

Kapitel 5: FIR- und IIR-Filterentwurf

Inhaltsverzeichnis

5.1. EINLEITUNG	2
5.2. FILTERSPEZIFIKATION	3
5.3. FIR-FILTER.....	4
5.3.1. TYPISIERUNG	4
5.3.2. ENTWURF MIT DER FENSTERMETHODE (WINDOWING)	5
5.3.3. TRANSFORMATION: TIEFPASS > HOCHPASS, BANDPASS, BANDSPERRE	9
5.3.4. KAISER-WINDOW	9
5.3.5. FENSTERENTWURF SPEZIELLER FIR-FILTER (FAKULTATIV).....	11
5.3.6. ENTWURF MIT FREQUENZABTASTUNG (FREQUENCY SAMPLING)	13
5.3.7. ENTWURF IM Z-BEREICH.....	13
5.4. IIR-FILTER	14
5.4.1. ENTWURF MIT ANALOGEN PROTOTYP-FILTERN	14
5.4.2. DIREKTER ENTWURF IM DISKRETEN.....	17
5.5. REALISIERUNGSSTRUKTUREN	18
5.5.1. DIREKT-STRUKTUREN	18
5.5.2. TRANSPONIERTE DIREKT-STRUKTUR.....	19
5.5.3. BIQUADS (SECOND ORDER SECTIONS)	20
5.5.4. PARALLEL-STRUKTUREN.....	21
5.5.5. LATTICE-FILTER (GITTER- UND LEITER-STRUKTUREN) (FAKULTATIV)	22
5.6. NICHTIDEALITÄTEN.....	23
5.6.1. BEISPIEL „FESTKOMMA-REALISIERUNG EINES IIR-FILTERS“	25
5.7. ZUSAMMENFASSUNG FILTER-ENTWURF UND MATLAB-BEFEHLE	28
5.7.1. WAHL FIR- ODER IIR-FILTER.....	28
5.7.2. FIR-FILTER-ENTWURFMETHODEN.....	29
5.7.3. IIR-FILTER-ENTWURFMETHODEN.....	30
5.7.4. FIR-FILTER: VERGLEICH DER FREQUENZGÄNGE	31
5.7.5. IIR-FILTER: VERGLEICH DER FREQUENZGÄNGE.....	32

Literatur- bzw. Quellenverzeichnis

- [1] A.V. Oppenheim, R.W. Schaffer, J. R. Buck: „Zeitdiskrete Signalverarbeitung“, 2., überarbeitete Auflage, ISBN 3-8273-7077-9, Neuübersetzung der aktuellen US-Ausgabe, Pearson Studium, 2004.
- [2] M. Meier: „Signalverarbeitung“, ISBN 3-528-16955-9, Vieweg Verlag, Oktober 2000.
- [3] S.J. Orfanidis: „Introduction to Signal Processing“, ISBN 0-13-209172-0, Prentice Hall, 1996.

5.1. Einleitung

In diesem Kapitel betrachten wir den klassischen Digitalfilterentwurf, in dem primär *ideale* Tiefpass- (TP), Hochpass- (HP), Bandpass- (BP) und Bandsperre-(BS)-Filter approximiert werden.

Diese Filter werden im Frequenzbereich spezifiziert. Wir betrachten deshalb zuerst den Frequenzgang eines Filters im Allgemeinen und dann die Art und Weise der Filterspezifikation.

Die Digitalfilter werden in zwei grosse Klassen unterteilt, nämlich in die Klasse der rekursiven bzw. IIR-Filter und in die Klasse der nichtrekursiven bzw. FIR-Filter.

FIR-Filter sind nichtrekursive LTD-Systeme und werden meistens in der Transversalstruktur bzw. der Direktform 1 realisiert. Sie haben die folgenden Vor- und Nachteile:

- + Mit FIR-Filtern kann man vollständig lineare Phasengänge realisieren.
- + FIR-Filter sind immer stabil. Alle Pole befinden sich im Ursprung, also immer innerhalb des Einheitskreises.
- + FIR-Filter sind toleranter gegenüber Quantisierungseffekten als IIR-Filter.
- FIR-Filter weisen eine höhere Filterordnung auf als vergleichbare IIR-Filter und benötigen damit mehr Realisierungsaufwand. Andererseits können FIR-Filter sehr effizient in DSPs implementiert werden.
- Die Zeitverzögerung bzw. die Gruppenlaufzeit ist meistens relativ gross.

Da man fast immer linearphasige FIR-Filter einsetzt, betrachten wir hier nur den Entwurf von linearphasigen FIR-Filtern, insbesondere mit der Fenstermethode, dem Frequenzabtastverfahren und der direkten Approximation im z-Bereich.

IIR-Filter sind rekursive LTD-Systeme und werden meistens als Biquad-Kaskade realisiert. Sie haben die folgenden Vor- und Nachteile:

- + kleiner Realisierungsaufwand bzw. kleine Filterordnung dank der Selektivität der Pole.
- + kleine Zeitverzögerungen.
- kein linearer Phasengang für kausale Filter.
- deutlich mehr Probleme mit Quantisierungseffekten (Grenzyklen, Rundungsrauschen) und dem Abklingen von Störungen im Vergleich zu den FIR-Filtern.

Digitalfilter werden heute fast ausschliesslich mit viel Computerunterstützung entworfen. Deshalb werden auch ein paar Matlab-Programmbeispiele angefügt.

5.2. Filterspezifikation

Der Amplitudengang eines Digitalfilters wird oft mit Hilfe eines Stempel-Matrizen-Schemas spezifiziert, siehe Abbildung 5-1.

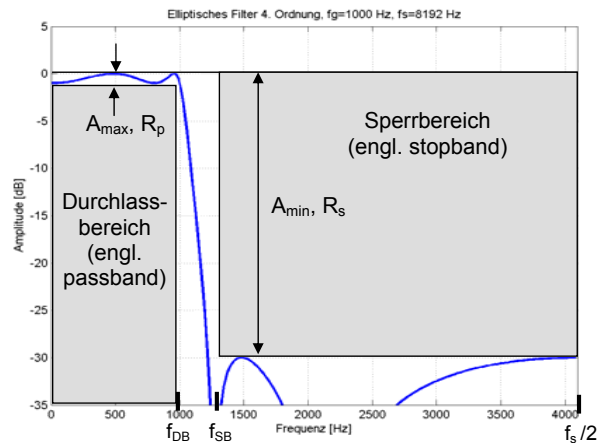


Abbildung 5-1: Filterspezifikation mit Stempel-Matrizen-Schema.

Der Durchlassbereich (engl. passband) eines TP- oder HP-Filters wird normalerweise spezifiziert

- durch die maximal tolerierbare Dämpfung A_{\max} bzw. den Rippel R_p gegenüber einer bestimmten Referenz (normalerweise 0 dB)
- und durch die Eckfrequenz f_{DB} .

Der Sperrbereich (engl. stopband) eines TP- oder HP-Filters wird normalerweise spezifiziert

- durch die minimal erforderliche Dämpfung A_{\min} bzw. den Rippel R_s gegenüber einer bestimmten Referenz (normalerweise 0 dB)
- und durch die Eckfrequenz f_{SB} .

Beispiel

Der in Abbildung 5-1 dargestellte Amplitudengang erfüllt das Stempel-Matrizen-Schema mit den Vorgaben $f_{DB} = 1$ kHz, $f_{SB} = 1.3$ kHz, $A_{\max} = R_p = 1$ dB und $A_{\min} = R_s = 30$ dB.

Die Filterordnung und der Realisierungsaufwand hängen hauptsächlich von der Steilheit der Filterflanke zwischen dem Durchlass- und dem Sperrbereich ab.

Bandpässe bzw. Bandsperren werden analog spezifiziert, wobei der Durchlass- bzw. der Sperrbereich mit 2 Eckfrequenzen am unteren und oberen Rand spezifiziert werden muss.

Der Phasengang wird normalerweise durch den Grad der Linearität (im Durchlassbereich) und/oder durch die maximale Zeitverzögerung spezifiziert.

Wenn die Linearität des Phasengangs wirklich wichtig ist und die Zeitverzögerung nicht allzu klein sein muss, wählt man normalerweise ein FIR-Filter.

5.3. FIR-Filter

5.3.1. Typisierung

Ein kausales FIR-Filter der Ordnung N besitzt $N+1$ Filterkoeffizienten b_0, \dots, b_N .

Man kann zeigen, dass ein FIR-Filter einen vollständig linearen Phasengang aufweist, wenn die Filterkoeffizienten entweder symmetrisch oder anti-symmetrisch sind, d.h. wenn

$$b_n = b_{N-n} \quad \text{oder} \quad b_n = -b_{N-n} \quad (5.1)$$

für $n=0, \dots, N$. Aber auch der Phasengang eines linearphasigen FIR-Filters besitzt Sprungstellen bei den Nullstellen des Amplitudengangs, was aber keine Bedeutung hat, weil die betreffenden Frequenzkomponenten ja vollständig unterdrückt werden.

Die Zeitverzögerung bzw. die Gruppenlaufzeit eines linearphasigen FIR-Filters ist für alle Frequenzkomponenten gleich und beträgt

$$\Delta = \frac{N}{2} \cdot T_s. \quad (5.2)$$

Beispiel:

Ein FIR-Filter der Ordnung $N=1$ mit symmetrischen Filterkoeffizienten besitzt die Übertragungsfunktion $H(z)=b_0(1+z^{-1})$ und damit den Frequenzgang

$$H(f) = b_0(1 + e^{-j2\pi f T_s}) = \frac{2b_0}{2} \cdot (e^{j\pi f T_s} + e^{-j\pi f T_s}) \cdot e^{-j\pi f T_s} = 2b_0 \cdot \cos(\pi f T_s) \cdot e^{-j\pi f T_s}. \quad (5.3)$$

Im der 1. Nyquist-Zone $-f_s/2 \leq f \leq f_s/2$ ist der Cosinus-Term nicht negativ. Für $b_0 > 0$ ist der Phasengang $\varphi(f) = -\pi f T_s$ und damit linear in der Frequenz f . Alle Frequenzkomponenten erleiden die gleiche Zeitverzögerung $\Delta = -\varphi(f)/(2\pi f) = T_s/2$.

Man kann zeigen, dass der Frequenzgang je nach Art der Koeffizientensymmetrie und je nachdem, ob die Filterordnung N gerade und ungerade ist, die in Tabelle 5-1. aufgeführten Restriktionen aufweist.

Typ	Symmetrie	Ordnung N	$H(f=0)$	$H(f=f_s/2)$
1	$b_n = b_{N-n}$	gerade	beliebig	beliebig
2	$b_n = b_{N-n}$	ungerade	beliebig	Nullstelle
3	$b_n = -b_{N-n}$	gerade	Nullstelle	Nullstelle
4	$b_n = -b_{N-n}$	ungerade	Nullstelle	beliebig

Tabelle 5-1: 4 Typen linearphasiger FIR-Filter und Restriktionen im Frequenzgang.

Mit einem FIR-Filter vom Typ 3 mit antisymmetrischen Filterkoeffizienten und gerader Filterordnung N kann man also nur Bandpässe realisieren, wegen den Nullstellen bei $f=0$ bzw. $f=f_s/2$ aber keine Tiefpässe, Hochpässe oder Bandsperren.

Beispiel (Fortsetzung)

Das oben betrachtete FIR-Filter ist vom Typ 2 und besitzt eine Nullstelle bei $f=f_s/2$, weil der Cosinus-Term in Gleichung (5.3) für $f=f_s/2$ Null wird.

5.3.2. Entwurf mit der Fenstermethode (Windowing)

Das Ziel von jedem Filterentwurf ist die Bestimmung der Filterkoeffizienten, so dass die resultierende Impulsantwort bzw. der resultierende Frequenzgang den gewünschten Verlauf aufweist.

Bei einem FIR-Filter stimmen die Filterkoeffizienten b_n aber direkt mit der Impulsantwort $h[n]$ überein. Es liegt deshalb nahe, FIR-Filter wie folgt zu entwerfen:

1. *Abtasten einer analogen (Referenz-) Stossantwort $h(t)$.*

Die resultierende, digitale Impulsantwort ist normalerweise unendlich lang, d.h.

$$h_d[n] = T_s \cdot h(t=nT_s), \quad -\infty \leq n \leq \infty. \quad (5.4)$$

Die Normierung mit T_s ist hier erforderlich, weil das Spektrum des abgetasteten Signals im Nyquistband mit dem Spektrum des analogen Signals übereinstimmen soll, Gleichung (2.5) aber einen Faktor $1/T_s$ beinhaltet.

2. *Ausschneiden des relevanten Anteils von $h_d[n]$, der einen grossen Anteil der Signalleistung enthält, bzw. Abschneiden der kleinen $h_d[n]$ -Werte.*

Wir betrachten hier nur FIR-Filter mit gerader Ordnung N bzw. ungerader Länge $N+1$. Die Fenstermethode kann aber auch für ungerade N modifiziert werden.

Für die abgeschnittene Impulsantwort $h_c[n]$ der Länge $N+1$ gilt:

$$h_c[n] = w[n] \cdot h_d[n], \quad -N/2 \leq n \leq N/2. \quad (5.5)$$

Die Fensterfunktion $w[n]$ weist nur für diskrete Zeitpunkte n im Bereich $-N/2 \leq n \leq N/2$ von Null verschiedene Werte auf. Der relevante Anteil von $h_d[n]$ kann mit einem rechteckigen Fenster $w[n]=1$, $-N/2 \leq n \leq N/2$, „hart“ oder mit verschiedenen auslaufenden Fenstern $w[n]$ „weich“ ausgeschnitten werden.

3. *FIR-Filter durch Zeitverschiebung der abgeschnittenen Impulsantwort kausal machen.*

Für die resultierende Impulsantwort bzw. die Filterkoeffizienten gilt

$$b_n = h[n] = h_c[n-N/2]. \quad (5.6)$$

Die FIR-Filterkoeffizienten sind symmetrisch oder anti-symmetrisch bezüglich $h[N/2]$, wenn die analoge Stossantwort $h(t)$ eine gerade oder eine ungerade Funktion ist und die Fensterfunktion $w[n]$ gerade ist.

Das Bestimmen der analogen Stossantwort $h(t)$ bzw. der gewünschten digitalen Impulsantwort $h_d[n]$ im 1. Schritt oben ist im Allgemeinen nicht einfach. Für gewisse Referenzfilter liegt $h(t)$ aber analytisch vor. Die mit f_s abgetastete und mit T_s normierte Stossantwort $h_d[n]$ eines idealen Tiefpasses mit der Grenzfrequenz f_{DB} ist z.B. gegeben durch

$$h_d[n] = \sin[n\pi \cdot f_{DB}/(f_s/2)] / (n\pi), \quad -\infty \leq n \leq \infty, \quad (5.7)$$

und weist einen $\sin(x)/x$ -förmigen Verlauf auf, der umso schneller abklingt, je breiter der Durchlassbereich ist. Mit Hilfe der Fenstermethode können nun ausgehend von Gleichung (5.7) beliebige, linearphasige FIR-Tiefpassfilter entworfen werden.

Beispiel:

In Abbildung 5-2 ist der Entwurf eines FIR-Tiefpassfilters der Ordnung $N=50$ mit der Grenzfrequenz $f_{DB} = 2$ kHz dargestellt, wobei die Abtastfrequenz $f_s = 8$ kHz ist.

Links oben sind die mit einem Rechteckfenster ausgeschnittenen $N+1=51$ Werte der Impulsantwort $h[n]$, $n=0, \dots, 50$, abgebildet. Sie sind symmetrisch zu $h[25]$ und klingen rasch ab.

Links unten ist der resultierende Amplitudengang im Frequenzbereich $[0, f_s/2]$ dargestellt. Er weist im Übergangsbereich ein grosses Überschwingen auf, das als Gibbs'sches Phänomen bezeichnet wird.

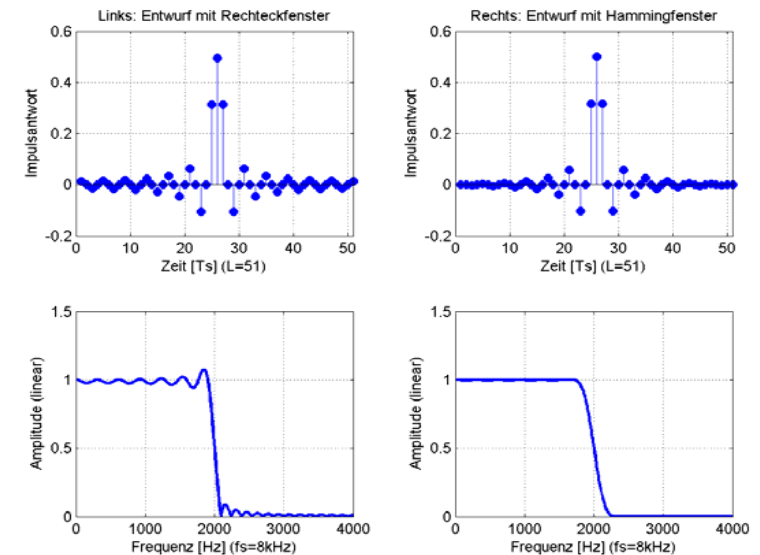


Abbildung 5-2: FIR-Filterentwurf mit der Fenstermethode.

Das Gibbs'sche Phänomen verschwindet nicht, selbst wenn die Filterlänge N viel grösser gemacht würde. Es entsteht wegen dem abrupten Abschneiden der Impulsantwort. Das erkennt man am einfachsten im Spektrum.

Die Multiplikation der Impulsantwort $h_d[n]$ mit dem Fenster $w[n]$ in Gleichung (5.5) ist äquivalent zur Faltung des (gewünschten) Frequenzgangs $H_d(f)$ mit der Fouriertransformierten $W(f)$ des Fensters. Das Spektrum $W(f)$ eines rechteckigen Fensters weist einen $\sin(x)/x$ -förmigen Verlauf auf. Faltet man nun $W(f)$ mit dem gewünschten rechteckigen Frequenzgang $H_d(f)$, so resultiert ein Überschwingen im Übergang zwischen Durchlass- und Sperrbereich.

Das Gibbs'sche Phänomen kann mit „weichem“ Abschneiden der Impulsantwort bzw. mit auslaufenden Fenstern $w[n]$ verringert werden. In Abbildung 5-3 sind 4 verschiedene, kausale Fenster mit 51 Abtastwerten dargestellt. Sie sind wie folgt definiert:

Rechteckfenster: $w[n] = 1, \quad 0 \leq n \leq N$

Hamming-Fenster: $w[n] = 0.54 - 0.46 \cdot \cos(2\pi n/N), \quad 0 \leq n \leq N$

Hanning-Fenster: $w[n] = 0.5 \cdot [1 - \cos(2\pi n/N)], \quad 0 \leq n \leq N$

Blackmann-Fenster: $w[n] = 0.42 - 0.5 \cdot \cos(2\pi n/N) + 0.08 \cdot \cos(4\pi n/N), \quad 0 \leq n \leq N$

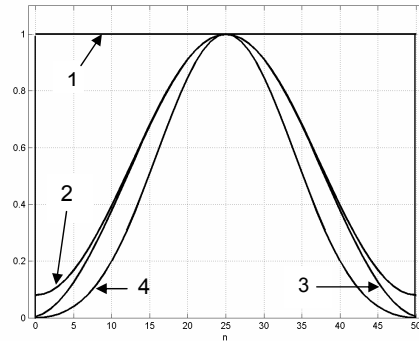


Abbildung 5-3: Rechteck- (1), Hamming- (2), Hanning- (3) und Blackman-Fenster (4).

Es gibt noch viele weitere Fensterfunktionen $w[n]$, siehe z.B. die Fensterfunktionen in der Matlab Signal Processing Toolbox, die mit dem Befehl `help signal` aufgelistet werden können.

Das parametrierbare Kaiser-Fenster wird in einem separaten Unterkapitel weiter hinten in diesem Dokument behandelt.

Beispiel (Fortsetzung)

Rechts oben in Abbildung 5-2 sind die mit einem auslaufenden Hamming-Fenster geschnittenen $N+1=51$ Werte der Impulsantwort $h[n]$, $n=0, \dots, 50$, abgebildet.

Rechts unten ist der resultierende Amplitudengang dargestellt. Das Überspringen ist dank dem weichen Abschneiden verschwunden und die Dämpfung im Sperrbereich ist deutlich grösser geworden, allerdings auf Kosten der Filtersteilheit.

In Abbildung 5-4 sind die Betragsspektren $|W(f)|$ der in Abbildung 5-3 abgebildeten Fenster der Länge 51 abgebildet. Der Parameter A bezeichnet das dB-Grössenverhältnis zwischen der „Hauptkeule“ und der grössten „Nebenkeule“. Wenn man die oben erwähnte Faltung ausführt, sieht man, dass die Sperrbereichsdämpfung des resultierenden FIR-Filters umso grösser ist, je kleiner A ist. In Abbildung 5-4 schneidet das Rechteckfenster in dieser Hinsicht schlechter ab als das Hamming-Fenster.

In Abbildung 5-4 ist auch die Breite Δf der „Hauptkeule“ in Funktion der Abtastfrequenz f_s und der Filterlänge $L=N+1$ dargestellt. Wenn man die oben erwähnte Faltung ausführt, sieht man, dass die Filtersteilheit umso grösser ist, je kleiner Δf bzw. schmaler die „Hauptkeule“ ist. In Abbildung 5-4 schneidet das Rechteckfenster in dieser Hinsicht besser ab als das Hamming-Fenster.

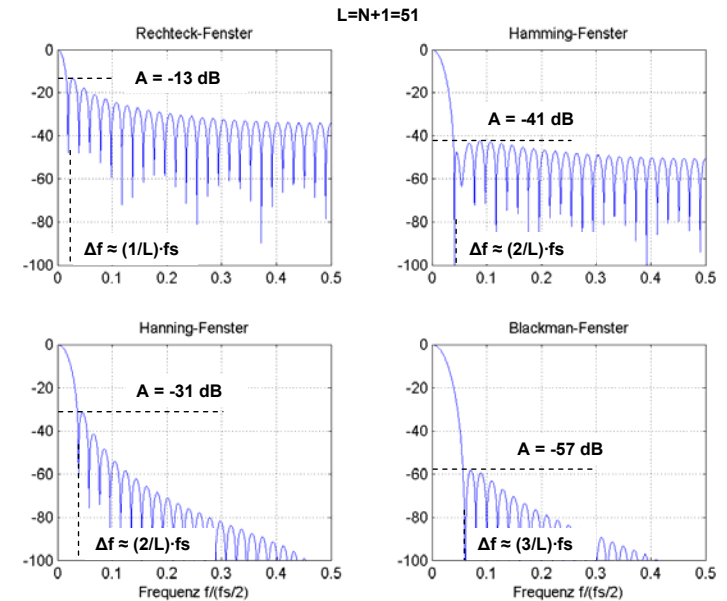


Abbildung 5-4: Betragsspektren einiger Fenster der Länge $L=N+1=51$.

Korrigenda: Die Frequenzachse ist f/f_s und nicht $f/(f_s/2)$

Mit Gleichung (5.7) lassen sich nur Tiefpass-Filter realisieren. Es stellt sich nun die Frage, wie die anderen Filtertypen realisiert werden können. Zu diesem Zweck sucht man Tiefpass-Bandpass, Bandpass-Bandsperr und Tiefpass-Hochpass-Transformationen, die die lineare Phase erhalten.

5.3.3 Transformation: Tiefpass > Hochpass, Bandpass, Bandsperre

Für den Entwurf von Tiefpass-Filtern gibt es viele ausgereifte Methoden.

Mit Hilfe von mathematischen Transformationen können diese Methoden auch für den Entwurf von linearphasigen Hochpässen, Bandpässen und -sperren verwendet werden.

Tiefpass-Bandpass-Transformation

Mit Hilfe der Frequenzverschiebungseigenschaft der Fouriertransformation (siehe Anhang zu Kapitel 2) kann man zeigen, dass die beiden folgenden Transformationen einen Tiefpass mit der Grenzfrequenz f_{DB} in einen Bandpass mit der Mittenfrequenz f_0 und der unteren bzw. der oberen Grenzfrequenz $f_0 \pm f_{DB}$ transformieren.

$$\begin{aligned} \text{Typ 1 und 2: } b_{BP}[n] &= 2 \cdot \cos(\omega_0 \cdot nT_s) \cdot b_{TP}[n] \\ \text{Typ 3 und 4: } b_{BP}[n] &= 2 \cdot \sin(\omega_0 \cdot nT_s) \cdot b_{TP}[n] \end{aligned} \quad (5.8)$$

Tiefpass-Hochpass-Transformation

Wenn man in (5.8) $f_0 = f_s/2$ setzt, erhält man die Tiefpass-Hochpass-Transformation

$$b_{HP}[n] = (-1)^n \cdot b_{TP}[n]. \quad (5.9)$$

Bandpass-Bandsperre-Transformation

Bandsperren können als Komplementärfilter zu einem Bandpassfilter realisiert werden, d.h.

$$\begin{aligned} H_{BS}(z) + H_{BP}(z) &= z^{-N/2} \\ b_{BS}[n] &= \delta[n - N/2] - b_{BP}[n] \end{aligned} \quad (5.10)$$

Die Fenstermethode zum Entwurf von FIR-Filtern kann vollständig mit Computerunterstützung vorgenommen werden, siehe z.B. die Funktion `fir1()` in der Matlab Signal Processing Toolbox.

Beispiel

Mit dem Matlab-Befehl `b=fir1(50,0.5,rectwin(51))`; kann man ein FIR-TP-Filter der Ordnung $N=50$ mit der Fenstermethode entwerfen, wobei ein Rechteckfenster der Länge 51 verwendet wird (Default ist das Hamming-Fenster).

Mit dem Matlab-Befehl `b=fir1(9,0.5,'high')` wird mit der Fenstermethode ein FIR-HP-Filter der Ordnung $N=10$ generiert, weil kein entsprechendes Filter mit der Ordnung $N=9$ existiert, siehe Tabelle 5-1.

5.3.4 Kaiser-Window

Beim Entwurf eines Filters mit der Fenstermethode muss immer ein Kompromiss zwischen Filtersteilheit und Dämpfung im Sperrbereich gefunden werden.

Neben den oft eingesetzten Fensterfunktionen „Rechteck“, „Hamming“, „Hanning“ und „Blackman“ ist das „Kaiser“-Fenster eine interessante Alternative.

Beim Kaiser-Fenster kann mit dem Faktor β der Kompromiss zwischen Breite der Hauptkeule (= Filter-Steilheit) und Höhe der Nebenkeulen (= Dämpfung im Sperrbereich) eingestellt werden.

Wie bei allen Filtern können durch Erhöhen der Filterordnung sowohl die Filter-Steilheit, als auch die Sperrbereichs-Dämpfung verbessert werden. Proportional zur Filterordnung steigen aber Rechenleistung und Signal-Verzögerung.

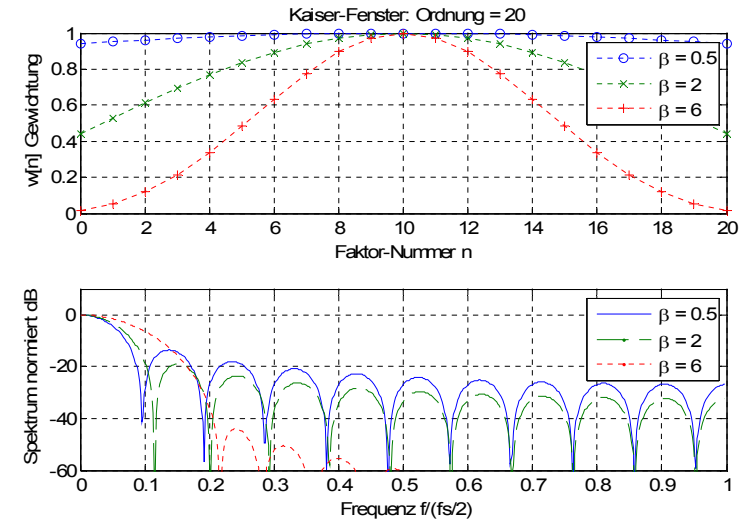


Abbildung 5-5: Kaiser-Window

Gewichtungs-Faktoren und normiertes Spektrum für verschiedene β

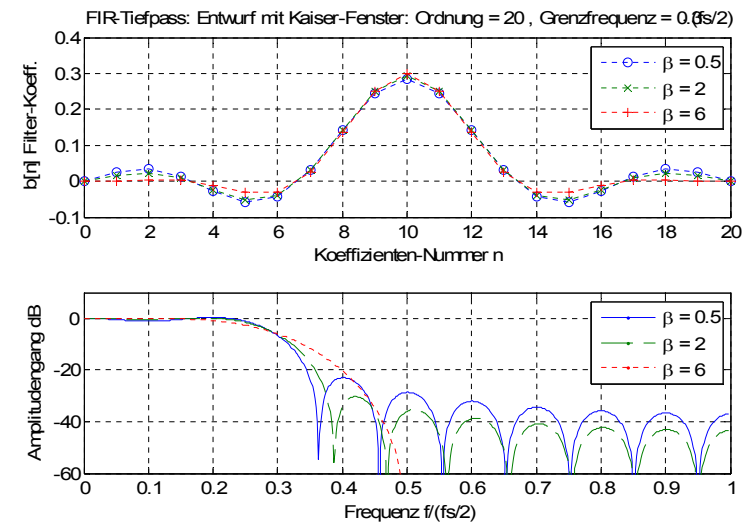


Abbildung 5-6: FIR-Tiefpass-Filter: Filter-Koeffizienten und Amplitudengang entworfen durch Gewichtung mit Kaiser-Windows mit verschiedenen β

5.3.5. Fensterentwurf spezieller FIR-Filter (fakultativ)

Bis jetzt sind wir immer von der Impulsantwort des Referenz-TP-Filters in Gleichung (5.7) ausgegangen. Es gibt aber noch andere, interessante Referenz-Filter. Drei davon sollen hier kurz vorgestellt werden.

Die mit f_s abgetastete und mit T_s normierte Stossantwort $h_d[n]$ eines idealen Differentiators $H_d(f) = j2\pi f$ ist gegeben durch

$$h_d[n] = \cos(n\pi)/n - \sin(n\pi)/(n^2\pi), \quad -\infty \leq n \leq \infty, \quad (5.11)$$

wobei $h_d[0] = 0$ ist. Die gewünschte Impulsantwort $h_d[n]$ ist eine ungerade Funktion bzw. besitzt antisymmetrische Werte bezüglich des Nullpunkts. Mit Windowing resultiert also ein linearphasiges FIR-Filter mit frequenzunabhängiger Gruppenlaufzeit.

Beispiel

In Abbildung 5-7 sind die antisymmetrische Impulsantwort und der Amplitudengang eines FIR-Differentiators 10. Ordnung dargestellt, der mit Gleichung (5.11) und einem Hamming-Fenster entworfen worden ist.

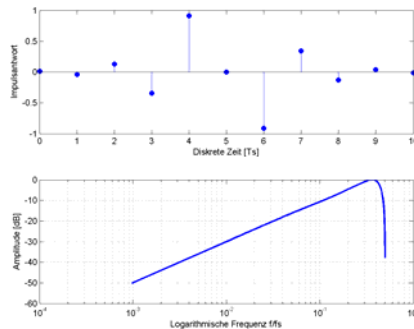


Abbildung 5-7: Impulsantwort und Amplitudengang eines FIR-Differentiators.

Bei der Differentiation hat man oft das Problem, dass man das höherfrequente Rauschen ungewollt verstärkt. Man kann nun linearphasige FIR-Filter entwerfen, die die tieferen Frequenzen differenzieren, die höheren Frequenzen aber gleichzeitig unterdrücken (TP-Filterung), siehe z.B. Matlab Filter Design Tool `fdatool`.

In der Nachrichtentechnik werden breitbandige 90° -Phasenschieber bzw. Hilbert-Filter mit dem folgenden Frequenzgang eingesetzt,

$$H_d(f) = \begin{cases} j & -f_s/2 \leq f \leq 0 \\ -j & 0 < f \leq f_s/2 \end{cases} \quad (5.12)$$

Die mit f_s abgetastete und mit T_s normierte Impulsantwort $h_d[n]$ des idealen Hilbert-Filters ist gegeben durch

$$h_d[n] = [1 - \cos(n\pi)] / n\pi, \quad -\infty \leq n \leq \infty, \quad (5.13)$$

wobei $h_d[0] = 0$ ist. Die gewünschte Impulsantwort $h_d[n]$ ist eine ungerade Funktion bzw. besitzt antisymmetrische Werte bezüglich des Nullpunkts.

Beispiel

In Abbildung 5-8 sind der Amplitudengang und der Phasengang eines FIR-Hilbert-Filters 10. Ordnung dargestellt, das mit Gleichung (5.13) und einem Rechteckfenster entworfen worden ist.

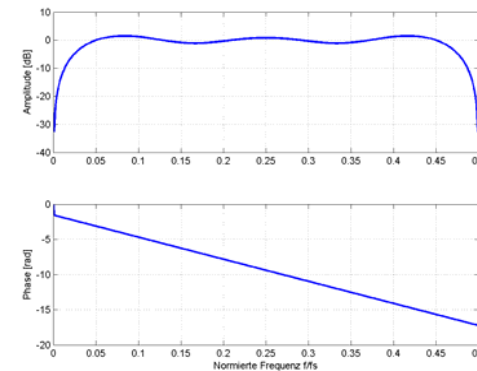


Abbildung 5-8: Frequenzgang eines FIR-Hilbert-Filters.

Der Amplitudengang ist für fast alle Frequenzen in der 1. Nyquistzone ein Allpass. Der Allpass-Charakter und das leichte Überschwngen könnten mit einer grösseren Filterordnung N und einem auslaufenden Fenster noch weiter optimiert werden.

Der Phasengang weist bei $f=0$ wie gewünscht einen Phasensprung auf $-\pi/2$ auf. Danach fällt er aber wegen der Zeitverschiebung im 3. Schritt der Fenstermethode linear ab. Es resultiert ein breitbandiger 90° -Phasenschieber mit einer „kleinen“ Zeitverzögerung.

In der Signalübertragung werden oft raised-cosine-Pulse eingesetzt. Mit dem Matlab-Befehl `b = firrcos(N,R/2,roll,fs,'rolloff')`; kann man linearphasige raised-cosine-FIR-Filter der Ordnung N mit TP-Spektren entwerfen, die einen cosinusförmigen Übergangsbereich $R/2 \cdot (1 \pm r)$ aufweisen, wobei R die Datenrate und r der sogenannte Rolloff-Faktor bezeichnen. Per Default verwendet `firrcos()` ein Rechteckfenster.

Beispiel:

In Abbildung 5-9 sind die symmetrische Impulsantwort und der Amplitudengang eines Raised-Cosine-FIR-Filters 20. Ordnung dargestellt, wobei die Datenrate $R=2$ kb/s, der Rolloff-Faktor $r=0.5$ und die Abtastfrequenz $f_s=8$ kHz gewählt worden sind.

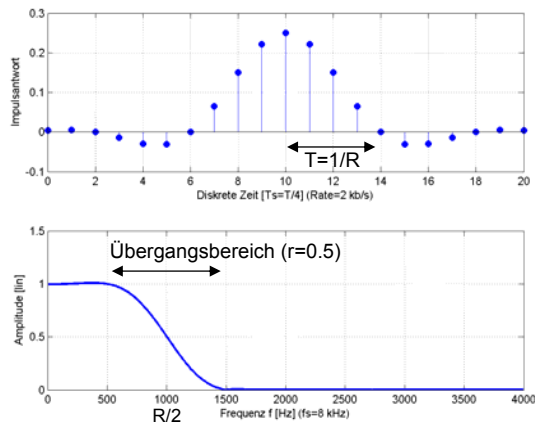


Abbildung 5-9: Raised-Cosine FIR-Filter.

5.3.6. Entwurf mit Frequenzabtastung (Frequency Sampling)

Bei dieser Entwurfsmethode werden $N+1$ äquidistante Abtastpunkte des gewünschten Frequenzgangs $H(f)$ vorgegeben. Mit Hilfe der inversen diskreten Fouriertransformation IDFT bzw. IFFT kann dann die Impulsantwort $h[n]$, $n=0, \dots, N$, des FIR-Filters bestimmt werden. Der Frequenzgang des FIR-Filters hält die Vorgabepunkte natürlich genau ein, dazwischen kann es aber „Überschwinger“ geben, vor allem im Übergang zwischen dem Durchlass- und dem Sperrbereich.

Mit dem Frequenzabtastverfahren können auch „beliebig“ komplexe Impulsantworten $h_d[n]$ generiert und als Input für die Fenstermethode verwendet werden.

5.3.7. Entwurf im z-Bereich

Es gibt einige iterative Optimierungsverfahren, die nur noch mit Computerunterstützung durchgeführt werden können.

Am bekanntesten ist wohl der Remez-Algorithmus von Parks-McClellan. Dieser Algorithmus minimiert die maximale Abweichung zwischen dem Amplitudengang des Filters und der Vorgabe (gegeben in Form eines [Multiband] Stempel-Matrizen-Schemas). Die resultierenden FIR-Filter weisen gleichmässige Rippel im Durchlass- und im Sperrbereich auf. Sie werden deshalb auch Equiripple FIR-Filter genannt. Bei gegebener minimaler Sperrbereichsdämpfung ist die Filterordnung kleiner als bei allen anderen FIR-Filter-Entwurfsmethoden. Die betreffende Funktion in der Matlab Signal Processing Toolbox heisst `firpm()`.

Andere Optimierungsverfahren minimieren die mittlere quadratische Abweichung zwischen dem Amplitudengang des Filters und der Vorgabe, siehe z.B. die Funktion `firls()` in der Matlab Signal Processing Toolbox.

5.4. IIR-Filter

5.4.1. Entwurf mit analogen Prototyp-Filtern

Die Approximation von idealen „Rechteck“-Filtern ist im Analogen auf verschiedene Arten gelöst worden, die alle in irgendeiner Form optimal sind. Von diesen Methoden will und kann man auch im Digitalen profitieren.

Beim klassischen IIR-Filterentwurf geht man deshalb von einem analogen Prototyp-TP-Filter aus, nimmt allenfalls eine TP - HP / BP / BS Transformation vor und transformiert dann das resultierende Filter ins Digitale. Die am häufigsten eingesetzte analog-digital- bzw. sz-Transformation ist die bilineare Transformation.

In Abbildung 5-10 sind die Amplitudengänge von vier IIR-Filtern 4. Ordnung dargestellt, die mit Hilfe von verschiedenen Prototyp-Filtern entworfen worden sind und folgende Anforderungen erfüllen:

- Abtastfrequenz $f_s = 8192$ Hz
- Durchlassbereich mit $f_{DB} = 1$ kHz, $A_{max} = R_p = 1$ dB (Butterworth-Filter 3 dB)
- Sperrbereich mit $A_{min} = R_s = 30$ dB

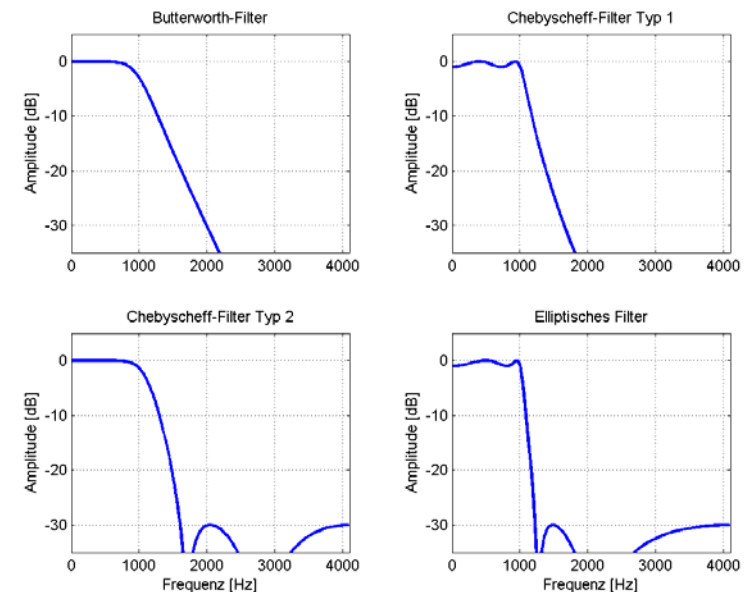


Abbildung 5-10: Amplitudengang von analogen Prototyp-Filtern.

Das elliptische Filter (Cauer-Filter) ist am steilsten und approximiert den idealen Tiefpass am besten. Es weist aber gleichmässige Rippel im Durchlass- und im Sperrbereich auf. Letztere sind das Resultat von Nullstellen der Übertragungsfunktion $H(z)$ auf dem Einheitskreis.

	Butterworth-Filter	Chebyscheff-Filter, Typ 1	Chebyscheff-Filter, Typ 2	Elliptisches-Filter	Bessel-Filter
Durchlassbereich	monoton	Rippel	monoton	Rippel	monoton
Sperrbereich	monoton	monoton	Rippel	Rippel	monoton
Steilheit	klein	mittel	mittel	gross	sehr klein
Linearität $\varphi(f)$	gross	mittel	mittel	klein	sehr gross

Tabelle 5-2: Wichtige Eigenschaften der analogen Prototyp-Filter.

Es besteht die Konvention, auch bei BP- und BS-Filtern mit der Filterordnung die Ordnung des Prototyp-TP-Filters zu bezeichnen. Bei der TP-BP / BS Transformation wird allerdings die „eigentliche“ Ordnung verdoppelt. Ein BP-Filter N-ter Ordnung besitzt deshalb $2N+1$ Koeffizienten im Nenner der UTF.

Bilineare Transformation

Ziel der bilinearen Transformation ist es, die eigentliche sz-Transformation $z=e^{sT_s}$ so zu approximieren, dass eine realisierbare bzw. rational gebrochene Übertragungsfunktion $H(z)$ resultiert. Dazu entwickelt man die e-Funktion in eine Reihe und approximiert sie wie folgt:

$$z = e^{sT_s} = \frac{e^{sT_s/2}}{e^{-sT_s/2}} \approx \frac{1+sT_s/2}{1-sT_s/2} \quad (5.14)$$

Durch Auflösen der Gleichung (5.14) nach s erhält man die sz-Transformation

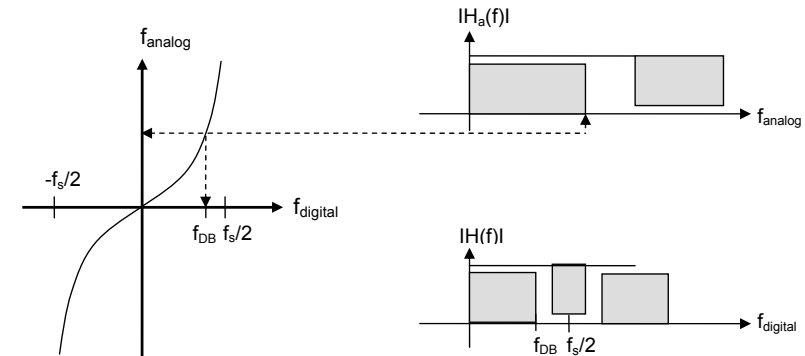
$$s = \frac{2}{T_s} \cdot \frac{z-1}{z+1}. \quad (5.15)$$

Diese Transformation wird *bilineare* Transformation genannt, weil Zähler und Nenner im obigen Ausdruck je lineare Funktionen sind. Die Transformation selbst ist nicht linear.

Die Eigenschaften der bilinearen Transformation erkennt man, wenn man die Frequenz-Transformation analysiert, d.h. $s=j2\pi f_{\text{analog}}$ und $z=\exp(j2\pi f_{\text{digital}}T_s)$ in Gleichung (5.15) einsetzt. Dann folgt

$$j2\pi f_{\text{analog}} = j \frac{2}{T_s} \cdot \tan(\pi f_{\text{digital}} T_s). \quad (5.16)$$

Die resultierende Frequenztransformation ist in Abbildung 5-11 dargestellt. Die ganze analoge Frequenzachse wird nur einmal in die 1. Nyquist-Zone $-f_s/2 \leq f \leq f_s/2$ abgebildet.

**Abbildung 5-11:** Bilineare Frequenztransformation.

Bei der bilinearen Transformation entsteht kein Aliasing, dafür eine Frequenzstauchung. Die Eckfrequenzen des Durchlass- und des Sperrbereichs des analogen Prototyp-Filters müssen deshalb gemäss Gleichung (5.16) vorverzerrt werden. Diese Vorverzerrung wird im Englischen *prewarping* genannt.

Bilinear transformierte IIR-Filter können vollständig mit Computerunterstützung entworfen werden, siehe z.B. die Funktion `bilinear()` in der Matlab Signal Processing Toolbox.

Matlab-Beispiele

In der Matlab Signal Processing Toolbox stehen komplette Entwurfsfunktionen bereit, die alle die bilineare Transformation enthalten. Alle Eckfrequenzen müssen auf $f_s/2$ normiert werden! Auch das GUI `fdatool` steht zur Verfügung.

Wenn N die Filterordnung, f_{DB} die 3dB-Eckfrequenz des Durchlassbereichs und f_s die Abtastfrequenz bezeichnen, können die Koeffizienten des entsprechenden Butterworth-TP- bzw. -HP-Filters einfach mit

$$\begin{aligned} [b,a] &= \text{butter}(N, f_{DB}/(f_s/2)); \\ \text{bzw. mit } [b,a] &= \text{butter}(N, f_{DB}/(f_s/2), 'high'); \end{aligned}$$

bestimmt werden. Die Entwurfsfunktionen für die anderen, oben erwähnten Standard-Prototypen lauten `ellip()`, `cheby1()` und `cheby2()`. Digitale Besselfilter müssen mit Hilfe von Teilfunktionen entworfen werden.

Matlab und viele Bücher arbeiten mit der normierten Kreisfrequenz Ω im Ausdruck

$$z = e^{j\Omega} = e^{j2\pi f T_s}. \quad (5.17)$$

Die Umformung

$$\Omega = \frac{\pi f}{f_s/2} \quad (5.18)$$

verdeutlicht, dass Ω im Bereich $[0, \pi]$ liegt, wenn f im Bereich $[0, f_s/2]$ variiert wird. Sehr nützlich sind auch die Funktionen zur Ermittlung der minimalen Filterordnung N , die zur Realisierung der Filterspezifikation $\{f_{DB}, f_{SB}, R_p, R_s\}$ erforderlich ist. Die minimale Ordnung eines Butterworth-Filters kann z.B. mit


```
[N, Wn] = buttord(fDB/(fs/2), fSB/(fs/2), Rp, Rs);
```

bestimmt werden. Die anderen Funktionen zur Ermittlung der minimalen Filterordnung lauten `ellipord()`, `cheblord()` und `cheb2ord()`.

5.4.2. Direkter Entwurf im Diskreten

IIR-Filter können mit CAD-Unterstützung auch direkt im Diskreten entworfen werden.

Mit den Algorithmen von Fletcher-Powell und Yule-Walker können die Filterkoeffizienten einer Übertragungsfunktion $H(z)$ so optimiert bzw. variiert werden, dass der resultierende Amplitudengang mit einer Vorgabe möglichst gut übereinstimmt (least-square Verfahren).

In Abbildung 5-12 ist der Amplitudengang eines Yule-Walker-Filters 10. Ordnung sowie die Vorgabe dargestellt.

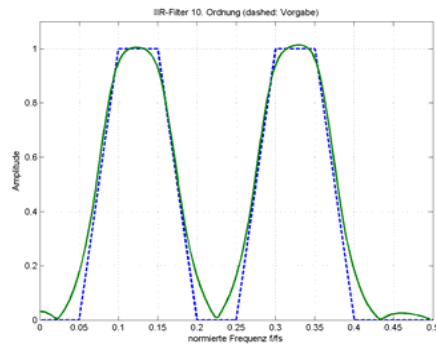


Abbildung 5-12: Amplitudengang eines Yule-Walker-IIR-Filters.

Mit dem direkten Entwurf im z -Bereich können auch nicht-standard-Filtertypen wie z.B. Multi-band-Filter realisiert werden, siehe Abbildung 5-12.

Die entsprechende Funktion in der Matlab Signal Processing Toolbox lautet `yulewalk()`.

5.5. Realisierungsstrukturen

LTD-Systeme können mit einer Differenzgleichung oder einer Übertragungsfunktion eindeutig beschrieben werden. Es gibt aber viele Möglichkeiten, LTD-Systeme zu realisieren bzw. zu implementieren.

Im Idealfall sind die folgenden Realisierungsstrukturen alle gleichwertig. Im Realfall mit endlich genauen Filterkoeffizienten bzw. beschränkten Wortlängen verhalten sie sich aber unterschiedlich.

Im Folgenden werden die Realisierungsstrukturen mit vereinfachten Signalflussgraphen wie in Abbildung 5-13 dargestellt. Ein Zweig mit einem beschrifteten Pfeil stellt eine Multiplikation mit einer Konstanten dar. Fehlt die Beschriftung, ist die Konstante 1. Addierer werden mit einem Plus-Symbol gekennzeichnet und Abzweigpunkte mit einem Punkt. Verzögerungselemente schliesslich werden mit einem Pfeil und dem Symbol z^{-1} dargestellt.

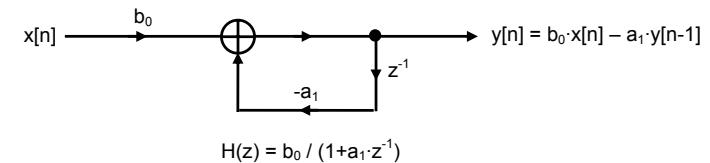


Abbildung 5-13: Beispiel eines Signalflussgraphen.

5.5.1. Direkt-Strukturen

Die Differenzgleichung (4.1) kann einfach mit der in Abbildung 5-14 dargestellten Direktstruktur 1 realisiert werden. Die Direktstruktur 1 wird in der Praxis meistens nur für die Realisierung von FIR-Filtern verwendet.

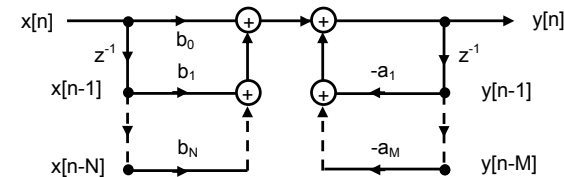


Abbildung 5-14: Direktstruktur 1.

Die beiden kaskadierten Teilsysteme in der Direktstruktur 1 dürfen vertauscht werden, ohne dass sich das Verhalten des Gesamtsystems verändert. Dann liegen aber die beiden Verzögerungsketten nebeneinander und können zu einer Kette zusammengefasst werden.

Die resultierende Struktur in Abbildung 5-15 wird Direktstruktur 2 genannt. Im Vergleich zur Direktstruktur 1 hat sie nur noch die halbe Anzahl Speicherzellen. Die Anzahl Speicherzellen ist minimal, weil sie gleich der Filterordnung ist (ohne Beweis).

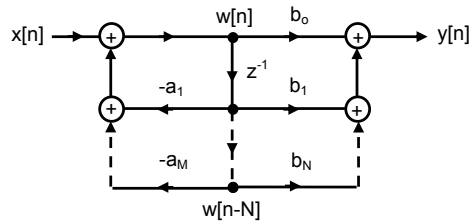


Abbildung 5-15: Direktstruktur 2.

Mit Hilfe der in Abbildung 5-15 eingezeichneten Werte $w[n]$, $w[n-1]$, ..., $w[n-N]$ kann der Ausgangswert $y[n]$ in Funktion des Eingangswerts $x[n]$ wie folgt berechnet werden:

Schritt 0: (zirkulärer) w -Buffer $\{w[n-1], \dots, w[n-N]\}$ mit Nullen initialisieren

Schritt 1: $w[n] = x[n] - a_1 \cdot w[n-1] - \dots - a_M \cdot w[n-M]$

Schritt 2: $y[n] = b_0 \cdot w[n] + \dots + b_N \cdot w[n-N]$ ausgeben

Schritt 3: w -Buffer schieben, $w[n]$ speichern
(effizienter: im Ringbuffer ältesten Wert $w[n-N]$ mit $w[n]$ überschreiben)

Schritt 4: neuen Eingangswert $x[n+1]$ lesen und mit Schritt 1 weiterfahren

FIR-Filter-Strukturen findet man durch Nullsetzen der Feedback-Koeffizienten a_k , $k=1, \dots, M$, in den oben dargestellten Realisierungsstrukturen. In Abbildung 5-16 ist die Realisierung eines FIR-Filters mit der Direktstruktur 1 dargestellt, die häufig verwendet wird.

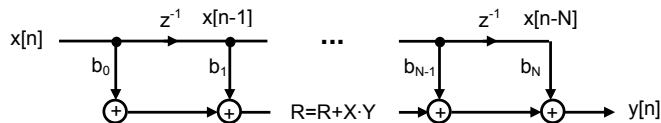


Abbildung 5-16: FIR-Filter-Realisierung in Direktstruktur 1.

FIR-Filter mit mehr als 100 Taps (Abzapfpunkten) sind in der Praxis keine Seltenheit. In Echtzeit-Anwendungen ist es darum wichtig, dass die N Multiply-And-Accumulate (MAC) Operationen $R=R+X \cdot Y$ in Abbildung 5-16 sehr schnell ausgeführt werden können. Digitale Signalprozessoren sind darauf spezialisiert.

Es gibt noch weitere Strukturen für spezielle Realisierungsvarianten von digitalen Systemen bzw. Digitalfiltern, siehe z.B. [2].

5.5.2. Transponierte Direkt-Struktur

Das Transponierungstheorem für (lineare) Signalflussgraphen besagt, siehe [1] oder [2]: „Werden alle Signalflussrichtungen umgekehrt, alle Addierer durch Knoten und alle Knoten durch Addierer ersetzt sowie Ein- und Ausgang vertauscht, dann bleibt die Übertragungsfunktion $H(z)$ unverändert.“

Durch Transposition der Direktstruktur 2 erhält man die in Abbildung 5-17 dargestellte transponierte Struktur 2.

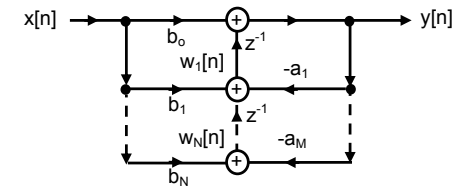


Abbildung 5-17: Transponierte Direktstruktur 2.

Mit Hilfe der in Abbildung 5-17 eingezeichneten Werte $w_1[n]$, $w_2[n]$, ..., $w_N[n]$ kann der Ausgangswert $y[n]$ in Funktion des Eingangswerts $x[n]$ wie folgt berechnet werden (Annahme $M=N$):

$$y[n] = b_0 \cdot x[n] + w_1[n-1]$$

$$w_1[n] = b_1 \cdot x[n] - a_1 \cdot y[n] + w_2[n-1]$$

:

$$w_{N-1}[n] = b_{N-1} \cdot x[n] - a_{M-1} \cdot y[n] + w_N[n-1]$$

$$w_N[n] = b_N \cdot x[n] - a_M \cdot y[n]$$

5.5.3. Biquads (second order sections)

Die in Gleichung (4.28) gegebene Pol-Nullstellen-Darstellung der Übertragungsfunktion $H(z)$ ermöglicht eine weitere Realisierungsstruktur. Die komplexen Pole und Nullstellen treten immer in konjugiert komplexen Paaren auf. Diese Paare können so zusammengefasst werden, dass eine Kaskade von Biquads 2. Ordnung mit reellen Teil-Übertragungsfunktionen $H_k(z)$, $k=1, \dots, L$, entsteht, d.h.

$$H(z) = K \cdot \frac{(z - z_1) \cdot (z - z_1^*)}{(z - p_1) \cdot (z - p_1^*)} \dots \frac{(z - z_L) \cdot (z - z_L^*)}{(z - p_L) \cdot (z - p_L^*)} = K \cdot H_1(z) \dots H_L(z). \quad (5.19)$$

K ist ein beliebiger Verstärkungs- bzw. Abschwächungsfaktor. Bei der Realisierung von IIR-Filtern ungerader Ordnung degeneriert ein Biquad zu einem System 1. Ordnung.

Die einzelnen Biquads werden wieder in einer Direktstruktur realisiert, siehe Abbildung 5-18.

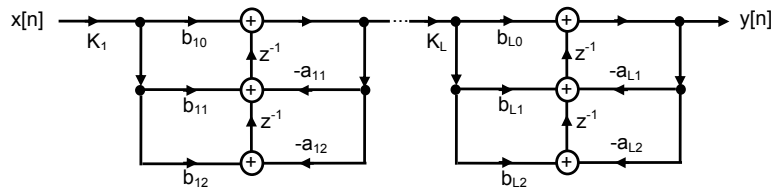


Abbildung 5-18: Kaskadierung von Biquads in transponierter Direktform 2.

Kaskaden- und Parallelstrukturen verhalten sich in Bezug auf Quantisierungseffekte (siehe unten) günstiger als Direktstrukturen. Die Quantisierung der Filterkoeffizienten beeinflusst nicht alle Pole und Nullstellen gleichzeitig wie bei der Direktstruktur, sondern immer nur die Paare eines Biquads, und hat damit einen kleineren Einfluss auf den Frequenzgang.

Es gibt zahlreiche Möglichkeiten der Pol-Nullstellenpaarung für die Biquads einer Kaskade. In der Praxis wird normalerweise wie folgt verfahren:

- Der letzte Biquad enthält das komplexe Polpaar, das am nächsten beim Einheitskreis liegt, und das dazu nächstgelegene konjugiert komplexe Nullstellenpaar.
- Die übrig gebliebenen Pole und Nullstellen werden nach der Regel a) kombiniert. Mit Regel a) wird die grösste Überhöhung im Frequenzgang mit einer Abschwächung ganz in der Nähe der Überhöhung kompensiert. Damit wird der Dynamikbereich des Biquads eingeschränkt, was vorteilhaft ist, wenn die Signale nur mit endlicher Genauigkeit repräsentiert werden können.

5.5.4. Parallel-Strukturen

Die rationale Übertragungsfunktion $H(z)$ in Gleichung (4.18) kann auch als Summe von Partialbrüchen dargestellt werden, siehe Gleichung (5.20). Wenn man wieder die konjugiert komplexen Pole zusammenfasst, kann die Übertragungsfunktion $H(z)$ als Summe von reellen Biquads dargestellt und in einer Parallelstruktur implementiert werden.

$$H(z) = \sum_{k=1}^L \frac{A_k}{z - p_k} + \frac{A_k^*}{z - p_k^*} = \sum_{k=1}^L H_k(z) \quad (5.20)$$

5.5.5. Lattice-Filter (Gitter- und Leiter-Strukturen) (fakultativ)

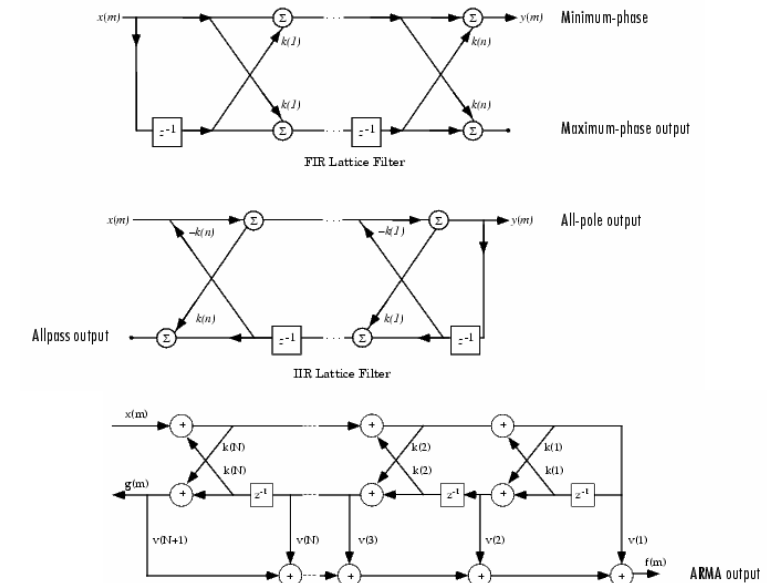


Abbildung 5-19: Lattice-Filter (Auswahl)

Quelle der Grafiken: Matlab-Hilfe:

http://www.mathworks.com/access/helpdesk/help/pdf_doc/signal/signal_tb.pdf Seite 1-40

Lattice-Filter stellen eine grosse Familie von Filtern dar mit gekreuzten Signalpfaden. Entsprechend schwierig sind Entwurf und Stabilitätsbetrachtungen. Lattice-Filtern sind vorteilhaft wenn es auf einen genau definierten Phasengang ankommt.

5.6. Nichtidealitäten

Viele Nichtidealitäten bei der Realisierung von digitalen Systemen bzw. Digitalfiltern hängen mit der endlichen Wortlänge in der Zahlendarstellung zusammen.

Festkomma-Zahlendarstellung

Bei der Festkomma-Zahlendarstellung mit einem Wort der Länge W Bits wird die Position des binären Kommas fest vereinbart.

Bei allen gängigen Signalprozessoren wird die Zweierkomplementdarstellung verwendet, da sie viele Vorteile hat (eindeutige Null, einfache Subtraktion, effiziente Algorithmen für die Multiplikation).

In Abbildung 5-20 sind die am häufigsten verwendeten Kommastellen und die Gewichtung der einzelnen Bitpositionen für Zweierkomplement-Zahlen mit Vorzeichen und für Zweierkomplement-Zahlen ohne Vorzeichen dargestellt.

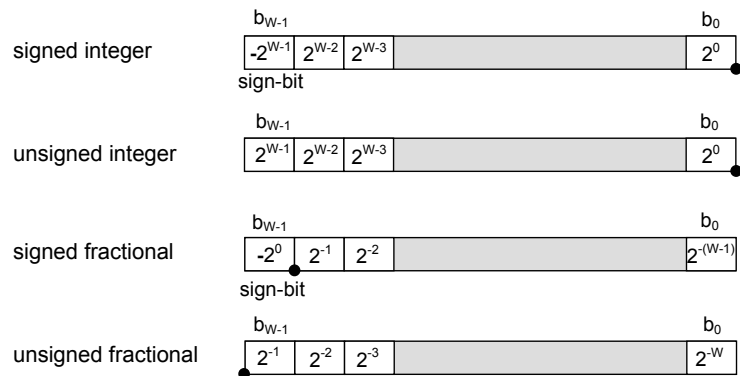


Abbildung 5-20: Festkomma-Zahlendarstellungsarten (Zweierkomplement).

In Tabelle 5-3 sind für eine Wortlänge von $W=3$ Bits die Zahlenwerte für die vier oben erwähnten Formate aufgelistet. Vergewissern Sie sich, dass eine binäre Zahl b einfach negiert werden kann, indem alle Bits von b invertiert und 1 LSB addiert wird.

$[b_2 \ b_1 \ b_0]$	signed integer	unsigned integer	signed fractional	unsigned frac.
[0 0 0]	0	0	0.00	0.000
[0 0 1]	1	1	0.25	0.125
[0 1 0]	2	2	0.50	0.250
[0 1 1]	3	3	0.75	0.375
[1 0 0]	-4	4	-1.00	0.500
[1 0 1]	-3	5	-0.75	0.625
[1 1 0]	-2	6	-0.50	0.750
[1 1 1]	-1	7	-0.25	0.875

Tabelle 5-3: Worte der Länge $W=3$ und entsprechende Festkomma-Zahlenwerte.

Die Festkomma-Zahlendarstellung hat einen gewichtigen Vorteil. Die HW-Realisierung der Zahlenarithmetik braucht weniger Chipfläche als die HW-Realisierung der Gleitkomma-Zahlenarithmetik. Deshalb sind Festkomma-DSP's im Allgemeinen kostengünstiger als vergleichbare Gleitkomma-DSP's.

Die Festkomma-Zahlendarstellung hat aber die folgenden Nachteile:

- Die Filterkoeffizienten müssen quantisiert bzw. können nicht beliebig genau dargestellt werden. Dadurch verändert sich die Lage der Pole und Nullstellen in der z -Ebene bzw. der resultierende Frequenzgang.

Wie erwähnt sind die Kaskaden- und die Parallelstruktur weniger sensitiv auf die Koeffizientenquantisierung als die Direktstrukturen.

Besonders selektive IIR-Filter mit Polen ganz nahe beim Einheitskreis können im Extremfall durch die Quantisierung der Filterkoeffizienten ausserhalb des Einheitskreises zu liegen kommen und so sogar instabil werden.

- Die Additionsergebnisse können überlaufen. Die resultierenden nichtlinearen Verzerrungen können bei IIR-Filtern im Extremfall grosse Überlaufschwingungen anregen. Dagegen kann man Folgendes unternehmen:

=> Das Eingangssignal des Filters bzw. eines Biquads kann gezielt so skaliert bzw. abgeschwächt werden, dass die Addierer nicht mehr überlaufen können. Dadurch sinkt aber das Signal-zu-Quantisierungsgeräuschverhältnis (SNR).

=> Neuere DSP's weisen ein grosses MAC-Register der Länge $2 \cdot W + 8$ Bits auf. Dadurch können viele Multiplikationsergebnisse ohne Rundungsfehler berechnet und temporäre Überläufe toleriert werden.

Die Zweierkomplement-Arithmetik hat nämlich die folgende, nützliche Eigenschaft: Werden mehrere Zweierkomplement-Zahlen addiert, deren Summe nicht überläuft, ist das Resultat der Zweierkomplement-Addition dieser Zahlen auch richtig, wenn Zwischensummen überlaufen.

- Das Resultat der Multiplikation von 2 Zahlen der Länge W -Bits ist im Allgemeinen $2 \cdot W$ -Bits lang. Irgendwann muss deshalb eine Rundung vorgenommen werden. Das resultierende Rundungsrauschen kann bei IIR-Filtern im Extremfall zu Grenzschränkungen mit kleinen Amplituden führen, die z.B. bei der digitalen Audiosignalverarbeitung stören.

Gleitkomma-Zahlendarstellung

Die Gleitkomma-Zahlendarstellung soll am Beispiel des IEEE Standards 754/854 (32 Bit) dargestellt werden, siehe Abbildung 5-21.

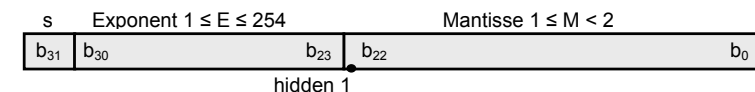


Abbildung 5-21: IEEE 754/854 Gleitkomma-Zahlendarstellung.

Der (float)-Wert F der Gleitkomma-Zahl wird bestimmt durch

$$F = (-1)^s \cdot M \cdot 2^{E-127}, \tag{5.21}$$

wobei M die Mantisse und E den Exponenten darstellen.

Beim IEEE-Standard ist die Mantisse 23 Bit lang, wobei man sich vor dem Komma eine 1 denkt. Dadurch nimmt die Mantisse Werte zwischen 1 und 2 an.

Der Exponent ist 8 Bit lang. Die Werte E=0 und E=255 werden zur Darstellung von Null, keine Zahl (engl: nan bzw. not a number) und Unendlich reserviert. In Gleichung (5.21) liegt E deshalb im Bereich $1 \leq E \leq 254$.

Die grösste darstellbare Zahl ist etwas kleiner als $2 \cdot 2^{254-127} \approx 3.4 \cdot 10^{38}$. Die kleinste darstellbare Zahl ist $-1 \cdot 2^{1-127} \approx -1.17 \cdot 10^{-38}$.

Der Dynamikbereich ist damit sehr gross und betragsmässig kleine Zahlen können sehr genau dargestellt werden.

Das Überlaufproblem kann im Wesentlichen ausgeschlossen werden. Allerdings entstehen kleine Quantisierungsfehler beim Addieren und Multiplizieren von 2 Gleitkomma-Zahlen.

5.6.1. Beispiel „Festkomma-Realisierung eines IIR-Filters“

Es soll ein Digitalfilter mit folgender Spezifikation entworfen und als Biquad-Kaskade mit quantisierten Filterkoeffizienten realisiert werden.

Filterart	(elliptisches) Tiefpass-Filter
Filterordnung	N=4
Abtastfrequenz	f _s =8000 Hz
Eckfrequenz Durchlassbereich	f _{DB} =1000 Hz
Eckfrequenz Sperrbereich	f _{SB} =1300 Hz
max. Rippel im Durchlassbereich	R _p =3 dB
min. Rippel im Sperrbereich	R _s =40 dB
Wortbreite	W=8 Bit

Tabelle 5-4: Filterspezifikation Beispiel-Filter.

In Abbildung 5-22 ist das Pol-Nullstellendiagramm der resultierenden UTF H(z) dargestellt, wobei die Filterkoeffizienten noch nicht quantisiert worden sind.

Dieses IIR-Filter wird nun als Biquad-Kaskade realisiert. Die UTF H(z) wird dazu in die beiden Teil-UTF H₁(z) und H₂(z) faktorisiert, d.h.

$$H(z) = k_1 \cdot H_1(z) \cdot k_2 \cdot H_2(z), \tag{5.23}$$

wobei die in Abbildung 5-22 dargestellte Pol-Nullstellenpaarung vorgenommen wird (Matlab Signal Processing Toolbox `tf2sos`).

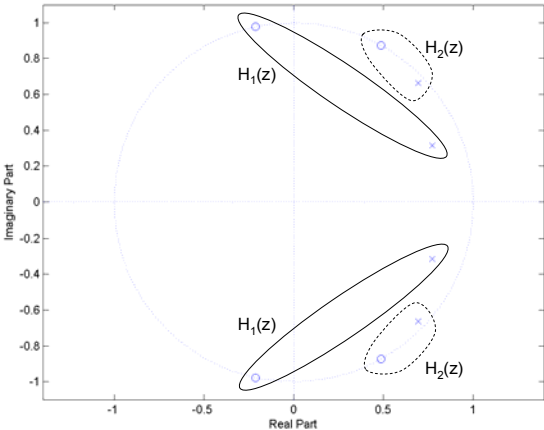


Abbildung 5-22: Pol-Nullstellendiagramm zu Beispiel-Filter.

An den Ausgängen der beiden Biquad-Filter darf kein Wert grösser als 1 werden. Die beiden Skalierungsfaktoren k₁ und k₂ der beiden Biquad-Filter H₁(z) und H₂(z) in Gleichung (5.22) werden deshalb so gewählt, dass

$$k_1 \cdot \max(|H_1(f)|) < 1 \text{ und } k_1 \cdot k_2 \cdot \max(|H_1(f) \cdot H_2(f)|) < 1 \tag{4.26}$$

Damit wird ein Überlauf unwahrscheinlich, aber nicht unmöglich. Die beiden Skalierungsfaktoren k₁ und k₂ können in die b-Koeffizienten der beiden Biquad-Filter integriert werden.

In Abbildung 5-23 ist das Spektrum [dB] von k₁·|H₁(f)| und k₂·|H₂(f)| dargestellt, wobei nun alle Filterkoeffizienten auf W=8 Bit quantisiert worden sind. Im 2. Biquadfilter ist die Pol-Überhöhung gut erkennbar.

Die Summe der beiden Teil-Spektren ergibt das Spektrum [dB] des Gesamt-Filters H(z). Es ist gut erkennbar, dass die Spezifikation aus Tabelle 5-4 erfüllt werden kann.

In Abbildung 5-23 ist das Spektrum [dB] des Filters |H_{dir}(f)| dargestellt, das ausgehend von der Filterspezifikation in Tabelle 5-4 nicht als Biquad-Kaskade, sondern in einer Direktform mit 8-Bit quantisierten Filterkoeffizienten realisiert worden ist. Man erkennt, dass die Spezifikationen aus Tabelle 5-4 nicht mehr erfüllt sind.

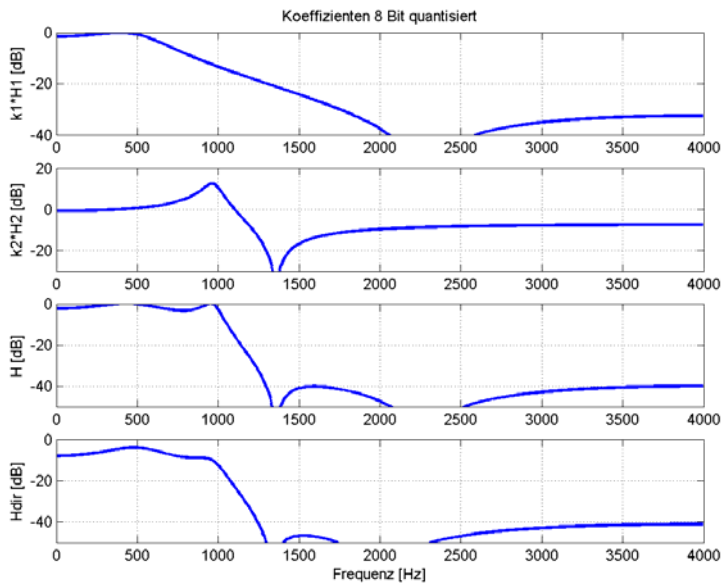


Abbildung 5-23: Betragsspektren der beiden Biquadfilter, des Biquad-Kaskaden-Filters sowie eines äquivalenten Filters in Direktstruktur.

5.7. Zusammenfassung Filter-Entwurf und Matlab-Befehle

5.7.1. Wahl FIR- oder IIR-Filter

Zuerst soll entschieden werden, ob ein FIR- oder IIR-Filter besser geeignet ist.

FIR-Filter	IIR-Filter
Hohe Filterordnung, da der Frequenzgang mit Nullstellen nur abgesenkt werden kann	Kleine Filterordnung (wenig Rechenzeit), da Frequenzgang mit Nullstellen abgesenkt und mit Polen angehoben werden kann
Lange Verzögerung = Filterordnung/2 / fs	Kleine Zeitverzögerung (kurze Latenzzeit)
Linearphasige Filter möglich Allpässe nicht realisierbar	Linearer Phasengang nur akausal möglich Minimalphasige Systeme einfach machbar
Immer stabil, da nicht rückgekoppelt, unempfindlich auf Quantisierung/Störungen	Stabilitätsprobleme bei Polen nahe am Einheitskreis (Quantisierung, Störungen)
Endliche Impulsantwort	Impulsantwort klingt theoretisch nie ab
Transversal-Struktur bevorzugt	Biquad-Kaskade bevorzugt

Tabelle 5-5: Kriterien für Auswahl FIR- oder IIR-Filter

Die folgenden Darlegungen beschränken sich auf Tiefpass-Filter, da die Filter-Koeffizienten von Hoch- und Bandpässen durch Transformation einfach aus den Filter-Koeffizienten eines Tiefpasses berechnen lassen (Matlab erledigt das im Hintergrund).

5.7.2. FIR-Filter-Entwurfsmethoden

Bei einem FIR-Filter sind die **Filterkoeffizienten b_k gleich der Impulsantwort $h[n]$** . Der Entwurf eines FIR-Filters kann somit auf das Bestimmen der Impulsantwort reduziert werden.

Es muss ein **Kompromiss** gefunden werden zwischen kurzer Latenzzeit und kleinem Realisierungsaufwand einerseits und schmalem Übergangsbereich und hoher Genauigkeit im Durchlass- und Sperrbereich andererseits. Deshalb stehen verschiedene Entwurfs-Methoden zur Auswahl.

FIR-Filter: Approximations-Methode	Matlab
Fenstermethode Ein idealer Tiefpass hat im Durchlassbereich keine Dämpfung und lässt im Sperrbereich kein Signal passieren, ein Übergangsbereich ist nicht vorhanden. Die Rücktransformation dieses „Rechtecks“ aus dem Frequenzbereich liefert als Impulsantwort einen $\sin(x)/x$ -Impuls. $\sin(x)/x$ dehnt sich in beide Richtungen bis ins Unendliche aus. Für kausale Filter würde das eine unendlich lange Latenzzeit bedeuten. Der $\sin(x)/x$ -Impuls muss also bei abgeschnitten werden. Die durch das Abschneiden fehlende Kompensation hochfrequenter Anteile führt zum Gibbs'schen Phänomen. Dem kann durch Gewichtung mit einer Fensterfunktion entgegengewirkt werden. Im gleichen Mass wie das Gibbs'sche Phänomen verschwindet verbreitert sich der Übergangsbereich.	fir1
Frequenz-Abtastung Statt einen idealen Tiefpass nachzuahmen, kann man den Frequenzgang durch einige Eckpunkte vorgeben und mit Hilfe der Frequenzabtastung und der inversen FFT die Impulsantwort bestimmen. Das Gibbs'sche Phänomen wird auch hier durch gewichten mit einer Fensterfunktion vermieden.	fir2
Least-Square-Methode Der Frequenzgang soll nicht nur an den Eckpunkten, sondern über den ganzen Bereich möglichst gut stimmen. Die Näherung wird mit der Methode der kleinsten Fehlerquadrate erreicht.	firls fircls firls1
Raised-Cosine Ein Übergangsbereich mit cosinus-förmigen Verlauf (raised cosine) hat im Frequenzgang keine Ecken und eignet sich deshalb gut für Anwendungen, wo Bandbreite gespart werden muss (Funkstrecken).	firrcos
Filterordnung abschätzen Die benötigte Filterordnung für ein FIR-Filter kann mit dem Befehl kaiserord abgeschätzt werden. Es ist vorteilhaft kaiserord vor dem Filter-Entwurf-Befehl aufzurufen, da die zurückgegebenen Parameter direkt an den Entwurfsbefehl weitergegeben werden können.	kaiserord
Synthese im z-Bereich Man kann sich auch ganz lösen von der Vorstellung eines idealen Tiefpass-Filters und ein Filter mit beliebigem Frequenzgang direkt in der z-Ebene entwerfen. Mit dem firpm-Befehl kann der Amplitudengang und mit cfirpm zusätzlich der Phasengang vorgegeben werden. Mit firpmord kann die benötigte Filterordnung ermittelt werden. firpmord sollte vor firpm resp. cfirpm aufgerufen werden, damit die zurückgegebenen Parameter direkt weiterverwendet werden können.	firpm cfirpm firpmord

Tabelle 5-6: FIR-Entwurfs-Methoden

5.7.3. IIR-Filter-Entwurfsmethoden

Filter aus der Analog-Technik können mit Differential-Gleichungen oder mit komplexen Frequenzgängen (Zähler- und Nenner-Polynome in s) beschrieben werden. Der Filter-Entwurf im Analogen ist ausgereift und lässt sich für den Entwurf von IIR-Filtern nutzen. Die Differential-Gleichungen können durch Differenzen-Gleichungen approximiert werden oder die Polynome in s können zu Polynomen in z umgewandelt werden. Beim Übergang vom kontinuierlichen s - in den diskreten z -Bereich wird der Frequenzgang in z mit 2π periodisch und die Zuordnung $s \rightarrow z$ ist nicht linear. Die Impulsantwort und/oder der Frequenzgang sind deshalb unterschiedlich für analoge und digitale Systeme.

IIR-Filter: Approximations-Methode	Matlab
Impulsinvariante Transformation Ziel: für einen Impuls stimmen das digitale und das analoge System überein. Die Impulsantworten sind jeweils zu den Abtastzeiten $t = n \cdot T_s$ gleich: $h[n] = T_s \cdot h(t)$ Der gewünschte Frequenzgang $H(s)$ wird definiert/vorgegeben. Die analogen Filterkoeffizienten b und a sind damit bekannt. Daraus wird $h(t)$ durch Rücktransformation berechnet. $h[n] = T_s \cdot h(t)$ ist die digitale Variante der Impulsantwort. $H(z)$ wird durch Fourier-Transformation ermittelt. Die digitalen Filterkoeffizienten b und a werden nun berechnet.	impinvar
Bilineare Transformation Ziel: Frequenzgänge des digitalen und des analogen System stimmen überein. Wenn der genaue Zusammenhang $z = e^{s \cdot T_s}$ angewendet wird, entsteht für $H(z)$ kein Polynom in z . Die Filter-Koeffizienten b und a lassen sich nicht bestimmen. Bei der bilinearen Transformation wird anstelle der Definitionsgleichung $z = e^{s \cdot T_s}$ resp. $s = fs \cdot \ln(z)$ die Näherung $s = 2 \cdot fs \cdot (z-1)/(z+1)$ verwendet ($= 1$. Glied der Reihenentwicklung des Logarithmus). Damit entsteht aus einem Polynom in s (analoge Filterkoeffizienten b und a sind bekannt) ein Polynom in z (digitalen Filterkoeffizienten b und a sind berechenbar). $H(z) = H(s)$ mit $s = 2 \cdot fs \cdot (z-1)/(z+1)$ Die ganze imaginäre Achse in s wird einmal auf dem z -Einheitskreis abgebildet und die analoge Frequenz-Achse wird mit einer Tangens-Funktion gestaucht. Meist wird die Frequenzachse vor der Transformation so gedehnt, dass die analoge und die digitale Grenzfrequenz des Filters gleich sind (prewarping).	bilinear
Analoge Prototypen Statt den Frequenzgang $H(s)$ festzulegen und mit der impulsinvarianten oder der bilinearen Transformation $H(z)$ zu bestimmen, kann man auch analoge Prototypen mit genau bekannten Eigenschaften als Grundlage benutzen.	
Butterworth-Filter Maximal flacher Amplitudengang im Durchlassbereich	butter buttord
Chebyscheff-1-Filter Definierte Welligkeit im Amplitudengang im Durchlassbereich. Steiler und monotoner Abfall im Sperrbereich.	cheby1 cheb1ord
Chebyscheff-2-Filter Definierte Welligkeit im Amplitudengang im Sperrbereich. Monotoner Verlauf im Durchlassbereich. Steiler Abfall im Übergangsbereich.	cheby2 cheb2ord
Elliptisches oder Kauser-Filter Definierte Welligkeit im Amplitudengang im Durchlass- und im Sperrbereich. Sehr steiler und monotoner Abfall im Übergangsbereich. Kleinste Filterordnung.	ellip ellipord
Synthese im z-Bereich Wie bei FIR-Filtern lassen sich die Koeffizienten b und a für einen beliebigen Frequenzgang $H(z)$ direkt im z -Bereich errechnen (kleinste Fehlerquadrate).	yulewalk

Tabelle 5-7: IIR-Entwurfs-Methoden

5.7.4. FIR-Filter: Vergleich der Frequenzgänge

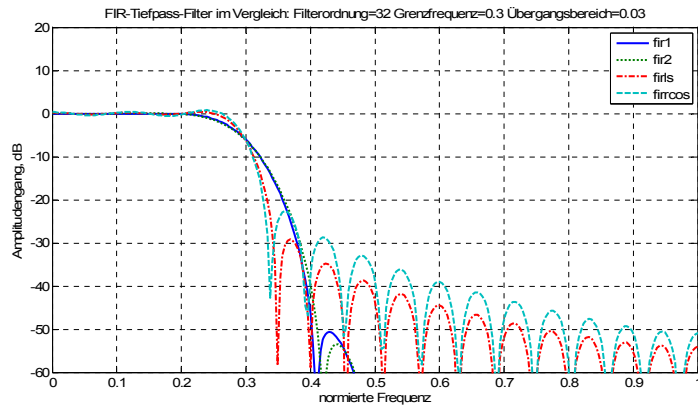


Abbildung 5-24: FIR-Tiefpass-Filter
Demo: dsv1kap5_filter_vergleichFIR_TP.m

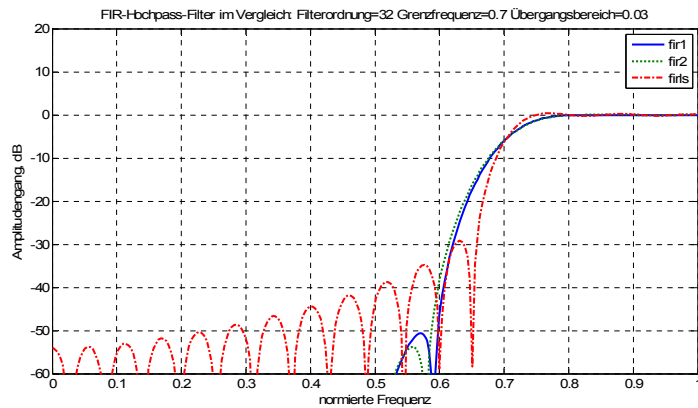


Abbildung 5-25: FIR-Hochpass-Filter
Demo: dsv1kap5_filter_vergleichFIR_HP.m

Aus den Abbildungen 5-24 und 5-25 ist gut ersichtlich, dass bei gleicher Filterordnung entweder ein schmaler Übergangsbereich oder ein hohe Dämpfung im Sperrbereich erreichbar sind.

5.7.5. IIR-Filter: Vergleich der Frequenzgänge

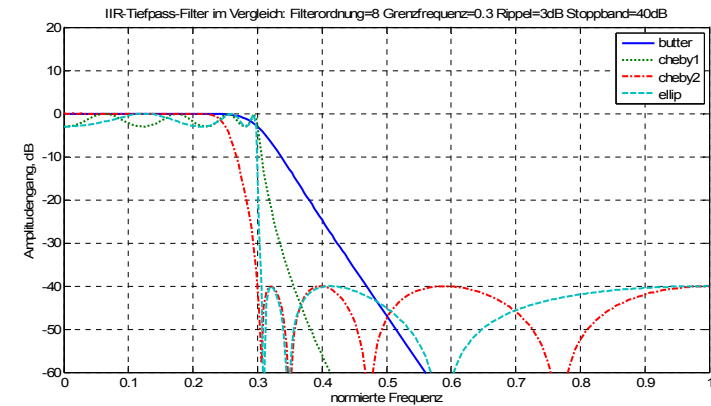


Abbildung 5-26: IIR-Tiefpass-Filter
Demo: dsv1kap5_filter_vergleichIIR_TP.m

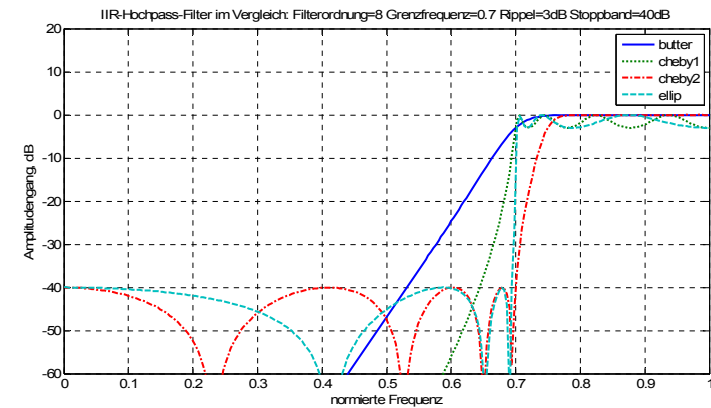


Abbildung 5-27: IIR-Hochpass-Filter
Demo: dsv1kap5_filter_vergleichIIR_HP.m

Aus den Abbildungen 5-24 bis 5-27 geht hervor, dass im Vergleich zu FIR-Filtern bei IIR-Filtern eine wesentlich kleinere Ordnung genügt.
Aus den Abbildungen 5-26 und 5-27 ist ersichtlich, dass der Übergangsbereich schmaler wird, je mehr Ripple im Durchlass- und Sperrbereich toleriert werden.