

Digitale Signalverarbeitung auf FPGAs

Kap. 2 – Zeitdiskrete Signale und Systeme
im Frequenzbereich

2016

Dr. Christian Münker

- $H(z)$, $|H(z)|$, $H(e^{j\Omega})$ und $H(\Omega)$...
- Schreibformen von $H(z)$
- Pol / Nullstellendiagramm
- Kausalität
- Matlab / Python – Darstellung von $H(z)$
- Einfache Filter

Digitale Signalverarbeitung auf FPGAs

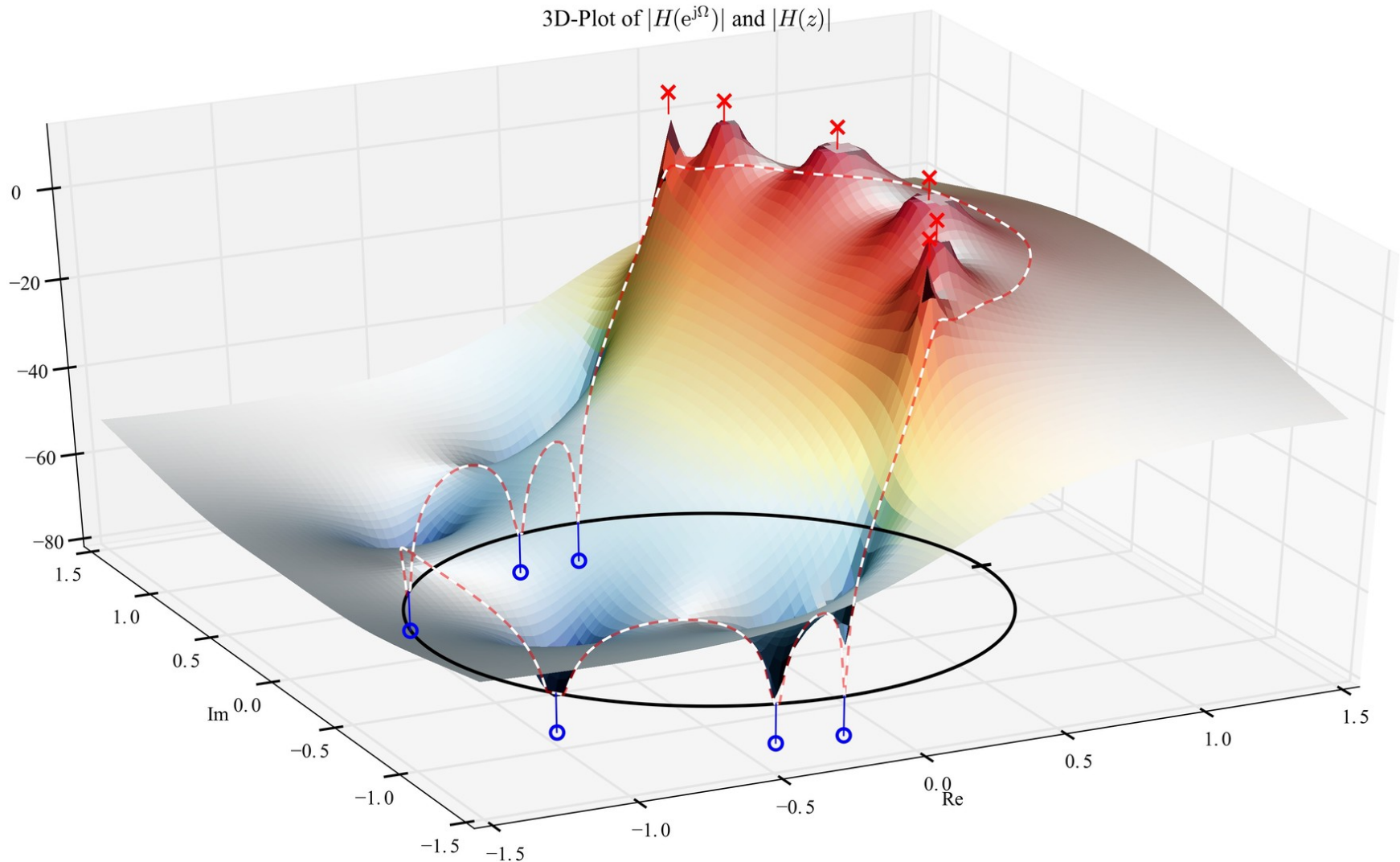
Kap. 2 Zeitdiskrete Signale und LTI-Systeme im Frequenzbereich

Teil 1 $H(z)$, $|H(z)|$, $H(e^{j\Omega})$ und $H(\Omega)$...

2016

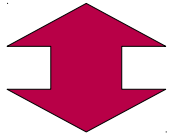
Dr. Christian Münker

$H(z)$, $|H(z)|$, $H(e^{j\Omega})$ und $H(\Omega)$...

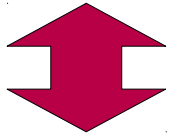


Zeitdiskrete Signale und LTI-Systeme lassen sich auf verschiedene Arten darstellen (vgl. zeitkontinuierliche Signale und LTI-Systeme):

Zeitbereich: Folge + Impulsantwort $h[k]$ → **Kapitel 1**



Komplexe z-Ebene: Systemfunktion $H(z)$, $|H(z)|$ → **Kapitel 1 + 2**



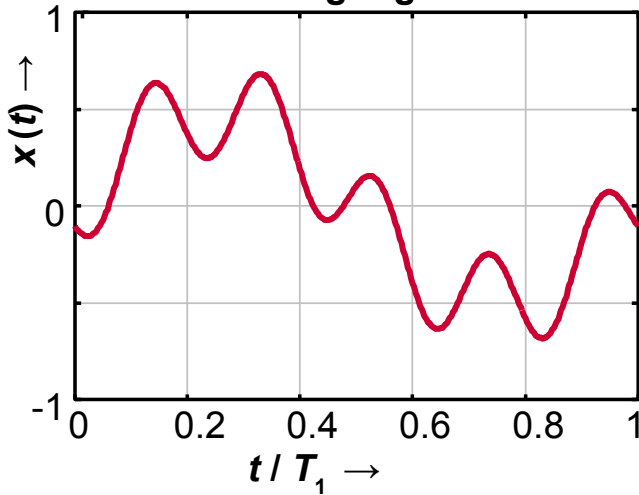
Frequenzbereich: Komplexer Frequenzgang $\underline{H}(e^{j\Omega})$ → **Kapitel 2**

Warum $\underline{H}(e^{j\Omega})$ und nicht $\underline{H}(\Omega)$?

Spektrum abgetasteter Signale

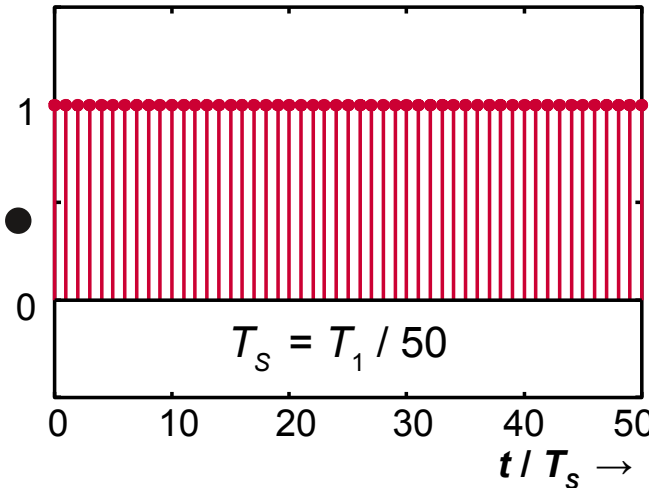


Analog Signal

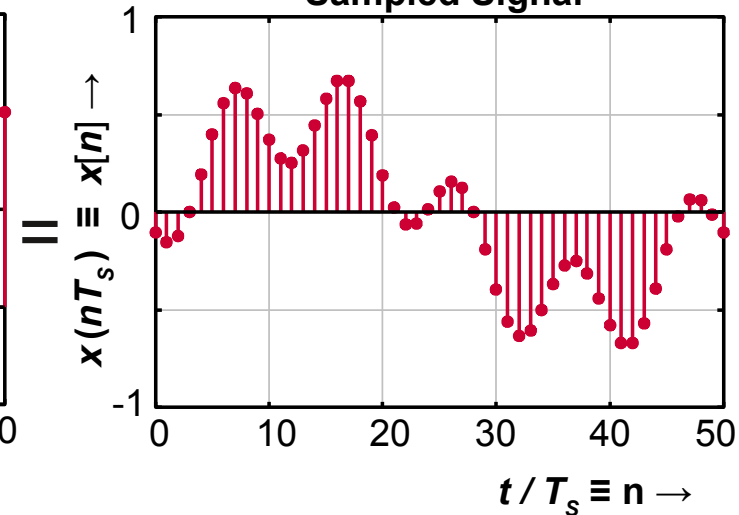


$$\sin\left(\frac{2\pi t}{T_1}\right)/2 + \cos\left(\frac{10\pi t}{T_1} + \varphi_0\right)/4$$

Ideal Sampler

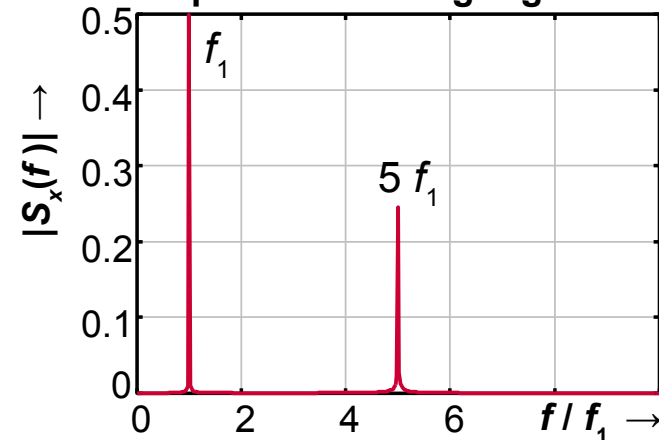


Sampled Signal

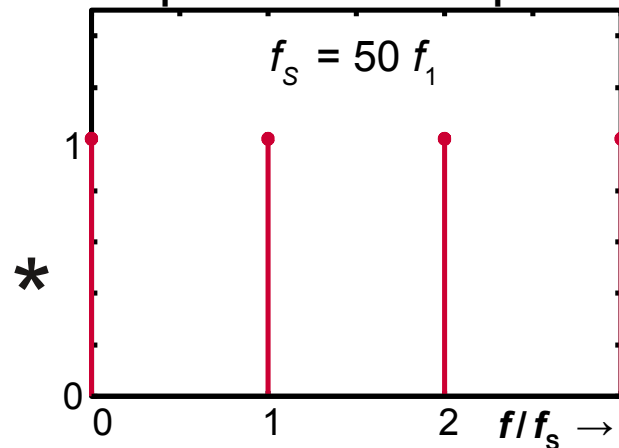


Periodisches Spektrum !

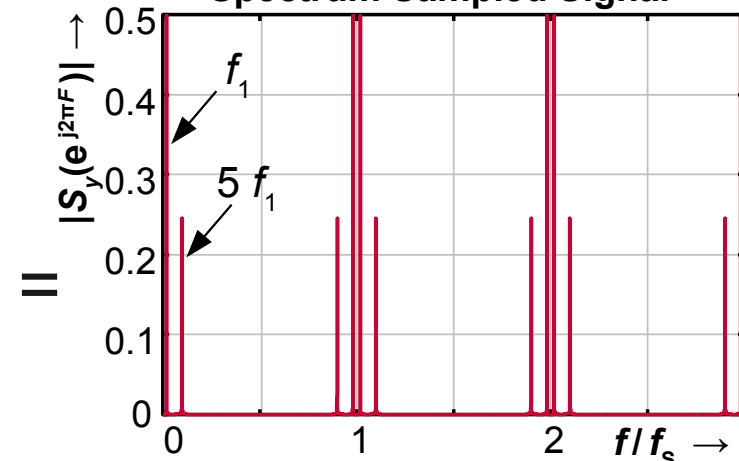
Spectrum Analog Signal



Spectrum Ideal Sampler



Spectrum Sampled Signal



Beispiel: $f = 1$ kHz (Perioden / s), $f_s = 20$ kHz (Samples / s)

Winkelfrequenz $\omega = 2\pi f = 6,28$ k rad / s

Normierte Frequenz $F = f / f_s = 0,05$ Perioden / Sample

Norm. Winkelfrequenz $\Omega = 2\pi f / f_s = 0,314$ rad / Sample
 $= 0,1 \pi$ rad / Sample

Achtung!

- Matlab bezieht normierte Frequenzen auf die Nyquistfrequenz $f_s/2$!
- Ω und ω werden in der Literatur nicht einheitlich verwendet!

Frequenzgang in der z-Ebene



Beliebiger Punkt ★ in z-Ebene:

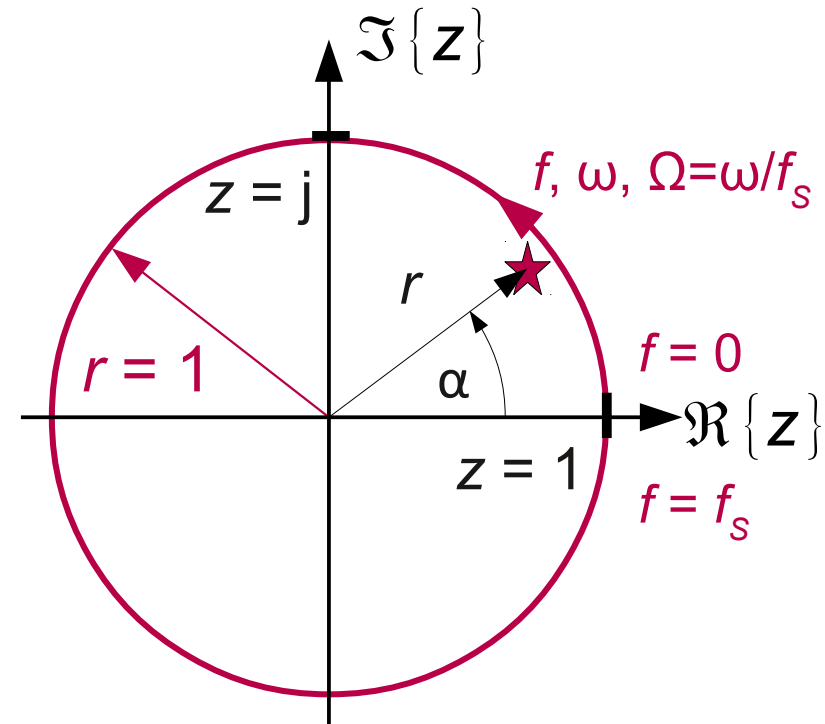
$$\begin{aligned} z &= e^{sT_s} = e^{(\sigma + j\omega)T_s} = e^{\sigma T_s} e^{j\omega T_s} = r e^{j\alpha} \\ &= x + jy = r(\cos \alpha + j \sin \alpha) \end{aligned}$$

Für $\sigma = 0$, $r = 1$ Frequenzachse entlang des Einheitskreises (EK):

$$\Rightarrow z = e^{j\omega T_s} = e^{j2\pi f/f_s}$$

$$\Rightarrow H(f) = H(z = e^{j2\pi f/f_s} = e^{j\omega/f_s})$$

$$z = e^{j2\pi f/f_s} = e^{j2\pi(f + kf_s)/f_s} \Rightarrow H(f) = H(f + kf_s); \quad k = \dots, -1, 0, 1, \dots$$



Zeitdiskrete Spektren bzw. Frequenzgänge werden entlang des Einheitskreises abgelesen und sind periodisch mit f_s !

Schreibweise: $H(z = e^{j2\pi f T_s}) = H(e^{j2\pi F}) = H(e^{j\Omega})$

Normierte Frequenzen auf dem EK



$$r = 1 \Rightarrow z = e^{j\omega T_s} = e^{j\Omega} = e^{j2\pi f/f_s} = e^{j2\pi F}$$

Normierte Frequenz:

$$F = f / f_s = 0 \dots 1 \text{ (oder } \pm 0,5)$$

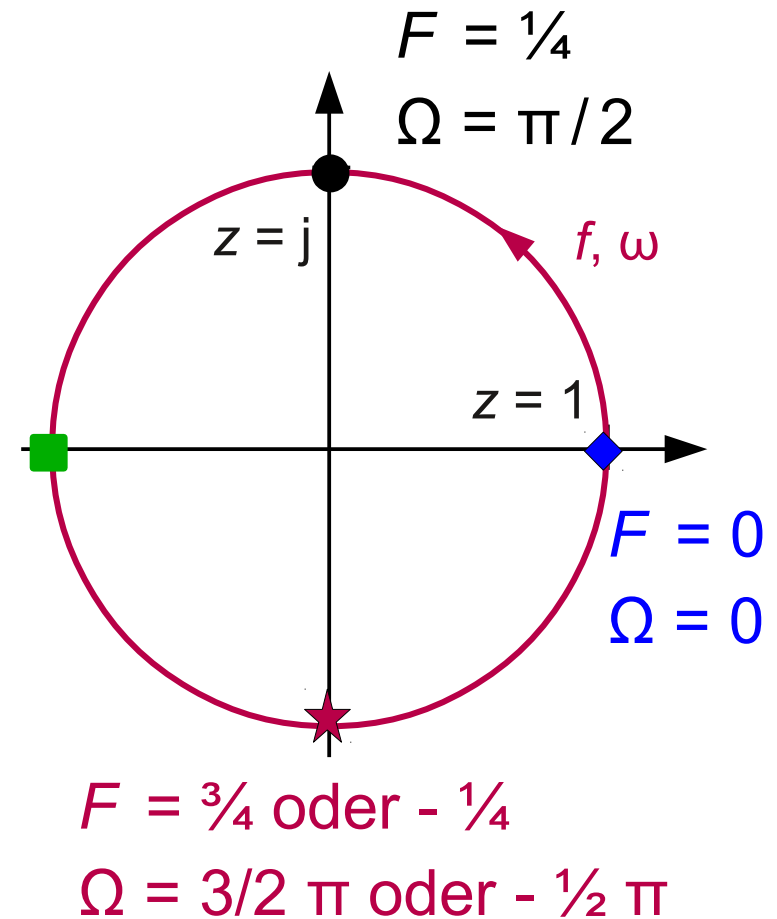
Normierte Kreisfrequenz:

$$\Omega = \omega / f_s = 0 \dots 2\pi \text{ (oder } \pm \pi)$$

Wichtige Punkte auf dem EK:

- ◆ $z = 1 \rightarrow f = 0 \text{ oder } f_s$
- $z = j \rightarrow f = f_s / 4$
- $z = -1 \rightarrow f = f_s / 2$
- ★ $z = -j \rightarrow f = 3/4 f_s \text{ oder } -f_s / 4$

$$F = 1/2$$
$$\Omega = \pi$$

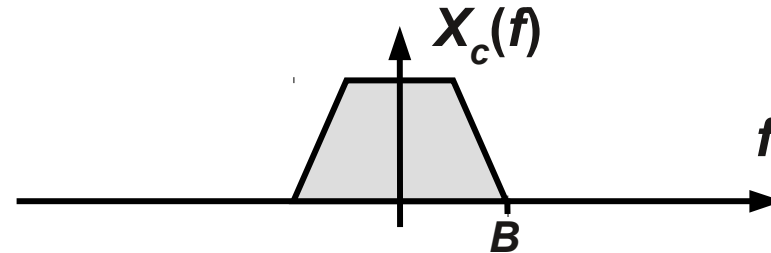


Achtung: manchmal auch Normierung auf $f_s / 2$ (Matlab) !

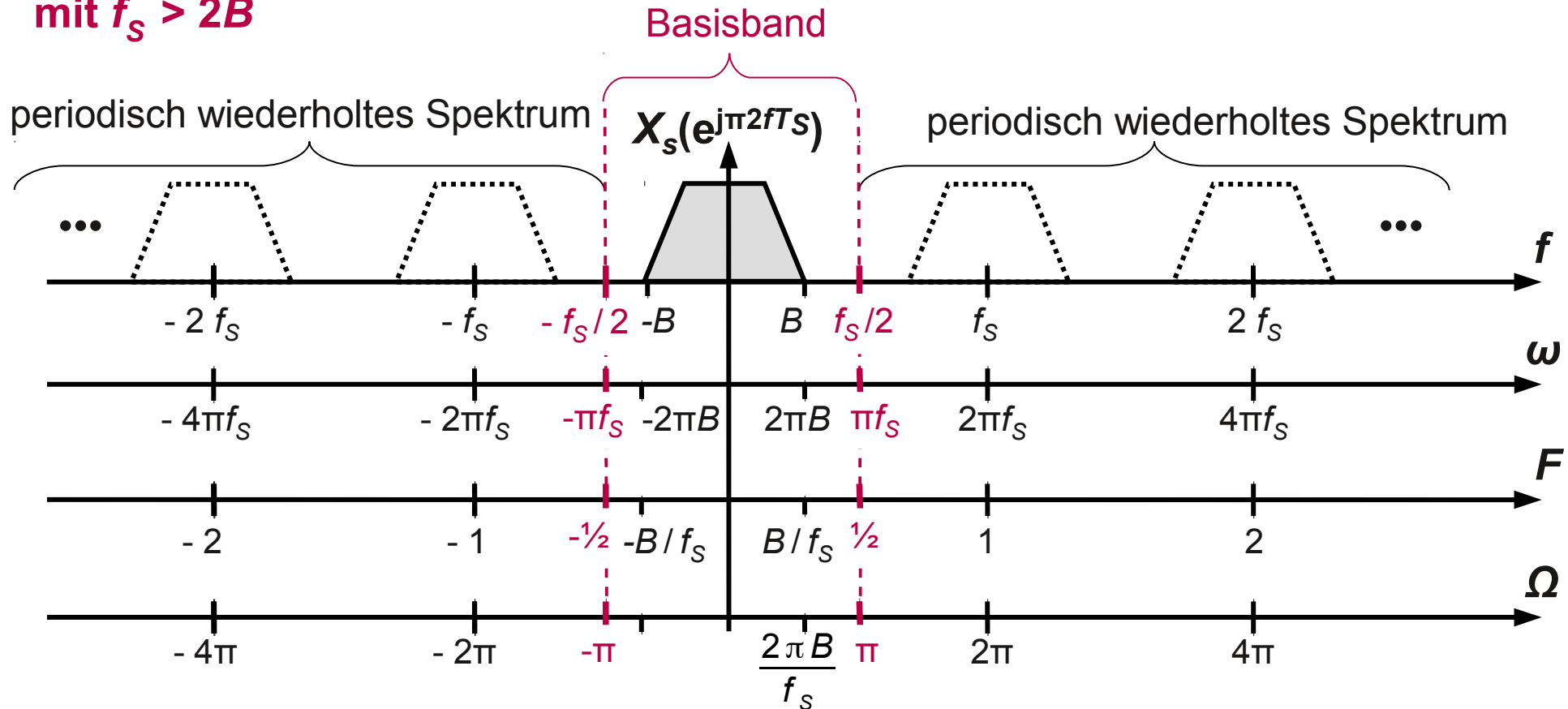
Frequenzdarstellung periodischer Spektren



Zeitkontinuierliches Eingangssignal mit Bandbreite B



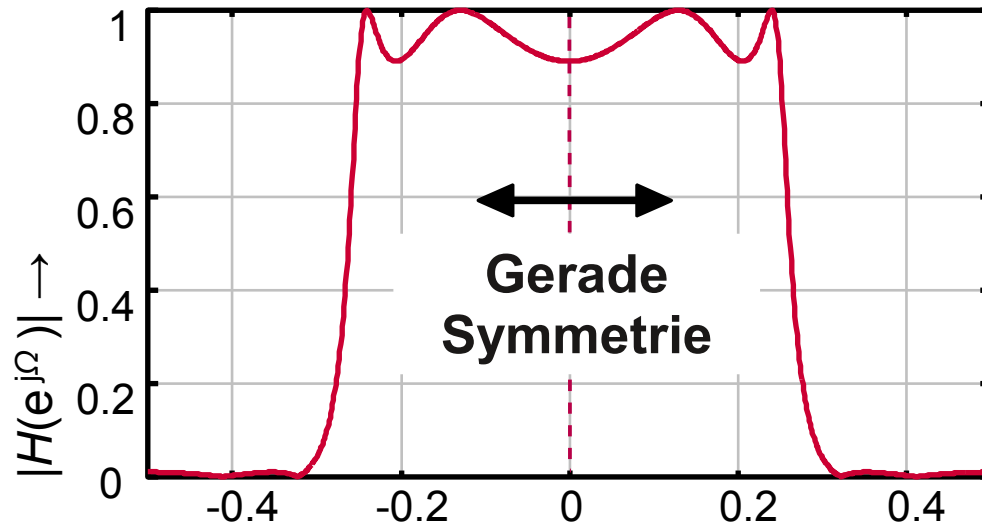
Abgetastetes Signal mit $f_s > 2B$



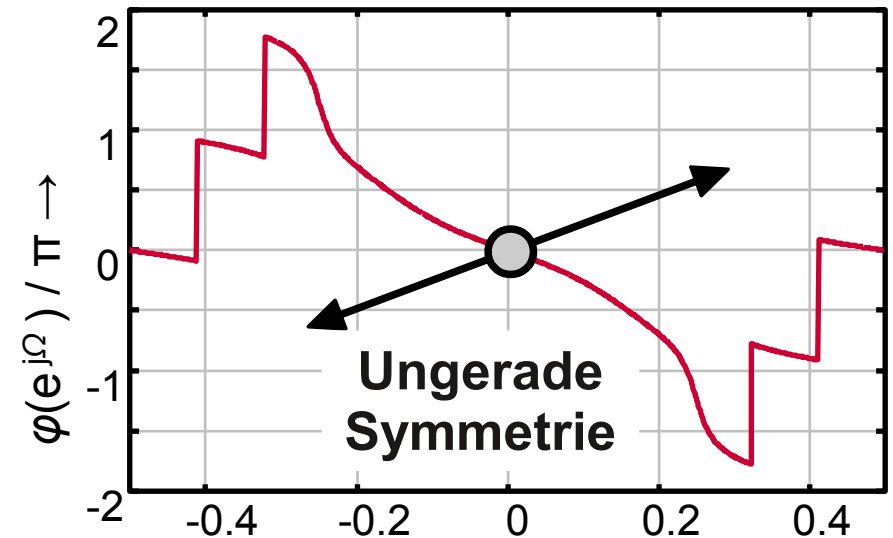
Spektren reellwertiger Signale und Systeme (1)



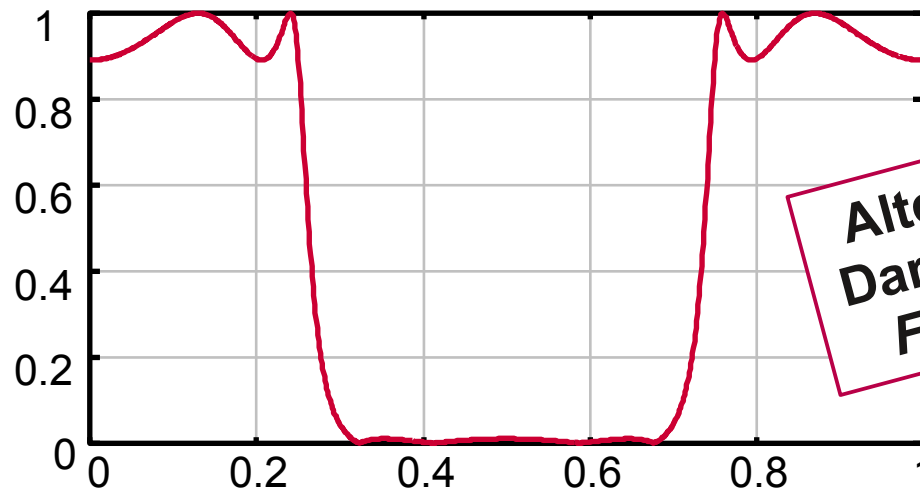
Betragsgang



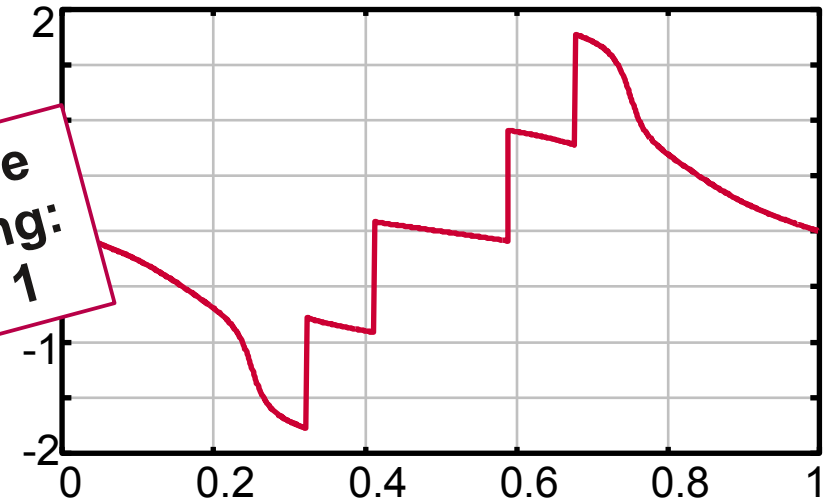
Phasengang



Normierte Frequenz $\Omega / 2\pi \rightarrow$



Alternative
Darstellung:
 $F = 0 \dots 1$

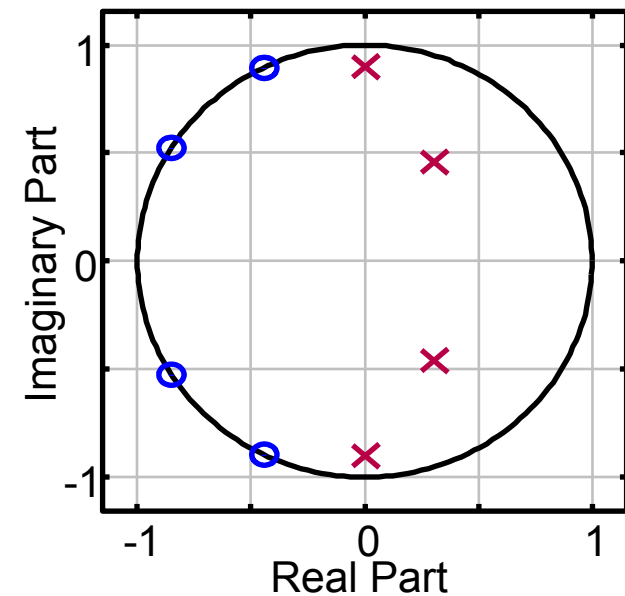
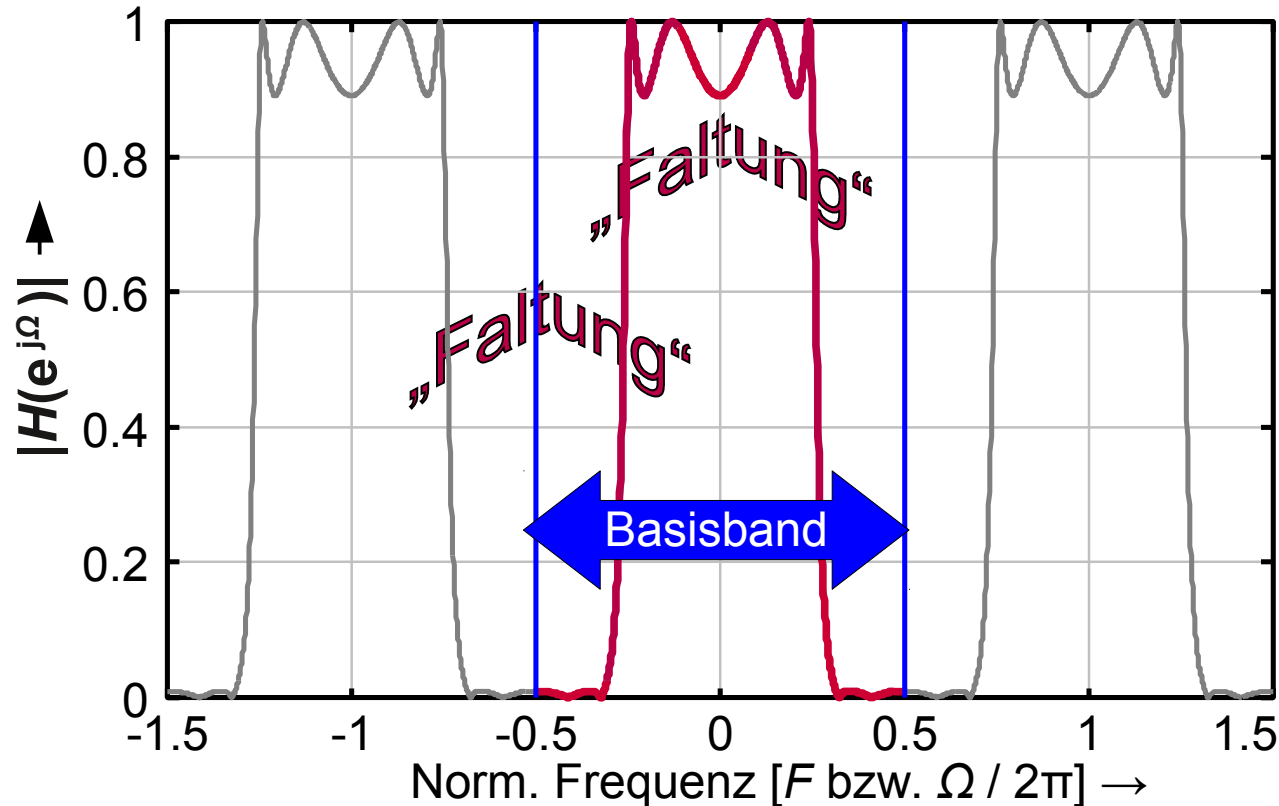


Kap2_PN_Plan_3D.m

Spektren reellwertiger Signale und Systeme (2)



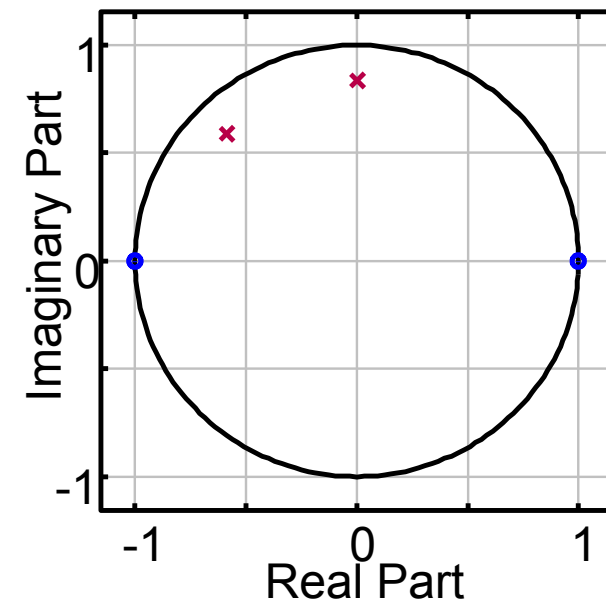
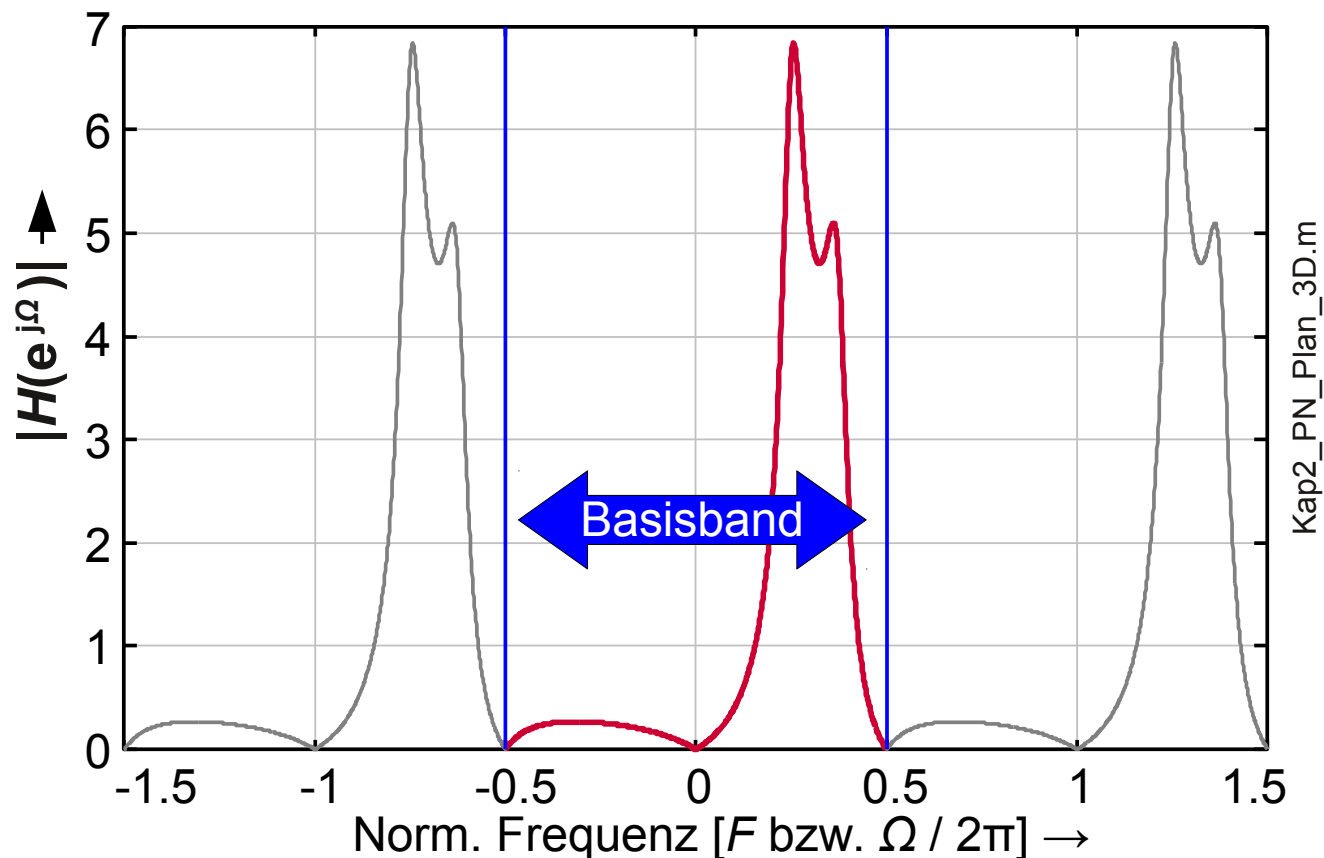
- Wiederholungspektren bei kf_s – immer bei zeitdiskreten Systemen
- Symmetrien um $kf_s/2$ - „Faltung“ - nur bei reellwertigen Systemen!
- Reellwertig \leftrightarrow P / N symm. zur x-Achse (konjugiert komplex)



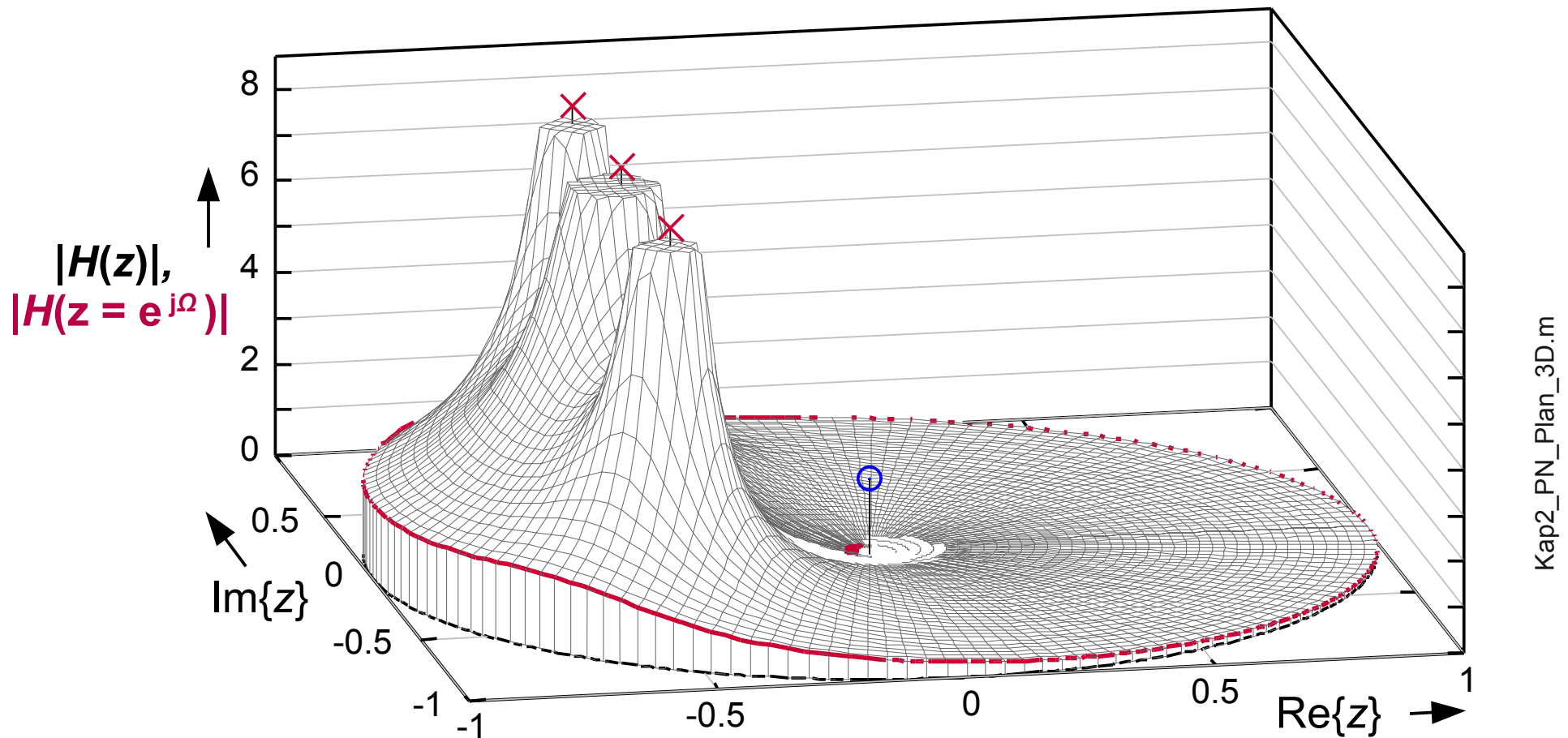
Spektren komplexwertiger Systeme



- Keine Symmetrien, nur Wiederholung bei kf_s - keine „Faltung“!
- In dieser Vorlesung fast ausschließlich reellwertige Systeme

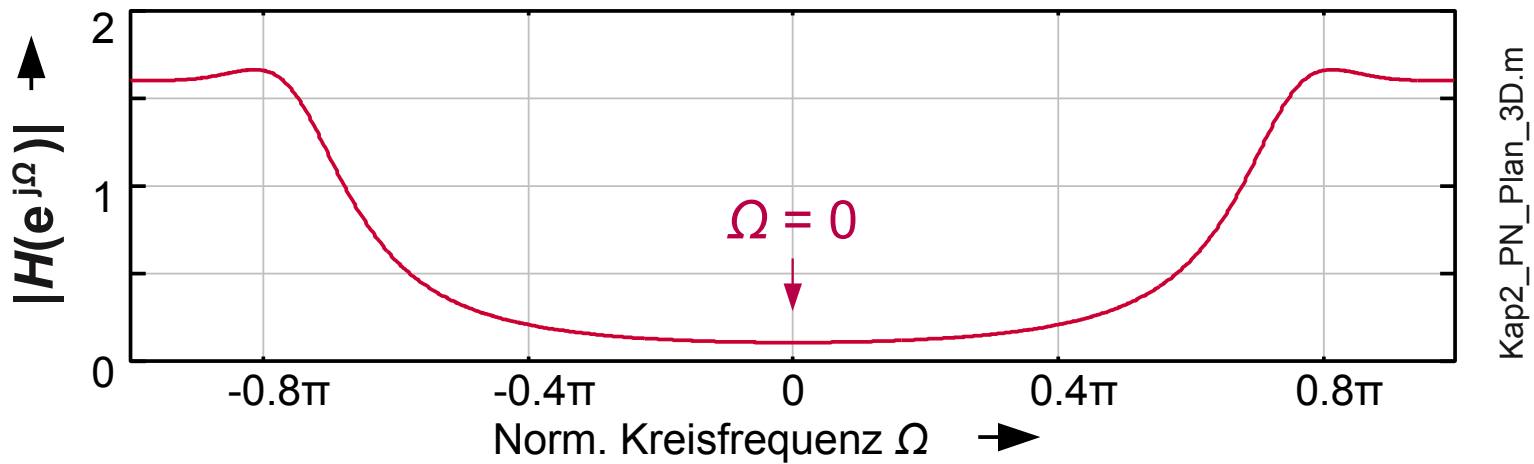
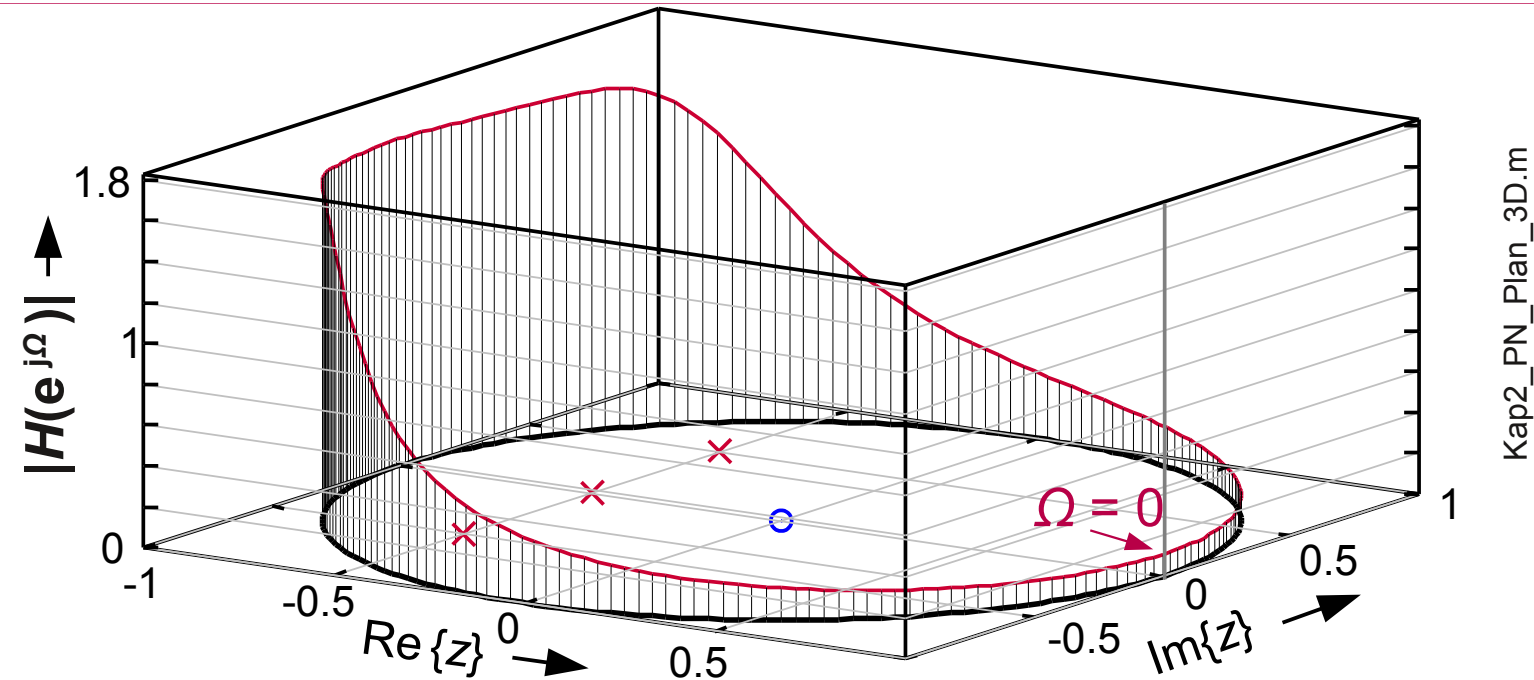


3D-Frequenzgang in der z-Ebene (1)



Frequenzgang $|H(e^{j\Omega})|$ wird entlang des Einheitskreises abgelesen

3D-Frequenzgang in der z-Ebene (2)



Digitale Signalverarbeitung auf FPGAs

Kap. 2 Zeitdiskrete Signale und
LTI-Systeme im Frequenzbereich

Teil 2 *Schreibformen für die Systemfunktion $H(z)$*

2016

Dr. Christian Münker

$H(z)$ kann durch mathematische Umformungen u.a. in die folgenden Formen gebracht werden:

Polynomform mit negativen Exponenten \leftrightarrow DZGL, SFG in
Direktform

Polynomform mit positiven Exponenten durch Erweitern mit z^N
 \leftrightarrow Produktform, Matlab / Python-Repräsentation

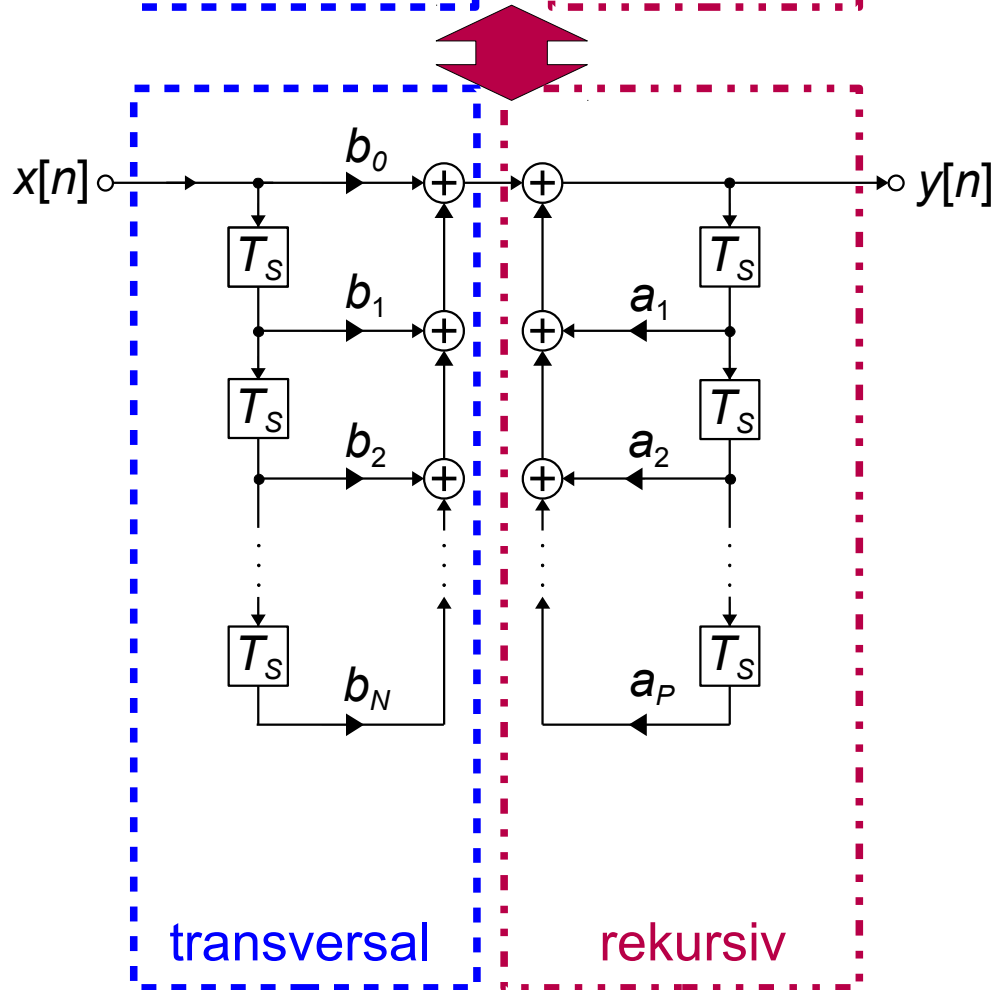
Produktform durch Faktorisierung \rightarrow Bestimmung von Pol/Nullstellen

Summenform durch Polynomdivision \rightarrow Summe aus Termen erster
und zweiter Ordnung (hier nicht behandelt)

Polynomform (neg. Exponenten) von $H(z)$



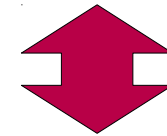
$$y[n] = \sum_{k=0}^N b_k x[n-k] + \sum_{i=1}^P a_i y[n-i] \quad \Leftrightarrow \quad Y(z) = \sum_{k=0}^N b_k X(z) z^{-k} + \sum_{i=1}^P a_i Y(z) z^{-i}$$



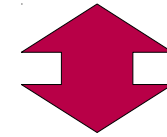
$$\Rightarrow H(z) = \frac{Y(z)}{X(z)} = \frac{\sum_{k=0}^N b_k z^{-k}}{1 - \sum_{i=1}^P a_i z^{-i}}$$

Nutzen:

Differenzengleichung



Übertragungsfunktion $H(z)$



Direkte Hardware-Konstruktion

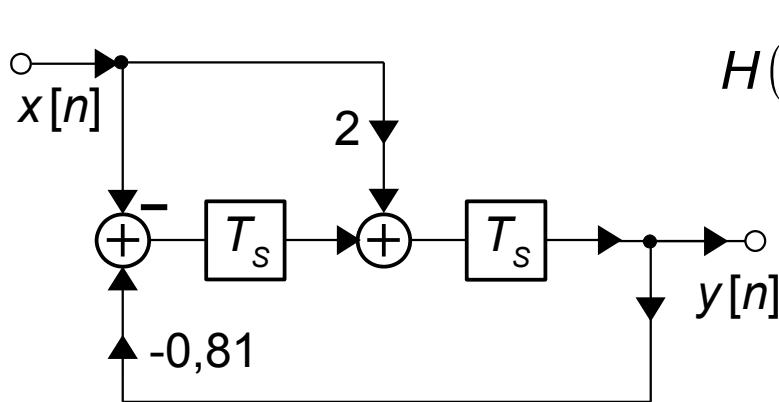
Polynomform (pos. Exponenten) von $H(z)$



Erweitern von Nenner und Zähler
mit $z^P z^N$ führt zu Polynomform
mit *positiven Exponenten*:

$$\Rightarrow H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 - \sum_{i=1}^P a_i z^{-i}} = \frac{\sum_{k=0}^N b_k z^{N-k}}{z^P - \sum_{i=1}^P a_i z^{P-i}} \frac{z^P}{z^N}$$

Nutzen: Diese Form kann leichter in die Produktform zerlegt werden
und gibt die Koeffizienten für Matlab / Python-Repräsentation an:



$$H(z) = \frac{2z^{-1} - z^{-2}}{1 + 0,81z^{-2}} = \frac{2z - 1}{z^2 + 0,81} = \frac{2(z - 0,5)}{(z - 0,9j)(z + 0,9j)}$$



ML / Python: $\mathbf{b} = [2, -1]$

$\mathbf{a} = [1, 0, 0.81]$

$$H(z) = \frac{\sum_{k=0}^N b_k z^{N-k}}{z^P - \sum_{i=1}^P a_i z^{P-i}} \frac{z^P}{z^N} = b_0 \frac{\prod_{k=1}^N (z - z_{0,k})}{\prod_{i=1}^P (z - z_{\infty,i})} z^{P-N}$$

$H(z)$ ist eindeutig definiert durch Koeffizienten a_k , b_k oder durch Nullstellen $z_{0,k}$ / Polstellen $z_{\infty,i}$ und Faktor b_0 !

Nutzen:

- **Stabilitätsanalyse:** Alle Pole innerhalb des EK ($|z_{\infty,i}| < 1$)?
- **Frequenzgang:** Geometrisches Abschätzen des Einflusses von P / N auf Punkte auf EK (nächster Abschnitt)
- **Systementwurf** im Frequenzbereich durch Platzieren von P / N

Digitale Signalverarbeitung auf FPGAs

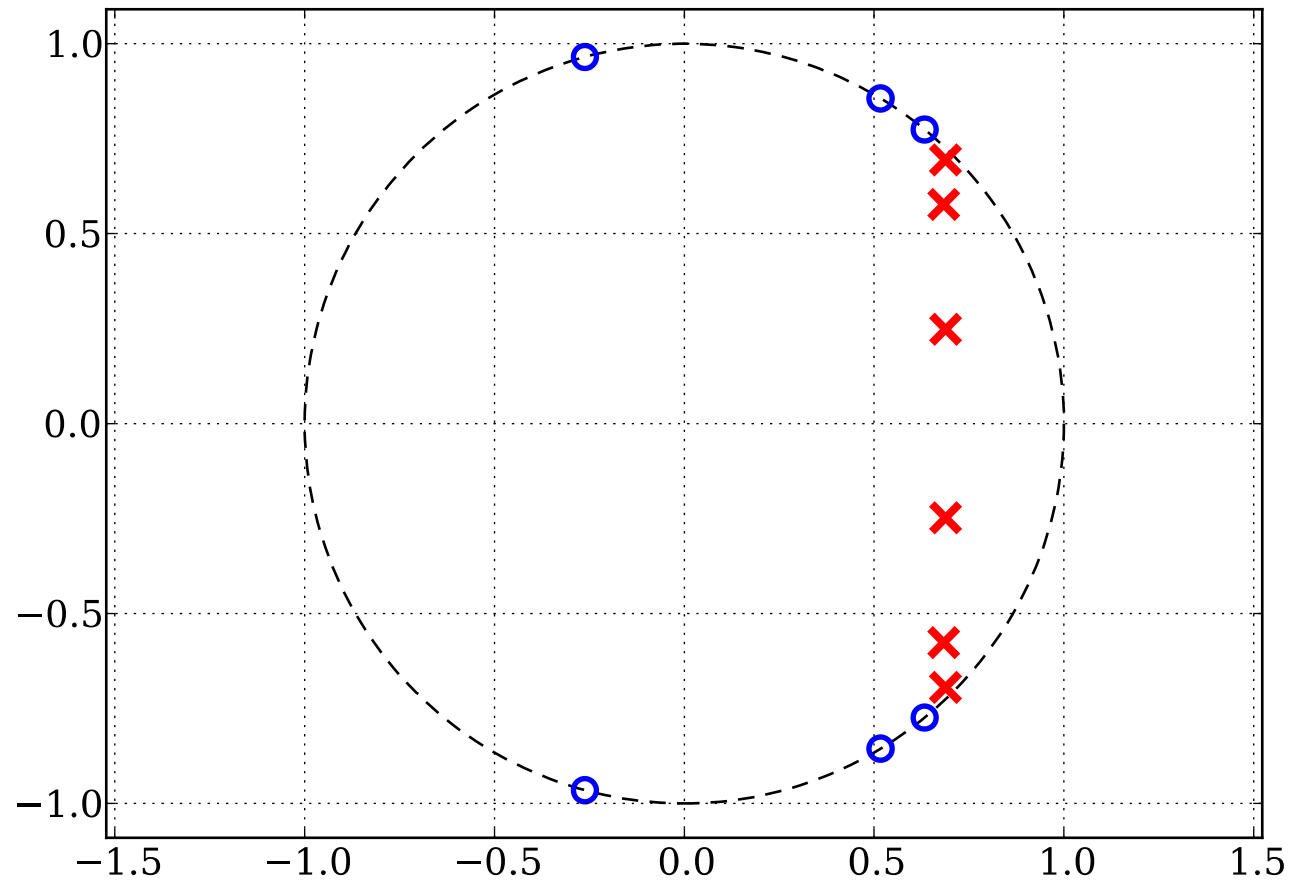
Kap. 2 Zeitdiskrete Signale und
LTI-Systeme im Frequenzbereich

Teil 3 *Pole und Nullstellen*

2016

Dr. Christian Münker

Pole und Nullstellen



Ausgangspunkt für Rechnungen / Simulationen: Zeitbereich

Impulsantwort / SFG eines transversalen (FIR) - Filters mit N Verzögerungen:

$$h[n] = \sum_{k=0}^N b_k \delta[n-k] \Leftrightarrow H(z) = \sum_{k=0}^N b_k z^{-k}, \quad b_k \in \mathbb{C} \quad \text{Systemfunktion in Polynomform mit neg. Exp.}$$

$$\Leftrightarrow H(z) = z^{-N} \sum_{k=0}^N b_k z^{N-k} \quad \text{Systemfunktion in Polynomform mit positiven Exponenten} \rightarrow \text{einfachere Zerlegung in Produktform}$$

$$\Leftrightarrow H(z) = b_0 z^{-N} \prod_{k=1}^N (z - z_{0,k}) \quad \text{Übertragungsfunktion in Produktform}$$

$\rightarrow N$ Nullstellen

$\rightarrow N$ Polstellen im Ursprung (z^{-N}) erkennbar

$H(z)$ **vollständig** definiert durch Koeff. b_k oder Nullstellen $z_{0,k}$ und Faktor b_0 !

Ausgangspunkt für Rechnungen / Simulationen: Zeitbereich

DZGL / SFG eines rekursiven (IIR) Filters mit N Verzögerungen im transversalen und P Verzögerungen im rekursiven Teil:

$$y[n] = \sum_{k=0}^N b_k x[n-k] + \sum_{i=1}^P a_i y[n-i] \Rightarrow H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 - \sum_{i=1}^P a_i z^{-i}} = b_0 \frac{\prod_{k=1}^N (z - z_{0,k})}{\prod_{i=1}^P (z - z_{\infty,i})} z^{P-N}$$

Übertragungsfunktion in Produktform:

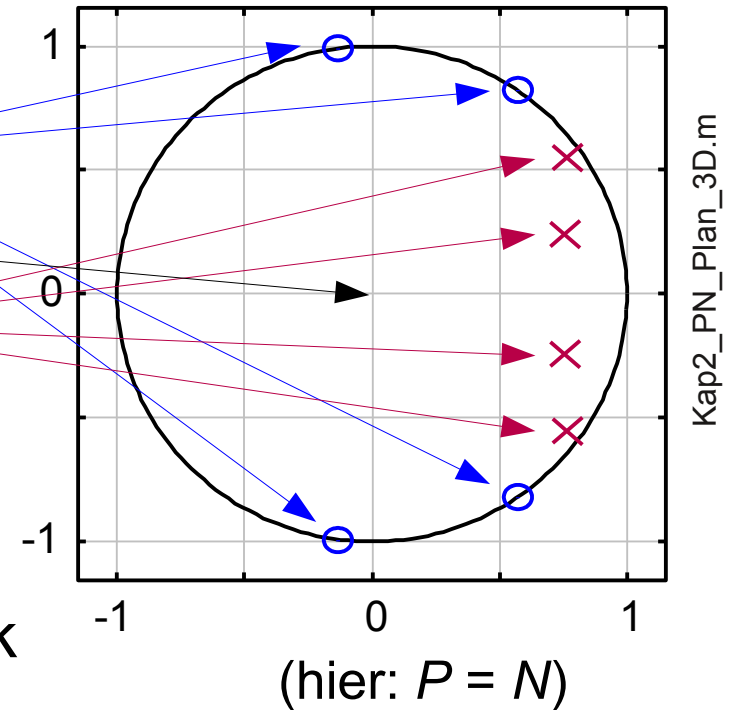
→ N Nullstellen, P Polstellen, zusätzlich

→ $(P - N)$ Nullstellen bzw.

$(N - P)$ Polstellen im Ursprung (z^{P-N}) → Abschnitt über Kausalität!

$H(z)$ **vollständig** definiert durch Null- und Polstellen $z_{0,k}$, $z_{\infty,i}$ und Faktor b_0 !

$$H(z) = \frac{\sum_{k=0}^N b_k z^{N-k}}{z^P - \sum_{i=1}^P a_i z^{P-i}} \frac{z^P}{z^N} = b_0 \frac{\prod_{k=1}^N (z - z_{0,k})}{\prod_{i=1}^P (z - z_{\infty,i})} z^{P-N}$$



Grafische Darstellung von Pol- und Nullstellen gibt viele Informationen auf einen Blick

- Stabilität
- Reellwertigkeit
- Frequenzgang
- Kausalität
- ...

empirisch:

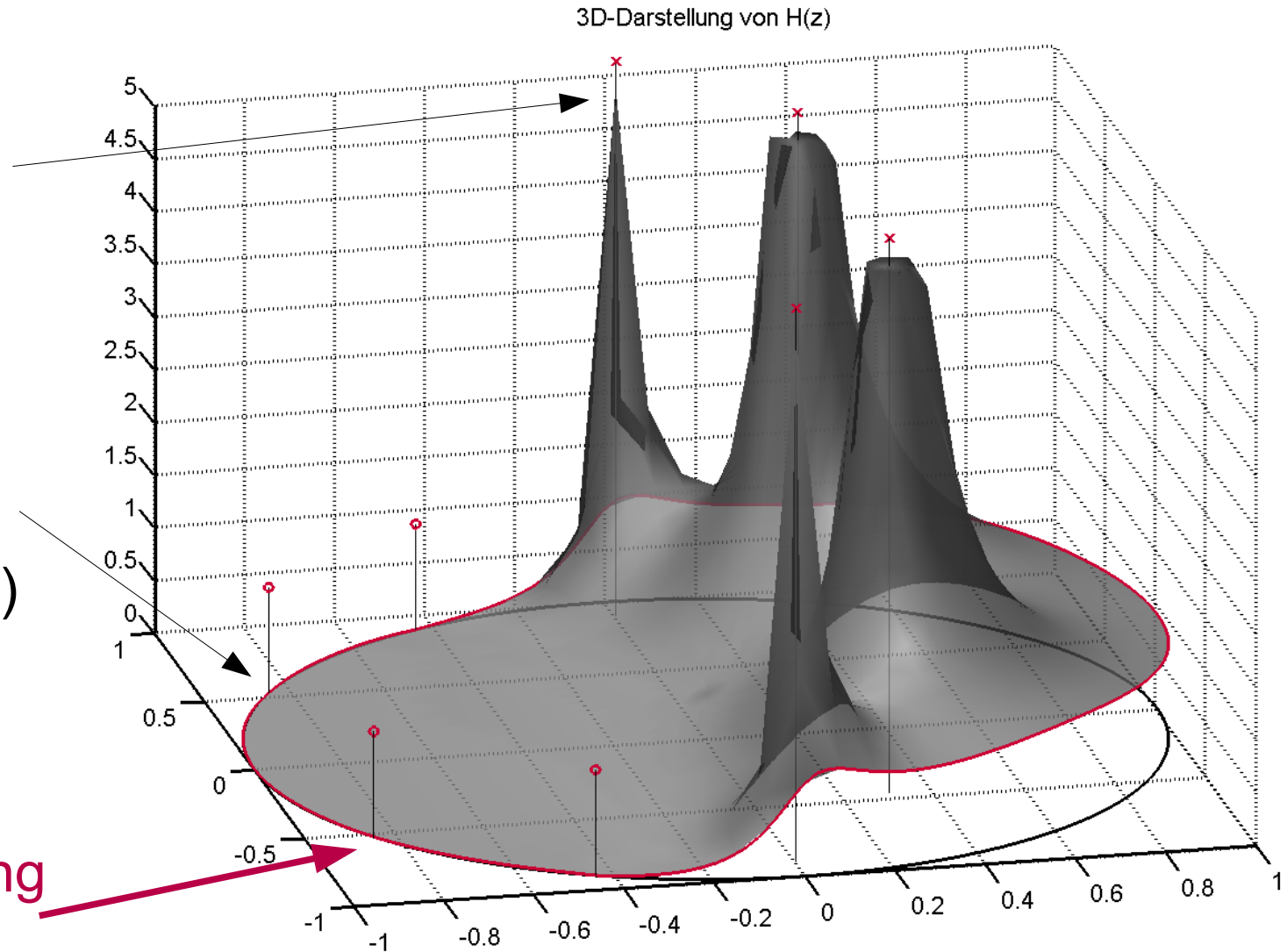
Polstellen =
Zeltstangen

+

Nullstellen =
Heringe
(Befestigung)

=

Frequenzgang
entlang EK

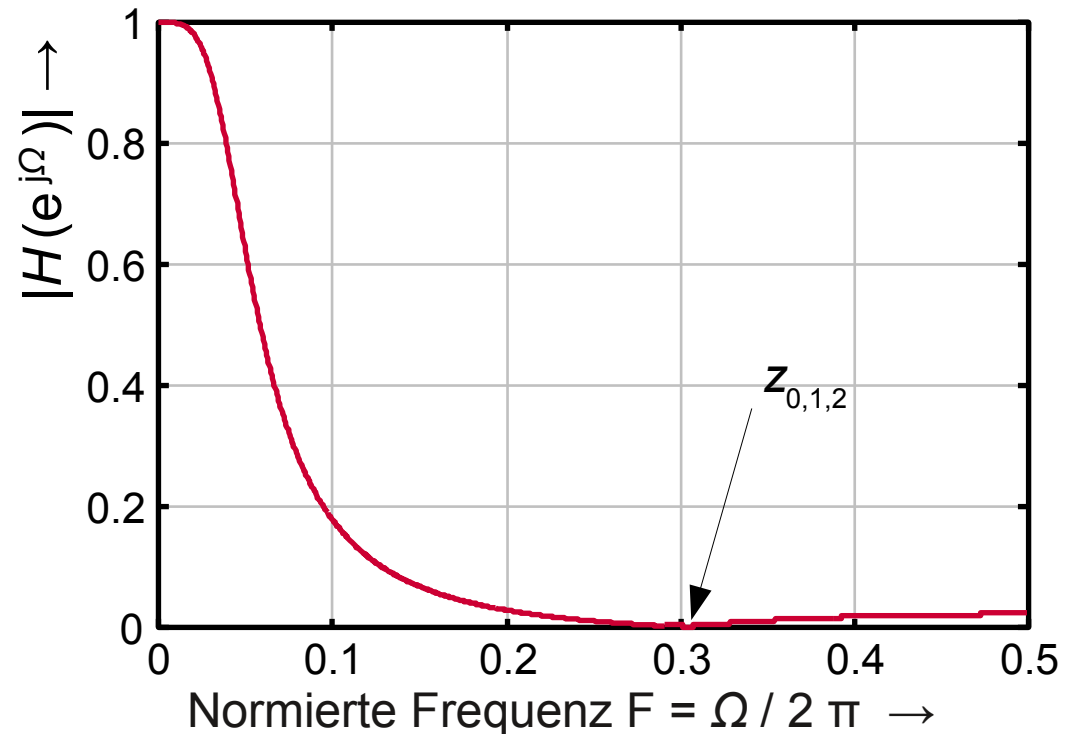
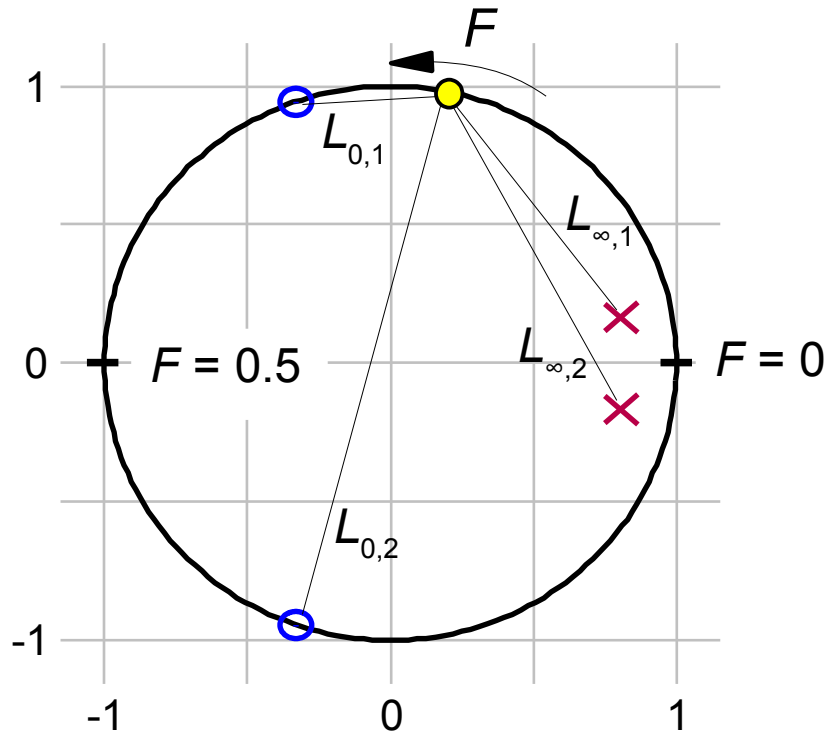


Einfluss von Polen / Nullstellen auf Betragsgang (2)



$$H(z) = b_0 \frac{\prod_{k=1}^N (z - z_{0,k})}{\prod_{i=1}^P (z - z_{\infty,i})} z^{P-N} \Rightarrow |H(z = e^{j\Omega})| = b_0 \frac{\prod_{k=1}^N \underbrace{|e^{j\Omega} - z_{0,k}|}_{L_{0,i}}}{\prod_{i=1}^P \underbrace{|e^{j\Omega} - z_{\infty,i}|}_{L_{\infty,i}}} \underbrace{|e^{j\Omega(P-N)}|}_{=1}$$

Geometrische Deutung:

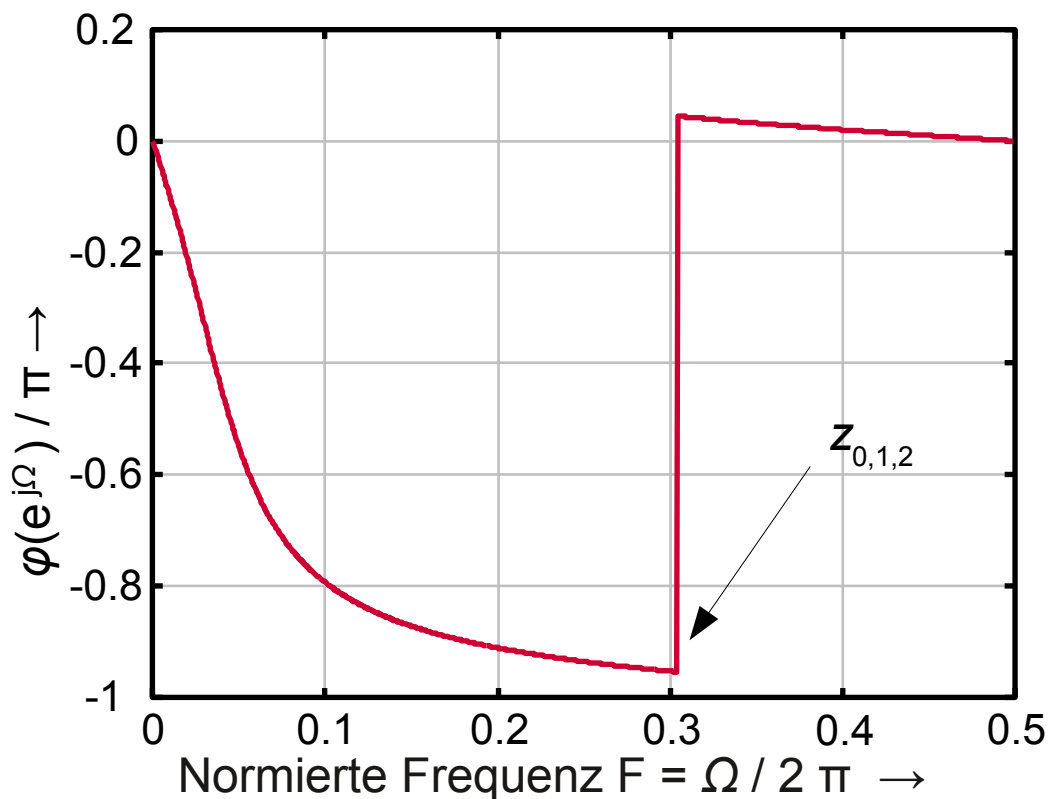
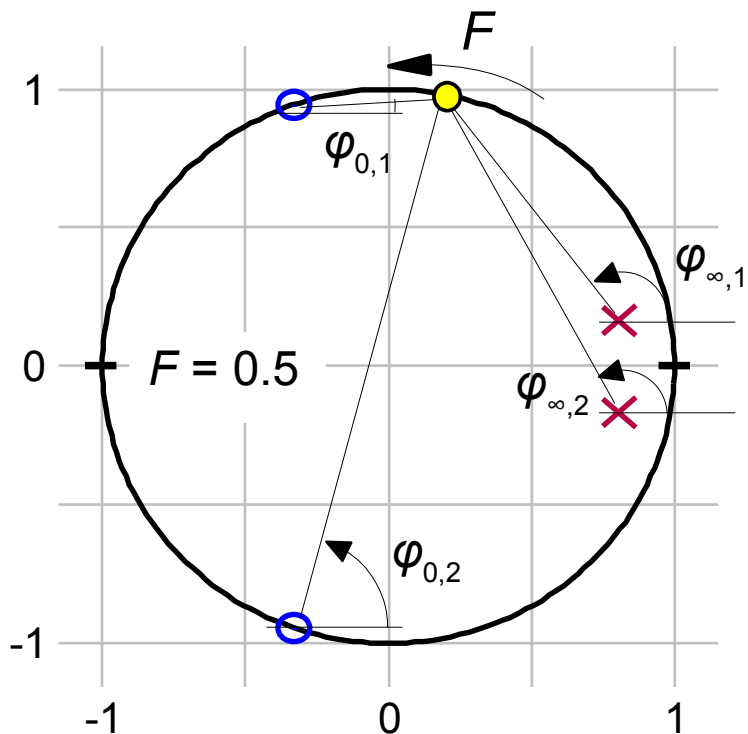


Einfluss von Polen / Nullstellen auf Phasengang

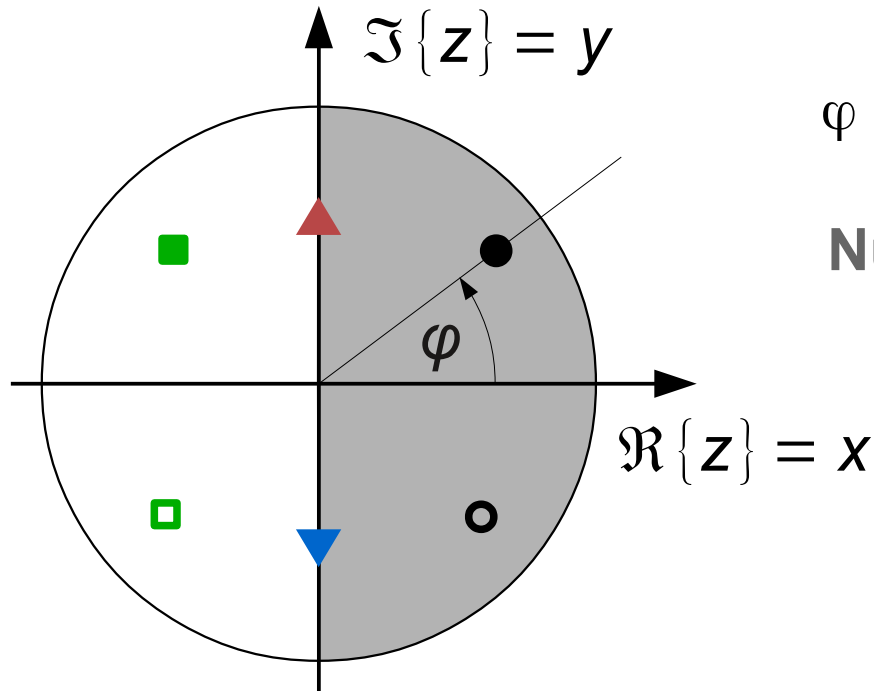


$$H(e^{j\Omega}) = b_0 \frac{\prod_{k=1}^N e^{j\Omega} - z_{0,k}}{\prod_{i=1}^P e^{j\Omega} - z_{\infty,i}} e^{j\Omega(P-N)} \Rightarrow \varphi(e^{j\Omega}) = \sum_{k=1}^N \underbrace{\angle(e^{j\Omega} - z_{0,k})}_{\varphi_{0,k}} - \sum_{i=1}^P \underbrace{\angle(e^{j\Omega} - z_{\infty,i})}_{\varphi_{\infty,i}} + \underbrace{\Omega(P-N)}_{= \text{lineare Phase}}$$

Geometrische Deutung:



Atan2 und arctan - Funktion



$$\varphi = \arctan \frac{y}{x} = \arctan \frac{-y}{-x}$$

Nur eindeutig für $x > 0$!

Neue Definition: $\varphi = \text{atan2}(y, x) = \begin{cases} \arctan \frac{y}{x} & \text{für } x > 0 \\ \arctan \frac{y}{x} + \pi & \text{für } x < 0, y \geq 0 \\ \arctan \frac{y}{x} - \pi & \text{für } x < 0, y < 0 \\ \frac{\pi}{2} & \text{für } x = 0, y > 0 \\ -\frac{\pi}{2} & \text{für } x = 0, y < 0 \end{cases}$

Matlab: `atan2(y, x)` oder `angle(z)`

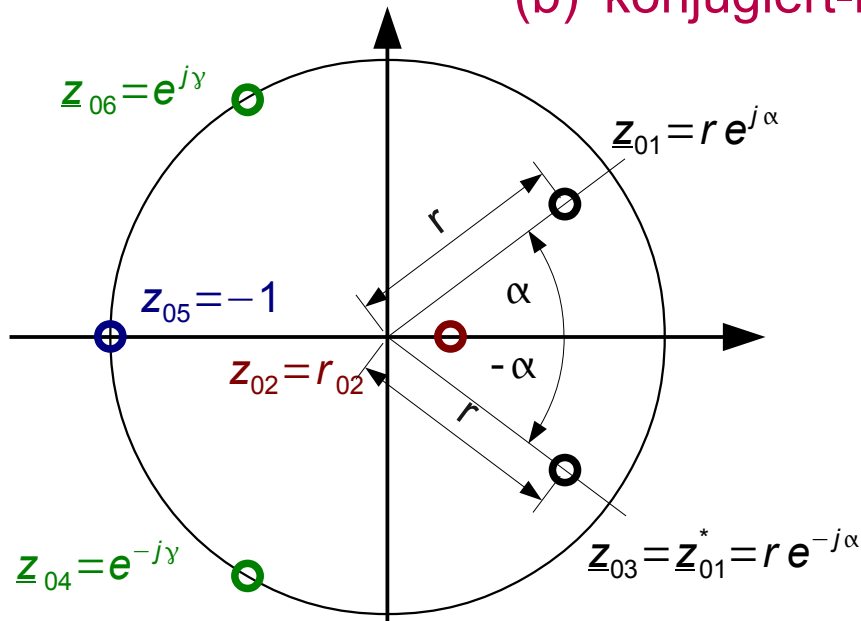
Python: `np.arctan2(y, x)`
oder `np.angle(z)`

Wo müssen Nullstellen z_{0k} der Übertragungsfunktion $H(z)$ liegen, damit ein Filter nur reellwertige Koeffizienten b_k hat (analog für Polstellen)?

$$h[n] = \sum_{k=0}^N b_k \delta[n-k] \Rightarrow H(z) = \sum_{k=0}^N b_k z^{-k} = \sum_{k=0}^N b_k \frac{z^{N-k}}{z^N} = b_0 z^{-N} \prod_{k=1}^N (z - z_{0k}), \quad b_k \in \mathbb{R}$$

Erfüllbar für: (a) reellwertige Nullstellen (z_{02} und z_{05})

(b) konjugiert-komplexe Nullstellenpaare ($z_{01} = z_{03}^*$, $z_{04} = z_{06}^*$)



Produkt von zwei konjugiert komplexen Nullstellen liefert *reellwertigen* Beitrag!

$$(z - z_{0k})(z - z_{0k}^*) = z^2 - 2\Re\{z_{0k}\}z + |z_{0k}|^2$$

Gilt genauso für Polstellen(paare) bei rekursiven Systemen!

Digitale Signalverarbeitung auf FPGAs

Kap. 2 Zeitdiskrete Signale und LTI-Systeme im Frequenzbereich

Teil 4 *Kausalität*

2016

Dr. Christian Münker

caffeine causality loop



wronghands1.wordpress.com

© John Atkinson, Wrong Hands

„caffeine causality loop“
(<http://wronghands1.com/2014/01/31/caffeine-causality-loop/>)
von John Atkinson (wronghands1@gmail.com)
unter CC-BY-NC-ND-3.0 Lizenz

$$H(z) = \frac{\sum_{k=0}^N b_k z^{-k}}{1 - \sum_{i=1}^P a_i z^{-i}} = b_0 \frac{\prod_{k=1}^N (z - z_{0,k})}{\prod_{i=1}^P (z - z_{\infty,i})} z^{P-N}$$

- Anzahl der Nullstellen außerhalb des Ursprungs ist Grad des Zählerpolynoms (N)
- Anzahl der Pole außerhalb des Ursprungs ist Grad des Nennerpolynoms (P)
- „Ordnung“ des Filters ist $\max[N, P]$
 - falls $P > N$: es gibt $P - N$ zusätzliche Nullstellen im Ursprung
 - falls $P < N$: es gibt $N - P$ zusätzliche Pole im Ursprung (z.B. FIR: $P = 0$)

Da P / N im Ursprung keinen Einfluss auf Betragsgang haben (warum?), werden sie im P / N -Diagramm häufig nicht eingetragen.
- Aber welchen Einfluss haben denn P / N im Ursprung überhaupt ???

Entwurf ausgehend von der Differenzengleichung oder der Struktur ergibt immer ein **kausales System** (= keine Reaktion am Ausgang bevor Stimulus am Eingang angelegt wird).

Voraussetzung: Es werden nur Verzögerungen verwendet, z.B. $x[n-3]$

Das gleiche gilt, wenn man mit der Übertragungsfunktion $H(z)$ in Polynomform mit *negativen Koeffizienten* beginnt, z.B. z^{-3} (warum?)

Startet man mit der Polynomform mit positiven Koeffizienten oder durch Angabe der P/N, können **akausale Systeme** entstehen.

Kausalität von $H(z)$ (2)



Kausalität lässt sich am Einfachsten im Zeitbereich (Differenzengleichung) beurteilen, $y[n]$ darf nicht abhängen von $x[n + k]$:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{k \prod_{i=1}^N (z - z_{0,i})}{\prod_{i=1}^P (z - z_{\infty,i})} = \frac{\sum_{i=0}^N \tilde{b}_i z^i}{z^P + \sum_{i=1}^P a_i z^i} = \frac{z^N \sum_{i=0}^N \tilde{b}_{N-i} z^{-i}}{z^P \left(1 + \sum_{i=1}^P a_i z^{-i} \right)} = \frac{z^{N-P} \sum_{i=0}^N b_i z^{-i}}{1 + \sum_{i=1}^P a_i z^{-i}}; \quad b_i = \tilde{b}_{N-i}$$

$$\Rightarrow Y(z) = z^{N-P} \sum_{i=0}^N b_i z^{-i} X(z) - \sum_{i=1}^P a_i z^{-i} Y(z)$$



$$y[n] = \delta[n+N-P] * \sum_{i=0}^N b_i x[n-i] - \sum_{i=1}^P a_i y[n-i] = \sum_{i=0}^N b_i x[n+N-P-i] - \sum_{i=1}^P a_i y[n-i]$$

→ $N \leq P$, ansonsten wird System akausal!

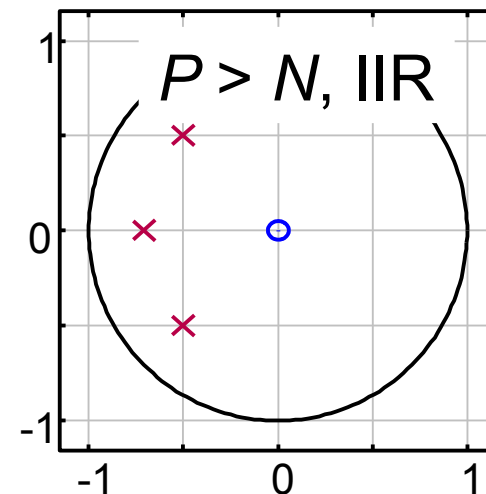
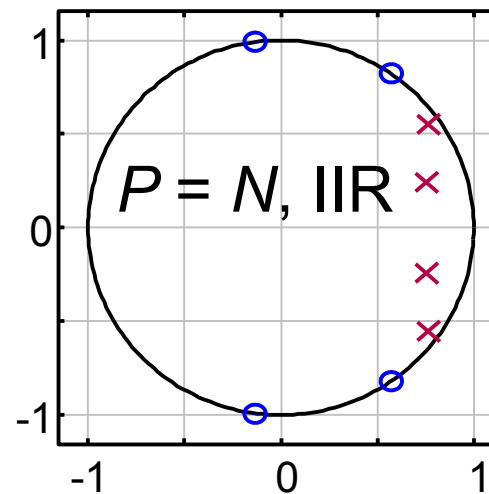
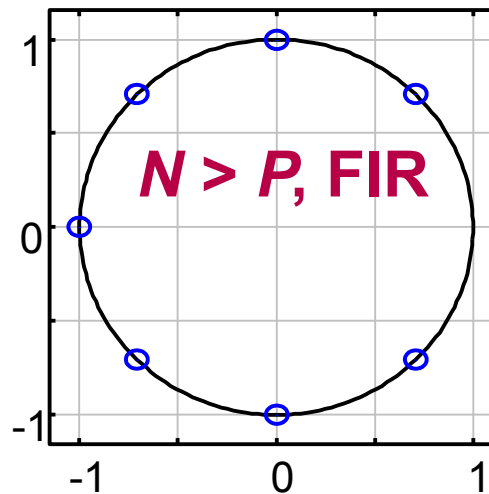
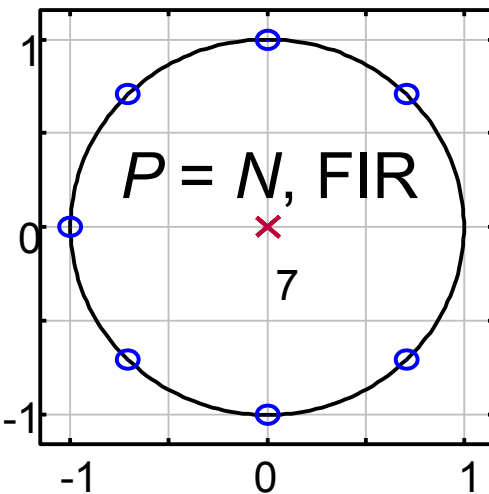
Kausalität von $H(z)$ (3)



$$H(z) = \frac{Y(z)}{X(z)} = k \frac{\prod_{i=1}^N (z - z_{0,i})}{\prod_{i=1}^P (z - z_{\infty,i})} = \frac{\sum_{i=0}^N \tilde{b}_i z^i}{z^P + \sum_{i=1}^P a_i z^i} = \frac{\sum_{i=0}^N b_i z^{-i}}{1 + \sum_{i=1}^P a_i z^{-i}} z^{N-P}; \quad b_i = \tilde{b}_{N-i}$$

- $P > N$: zusätzliche Verzögerung, $z^{-(N-P)}$
- $N > P$: nicht-kausales System, $z^{+(N-P)}$

Kap2_PN_Plan_3D.m



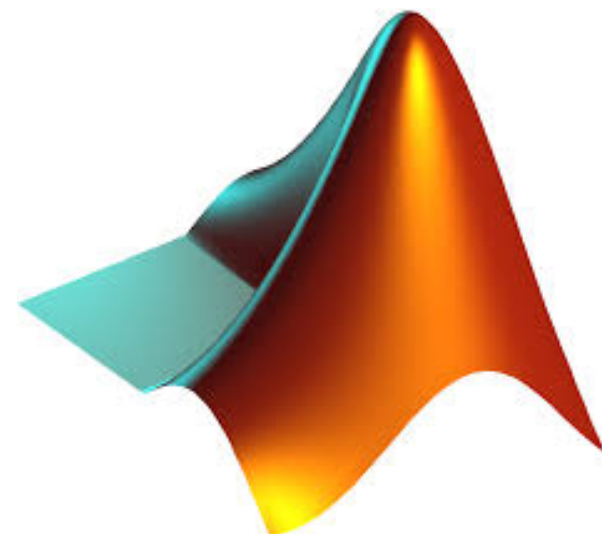
Digitale Signalverarbeitung auf FPGAs

Kap. 2 Zeitdiskrete Signale und
LTI-Systeme im Frequenzbereich

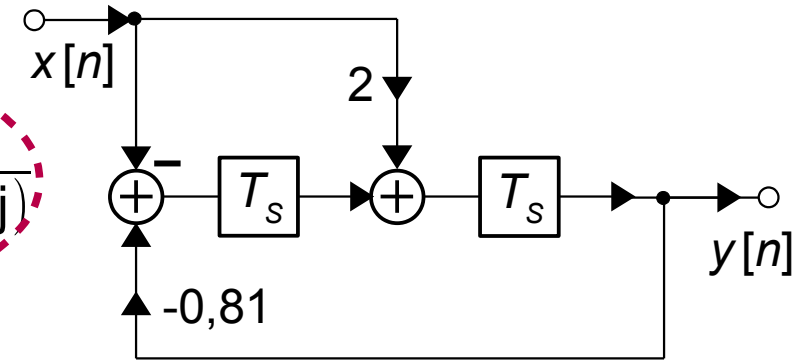
Teil 5 *Darstellung von $H(z)$ mit Python und Matlab*

2016

Dr. Christian Münker



$$H(z) = \frac{2z^{-1} - z^{-2}}{1 + 0,81z^{-2}} = \frac{2z - 1}{z^2 + 0,81} = \frac{2(z - 0.5)}{(z - 0.9j)(z + 0.9j)}$$



Zähler- / Nennerkoeff.: `b = [2, -1]; a = [1, 0, 0.81];`

Nullstellen / Pole: `N = [0; 0.5]; P = [-0.9*j; 0.9*j];`

Umrechnung:
`N = roots(b); P = roots(a);`
`b = k0 * poly(P); a = poly(N);`

oder `[b, a] = zp2tf(z, p, k0); bzw. [z, p, k0] = tf2zp(b, a);`

Achtung: In Matlab sind Koeffizienten Zeilenvektoren, P/N Spaltenvektoren!

Python: `np.roots()`, `np.poly()`, `scipy.signal.zpk2tf()` und `tf2zpk`

Pol/Nullstellen-Diagramm

Nullstellen / Pole: `zplane(N, P) ;`

Zähler- / Nennerkoeff.: `zplane(b, a) ;`

Je nachdem ob `zplane` mit Spalten- oder Zeilenvektoren aufgerufen wird, werden die Vektoren als P/N oder Koeffizienten interpretiert!

P bzw. a sind optional. Welche Werte haben P und a bei FIR-Filtern?

Frequenzgang

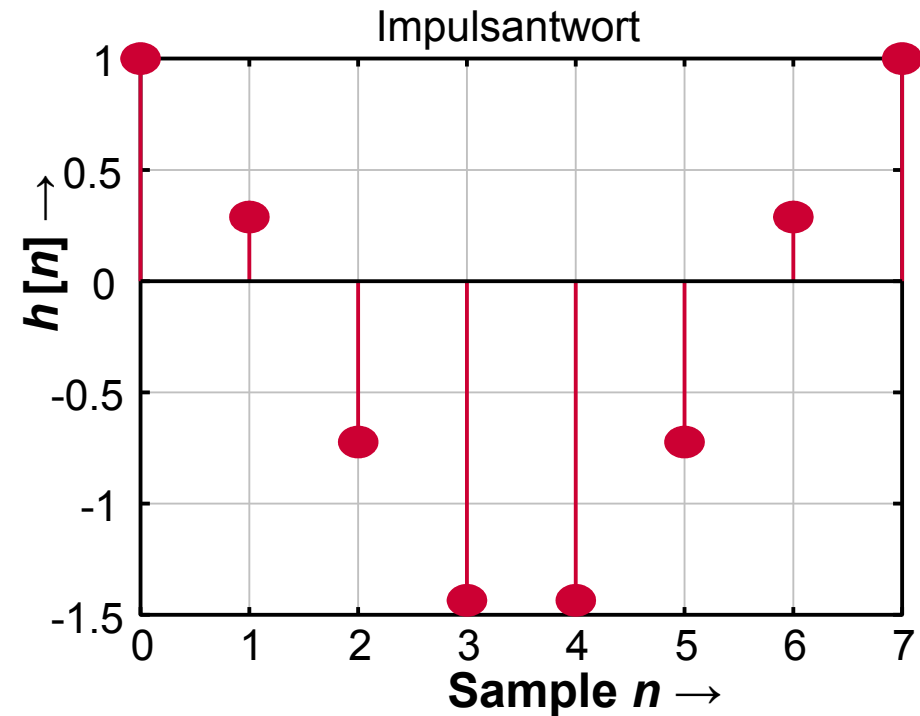
Kombiniert Betrag / Phase: `freqz(b, a) ;`

Detaillierte Plots: `[F, H] = freqz(b, a) ;`

Weitere Optionen sind die Anzahl der Frequenzpunkte und die Abtastfrequenz (Python: `scipy.signal.freqz()`).

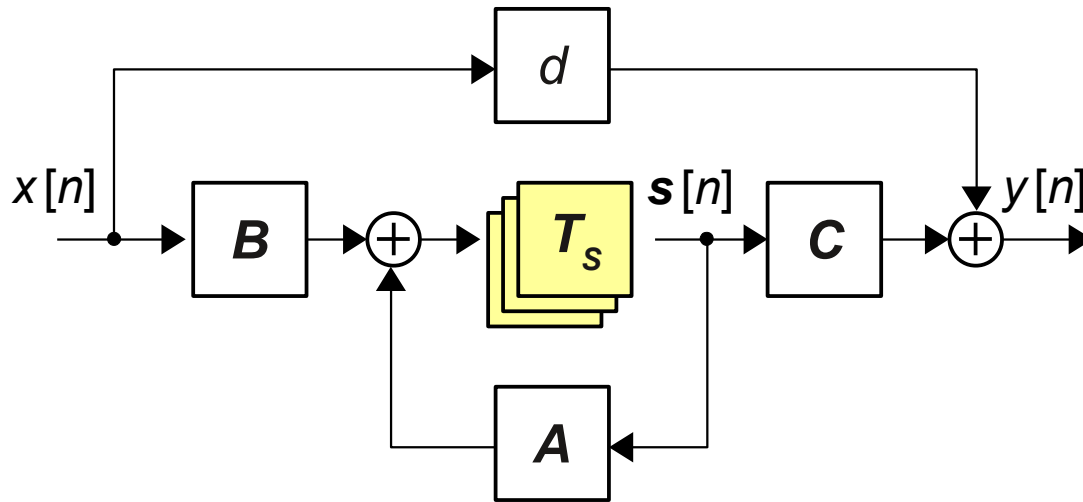

```
[himp, t]=impz(b, a, n, f_S);  
stem(t, himp);
```

- Wenn P / N (Produktform) gegeben sind: Bestimmung der Koeffizienten \mathbf{b} , \mathbf{a} mit `poly()`
- Koeffizienten \mathbf{b} entsprechen Impulsantwort bei FIR-Filter



a , n , f_S sind optional. n gibt die Anzahl der zu plottenden Impulse an, ansonsten versucht Matlab eine gute Darstellung zu finden. Durch Angabe von f_S wird die t -Achse mit absoluten Zeitangaben skaliert.

State-Space Darstellung von $H(z)$



$$\mathbf{s}[n+1] = \mathbf{A}\mathbf{s}[n] + \mathbf{B}x[n]$$

$$y[n] = \mathbf{C}\mathbf{s}[n] + d x[n]$$



$$\mathbf{S}(z)\mathbf{z} = \mathbf{A}\mathbf{S}(z) + \mathbf{B}X(z)$$

$$Y(z) = \mathbf{C}\mathbf{S}(z) + d X(z)$$

- Beschreibung eines beliebigen LTI Netzwerks mit Hilfe seiner **Zustandsvariablen** $\mathbf{s}[n]$ (= **state** variables, Register) und Differenzen/Differenzialgleichungen erster Ordnung
- Eingangssignal $x[n]$, Ausgangssignal $y[n]$ und die Zustandsvariablen $\mathbf{s}[n]$ sind über die Vektoren / Matrizen \mathbf{A} , \mathbf{B} , \mathbf{C} , d miteinander verknüpft
- Hier: Single-Input, Single-Output (SISO), aber MIMO genauso möglich

Digitale Signalverarbeitung auf FPGAs

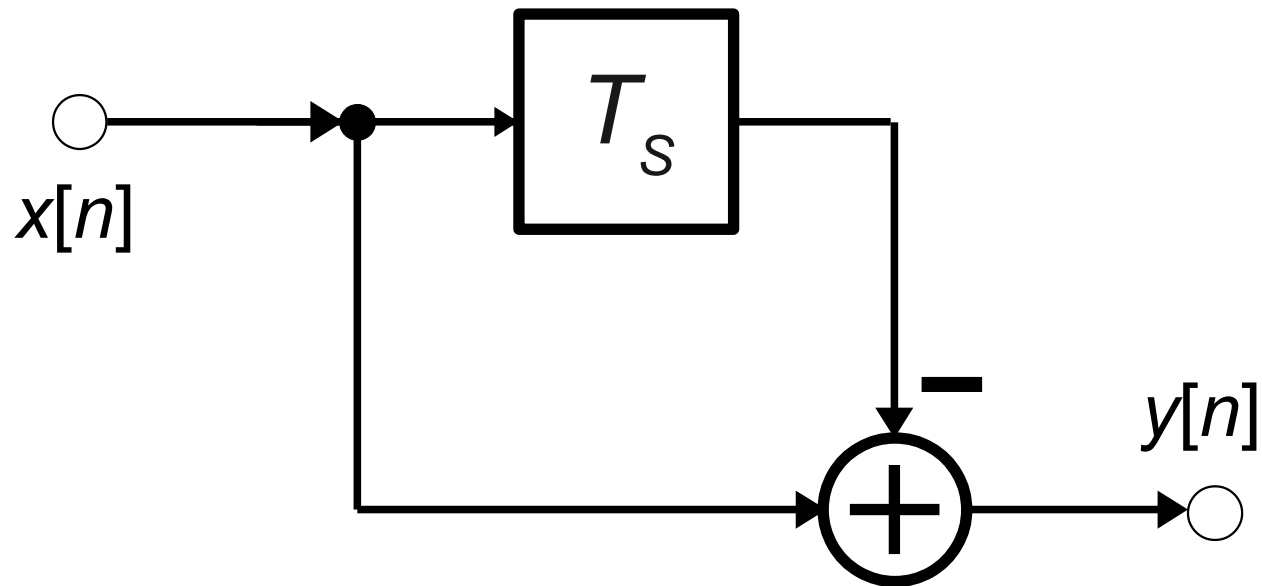
Kap. 2 Zeitdiskrete Signale und
LTI-Systeme im Frequenzbereich

Teil 6 *Einfache Filter*

2016

Dr. Christian Münker

Einfache Filter



Euler

$$\begin{aligned}e^{\pm j\Omega} &= \cos \Omega \pm j \sin \Omega \\e^{j\Omega} + e^{-j\Omega} &= 2 \cos \Omega \\e^{j\Omega} - e^{-j\Omega} &= 2j \sin \Omega\end{aligned}$$

Endliche geometrische Reihe

$$\begin{aligned}H(z) &= \sum_{n=0}^N z^{-n} = \frac{1 - z^{-(N+1)}}{1 - z^{-1}} \\ \Rightarrow H(e^{j\Omega}) &= \sum_{n=0}^N e^{-jn\Omega} = \frac{1 - e^{-j(N+1)\Omega}}{1 - e^{-j\Omega}}\end{aligned}$$

**Zerlegung in
„Spiegelpolynome“**

$$\begin{aligned}H(z) &= 1 + z^{-N} = z^{-N/2} (z^{N/2} + z^{-N/2}) \\ \Rightarrow H(e^{j\Omega}) &= e^{-jN\Omega/2} (e^{jN\Omega/2} + e^{-jN\Omega/2}) = 2e^{-jN\Omega/2} \cos(N\Omega/2)\end{aligned}$$

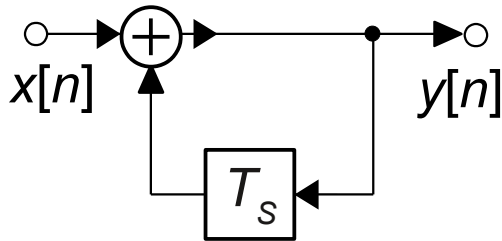
N-fache Nullstelle auf EK

$$1 - z^N = 0 \Rightarrow z = \sqrt[N]{1} = \sqrt[N]{e^{j2k\pi}} \Rightarrow z = e^{j2k\pi/N} \text{ mit } k = \dots, -1, 0, 1, \dots$$

$$1 + z^N = 0 \Rightarrow z = \sqrt[N]{-1} = \sqrt[N]{e^{j\pi(2k+1)}} \Rightarrow z = e^{j\pi(2k+1)/N} \text{ mit } k = \dots, -1, 0, 1, \dots$$

$\Rightarrow N$ Nullstellen auf EK verteilt mit $\Delta\varphi = 2\pi/N$

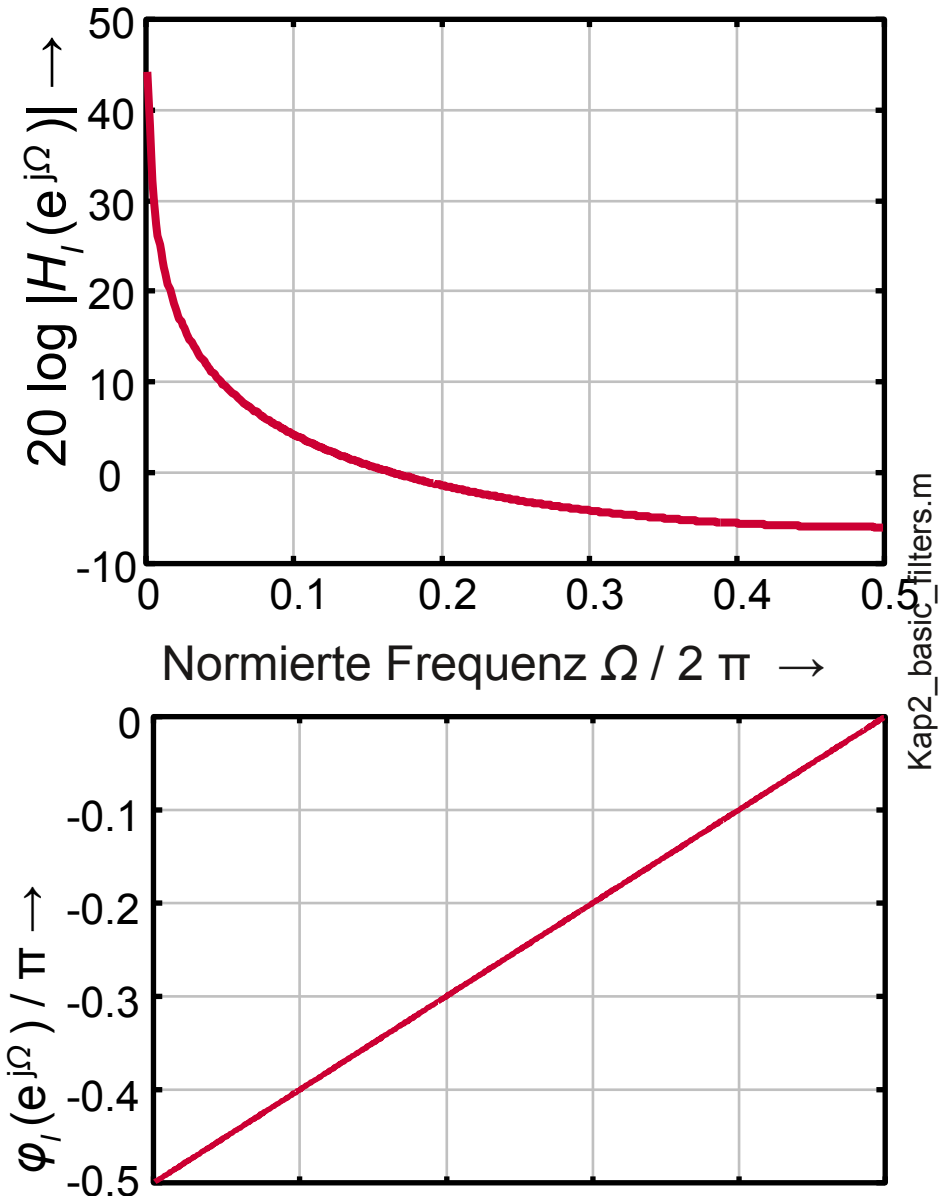
Einfache Filter: Integrator



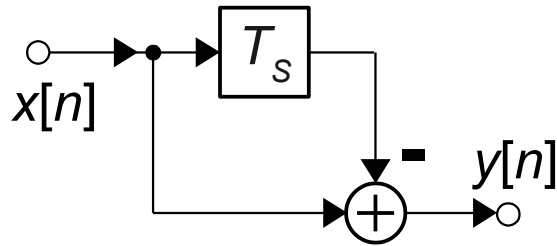
$$H_I(z) = \frac{1}{1 - z^{-1}} = \frac{z}{z - 1}$$

$$\begin{aligned} H_I(e^{j\Omega}) &= \frac{1}{1 - e^{-j\Omega}} = \frac{1}{e^{-j\Omega/2}(e^{+j\Omega/2} - e^{-j\Omega/2})} \\ &= \frac{e^{j\Omega/2}}{2j \sin(\Omega/2)} \approx \frac{1}{j\Omega} \quad \text{für } \Omega \ll 1 \end{aligned}$$

- P/N-Diagramm?
- Stabilität?
- Lineare Phase??



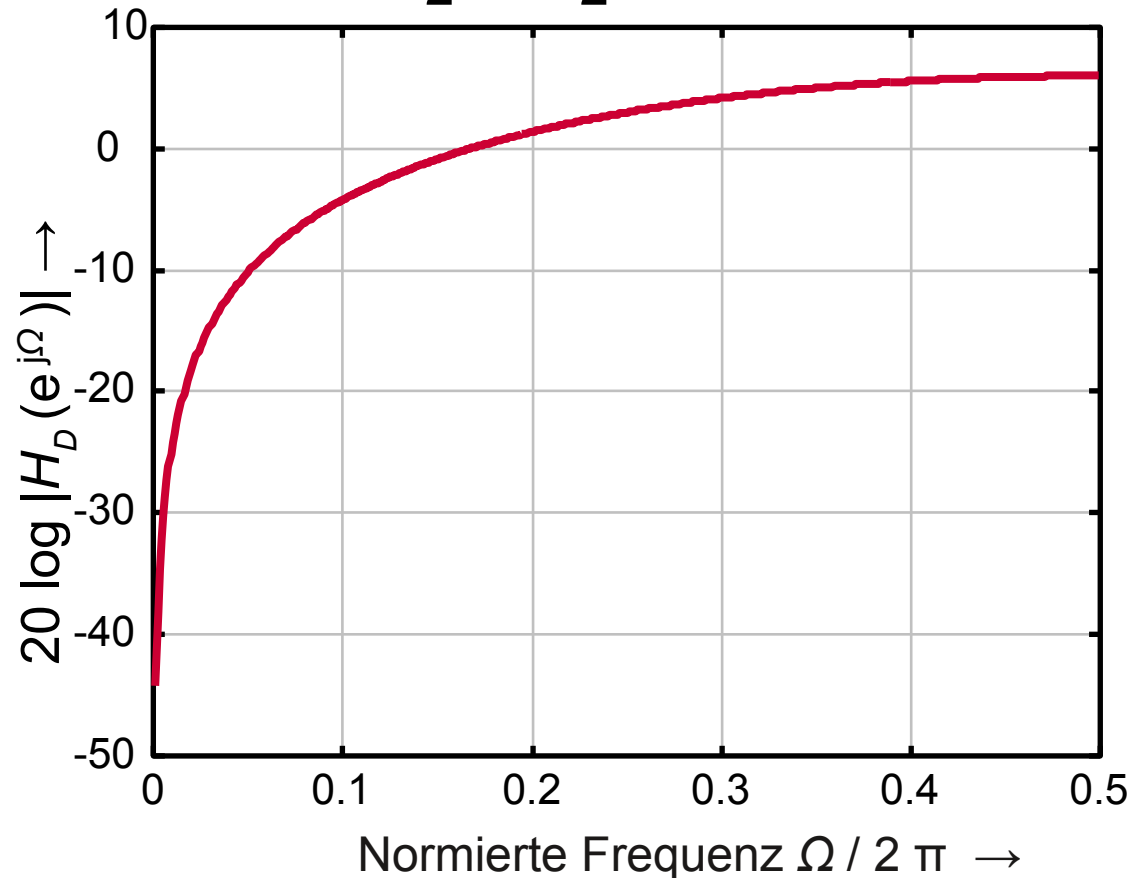
Einfache Filter: Differenzierer



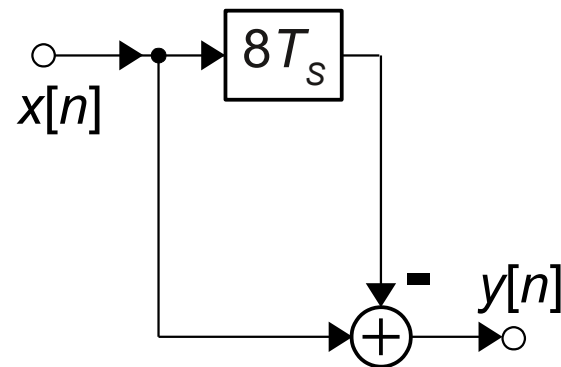
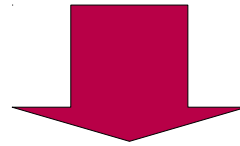
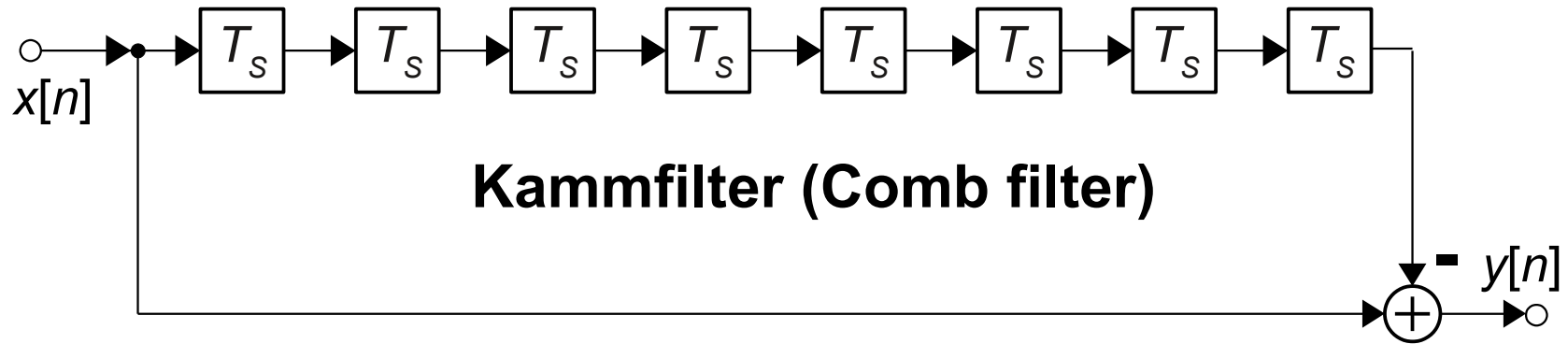
$$H_D(z) = 1 - z^{-1}$$

- $H_D(z = 1)$? $H_D(z = -1)$?
- P/N-Diagramm?
- Stabilität?
- Phasengang?

$$\begin{aligned} H_D(e^{j\Omega}) &= 1 - e^{-j\Omega} = e^{-j\Omega/2} (e^{+j\Omega/2} - e^{-j\Omega/2}) \\ &= 2j e^{-j\Omega/2} \sin(\Omega/2) \approx j\Omega \text{ für } \Omega \ll 1 \\ \varphi_D(\Omega) &= \frac{\pi - \Omega}{2} - \frac{\pi}{2} (1 - \text{sgn}(\sin(\Omega/2))) \end{aligned}$$



Einfache Filter: Kammfilter mit $N = 8$ Delays (1)



$$H_C(z) = 1 - z^{-8}$$

Einfache Filter: Kammfilter mit $N = 8$ Delays (2)



$$H_C(z) = 1 - z^{-N} \Rightarrow$$

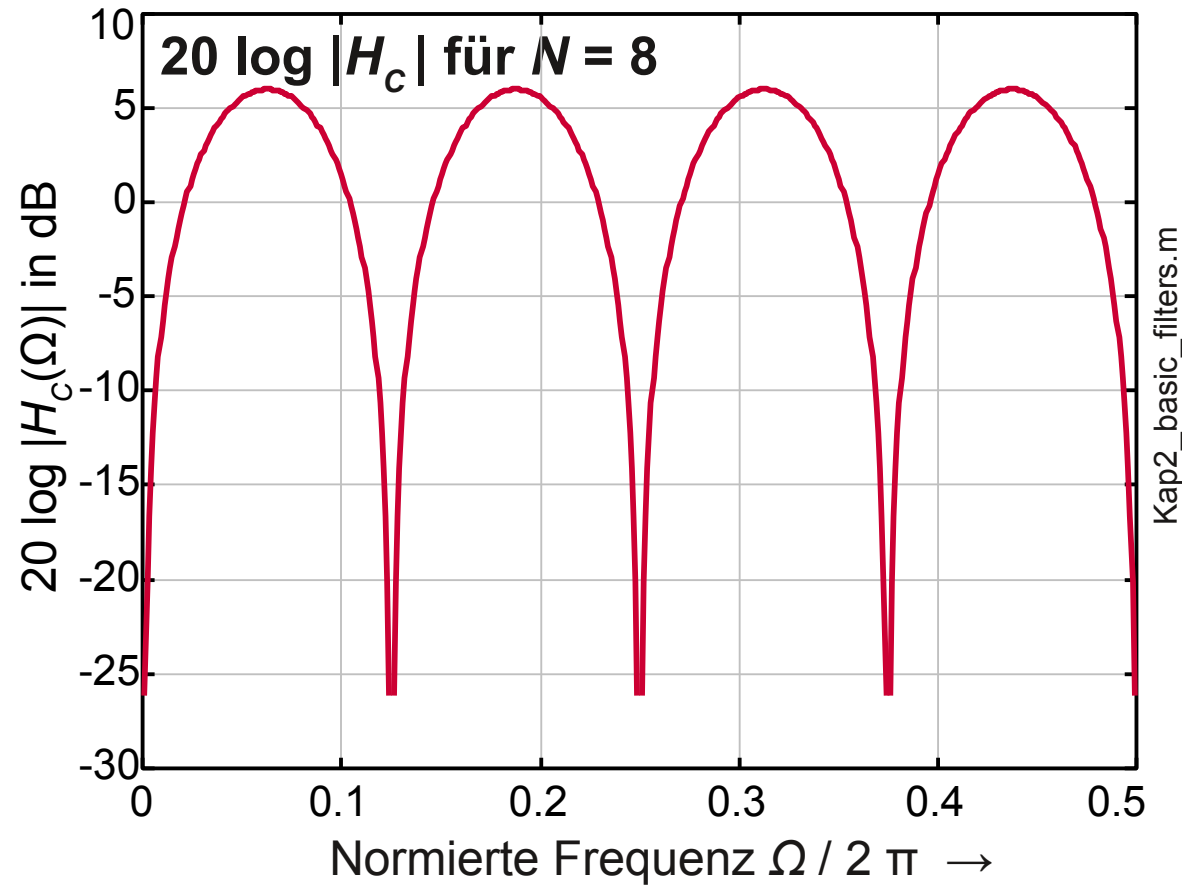
$$H_C(e^{j\Omega}) = 1 - e^{-jN\Omega}$$

$$= e^{-jN\Omega/2} \left(e^{+jN\Omega/2} - e^{-jN\Omega/2} \right)$$

$$= 2j e^{-jN\Omega/2} \sin(N\Omega/2)$$

\Rightarrow **N Nullstellen** in $0 \leq \Omega < 2\pi$

- P/N-Diagramm?
- Stabilität?
- Phasengang?



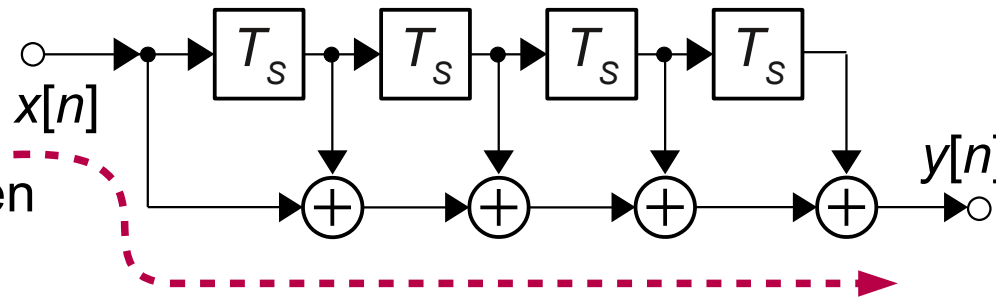
Anwendungen: Unterdrückung von periodischen Spektren (z.B. Fernsehtechnik), Teil des Cascaded Integrator Comb (CIC) Filters

Einfache Filter: Moving Average (MA) Filter



Ordnung $N = 4$:

N Delays,
 $N+1$ Koeffizienten
mit Wert 1



Hardware:

N Register und Addierer,
 N Additionen / Sample

Kritischer Pfad:

N Addierer

Grundlegende Struktur für Digitale Signalverarbeitung:

- Gleiche Problemstellung bei rect-Puls und rect-Fenster !
- Multipliziererlose Filter sind wichtig vor allem für FPGA-Implementierungen !

MA-Filter ($N = 4$): Pol- und Nullstellen



$$h[n] = \{1; 1; 1; 1; 1\} = \sum_{m=0}^4 \delta[n-m]$$

$$\Rightarrow H(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} = \frac{1 - z^{-5}}{1 - z^{-1}} \quad (\text{mit endl. geom. Reihe}) = \frac{z^5 - 1}{(z - 1)z^4}$$

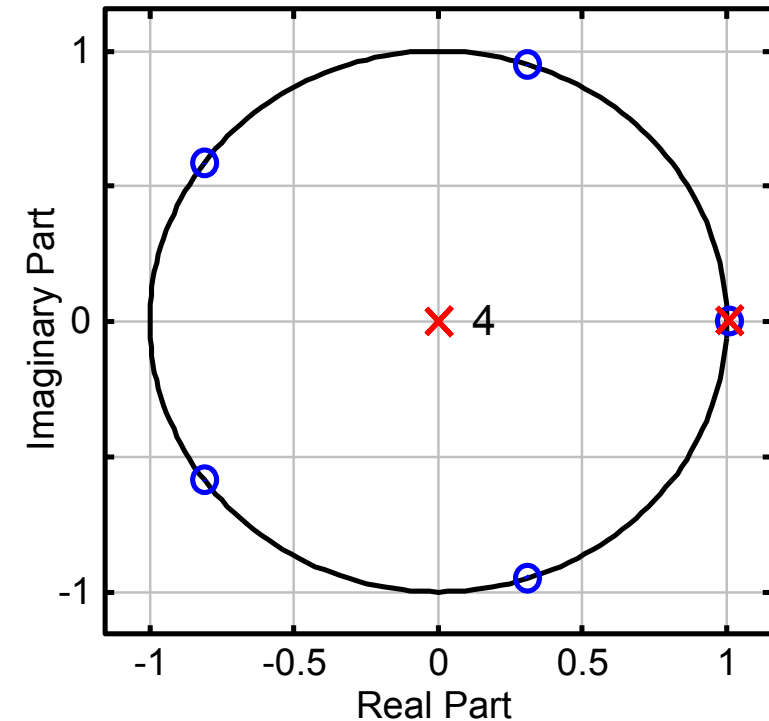
Nullstellen: $z^5 - 1 = 0 \Rightarrow z = \sqrt[5]{1} = \sqrt[5]{e^{j2k\pi}} \Rightarrow z = e^{j2k\pi/5}$ mit $k = \dots, -1, 0, 1, \dots$
 \Rightarrow 5 Nullstellen auf EK verteilt mit $\Delta \varphi = 2\pi/5$

Polstellen: Vier Polstellen bei $z = 0$,
Eine Polstelle und eine Nullstelle
bei $z = 1$ ($f = 0$) **kompensieren sich**

\rightarrow 4 wirksame Nullstellen!

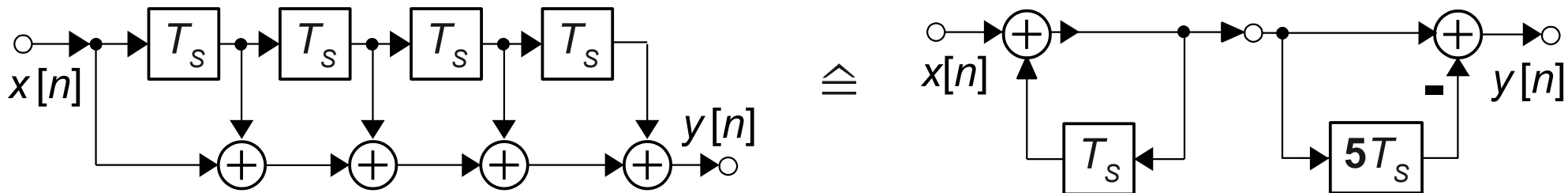
Matlab / Python

```
b = [1, 1, 1, 1, 1];  
zplane(b);
```



- MA-Filter kann auch als Kettenschaltung aus Kammfilter und Integrator realisiert werden
- Pol des Integrators würde Nullstelle bei $z = 1$ aufheben
- Theoretisch gleiche ÜF wie MA-Filter mit weniger Addierern
- Praktisch implementierbar? Reihenfolge Kammfilter / Integrator?

Siehe auch: Cascaded Integrator-Comb-Filter mit Überlauf-Arithmetik (Kap. 9)



MA-Filter ($N = 4$): Frequenzgang (1)



$$H(z = e^{j2\pi f T_s}) \Rightarrow \begin{cases} H(f=0) & = H(z=+1) = 5 \\ H(f=f_s/4) & = H(z=j) = 1 \\ H(f=f_s/2) & = H(z=-1) = 1 \end{cases}$$
$$= \sum_{n=0}^4 e^{-jn\Omega} = e^{-j2\Omega} \sum_{n=-2}^2 e^{-jn\Omega} = \underbrace{e^{-j2\Omega}}_{\text{Phase}} \underbrace{(1 + 2\cos(\Omega) + 2\cos(2\Omega))}_{\text{Amplitude und Vorzeichen}}$$

Berechnung einzelner Werte von $H(f)$ aus Summe von cos-Funktionen leicht möglich, aber unpraktische Form für weitere Umformungen und Analysen.

$$H(z) = \frac{1-z^{-5}}{1-z^{-1}} \Rightarrow H(z=e^{j\Omega}) = \frac{1-e^{-j5\Omega}}{1-e^{-j\Omega}} = \frac{e^{+j5\Omega/2} - e^{-j5\Omega/2}}{e^{+j\Omega/2} - e^{-j\Omega/2}} \cdot \frac{e^{-j5\Omega/2}}{e^{-j\Omega/2}}$$
$$= \underbrace{\frac{\sin(5\Omega/2)}{\sin(\Omega/2)}}_{\text{Amplitude}} \underbrace{e^{-j2\Omega}}_{\text{Phase}} = 5 \underbrace{\frac{\sin(5\Omega/2)}{5\sin(\Omega/2)}}_{:= di_5(\Omega)} \underbrace{e^{-j2\Omega}}_{\text{Phase}}$$

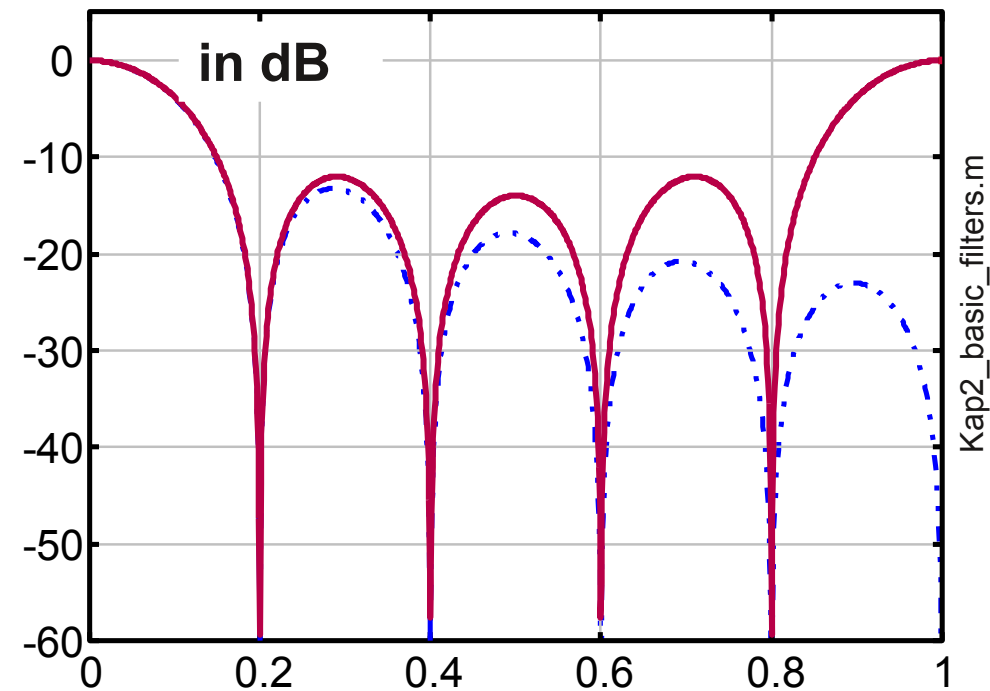
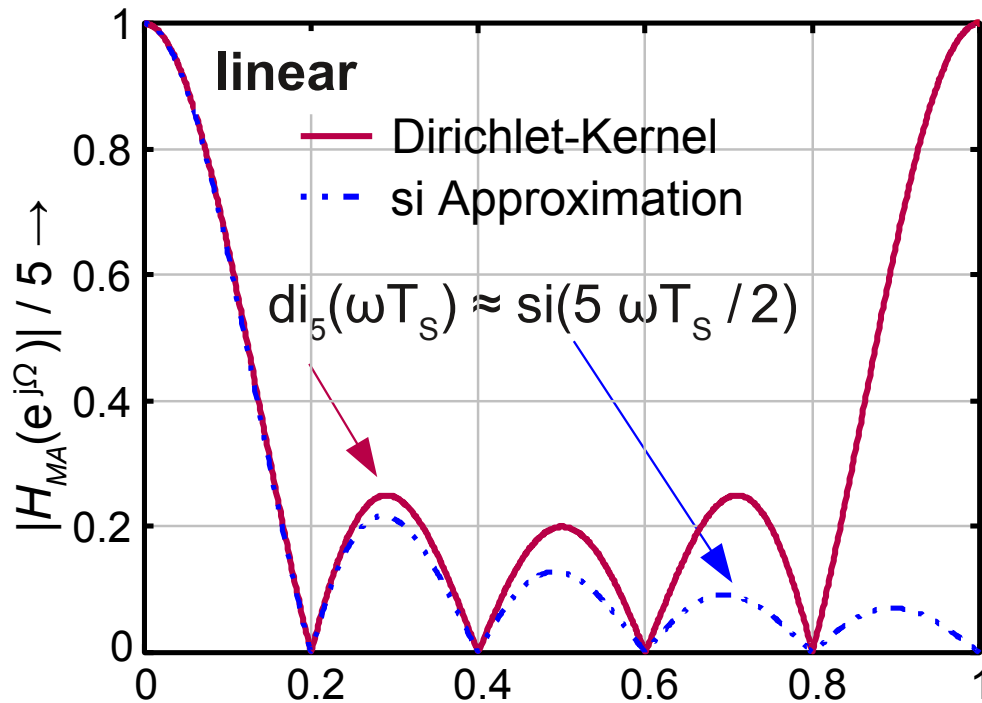
MA-Filter ($N = 4$): Frequenzgang (2)



Beispiel: MA Filter mit $N = 4$ Delays und $N + 1 = 5$ Koeffizienten

➔ $h[n]$ ist rect-Puls der Länge $N + 1 = 5$

➔ $H(z)$ hat $N = 4$ Nullstellen im Bereich $f = 0 \dots f_s$ ($F = 0 \dots 1$)



Normierte Frequenz $\Omega / 2 \pi$ bzw. $F = f / f_s \rightarrow$

MA-Filter der Ordnung N / Dirichlet-Kernel



Spektrum eines MA-Filters / Rechteckpulses mit Länge $N + 1$

$$\begin{aligned} h[n] &= \sum_{m=0}^N \delta[n-m] \Rightarrow H(z) = \sum_{m=0}^N z^{-m} \\ \Rightarrow H(e^{j\Omega}) &= \sum_{n=0}^N e^{-jn\Omega} = \frac{1 - e^{-j(N+1)\Omega}}{1 - e^{-j\Omega}} = \frac{e^{-j(N+1)\Omega/2}}{e^{-j\Omega/2}} \cdot \frac{e^{+j(N+1)\Omega/2} - e^{-j(N+1)\Omega/2}}{e^{+j\Omega/2} - e^{-j\Omega/2}} \\ &= e^{-jN\Omega/2} \cdot \frac{\sin((N+1)\Omega/2)}{\sin(\Omega/2)} = \underbrace{e^{-jN\Omega/2}}_{\text{lineare Phase}} \cdot \underbrace{(N+1)}_{H(f=0)} \cdot \underbrace{\text{di}_{N+1}(\Omega)}_{\text{Amplitudengang}} \end{aligned}$$

mit **Dirichlet-Kernel** $\text{di}_K(\mathbf{x})$ („periodische si-Funktion“):

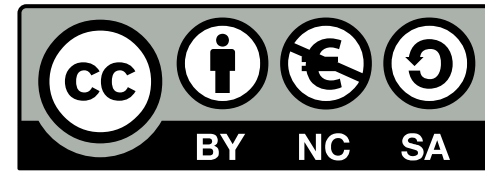
$$\text{di}_K(x) := \frac{\sin(Kx/2)}{K \sin(x/2)} \approx \text{si}(Kx/2) \text{ für } x/2 \ll 1$$

Achtung: Verschiedene Definitionen für di_K ; hier: Matlab $\text{diric}(x, K) \equiv \frac{\sin(Kx/2)}{K \sin(x/2)}$

Berechnung von $\text{di}_K(0)$ über Regel von L'Hospital:

$$\left. \frac{f(x)}{g(x)} \right|_{x \rightarrow x_0} = \left. \frac{f'(x)}{g'(x)} \right|_{x \rightarrow x_0} \Rightarrow \text{di}_K(\Omega = 0) = \left. \frac{K/2 \cos(K\Omega/2)}{K/2 \cos(\Omega/2)} \right|_{\Omega \rightarrow 0} = 1$$

Diese Folien und die zugehörigen Videos sind unter
Creative-Commons-Lizenz **CC-BY-NC-SA 3.0 de** veröffentlicht.



Bei Verwendung dieses Werks müssen Sie auf die entsprechende **CC-Lizenzurkunde** verweisen, in diesem Fall <http://creativecommons.org/licenses/by-nc-sa/3.0/de/> .

Sie müssen ferner die folgenden Angaben machen ("BY", attribution)

- **Author** („Christian Münker“)
- **Titel** („Digitale Signalverarbeitung auf FPGAs“)
- **URL** zu Werk (https://github.com/chipmuenk/dsp_fpga)
und / oder Author (<http://www.chipmuenk.de>)

Außerdem ist die Verwendung auf folgende Weise eingeschränkt:

- Diese Materialien dürfen nur **nicht kommerziell** genutzt werden („NC“, non-commercial).
- Dieses Werk oder Teile daraus dürfen nur **unter gleichen Lizenzbedingungen** weiterverteilt werden („SA“, share alike).

Fragen, Anmerkungen, Anregungen, Bugs, Bierbons bitte an mail@chipmuenk.de.

Ich wünsche viel Erfolg und Spaß (!) mit den Materialien!