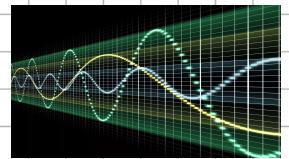


Einleitung

Geschichte



1600 - 1800: math. Grundlagen

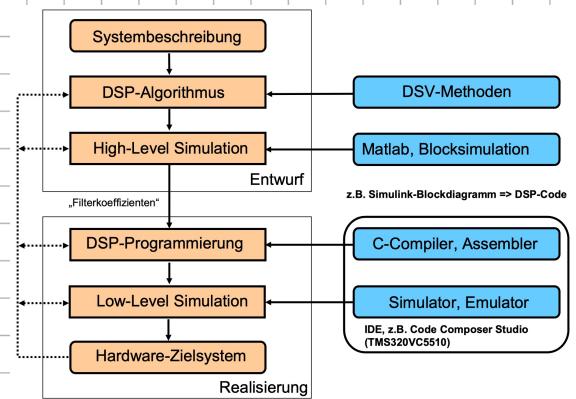
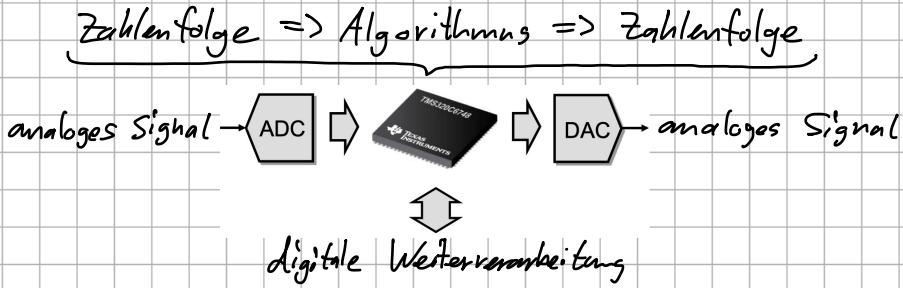
1900 - 1950: Fundamente wie Abtasttheorie, Transversalfilter, Puls-Code-Modulation
ab 1950: analoge Systeme vor Realisierung auf digitalen Rechnern simulieren (viele grösse Fflexibilität)

1964: adaptive Entzerrung bei Datenübertragung

1965: Fast Fourier Transform

Anfang 1980er: Durchbruch d. Echtzeit-Signalverarbeitung mit ersten frei programmierbaren DS-Prozessoren

prinzipieller Ablauf d. DSV



DSP

für die Ausführung von DSV-Funktionen in Echtzeit optimierter elektronischer Bausteine
diese Funktionen basieren auf wiederholten Multiply-And-Accumulate (MAC) Operationen

wichtige digitale Signale und deren z-Transformierte

$$u[n] = \begin{cases} 1 & n \geq 0 \\ 0 & n < 0 \end{cases} \quad \bullet \longrightarrow 1 \quad \text{ROC: all } z$$

$$\delta[n] = \begin{cases} 1 & n=0 \\ 0 & n \neq 0 \end{cases} \quad \bullet \longrightarrow \frac{1}{1-z^{-1}} \quad \text{ROC: } |z| > 1$$

wichtige Formeln

geometrische Reihe:

$$\sum_{k=0}^n q^k = \frac{1-q^{n+1}}{1-q}$$

Frequenzgang $H(f)$
amz UTF $H(z)$:

$$H(f) = H(z = e^{j2\pi f T_s})$$

Mathematische Grundlagen AD/DA-Umsetzung

Dirac-Impuls

Der Dirac-Impuls $\delta(t)$ kann als genügend kurzer Rechteck-Impuls hoher Amplitude gedeutet werden. Er wird üblicherweise mit einem Pfeil und allenfalls einem „Gewicht“ dargestellt.

Mathematisch gesehen ist der Dirac-Impuls $\delta(t)$ eine verallgemeinerte Funktion, eine sogenannte Distribution. Er kann durch das folgende Integral definiert werden:

$$\int_{-\infty}^{\infty} x(t) \cdot \delta(t-t_0) dt = x(t_0)$$

Der Dirac-Impuls verfügt über eine eigentliche *Sieb-* bzw. *Ausblendeigenschaft*. Im Integral oben wird aus der Funktion $x(t)$ genau der Wert an der Impulsstelle $t=t_0$ „herausgesiebt“. Es kann auch gezeigt werden, dass die *Siebeigenschaft* für ein Produkt von einem Dirac-Impuls $\delta(t)$ mit einem Signal $x(t)$ gilt, d.h.

$$x(t) \cdot \delta(t-t_0) = x(t_0) \cdot \delta(t-t_0).$$

Beispiel

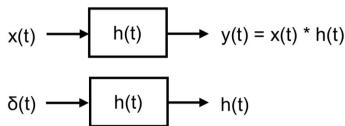


Abbildung: Stossantwort eines linearen, zeitkontinuierlichen Systems.

Das Signal $y(t)$ am Ausgang eines linearen, zeitkontinuierlichen Systems ist durch die Faltung des Eingangssignals $x(t)$ mit der Stossantwort $h(t)$ des Systems gegeben, d.h.

$$y(t) = x(t) * h(t) = \int_{-\infty}^{\infty} x(\tau) \cdot h(t-\tau) d\tau.$$

Wenn man einen Dirac-Impuls $\delta(t)$ als Eingangssignal verwendet, antwortet das System mit der Stossantwort $h(t)$, d.h.

$$y(t) = \delta(t) * h(t) = h(t),$$

siehe Abbildung oben.

Fourierreihe

Signale $x(t)$ mit Periode T , d.h. $x(t) = x(t+nT)$ für alle ganzen Zahlen n , können als komplexe Fourierreihe dargestellt werden:

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\frac{2\pi}{T}t}$$

$$c_n = \frac{1}{T} \int_{-T/2}^{T/2} x(t) e^{-jn\frac{2\pi}{T}t} dt$$

Periodische Signale $x(t)$ haben ein diskretes Linienspektrum c_n bei den ganzzahligen Vielfachen der Grundfrequenz $1/T$. Alternativ können periodische Signale $x(t)$ auch als reelle Fourierreihe dargestellt werden:

$$x(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos(n \frac{2\pi}{T} t + \varphi_n)$$

Der Zusammenhang zwischen den reellen und den komplexen Fourierkoeffizienten lautet:

$$A_0 = c_0, \quad A_n = 2|c_n| \text{ für } n=1,2,\dots \quad \varphi_n = \arg(c_n)$$

Für die normierte Leistung (an 1Ω) gilt:

$$\frac{1}{T} \int x^2(t) dt = A_0^2 + \frac{1}{2} \sum_{n=1}^{\infty} A_n^2 = \sum_{n=-\infty}^{\infty} |c_n|^2.$$

Beispiel

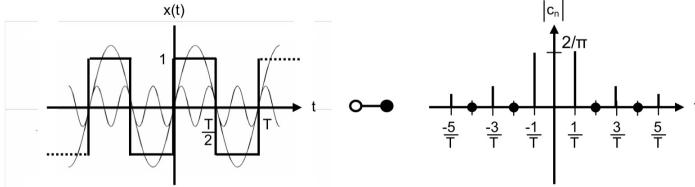


Abbildung: Fourierreihendarstellung und Fourierkoeffizienten eines periodischen Rechtecksignals.

Das in Abbildung A2-2 dargestellte periodische Rechtecksignal besitzt die reelle Fourierreihendarstellung

$$x(t) = \frac{4}{\pi} \left[\sin\left(\frac{2\pi}{T}t\right) - \frac{1}{3} \sin\left(3\frac{2\pi}{T}t\right) + \frac{1}{5} \sin\left(5\frac{2\pi}{T}t\right) - \dots \right]$$

und ein diskretes Linienspektrum bei den ungeraden Vielfachen der Grundfrequenz $1/T$, das betragsmäßig mit $1/n$ abnimmt. Die Phasenwerte betragen $\varphi_n = -n\pi/2$ für $n=\pm 1, \pm 3, \dots$

In der obigen Abbildung ist die Approximation des periodischen Rechtecksignals mit den ersten beiden Termen der Fourierreihe dargestellt.

Fouriertransformation

Für die allermeisten zeitkontinuierlichen Signale $x(t)$ mit endlicher Energie existiert die Fouriertransformation und die inverse Fouriertransformation:

$$X(f) = \int_{-\infty}^{\infty} x(t) \cdot e^{-j2\pi ft} dt$$

$$x(t) = \int_{-\infty}^{\infty} X(f) \cdot e^{j2\pi ft} df$$

Das Fourierspektrum wird manchmal auch in Funktion der Kreisfrequenz $\omega=2\pi f$ statt der Frequenz f angegeben.

Zeitkontinuierlichen Signale $x(t)$ mit endlicher Energie haben ein kontinuierliches Amplitudendichtespektrum [Amplitude/Hz].

In der folgenden Tabelle sind einige nützliche Eigenschaften der Fouriertransformation zusammengefasst:

Eigenschaft	Zeitbereich	Frequenzbereich
Faltung	$x(t) * h(t)$	$X(f) \cdot H(f)$
Multiplikation	$x(t) \cdot h(t)$	$X(f) * H(f)$
Symmetrie	$X(t)$	$X(-f)$
Superposition	$a \cdot x(t) + b \cdot y(t)$	$a \cdot X(f) + b \cdot Y(f)$
Ähnlichkeit	$x(a \cdot t), a \neq 0$	$(1/ a) \cdot X(f/a)$
Zeitverschiebung	$x(t-t_0)$	$X(f) \cdot e^{-j2\pi f t_0}$
Frequenzverschiebung	$x(t) \cdot e^{j2\pi f_0 t}$	$X(f-f_0)$
Differentiation	$d^n / dt^n x(t)$	$(j2\pi f)^n \cdot X(f)$

Tabelle: Eigenschaften der Fouriertransformation.

Beispiel

Das Signal $y(t)$ am Ausgang eines linearen, zeitkontinuierlichen Systems ist durch die Faltung des Eingangssignals $x(t)$ mit der Stossantwort $h(t)$ des Systems gegeben, d.h. $y(t) = x(t) * h(t)$, siehe Abbildung.

Mit Hilfe der Fouriertransformation bzw. der Faltungseigenschaft gilt für das Fourier-Spektrum des Ausgangssignals $Y(f)$ die einfache Beziehung

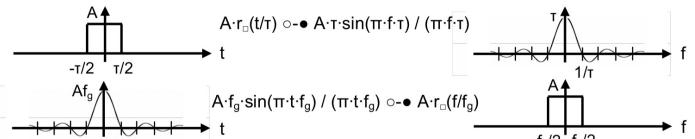
$$Y(f) = X(f) \cdot H(f),$$

wobei $X(f)$ das Fourierspektrum des Eingangssignals $x(t)$ und $H(f)$ das Fourierspektrum der Stossantwort $h(t)$ darstellen. $H(f)$ wird üblicherweise als *Übertragungsfunktion* bezeichnet, die oft mit der Methode der symbolischen Wechselspannungsrechnung bestimmt wird.

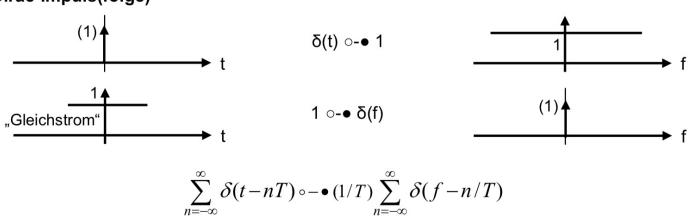
Spezielle Signalfunktionen im Zeit- und Frequenzbereich

In der folgenden Tabelle sind einige nützliche Signalfunktionen im Zeit- und im Frequenzbereich dargestellt.

Rechteckimpuls bzw. idealer TP



Dirac-Impuls(folge)



Cosinus

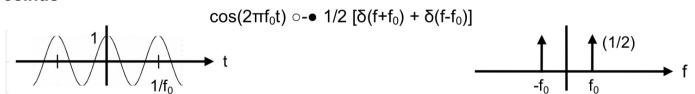
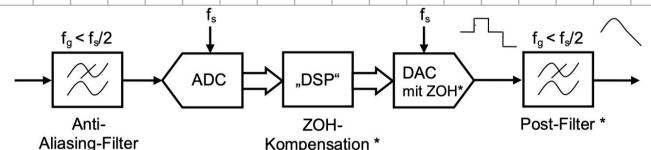


Tabelle: Spezielle Signalfunktionen im Zeit- und im Frequenzbereich.

AD/DA - Umsetzung



Abtastung

ideal abgetastetes Signal: $x_s(t) = x(t) \sum_{n=-\infty}^{\infty} \delta(t-nT_s)$

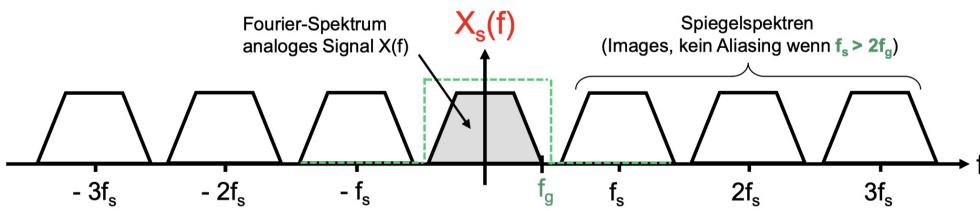
mit der FR des Dirac-Kammes $\sum_{n=-\infty}^{\infty} c_n \exp(jk \frac{2\pi}{T_s} t)$, $c_k = \frac{1}{T_s}$ kann man x_s auch schreiben als

$$x_s(t) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} x(t) e^{jn \frac{2\pi}{T_s} t} \quad \text{oder} \quad X_s(f) = \frac{1}{T_s} \sum_{n=-\infty}^{\infty} X(f - n f_s)$$

mit der Siebeigenschaft kann man x_s auch schreiben als

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT_s) \delta(t-nT_s) \quad \text{oder} \quad X_s(f) = \sum_{n=-\infty}^{\infty} x(nT_s) e^{-jn 2\pi \frac{f}{f_s}}$$

wichtig im Zusammenhang mit Z-Transformation



Aliasing

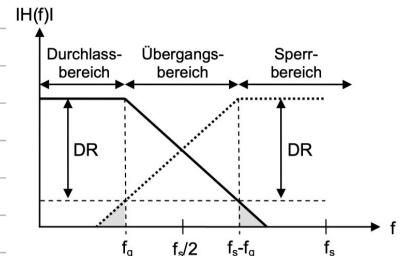
interessierendes Frequenzband: $[0, f_g]$

=> falls analoges Eingangssignal Komponenten darüber $f_s - f_g$ aufweist entstehen Aliaskomponenten im interessierenden Frequenzband und limitieren den DR

Aliaskomponenten in $[f_g, f_s/2]$ hingegen sind uninteressant und limitieren den DR nicht

Die Dämpfung ab $f_s - f_g$ muss also > DR sein

mit Oversampling-Methode kann der Filterabstand vom Analogen ins Digitale verschoben werden.



Dämpfung eines Butterworth-Anti-Aliasing-Filters N-ter Ordnung

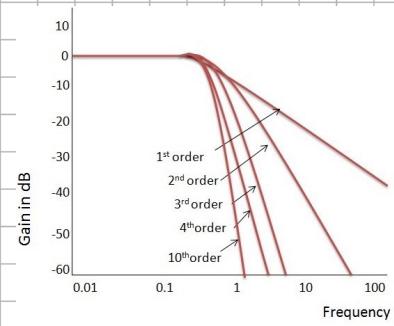
$$A(f) \approx 20N \text{dB} \log_{10}\left(\frac{f}{f_g}\right) \quad \text{für } f > f_g$$

$\rightarrow N \cdot 20 \text{dB}/\text{Dek}$ resp. $N \cdot 6 \text{dB}/\text{Oct}$
 ↳ Frequenzverzehnfachung ↳ Frequenzverdopplung

Achtung:

beim Auslegen TP-Wirkung des ZOH berücksichtigen

$$\hookrightarrow \sin(\pi f/f_s) / (\pi f/f_s)$$



Undersampling (auch Bandpass Sampling)

Abtasten eines Signals außerhalb der 1. Nyquistzone

→ (B) und (C) in der Abbildung rechts

bei Abtastung in gerader Nyquistzone ist das Spektrum umgedreht (Kehrlage)

in Mobilkommunikation bspw. weit verbreitet, weil man mit einer tiefen Abtastrate auskommt und gleichzeitig das Empfangssignal ins Basisband verschieben kann

das Abtasttheorem für die Unterabtastung eines Bandpass-Signals mit Mittelfrequenz f_c und Bandbreite B lautet:

$f_s > 2B$ Abtasttheorem für Bandpass-Signal

zusätzlich muss f_c in der Mitte der Nyquistzone $N/2$ liegen:

$$f_s = \frac{4f_c}{2N-1}$$

auch beim Undersampling ist normalerweise ein Anti-Aliasing-Filter am Eingang erforderlich

→ es muss natürlich Bandpass-Charakter aufweisen!

Achtung: ADC muss Eingangs frequenz von $f_c + B$ unterstützen !

Hardware

gibt verschiedene Möglichkeiten ADCs und DACs zu realisieren, welche sich bezüglich Geschwindigkeit, Chipflächenbedarf und Energieverbrauch unterscheiden

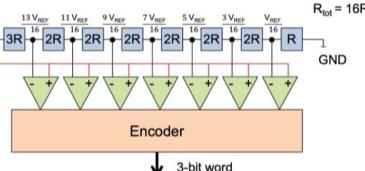
Example Flash ADC

- Network of 2^N resistors to divide V_{REF} into 2^N levels
- $2^N - 1$ analog comparators
 - Compare input signal to divided reference voltages
- Encoder transforms digital comparator results into N-bit word

Fast conversion

Requires many elements

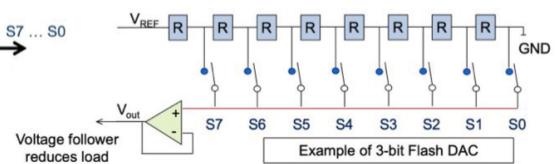
- e.g. 255 comparators for 8-bit resolution
- Power hungry
- Consumes large chip area



Example Flash DAC

- Network of resistors (of same value) creates 2^N voltage levels
- N-bit digital input decoded into 2^N values (S0 ... Sx)
 - Select single voltage level as DAC output

3-bit word → 3 to 8 decoder → S7 ... S0



Direkte Digitale Synthese DDS

DDS: digitale Frequenzsynthese

Frequenzsynthese:

Generierung von Sinus-Signalen mit unterschiedlichen Frequenzen ausgehend von einer Referenzfrequenz
→ wichtig für bspw. Messtechnik, Nachrichtentechnik

analog mit Phase-Locked-Loops (PLLs) erreichbar
→ hier werden Sinus-Signale jedoch verändert
auch digital generiert

Grundprinzip der DDS:

1 Sinusperiode als diskrete Werte in Lookup-Tabelle gespeist

mit (Modulo-)Adress-Zähler werden diese Abtastwerte nacheinander selektiert und mit einem DAC umgesetzt, nach DAC mit TP geöffnet

Wert des Zählers zum Zeitpunkt n entspricht der Phase, wird gemäß folgender Formel aktualisiert:

$$\text{Phase}[n] = (\text{Phase}[n-1] + M) \bmod N \quad \text{wobei } N = 2^W$$

von dieser Phase wird der hintere Teil abgeschnitten um den Index für die Look-Up-Tabelle zu erhalten

⇒ es werden immer alle Werte benötigt aber je nach Frequenz wird jeder individuelle Wert unterschiedlich oft benötigt

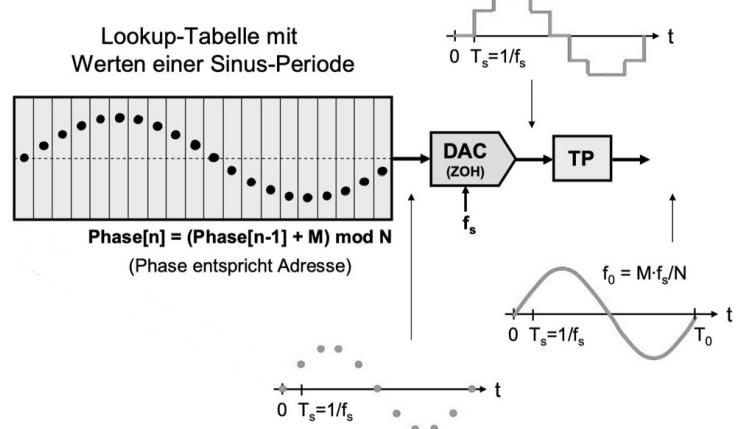
Frequenz kann durch Wahl des Tuning-Worts M eingestellt werden
↳ «Numerically Controlled Oscillator»

mit der sogenannten Tuning-Gleichung kann die Frequenz berechnet werden:

$$f_0 = M \cdot \Delta f = M \cdot f_s / N = M \cdot f_s / 2^W \quad \text{wobei } W = \text{Wortlänge d. Phasenzähls.}$$

Δf : Frequenzauflösung

DDS-System kann SW-mäßig mit einem Prozessor, Speicher und DAC realisiert werden
↳ um Prozessor zu entlasten ist es aber oft vorteilhaft einen CMOS-DDS-Baustein einzusetzen

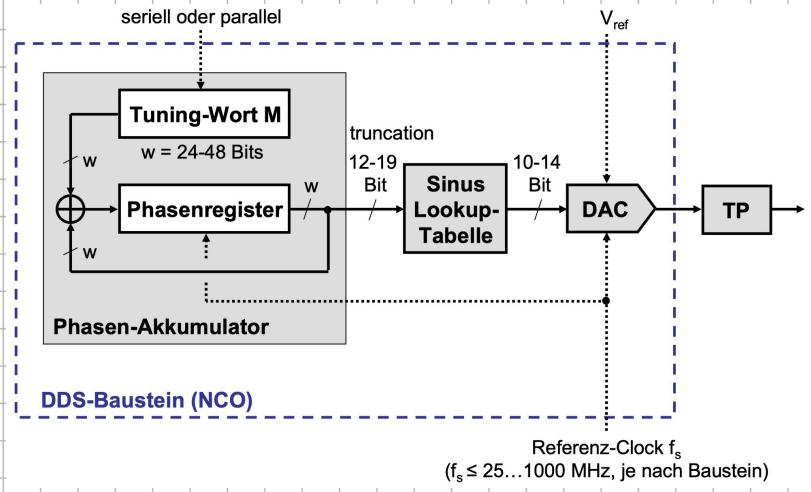


DDS-Baustein (NCO)

Adresse für Sinus-Lookup-Tabelle wird im Phasenakkumulator gemäß der Tuning-Gleichung generiert

um aus Akkumulator-Wert die tatsächliche Adresse zu erhalten müssen einige (viele) LSBits abgeschnitten

es muss aussuchen nur ein Quadrant der Periode gespeist werden



rechts ist Bezugsspektrum für den Fall $f_0 = 30 \text{ MHz}$ und $f_s = 100 \text{ MHz}$ abgebildet

neben f_0 auch die Kopien bei Vielfachen der Abtastrate, nämlich $f = k \cdot f_s \pm f_0$, zu sehen

Gewichtung mit $\sin(x)/x$ -förmiger Amplitudengang des ZOH

die Komponente bei $f_s - f_0$ und die anderen Kopien müssen mit TP weg gefiltert werden

Anforderungen an Steilheit & TP steigen, je näher f_0 bei $f_s/2$ gewählt wird

wegen Abtasttheorem ist Ausgangsfrequenz f_0 theoretisch auf halbe Abtastfrequenz beschränkt

→ wegen der TP-Filtrierung ist jedoch typischerweise $f_0 \leq 0.4 \cdot f_s$

aktuell (2023) sind DDS-Bausteine mit maximalen Abtastfrequenzen 2MHz bis 1000MHz erhältlich

Spanning Free Dynamic Range (SFDR): Maß für Reinheit des Spektrums

→ abhängig von Wortbreite des DAC (→ Quantisierungsrauschen)

→ gibt aber auch sekundäre Effekte, bspw. ist Quantisierung in DA-Umsetzung n. ganz linear

→ dadurch entstehen Oberwellen in 1. Nyquistzone (in der Abb bei 60MHz (+Kopie bei 40MHz))

integral non-linearity INL

DFT und FFT

Zusammenhang DFT und FR

tastet man eine periodische Funktion $x_p(t)$ mit N Werten pro Periode ab, also $T_s = T/N$
bzw. $f_s = N/T$ erhält man das diskrete Signal $x_{np}[n]$, dessen Spektrum aus dem Originalspektrum C_n und Kopien bei Vielfachen von f_s besteht

wenn beim Abtasten kein Aliasung entsteht, können die Spektrallinien C_n mit der DFT berechnet werden:

$$C_n = X[n]/N$$

Koeffizienten der FR (komp. Schreibweise) aus DFT

Filterfunktion der DFT

ein Cosinus-Signal $x(t) = \cos(2\pi f t)$ wurde mit f_s N mal abgetastet und die DFT berechnet, d.h. Frequenzauflösung ist $\Delta f = f_s/N$

Berechnung d. einzelnen DFT-Werte $X[m]$ entspricht einer Filterung

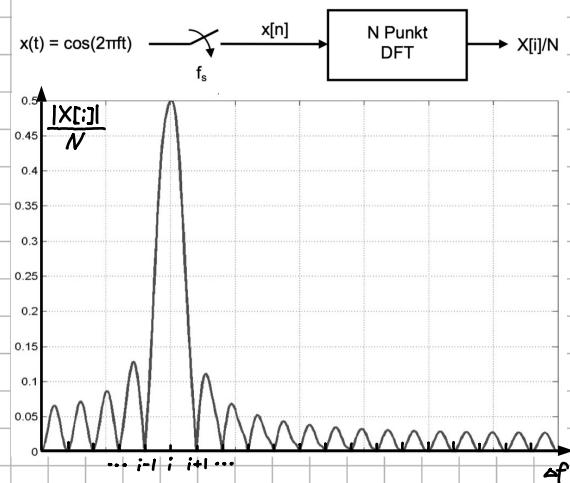
rechts dargestellt ist $|X[i]|/N$, also der Betrag des i^{ten} Wertes der normierten DFT in Abhängigkeit der Frequenz f des Cosinus-Signals

der i^{te} Wert entspricht einer Frequenz von $i \cdot \Delta f = i \cdot f_s/N$

falls $f = i \cdot \Delta f$, ist $|X[i]|/N$ wie erwartet = 0.5

solange $f \in [i \cdot \Delta f - \Delta f/2, i \cdot \Delta f + \Delta f/2]$ ist, gilt $|X[i]|/N \approx 0.5$

falls f außerhalb des Δf breiten Bereichs (eng. bin) um $i \cdot \Delta f$ liegt, spricht $|X[i]|/N$ weniger an und bei den anderen Vielfachen von Δf gilt sogar $|X[i]|/N = 0$
 → dort sprechen die anderen DFT-Spektralfilter an



Leakage

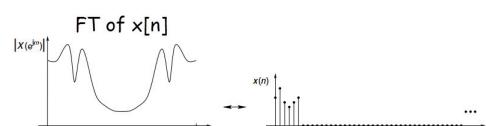
falls Fensterlänge unpassend gewählt wird, kann es aufgrund d. periodischen Fortsetzung, der Zeitfenster-Sequenz $x_{np}[n]$ zu Sprungstellen kommen, die im urspr. Signal n. vorhanden waren

→ resultierendes Spektrum weist zusätzliche, auslaufende Spektrallinien auf (leakage)

der Effekt kann verminderd werden, wenn die Zeitfenster-Sequenz mit einem ans laufenden Fenster (window) gewichtet wird:

$$x_w[n] = x_{np}[n] \cdot w[n], \quad n \in \{0, \dots, N-1\}$$

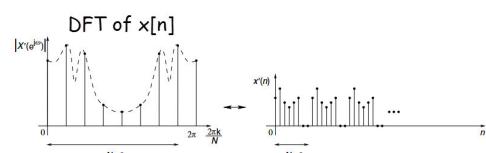
Widnowing



DFT

discrete Frequenzen

$$X[n] = \sum_{n=0}^{N-1} x[n] e^{-j 2\pi \frac{mn}{N} T_s} = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} mn}$$



Anzahl zur Berechnung der N Spektralwerte beträgt mit der DFT N^2 komplexe Multiplikationen

falls nur wenige Spektralwerte interessieren kann es vorteilhaft mit DFT die wenigen Spektralwerte direkt und nicht mit FFT alle möglichen DFT-Werte zu berechnen

Zeitfenster: $T_{\text{DFT}} = N \cdot T_s = N/f_s$
 Frequenzauflösung: $\Delta f = 1/T_{\text{DFT}} = f_s/N$

decimation-in-time FFT

Aufwand: ca. $N \cdot \log_2(N)$

N sei 2er-Potenz

Funktionsweise:

sukzessive N -Punkt DFT in zwei $N/2$ -Punkt DFTs aufteilen, ohne über die geordneten und eine über die ungeraden Abtastwerte

$$X[m] = \sum_{n=0}^{N-1} x[n] e^{-j\frac{2\pi}{N} mn} = \sum_{n=0}^{N-1} x[n] W_N^{mn}$$

$$X[m] = \sum_{n=0}^{N/2-1} x[2n] \cdot W_N^{m2n} + \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_N^{m(2n+1)}$$

$$W_N^{m(2n+1)} = W_N^{2mn+m} = W_N^m \cdot W_N^{2mn}$$

mit $W_N = e^{-j\frac{2\pi}{N}}$ «weight» oder auch «twiddle factor»

die FFT nutzt nun verschiedene spezielle Eigenschaften des Ausdrucks W_N^{mn} u.a.:

- $W_N^{mN} = e^{-j2\pi m} = 1$
- $W_N^{2mn} = W_{N/2}^{mn}$
- $W_N^{mn} = W_N^{(m+N)n} = W_N^{m(n+N)}$ periodisch in m, n
- $W_N^{m+N/2} = -W_N^m$

weil $W_N^{2mn} = W_{N/2}^{mn}$ kann man die Berechnung von $X[m]$ oben vereinfachen:

$$X[m] = \sum_{n=0}^{N/2-1} x[2n] \cdot W_{N/2}^{mn} + W_N^m \sum_{n=0}^{N/2-1} x[2n+1] \cdot W_{N/2}^{mn}$$

die beiden Summen stellen $N/2$ -Punkt DFTs dar

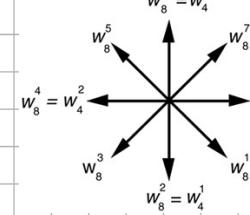
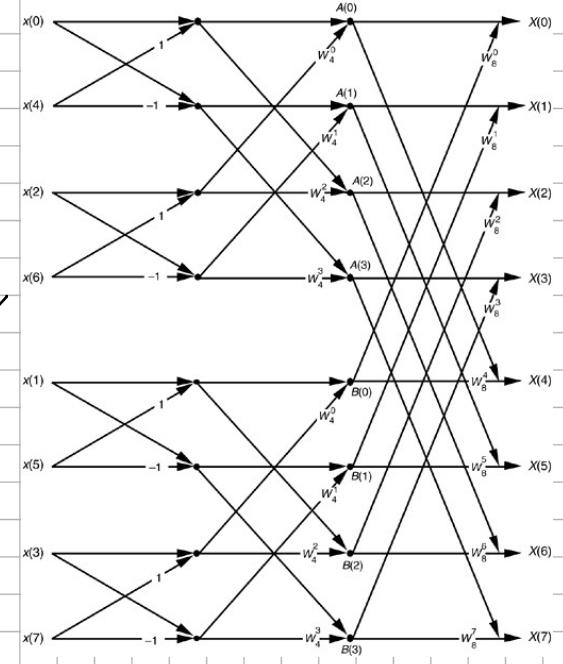
Spektrum kann somit in $\log_2(N)$ unabhängigen Teilschritten mithilfe von je $N/2$ sogenannten butterflies berechnet werden

es ist nur 1 Vektor mit N komplexen Spektrumarten erforderlich (in-place computation)

die korrekte Reihenfolge erhält man, wenn man den Zeitindex n von $x[n]$ binär darstellt und für die Spektrum rückwärts liest (bit-reversed-Adressierung)

Pseudocode:

```
function fft( $n, f$ ) :
    if ( $n = 1$ )
        return  $f$ 
    else
         $\vec{g} = \text{fft}\left(\frac{n}{2}, (f_0, f_2, \dots, f_{n-2})\right)$ 
         $\vec{u} = \text{fft}\left(\frac{n}{2}, (f_1, f_3, \dots, f_{n-1})\right)$ 
        for  $k = 0$  to  $\frac{n}{2} - 1$ 
             $c_k = g_k + u_k \cdot e^{-j\frac{2\pi}{n} k \cdot \frac{n}{2}}$ 
             $c_{k+\frac{n}{2}} = g_k - u_k \cdot e^{-j\frac{2\pi}{n} k \cdot \frac{n}{2}}$ 
        return  $\vec{c}$ 
```



Wichtiges

DFT:

$$X[m] = \sum_{n=0}^{N-1} x[n] e^{-j \frac{2\pi}{N} mn}$$

IDFT:

$$x[n] = \frac{1}{N} \sum_{m=0}^{N-1} X[m] e^{j \frac{2\pi}{N} mn}$$

DFT berechnet aus N Abtastwerten $x[n]$ im Zeitfenster der Länge $T_{DFT} = N \cdot T_s$ N äquidistante (komplexe) Spektralwerte im f-Bereich $[0, f_s]$

Zeitfenster: $T_{DFT} = N \cdot T_s = N / f_s$

Frequenzauflösung: $\Delta f = f_s / N = 1 / (NT_s) = 1 / T_{DFT}$

z - Transformation

ist aus der Laplace-Transformation entstanden
erfüllt ähnlichen Zweck wie LT, aber für zeitdiskrete Signale

ideal abgetastetes Signal und dasson LT:

$$x_s(t) = x(t) \mid_{t=t_s} = x(t) \sum_{n=-\infty}^{\infty} \delta(t-nT_s) = \sum_{n=-\infty}^{\infty} x(nT_s) \delta(t-nT_s) = \sum_{n=-\infty}^{\infty} x[n] \delta(t-nT_s)$$

$$\bullet X_s(s) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-s \cdot nT_s}$$

wenn man oben e^{sT_s} durch z ersetzt erhält man die z-Transformation:

$$X(z) := \sum_{n=-\infty}^{\infty} x[n] \cdot z^{-n}$$

sie existiert nur in ihrem Konvergenzbereich

die FT erhält man aus der zT folgendermassen: $X(f) = X(z=e^{j2\pi f T_s})$

→ man muss also $X(z)$ auf dem Einheitskreis $z=e^{j2\pi f T_s}$ auswerten um $X(f)$ zu erhalten

→ $X(f)$ ist dann wie erwartet periodisch mit f_s (Spiegelachsen)

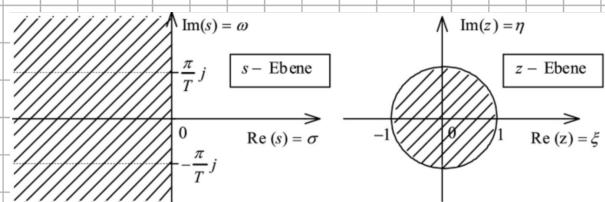
für die Bestimmung von $x[n]$ aus $X(z)$ gibt es mehrere Möglichkeiten:

- $x[n] = \frac{1}{2\pi j} \oint_C X(z) \cdot z^{n-1} dz$ dabei muss C vom Ursprung gehen und im Gegenurzeigersinn durchlaufen werden
- Integral-Inversionsformel
- Fortlaufende Division
- Partialbruchzerlegung → für jeden Partialbruch resultiert im Zeitbereich eine geometrische Reihe:

$$H(z) = A_0 + \sum_{k=1}^K \frac{A_k \cdot z}{z-p_k} \quad \bullet \circ h[n] = A_0 \cdot \delta[n] + \sum_{k=1}^K A_k \cdot p_k^n \quad (n \geq 0)$$

Eigenschaften der z-Transformation

- Linearität: $a_1 \cdot x_1[n] + a_2 \cdot x_2[n] \circ \bullet a_1 \cdot X_1(z) + a_2 \cdot X_2(z)$
- Zeitverschiebung: $x[n-k] \circ \bullet z^{-k} \cdot X(z)$
- Faltung: $x[n] * h[n] \circ \bullet X(z) \cdot H(z)$
- Multiplikation mit einer Exponentielle: $a^n \cdot x[n] \circ \bullet X(z/a)$
- Multiplikation mit der Zeit: $n \cdot x[n] \circ \bullet -z \cdot dX(z)/dz$
- Zeitumkehr $x[-n] \circ \bullet X^*(1/z^*)$
- Anfangswerttheorem für einseitige zT $x[0] = \lim_{z \rightarrow \infty} X(z)$
- Endwerttheorem für einseitige zT $\lim_{n \rightarrow \infty} x[n] = \lim_{z \rightarrow 1} (z-1) \cdot X(z)$



wichtige z-Transformierte

$$x[n] \xrightarrow{\quad} X(z)$$

$x[n]$	$X(z)$	ROC
$A\delta[n]$	A	Everywhere
$A\delta[n - n_0]$	Az^{-n_0}	Everywhere
$A \cdot u[n]$	$\frac{A}{1-z^{-1}} = \frac{Az}{z-1}$	$ z > 1$
$Aa^n \cdot u[n]$	$\frac{A}{1-az^{-1}} = \frac{Az}{z-a}$	$ z > a $
$Aa^n e^{j\omega n} \cdot u[n]$	$\frac{A}{1-ae^{j\omega}z^{-1}} = \frac{Az}{z-ae^{j\omega}}$	$ z > a $
$Aa^n \cos(\omega n) \cdot u[n]$	$\frac{Az(z-a \cos \omega)}{z^2 - (2a \cos \omega)z + a^2}$	$ z > a $
$Aa^n \cos(\omega n + \alpha) \cdot u[n]$	$\frac{\frac{A}{2}e^{j\alpha}}{1-ae^{j\omega}z^{-1}} + \frac{\frac{A}{2}e^{-j\alpha}}{1-ae^{-j\omega}z^{-1}}$	$ z > a $
$A \cos(\omega n) \cdot u[n]$	$\frac{Az(z-\cos \omega)}{z^2 - (2 \cos \omega)z + 1}$	$ z > 1$
$A \sin(\omega n) \cdot u[n]$	$\frac{Az \sin \omega}{z^2 - (2 \cos \omega)z + 1}$	$ z > 1$
$An \cdot u[n]$	$\frac{Az^{-1}}{(1-z^{-1})^2} = \frac{Az}{(z-1)^2}$	$ z > 1$
$An^2 \cdot u[n]$	$\frac{Az^{-1}(1+z^{-1})}{(1-z^{-1})^3} = \frac{Az(z+1)}{(z-1)^2}$	$ z > 1$
$Ana^n \cdot u[n]$	$\frac{Aaz^{-1}}{(1-az^{-1})^2} = \frac{Aaz}{(z-a)^2}$	$ z > a $
$Aa^n \cdot u[-n-1]$	$\frac{-A}{1-az^{-1}} = \frac{-Az}{z-a}$	$ z < a $

Digitale Systeme

manchmal auch „Digitalfilter“ genannt
 ↳ auch wenn Filter funktionieren nicht im Vierlag und stabil



hier geht es um (lineare, zeitinvariante, zeitdiskrete (LTD) Systeme:

$$L: x_1[n] \rightarrow y_1[n] \quad \& \quad x_2[n] \rightarrow y_2[n] \Rightarrow c_1 x_1[n] + c_2 x_2[n] \rightarrow c_1 y_1[n] + c_2 y_2[n]$$

$$T: x[n] \rightarrow y[n] \Rightarrow x[n-k] \rightarrow y[n-k] \quad \forall k \quad (\text{Systemparameter bzw. -koeffizienten bleiben konstant})$$

es ist kausal falls: aktueller Eingangswert beeinflusst nur aktuellen und zukünftige Ausgangswerte

es ist stabil falls: es beschränkte Eingangsfolgen ($|x[n]| < \infty \forall n$) in beschränkte Ausgangsfolgen transformiert

Differenzengleichung

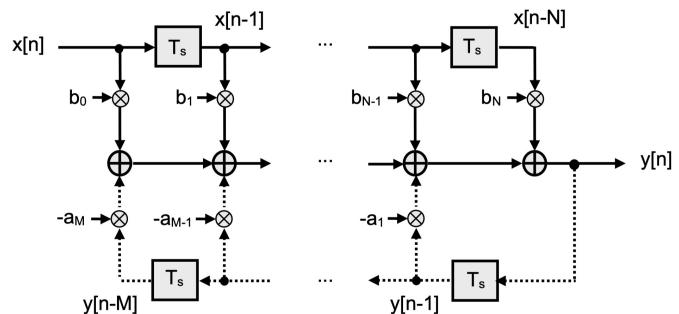
$$y[n] = \sum_{k=0}^N b_k \cdot x[n-k] - \sum_{k=1}^M a_k \cdot y[n-k]$$

analog zu Differenzengleichung bei analogen Systemen

man unterscheidet 2 Klassen von LTD-Systemen:

Nichtrekursive LTD-Systeme (Finite Impulse Response (FIR) Filter)

- $a_k = 0 \quad \forall k \in \{1, \dots, M\}$ (gestrichelter Teil in Abb. fehlt)
- Impulsantwort endlich lang
- o auch Transversalfilter, analog angelebt MA-System, All-Zero-Fiter genannt
- M Polstellen im Ursprung

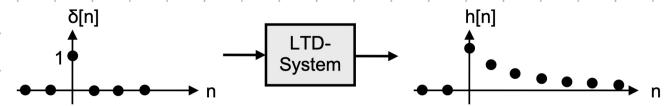


Rekursive LTD-Systeme (Infinite Impulse Response (IIR) Filter)

- aktueller Ausgangswert auch von vergangenen Ausgangswerten abhängt
- Impulsantwort normalerweise unendlich lang
- gestrichelter eingezeichnetes Feedback-Teil manchmal auch Auto-Regressiv- bzw. AR-System oder All-Pole Filter genannt

Impulsantwort h[n]

Reaktion auf Kronecker-Delta bzw. Einheitsimpuls $\delta[n] := 1$ falls $n=0$, 0 falls $n \neq 0$



aufgrund der L und T kann mit $h[n]$ die Reaktion $y[n]$ auf ein beliebiges $x[n]$ berechnet werden:

$$x[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot \delta[n-k] \rightarrow \text{LTD-System} \rightarrow y[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h[n-k] \stackrel{n=n-k}{=} \sum_{k=-\infty}^{\infty} h[k] \cdot x[n-k]$$

die Summen im Ausdruck rechts werden Faltungssummen genannt

→ Impulsantwort $h[k]$ (oder alternativ das Eingangssignal $x[k]$) muss zur Auswertung von $y[n]$ in der Zeit umgedreht bzw. gefaltet und n Positionen nach rechts

für FIR-Filter wird die Impulsantwort direkt durch die Filterkoeffizienten gebildet:

$$h_{\text{FIR}}[n] = \{b_0, b_1, \dots, b_M, b_N\} \quad \text{für } n = 0, \dots, N$$

Übertragungsfunktion

z -Transformiert $H(z)$ der Impulsantwort wird (z -) Übertragungsfunktion genannt

Berechnung:

$$h[n] = \sum_{k=-\infty}^{\infty} h[k] \delta[n-k] \quad \rightarrow \quad H(z) = \sum_{k=-\infty}^{\infty} h[k] z^{-k}$$

$$\Leftrightarrow \text{gilt: } H(z) = \frac{Y(z)}{X(z)}$$

mithilfe der Differenzengleichung kann man $H(z)$ in Funktion der Filterkoeffizienten bestimmen:

$$\begin{aligned} y[n] &= \sum_{k=0}^N b_k \cdot x[n-k] - \sum_{k=1}^M a_k \cdot y[n-k] \\ \Leftrightarrow Y(z) &= \sum_{k=0}^N b_k \cdot z^{-k} \cdot X(z) - \sum_{k=1}^M a_k \cdot z^{-k} \cdot Y(z) \quad \Leftrightarrow \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k \cdot z^{-k}}{1 + \sum_{k=1}^M a_k \cdot z^{-k}} \\ H(z) &= \frac{b_0 + b_1 \cdot z^{-1} + \dots + b_N \cdot z^{-N}}{1 + a_1 \cdot z^{-1} + \dots + a_M \cdot z^{-M}} \end{aligned}$$

Pol-Nullstellen-Darstellung:

$$H(z) = b_0 \frac{\prod_{k=1}^N (z - z_k)}{\prod_{k=1}^M (z - p_k)} z^{M-N} \quad \text{stabil falls alle } p_k \text{ innerhalb Einheitskreis}$$

$$\text{Frequenzgang: } H(f) = H(z = e^{j2\pi f T_s}) = b_0 \frac{\prod_{k=1}^N (e^{j2\pi f T_s} - z_k)}{\prod_{k=1}^M (e^{j2\pi f T_s} - p_k)} e^{j2\pi f T_s(M-N)}$$

$$\text{Betragsspektrum: } |H(f)| = K \frac{\prod_{k=1}^N |e^{j2\pi f T_s} - z_k|}{\prod_{k=1}^M |e^{j2\pi f T_s} - p_k|} \quad \text{«Abstände von } e^{j2\pi f T_s} \text{ zu Nullstellen multiplizieren, Abstände zu Polen dividieren»}$$

Korrelation

unnormalierte (Kreuz-)Korrelation zwischen 2 Signalen $x[n]$ und $y[n]$ ist definiert als:

$$r_{xy}[m] = \sum_{n=-\infty}^{\infty} x[n] \cdot y[n+m]$$

Mass für Ähnlichkeit von $x[n]$ und $y[n]$

falls $x[n]$ im Zeitintervall $n=0, \dots, N$ besteht, $y[n]$ davon und m lang ist gilt

$$r_{xy}[m] = x[0] \cdot y[m] + x[1] \cdot y[m+1] + \dots + x[N] \cdot y[m+N]$$

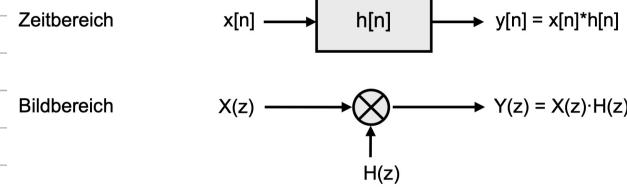
dies kann wie folgt umformuliert werden:

$$r_{xy}[m] = x[N] \cdot y[m+N] + \dots + x[1] \cdot y[m+1] + x[0] \cdot y[m] = x[N-m] * y[m]$$

\Rightarrow Faltung von $y[n]$ mit zeitlich umgedrehten $x[n]$, also $x[N-m]$

Transpositionstheorem für lineare Signalflussgraphen

«werden alle Signalflussrichtungen umgedreht, alle Addierer durch Knoten und alle Knoten durch Addierer ohne \ominus sowie Ein- und Ausgang vertauscht, dann bleibt die Übertragungsfunktion $H(z)$ unverändert»



Realisierungsstrukturen

theoretisch alle äquivalent, aufgrund von beschränkter Wortlänge in der Realität aber unterschiedlich:

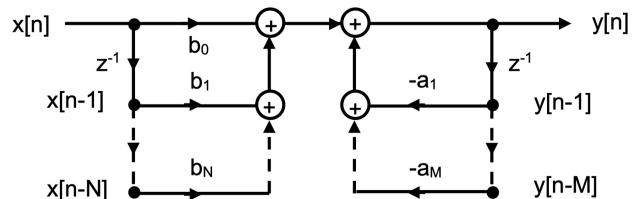
Kaskaden- und Parallelstrukturen verhalten sich im Bezug auf Quantisierungseffekte günstiger als Direktstrukturen

Direktstruktur 1:

- direkte Realisierung der Differenzengleichung
- wird nur für FIR-Filter verwendet

die beiden kaskadierten Strukturen können vertauscht werden ohne dass sich das Verhalten ändert

→ dann können über die beiden Verzögerungsketten zu einer zusammengefasst werden
→ es resultiert die



Direktstruktur 2:

- nur halb so viele Speicherzellen wie Direktstruktur 1
- Anzahl Speicherzellen unabh. von der Filterordnung

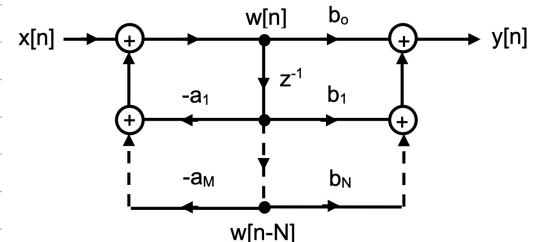
Schritt 0: (zirkulärer) w-Buffer $\{w[n-1], \dots, w[n-N]\}$ mit Nullen initialisieren

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)

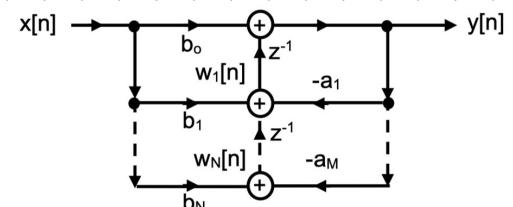
Schritt 4: neuen Eingangswert $x[n+1]$ lesen und mit Schritt 1 weiterfahren



transponierte Direktstruktur 2:

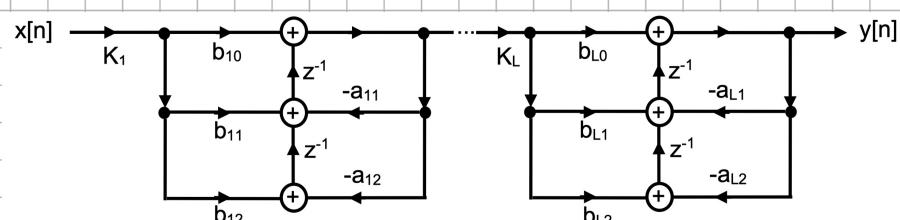
- Anwenden des Transponierungstheorems auf Direktstruktur 2
- Annahme: $M=N$

$$\begin{aligned} y[n] &= b_0 \cdot x[n] + w_1[n-1] \\ w_1[n] &= b_1 \cdot x[n] - a_1 \cdot y[n] + w_2[n-1] \\ &\vdots \\ w_{N-1}[n] &= b_{N-1} \cdot x[n] - a_{M-1} \cdot y[n] + w_N[n-1] \\ w_N[n] &= b_N \cdot x[n] - a_M \cdot y[n] \end{aligned}$$



Kaskadierung von Biquads:

- Realisierung der Pol-Nullstellen-Darstellung
- Zusammenfassung der komplex konjugierten Paare zu Kaskade von L reellen Biquads
- bei IIR-Filter ungerader Ordnung degeniert ein Biquad zu System 1. Ordnung
- $H(z) = K \cdot \frac{(z-z_1)(z-z_1^*)}{(z-p_1)(z-p_1^*)} \cdots \frac{(z-z_L)(z-z_L^*)}{(z-p_L)(z-p_L^*)} = K \cdot H_1(z) \cdots H_L(z)$
- Reihenfolge:
 - a) Der letzte Biquad enthält das komplexe Polpaar, das am nächsten beim Einheitskreis liegt, und das dazu nächstgelegene konjugiert komplexe Nullstellenpaar.
 - b) Die übrig gebliebenen Pole und Nullstellen werden nach der Regel a) kombiniert.



Parallelstruktur von Biquads:

- resultiert wenn man eine Partialbruchzerlegung der rationalen $H(z)$ durchführt
- $H(z) = \sum_{k=1}^L \left(\frac{A_k}{z-p_k} + \frac{A_k^*}{z-p_k^*} \right) = \sum_{k=1}^L H_k(z)$

Nichtdezimalzahlen

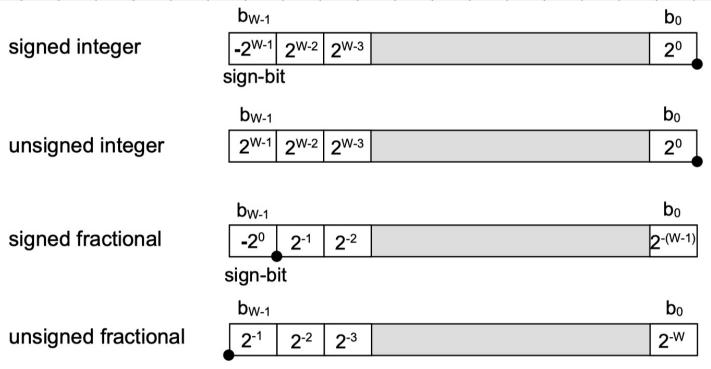
Festkoma - Zahlendarstellung

am häufigsten verwandte Kommastellen →

Vorteil: wenig Chipfläche \Rightarrow kostengünstig

Nachteile:

- Quantisierung der Filterkoeffizienten
- Überläufe
- Resultat der Multiplikation von 2 Zahlen der Länge W i.A. zu lang



Gleitkoma - Zahlendarstellung

am Beispiel des IEEE-Standards 754/854

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

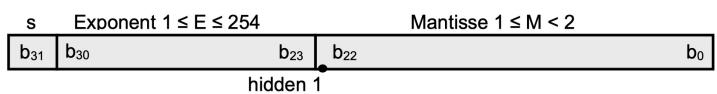
E = 0 und E = 255 für Null, keine Zahl (nan), Unendlich reserviert

Vorteile:

- Dynamikubereich sehr gross, auch befragsmässig sehr kleine Zahlen können dargestellt werden
- Überlaufproblem im Wesentlichen nicht vorhanden

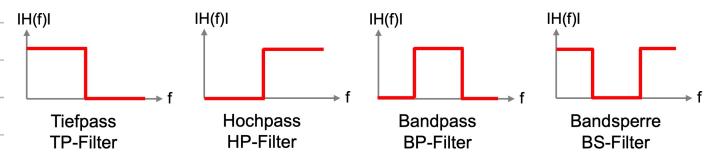
Nachteile:

- teurer als Festkoma
- kleine Quantisierungsfehler bei Addition / Multiplikation von 2 Zahlen



Digitalfilterentwurf

im klassischen Digitalfilterentwurf werden primär ideale TP, HP, BP, BS approximiert



ideale Filter n. möglich, da rechteckförmigen Frequenzbereich \Leftrightarrow sinc-förmig im Zeitbereich (als anal.)

diese Filter werden im Frequenzbereich spezifiziert

FIR-Filter: werden meist in der Transversalstruktur bzw. Direktform I realisiert

- ⊕ man kann vollständig lineare Phasengänge realisieren
- ⊕ immer stabil (alle Pole innerhalb Einheitskreis)
- ⊕ toleranter gegenüber Quantisierungseffekten als IIR-Filter
- ⊖ weisen höhere Filterordung auf als vergleichbare IIR-Filter und benötigen damit mehr Realisierungsaufwand
→ können andererseits sehr effizient in DSPs implementiert werden
- ⊖ Zeitverzögerung bzw. Gruppenlaufzeit ist meistens relativ gross

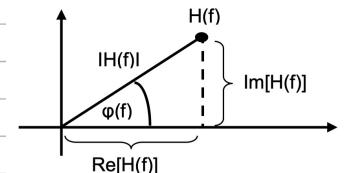
IIR-Filter: werden meistens als Biqdard-Kaskade realisiert

- ⊕ kleiner Realisierungsaufwand bzw. kleine Filterordnung durch die Selektivität der Pole
- ⊕ kleine Zeitverzögerungen
- ⊕ linearer Phasengang für kausale Filter
- ⊖ mehr Probleme mit Quantisierungseffekten (Grenzyklen, Rundungsrauschen) und dann Ableitungen von Störungen

Frequenzgang eines Digitalfilters

Fouriertransformierte der Impulsantwort $h[n]$

periodisch mit Abtastfrequenz f_s



eine einzelne Frequenzkomponente bei f_0 wird in der Amplitude mit $|H(f_0)|$ gewichtet und erleidet eine Phasenverschiebung von $\varphi(f_0)$ bzw. eine Zeitverzögerung von Δ_0 :

$$\cos(2\pi f_0 \cdot nT_s) \xrightarrow{H(f)} |H(f_0)| \cdot \cos[2\pi f_0 \cdot nT_s + \varphi(f_0)] = |H(f_0)| \cdot \cos[2\pi f_0 \cdot (nT_s - \Delta_0)]$$

$$\Delta_0 = T_p(f_0) = \frac{-\varphi(f_0)}{2\pi \cdot f_0}$$

unterschiedlich verzögerte Frequenzkomponenten können sich zu störenden Signalverzerrungen summieren

Phasenverschiebung: $\varphi(f_0)$

Zeitverzögerung: $T_p(f_0) = \frac{-\varphi(f_0)}{2\pi \cdot f_0}$

Gruppenlaufzeit $T_g(f_0) = \frac{-\partial \varphi(f_0)}{2\pi \cdot \partial f_0}$

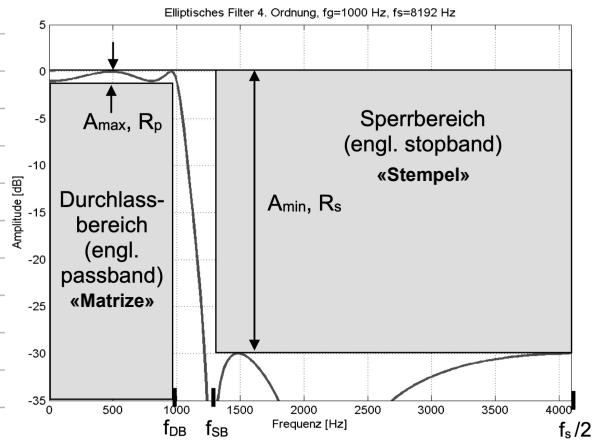
wenn der Phasengang in gewissem Frequenzbereich linear ist, bzw. $\varphi(f) = -K \cdot f$ gilt, wobei K eine Konstante ist, werden gemäss der Formel oben alle betreffenden Frequenzkomponenten beim Durchlaufen des Filters um die gleiche Konstante Zeit $\Delta = K / (2\pi)$ verzögert

mit (kausalen) IIR-Filtren können keine vollständig linearen Phasengänge realisiert werden, mit FIR-Filtren hingegen schon!

Filterspezifikation

Amplitudengang wird oft mit Stempel-Matrizen-Schema spezifiziert:

- Durchlassbereich
 - maximal tolerierbare Dämpfung A_{max} bzw. Rippel R_p gegenüber einer bestimmten Referenz
- Eckfrequenz f_{DB}
- Sperrbereich
 - minimal erforderliche Dämpfung A_{min} bzw. Rippel R_s gegenüber einer bestimmten Referenz
- Eckfrequenz f_{SB}



Filterordnung und Realisierungsaufwand hängen hauptsächlich von Steilheit der Filter flanke zwischen Durchlass- und Sperrbereich ab

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

wenn Linearität des Phasengangs wirklich wichtig ist und Zeitverzögerung nicht allzu klein sein muss wählt man normalerweise ein FIR-FILTER

FIR-Filtekentwurf

Typisierung von FIR-Filtern

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

Phasengang ist vollständig linear, wenn die Filterkoeffizienten entweder symmetrisch oder antisymmetrisch sind, d.h. wenn $b_n = b_{N-n}$ oder $b_n = -b_{N-n}$ für $n = 0, \dots, N$

Zeitverzögerung bzw. Gruppenlaufzeit eines linearphasigen FIR-Filters ist für alle Frequenzkomponenten gleich und beträgt $\Delta = \frac{1}{2}NT_s$

4 Typen linearphasiger FIR-Filter:

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

Fenstermethode (Winddoung)

Methode zur Bestimmung der Filterkoeffizienten von FIR-Filtern

da bei FIR-Filter die Filterk. b_n aber direkt mit der Impulsantwort $h[n]$ übereinstimmen kann man die Koeff. anhand einer Referenz-Stossantwort wie folgt bestimmen

1. Abtasten der analogen (Referenz-)Stossantwort $h(t)$:

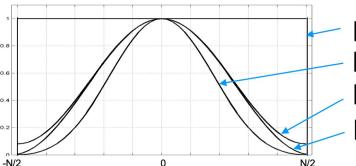
$$h_d[n] = T_s \cdot h(t=nT_s), \quad -\infty < n < \infty \quad \text{digitale Impulsantwort}$$

Normierung mit T_s ist erforderlich, weil Spektrum des abgetasteten Signals im Nyquistband mit Spektrum des analogen Signals übereinstimmen soll, $h_s(t) = h(t=nT_s)$ aber ein mit $1/T_s$ verzerrtes Spektrum aufweist: $H_s(f) = \frac{1}{T_s} \sum H(f-nf_s)$

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

$$h_c[n] = w[n] \cdot h_d[n], \quad -N/2 \leq n \leq N/2 \quad \text{abgeschnittene Impulsantwort}$$

$w[n]$ ist eine Fensterfunktion welche nur für n Zeitschritte im Bereich $-N/2 \leq n \leq N/2$ von Null verschieden ist:



- Rechteckfenster:
- Blackmann-Fenster:
- Hamming-Fenster:
- Hanning-Fenster:

$$\begin{aligned} w[n] &= 1, \quad 0 \leq n \leq N \\ w[n] &= 0.42 - 0.5 \cos(2\pi n/N) + 0.08 \cos(4\pi n/N), \quad 0 \leq n \leq N \\ w[n] &= 0.54 - 0.46 \cos(2\pi n/N), \quad 0 \leq n \leq N \\ w[n] &= 0.5 \cdot [1 - \cos(2\pi n/N)], \quad 0 \leq n \leq N \end{aligned}$$

3. FIR-Filter durch Zeitverschiebung der abgeschnittenen Impulsantwort herstellen:

$$b_n = h_{FIR}[n] = h_c[n-N/2]$$

die b_n sind sym. oder anti-sym. bezügl. $h[N/2]$, wenn analoge Stossantwort $h(t)$ gerade oder ungerade Funktion ist und Fensterfunktion $w[n]$ gerade ist

Bestimmen der analogen Stossantwort $h(t)$ bzw. der gewünschten digitalen Impulsantwort $h_d[n]$ im 1ten Schritt ist i.A. nicht einfach
→ für gewisse Referenzfilter liegt $h(t)$ über analytisch vor

Referenzfilter idealer TP:

$$H(f) = r_0(f/(2f_{\text{fs}})) \rightarrow 2f_{\text{fs}} \frac{\sin(\pi f/2f_{\text{fs}})}{\pi f/2f_{\text{fs}}} = \frac{\sin(\pi f/2f_{\text{fs}})}{\pi f} = h(t)$$

mit f_s abtasten und T_s normieren ergibt:

$$h_d[n] = \frac{\sin(\pi n T_s / 2f_{\text{fs}})}{\pi n}, -\infty \leq n \leq \infty$$

ausgehend von dieser Gleichung können beliebige linearphasige FIR-TP-Filter entworfen werden

beim Rechteckfenster tritt das Gibbs'sche Phänomen auf, welches auch für grosse N nicht verschwindet

es entsteht wegen dem abrupten Abschneiden der Impulsantwort, wie man es am besten im Spektrum erkennt: $h_d[n] \cdot w[n] \rightarrow H(f) * W(f)$

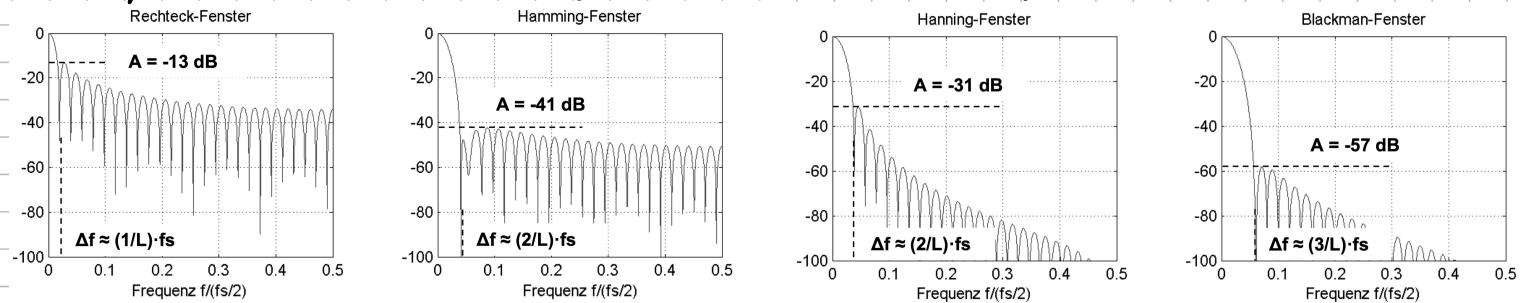
$W(f)$ ist für rechteckiges Fenster sinc-förmig, die Faltung mit dem gewünschten Frequenzgang $H(f)$ führt also zu einem Überschwingen im Übergang zw. DB und SB

das Gibbs'sche Phänomen kann mit weitem Abschneiden bzw. mit ausläufenden Fenster verhindert werden, allerdings resultiert bei diesen eine geringere Filterstabilität

es gelten folgende 2 Zusammenhänge:

- Nebenkeule von $W(f)$ klein \Rightarrow Überschwingungen von $H_{\text{FIR}}(f)$ klein
- Hauptkeule von $W(f)$ schmal \Rightarrow Übergangsbereich von $H_{\text{FIR}}(f)$ steil

Betragsspektrum $|W(f)|$ der oben abgebildeten Fenster der Länge $L = N + 1$:



Um nun auch die anderen Filtertypen zu realisieren sucht man Transformationen, welche die lineare Phase erhalten

TP-BP-Transformation:

mithilfe der Frequenzverschiebungseigenschaft der FT kann man einen TP mit Grenzfrequenz f_{fs} in einen BP mit Mittelfrequenz f_0 und unter bzw. oberer Grenzfrequenz $f_0 \pm f_{\text{fs}}$ transformieren

$$\text{Typ 1,2: } b_{\text{BP}}[n] = 2 \cos(\omega_0 \cdot n T_s) \cdot b_{\text{TP}}[n]$$

$$\text{Typ 3,4: } b_{\text{BP}}[n] = 2 \sin(\omega_0 \cdot n T_s) \cdot b_{\text{TP}}[n]$$

TP-HP-Transformation:

$$\text{TP-BP-Trafo mit } f_0 = f_s/2: b_{\text{HP}}[n] = (-1)^n \cdot b_{\text{TP}}[n]$$

BP-BS-Transformation:

BS können als Komplementär filter zu BP realisiert werden

$$H_{\text{BS}}(z) + H_{\text{BP}}(z) = z^{-N/2} \Rightarrow b_{\text{BS}}[n] = \delta[n-N/2] - b_{\text{BP}}[n]$$

neben des Referenz-TP gibt es auch andere interessante Referenz-Filter, wie zB Differentiator, Hilbertfilter oder raised-cosine-Pulse

Frequenzabtastung (Frequency Sampling)

es werden $N+1$ äquidistante Abtastpunkte des gewünschten Frequenzgangs $H(f)$ vorgegeben mithilfe der IDFT bzw. IFFT kann dann die Impulsantwort $h[n]$, $n = 0, \dots, N$ des FIR-Filters bestimmt werden

$H_{FIR}(f)$ hält die Vorgabepunkte natürlich genau ein, dazwischen kann es aber „Überschüsse“ geben, vor allem ein Übergang zwischen Durchlass- und Sperrbereich

es können auch „belädt“ komplexe Impulsantworten $h_d[n]$ generiert und als Input für die Festermethode verwendet werden

Entwurf im z-Bereich

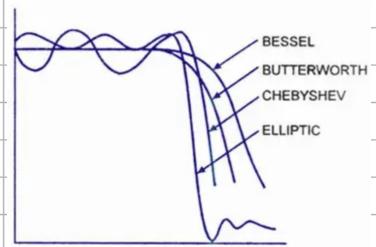
gibt einige iterative Optimierungsverfahren:

- Remez-Algorithmus von Parks-McClellan:
minimiert maximale Abweichung zw. Amplitudengang und der Vorgabe (gegeben als (Multi-)Stempelmatrix-Schema)
resultierende FIR-Filter nutzen gleichmäßige Rippel im Durchlass- und im Sperrbereich aus
→ werden daher auch Equiripple FIR-Filter genannt
bei geplanter minimaler Sperrbereichsdämpfung ist die Filterordnung kleiner als bei allen anderen FIR-Filter Entwurfmethoden
- Least-Square Verfahren:
minimiert mittlere quadratische Abweichung zwischen Amplitudengang und Vorgabe

IIR-Filfilterentwurf

Entwurf mit analogen Prototyp-Filttern

Approximation von idealen „Rectangular“-Filttern ist im Analogen auf verschiedene Arten gelöst werden.
→ daraus kann man auch den Digitalen profitieren

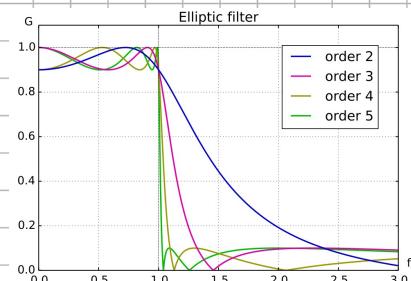


Beim klassischen IIR-Filfilterentwurf geht man vom analogen Prototyp-TP-Filter aus, nimmt allenfalls eine TP-HP/BP/BS Transformation vor und transformiert dann das resultierende Filter ins Digitale

die am häufigsten eingesetzte analog-digital- bzw. s2z-Transformation ist die bilineare Transformation, seltener die impulse-invariante Transformation

analoge Prototyp-Filter:

	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 $\phi(f)$	gross	mittel	mittel	klein	sehr gross



elliptische Filter (Cauer-Filter) am steilsten, hat aber gleichmässige Rippel im BB und SB
→ diese sind das Resultat von Nullstellen der UTF $H(z)$ auf Einheitskreis

Konvention: auch bei BP und BS wird mit Filterordnung die Ordnung des analogen Prototyp-Filters bezeichnet, obwohl bei TP-BP/BS Trafo die eigentliche Ordnung verdoppelt wird

Ziel der bilinearen Trafo ist es, die eigentliche s2z-Trafo $z = e^{sT_s}$ so zu approximieren, dass eine realisierbare bzw. rational gebrochene Funktion $H(z)$ resultiert
dazu benutzt man Taylorpolynome 1^{er} Ordnung:

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

durch auflösen nach s erhält man die bilineare Trafo

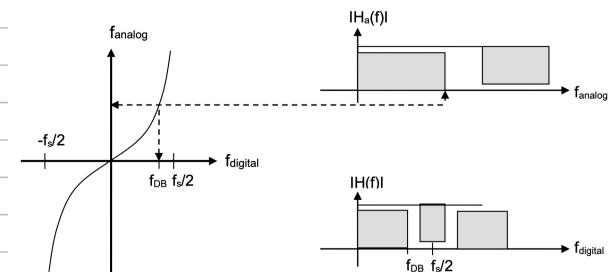
$$s = \frac{2}{T_s} \cdot \frac{z - 1}{z + 1}$$

wird bilinear genannt weil Zähler und Nenner je linearer Funktionen sind, die Trafo selbst ist aber nicht linear!

die Eigenschaften erkennt man, indem man die Frequenz-Trafo analysiert, d.h. $s = j\omega T_{\text{analog}}$ und $z = \exp(j\omega T_{\text{digital}})$ in obige Gleichung eingesetzt:

$$j\omega T_{\text{analog}} = j \frac{2}{T_s} \tan(\pi f_{\text{digital}} T_s)$$

die ganze analoge Frequenzachse wird nur einmal in 1. Nyquist-Zone $-f_s/2 \leq f \leq f_s/2$ abgebildet



bei der bilinearen Trafo entsteht kein Aliasing, stattdessen Frequenzverzerrung

Eckfrequenzen des DB und SB des analogen Prototyp-Filters müssen deshalb gemäß obiger Formel vorverzerrt werden (prewarping)

Direkter Entwurf im Diskreten

IIR-Filter können mit Computer-Hilfe auch direkt im Diskreten entworfen werden

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

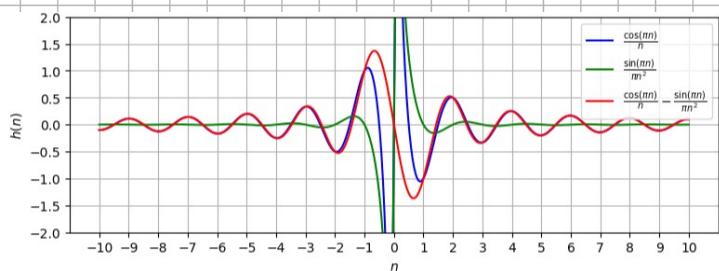
es können auch nicht-standard-Filtertypen wie Multiband-FILTER realisiert werden

Spezielle FIR-Filter

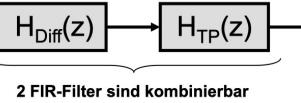
Differenzfilter

mit f_s abgetastet und T_s normierte Stoßantwort $h_d[n]$ eines idealen (analogen) Differenzfilters mit Amplitudengang $H_d(f) = j2\pi f$:

$$h_d[n] = \cos(n\pi)/n - \sin(n\pi)/(n^2\pi), \quad -\infty < n < \infty, \text{ wobei } h_d[0] = 0 \text{ ist}$$



bei Differentiation hat man oft das Problem, dass man das Höchstfrequenz-Rauschen ungewollt verstärkt
 → kann man verhindern indem man gleichzeitig TP-Filtrierung einsetzt



je höher Ordnung eines FIR-Differenzfilters, desto länger folgt der Amplitudengang dem idealen Amplitudengang

alle mit dater Formel entworfene FIR-Differenzfiltern haben antisymmetrische Koeffizienten

falls die Ordnung gerade ist (Typ 3) haben sie ohne Nullstelle bei $f_s/2$

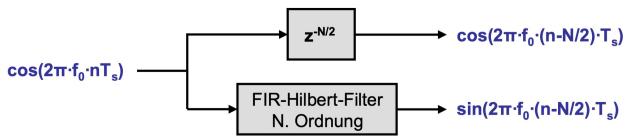
falls Ordnung ungerade (Typ 4) können sie dem idealen Differenzfilter bis zu ganz hohen Frequenzen folgen

Hilbert-Filter

in Nachrichtentechnik werden breitbandige 90° -Phasenschreiber bzw. Hilbert-Filter eingesetzt um bspw. aus einem Cosinus- oder Sinus-Signal der gleichen Frequenz zu generieren

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

$$h_d[n] = (1 - \cos(n\pi))/(n\pi), \quad -\infty < n < \infty, \text{ wobei } h_d[0] = 0 \text{ ist}$$



ein Signal $z(t)$, das keine negativen Frequenzkomponenten aufweist, nennt man analytisches Signal
 → jedes analytische Signal muss komplexeartig sein!

jedes reellwertige Signal $x(t)$ kann in ein zugehöriges analytisches Signal $z(t)$ konvertiert werden indem man alle Komponenten um 90° phasenverschiebt, mit j multipliziert und dann zu sich selbst daran addiert.

$$z(t) = x(t) + j H(x(t))$$

$$\text{env}(x(t)) = \text{abs}(\text{hilbert}(x(t)))$$

Raised-cosine-Filter

in Nachrichtentechnik werden praktisch nie rechteckförmige Bit-Signale verwandt, da das zugehörige $\sin(x)/x$ -förmige Spektrum viel zu breit ist

durch Verwendung von besser geeigneten Bit-Pulsesignalen können viel kompaktere Spalten realisiert werden, ohne dass die Datenrate reduziert werden müsste

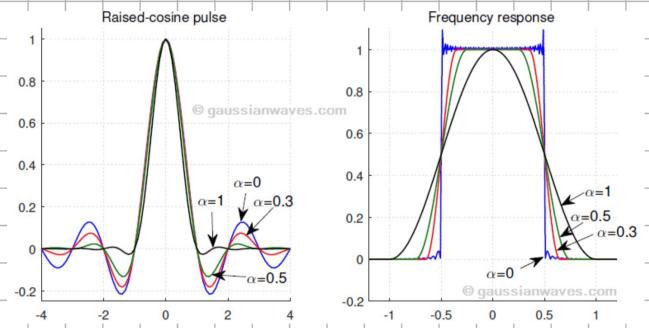
Generierung der Bit-Pulsesignale wird normalerweise als Puls-Shaping bezeichnet

in der Signalübertragung werden oft raised-cosine-Pulse eingesetzt:

$$p(t) = \frac{\sin(\pi t/T_{sym})}{\pi t/T_{sym}} \cdot \frac{\cos(\pi \alpha t/T_{sym})}{1 - (2\alpha t/T_{sym})^2}, \quad 0 \leq \alpha \leq 1$$

hierbei bezeichnet T_{sym} die Symboldauer, α nennt man Roll-off-Faktor

weiteres Pulse-Shaping-Filter, das in der Signalübertragung oft eingesetzt wird: Gauss-FIR-Filter



Kamm- und Schmalband-IFIR-Filter

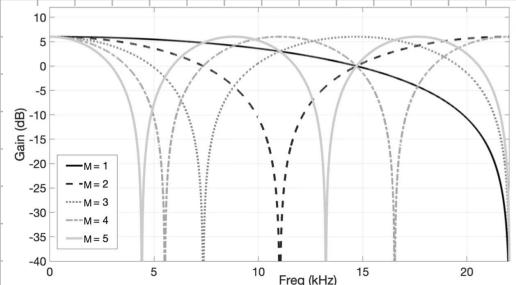
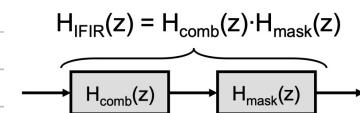
wenn man bei Digitalfilter $H(z)$ jedes Verzögerungselement z^{-1} durch Kaskade von M Verzögerungselementen ersetzt, erhält man das Kammfilter

$$H_{comb}(z) = H(z^M)$$

Frequenzbandbreite

es weist kammförmigen Frequenzgang auf, bei M Perioden des urspr. Digitalfilters $H(z)$ enthalten, weil $H_{comb}(f) = H(z = e^{j2\pi f M T_s})$

der Frequenzgang des urspr. Digitalfilters ist also in der Frequenz gestrafft, es hat nun die Bandbreite B/M , wobei B die Bandbreite des urspr. Filters $H(z)$ bezeichnet



durch Kaskadierung mit Maskierungsfilter lassen dann $M-1$ Kämme wegfiltert werden und man erhält ein Schmalbandfilter

weil das Maskierungsfilter die $M-1$ Nullen in der Impulsantwort des Kammfilters interpoliert spricht man auch von interpolated FIR-Filter bzw. IFIR-Filter:

$$H_{IFIR}(z) = H_{comb}(z) \cdot H_{mask}(z)$$

Echtzeit - Signalverarbeitung

direkte Filterung im Zeitbereich

bei FIR-Filters kann die Ausgangsfolge direkt im Zeitbereich bestimmt werden, indem die Eingangsfolge $x[n]$ mit der Impulsantwort $h[n]$ gefaltet wird:

$$y[n] = x[n] * h[n] = \sum_{k=0}^{N-1} h[k] \cdot x[n-k]$$

dabei bezeichnet N die Filterordnung und $N+1$ entspricht der Filterlänge bzw. der Anzahl „taps“ in der Verzögerungskette

ein DSP muss also $(N+1) \cdot f_s$ MAC-Operationen pro Sekunden ausführen

bei grossen FIR-Systemen wird dies schnell zum Problem

→ effizienter ist der fast-convolution-Algorithmus, lohnt sich schon ab $N > 100$

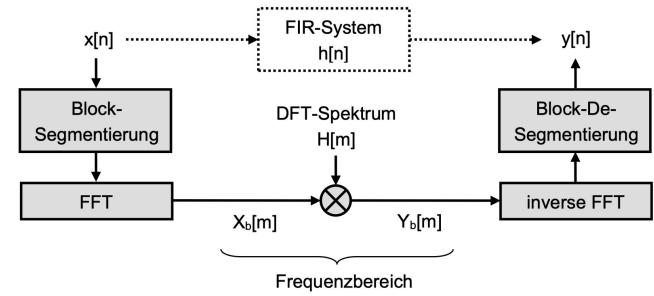
fast convolution

basiert darauf, dass

- Eingangsfolge $x[n]$ geziert in Blöcke der Länge M segmentiert werden
- Blöcke $x_b[n]$ je mit L -Punkt-FFT effizient im Frequenzbereich transformiert werden ($L \cdot \log_2(L)$ komplexe Multiplikationen)
- resultierende $X_b[m]$ mit DFT-Spektrum des FIR-Systems multipliziert werden (L komplexe Multiplikationen)
- resultierende $Y_b[m]$ mit IFFT zurück transformiert werden ($L \cdot \log_2(L)$ komplexe Multiplikationen)
- die Blöcke $y_b[n]$ geziert desegmentiert werden

ein Nachteil ist der Zeitzwischenraum 1 Block

für Verständnis des fast-convolution-Algorithmus und der Wahl der FFT-Länge L ist es wichtig zwischen zirkulärer und linearer Faltung zu unterscheiden



Zirkuläre und Lineare Faltung

Multiplikation zweier N -Punkt DFT-Spektren im Frequenzbereich $X[m] = X_1[m] \cdot X_2[m]$, $m = 0, \dots, N-1$ entspricht der zirkulären Faltung der Zeitsignale, d.h.

$$x[n] = x_1[n] \otimes x_2[n] = \sum_{k=0}^{N-1} x_1[k] \cdot x_2[(n-k) \bmod N], \quad n = 0, \dots, N-1$$

da die DFT-Spektren ja die Spektren der periodisch fortgesetzten Zeitschreiter-Folgen darstellen

für Echtzeit-Verarbeitung braucht man aber die übliche lineare Faltung ohne Modulo-Operation

damit die FFT trotzdem eingesetzt werden kann, bedient man sich folgenden Tricks: man kann x_1 und x_2 so mit Nullen ergänzen, dass der zirkulär verschobene Anteil beim zirkulären Falten keinen Einfluss hat

dadurch wird die DFT mindestens so lang wie die Summe der beiden Zeitsignalen minus 1, eben mindestens so lang wie die lineare Faltung von x_1 und x_2

Overlap-Add-Algorithmus

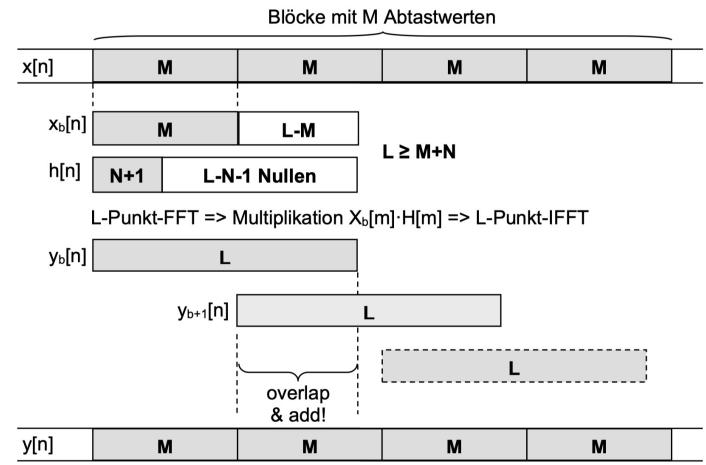
Annahmen:

- $x[n]$ im Zeitraum $[0, \infty)$ definiert
- FIR-Filter habe die Länge $N+1$

$x[n]$ in Blöcke mit M Abtastwerte segmentieren:

$$x[n] = \sum_{b=0}^{\infty} x_b[n-bM] \quad \text{wobei:}$$

$$x_b[n] = \begin{cases} x[n+bM] & 0 \leq n \leq M-1 \\ 0 & \text{sonst} \end{cases}$$



mit der Linearität der Faltung folgt

$$y[n] = x[n] * h[n] = \sum_{b=0}^{\infty} x_b[n-bM] * h[n] = \sum_{b=0}^{\infty} y_b[n]$$

$y[n]$ setzt sich also aus der Summe der linearen Faltungen der Eingangsblöcke $x_b[n]$ mit der Filter-Impulsantwort $h[n]$ zusammen

einzelne lineare Faltungen können nun mit zirkulären Faltungen realisiert werden

dazu müssen Eingangsblöcke $x_b[n]$ mit $L-M$ Nullen und Filter-Impulsantwort mit $L-N-1$ Nullen ergänzt, mit einer L-Punkt-FFT im Frequenzbereich transformiert, dort multipliziert und schließlich mit einer L-Punkt-IFFT wieder im Zeitbereich transformiert werden

damit die zirkuläre der linearen Faltung entspricht muss $L \geq M+N$ gelten

pro Input-Block $x_b[n]$ der Länge M resultiert ein Output-Block $y_b[n]$ der Länge L
 → die Output-Blöcke überlappen sich also!

wegen Superpositionsprinzip müssen die Output-Blöcke überlappend aufaddiert werden

es gibt noch einen weiteren Algorithmus zur Echtzeit-Filtrierung mit FFT (overlap-save), der genau dem gleichen Aufbau hat

Vergleich: Faltung mit FIR im Zeitbereich: $(N+1) \cdot f_s \text{ MAC pro Sekunde}$

↪ Fast-Conv. kostet sch. als $N > 100$

Korrelation

Kreuzkorrelation

Kreuzkorrelationsfunktion für Signale $x[n]$ und $y[n]$: $r_{xy}[m] = \sum_{n=-\infty}^{\infty} x[n+m] \cdot y[n]$

falls $y[n] = (y[0], \dots, y[N-1])$: $r_{xy}[m] = \sum_{n=0}^{N-1} x[n+m] \cdot y[n], m \geq -(N-1)$

es gilt $r_{xy}[m] = r_{yx}[-m]$, d.h. wenn x und y vertauscht werden, ist r_{xy} am Originus gespiegelt

$\curvearrowleft \curvearrowright$ lang

Autokorrelation

Autokorrelationsfunktion: $r_{xx}[m]$

↳ Maß für Ähnlichkeit eines Signals $x[n]$ mit einer um m Abstandintervalle verschobene Kopie von sich selbst
Wert der Autokorrelation ist maximal für $m=0$ und entspricht dann der Signalleistung:

$$r_{xx}[m=0] = \sum_{n=-\infty}^{\infty} x[n]^2$$

Zusammenhang Faltung

im Gegensatz zu Faltung ist Kreuzkorrelation nicht kommutativ

$r_{xy}[m]$ kann man durch Faltung von $x[n]$ mit $y'[n] = y[-n]$ berechnen:

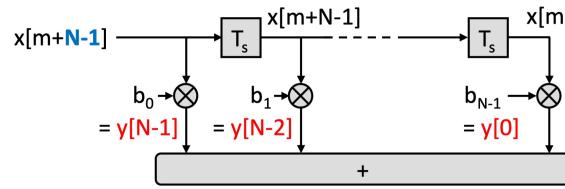
$$r_{xy}[m] = \sum_k x[k+m] \cdot y'[k] \stackrel{k=-n}{=} \sum_n x[m-n] \cdot y[-n] = x[n] * y'[n]$$

FIR-Korrelationsfilter

Ordnung: $N-1$

$$b_n = y[N-1-n], n=0, \dots, N-1$$

es wird das Ende des Musters $y[n], n=0, \dots, N-1$ im Signal $x[n]$ berücksichtigt



Matlab:
 $b = y(end:-1:1);$
 $% b = fliplr(y);$
 $r_{xy} = filter(b, 1, x);$

$$r_{xy}[m] = \sum_{n=0}^{N-1} x[n+m] \cdot y[n]$$

z-Trafo

$$R_{xy}(z) = \mathbb{E}\{r_{xy}[m]\} = \mathbb{E}\{x[n] * y[-n]\} = X(z) \cdot Y(\frac{1}{z}) = X(z) \cdot Y(z^{-1})$$

Spektrum: $R_{xy}(f) = X(f) \cdot Y^*(f)$ und $R_{xx}(f) = |X(f)|^2$

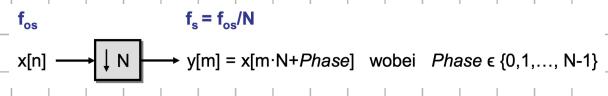
Multitakt-Signalverarbeitung

Downsampling

$$X(f) = \frac{1}{T_{os}} \sum_{n=-\infty}^{\infty} x[n] \delta(f - n f_{os})$$

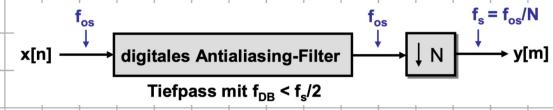
$$Y(f) = \frac{1}{N T_{os}} \sum_{n=-\infty}^{\infty} X_n(f - n f_{os}/N)$$

$$y[m] = \sum_{k=0}^{N-1} b_k \cdot x[mN - k]$$



Decimation

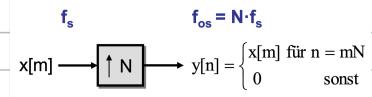
Kaskade Anti-Aliasing-TF-Filte & Downampler



Upsampling

$$Y(z) = \sum_{m=-\infty}^{\infty} y[m] z^{-m} = \sum_{n=-\infty}^{\infty} x[n] z^{-nN} = X(z^N)$$

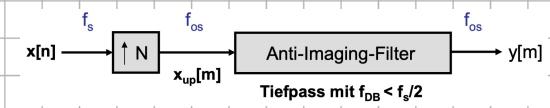
$$Y(f) = Y(z=e^{j2\pi f T_{os}}) \Rightarrow NT_{os} = T_s \models X(f) \text{ aber mit Images in 1. Nyquistzone}$$



Interpolation

Kaskade Upsampler & digitalem A-A-Filte

Images in 1. Nyquistzone vom Upsampler müssen mit digitalem A-A-Filte entfernt werden



$\Sigma\Delta$ -Wandler

