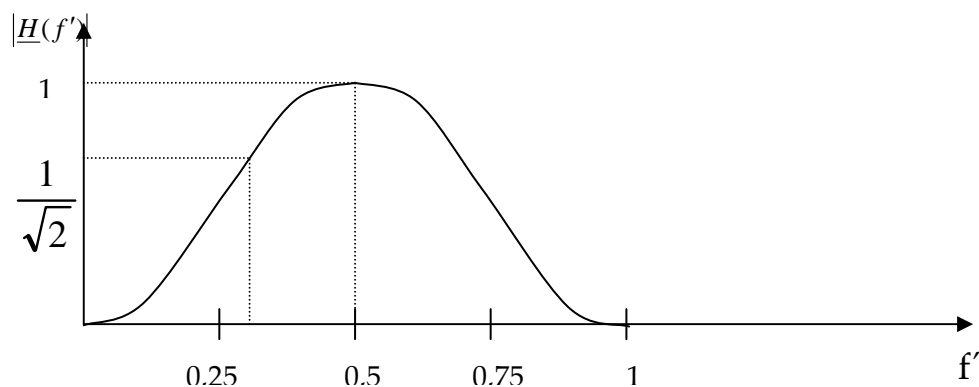


Abbildung 82: Frequenzgang des Hochpasses 2. Ordnung aus Abbildung 81

Dateiname „hochpass 2 Ordnung tietze-schenk.sce“



```
hochpass 2_Ordnung tietze-schenk.sce - Scilab Text Editor
Datei Bearbeiten Suchen Ansicht Dokument Anwenden ?
hochpass 2_Ordnung tietze-schenk.sce - Scilab Text Editor
Unbenannt 1 hochpass 2_Ordnung tietze-schenk.sce
1 // Datei: hochpass 2_Ordnung tietze-schenk.sce
2 // Funktion: Frequenzgang Hochpass 2. Ordnung
3 // Autor: HWN
4 // Datum: 16.8.2010
5 fa = 44100; // Definition der Abtastfrequenz
6 hz=(0.25 - 0.5*z^-1 + 0.24999999999999999*z^-2); // Definition der z-Übertragungsfunktion
7 // Eigentlich sollte bei z^-2 der Multiplikator 0,25 stehen
8 // Dies ergibt aber wie beim Hochpass 1. Ordnung (Datei "hochpass 1_Ordnung tietze-schenk.sce") eine
9 // Division durch 0 in SCILAB!
10
11 lisis = syslin (1/fa, hz); // Umwandlung in lineares System
12
13 //Definition von 3 Grafik Ausgabefenstern
14 fenster0=scf(0); // Fenster 0 = für Bodeplot
15 fenster1=scf(1); // Fenster 1 = für Gruppenlaufzeit
16 fenster2=scf(2); // Fenster 2 = für linearen Frequenzgang
17
18 // Anzeige Bodediagramm in Fenster 0
19 fp=scf(fenster0); // Aktiviere Fenster 0 zur Ausgabe
20 bode(lisis,10,100000);
21 fp.figure_position=[100,100]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
22
23 // Berechnung und Anzeige der Gruppenlaufzeit
24 [tg,fr]=group(100,hz); // Berechnung der Gruppenlaufzeit
25 fp=scf(fenster1); // Aktiviere Fenster 1 zur Ausgabe
26 // Wertebereich skaliert auf 0 bis 0.5 für die normierte Frequenz
27 // und 0 bis 1 für die Gruppenlaufzeit
28 p=gca(); // Lade Handle auf Grafikfensterausgabe
29 p.thickness=3; // Linienstärke dicker für Grupp delayplot
30
31 plot2d(fr,tg,rect=[0,0,0.5,1],style = 5);
32 p.thickness=1; // Linienstärke dünn für Gitterrasterplot
33 // Beschriftung der Plotachsen
34 xtitle('Gruppenlaufzeit in Abtasttakten', 'Normierte Frequenz', 'Wert in Abtasttakten');
35 xgrid(2); //Raster in blau wegen (2) einzeichnen
36 fp.figure_position=[500,300]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
37
38 //Anzeige des Frequenzganges linear in Fenster 2
39 fp=scf(fenster2); // Aktiviere Fenster 2 zur Ausgabe
40 [hzm,fr]=frmag(hz,256);
41 plot(fr,hzm,'blue');
42 // Beschriftung der Plotachsen
43 xtitle('Frequenzgang linear', 'Normierte Frequenz', 'Betrag der Verstärkung');
44 fp.figure_position=[900,500]; // Ausgabefenster verschieben gegenüber vorherigem Fenster
```

Abbildung 83: SCILAB Kommandodatei des Hochpasses 2. Ordnung

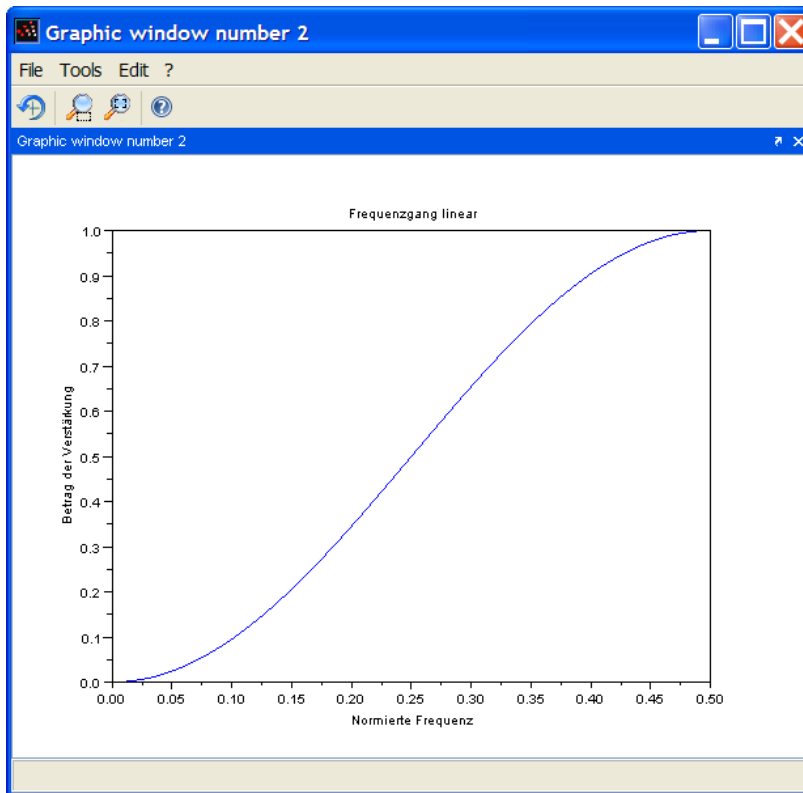


Abbildung 84: Mit SCILAB simulierter Frequenzgang des Hochpasses 2. Ordnung (linear)

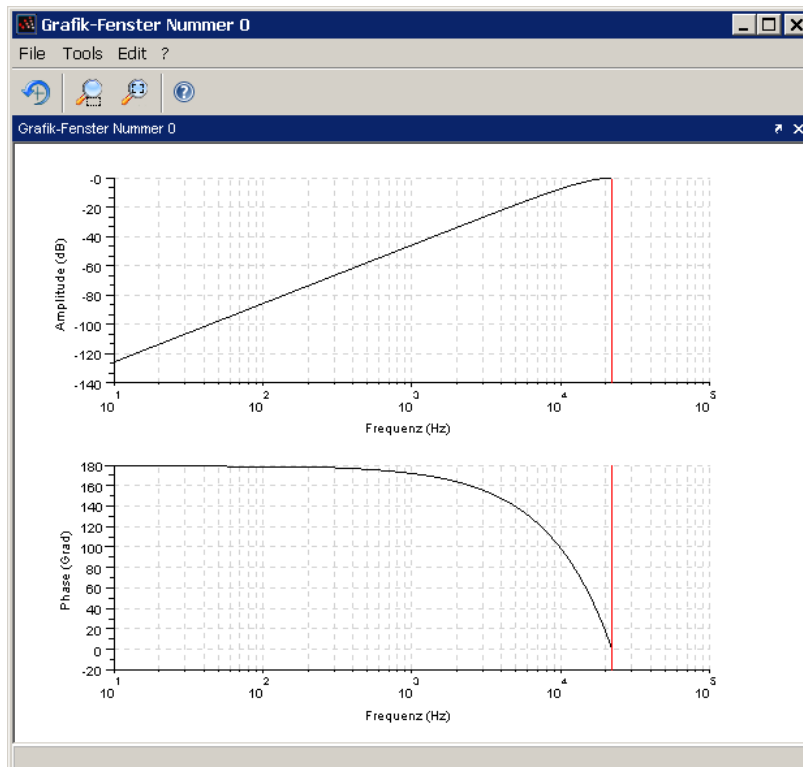


Abbildung 85: Mit SCILAB simuliertes Bodediagramm des Hochpasses 2. Ordnung

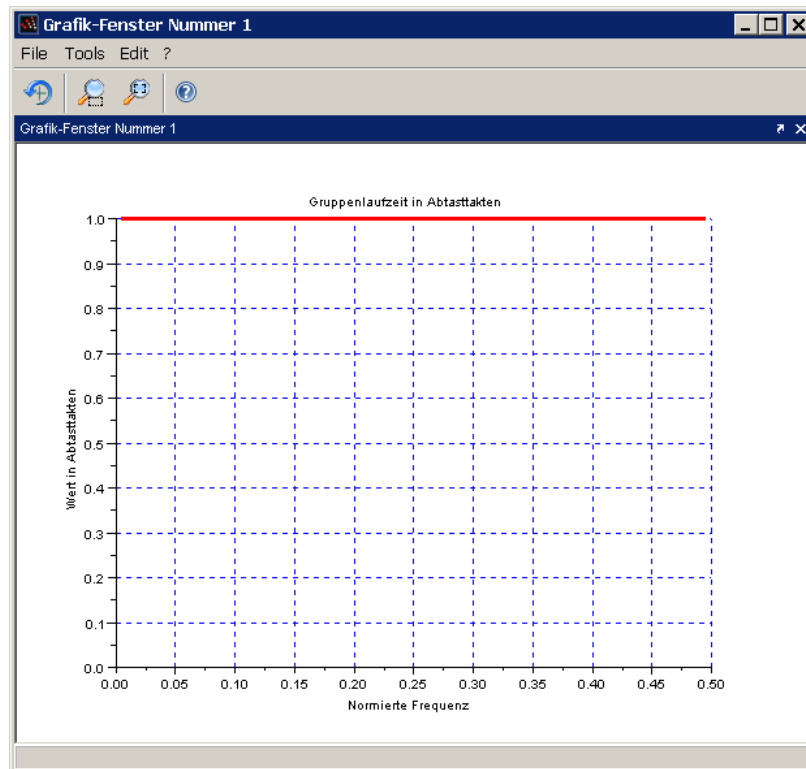


Abbildung 86: Mit SCILAB simulierte Gruppenlaufzeit des Hochpasses 2. Ordnung

8.6. Bandsperre 2. Ordnung

Die Schaltung in Abbildung 87 zeigt eine Bandsperre 2. Ordnung ($N=2$). Die Verstärkung für Gleichspannung ist 1 (Summe aller Filterkoeffizienten).

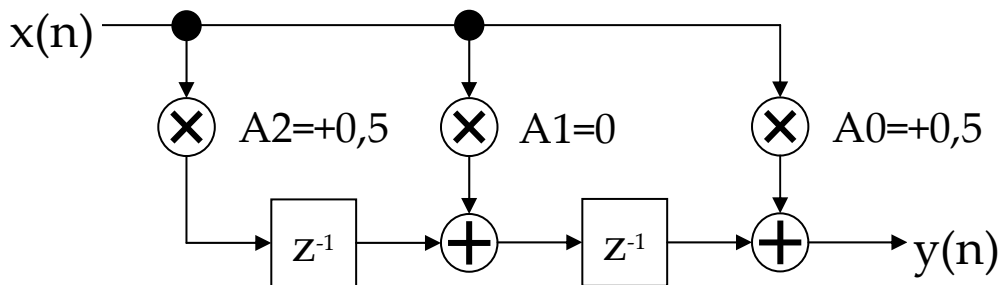


Abbildung 87: Bandsperre 2. Ordnung

$$H(z) = 0,5 + 0,5 \cdot z^{-2}$$

$$|H(f')| = |\cos(2 \cdot \pi \cdot f')|$$

$$f'_{r} = 0,25 = \text{Sperrfrequenz (Resonanzfrequenz)}$$

$$\varphi = -2 \cdot \pi \cdot f'$$

$$t_{gr} = T_A$$

$$Q = \frac{f'_{r}}{B} = 1$$

Gleichung 71: Berechnung der Bandsperre 2. Ordnung

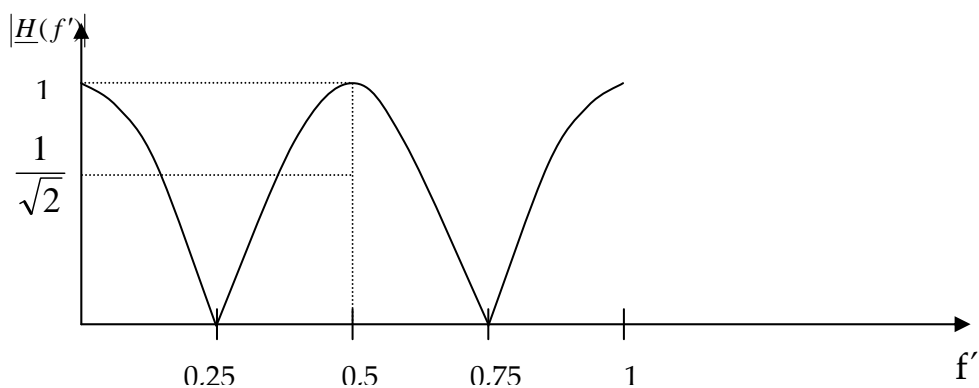


Abbildung 88: Frequenzgang der Bandsperre 2. Ordnung aus Abbildung 87

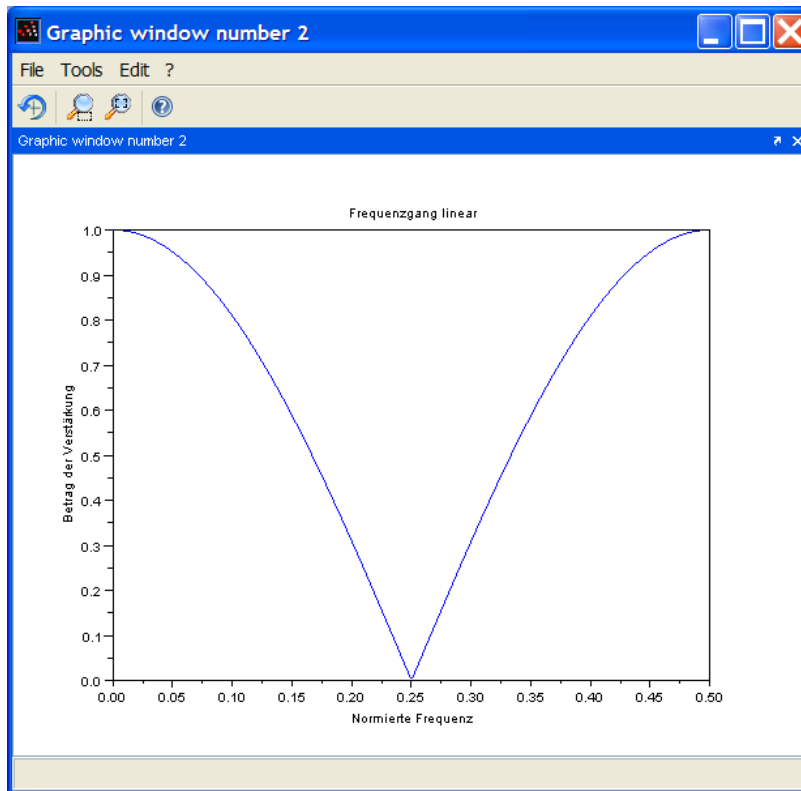


Abbildung 90: Mit SCILAB simulierter Frequenzgang der Bandsperre 2. Ordnung (linear)

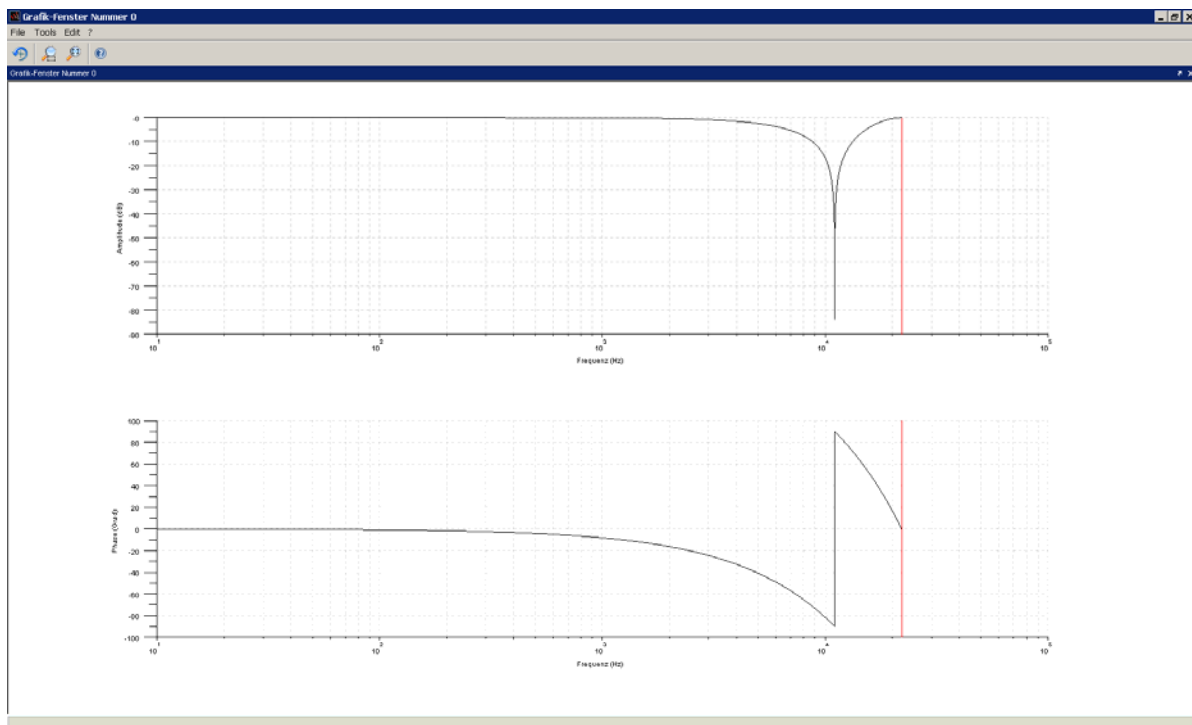


Abbildung 91: Mit SCILAB simuliertes Bodediagramm der Bandsperre 2. Ordnung

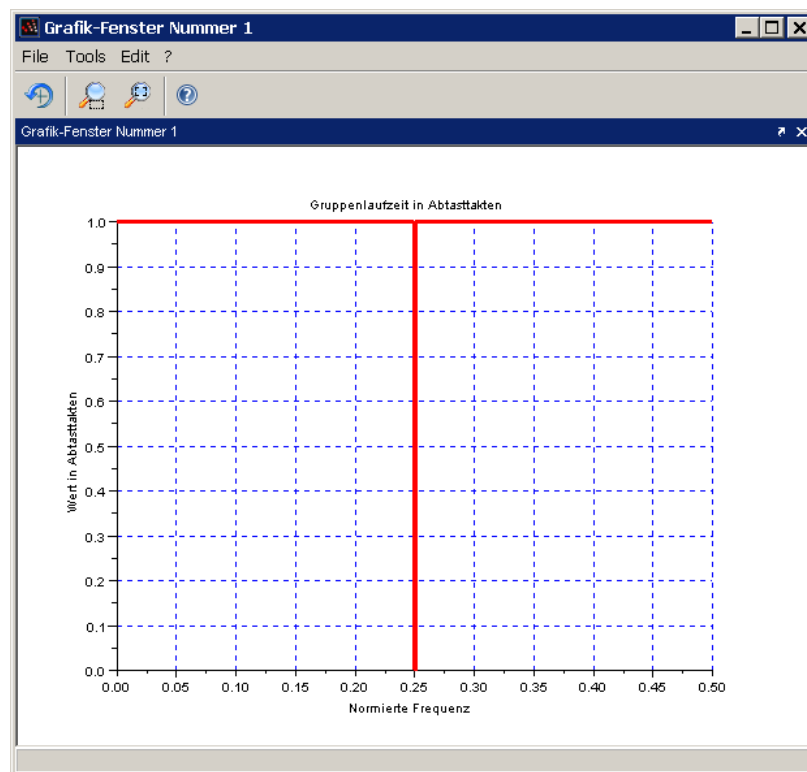


Abbildung 92: Mit SCILAB simulierte Gruppenlaufzeit der Bandsperre 2. Ordnung

Die Breite der Nadel nach 0 bei der Gruppenlaufzeit (bei Frequenz 0,25) ist abhängig von der Anzahl der in Zeile 24 der Kommandodatei angegebenen Stützstellen zur Berechnung der Gruppenlaufzeit.

```
[tg,fr]=group(10000,hz); // Berechnung der Gruppenlaufzeit
```

Wir haben hier den Wert 10000 angegeben. Je kleiner der Wert umso breiter die Nadel!

Literaturverzeichnis



Digitale Signalverarbeitung

Daniel CH. Von Grünigen

Fachbuchverlag Leipzig

ISBN 3-446-21445-3

Gut verständliche Einführung in die digitale Signalverarbeitung sowie die Theorie der digitalen Filter. Die beiliegende CD enthält verschiedene Testprogramme für MATLAB + DSP der Fa. Texas Instruments.



Digitale Signalverarbeitung

James D. Broesch

Elektor Verlag

ISBN 3-89576-066-8

Sehr leicht verständliche Einführung in Anwendung und Arbeitsweise von Signalprozessoren. Die beiliegende CD enthält verschiedene Testprogramme (DSP Calculator genannt) wie Signalgeneratoren, Fouriertransformation, Filterentwurf, Diskrete Faltung usw.



DIGITAL SIGNAL PROCESSING

PROAKIS/MANOLAKIS

Prentice Hall International Editions

ISBN 0-13-394289-9

Grundlagenwerk. Eine umfassende Behandlung der digitalen Signalverarbeitung. Nach meiner Meinung momentan das beste Buch am Markt.



Digitale Signalverarbeitung in der Nachrichtentechnik

Peter Gerdson, Peter Kröger

Springer-Verlag

ISBN 3-540-61194-0

49,95 €

Sehr gut für Nachrichtentechniker, welche mit MATLAB üben wollen. Es werden die wichtigsten Algorithmen der Nachrichtentechnik behandelt.



Digital Signal Processing and the Microcontroller

Dale Grover, John R. Deller

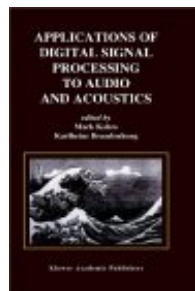
MOTOROLA UNIVERSITY, Prentice Hall

www.phptr.com

ISBN 0-13-081348-6

62,95 €

Hier wird die digitale Signalverarbeitung sehr anschaulich erklärt! Allerdings sind die Programmbeispiele alle für einen Microcontroller (68HC16) in Maschinensprache.



APPLICATIONS OF DIGITAL SIGNAL PROCESSING TO AUDIO AND ACOUSTICS

Herausgegeben von M. Kahrs und K. Brandenburg

Kluwer Academic Publishers Boston, Dordrecht, London

ISBN 0-7923-8130-0

Ein Werk von mehreren Spezialisten. Hier stellt jeder sein spezielles Arbeitsgebiet in der Audiosignalverarbeitung dar (Wavetablesynthese, Raumhall, Audiorestauration usw.)



Digitale Verarbeitung analoger Signale

Stearn/Hush

Oldenbourg Verlag

ISBN 3-486-22027-6

49,80 €

Grundlagenwerk der digitalen Signalverarbeitung



Digitale Audiosignalverarbeitung

Udo Zölzer

B.G. Teubner Stuttgart

ISBN 3-519-16180-X

44,90 €

Grundlagenwerk der digitalen Audiotechnik

Stichwortverzeichnis

A	
Abtaster	16
Abtastfrequenz.....	11
Abtasttheorem.....	9
Adaptive Filter	1
Allpass.....	1
Amplitudengang	
Ideal	4
Amplitudenquantisierung	11
Analog-Digitalwandler.....	9
Analoge Signalverarbeitung	8
Analoges Referenzfilter	49
Approximationsarten.....	5
Audiosignal	9
B	
Bandpass.....	1
Bandsperr	
Ideal	4
Bandsperr.....	1
Bandsperr 2. Ordnung	94
Bodediagramm	95
Gruppenlaufzeit.....	96
Phasengang	94
SCILAB-Kommandodatei	94
bilineare Transformation.....	49
Bilineare Transformation.....	47
Beispiel	51
Fehler	52
Biquad	59
Blockschaltbild eines FIR-Filters	73
D	
Differentialgleichungen.....	8
Differenzgleichung.....	48
Differenzgleichung.....	17
Differenzgleichungen	8, 10
Digital-Analogwandler.....	10
Digitale Signalverarbeitung	8
Diracfolge.....	20
Dirac-Folge	14
DSP	7, 62
Durchlassbereich	4
E	
Einheits-Impulsfolge	14
Einheits-Sprungfolge.....	14
Eulersche Beziehung	26
F	
Faltung anschaulich	23
Faltungsoperator.....	21
Faltungssumme.....	21
Fenstermethode.....	32
Filter	
Toleranzbereich.....	5
Filterkoeffizienten.....	1
Filterkoeffizienten in Datei	
schreiben	70
FIR	
Fenstermethode.....	32, 68
Frequenzgang.....	26
Hamming-Fenster.....	36
Impulsantwortfolge.....	27, 33
Komplexer Frequenzgang	27
Phasengang.....	31
punktsymmetrisch.....	27, 31
Remez-Exchange.....	32
Rohkoeffizienten.....	35
spiegelsymmetrisch.....	27, 30
Symmetrie.....	27
FIR-Filter	
Bandsperr 2. Ordnung	93
Beispielberechnung eines	
Tiefpasses.....	70
einfache Beispiele.....	76
Gruppenlaufzeit.....	31

Hochpass 1. Ordnung	81	Biquad-Realisierung	61
Hochpass 2. Ordnung	89	Bodediagramm	59
Tiefpass 1. Ordnung	44, 76	Hochpass 1. Ordnung	74
Tiefpass 1. Ordnung		Pole und Nullstellen	61
Gruppenlaufzeit	44	z-Übertragungsfunktion	59
Tiefpass 1. Ordnung Phasengang		IIR-Kurzdaten der rekursiven Filter	
.....	44	(IIR-Filter)	47
Tiefpass 2. Ordnung	85	Impulsantwortfolge	20, 22, 33
FIR-Kurzdaten der nichtrekursiven		inverse Fouriertransformierte	33
Filter (FIR-Filter)	26	Inverse z-Transformation	75
Fourier-Transformierte	11		
Frequenz		K	
Normiert	6	Kodierung	11
Frequenz des analogen		Koeffizienten der Filter	1
Referenzfilters	49		
Frequenz des digitalen		L	
Wunschfilters	49	Laplacetransformation	16
Frequenzabtastfilter	1	linearer Phasengang	26
Frequenznormierung	48	ln(z)	48
		LTSpice	44
G			
Grenzyklen	26, 47	N	
		nichtlinearer Phasengang	47
H		nichtrekursive Filter	25
H(z)	18	Nichtrekursives System	2
Hochpass	1	Normierte Frequenz	6
Hochpass 1. Ordnung		Nyquistfrequenz	5
Frequenzgang	84		
Gruppenlaufzeit	84	P	
SCILAB-Kommandodatei	82	Prinzip digitales Filter	44
Hochpass 2. Ordnung	89		
Bodediagramm	92	R	
Gruppenlaufzeit	92	Realisierung digitaler Filter	73
SCILAB-Kommandodatei	90	Referenzfilter	49
		Rekonstruktion des analogen	
		Ausgangssignals	9
I		Rekursives System	2
Ideale Bandsperre	4	Remez-Exchange	32
Idealer Tiefpass	3		
IIR	47	S	
IIR-Filter		SCILAB	
Beispielentwurf als Bandpass ..	66	Biquadfilter	60
Biquad	59		

Bodediagramm	53	Gruppenlaufzeit.....	80
cells	60, 67	Tiefpass 2. Ordnung	
Datei schreiben.....	71	Frequenzgang.....	87
Editor.....	55, 66	Gruppenlaufzeit.....	88
fact	62	SCILAB-Kommandodatei.....	86
Fenster, mehrere	78	Transversalfilter	25
Filterentwurfsvfunktionen	68		
Filtergrundverstärkung	62	U	
FIR-Beispiel	70	Übergangsbereich	4
FIR-Frequenznormierung.....	70	Übertragungsfunktion $H(z)$	18
Koeffizientendatei	64, 72		
Koeffizientendatei erzeugen	71	V	
Kommandodatei	55	Verzögerer	75
wfir	69	VISUAL DSP	62
z-Übertragungsfunktion.....	53		
si-Funktion.....	12	W	
Signalprozessor.....	7, 62	wfir.....	69
Koeffizientendatei	62		
Sperrbereichbereich.....	4	Z	
Systemantwort	20	Z-Definition	47
		Zeitquantisierung	11
T		z-Transformation	15, 17
Tiefpass	1, 9	Z-Übertragungsfunktion	1
Ideal.....	3	Zuordnungssymbol der z-	
Tiefpass 1. Ordnung		Transformation	17
Frequenzgang.....	80		