

Formelsammlung DSVB

Mario Felder
Michi Fallegger

Manuel Ziegler
Thomas Annen

19. Januar 2020

Inhaltsverzeichnis

1	Digitale Signale im Zeitbereich	5
1.1	Signal-Analyse	5
1.1.1	Sampling	5
1.1.2	Grundliegende Signale	5
1.1.3	Statistische Signalparameter	6
1.1.4	Messverhältnisse und dB	6
1.2	Signal Operationen	7
1.2.1	Korrelation (correlation)	7
1.2.2	Faltung (convolution)	7
2	Analog-Digital & Digital-Analog Wandlung	8
2.1	Schritte der A/D- und D/A-Wandlung	8
2.2	Digitale Signalprozessoren (DSP)	8
2.3	Abtasten und Aliasing	9
2.3.1	Abtasten von Tiefpass-Signalen	9
2.3.2	Aliasing & Gegenmassnahme	9
2.3.3	Abtasten von Bandpass-Signalen	10
2.4	Quantisierung von Signalen	10
2.4.1	Uniforme Quantisierung	10
2.4.2	Quantisierungsrauschen	11
2.4.3	Logarithmische Quantisierung	12
2.4.4	Dithering	12
3	Digitale Signale im Frequenzbereich	13
3.1	Von Fouriertransformation zur DFT	13
3.1.1	Übergang zu diskreter Zeit (DTFT)	13
3.1.2	Übergang zu endlichem Messintervall (DFT)	13
3.1.3	Vergleich der Fourier-Methoden	14
3.1.4	Intuitiver Ansatz zum Verständnis der DFT	14
3.2	Eigenschaften der DFT	15
3.2.1	Gültigkeitsbereich der DFT	15
3.3	Anwendungs-Aspekte	16
3.3.1	Fensterlänge & Leakage	16
3.3.2	Zero-Padding	16
3.3.3	Fensterfunktion	16
3.4	Short-Time DFT	17
3.5	Fast Fourier Transformation (FFT)	17
3.5.1	Twiddle-Faktoren	17
3.5.2	Radix-2 FFT	18
3.6	Goertzel-Algorithmus	19
4	Digitale LTI Systeme	20
4.1	Definition	20
4.2	Beschreibungsarten von LTI-Systemen	20
4.3	Systembeschreibung im Zeitbereich	20
4.3.1	Impulsantwort / Faltung	20
4.3.2	Differenzengleichung	20
4.3.3	Signalflussdiagramm	20
4.4	Systembeschreibung im Frequenzbereich	21
4.4.1	Übertragungsfunktion	21
4.4.2	Pol- Nullstellenplan	21
4.4.3	Frequenzgang	21

4.4.4 Beziehung zwischen Frequenzgang & Übertragungsfunktion	21
5 Digitales Filterdesign	22
5.1 Übertragungsfunktion	22
5.2 FIR (finite impulse response) Filter	24
5.2.1 Symmetrische FIR-Filter	24
5.2.2 Window Design Methode	25
5.2.3 Alternative Designmethoden	26
5.2.4 FIR-Kammfilter	26
5.2.5 Kanalausgleich mittels FIR	26
5.3 IIR-Filter	27
5.3.1 Eigenschaften	27
5.3.2 IIR-Design mittels Impulsinvarianz-Transformation (Tiefpass)	27
5.4 Implementationsaspekte	28
5.4.1 Implementation von IIR-Filttern	28
5.4.2 Fixkomma-Implementation	28
6 Multirate und Filterbänke	29
6.1 Downsampling/Decimation	29
6.2 Upsampling/Interpolation	30
6.3 Polyphasen Filter	31
6.4 Implementation rationaler Abtastraten	32
6.5 Quadratur Spiegel Filterbank	32
6.6 DFT Filterbank	33
7 Zufallssignale	36
7.1 Autokorrelation und Spektrum	36
7.1.1 Betrachtung im Zeitbereich	36
7.1.2 Betrachtung im Frequenzbereich	36
7.2 Spectral Shaping durch LTI Systeme	37
7.3 Lineare Modelle für Stochastische Prozesse	37
7.3.1 Filtern von White Noise	37
7.3.2 Noise Whitening	38
7.3.3 Moving average model (MA)	38
7.3.4 Autoregressive model (AR)	39
7.3.5 ARMA model	39
7.4 Abschätzung der Spektrums-Dichtefunktion	39
7.4.1 Parameterlose Methode	39
7.4.2 Parameterbehaftete Methode	39
8 Optimale Lineare Filter	40
8.1 Wiener Filter	40
8.2 Kalman-Filter	41
9 Adaptive Filter	42
9.1 Linear Predictive Coding (LPC-10e)	42
9.2 LMS Algorithmus (least mean square)	43
9.3 Echo cancellation	45
10 Neuronale Netze	46
10.1 Einführung	46
10.1.1 Anwendungsgebiete	46
10.1.2 Gründe für Einsatz & Popularität	46
10.1.3 Typen-Übersicht	46
10.1.4 Vergleich: Nature vs. AI Networks	47
10.2 Prinzipielle Architektur für Bildklassifikation	47
10.3 Layer-Übersicht	47

10.3.1 Convolutional Layer	47
10.3.2 Pooling Layer	49
10.3.3 Activation Layer	49
10.4 Layer-Entwicklung	49
11 Matlab	50

Kapitel 1

Digitale Signale im Zeitbereich

1.1 Signal-Analyse

1.1.1 Sampling

Die Sample-Frequenz f_S ist durch die Sample-Periode T_S gegeben:

$$f_S = \frac{1}{T_S}$$

Aus dem Signal $x(t)$ wird durch die Abtastung:

$$x(n \cdot T_S) = x[n]$$

Das Signal $x[n]$ ist kausal wenn:

$$x[n] = 0 \quad \text{für alle } n < 0$$

1.1.2 Grundliegende Signale

Einheitsimpuls oder Diracimpuls:

$$\delta = \begin{cases} 0 & : n \neq 0 \\ 1 & : n = 0 \end{cases}$$

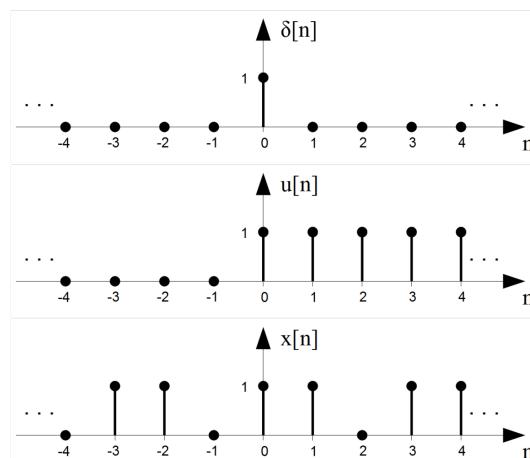
Einheitsschritt:

$$\sigma[n] \text{ oder } u[n] = \begin{cases} 0 & : n < 0 \\ 1 & : n \geq 0 \end{cases}$$

Ein periodisches Signal ist beschrieben durch:

$$x[n] = x[n + T_0/T_S] \quad \text{mit } T_0/T_S = k$$

Ein periodisches Signal muss eine Periodendauer T_0 haben, welche ein ganzzahliges Vielfaches der Sample-Periode T_S ist.



Komplexe harmonische Sequenz für Fourier-Analyse:

$$x[n] = \hat{X} \cdot e^{j2\pi f_0 n T_S}$$

Real- und Imaginäranteil der komplexen Harmonischen:

$$Re\{\} = \hat{X} \cdot \cos 2\pi f_0 n T_S$$

$$Im\{\} = \hat{X} \cdot \sin 2\pi f_0 n T_S$$

1.1.3 Statistische Signalparameter

Observations-Intervall T : Der Parameter N entspricht der Anzahl Samples.

$$T = N \cdot T_S$$

Der Mittelwert (**mean value**) μ_x repräsentiert den DC-Anteil des Signals $x[n]$:

$$\mu_x = \frac{1}{N} \sum_{i=0}^{N-1} x[i]$$

Der quadratische Mittelwert (**quadratic mean value**) ρ_x^2 korrespondiert zur durchschnittlichen Leistung des Signal $x[n]$ inklusive dem DC-Anteil:

$$\rho_x^2 = \frac{1}{N} \sum_{i=0}^{N-1} x[i]^2$$

Die **Varianz** σ_x^2 repräsentiert die durchschnittliche AC-Leistung des Signals $x[n]$:

$$\sigma_x^2 = \frac{1}{N} \sum_{i=0}^{N-1} (x[i] - \mu_x)^2$$

1.1.4 Messverhältnisse und dB

Leistungsverhältnis (**power ratio**):

$$A_{dB} = 10 \cdot \log_{10} \left(\frac{P_1}{P_2} \right)$$

Rauschabstand, Signal-to-noise ratio (**SNR**):

$$SNR_{dB} = 10 \cdot \log_{10} \left(\frac{P_{signal}}{P_{noise}} \right) = 20 \cdot \log_{10} \left(\frac{U_{signal}}{U_{noise}} \right)$$

Die Anwendung der Formel mit dem Spannungsverhältnis gilt nur bei Messung über der gleichen Impedanz!

Matlab-Befehle für Berechnungen mit Leistungsgrößen:

```
snr    = 10 * log10(sum(s.^2)/sum(r.^2)); (Anwenden des quadratischen Mittelwertes)
snrac = 10 * log10(sum((s-mean(s)).^2)/sum((r-mean(r)).^2)); (Anwenden der Varianz)
```

Matlab-Befehle für Berechnungen mit Signalgröße (Spannung **s** und Rauschen **r**):

```
snr = 20*log10(sqrt(sum(s.^2))/sqrt(sum(r.^2)));
```

Linear a/b	Power ratio [dB]	
	Power	Voltage
1/1000	-30	-60
1/100	-20	-40
1/10	-10	-20
1/2	≈ -3	≈ -6
1	0	0
2/1	≈ 3	≈ 6
10/1	10	20
100/1	20	40
1000/1	30	60

1.2 Signal Operationen

1.2.1 Korrelation (correlation)

Statische Korrelation (static correlation): drückt die Ähnlichkeit zweier Signale $x[n]$ und $y[n]$ der selben Länge N als Skalar aus. Bei ungleicher Länge ist zero-padding erlaubt.

$$R = \frac{1}{N} \sum_{i=0}^{N-1} x[i] \cdot y[i]$$

Je ähnlicher sich die Signale sind, desto grösser der Wert von R .

Lineare Korrelation (linear correlation): Falls gilt, dass $x[n] = y[n]$, spricht man von einer Autokorrelation (entspricht dem quadratischen Mittelwert und somit der durchschnittlichen Leistung), sonst auch von einer Kreuzkorrelation.

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

Die Korrelation ist nicht kommutativ, d.h. $r_{xy}[n] \neq r_{yx}[n]$.

Resultierende Länge von r_{xy} (N_{xy}) und Definitionsbereich:

$$N_{xy} = N_x + N_y - 1 \quad r_{xy}[n] \neq 0 \text{ gilt für } \{-N_x + 1 \leq n \leq N_y - 1\}$$

Matlab-Befehl für Kreuz- und Autokorrelation: `xcorr()`

1.2.2 Faltung (convolution)

Die Faltung ist der Korrelation ähnlich, das verzögerte Signal wird jedoch an der Y-Achse ($n = 0$) gespiegelt. Definition:

$$z[n] = \sum_{i=-\infty}^{\infty} x[i] \cdot y[-i+n]$$

Die Faltung ist kommutativ:

$$z[n] = x[n] * y[n] = y[n] * x[n]$$

Resultierende Länge für $z[n]$ (N_z) und Definitionsbereich:

$$N_z = N_x + N_y - 1 \quad z[n] \neq 0 \text{ gilt für } \{0 \leq n \leq N_x + N_y - 2\}$$

Beispiel einer Faltung mittels Polynommultiplikation zweier Signale $x[i] = \{1, 1, 1, -1\}$ und $y[i] = \{1, 1, 0.5, 0.5\}$:

$$\begin{array}{r} (1 + p + p^2 - p^3) \cdot (1 + p + 0.5p^2 + 0.5p^3) = \\ 1 + p + p^2 - p^3 \\ + p + p^2 + p^3 - p^4 \\ + 0.5p^2 + 0.5p^3 + 0.5p^4 - 0.5p^5 \\ + 0.5p^3 + 0.5p^4 + 0.5p^5 - 0.5p^6 \\ \hline 1 + 2p + 2.5p^2 + p^3 + 0p^4 + 0p^5 - 0.5p^6 \end{array}$$

Bei der zyklischen Faltung (**circular convolution**) müssen die beiden Signale die selbe Länge N haben, oder vorab mit zero-padding auf die selbe Länge $N = N_x + N_y - 1$ gebracht werden:

$$z[n] = x[n] \circledast_N y[n] = y[n] \circledast_N x[n]$$

Lösung durch Matrix:

$$\begin{bmatrix} y[N] & y[N-1] & \dots & y[0] \\ y[0] & y[N] & \ddots & y[1] \\ \ddots & \ddots & \ddots & \ddots \\ y[N-1] & y[N-2] & \dots & y[N] \end{bmatrix} \cdot \begin{bmatrix} x[0] \\ x[1] \\ \vdots \\ x[N] \end{bmatrix} = \begin{bmatrix} z[0] \\ z[1] \\ \vdots \\ z[N] \end{bmatrix}$$

Matlabbefehl für Faltung: `conv()`

Matlabbefehl für zyklische Faltung: `convmtx()`

Kapitel 2

Analog-Digital & Digital-Analog Wandlung

2.1 Schritte der A/D- und D/A-Wandlung

Sample: Kontinuierliche Signalwerte werden mit der Samplefrequenz f_S aufgezeichnet. Dies erzeugt eine Sequenz von diskreten Signalwerten.

Quantize: Die diskreten Signalwerte werden einer bestimmten Anzahl Quantisierungsleveln zugeordnet.

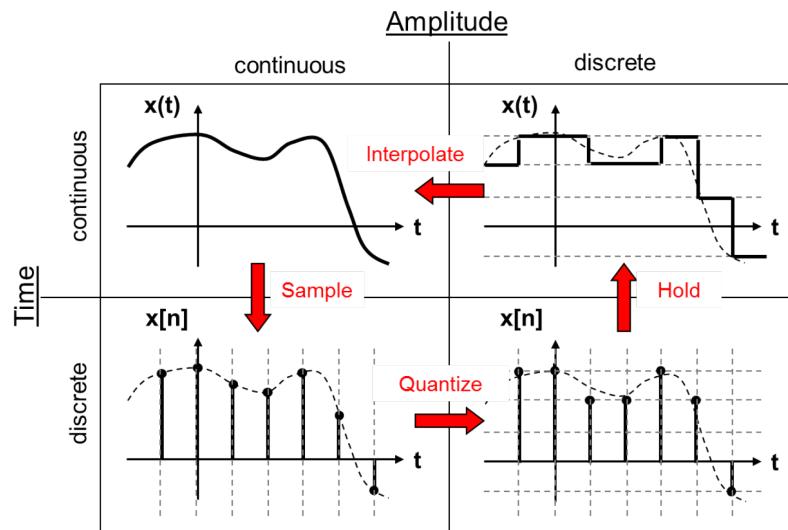
Code: Die quantisierten Abtastwerte können verwendet werden, um die erhaltene Pulsfolge zu modulieren (Pulse Code Modulation PCM). Meistens wird die Signalverarbeitung direkt mit den quantisierten Abtastwerten vorgenommen, so dass diese ohne Modulation gespeichert werden. Dazu wird die Repräsentierung der Quantisierungsleveln benötigt.

Die Digital-Analog Wandlung enthält folgende Schritte:

Decode: Die digitalen Werte werden in einer für die Digital-Analog Wandlung repräsentativer Form benötigt.

Hold: Das diskrete Signal muss über die Sampleperiode T_S konstant gehalten werden, ein Treppen-ähnlicher Output entsteht.

Interpolate: Das kontinuierliche Treppensignal wird durch Mittelwerte von Tiefpass-Filttern geglättet.



2.2 Digitale Signalprozessoren (DSP)

Ein DSP prozessiert das digitale Signal nach einem vorgegebenen Algorithmus in SW-Form. Übliche Anwendungen:

- **Signalgenerierung:** Harmonische & Random-Signale, wichtig im Bereich Mess- und Kommunikationstechnik.
- **Signalanalyse:** Analyse im Zeit- oder Frequenzbereich
- **Signalkomposition:** Zusammenführen, in der Zeitdomäne üblicherweise mittels Korrelationstechnik. In der Kommunikationstechnik wird im Frequenzbereich korreliert, um eine Modulation des Signals zu erreichen.

Vorteile:

- **Programmierbarkeit:** Selbe HW kann für Anwendung verwendet und einfach neu programmiert werden.
- **Parametrisierbarkeit:** Parameter können ohne Hardware-Anpassungen geändert werden, teils sogar automatisch.
- **Wiederholbarkeit:** Resultate können exakt reproduziert werden, keine Alterungserscheinungen o.ä.

Nachteile:

- Zusätzlicher Aufwand für AD und DA-Wandlung.
- (Noch) keine Prozessierung von HF-Signalen möglich.
- Potentielle EMV-Probleme mit analoger HW.

2.3 Abtasten und Aliasing

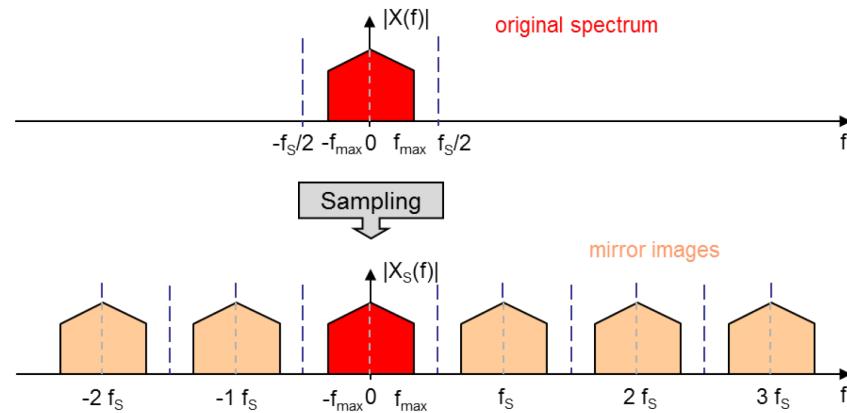
2.3.1 Abtasten von Tiefpass-Signalen

Mathematisch wird die Abtastung des Signals $x(t)$ durch eine Multiplikation mit Dirac-Impulsen der Periode T_S dargestellt. Es entsteht ein Zug aus gewichteten Dirac-Impulsen.

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

Das Frequenzspektrum des abgetasteten Signals ist:

$$X_S(f) = \frac{1}{T_S} \sum_{k=-\infty}^{\infty} X(f - kf_S)$$



Um das Signal rekonstruieren zu können, müssen die Spiegelfrequenzen von $X(f)$ mit einem Tiefpassfilter unterdrückt werden. Das geht nur, wenn die grösste Frequenz f_{max} des Signals kleiner als die halbe Abtastfrequenz $f_S/2$ ist, sich die "mirror images" also nicht überlappen. Aus dem **Nyquist-Kriterium** oder **Abtasttheorem** folgt:

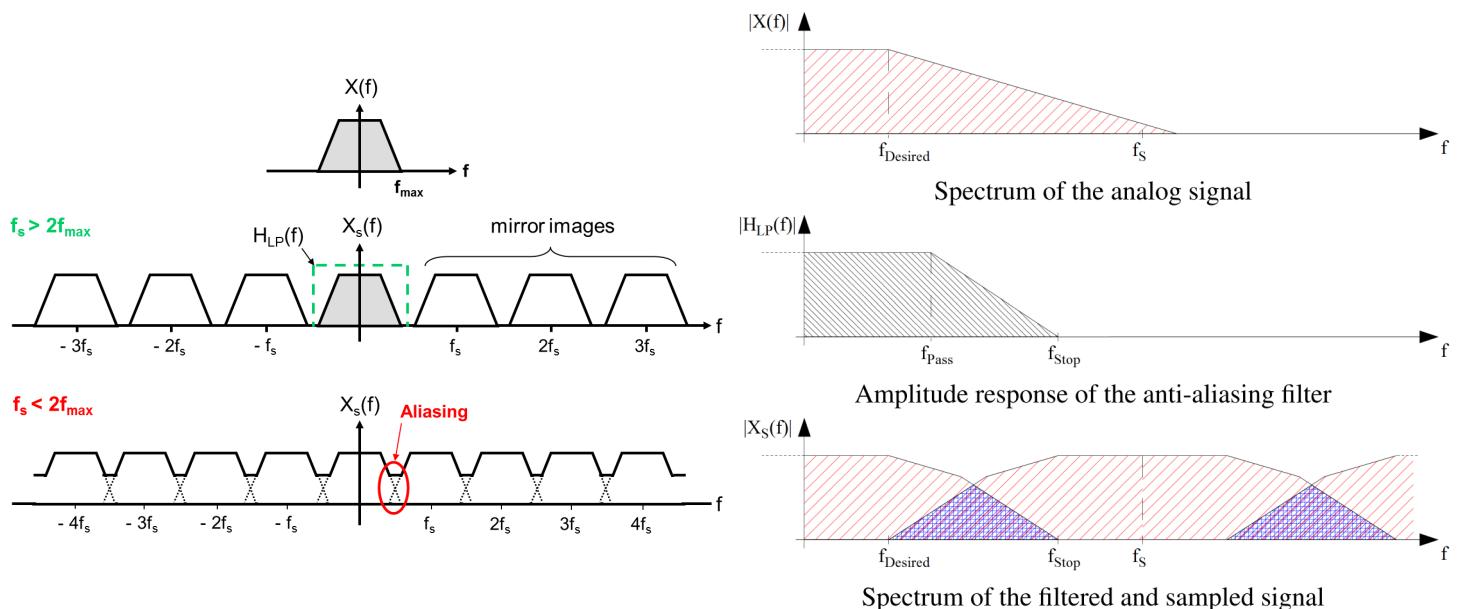
$$f_S > 2 \cdot f_{max}$$

2.3.2 Aliasing & Gegenmassnahme

Aliasing entsteht, wenn das Abtasttheorem verletzt wird. Die Frequenzanteile über $f_S/2$ werden in das Basisband gespiegelt und überlagern mit den gewünschten Frequenzen. Das analoge Signal kann nicht rekonstruiert werden.

Dem kann mit einer Tiefpassfilterung *vor* der Abtastung entgegengewirkt werden. Ausserhalb des gewünschten Frequenzbandes $f_{desired}$ darf es hingegen zu Aliasing kommen. Es gelten die Filter-Spezifikationen:

$$f_{pass} \geq f_{desired} \quad \text{sowie} \quad f_{stop} \leq f_S - f_{desired}$$



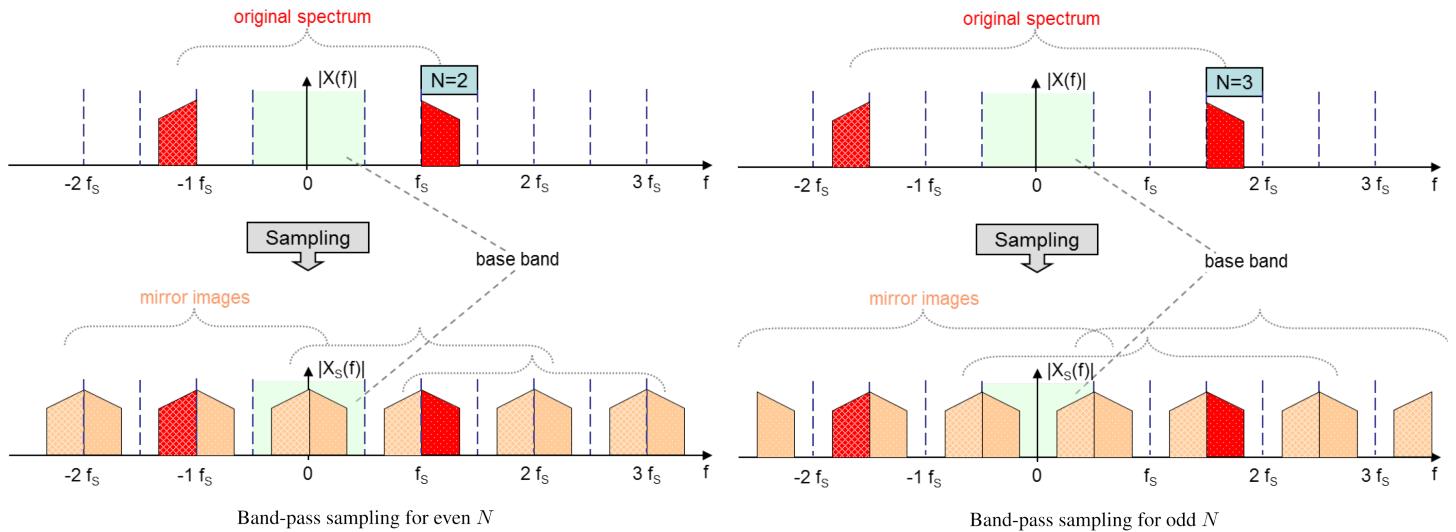
2.3.3 Abtasten von Bandpass-Signalen

Um Bandpass-Signale zu sampeln, wird das Abtasttheorem angepasst (für $N \geq 1$), f_S muss die Bedingung erfüllen:

$$\frac{2 \cdot f_{min}}{N} \geq f_S \geq \frac{2 \cdot f_{max}}{N + 1}$$

Beim Sampeln von Bandpass-Signalen wird der sonst unerwünschte Effekt des Aliasing ausgenutzt. Für ungerade N erscheint das originale Spektrum invertiert im Basisband. Die originale Struktur des Spektrums wird zurückgewonnen durch Invertierung jedes zweiten Samples im Zeitbereich. Dies entspricht einer Frequenzverschiebung von der halben Abtastfrequenz f_S .

$$\tilde{x}[n] = (-1)^n \cdot x[n]$$



2.4 Quantisierung von Signalen

2.4.1 Uniforme Quantisierung

Der Quantisierungsschritt (**quantization step**) Δ ist gegeben durch die Auflösung mit W Bits und der dynamischen Reichweite R des abgetasteten Signals $x[n]$:

$$\Delta = \frac{R}{2^W}$$

Bei der uniformen Quantisierung ist die Schrittgrösse entlang der x- und y-Achse uniform (gleichmässig) verteilt. Es existieren zwei unterschiedliche Arten der Quantisierung:

- Mid-tread quantizer: Entscheidungslevel bei $(\pm 0.5\Delta, \pm 1.5\Delta, \pm 2.5\Delta, \dots)$, 0-Level Output möglich
- Mid-rise quantizer: Entscheidungslevel bei $(\pm 0\Delta, \pm 1\Delta, \pm 2\Delta, \dots)$, einfache Implementation in HW & SW

Durch das Abbilden eines amplitudenkontinuierlichen Signals auf eine endliche Anzahl von Rekonstruktionsleveln können zweier Fehler entstehen:

Clipping: Werte von $x[n]$ ausserhalb des Bereichs R werden mit dem maximum bzw. minimum Rekonstruktionslevel dargestellt.

Quantization error ϵ : Dieser Fehler tritt immer auf und kann nicht verhindert werden. Die Grösse des Fehlers ist gegeben durch die Quantisierungsgrösse Δ .

Für den mid-tread quantizer (runden zum nächsten Wert):

$$-\Delta/2 < \epsilon \leq \Delta/2$$

Für den mid-rise quantizer (runden Richtung $-\infty$):

$$-\Delta < \epsilon \leq 0$$

2.4.2 Quantisierungsrauschen

Der Quantisierungsfehler zeigt sich als Rauschen überlagert zum quantisierten Signal:

$$\epsilon[n] = x_q[n] - x[n]$$

Die Leistung P_ϵ des Quantisierungsrauschen (quantization noise) wird ausgedrückt mit:

$$P_\epsilon = \sigma_\epsilon^2 = \int_{-\infty}^{\infty} (\epsilon - \sigma_\epsilon)^2 \cdot p(\epsilon) d\epsilon = \frac{\Delta^2}{12}$$

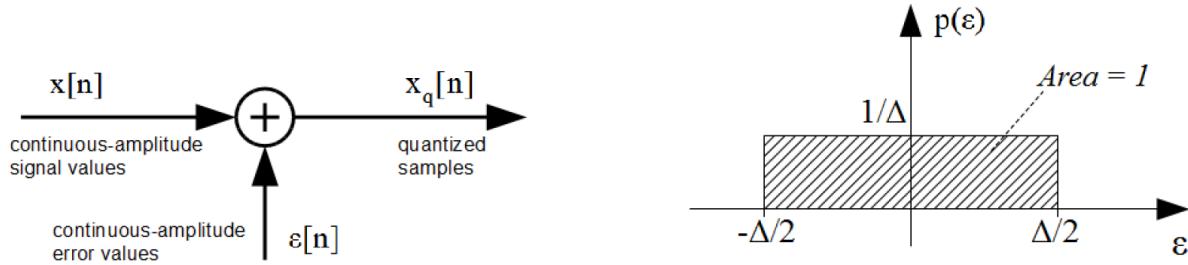
wobei $p(\epsilon)$ die Wahrscheinlichkeitsdichte ist. Angenommen die Werte von $\epsilon[n]$ sind statistisch unkorreliert und uniform verteilt über den Intervall $(-\Delta/2, \Delta/2]$, ist die Wahrscheinlichkeitsdichte für einen mid-tread quantizer:

$$p(\epsilon) = 1/\Delta$$

Der Erwartungswert des Quantisierungsfehlers $\mu(\epsilon)$ ergibt somit:

$$\mu(\epsilon) = \int_{-\infty}^{\infty} \epsilon \cdot p(\epsilon) d\epsilon = \frac{1}{2\Delta} \epsilon^2 \Big|_{-\Delta/2}^{\Delta/2} = 0$$

Das Modell zur Berechnung des Quantisierungsfehlers sowie die Fehlerverteilung bei einem mid-tread quantizer.



Für das Signal-Rausch-Verhältnis (SNR) gilt:

$$SNR = \frac{P_x}{P_\epsilon} = 2^{2W} \cdot \frac{12 \cdot P_x}{R^2}$$

Durch Ausdrücken des SNR in dB ergibt sich:

$$\begin{aligned} SNR_{dB} &= 10 \cdot \left(\log_{10} 2^{2W} + \log_{10} \frac{12P_x}{R^2} \right) \\ &\approx 6W + 10 \cdot \log_{10} \frac{12P_x}{R^2} \end{aligned}$$

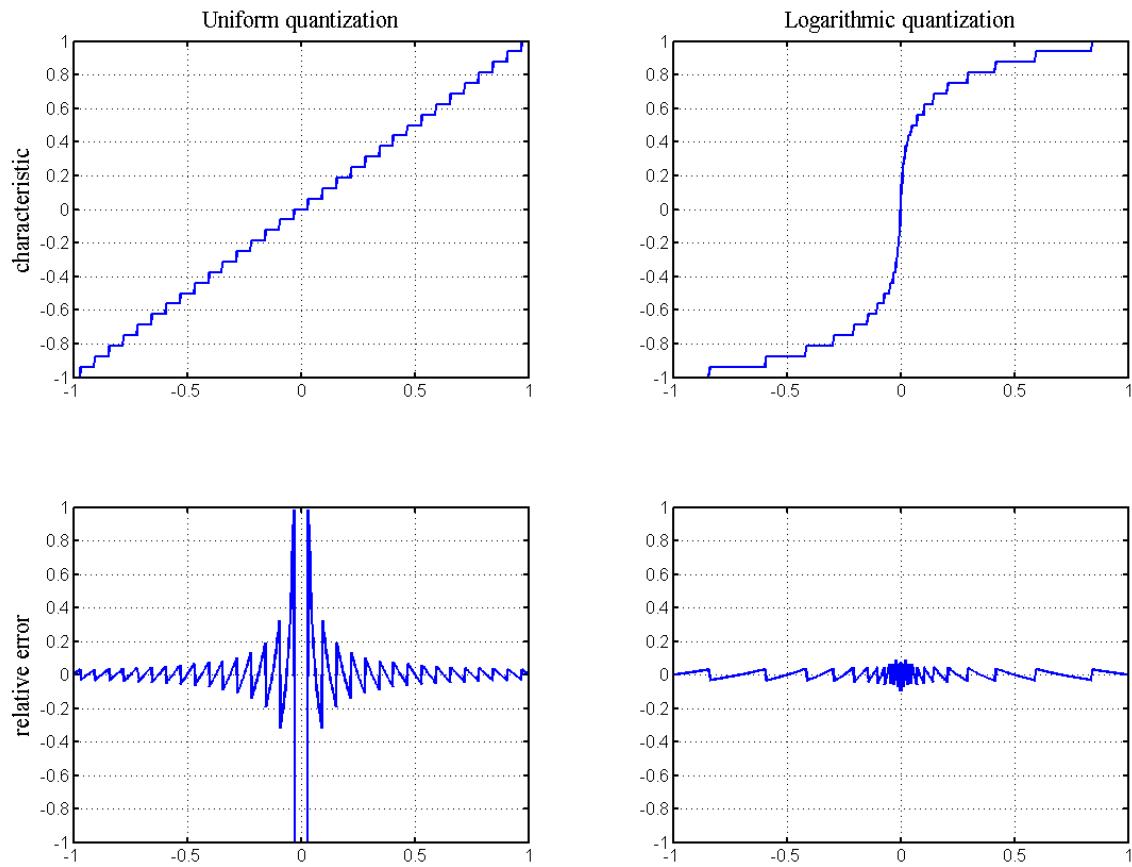
Bei einem harmonischen Inputsignal $x[n]$ ergibt sich folgendes SNR für eine uniforme Quantisierung:

$$SNR_{db} = 6W + 1.76 \approx 6W$$

Wichtige Faustformel: Mit jedem zusätzlichen Bit und uniformer Quantisierung erhöht sich das SNR um 6 dB.

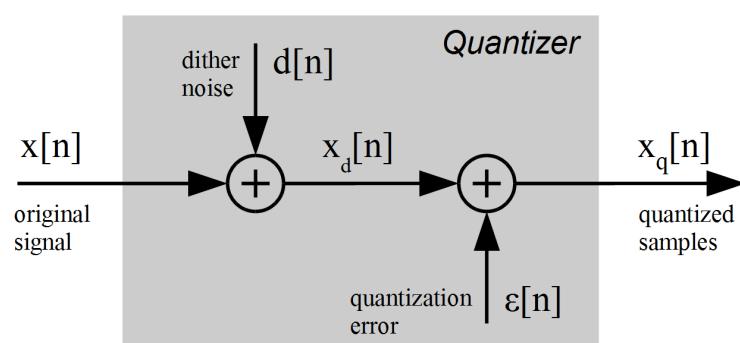
2.4.3 Logarithmische Quantisierung

Eine Möglichkeit das mit der Quantisierung verbundene SNR zu verbessern, besteht darin, die Quantisierung auf die statischen Eigenschaften des Signales anzupassen. Dieser Ansatz ist besonders vielversprechend für Signalklassen, die erheblich von einer gleichmäßigen Verteilung abweichen. Ein bekanntes Beispiel für eine solche Signalklasse sind menschliche Sprachsignale. Bei dieser Art von Signalen sind kleine Amplitudenwerte wahrscheinlicher als große Amplitudenwerte.



2.4.4 Dithering

Beim Dithering wird einem Signal vor der Quantisierung absichtlich Rauschen hinzugefügt. Dies wird oft in der Audio oder Bildverarbeitung eingesetzt um periodisches Rauschen zu vermindern. Das menschliche Gehör nimmt harmonisches Rauschen als störender wahr als zufälliges Rauschen. Das SNR wird durch Dithering verringert, jedoch nimmt der Mensch das Signal trotzdem als angenehmer wahr.



Kapitel 3

Digitale Signale im Frequenzbereich

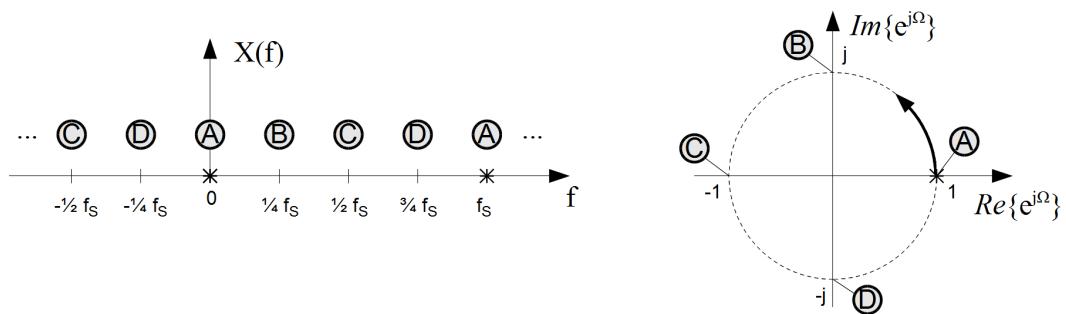
3.1 Von Fouriertransformation zur DFT

3.1.1 Übergang zu diskreter Zeit (DTFT)

Die zeitdiskrete Fourier Transformation (DTFT) berechnet sich mit der normalisierten Winkelfrequenz Ω :

$$\Omega = 2\pi f T_S = 2\pi \frac{f}{f_S} \left[\frac{\text{rad}}{\text{sample}} \right] \quad X(\Omega) = \sum_{n=-\infty}^{\infty} x[n] \cdot e^{-j\Omega n}$$

Die DTFT produziert ein 2π -periodisches, kontinuierliches Spektrum. Das Mapping sieht wie folgt aus:



3.1.2 Übergang zu endlichem Messintervall (DFT)

Je mehr Abtastpunkte bzw. je länger das Messintervall T , desto höher die Auflösung. Da in der Praxis nur über N Abtastpunkte eine Fourier Analyse gemacht werden kann, ist die kleinste Frequenzänderung Δf , welche festgestellt werden kann:

$$\Delta f = \frac{1}{T} = \frac{1}{N \cdot T_S} = \frac{f_S}{N}$$

Die Diskrete Fourier Transformation (DFT) kann geschrieben werden als:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot e^{-j2\pi n \frac{k}{N}} \quad k = 0, 1, 2, \dots, N-1$$

Dabei beschreibt n den diskreten Zeitindex, k den diskreten Frequenzindex.

Die DFT produziert ein diskretes und periodisches Linienspektrum mit Auflösung f_S/N . Die DFT berechnet die Spektralwerte an den folgenden Frequenzpunkten:

$$0, \frac{f_S}{N}, 2\frac{f_S}{N}, \dots, (N-1)\frac{f_S}{N} \quad \text{oder} \quad f_k = k \cdot \frac{f_S}{N} \text{ mit } 0 \leq k \leq N-1$$

Die DFT ist demnach beschränkt zu einer maximalen Frequenz von:

$$f_{max} = (N-1) \frac{f_S}{N}$$

Die inverse diskrete Fourier Transformation (IDFT) ist gegeben durch:

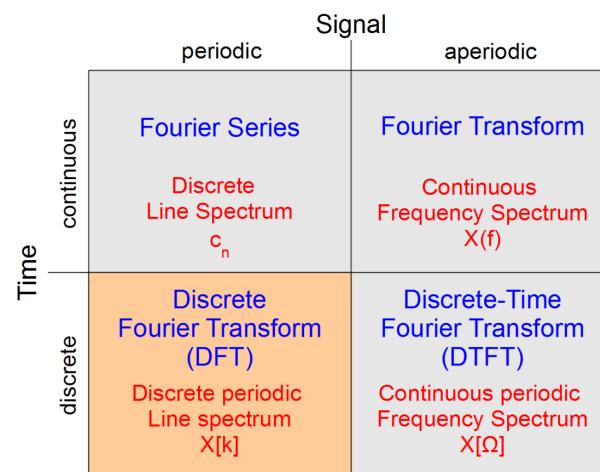
$$x[n] = \frac{1}{N} \sum_{k=0}^{N-1} X[k] \cdot e^{j2\pi n \frac{k}{N}} \quad n = 0, 1, 2, \dots, N-1$$

Achtung: Bei der IDFT wird über die Anzahl Samples normiert, bei der DFT jedoch nicht!

3.1.3 Vergleich der Fourier-Methoden

Die folgenden Punkte können festgehalten werden:

- Periodizität im Zeitbereich resultiert diskretes Linienspektrum im Frequenzbereich.
- Sampling im Zeitbereich resultiert in einem periodischen Spektrum.



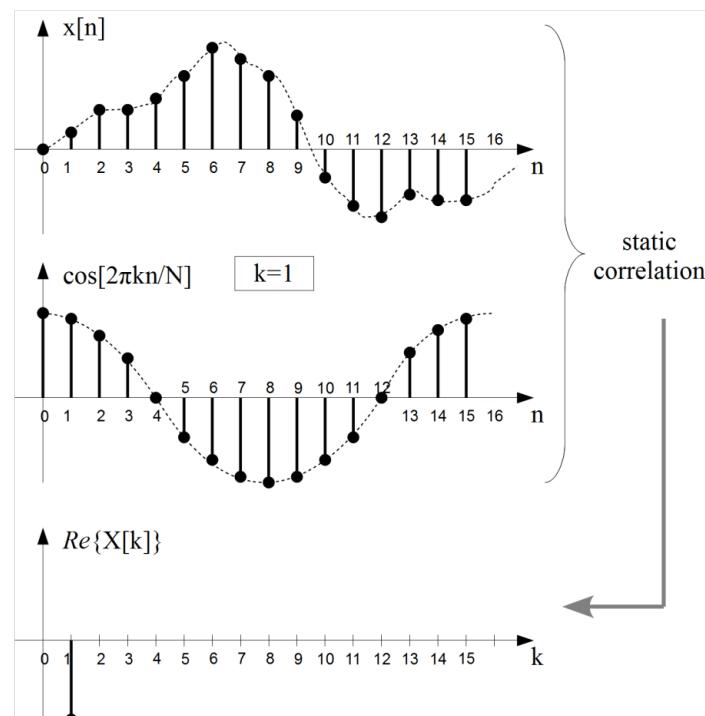
3.1.4 Intuitiver Ansatz zum Verständnis der DFT

Wendet man auf die Definition der DFT Analyse die eulersche Identität an, ergibt sich die Definition in Summen-Form, aufgeteilt in einen Real- und einen Imaginäranteil:

$$X[k] = \underbrace{\sum_{n=0}^{N-1} x[n] \cos(2\pi k \frac{n}{N})}_{\Re\{X[k]\}} + j \cdot \underbrace{\sum_{n=0}^{N-1} x[n](-1)^n \sin(2\pi k \frac{n}{N})}_{\Im\{X[k]\}}$$

Im Endeffekt handelt es sich beim Realanteil des Resultates der DFT $X[k]$ um eine statische Korrelation des Signals $x[n]$ mit $\cos(2\pi kn/N)$. Beim Imaginäranteil von $X[k]$ handelt es sich um eine statische Korrelation des Signals $x[n]$ mit $-\sin(2\pi kn/N)$. Der Index k gibt dabei an, wie oft die Schwingung in einer Periode platz hat.

Die folgende Graphik stellt den Zusammenhang graphisch dar für $k = 1$ & $N = 16$.



3.2 Eigenschaften der DFT

Periodizität: Das Spektrum der DFT ist aufgrund der diskreten Samples f_S -periodisch:

$$X[k] = X[k + N]$$

Das Spektrum wiederholt sich alle f_S Hertz. Für eine Frequenz f_1 entsteht Aliasing bei:

$$f_k = k \cdot f_S \pm f_1 \quad k = \{1, 2, 3, \dots\}$$

Dementsprechend ist IDFT periodisch mit $T = N \cdot T_S$:

$$x[n] = x[n + N]$$

Symmetrie: Die DFT eines realwertigen Signals ist symmetrisch um den Punkt $k = N/2$:

$$X[N/2 + m] = X^*[N/2 - m]$$

Zeit/Frequenz Verschiebung: Eine Verschiebung der Sequenz im Zeitbereich um n_0 Samples hat einen linearen Phasenoffset bei allen Spektralwerten zur Folge. Die zu addierende Phase ist $\varphi = 2\pi n_0 \frac{k}{N}$ und muss demnach für jedes k berechnet werden. Der Betrag bleibt unverändert.

$$x[n + n_0] \quad \circ \text{---} \bullet \quad e^{j2\pi n_0 \frac{k}{N}} \cdot X[k]$$

Die Multiplikation mit einem komplexen Exponent hat eine konstante Frequenzverschiebung zur Folge:

$$e^{j2\pi k_0 \frac{n}{N}} \cdot x[n] \quad \circ \text{---} \bullet \quad X[k - k_0]$$

Modulation: Konsequenz der Frequenzverschiebung ist die Modulation:

$$\cos\left(2\pi k_0 \frac{n}{N}\right) \cdot x[n] \quad \circ \text{---} \bullet \quad \frac{1}{2} (X[k + k_0] + X[k - k_0])$$

Parseval Theorem:

Zwischen Signalwerten $x[n]$ und Fourier Koeffizienten $X[k]$ besteht folgende Beziehung. Der linke Teil repräsentiert dabei die durchschnittliche Signalleistung des Signals $x[n]$. Das Parseval Theorem kann eingesetzt werden, um das SNR mittels den Fourier-Koeffizienten $X[k]$ zu bestimmen. Zuerst müssen dazu die Nutzsignal-Frequenzen von den Rausch-Frequenzen unterschieden werden.

$$\frac{1}{N} \sum_{n=0}^{N-1} x[n]^2 = \sum_{k=0}^{N-1} \left| \frac{X[k]}{N} \right|^2$$

Zusammenhang von Faltung und Multiplikation:

Punktweise Multiplikation zweier DFT Spektren $X[k]$ und $Y[k]$ im Frequenzbereich entspricht der zyklischen Faltung von $x[n]$ und $y[n]$ im Zeitbereich:

$$x[n] \circledast_N y[n] \quad \circ \text{---} \bullet \quad X[k] \cdot Y[k]$$

3.2.1 Gültigkeitsbereich der DFT

Wird die DFT mit endlichem N auf nicht-periodische Signale angewendet, muss $x[n]$ ausserhalb von $0 \leq n < N$ Null sein. Dann gilt, dass die DFT die DTFT an diskreten Punkten der normalisierten Kreisfrequenz Ω sampelt:

$$X[k] = X[\Omega] \Big|_{\Omega=2\pi \frac{k}{N}}$$

Fall 1: $x[n]$ periodisch

Der Messintervall $N \cdot T_S$ ist ein ganzzahliges Vielfaches der Periodendauer von $x[n]$.

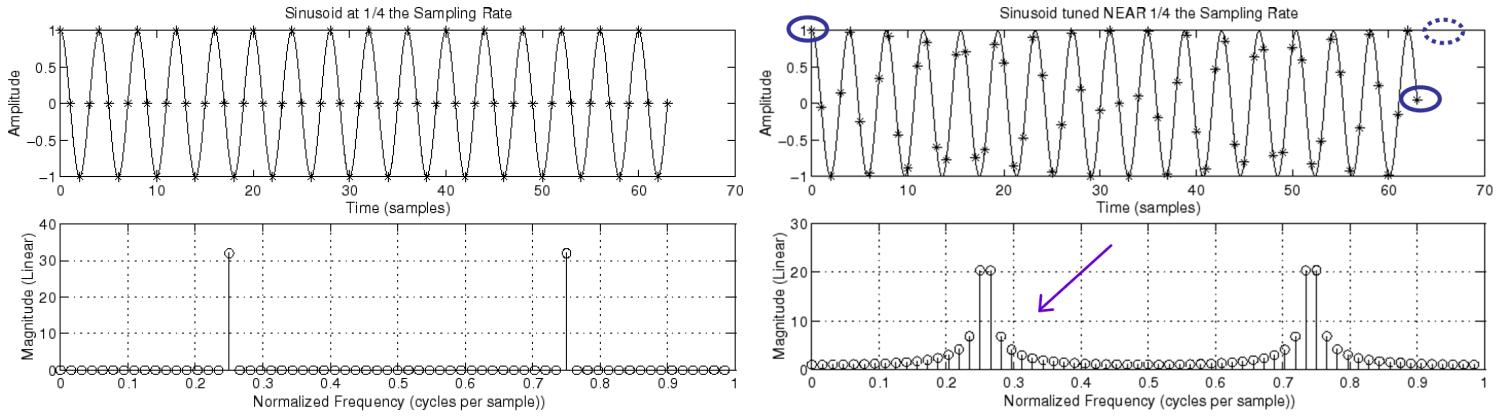
Fall 2: $x[n]$ aperiodisch

Alle Werte von $x[n]$ ausserhalb des Bereichs $0 \leq n < N$ sind Null. Ansonsten ergibt sich nur eine Annäherung. In der Praxis wird der Approximations-Fehler durch Fenster-Funktionen reduziert.

3.3 Anwendungs-Aspekte

3.3.1 Fensterlänge & Leakage

Wie im vorherigen Unterkapitel (Fall 1) beschrieben, muss das Messintervall ein ganzzahliges Vielfaches der Periodendauer von $x[n]$ sein. Da dies in der Praxis nicht möglich ist, entsteht sogenanntes Leakage. Der Effekt zeigt sich, indem im Bereich um die eigentlichen Signalfrequenzen “verschmiert” wird. Ohne Leakage (links) und mit Leakage (rechts):



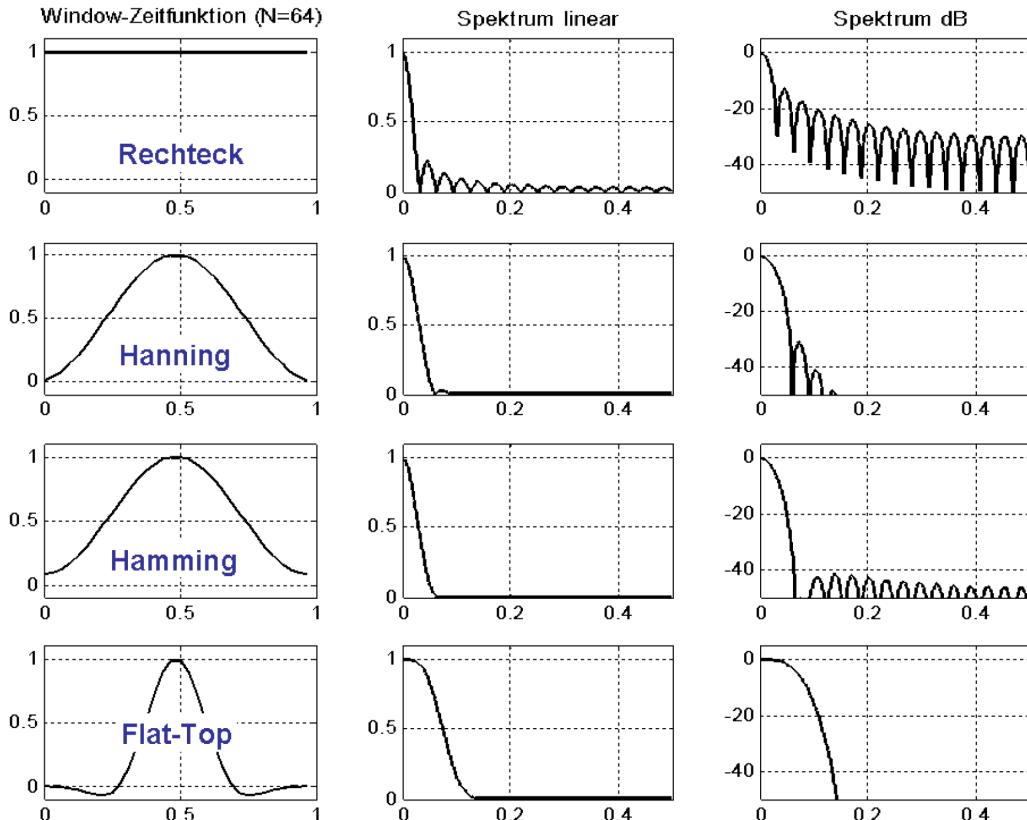
3.3.2 Zero-Padding

Um eine bessere Interpolation zwischen den N Frequenzpunkten in der DFT zu bekommen, kann das Signal im Zeitbereich mit Nullen aufgefüllt werden. Das Spektrum $X(\Omega)$ sowie die Auflösung des Spektrums wird nicht verändert, allerdings stehen zusätzliche Abtastpunkte auf der Achse der normalisierten Winkelfrequenz Ω zur Verfügung.

3.3.3 Fensterfunktion

Die DFT wendet per Definition ein Rechteckfenster an, um die N Samples auszuschneiden. Es können andere Fensterfunktionen auf das Signal $x[n]$ angewendet werden, allerdings gelten folgende Punkte:

- Je schmäler die Hauptkeule im Spektrum des Fensters, desto höher ist die spektrale Auflösung für $X[k]$.
- Je stärker die Dämpfung der Nebenkeulen im Spektrum des Fensters, desto besser die Unterdrückung der Leakage in $X[k]$.



3.4 Short-Time DFT

Wenn die Entwicklung bzw. der Verlauf des Frequenzspektrums über die Zeit interessiert, werden kontinuierliche Berechnungen des Spektrums von kurzen Signalsektoren durchgeführt. Da diese kurzen Sektoren meistens nicht ein ganzzahliges Vielfaches der Periode sind, werden Fensterfunktionen angewendet.

Bei der Länge N des Fensters ergibt sich der Kompromiss von:

- Grosser spektraler Auflösung (N gross).
- Grosser zeitlicher Auflösung (N klein).

Um beides zu erreichen, können die DFT-Fenster überlappt werden um maximal $N - 1$.

3.5 Fast Fourier Transformation (FFT)

Die FFT ist ein effizienter Algorithmus zur DFT-Berechnung nach dem Teile-und-Hersche Prinzip. Es wird rekursiv in r Subsequenzen unterteilt. r wird als "radix" der FFT bezeichnet. Meistens werden radix-2 Algorithmen eingesetzt.

Die Frequenzauflösung ist gegeben durch, wobei $N = 2^L$ mit einem ganzzahligen Wert für L ist: $\Delta f = \frac{f_s}{N}$

Geschwindigkeitsvergleich FFT-DFT:

Vereinfachungen & Annahmen:

- Eine Addition von 2 realen Zahlen ist so rechenintensiv wie eine Multiplikation von 2 realen Zahlen.
- Eine komplexe Multiplikation benötigt vier reale Multiplikationen und 2 reale Additionen.
- Eine komplexe Addition benötigt zwei reale Additionen.

Der speedup factor der FFT ist demnach wie folgt:

$$\text{speedup factor}_{FFT} = \frac{8N - 2}{5 \cdot \log_2 N} \approx 1.5 \cdot \frac{N}{\log_2 N}$$

Reduktion des Rechenaufwandes:

$$\begin{aligned} \text{Aufwand mit "normaler" DFT: } & [N^2]_{\text{cpl.Mul.}} + [N^2 - N]_{\text{cpl.Add.}} \\ \text{Aufwand mit FFT: } & \left[\frac{N}{2} \cdot \log_2 N \right]_{\text{cpl.Mul.}} + [N \cdot \log_2 N]_{\text{cpl.Add.}} \end{aligned}$$

3.5.1 Twiddle-Faktoren

Diese Drehfaktoren werden verwendet, um die Eigenschaften der komplexen Harmonischen zu beschreiben, mit denen die Korrelation zur Berechnung der DFT im Endeffekt durchgeführt wird. Definition:

$$W_N = e^{-j \frac{2\pi}{N}}$$

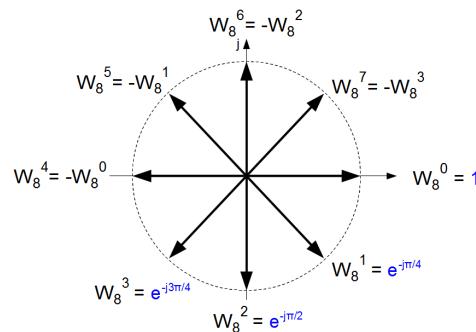
Die folgenden Eigenschaften werden bei der Anwendung der Twiddle-Faktoren ausgenutzt:

Periodizität: W_N kann nur für N verschiedene Zahlen ausgewertet werden und ist somit N -periodisch:

$$W_N^{k+N} = W_N^k$$

Symmetrie: Abgesehen vom Vorzeichen nimmt W_N nur $N/2$ verschiedene Werte an:

$$W_N^{k+N/2} = -W_N^k$$



3.5.2 Radix-2 FFT

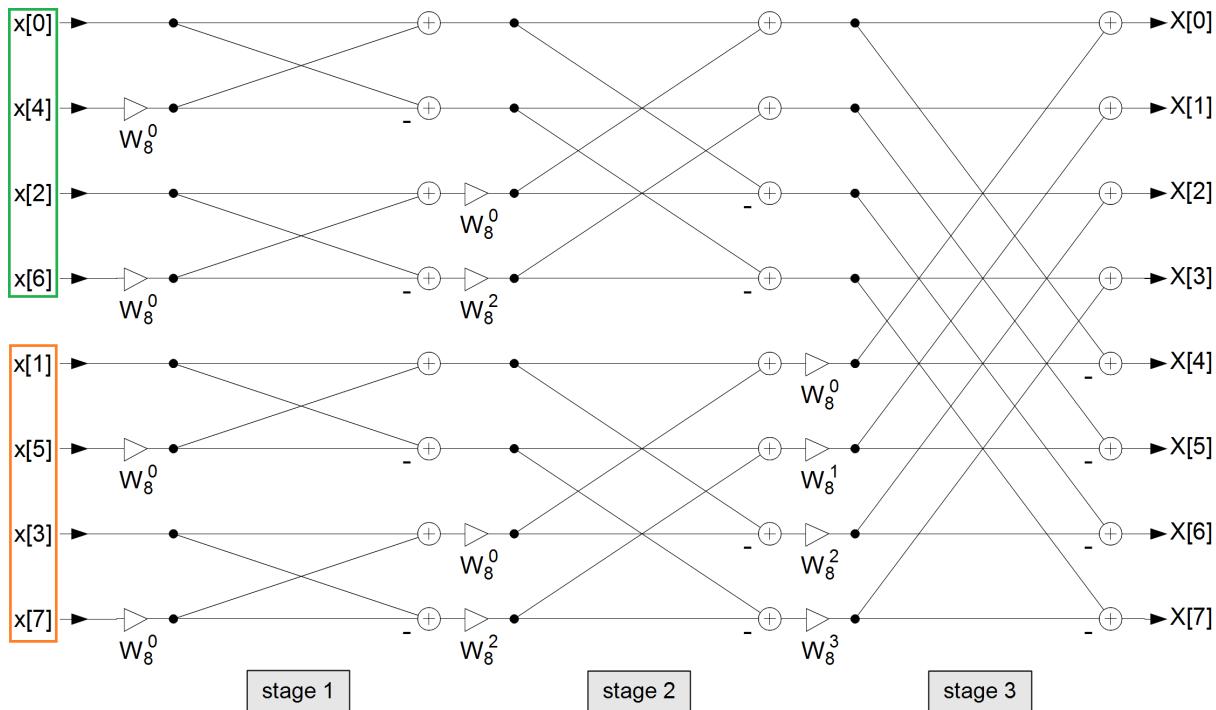
Die DFT wird mit dem Twiddle Faktor umgeschrieben:

$$X[k] = \sum_{n=0}^{N-1} x[n] \cdot W_N^{nk} = \sum_{n \text{ even}} x[n] \cdot W_N^{nk} + \sum_{n \text{ odd}} x[n] \cdot W_N^{nk}$$

Es können zwei neue Sequenzen $x_1[n]$ und $x_2[n]$ mit geraden und ungeraden n von der Länge $N/2$ erzeugt werden.

$$X[k] = \underbrace{\sum_{n=0}^{N/2-1} x_1[n] \cdot W_{N/2}^{nk}}_{x_1[\tilde{k}]} + W_N^k \cdot \underbrace{\sum_{n=0}^{N/2-1} x_2[n] \cdot W_{N/2}^{nk}}_{x_2[\tilde{k}]} \quad \text{wobei: } \tilde{k} = k \pmod{N/2}$$

Im Endeffekt kann die N -punkt DFT durch $N/2$ 2-Punkt DTFs ausgedrückt werden. Beispiel einer 8-Punkt radix-2 FFT nach Aufteilung in zwei Sequenzen selber Länge (ungerade & gerade). Die innere Struktur wird *Butterfly* genannt. Es handelt sich bereits um eine vereinfachte Struktur der inneren *Butterflys*.



Benötigter Speicher der FFT bei in-place (speicheroptimiert): $2N$ (N komplexwertige Koeffizienten) + N ($N/2$ komplexwertige Twiddle Faktoren)

Index bit-reversed in Matlab: `bitrevorder()`

Matlab-Befehle für Darstellung des Spektrum eines Signals $x1$:

```
n = length(x1); % # of samples in x1
plot([-n/2:1:n/2-1], 20*log10(abs(fftshift(fft(x1)))));
axis([-1e4 1e4 -200 0]);
ylabel('X_1(f)_dB');
title('Original Spectra');
```

linear order		bit-reversed order	
decimal	binary	binary	decimal
0	000	000	0
1	001	100	4
2	010	010	2
3	011	110	6
4	100	001	1
5	101	101	5
6	110	011	3
7	111	111	7

3.6 Goertzel-Algorithmus

Der Goertzel Algorithmus beschreibt einen anderen Ansatz zur DFT-Berechnung mittels linearer Filtertechnik. Er dient dazu, eine bestimmte Frequenzkomponenten $X[k]$ aus den N möglichen Komponenten zu berechnen. Definition:

$$X[k] = y_k[n]|_{n=N} = \sum_{i=0}^{N-1} x[i] \cdot W_N^{-k(N-i)}$$

Übertragungsfunktion des Filters:

$$H_k(z) = \frac{1}{1 - W_N^{-k} z^{-1}}$$

Differenzengleichung:

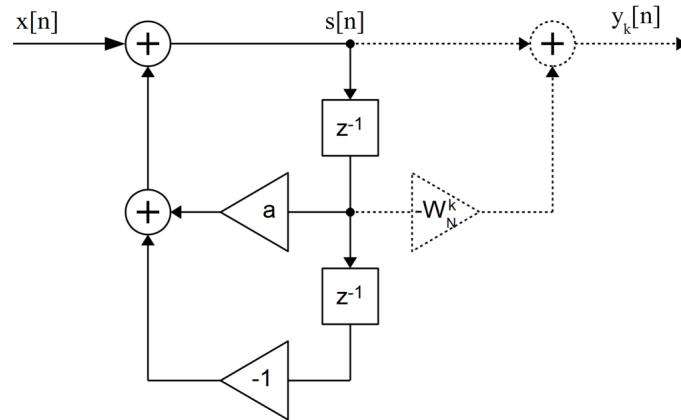
$$s[n] = x[n] + a \cdot s[n-1] - s[n-2]$$

$$y_k[n] = s[n] - W_N^k \cdot s[n-1]$$

mit

$$a = 2 \cos 2\pi \frac{k}{N}$$

$s[n]$ muss für alle Zeitpunkte $n = 0, 1, \dots, N$ berechnet werden, $y_k[n]$ nur für $n = N$ (unten gestrichelt).



Der Goertzel-Algorithmus ergibt die DFT Spektralkomponente bei der Frequenz:

$$f_k = k \frac{f_s}{N}$$

Mit dem Parseval Theorem kann der Leistungsgehalt P_k eines realwertigen Signals $x[n]$ bei einer Frequenz von f_k ermittelt werden:

$$P_k = 2 \left| \frac{X[k]}{N} \right|^2 = \frac{2}{N^2} (\Re\{X[k]\}^2 + \Im\{X[k]\}^2)$$

Rechenaufwand: Für reelle Eingangssignale $x[k]$ benötigt der Algorithmus für die Spektrumskomponente (positive und negative) dabei $N + 1$ reale Multiplikationen und eine komplexe Multiplikation.

Kapitel 4

Digitale LTI Systeme

4.1 Definition

Linearität:

$$x[n] = k_1 \cdot x_1[n] + k_2 \cdot x_2[n] \implies y[n] = k_1 \cdot y_1[n] + k_2 \cdot y_2[n]$$

Zeitinvarianz:

$$x[n] \rightarrow y[n] \implies x[n-d] \rightarrow y[n-d]$$

Erlaubte Operationen:

- Multiplikation eines Signals mit einer Konstanten.
- Addition von zwei Signalen.
- Zeitverzögerung eines Signals um $k \cdot T_S$.

4.2 Beschreibungsarten von LTI-Systemen

Je nach Anwendung eignen sich unterschiedliche Beschreibungsarten, zwischen denen transformiert werden kann.

Zeitbereich	Impuls- oder Sprungantwort ($h[n], u[n]$)	Systemidentifikation, Messung
	Differenzengleichung	Systemimplementation (Algorithmus)
Frequenzbereich	Signalflussdiagramm	Systemimplementation (Architektur)
	Übertragungsfunktion	Gekoppelte Analyse & Implementation
	Pol- Nullstellenplan	Intuitives Analyse- & Designvorgehen
	Frequenzantwort	Systemidentifikation, Analyse & Design

Nützliche Matlab-Befehle:

Definition: Impulsantwort mit $h[n] = \{1, -1\}$ $h = [1 -1];$

Plot: Impulsantwort mit Amplitude 1, Länge 10 $impz(h, 1, 10);$

Plot: Pol-Nullstellenplan $zplane(h, 1);$

Plot: Frequenzgang (Gain & Phase) $freqz(h, 1, 100); % freqz(b,a,N), b = Zähler, a = Nenner$

4.3 Systembeschreibung im Zeitbereich

4.3.1 Impulsantwort / Faltung

Das Ausgangssignal $y[n]$ kann durch Faltung des Eingangssignals $x[n]$ mit der Impulsantwort $h[n]$ ermittelt werden.

$$y[n] = \sum_{i=-\infty}^{\infty} x[i] \cdot h[n-i] = x[n] * h[n]$$

4.3.2 Differenzengleichung

Das $\max(N, M)$ wird als *Ordnung* des Systems bezeichnet. Falls $M \geq 1$ ist das System rekursiv, vergangene Ausgangswerte werden also in den aktuellen Ausgangswert miteinbezogen. a_k & b_k beeinflussen das Systemverhalten.

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

4.3.3 Signalflussdiagramm

Es existieren vier Grundlegende Varianten, welche jeweils Vor- und Nachteile für die praktische Implementation mit sich bringen. Im Kapitel "Digitales Filterdesign" sind die vier Diagramme abgebildet.

4.4 Systembeschreibung im Frequenzbereich

4.4.1 Übertragungsfunktion

Durch z-Transformation der Differenzengleichung ergibt sich:

$$y[n] = \sum_{k=0}^N b_k \cdot x[n-k] - \sum_{k=0}^M a_k \cdot y[n-k] \quad \circlearrowright \bullet \quad Y(z) = \sum_{k=0}^N b_k z^{-k} X(z) - \sum_{k=0}^M a_k z^{-k} Y(z)$$

Durch Umstellen und Normieren von a_0 auf 1 erhält man:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_N \cdot z^{-N}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_M \cdot z^{-M}}$$

4.4.2 Pol- Nullstellenplan

Durch Faktorisieren der obigen Übertragungsfunktion $H(z)$ ergibt sich die folgende Darstellung:

$$H(z) = K_0 \cdot \frac{(z - z_1)(z - z_2)\dots(z - z_N)}{(z - p_1)(z - p_2)\dots(z - p_M)} \cdot z^{M-N}$$

- Für $N > M$ gibt es $N - M$ zusätzliche Polstellen bei $z = 0$.
- Für $M > N$ gibt es $M - N$ zusätzliche Nullstellen bei $z = 0$.
- Für kausale Systeme gilt, dass das System stabil ist, wenn: $|p_i| < 1, i = 1, \dots, M$.

4.4.3 Frequenzgang

Beschreibt das Systemverhalten als Funktion der Frequenz des Eingangssignals (frequency sweep). Es gilt:

$$h[n] \quad \circlearrowright \bullet \quad H(\Omega)$$

In der Praxis wird die DTFT durch die DFT approximiert und als Polarkoordinaten dargestellt:

$$H(\Omega) = |H(\Omega)| \cdot e^{j\varphi(H(\Omega))}$$

Amplitudengang: Beschreibt die Verstärkung, die ein Signal der Frequenz Ω ausgesetzt ist:

$$|H(\Omega)|_{dB} = 20 \cdot \log_{10}(|H(\Omega)|)$$

Phasengang: Beschreibt die Phasenverschiebung, die ein Signal der Frequenz Ω ausgesetzt ist:

$$\varphi(Y(\Omega)) = \varphi(X(\Omega)) + \varphi(H(\Omega))$$

4.4.4 Beziehung zwischen Frequenzgang & Übertragungsfunktion

DTFT und z-Transformation sind mit folgender Beziehung verbunden:

$$z = r \cdot e^{j\Omega}$$

Der Frequenzgang lässt sich also durch Betrachtung von $H(z)$ entlang des Einheitskreises erhalten.

$$H(\Omega) = H(z)|_{z=e^{j\Omega}}$$

Durch Substitution von $z = e^{j\Omega}$ lässt sich die Verstärkung & Phase bei einer beliebigen Frequenz Ω berechnen.

Amplitudengang:

$$|H(z)| = |K| \cdot \frac{|(z - z_1)||z - z_2|\dots|(z - z_N)|}{|(z - p_1)||z - p_2|\dots|(z - p_M)|} \cdot |z|^{M-N}$$

Phasengang:

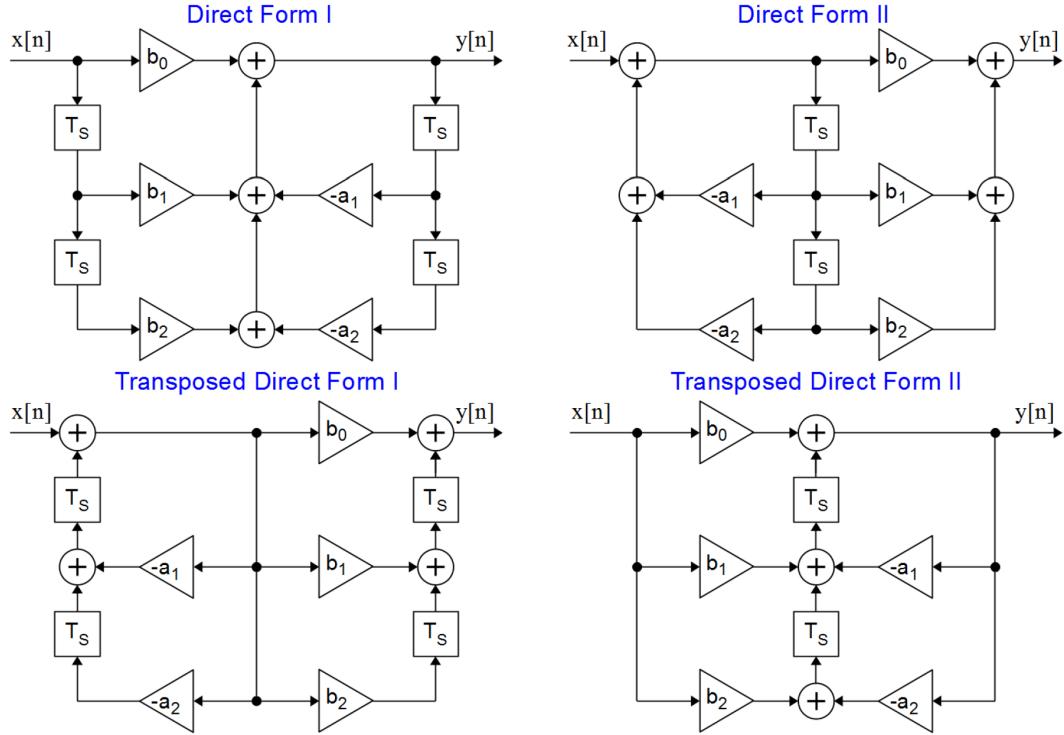
$$\varphi(H(z)) = \sum_{k=1}^N \varphi(z - z_k) - \sum_{k=1}^M \varphi(z - p_k) + \sum_{k=N+1}^M \varphi(z)$$

Trick 1 & 2 auf der Folgeseite beachten, wenn Verstärkungen bei bestimmtem Ω berechnet werden müssen!

Kapitel 5

Digitales Filterdesign

Ziel: Bestimmen der Parameter b_k & a_k , sowie der Ordnung N & M für gegebene Filterspezifikationen.



5.1 Übertragungsfunktion

Differenzengleichung:

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

Übertragungsfunktion (a_0 normiert auf 1):

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_N \cdot z^{-N}}{1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_M \cdot z^{-M}}$$

Amplituden, Phasengang:

$$H(\Omega) = H(z)|_{z=e^{j\Omega}}$$

Beim Amplitudengang wird der Betrag $|H(\Omega)|$ in Abhängigkeit von Ω geplottet:

$$|H(\Omega)| = |K| \cdot \frac{|(z - z_1)||z - z_2| \dots |z - z_N|}{|(z - p_1)||z - p_2| \dots |z - p_M|} \cdot |z|^{M-N}$$

Phasengang:

$$\varphi(H(z)) = \sum_{k=1}^N \varphi(z - z_k) - \sum_{k=1}^M \varphi(z - p_k) + \sum_{k=N+1}^M \varphi(z)$$

Trick 1:

$$1 - e^{-k \cdot jx} = e^{-\frac{k}{2} \cdot jx} \cdot (e^{\frac{k}{2} \cdot jx} - e^{-\frac{k}{2} \cdot jx})$$

Trick 2:

$$\sin(x) = \frac{e^{j \cdot x} - e^{-j \cdot x}}{2j} \quad \cos(x) = \frac{e^{j \cdot x} + e^{-j \cdot x}}{2}$$

Beispielaufgabe Phasenwinkel:

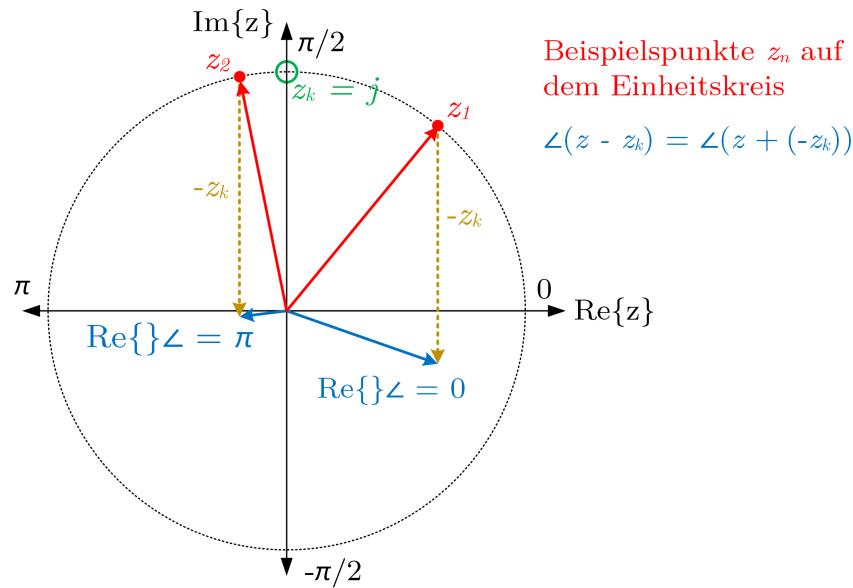
Analytisches Ermitteln des Phasenganges für die Übertragungsfunktion $H(z) = 1 - z^{-4}$:

$$\begin{aligned}\angle H(\Omega) &= \angle H(z)|_{z=e^{j\Omega}} \\ &= \angle(1 - e^{-4 \cdot j\Omega})\end{aligned}$$

Anwenden von Trick 1: $= \angle(e^{-2 \cdot j\Omega} \cdot (e^{2 \cdot j\Omega} - e^{-2 \cdot j\Omega}))$

$$\begin{aligned}\text{Anwenden von Trick 2: } &= \angle(\underbrace{e^{-2 \cdot j\Omega}}_{=-2\Omega} \cdot \underbrace{2 \cdot j}_{+\frac{\pi}{2}} \cdot \underbrace{\sin(2\Omega)}_{0 \text{ für: } 0 \leq \Omega \leq \frac{\pi}{2}}) \\ &\quad + \pi \text{ für: } \frac{\pi}{2} \leq \Omega \leq \pi\end{aligned}$$

Es gibt somit eine Fallunterscheidung für die zwei Bereiche von Ω . Dies ist zuständig für den Phasensprung bei $\pi/2$. Graphische Darstellung des Phasenwinkels für das letzte Element (sinus):

**Beispielaufgabe Verstärkung:**

Berechnen der Frequenz, bei der die Verstärkung $|H(z)|$ von $H(z) = 1 - z^{-1} = \sqrt{2}$ ist:

$$H(\Omega) = H(z)|_{z=e^{j\Omega}}$$

$$|H(\Omega)| = |1 - e^{-j\Omega}| = \frac{|e^{j\Omega} - 1|}{|e^{j\Omega}|} \implies |e^{j\Omega}| \text{ aus dem Nenner} = 1$$

Verinfacht sich somit zu: $|H(\Omega)| = |e^{j\Omega} - 1|$

Anwenden von Trick 1: $= |e^{j\frac{\Omega}{2}} \cdot (e^{j\frac{\Omega}{2}} - e^{-j\frac{\Omega}{2}})| \implies |e^{j\frac{\Omega}{2}}|$ vor der Klammer = 1

$$\text{Vereinfachen \& Anwenden von Trick 2: } = |2 \cdot j \cdot \sin\left(\frac{\Omega}{2}\right)| = 2 \cdot |\sin\left(\frac{\Omega}{2}\right)| \stackrel{!}{=} \sqrt{2}$$

$$\text{Auflösen der Gleichung nach } \Omega \text{ ergibt: } \Omega_0 = \frac{\pi}{2}, \Omega_1 = \frac{3}{2}\pi$$

5.2 FIR (finite impulse response) Filter

Bei einem FIR-Filter ist die Ordnung des Nenners immer $M = 0$, die Übertragungsfunktion für Ordnung N lautet:

$$H(z) = b_0 + b_1 z^{-1} + \dots + b_N z^{-N}$$

Die Impulsantwort ist $N + 1$ Zeitschritte lang und entspricht den Koeffizienten von $H(z)$:

$$h[n] = \{b_0, b_1, \dots, b_N, 0, 0, \dots\}$$

Stabilität: Da alle Pole bei $z = 0$ liegen, sind FIR-Filter per Definition stabil.

Lineare Phase: Mit einem FIR-Filter ist es einfach möglich, eine lineare Phasenübertragung zu realisieren.

Implementation: Die Realisierung von FIR-Filters in HW oder SW ist straightforward und unkritisch.

5.2.1 Symmetrische FIR-Filter

Ein FIR-Filter ist symmetrisch wenn:

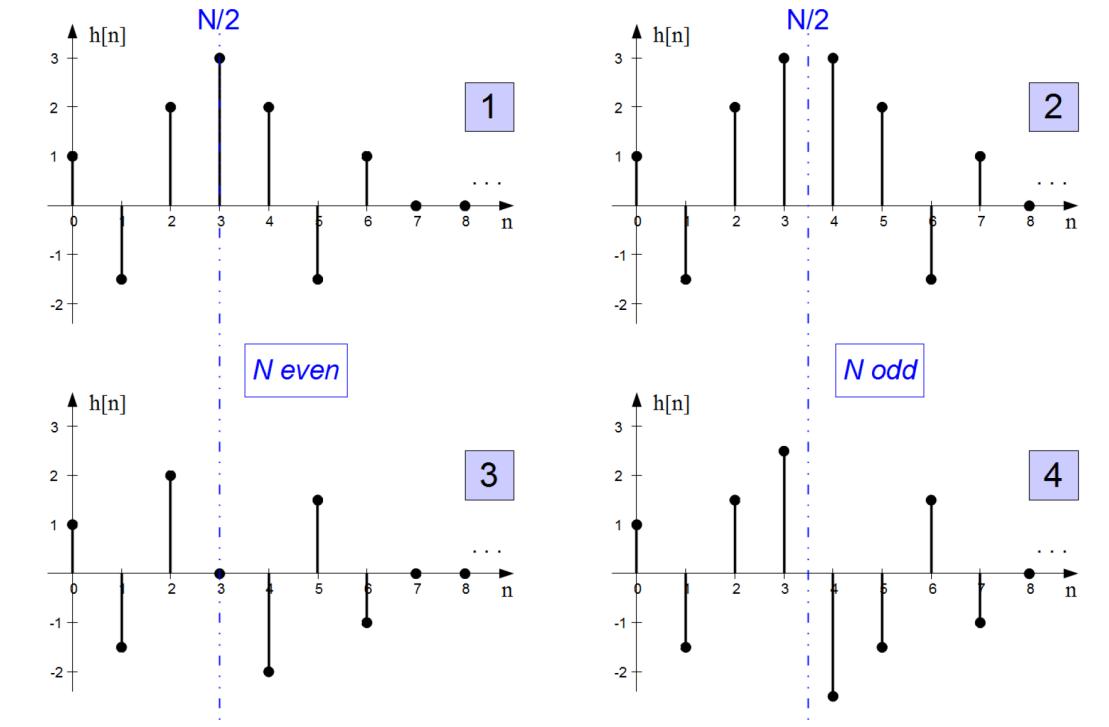
$$b_i = \pm b_{N-i} \quad i = 0, 1, \dots, N$$

- Gespiegelt-symmetrischer (oder einfach symmetrischer) FIR-Filter: Gleichung mit “+“ ist erfüllt.
- Anti-symmetrischer FIR-Filter: Gleichung mit “-“ ist erfüllt.

Alle symmetrischen Filter haben eine lineare Phasenantwort im Pass-Band und somit eine konstante Gruppenlaufzeit:

$$\tau_g = \frac{N}{2} \cdot T_S$$

Beispiel für Filter mit Ordnung 6 (links) & Ordnung 7 (rechts).



Type	Symmetry	Order N	$ H(f = 0) $	$ H(f = f_S/2) $	$H(\Omega)$
1	$h[n] = h[N - n]$ (symmetric)	even	any	any	$e^{-j\Omega \frac{N}{2}} \cdot H_{zp}(\Omega)$
2		odd	any	0	
3	$h[n] = -h[N - n]$	even	0	0	$e^{-j(\Omega \frac{N}{2} - \frac{\pi}{2})} \cdot H_{zp}(\Omega)$
4	(anti-symmetric)	odd	0	any	

Anmerkung Phasensprung & Gegenmassnahme:

Im Stop-Band symmetrischer FIR-Filter können 180° -Phasensprünge auftreten. Sie entstehen bei jedem Frequenzpunkt von komplex-konjugierten Nullstellen am Einheitskreis. Im Normalfall werden sie toleriert, da die Dämpfung im Stopp-Band entscheidend ist. Ansonsten können die Sprünge durch Duplikate der komplex-konjugierten Nullstellen-Paare eliminiert werden. Achtung: Dies erhöht jedoch die Gruppenlaufzeit durch die entstehende höhere Ordnung.

Anmerkung Gruppenlaufzeit:

Symmetrische FIR-Filter weisen eine lineare Gruppenlaufzeit auf mit $N/2$ Samples. Die Nullstellen dieser Filter treten in reziproken Paaren auf (z.B. bei $-1/2$ und -2).

5.2.2 Window Design Methode

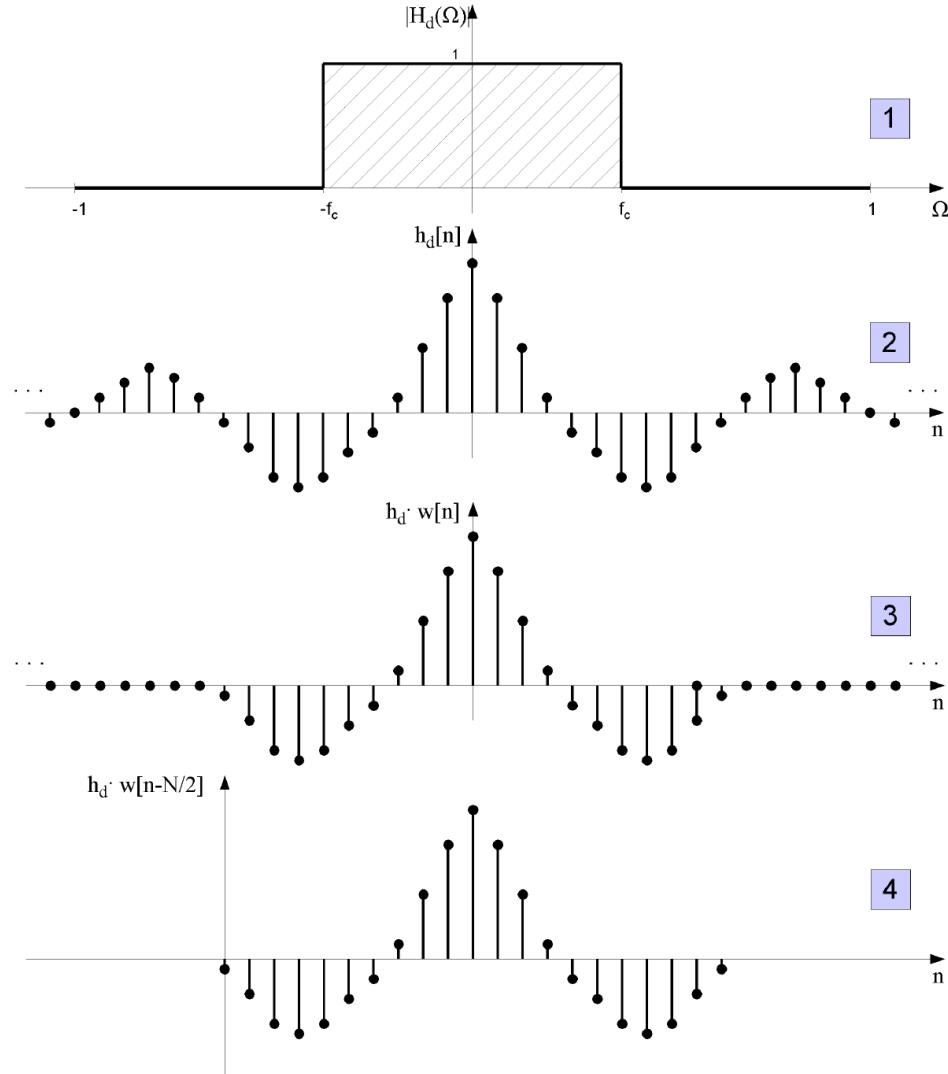
Der Matlab-Befehl `fir1()` verwendet die Window Design Methode. Es wird mit der Impulsantwort eines idealen Tiefpass-Filters mit Cutoff-Frequency f_c gestartet. Die ideale Impulsantwort ist nicht endlich. Sie wird auf eine endliche Länge von $N + 1$ begrenzt und um $N/2$ Samples geshiftet.

$$h_{d_{TP}}[n] = \frac{\sin(\Omega_c \cdot n)}{n \cdot \pi} \quad \Omega_c = 2 \cdot \pi \cdot \frac{f_c}{f_S}, \quad n = \left\{ \frac{-N}{2}, \dots, \frac{N}{2} \right\}$$

Für den Wert von $h_{d_{TP}}[0]$ muss aufgrund der Nulldivision die Regeln von L'Hospital angewendet werden (normalisiert auf f_S):

$$\frac{d}{dn} \left(\frac{\sin(2 \cdot f_c \cdot \pi \cdot n)}{\pi \cdot n} \right) \Big|_{n=0} = \frac{2 \cdot f_c \cdot \pi \cdot \cos(2 \cdot f_c \cdot \pi \cdot 0)}{\pi} = 2 \cdot f_c \quad h_{d_{TP}}[0] = \frac{2 \cdot f_c}{f_S}$$

1. Ideale Tiefpass Übertragungsfunktion
2. Ideale Impulsantwort (nicht endlich)
3. Begrenzen der Impulsantwort mittels Fenster
4. Geshiftete Impulsantwort zur praktischen Umsetzung



Vorgehen für Band- und Hochpass:

Band- und Hochpass-Filter werden durch die Summe oder Differenz von Tiefpass-Filtern mit unterschiedlichen Cutoff-Frequencies bestimmt. Beispiel von Hochpass: Tiefpass mit $f_c = f_S/2$ minus ein anderer Tiefpass mit der f_c . Die Impulsantwort des Hochpassfilters lautet demnach:

$$\begin{aligned} h_{d_{HP}}[n] &= \frac{\sin(n \cdot \pi)}{n \cdot \pi} - \frac{\sin(\Omega_c \cdot n)}{n \cdot \pi} = -h_{d_{TP}}[n], \text{ wobei: } h_{d_{HP}}[0] = 1 - h_{d_{TP}}[0] \\ &= \{..., -h_{d_{TP}}[-2], -h_{d_{TP}}[-1], 1 - h_{d_{TP}}[0], -h_{d_{TP}}[1], -h_{d_{TP}}[2], ... \} \end{aligned}$$

Zusammenhang zwischen Overshoot, Übergangsband, Ordnung & Window-Typ:

Das Anwenden des Windows entspricht einer Faltung des idealen Frequenzgangs mit einem Rechteckfenster. Dadurch entsteht ein Overshoot bei den Übergängen der Bänder. Der Overshoot kann durch erhöhen von N nicht reduziert werden, jedoch wird dadurch die Breite des Übergangsbereichs verringert. Um den Overshoot zu verringern, muss ein anderes Windows als das Rechteck eingesetzt werden. Das Übergangsband wird dadurch jedoch etwas breiter.

5.2.3 Alternative Designmethoden

Equiripple: Erzielt teils tiefere Filter-Ordnung N für dieselben Spezifikationen, da der Rippel im Durchlass- und Stoppband gewichtet werden kann. Der Filter weist dabei eine konstante Rippel-Amplitude auf.

Frequenz-Sampling: Man sampelt den idealen Frequenzgang $H_d(\Omega)$ an N Sample-Punkten in identisch grossem Abstand. Durch Anwenden der IDFT zwischen 0 ... 2π erhält man die Impulsantwort.

5.2.4 FIR-Kammfilter

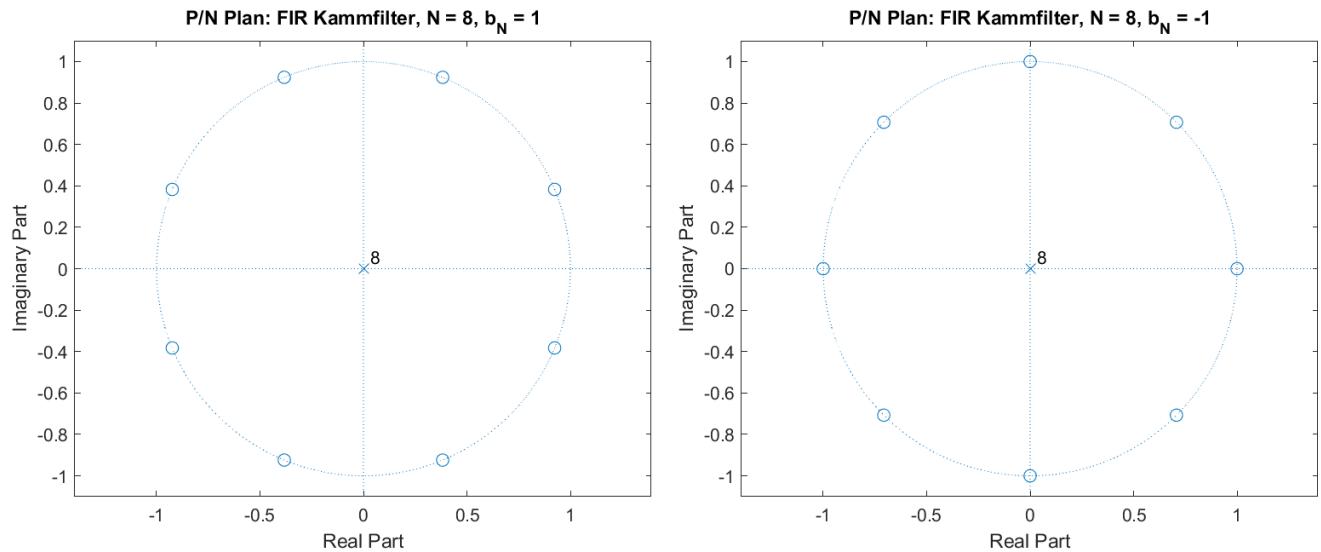
z-Übertragungsfunktion des Kammfilters N -ter Ordnung:

$$H(z) = 1 \pm z^{-N}$$

Impulsantwort:

$$h[k] = [b_0 \ 0 \dots 0 \ b_N] = [1 \ 0 \dots 0 \ \pm 1]$$

Alle N Pole sind im Ursprung. Die Nullstellen ergeben sich aus den Einheitswurzeln, bei $b_N = 1$ haben diese noch eine Phasenverschiebung (siehe linker Fall unten).



5.2.5 Kanalausgleich mittels FIR

Angenommen, ein Kanal verzerrt das Signal durch Echos. Die Impulsantwort ist endlich. Mittels Channel Equalization kann der Kanal entzerrt werden. Dazu eignet sich folgendes Vorgehen, um die Koeffizienten des Filters in Echtzeit zu berechnen. Der Aufwand bezieht sich auf die Anzahl realer Multiplikationen (1 cpl Mul = 2 real Mul):

1. FFT der Impulsantwort (Aufwand: $\left[\frac{N}{2} \cdot \log_2 N\right]$)
2. Invertierung aller Spektralwerte (Aufwand: $[N]$)
3. IFFT des invertierten Spektrums (Aufwand: $\left[\frac{N}{2} \cdot \log_2 N\right]$)

Totaler Aufwand an realwertigen Multiplikationen: $2N + 2N \log_2 N$

5.3 IIR-Filter

5.3.1 Eigenschaften

Bei IIR-Filtern ist die Ordnung des Nenners $M > 0$, es existieren also Polstellen bei $z \neq 0$. Die Impulsantwort dieser Filter ist theoretisch unendlich. Generell kann mit einem IIR-Filter mit tieferer Ordnung derselbe Amplitudengang wie mit einem FIR-Filter erreicht werden. Die Laufzeit verkürzt sich dadurch, eine konstante Gruppenlaufzeit lässt sich jedoch nicht realisieren. Durch Quantisierungsfehler können IIR-Filter instabil werden. Tabellarischer Vergleich:

criteria	FIR	IIR
stability	always stable	not guaranteed, must be verified for actual implementation
filter performance / filter order	higher order for same amplitude response	lower filter order for same amplitude response
phase response	exactly linear possible	linear phase not possible, introduces phase distortion
Arbitrary frequency response	easy to realize	more difficult to realize
design process	optimum design only with CAD, manual design possible	basic filter characteristics can be designed manually
design with analog filter model	no direct conversion possible	standard design method
finite word width effects	relatively low	sensitive to quantization and scaling
implementation	non-critical in HW and SW, direct support in DSPs	more complicated, requires detailed analysis of word widths

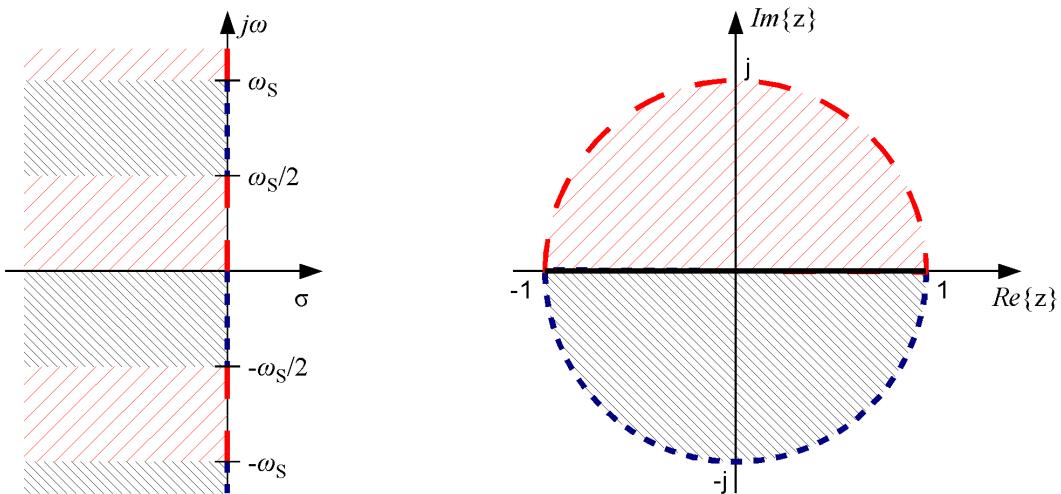
5.3.2 IIR-Design mittels Impulsinvarianz-Transformation (Tiefpass)

Die Impulsantwort des digitalen IIR-Filters wird durch Sampling der Impulsantwort des analogen Prototyp-Filters ermittelt. Um Aliasing zu verhindern, muss f_S mindestens doppelt so gross wie die höchste Durchlass-Frequenz des analogen Prototypen-Filters sein. Die Methode ist demnach nur für Tiefpass, jedoch nicht für andere Filter geeignet.

$$h[n] = h_a(n \cdot T_S) \quad n = 0, 1, 2, \dots$$

Umformung von der p -Ebene zur z -Ebene:

$$z = e^{pT_S}$$



5.4 Implementationsaspekte

Die Wahl der Sampling-Frequenz muss gut durchdacht sein. Bei zu hoher Sampling Frequenz entstehen Probleme:

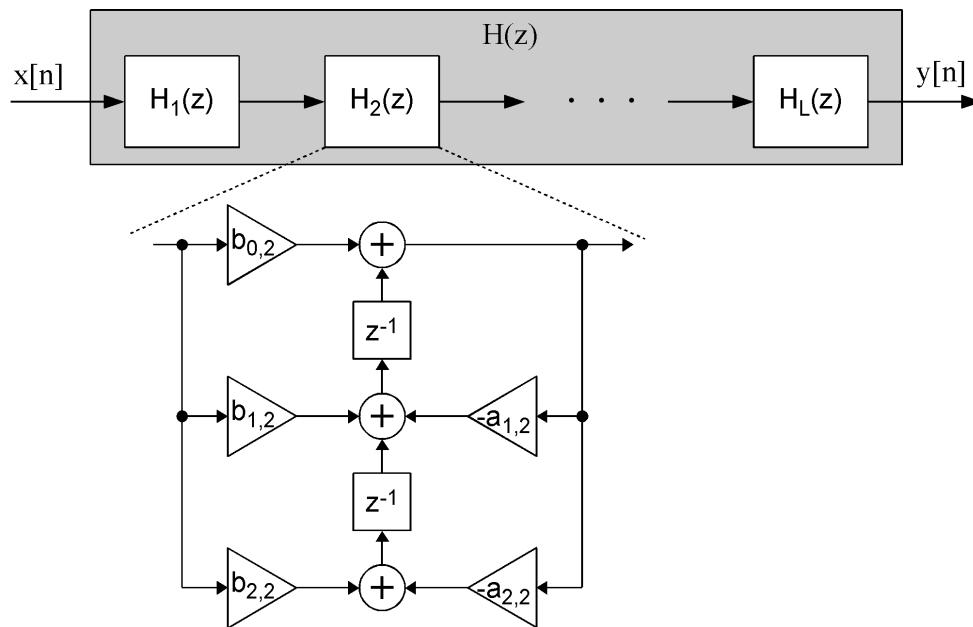
- Größerer Rechenaufwand, da mehr Samples in vorgegebener Zeit bearbeitet werden müssen.
- Je kleiner der Frequenzabstand zwischen Pass- und Stoppband wird, desto steiler wird die Filterkurve. Dies wiederum erzwingt eine höhere Ordnung, was erneut den Rechenaufwand erhöht.
- Je näher Pol- und Nullstellen der Transferfunktion in der z -Ebene kommen, desto höher die Sensitivität gegenüber Quantisierungseffekten der Koeffizienten.

5.4.1 Implementation von IIR-Filtern

Der IIR-Filter kann als eine Kette von "biquads" implementiert werden. Dabei wird die Übertragungsfunktion in L Blöcke aufgeteilt, welche jeweils einen konjugiert-komplexen Pol haben:

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

Die Sub-Übertragungsfunktionen werden dabei in der "Direct Form I" oder in der "Transposed Direct Form I" implementiert. Beispiel mit biquads zweiter Ordnung in der "Transposed Direct Form II":



Matlab-Befehl zur Umwandlung der Koeffizienten von $H(z)$ in die Koeffizienten von $H_1(z), \dots, H_L(z)$: `tf2sos()`

5.4.2 Fixkomma-Implementation

Die Implementation im Fixkomma-Format ist resourceneffizienter als das Fliesskomma-Format. Bei einer Fix-Point Implementierung mit W Bits (b_k) und F Nachkommastellen ist der dezimale Wert:

$$D_{ufix} = \sum_{k=0}^{W-1} b_k \cdot 2^{k-F}$$

Für Zahlen im Zweierkomplement ergibt sich der dezimale Wert wie folgt:

$$D_{sfix} = -b_{W-1} \cdot 2^{W-F-1} + \sum_{k=0}^{W-2} b_k \cdot 2^{k-F}$$

Overflow: Kann durch Hinzufügen zusätzlicher MSBs, einer Sättigung (Begrenzen der Zahlen auf grösste repräsentierbare Zahl) oder "wrap-around" (zählt beim kleinsten Wert weiter) erreicht werden.

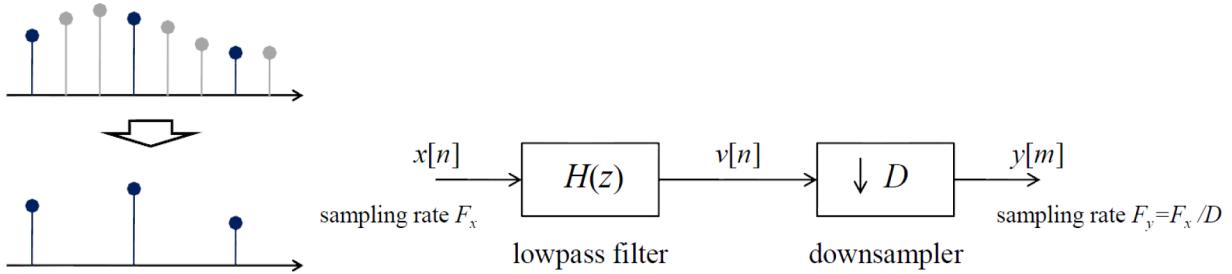
Underflow: Das Resultat kann aufgrund fehlender LSBs (Auflösung) nicht angezeigt werden. Zu den Massnahmen zählen Abschneiden (LSBs vernachlässigen) oder Runden (auf die nächst gelegene repräsentierbare Zahl).

Kapitel 6

Multirate und Filterbänke

6.1 Downsampling/Decimation

Um ein Signal downzusampeln, wird einfach nur jedes D -te Sample verwendet.



Dabei muss beachtet werden, dass das Abtasttheorem noch eingehalten wird. Das Signal muss zuerst mit einem Tiefpass gefiltert werden. Die Frequenzantwort des TP ist idealerweise:

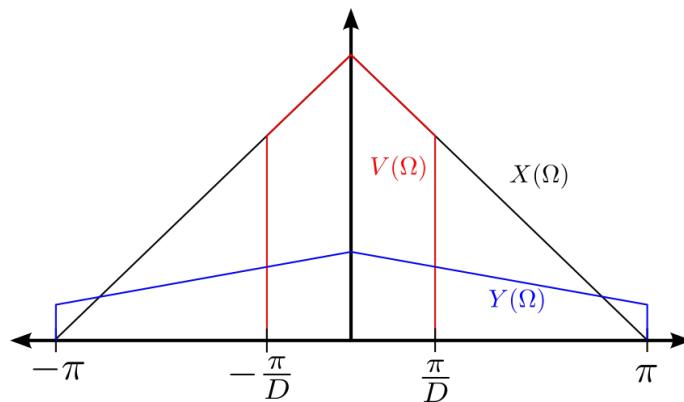
$$H(\Omega) = \begin{cases} 1 & \text{if } \Omega \in [-\pi/D, \pi/D] \\ 0 & \text{otherwise} \end{cases}$$

Das Resultat kann beschrieben werden als $y[m] = v[nD]$. Das Frequenzspektrum wird um den Faktor D horizontal gespreizt und gleichzeitig vertikal gestaucht. Für einen *idealen* TP-Filter gilt für das resultierende Spektrum $Y(\Omega)$:

$$Y(\Omega) = \frac{1}{D} V(\Omega/D)$$

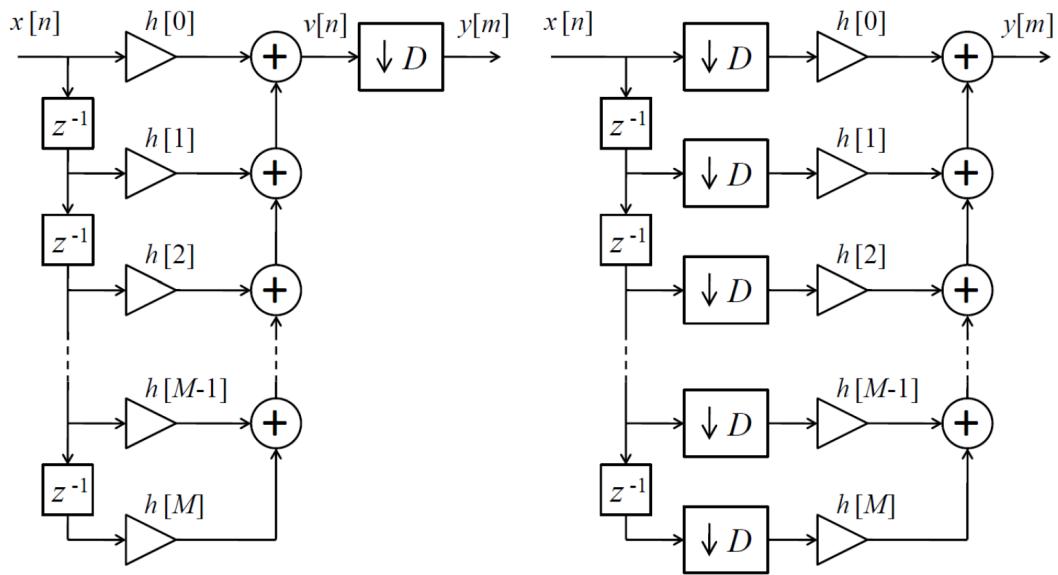
Allgemein für alle TP-Filter, wobei $d = 1, \dots, D-1$ Aliasing aufgrund nicht idealer TP-Filterung einbezieht:

$$Y(\Omega) = \frac{1}{D} \sum_{d=0}^{D-1} V(\Omega/D - 2\pi d/D)$$



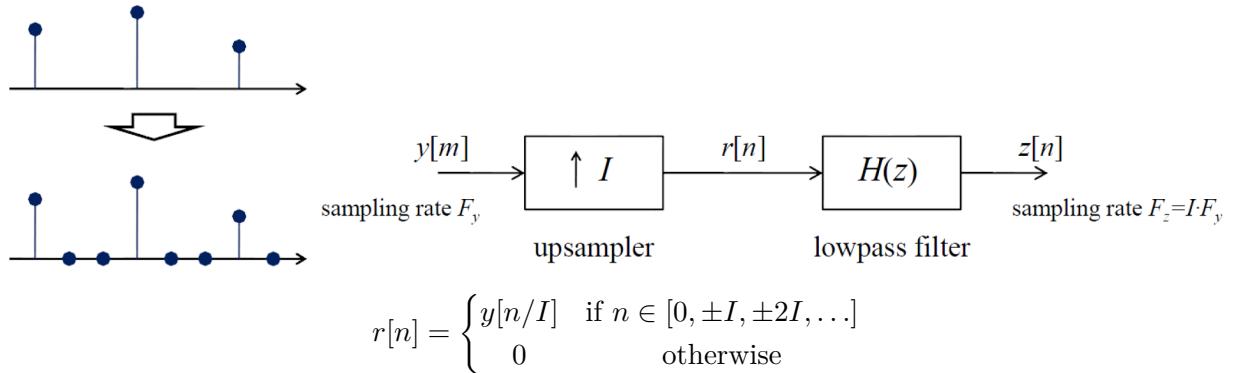
Als TP-Filter kann ein FIR Filter verwendet werden. Eine direkte Implementierung ist nicht effektiv, da $D-1$ vom Filter berechnete Werte vom Downampler weggeworfen werden. Mit dem Downampler vor dem Filter müssen weniger Berechnungen durchgeführt werden. Die Multiplikatoren können mit einer tieferen Samplingrate betrieben werden.

Direkte, ineffiziente Implementation (links) & effizientere Implementation (rechts):



6.2 Upsampling/Interpolation

Bei einem Upsampler mit dem Faktor I werden zwischen zwei Samples jeweils $I - 1$ Nullen eingefügt.

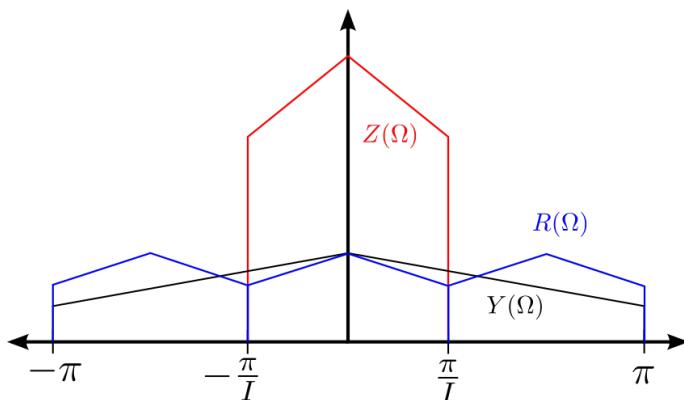


Die Abtastfrequenz F_z ist I -mal höher als die Abtastfrequenz F_y von $y[m]$. Das Spektrum des upgesampelten Signal ist:

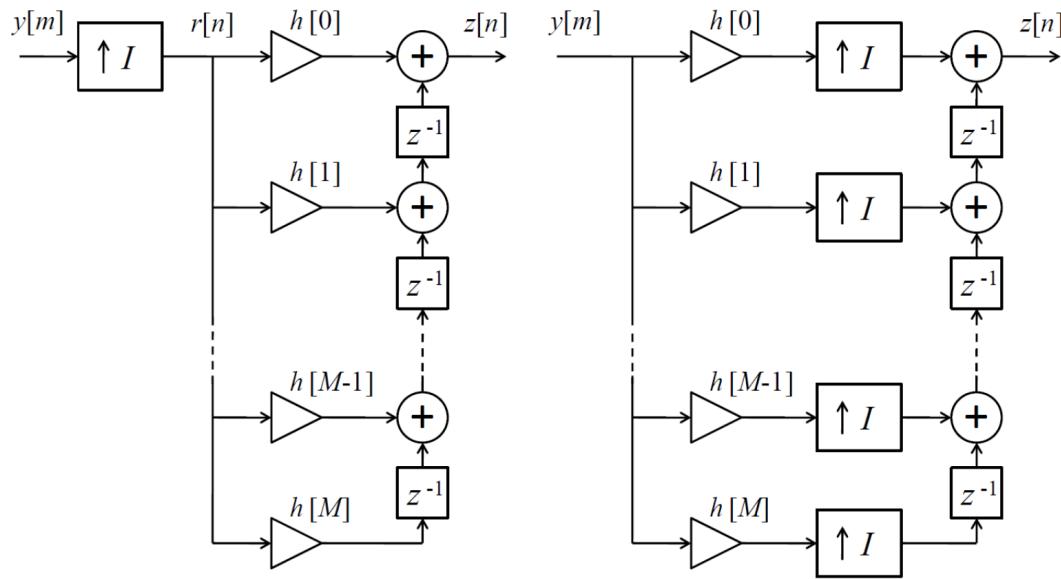
$$R(\Omega) = \sum_{n=-\infty}^{\infty} r[n] e^{-j\Omega n} = \sum_{m=-\infty}^{\infty} r[m] e^{-j\Omega m} = Y(I\Omega)$$

Damit das Frequenzspektrum von $y[m]$ nicht periodisch mit der Periode von $2\pi/I$ ist, ist eine Tiefpassfilterung nach dem Upsampling notwendig. Er glättet das Signal und führt dadurch die Interpolation durch. Der ideale TP-Filter hat die Frequenzantwort:

$$H(\Omega) = \begin{cases} I & \text{if } \Omega \in [-\pi/I, \pi/I] \\ 0 & \text{otherwise} \end{cases}$$



Ein FIR oder IIR Filter kann als TP-Filter verwendet werden. Eine direkte Implementierung (siehe linkes Blockschaltbild) ist ineffektiv, da der Filter auch auf die Null-Werte angewandt wird. Rechts muss der Multiplikator nur mit der Frequenz F_y arbeiten, im Vergleich zu links mit der höheren F_z nach dem Upsampling.



6.3 Polyphasen Filter

Dienen zur effizienteren Implementation von Filtern. Jede i -te Komponente von $p_i[k]$ beinhaltet den M -ten Koeffizienten der Impulsantwort $h[k]$, alle zusammen beinhalten die gesamten Informationen. Insgesamt existieren also immer M Polyphasenfilter. Die um M downgesampelte Impulsantwort $h[k]$ ist definiert als:

$$p_i[k] = h[kM + i] \quad i = 0, 1, \dots, M - 1$$

Dazugehörende z-Transformation für die Polyphasen-Komponenten $P_i(z)$:

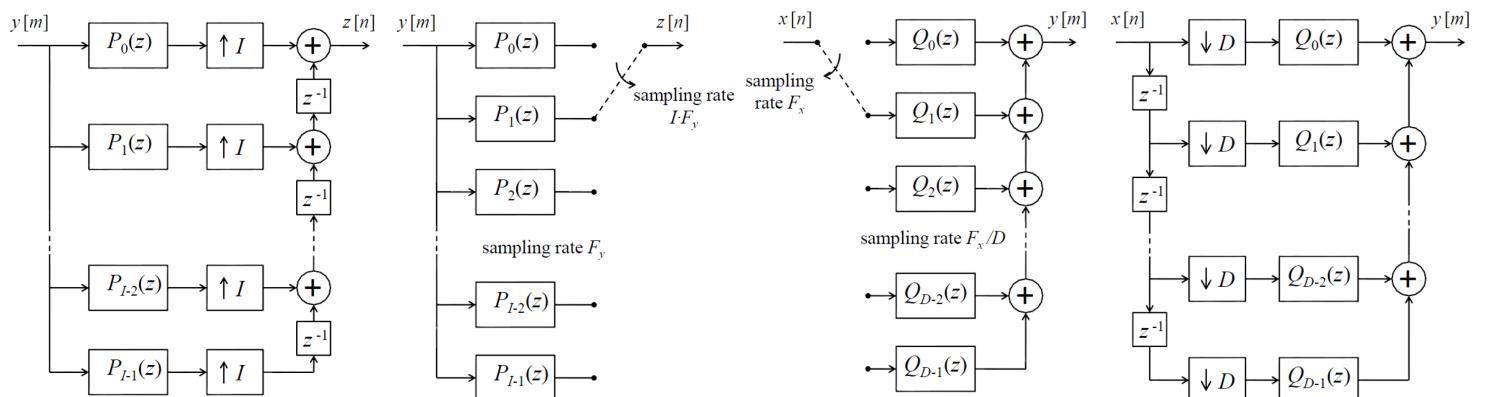
$$P_i(z) = \sum_{k=-\infty}^{\infty} p_i[k]z^{-k} \quad i = 0, 1, \dots, M - 1$$

Somit gilt für das Spektrum:

$$H(z) = \sum_{i=0}^{M-1} z^{-i} P_i(z^M)$$

Der Vorteil liegt darin, dass die Filter bei einem Upsampling mit der Rate F_y statt der höheren Rate F_z arbeiten können. Die einzelnen Werte müssen anschliessend nur noch mit der $F_z = I \cdot F_y$ zusammengesetzt werden.

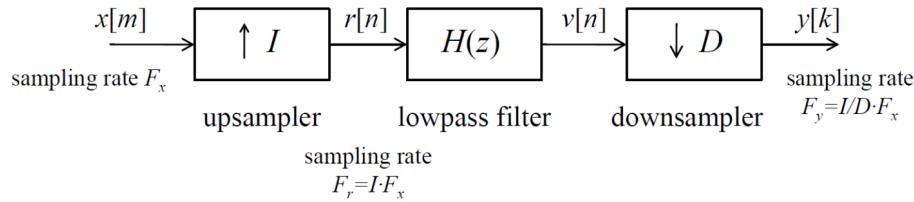
Bei einem Downsampling können die Filter mit F_x/D arbeiten, während die Werte mit F_x auf die einzelnen Polyphasen-Komponenten aufgeteilt werden.



6.4 Implementation rationaler Abtastraten

Wenn ein up- bzw. downsample Faktor gefordert ist, welcher nicht ganzzahlig ist, kann dieser mit $\frac{I}{D}$ dargestellt werden.

Wenn zuerst der Downampler kommt, gehen Informationen verloren. Andersrum entsteht eine hohe Abtastfrequenz dazwischen. Es ist jedoch vorzuziehen, zuerst das Upsampeln durchzuführen, danach das Downsamplen. So kann der Tiefpass der Interpolation mit dem Tiefpass der Decimation kombiniert werden.



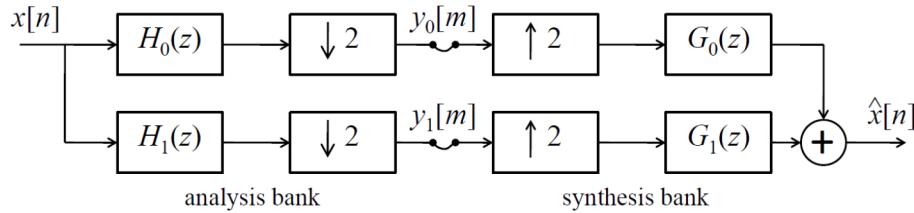
Die Anzahl an Koeffizienten p für einen FIR-TP-Filter mit Länge N berechnet sich wie folgt:

$$p = \frac{N}{I}$$

Es werden p Multiplikationen sowie $p - 1$ Additionen notwendig pro Output Sample des Filters.

6.5 Quadratur Spiegel Filterbank

Um den Datenverlust beim Downsamplen zu kompensieren, kann das Signal über zwei Kanäle übertragen werden. Der eine Kanal filtert das Signal mit einem TP $H_0(z)$, der Andere mit einem HP $H_1(z)$.



Die DTFT der zwei generierten Signale $y_0[m]$ und $y_1[m]$ sind:

$$Y_{0/1} = \frac{1}{2} \left(H_{0/1} \left(\frac{\Omega}{2} \right) X \left(\frac{\Omega}{2} \right) + H_{0/1} \left(\frac{\Omega}{2} - \pi \right) X \left(\frac{\Omega}{2} - \pi \right) \right)$$

Das Spektrum des synthetisierten Signals $\hat{x}[n]$ ist:

$$\begin{aligned} \hat{X}(\Omega) &= \frac{1}{2} \left(H_0(\Omega)G_0(\Omega) + H_1(\Omega)G_1(\Omega) \right) \cdot X(\Omega) \\ &\quad + \underbrace{\frac{1}{2} \left(H_0(\Omega - \pi)G_0(\Omega) + H_1(\Omega - \pi)G_1(\Omega) \right)}_{\text{alias term}} \cdot X(\Omega - \pi) \end{aligned}$$

Um keinen Aliasterm zu haben müssen die Bedingungen $G_0(\Omega) = H_1(\Omega - \pi)$ und $G_1(\Omega) = -H_0(\Omega - \pi)$ erfüllt sein. Oft wird ebenfalls nur ein Filter $H(\Omega)$ gewählt. Meist wird dieses im z -Bereich bestimmt/vorgegeben und kann dann mit $z = e^{j \cdot \Omega}$ umgewandelt werden. Die entsprechenden Filter können dann wie folgt abgeleitet werden:

$$H_0(\Omega) = H(\Omega)$$

$$H_1(\Omega) = H(\Omega - \pi)$$

$$G_0(\Omega) = H(\Omega)$$

$$G_1(\Omega) = -H(\Omega - \pi)$$

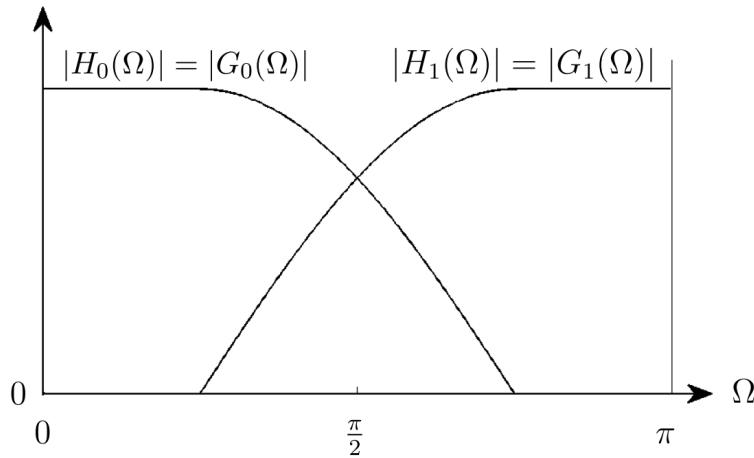
Anschliessend werden sie in den z -Bereich zurücktransformiert, z.B. $H_0(\Omega) \circlearrowleft H_0(z)$ oder $G_1(\Omega) \circlearrowleft G_1(z)$.

Dank dieser Transformation können die Impulsantworten einfach ermittelt werden.

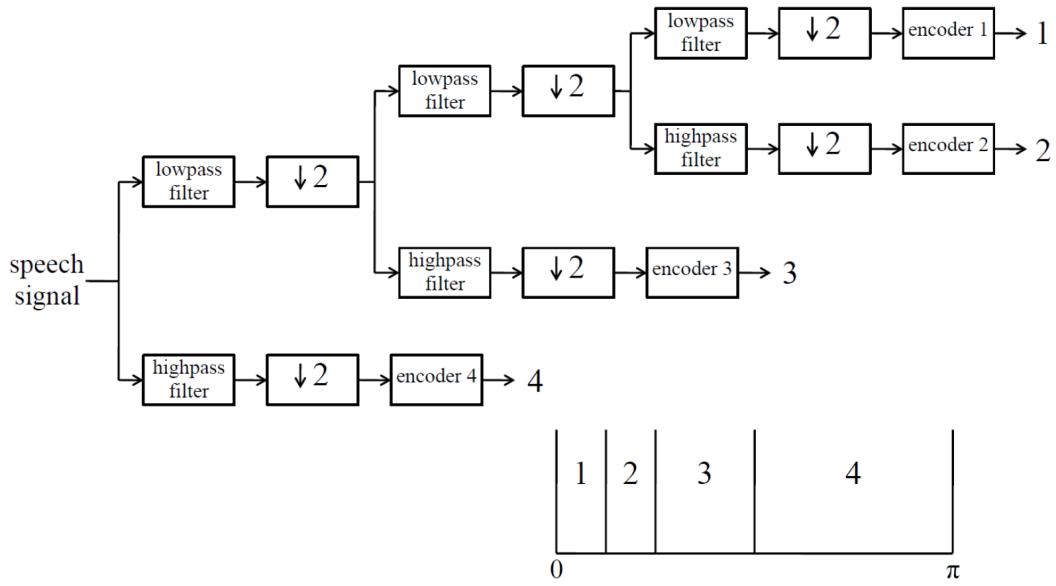
Weiter gilt für das synthetisierte Signalspektrum:

$$\hat{X}(\Omega) = T(\Omega)X(\Omega) \quad , \text{ wobei: } T(\Omega) = \frac{1}{2} \left(H^2(\Omega) - H^2(\Omega - \pi) \right)$$

Um perfekte Rekonstruktion zu haben, muss $T(\Omega)$ eine Allpass- Charakteristik mit linearer Phasenverschiebung aufweisen (konstante Amplitudenverstärkung, konstante Zeitverzögerung). Dies kann am besten mit einem symmetrischen FIR-Filter (ungerade Ordnung) erreicht werden. Der einseitige Frequenzgang sieht dann wie folgt aus:



Diese Filter werden z.B. in der Speicherung von Sprachsignalen verwendet. Dabei werden Signale in einzelne Frequenzbänder zerlegt. Tieferen Frequenzen kommen dabei vermehrt vor und können so besser (mehr Bits) quantisiert werden.

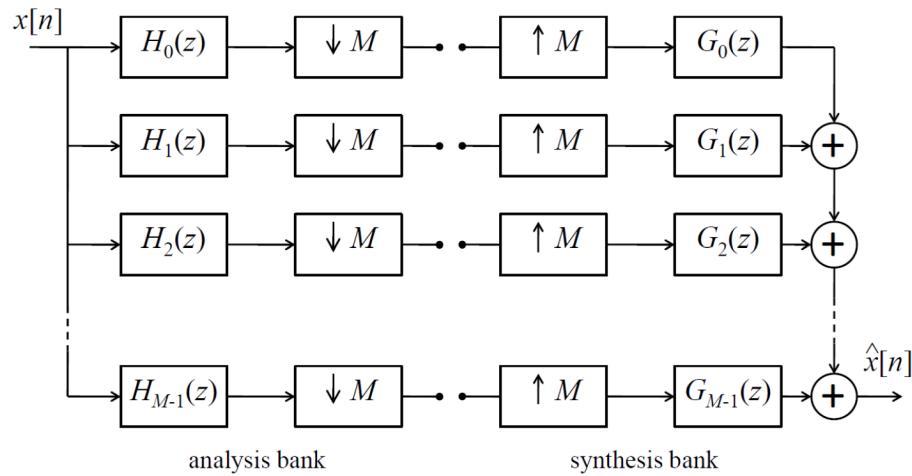


6.6 DFT Filterbank

Bei dieser Filterbank handelt es sich um eine Generalisierung der Quadratur Spiegel Filterbank. Die Anzahl Channels wird von 2 auf M erhöht. Es ergeben sich M Filter in der Analyse & Synthese. Die Subband-Signale können dadurch mit tieferer Sampling-Rate abgetastet werden. Es wird unterschieden zwischen:

- **Critical sampling:** Die Anzahl der Filter $H_0(z), H_1(z), \dots, H_{M-1}$ entspricht dem Downsampling M .
- **Oversampling:** Anzahl Kanäle ist grösser als der Downsampling Faktor.
- **Undersampling:** Anzahl Kanäle ist kleiner als der Downsampling Faktor.

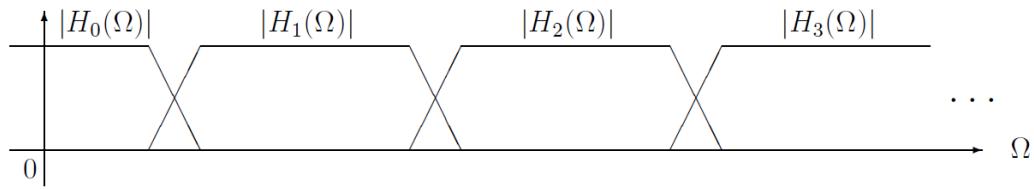
Mit Undersampling ist *keine* perfekte Rekonstruktion des Eingangssignals $x[n]$ möglich.



Um ein simpleres Design zu erreichen, wird der Prototype-Filter $H(z)$ (TP) jeweils frequenzmässig verschoben. Das selbe gilt für die Synthesis-Bank mit $G(z)$.

$$H_l(z) = H(z \cdot e^{-j2\pi l/M}) \quad l = 0, 1, \dots, M-1$$

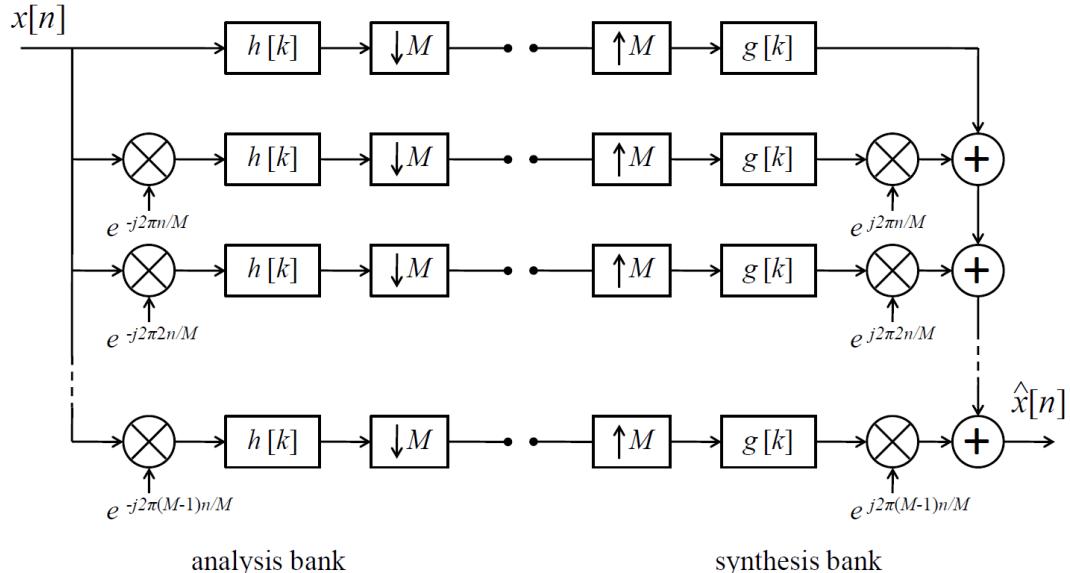
$$G_l(z) = G(z \cdot e^{-j2\pi l/M}) \quad l = 0, 1, \dots, M-1$$



Frequenzgang:

$$H_l(\Omega) = H(\Omega - 2\pi l/M)$$

Die Namensgebung ist auf die folgende Implementierung zurückzuführen. Das Eingangssignal wird im l -ten Kanal um $2\pi l/M$ geshiftet.



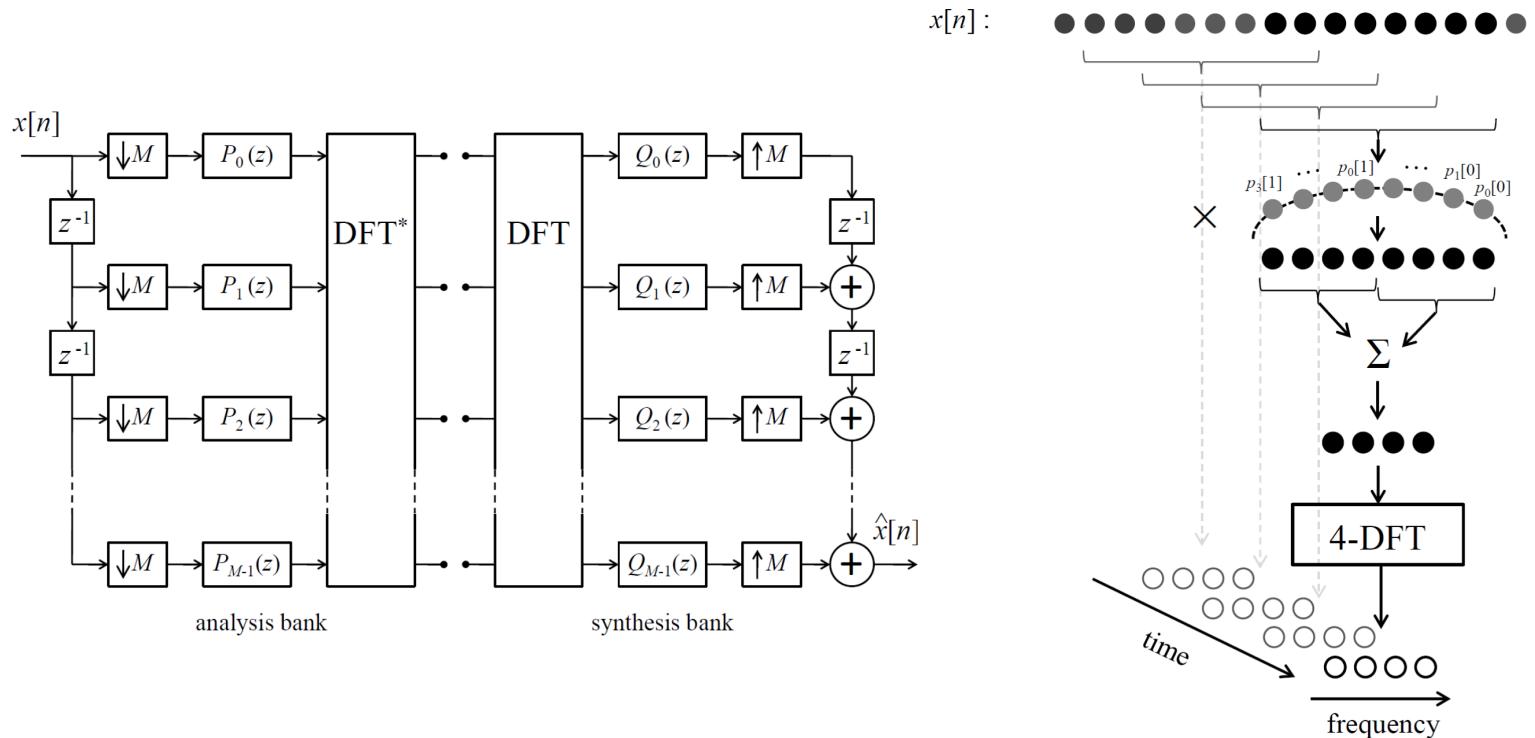
Die Impulsantworten der Analysis & Synthesis-Filter lauten:

$$h[k] = \begin{cases} 1 & \text{if } k \in \{0, 1, \dots, M-1\} \\ 0 & \text{otherwise} \end{cases}$$

$$g[k] = \begin{cases} \frac{1}{M} & \text{if } k \in \{0, 1, \dots, M-1\} \\ 0 & \text{otherwise} \end{cases}$$

Mit Polyphase kann dies in eine DFT mit Polyphase Filter aufgeteilt werden:

$$H_l(z) = \sum_{i=0}^{M-1} e^{j2\pi li/M} \cdot (z^{-i} \cdot P_i(z^M)) \quad l = 0, 1, \dots, M-1$$



Filterbank Ausgangssignal im z -Bereich:

$$\hat{X}(z) = \sum_{i=0}^{M-1} z^{-i} \cdot Q_i(z^M) \cdot \sum_{l=0}^{M-1} Y_l(z^M) \cdot e^{j2\pi li/M}$$

Vorteile der DFT-Filterbank mit Polyphasen-Struktur:

- Es handelt sich um FIR-Filter bei den Polyphase-Komponenten. Dabei handelt es sich um downgesampelte Varianten von $h[k]$ und $g[k]$. Diese Filter haben typischerweise einen kleinen Komplexitätsgrad.
- Für die DFT-Berechnungen sind effiziente FFT-Algorithmen verfügbar.

Kapitel 7

Zufallssignale

Stochastische Prozesse in Kombination mit LTI Systemen sind wichtige Werkzeuge zur Modellierung von Zufallssignalen. Sie sind besonders wichtig in der Audio- und Kommunikationstechnik.

7.1 Autokorrelation und Spektrum

7.1.1 Betrachtung im Zeitbereich

Mittelwert eines Zufallssignals $x[n]$:

$$m_x = E\{x[n]\}$$

Autokorrelation eines Zufallssignals $x[n]$:

$$\gamma_{xx}[m] = E\{x^*[n] \cdot x[n+m]\}$$

Wobei $\gamma_{xx}[0]$ die Leistung P_x repräsentiert.

Die Autokorrelation ist gespiegelt symmetrisch an der y-Achse. Für Zufallssignale gilt:

$$\gamma_{xx}[m] = \gamma_{xx}^*[-m]$$

Signale, welche den Mittelwert nicht bei Null haben, werden teils mit der Autokovarianz statt der Autokorrelation charakterisiert:

$$\begin{aligned} c_{xx} &= E\{(x[n] - m_x)^* \cdot (x[n+m] - m_x)\} \\ y_{xx} &= c_{xx}[m] + m_x^2 \end{aligned}$$

Die Autokovarianz bei $m = 0$ repräsentiert die Varianz:

$$E\{|x[n] - m_x|^2\}$$

7.1.2 Betrachtung im Frequenzbereich

Zufallssignale weisen eine nicht endliche Energie auf, es existiert also keine Fouriertransformierte. Es kann jedoch ein Intervall $[n_1, n_2]$ mit der DTFT betrachtet werden. Spektrum durch die DTFT im Intervall $[n_1, n_2]$:

$$X_{[n_1, n_2]}(\Omega) = \sum_{n=n_1}^{n_2} x[n] \cdot e^{-j\Omega n}$$

Das **Leistungsdichte-Spektrum** (power density spectrum) ergibt sich aus der DFT der Autokorrelationssequenz:

$$\Gamma_{xx}(\Omega) = \sum_{m=-\infty}^{\infty} \gamma_{xx}[m] \cdot e^{-j\Omega m}$$

Die Leistung lässt sich durch Integration über $\Gamma_{xx}(\Omega)$ oder $\gamma_{xx}[0]$ ermitteln. Bei realen Signalen ist die Form "gerade" (gespiegelt an der y-Achse). Es reicht also für reale Signale, im Bereich $\Omega = [0, \pi]$ zu berechnen.

$$P_x = \frac{1}{2} \int_{-\pi}^{\pi} \Gamma_{xx}(\Omega) d\Omega \quad \text{oder} \quad P_x = \gamma_{xx}[0]$$

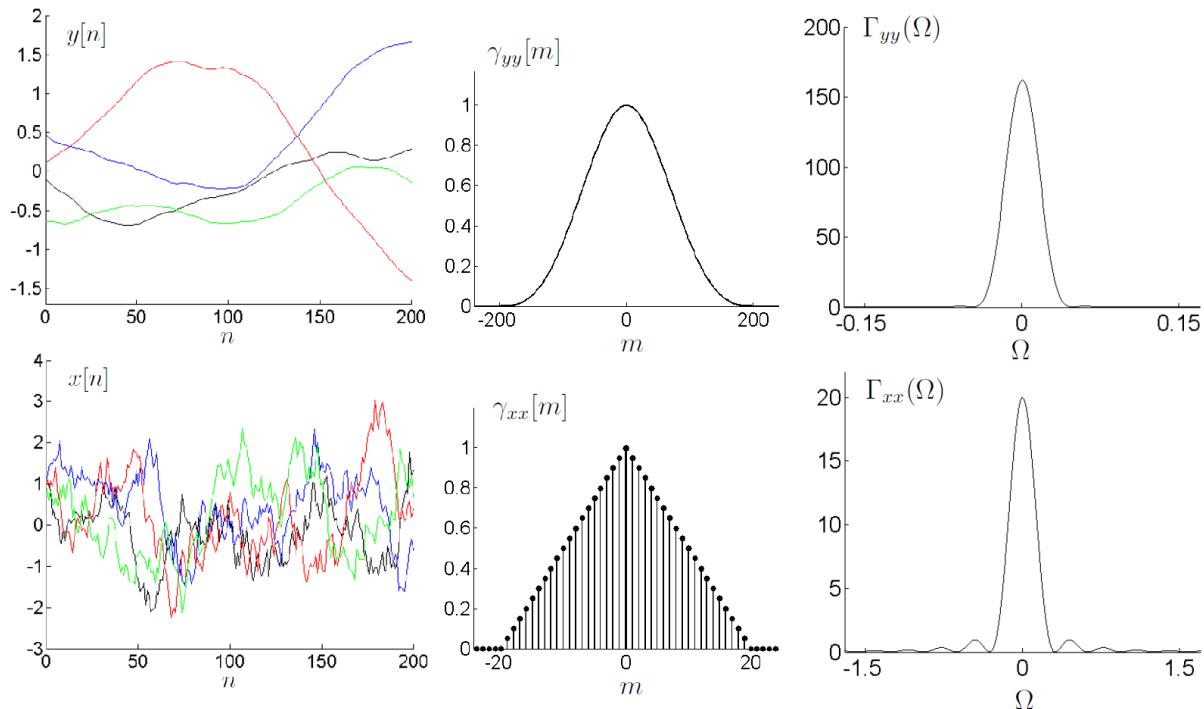
Die Autokorrelation und das Leistungsdichte-Spektrum enthalten also die selben Informationen. Aufgrund der fehlenden Phaseninformation ist es jedoch nicht möglich, $x[n]$ aus $\Gamma_{xx}(\Omega)$ oder $\gamma_{xx}[m]$ zu rekonstruieren.

Weisses Rauschen:

Aufeinander folgende Samples sind vollständig unkorreliert, das Leistungsdichte-Spektrum ist über Ω konstant. Es beinhaltet alle Frequenzen mit uniformer Verteilung. Die Autokorrelation lautet:

$$\gamma_{ww}[m] = \begin{cases} \sigma_w^2 & \text{if } m = 0 \\ 0 & \text{if } m \neq 0 \end{cases}$$

Im folgenden werden zwei unterschiedliche Signale miteinander verglichen. Besonders die Größenordnungen der x-Achsen bei der Autokorrelation und des Spektrums sind interessant.



7.2 Spectral Shaping durch LTI Systeme

Mittelwert des Ausgangssignals $y[n]$ nach einem LTI-System H :

$$E\{y[n]\} = m_y = m_x \cdot \sum_{k=-\infty}^{\infty} h[k] = H(0) \cdot m_x$$

, da der Frequenzgang gegeben ist als:

$$H(\Omega) = \sum_{k=-\infty}^{\infty} h[k] \cdot e^{-j\Omega k}$$

Autokorrelation von $y[n]$ (i geht wie k von $-\infty$ bis ∞):

$$\gamma_{yy}[m] = h^*[-i] * \gamma_{xx}[m] * h[k]$$

Aus der erhaltenen Formel für Mittelwert und der Autokorrelation lässt sich folgende Aussage treffen:

→ Der Ausgang eines LTI-Systems mit stationärem Zufallssignal am Eingang ist erneut stationär.

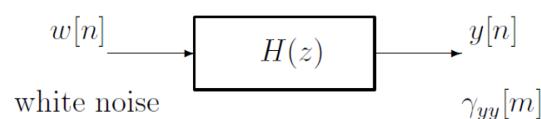
Leistungsdichte-Spektrum am Ausgang:

$$\Gamma_{yy}(\Omega) = H^*(\Omega) \cdot \Gamma_{xx}(\Omega) \cdot H(\Omega) = |H(\Omega)|^2 \cdot \Gamma_{xx}(\Omega)$$

7.3 Lineare Modelle für Stochastische Prozesse

7.3.1 Filtern von White Noise

Ein Whitenoise-Signal $w[n]$ durchläuft ein LTI-System. Am Ausgang entsteht farbiges Rauschen.



Die z -transformierte der Autokorrelations-Sequenz für Ein- und Ausgang des Systems lauten:

$$\Gamma_{ww}(z) = \sigma_w^2 \quad \text{sowie} \quad \Gamma_{yy}(z) = H(z^{-1}) \cdot \Gamma_{ww}(z) \cdot H(z)$$

Angenommen, ein Zufallssignal mit einem Spektrum sei gegeben, und das Filter $H(z)$ sei gesucht. Es gilt für die Übertragungsfunktion:

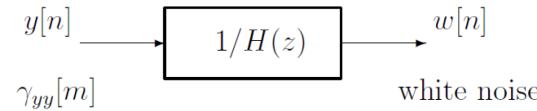
$$H(z) = \frac{B(z)}{A(z)} = \frac{\sum_{k=0}^M b_k \cdot z^{-k}}{1 + \sum_{k=1}^N a_k \cdot z^{-k}}$$

Die Null- und Polstellen von $A(z)$ und $B(z)$ seien alle im Einheitskreis, das System $H(z)$ demnach kausal und stabil. Ausgang des LTI:

$$\Gamma_{yy}(z) = \sigma_w^2 \cdot H(z) \cdot H(z^{-1}) = \sigma_w^2 \cdot \frac{B(z) \cdot B(z^{-1})}{A(z) \cdot A(z^{-1})}$$

7.3.2 Noise Whitening

In diesem Fall wird stationäres Zufallsrauschen in weisses Rauschen umgewandelt.



Der Filter kann umgesetzt werden, indem die inverse Übertragungsfunktion $\frac{1}{H(z)}$ angewendet wird.

Für das Ausgangssignal des Noise Whitening Filters ergibt sich:

$$w[n] = \frac{1}{b_0} \left(y[n] + \sum_{k=1}^N a_k \cdot y[n-k] - \sum_{k=1}^M b_k \cdot w[n-k] \right)$$

7.3.3 Moving average model (MA)

$H(z)$ ist ein FIR-Filter mit Ordnung M welcher ein weisses Rauschen $w[n]$ in ein Zufallssignal transformiert.

$$y[n] = \sum_{k=0}^M b_k \cdot w[n-k]$$

Es gilt:

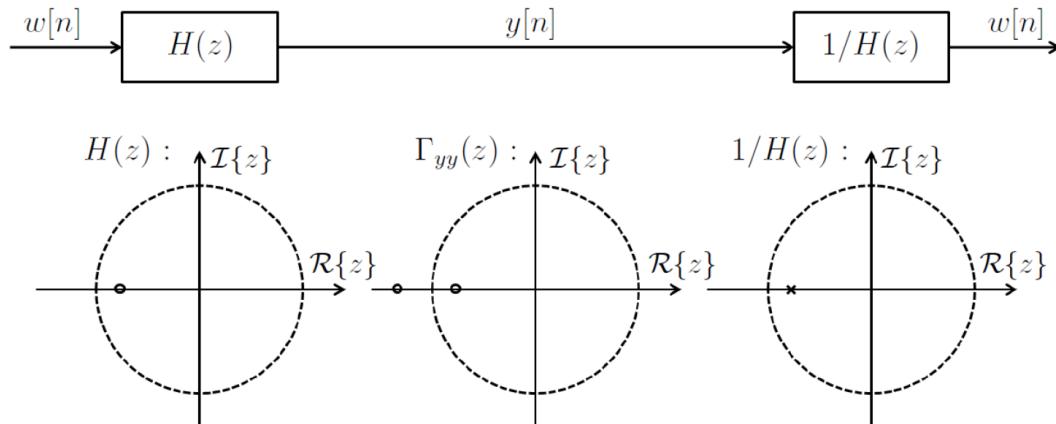
- $H(z)$ = kausales Filter *nur* mit Nullstellen
- $1/H(z)$ = kausales Filter *nur* mit Polstellen

Im Spezialfall $b_0 = b_1 = \dots = b_M = (M+1)^{-1}$ repräsentiert jedes Sample des Filterausgangs der Durchschnitt des Eingangssignals in einem verschiebbaren Fenster der Länge $M+1$.

Berechnung der Koeffizienten bei gegebener Autokorrelation γ_{yy} :

$$\gamma_{yy}[m] = \begin{cases} \sigma_w^2 \sum_{k=m}^M b_k \cdot b_{k-m}^* & \text{if } 0 \leq m \leq M \\ 0 & \text{if } m > M \\ \gamma_{yy}^*[-m] & \text{if } m < 0 \end{cases}$$

Beispiel eines MA-Modells:



7.3.4 Autoregressive model (AR)

Oft ist ein AR-Modell sinnvoller als MA. Die Zufallssequenz wird durch einen All-pole Filter generiert:

$$y[n] = w[n] - \sum_{k=1}^N a_k \cdot y[n-k]$$

Es gilt:

- $H(z) = \text{kausales Filter nur mit Polstellen}$
- $1/H(z) = \text{kausales Filter nur mit Nullstellen}$

Berechnung der Koeffizienten bei gegebener Autokorrelation γ_{yy} :

$$\gamma_{yy}[m] = \begin{cases} -\sum_{k=1}^N a_k \cdot \gamma_{yy}[m-k] & \text{if } m > 0 \\ \sigma_0^2 - \sum_{k=1}^N a_k \cdot \gamma_{yy}[m-k] & \text{if } m = 0 \\ \gamma_{yy}^*[-m] & \text{if } m < 0 \end{cases}$$

Yule-Walker Gleichungen: Das Gleichungssystem in Matrizenform lautet:

$$\begin{bmatrix} \gamma_{yy}[0] & \gamma_{yy}[-1] & \dots & \gamma_{yy}[-N] \\ \gamma_{yy}[1] & \gamma_{yy}[0] & \dots & \gamma_{yy}[-N+1] \\ \vdots & \vdots & & \vdots \\ \gamma_{yy}[N] & \gamma_{yy}[N-1] & \dots & \gamma_{yy}[0] \end{bmatrix} \cdot \begin{bmatrix} 1 \\ a_1 \\ \vdots \\ a_N \end{bmatrix} = \begin{bmatrix} \sigma_w^2 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

Achtung: Durch falsche Parametrisierung kann ein AR-Modell instabil werden!

7.3.5 ARMA model

Ist die Zusammensetzung des MA und AR Models. Dabei sind die Ordnungen der Polynome $B(z)$ und $A(z)$ 1 oder höher. Somit hat der Pol-Nullstellenplan Nullen und Pole. Dieser ist allerdings geeigneter als die Spezialfälle MA und AR, da weniger Koeffizienten für ein gutes Modell benötigt werden.

7.4 Abschätzung der Spektrums-Dichtefunktion

Gegeben sei eine Messserie $x[0], x[1], \dots, x[N-1]$ eines stationären stochastischen Prozesses.

7.4.1 Parameterlose Methode

Nachteil: Für jedes Ω_0 weisst die Schätzung $\hat{\Gamma}_{xx}(\Omega_0)$ eine grosse Varianz auf. Sie verändert sich auch nicht mit höherer Anzahl Samples.

Alternativ kann zuerst die Autokorrelation, daraus mittels Wiener-Khinchin-Theorem das Spektrum ermittelt werden. Es existieren die folgenden beiden Methoden:

- *Biased* autocorrelation estimator:

$$\hat{\gamma}_{xx}[m] = \frac{1}{N} \cdot \sum_{n=0}^{N-m-1} x^*[n] \cdot x[n+m] , \text{ für } m \text{ zwischen } 0 \text{ und } N-1$$
- *Unbiased* autocorrelation estimator:

$$\hat{\gamma}_{xx}[m] = \frac{1}{N-m} \cdot \sum_{n=0}^{N-m-1} x^*[n] \cdot x[n+m] , \text{ für } m \text{ zwischen } 0 \text{ und } N-1$$

Das Problem bei der “unbiased“ Variante ist, dass mit grösserem m der Bruch immer fragiler wird. Die “biased“ Methode erzielt meistens bessere Resultate aufgrund der Fehlerreduktion durch die höhere Anzahl an Samples.

7.4.2 Parameterbehaftete Methode

Falls Informationen über die Struktur der Rauschquelle vorhanden sind, können diese bei der Modellierung helfen. Auch ohne Informationen ist es aber möglich, mit einer parametrischen Methode ein passendes Modell zu ermitteln.

ARMA-Modelle sind weit verbreitet, da sie mit den Parametern fein angepasst werden können. Für Spektren mit Spitzen passt ein AR-Modell besser als ein MA-Modell.

Kapitel 8

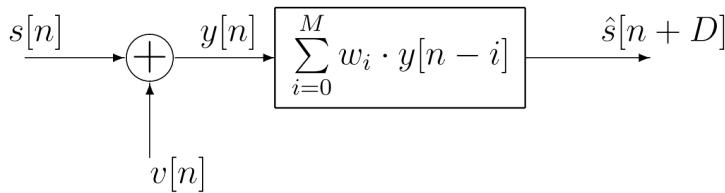
Optimale Lineare Filter

Problematik: Das ursprüngliche Signal aus einem verzerrten Signal mittels Filter abschätzen.

8.1 Wiener Filter

Ist optimal im mittleren quadratischen Fehler. Existieren für diskrete & kontinuierliche Signale.

Aufgabe ist es w_0, w_1, \dots, w_M so zu bestimmen, dass das Filter der Ordnung M die beste Schätzung ab liefert.



- $s[n]$: Ursprüngliches Signal
- $v[n]$: Störsignal (Rauschen)
- $\hat{s}[n + D]$: Schätzung von $s[n]$

Zurückgewonnenes Signal $\hat{s}[n + D]$.

- *Smoothing*: Bei $D < 0$ ist der Sinn Rauschen vom Signal zu eliminieren und eine Verzögerung von $-D$ in Kauf zu nehmen. Für jedes Sample erzeugt der Filter eine Schätzung auf der Basis von $M + 1$ Sampels von $y[n]$.
- *Filtering*: Bei $D = 0$ ist die Absicht das Signal $s[n]$ zu regenerieren. Es werden keine Verzögerungen erzeugt, da die Abschätzung aufgrund des aktuellen und vergangenen Werten gemacht wird.
- *Prediction*: Bei $D > 0$ soll das Signal vorhergesagt werden.

Filterausgang:

$$\hat{s}[n + D] = \sum_{i=0}^M w_i \cdot y[n - i]$$

Schätz-Fehler:

$$\hat{s}[n + D] - s[n + D]$$

Um die optimalen Parameter w_i zu finden muss der Gradient auf Null gesetzt werden:

$$\sum_{i=0}^M \tilde{w}_i \cdot \gamma_{yy}[m - i] = \gamma_{sy}[m + D], \quad m = 0, 1, \dots, M$$

mit der Autokorrelation des Filter-Eingangs, rechts der Fall für das obige Modell:

$$\gamma_{yy} = E\{y[n] \cdot y^*[n - m]\} \quad \gamma_{yy}[m] = \gamma_{ss}[m] + \gamma_{vv}[m]$$

und der Kreuzkorrelation, rechts der Fall für das obige Modell:

$$\gamma_{sy} = E\{s[n] \cdot y^*[n - m]\} \quad \gamma_{sy}[m] = \gamma_{ss}[m]$$

Wiener-Hopf Gleichung:

$$\mathbf{R}_{yy} = \begin{bmatrix} \gamma_{yy}[0] & \gamma_{yy}[-1] & \dots & \gamma_{yy}[-M] \\ \gamma_{yy}[1] & \gamma_{yy}[0] & \dots & \gamma_{yy}[1 - M] \\ \vdots & \vdots & & \vdots \\ \gamma_{yy}[M] & \gamma_{yy}[M - 1] & \dots & \gamma_{yy}[0] \end{bmatrix} \quad \mathbf{r}_{sy} = \begin{bmatrix} \gamma_{sy}[D] \\ \gamma_{sy}[D + 1] \\ \vdots \\ \gamma_{sy}[D + M] \end{bmatrix} \quad \tilde{\mathbf{w}} = \begin{bmatrix} \tilde{w}_0 \\ \tilde{w}_1 \\ \vdots \\ \tilde{w}_M \end{bmatrix}$$

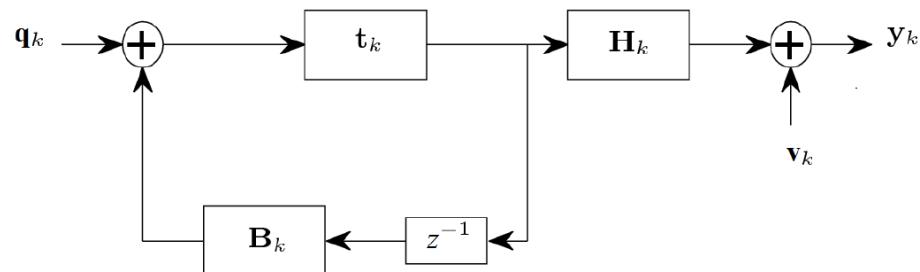
Der Vektor mit den optimalen Filterkoeffizienten $\tilde{\mathbf{w}}$ ist gegeben durch:

$$\tilde{\mathbf{w}} = \mathbf{R}_{yy}^{-1} \cdot \mathbf{r}_{sy}$$

Anmerkung: Die Invertierung der Matrix R_{yy} kann sich für hohe Ordnung M als problematisch erweisen. Mittels Toeplitz-Matrix lassen sich die Werte von \tilde{w} ohne die Invertierung berechnen.

8.2 Kalman-Filter

Jede zusätzliche Beobachtung wird für die Verbesserung der Filtercharakteristik verwendet, ohne dass die Ordnung limitierend wirkt. Basiert auf der Zustandsraumbeschreibung und eignet sich damit für dynamische Systeme.



Von aussen sichtbar sind die Messwerte $\mathbf{y}_1, \mathbf{y}_2, \dots$. Für die k -te Beobachtung gilt:

$$\mathbf{y}_k = \mathbf{H}_k \cdot \mathbf{t}_k + \mathbf{v}_k$$

Wobei \mathbf{H}_k die *Messmatrix* und \mathbf{v}_k den Messfehler abbildet.

Für die internen Zustandsvariablen zum Zeitpunkt k gilt:

$$\mathbf{t}_k = \mathbf{B}_k \cdot \mathbf{t}_{k-1} + \mathbf{q}_k$$

Wobei \mathbf{B}_k die *Zustandsübergangsmatrix* und \mathbf{q}_k den zufälliger Vektor ist.

Übersicht über die Filter-Bestandteile:

entity	symbol	deterministic/random
state vector	\mathbf{t}_k	random
covariance matrix of \mathbf{t}_k	\mathbf{R}_k	deterministic
observation	\mathbf{y}_k	random
measurement matrix	\mathbf{H}_k	deterministic
measurement error	\mathbf{v}_k	random with mean $\mathbf{0}$, covariance \mathbf{W}_k
state transition matrix	\mathbf{B}_k	deterministic
input	\mathbf{q}_k	random with mean \mathbf{u}_k , covariance \mathbf{P}_k

Vergleich zum Wiener-Filter:

Beide Filter sind optimale lineare Filter bezüglich dem mean-squared error. Der Wiener-Filter hat eine endliche Ordnung M , der Kalman-Filter hingegen ist rekursiv aufgebaut und hat keine endliche Ordnung. Der Kalman-Filter eignet sich im Gegensatz zum Wiener-Filter für dynamische Systeme statt nur für stationäre Systeme.

Kapitel 9

Adaptive Filter

Der Wiener- und Kalman-Filter sind optimale Filter unter der Annahmen, dass die Statistik über des involvierten Zufallsprozesses komplett bekannt sind. In der Praxis ist die üblicherweise nicht der Fall. Adaptive Filter passen sich unbekanntem und sich ändernder Umgebung automatisch an.

9.1 Linear Predictive Coding (LPC-10e)

Statt die Wellenform zu codieren, bildet der Vocoder ein Modell und extrahiert die Parameter davon.

Das menschliche Sprachsignal wird mit einem all-pole Filter nach dem AR-Modell mit Ordnung P abgebildet:

$$H(z) = \frac{g}{1 - \sum_{k=1}^P a_k \cdot z^{-k}}$$

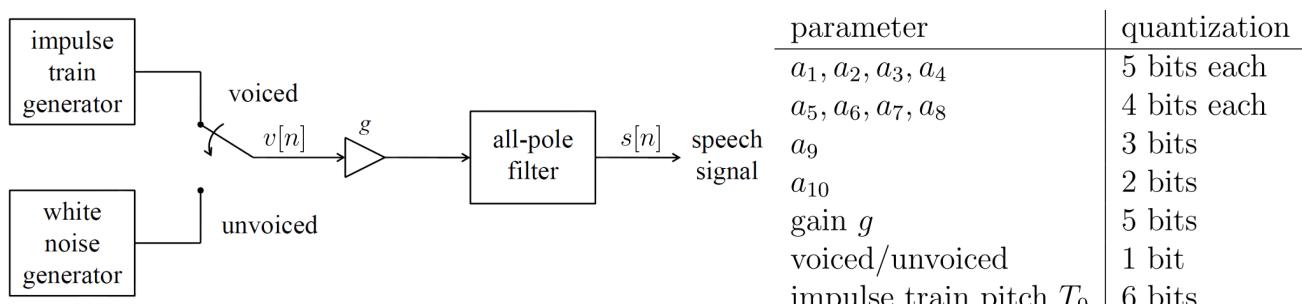
Die Anregung erfolgt mit einem Impulskamm oder einem Rauschprozess. LPC-10e Algorithmus:

- 8000 Samples/s
- 180 Samples/Segment
- Ordnung $P = 10$
- 54 Bits pro Segment für Encoding
- Bitrate von 2400 Bit/s

Entscheidet für jedes Segment, ob es voiced oder unvoiced ist, berechnet die Filterkoeffizienten a_1, a_2, \dots, a_{10} und bestimmt das Gain g . Konsonanten zählen als unvoiced, da das Signal einem Rauschen gleicht. Vokale erzeugen eine periodische Schwingung mit T_0 , welche pro Segment bestimmt wird.

Differenzengleichung des unten ersichtlichen Systems:

$$s[n] = \sum_{k=1}^P (a_k \cdot s[n-1] + g \cdot v[n])$$



Die Parameter a_k müssen so gefunden werden, dass der "one step linear predictor"

$$\hat{s}[n] = \sum_{k=1}^P a_k \cdot s_{in}[n-k]$$

möglichst nahe an $s_{in}[n]$ kommt. Dazu kann die Wiener-Filter-Theorie angewendet werden, es ergibt sich also:

$$\mathbf{R}_{ss} = \begin{bmatrix} \gamma_{ss}[0] & \gamma_{ss}[1] & \dots & \gamma_{ss}[P-1] \\ \gamma_{ss}[1] & \gamma_{ss}[0] & \dots & \gamma_{ss}[P-2] \\ \vdots & \vdots & & \vdots \\ \gamma_{ss}[P-1] & \gamma_{ss}[P-2] & \dots & \gamma_{ss}[0] \end{bmatrix}, \quad \mathbf{r}_{ss} = \begin{bmatrix} \gamma_{ss}[1] \\ \gamma_{ss}[2] \\ \vdots \\ \gamma_{ss}[P] \end{bmatrix} \quad \text{und} \quad \mathbf{a} = \mathbf{R}_{ss}^{-1} \cdot \mathbf{r}_{ss}$$

Es wird also ein Predictor mit $D = 1$ angewendet, $\gamma_{ss}[m]$ ist die Autokorrelation von $s_{in}[n]$. Die Werte der Autokorrelation müssten angenähert werden durch:

$$\hat{\gamma}_{ss}[m] = \frac{1}{N-m} \sum_{n=1}^{N-m} s_{in}[n] \cdot s_{in}[n+m], \quad \text{biased Variante mit: } \frac{1}{N}$$

mit N für die Anzahl Werte im Segment, beim LPC-10e also $N = 180$.

Der Fehler beträgt und ist abhängig von der Verstärkung:

$$d[n] = s_{in}[n] - \hat{s}[n]$$

Die Verstärkung ergibt sich nach:

$$g = \sqrt{\frac{1}{N} \sum_{n=1}^N d^2[n]}$$

Zur Unterscheidung zwischen "voiced" und "unvoiced" sowie für den Pitch wird die durchschnittliche Amplituden-Differenz-Funktion (average magnitude difference function) verwendet, definiert als:

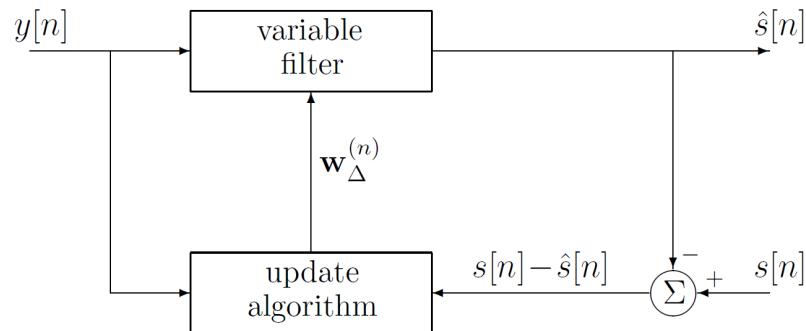
$$\bar{\gamma}_d[m] = \frac{1}{N-m} \sum_{n=1}^{N-m} \left| \frac{d[n]}{g} - \frac{d[n+m]}{g} \right|$$

Wenn das Minimum von $\bar{\gamma}_d[m]$ unter einen Schwellwert fällt, wird das Segment als "voiced" deklariert:

$$T_0 = \arg \min_{m \in [20, 160]} \bar{\gamma}_d[m]$$

9.2 LMS Algorithmus (least mean square)

Passt seine Filterkoeffizienten sample-für-sample basierend auf dem Fehler $s[n] - \hat{s}[n]$ an. Eignet sich beispielsweise für die Echo-Cancellation in der Telefonie. Beim variablen Filter handelt es sich oft um einen FIR-Filter.



Das Filter schaltet von einem Training-Modus, in dem die Preamble einer Nachricht analysiert wird, in den Entscheidungsmodus um, in dem die Nutzdaten ausgewertet werden.

Es wird der Eingangvektor und die Filterkoeffizienten benötigt:

$$\mathbf{y}_n = \begin{bmatrix} y[n] \\ y[n-1] \\ \vdots \\ y[n-M] \end{bmatrix} \quad \mathbf{w} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_M \end{bmatrix}$$

Der Ausgang des variablen FIR-Filters wird berechnet mit:

$$\hat{s}[n] = \mathbf{w}^T \cdot \mathbf{y}_n$$

Der mittlere quadratische Fehler ist:

$$\varepsilon_{MSE}(\mathbf{w}) = E\{(\hat{s}[n] - s[n])^2\}$$

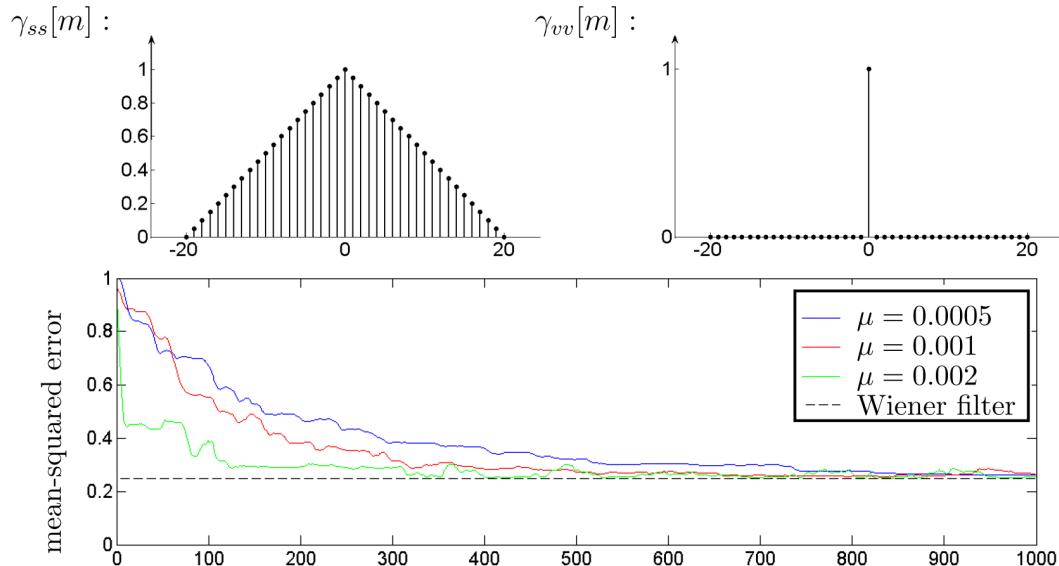
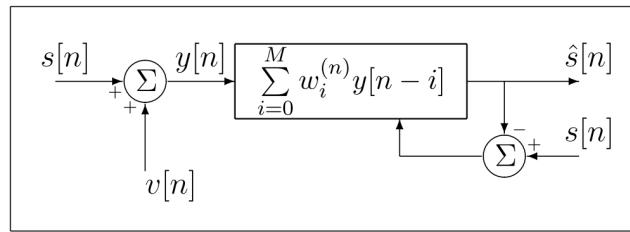
Der Vektor \mathbf{w} wird zu Beginn mit $M + 1$ Nullen initialisiert und berechnet sich danach aus:

$$\mathbf{w}^{(n+1)} = \mathbf{w}^{(n)} + \underbrace{\mu \left(s[n] - \left(\mathbf{w}^{(n)} \right)^T \mathbf{y}_n \right) \cdot \mathbf{y}_n}_{=\mathbf{w}_\Delta^{(n)}}$$

μ ist die Schrittweite und. Ein zu kleiner Wert resultiert in einer langsamen Anpassung, ein zu grosser Wert kann zu Schwingungen um das Optimum herum führen.

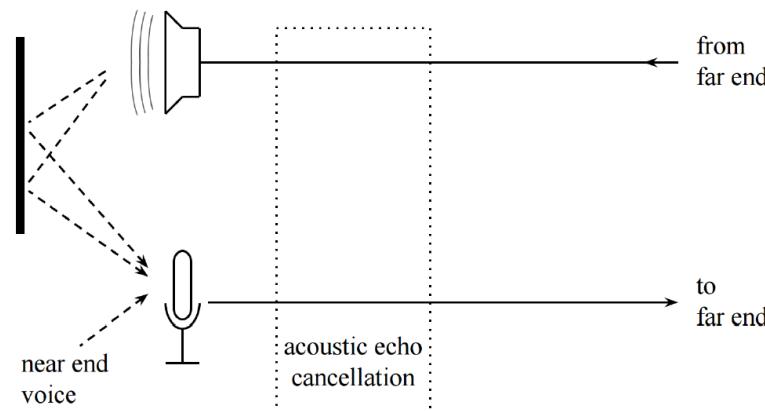
Im folgenden sind die Verläufe des Fehlers bei verschiedenen Schrittweiten μ aufgezeichnet. Es soll ein mit weissem Rauschen $v[n]$ überlagertes Signal $s[n]$ gefiltert werden. Das Filter hat Ordnung $M = 12$ und startet mit \mathbf{w} als Nullvektor. Mit steigender Anzahl Samples verringert sich der Fehler und kommt immer näher an den optimalen Filter (Wiener-Filter) heran.

Bessere Resultate (schnellere Konvergenz) wären mit einem RLS-Filter erzielbar. Dieser weisst jedoch eine deutlich höhere Komplexität auf als der LMS-Filter.

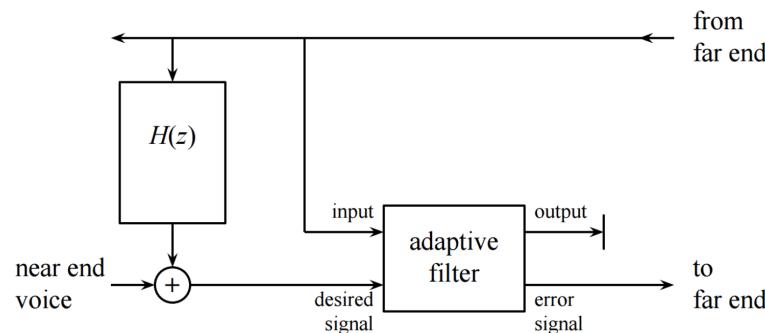


9.3 Echo cancellation

Konferenztelefone müssen das Echo, welches durch das vom Lautsprecher ausgestrahlte Audiosignal vom “anderen Ende der Leitung“ entsteht und erneut vom Mikrofon aufgenommen wird, herausfiltern können. Für diese Aufgabe wird typischerweise ein adaptives Filter eingesetzt.



$H(z)$ repräsentiert den Pfad zwischen Lautsprecher und Mikrofon inklusive allen möglichen Pfaden, welche die Schallwellen nehmen können. Es kann mit einem LTI-System modelliert werden. $H(z)$ versucht, das verzerrte Signal nachzubilden, welches vom Mikrofon wieder aufgenommen wird.



Das obige Modell wird erweitert, in dem unterschieden wird, ob eine Person spricht (am jeweiligen Ende), oder ob beide Personen sprechen. Ohne Unterscheidung verschlechtert sich durch das Filter die Qualität des Audiosignals. Einige Systeme fügen “comfort noise“ hinzu. So bleibt die Gegenseite nicht komplett still, was für den Menschen ungewohnt ist.

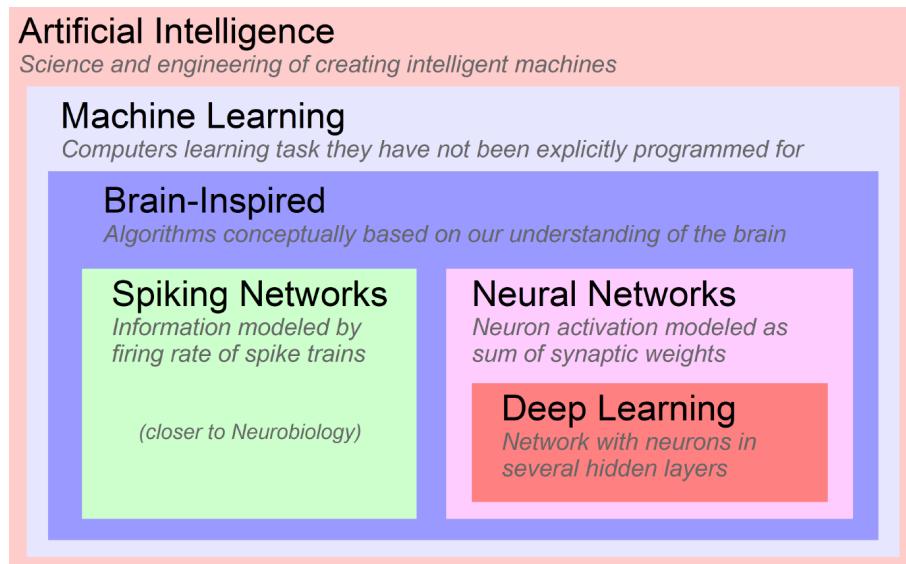
Kapitel 10

Neuronale Netze

10.1 Einführung

Deep Learning ist eine Möglichkeit, Machine Learning Aufgaben umzusetzen. Folgende Varianten existieren:

- Unsupervised: Finden von Mustern in Daten
- Supervised: Für einen neuen Input den Ausgang vorhersagen, basierend auf Beispielsdaten (Input/Output)
 - Regression: Bei der Ausgangsgröße handelt es sich um echte Werte, z.B. Temperatur am nächsten Tag.
 - Classification: Die Ausgangsgrößen werden mit Klassen gelabelt.



10.1.1 Anwendungsgebiete

- Objektklassifizierung/Detektion/Segmentierung
- Autonomes Fahren (Fahrzeuge)
- Geräuschunterdrückung in Echtzeit
- Automatisches Einfärben von schwarz/weiss Photos

10.1.2 Gründe für Einsatz & Popularität

- Deep Learning liefert für spezifische Aufgaben bessere Resultate als der Mensch
- Probleme, die mathematisch nur schwer modelliert werden können, werden lösbar
- Vortrainierte Netzwerke sind verfügbar (transfer learning)
- Rechenleistung ist mittlerweile durch GPUs vorhanden

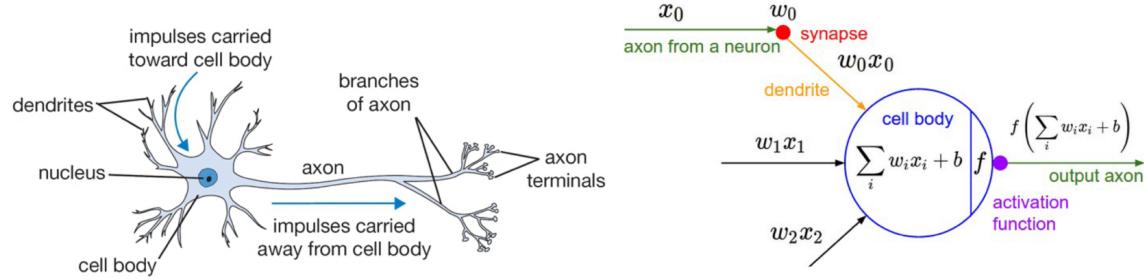
10.1.3 Typen-Übersicht

Weit verbreitete Strukturen sind das *Convolutional Neural Network* (CNN) sowie das *Deep Convolutional Network* (DCN), welches zu der Klasse der “feedforward Netzwerke” gehört. Im Bereich der Spracherkennung und dem zeitlichen Verlauf von Daten sind *Recurrent Neural Networks* (RNN), welche einen Feedback-Loop beinhalten.

CNN und RNN können mit FIR- und IIR-Filttern verglichen werden.

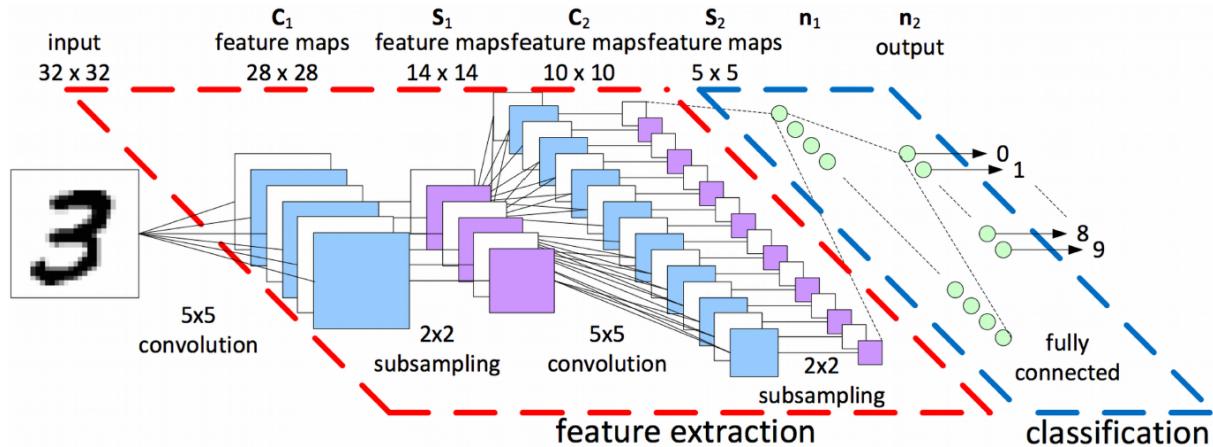
10.1.4 Vergleich: Nature vs. AI Networks

- Das Menschliche Gehirn beinhaltet $\approx 8.6 \cdot 10^{10}$ Neuronen.
- Jedes Neuron ist durchschnittlich mit 7000 Synapsen mit anderen Neuronen verbunden.
- Die "synaptic weights" sind lernbar und beeinflussen die Funktionalität des Gehirns.
- AI Netzwerke modellieren den Ausgang von Neuronen als gewichtete Summe von Eingängen und einer nicht-linearen Funktion. Diese nicht lineare Funktion wird "Activation function" genannt, der Ausgang eines Neurons einfach nur "Activation".



10.2 Prinzipielle Architektur für Bildklassifikation

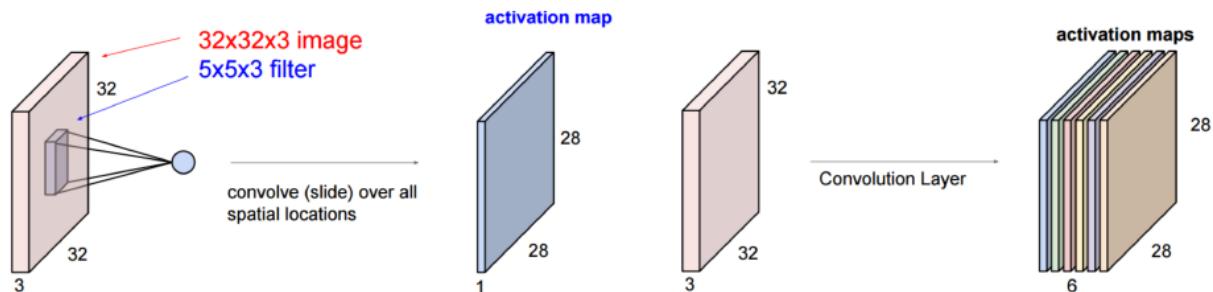
- Die Dimensionen eines Layers entsprechen der Anzahl Neuronen (Aktivierungen) am Layer-Ausgang.
- Die Anzahl Dimensionen des Eingangs-Layers entsprechen der Anzahl Pixel am Eingangsbild.
- Nach dem Eingangs-Layer führen mehrere Convolutional/Subsampling Layer die "Feature Extraction" durch.
- Aufeinanderfolgende "fully connected layers" führen die Klassifizierung durch.
- Die Anzahl Neuronen im Output Layer entspricht der Anzahl Klassen (Labels).



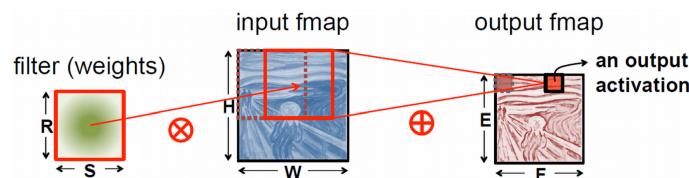
10.3 Layer-Übersicht

10.3.1 Convolutional Layer

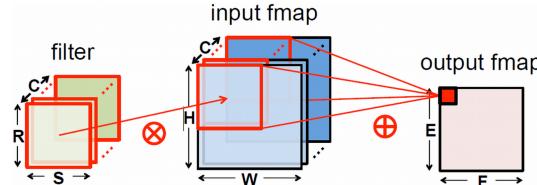
Sind die fundamentalen Bauteile eines aktuellen DL neuronalen Netzwerkes. Eine Filtermaske wird über das Bild gelegt und mit dem Input gefaltet (sliding window Verfahren).



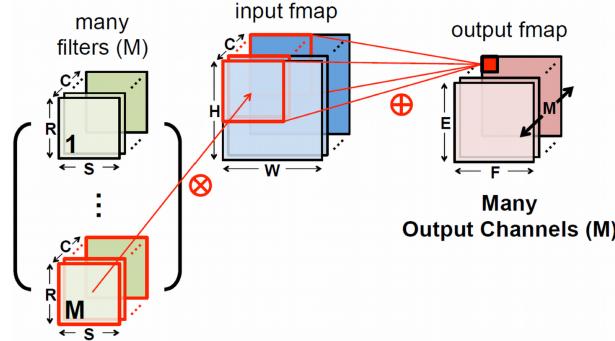
In jedem Schritt über die input feature map werden die Filterkoeffizienten mit dem unterliegenden Input elementweise multipliziert. Die Schrittgrösse nennt man "stride" und bestimmt die Ausgangs feature map, Grösse ($E \times F$). Falls $R = S$ gilt, wird typischerweise eine Filtermaske mit der Grösse 1x1, 3x3, ... gewählt.



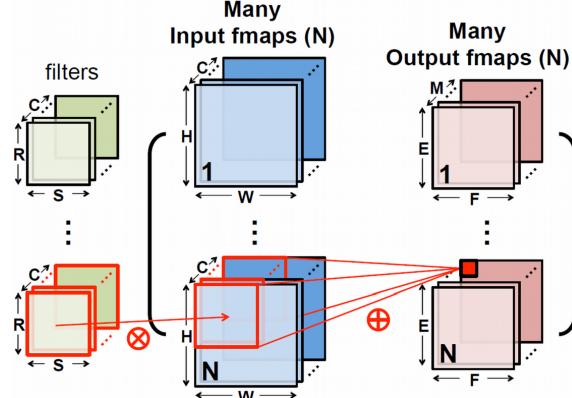
Falls mehr als eine input feature map existiert, werden alle feature maps auf einmal mit einem 3D-Filterkern berechnet. In jedem Convolutional Layer entspricht die Anzahl der input feature maps demnach der Anzahl der Filter Channels C . Typischerweise ist $1 \leq C \leq 128$. Nur am ersten Convolutional Layer ist typischerweise $C = 1$, gegen Ende kann sie grösser werden.



Oft wird mehr als ein 3D-Filter pro Convolutional Layer angewendet. Wenn im k -ten Convolutional Layer M Filter angewendet werden, dann entstehen aus den resultierenden M output feature maps von Layer k schlussendlich C input feature maps in Layer $k + 1$.



Falls $N > 1$, spricht man von "batch processing", es wird also mehr als ein Bild auf einmal durch das Netzwerk verarbeitet.



Die unten ersichtlichen Formeln gehen davon aus, dass kein Padding am Eingang durchgeführt wird.

$$\mathbf{O}[n][m][x][y] = \text{Activation} \left(\mathbf{B}[m] + \sum_{i=0}^{R-1} \sum_{j=0}^{S-1} \sum_{k=0}^{C-1} \mathbf{I}[n][k][Ux+i][Uy+j] \times \mathbf{W}[m][k][i][j] \right),$$

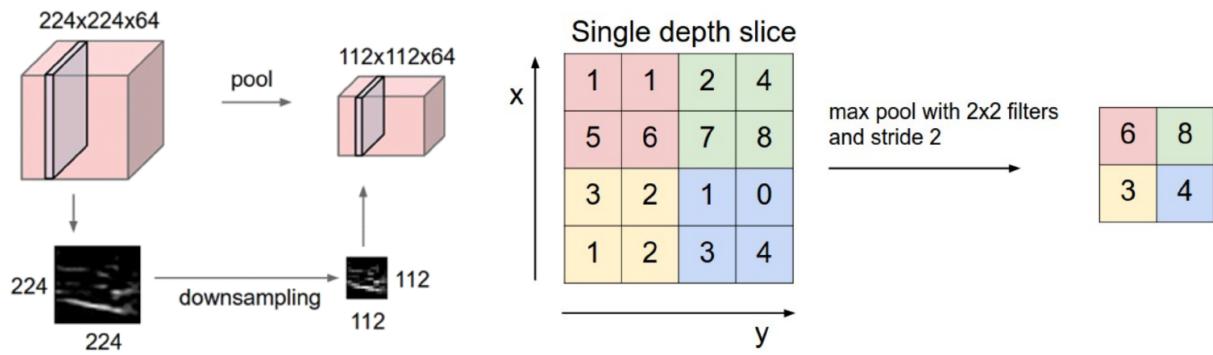
$$0 \leq n < N, 0 \leq m < M, 0 \leq y < E, 0 \leq x < F,$$

$$E = (H - R + U)/U, F = (W - S + U)/U$$

Shape Parameter	Description
N	fmap batch size
M	# of filters / # of output fmap channels
C	# of input fmap/filter channels
H/W	input fmap height/width
R/S	filter height/width
E/F	output fmap height/width
U	convolution stride

10.3.2 Pooling Layer

Werden zwischen Convolutional Layers platziert. Beim Pooling wird die Tiefe beibehalten, es findet aber ein Down-sampling für Breite & Höhe statt. Der zugehörige Faktor nennt sich "stride". Das Pooling reduziert die Anzahl Gewichte welche trainiert werden müssen und somit den Rechenaufwand.



10.3.3 Activation Layer

Repräsentiert eine nicht-lineare Operation in der Übertragungsfunktion eines Neurons. Es existieren verschiedene Funktionen, die ReLU-Funktion ist am weitesten verbreitet.

Binär-Schritt		$f(x) = \begin{cases} 0 & \text{für } x < 0 \\ 1 & \text{für } x \geq 0 \end{cases}$	$\{0,1\}$
ReLU		$f(x) = \begin{cases} 0 & \text{für } x < 0 \\ x & \text{für } x \geq 0 \end{cases}$	$(0, \infty)$
Sigmoid		$f(x) = \frac{1}{1 + e^{-x}}$	$(0,1)$

10.4 Layer-Entwicklung

Begriffserläuterung:

- Padding: Ergänzen der Daten am Rand, damit die ganze Filtermaske im Bild Platz findet.

Filtergrößen: Die Filtergrößen sind entscheidend für die Qualität, Speicherbedarf sowie Rechendauer.

- Pooling jeweils mit (2×2) Filter (stride = 2), Fully Connect mit 10 Ausgangsklassen
- Layer 2 (Conv 1) enthält 8 (3×3) Filter
- Layer 6 (Conv 2) enthält 16 (3×3) Filter
- Layer 10 (Conv 3) enthält 32 (3×3) Filter

Layer	L1 Input Image	L2 Convolution 1	L5 Pooling 1	L6 Convolution 2	L9 Pooling 2	L10 Convolution 3	L13 Fully Connected
Neuron structure (Tensor dimension)	$28 \times 28 \times 1$ Pixel 	$28 \times 28 \times 8$ Pixel 	$14 \times 14 \times 8$ Pixel 	$14 \times 14 \times 16$ Pixel 	$7 \times 7 \times 16$ Pixel 	$7 \times 7 \times 32$ Pixel 	Each output (10) is connected to each pixel
# of MAC (multiply accumulate) $\approx 550'000$	-	$8 \times (28 \times 28) \times (3 \times 3)$ $56'448$	-	$16 \times (14 \times 14) \times (3 \times 3 \times 8)$ $225'792$ $4 \times \text{MAC}_{L2}$	-	$32 \times (7 \times 7) \times (3 \times 3 \times 16)$ $225'792$ $4 \times \text{MAC}_{L2}$	$10 \times (7 \times 7 \times 32)$ $15'680$ $\approx \text{MAC}_{L2} / 4$
# of weights (trainig) $\approx 20'000$	-	$8 \times (3 \times 3)$ 72	-	$16 \times (3 \times 3 \times 8)$ $1'152$ $16 \times W_{L2}$	-	$32 \times (3 \times 3 \times 16)$ $1'152$ $64 \times W_{L2}$	$10 \times (7 \times 7 \times 32)$ $15'680$

Fazit: Faltung benötigt zur Laufzeit viel Rechenzeit, Training generiert die Gewichtungen und benötigt deshalb Speicher. Der Zugriff auf die extern gespeicherten Gewichtungen ist dabei Energieintensiv.

Kapitel 11

Matlab

Matrizen generierung

Befehl	Beschreibung	Syntax
<code>zeros</code>	Matrix mit Nullen	<code>zeros(N),</code> <code>zeros(M,N)</code>
<code>ones</code>	Matrix mit Einsen	<code>ones(N),</code> <code>ones(M,N)</code>
<code>rand</code>	Matrix mit gleichverteilten Zufallszahlen	<code>rand(N),</code> <code>rand(M,N)</code>
<code>randn</code>	Matrix mit normalverteilten Zufallszahlen	<code>randn(N),</code> <code>randn(M,N)</code>
<code>linspace</code>	Matrix mit linear steigenden Werten	<code>linspace(N),</code> <code>linspace(M,N)</code>
<code>logspace</code>	Matrix mit logarithmisch steigenden Werten	<code>logspace(N),</code> <code>logspace(M,N)</code>

Elementar Funktionen

Befehl	Beschreibung	Syntax
<code>log</code>	natürlicher Logarithmus	
<code>log10/2</code>	Logarithmus zur Basis 10 bzw. 2	
<code>round</code>	Rundung zur nächsten ganzen Zahl	
<code>ceil</code>	Rundung Richtung ∞	<code>ceil(n.x)</code> $= n + 1$
<code>floor</code>	Rundung Richtung $-\infty$	<code>floor(n.x)</code> $= n$
<code>mean</code>	Arithmetische Mittel	

Komplexe Zahlen

Befehl	Beschreibung	Syntax
<code>abs</code>	Betrag	
<code>angle</code>	Phase	
<code>real</code>	Realanteil	
<code>imag</code>	Imaginäranteil	
<code>conj</code>	conjungiert complex	
<code>polar</code>	Polarkoordinaten-Darstellung	

Graphik

Befehl	Beschreibung	Syntax
<code>figure</code>	Grafik-Fenster	
<code>plot</code>	2-D Grafik	<code>plot(X1,Y1,S1,X2,Y2,S2,...)</code>
<code>subplot</code>	Unterteilung des Grafik-Fensters	<code>subplot(z,s,p)</code>
<code>stem</code>	Darstellung diskreter Werte	
<code>hold</code>	Halten der Grafik zum Übereinanderzeichnen	

Transformation und Filterung

Befehl	Beschreibung	Syntax
<code>dftmtx</code>	Transformationsmatrix für DFT	<code>dftmtx(N)</code>
<code>fft/fft2</code>	schnelle DFT (FFT), 1-/2-dimensional	
<code>ifft/ifft2</code>	inverse FFT, 1-/2-dimensional	
<code>fftshift</code>	Verschiebung des Gleichanteils in die Mitte des Spektrums	
<code>spectrum</code>	Leistungsdichte-Spektrum	
<code>filter</code>	Filterung	<code>filter(B,A,X)</code> $(H(z) = B(z)/A(z))$
<code>conv</code>	Faltung	
<code>fftfilt</code>	schnelle Faltung	
<code>freqz</code>	Frequenzgang	<code>freqz(B,A,N)</code>
<code>impz</code>	Impulsantwort	<code>impz(B,A,N)</code>
<code>zplane</code>	Pol/Nullstellenverteilung	<code>zplane(B,A)</code>

Filterentwurf

Befehl	Beschreibung	Syntax
<code>fir1</code>	Hoch-, Tief-, Bandpass, Bandsperre	<code>fir1(N,Wn,'low')</code> <code>fir1(N,Wn,'high')</code> <code>Wn=[W1,W2]</code> <code>fir1(N,Wn,'bandpass')</code> <code>fir1(N,Wn,'stop')</code>
<code>fir2</code>	allgemeiner Frequenzverlauf	
<code>butter</code>	Butterworth IIR	<code>[B,A]=butter(N,Wn,'low')</code>
<code>cheby1</code>	Chebyshev I IIR	
<code>cheby2</code>	Chebyshev II IIR	
<code>ellip</code>	Elliptisch Cauer	

Fensterung

Befehl	Beschreibung	Syntax
<code>rectwin</code>	Rechteckfenster	<code>rectwin(N)</code>
<code>bartlett</code>	Dreieck-Fenster	<code>bartlett(N)</code>
<code>hamming</code>	Hamming-Fenster	<code>hamming(N)</code>
<code>kaiser</code>	Kaiser-Fenster	<code>kaiser(N)</code>