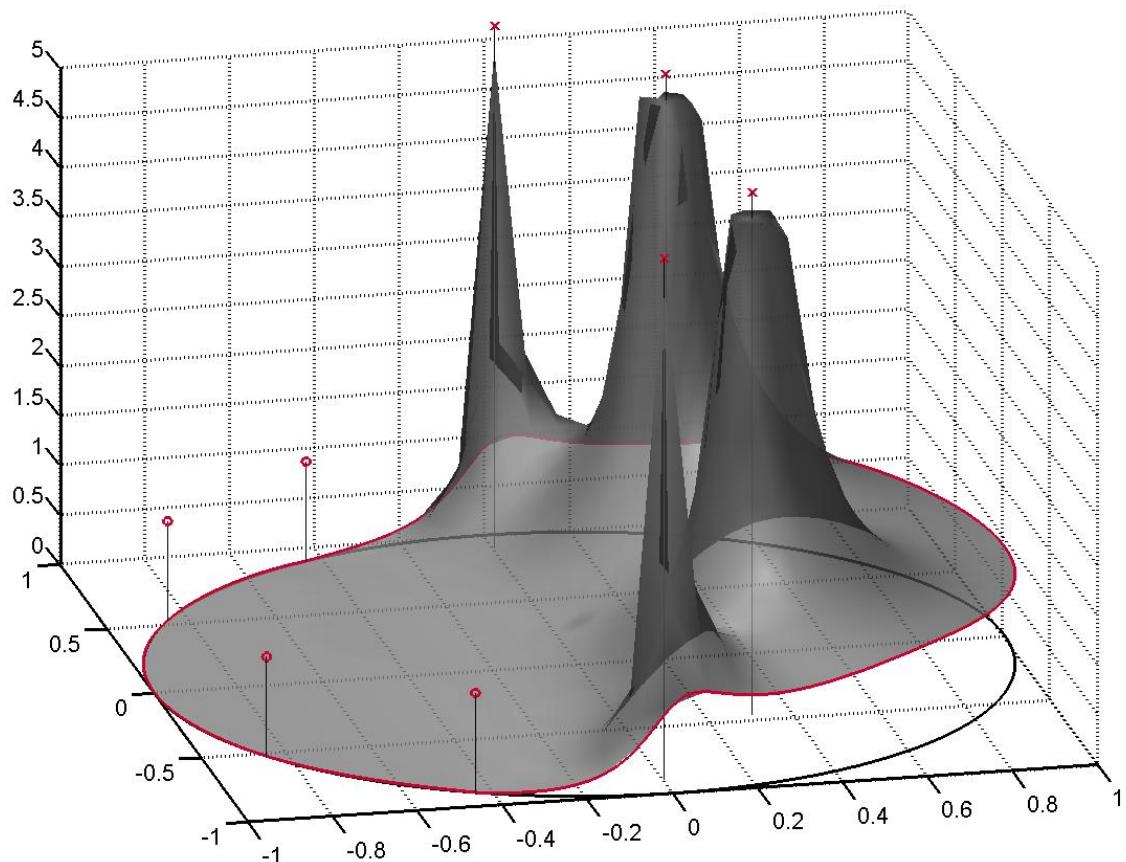


# Übungsaufgaben mit Python-Beispielen zu

## Digitale Signalverarbeitung auf FPGAs

Prof. Dr. Christian Münker



Kurs im Sommersemester 2020 an der Hochschule München

Zuletzt aktualisiert: 17. Juni 2020

[chipmuenk@gmail.de](mailto:chipmuenk@gmail.de)

Übungsaufgaben zu Digitale Signalverarbeitung auf FPGAs

(c) 2008-2020 Prof. Dr. Christian Münker

Überarbeitete Version vom 17. Juni 2020.

Prof. Dr. Christian Münker lehrt digitale Signalverarbeitung und analoge Schaltungstechnik an der Fakultät Elektrotechnik und Informationstechnik der Hochschule München.

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Unbenommen davon gilt für das Gesamtwerk eine Creative Commons Namensnennung – Weitergabe unter gleichen Bedingungen 4.0 International Lizenz (CC BY-SA 4.0):



<https://creativecommons.org/licenses/by-sa/4.0/deed.de>.

- Bei Verwendung dieses Werks müssen Autor, Titel und URL zu Werk und / oder Autor genannt werden und es muss auf die entsprechende CC-Lizenzurkunde verwiesen werden („BY“, attribution):  
„Christian Münker, Übungsaufgaben zu Digitale Signalverarbeitung auf FPGAs, [https://github.com/chipmuenk/dsp\\_fpga](https://github.com/chipmuenk/dsp_fpga), lizenziert unter Creative Commons Lizenz BY-SA 4.0“.
- Dieses Werk oder Teile daraus dürfen nur unter gleichen Lizenzbedingungen weiterverteilt werden („SA“, share alike).

(c) der genialen XKCD-Comics in diesem Skript: Randall Munroe (<https://xkcd.com/>) unter CC BY-NC 2.5 Lizenz (<http://creativecommons.org/licenses/by-nc/2.5/>).

Dieses Dokument kann frei heruntergeladen werden unter [https://github.com/chipmuenk/dsp\\_fpga](https://github.com/chipmuenk/dsp_fpga). Um die L<sup>A</sup>T<sub>E</sub>X- Quelltexte zu bekommen, schicken Sie einfach eine Mail an den Autor.

Diese Unterlagen wurden mit L<sup>A</sup>T<sub>E</sub>X verfasst, weil ...

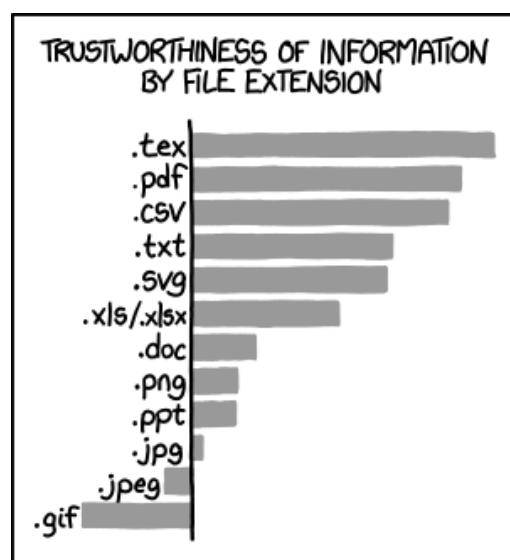


Abb. 1: [<http://xkcd.com/1301/>]



# Inhaltsverzeichnis

<b>Abkürzungs- und Begriffsverzeichnis</b>	<b>xi</b>
<b>Unterlagen und Hinweise zu diesem Kurs</b>	<b>xix</b>
1    Schriftlich (diese Unterlagen) . . . . .	xix
2    InterAKTIV . . . . .	xx
2.1    Jupyter Notebooks . . . . .	xxi
2.2    PYNQ-Board . . . . .	xxii
2.3    Screencasts . . . . .	xxii
2.4    pyFDA . . . . .	xxiii
3    Noch mehr?! . . . . .	xxiii
<b>I Übungsaufgaben</b>	<b>1</b>
<b>1 LTI: LTI-Systeme im Zeitbereich</b>	<b>3</b>
Theorie . . . . .	4
Spezielle Python-Funktionen . . . . .	5
Notebooks und Labs . . . . .	5
1.1    Filterung abgetasteter Signale . . . . .	6
1.2    Faltung mit einfachem FIR-Filter . . . . .	8
1.3    Systemeigenschaften aus Impulsantwort . . . . .	8
1.4    Aufstellung der Differenzengleichung für Biquads aus SFG . . . . .	9
1.5    Kritischer Pfad und Ressourcenverbrauch 1 . . . . .	10
1.6    Kritischer Pfad und Ressourcenverbrauch 2 . . . . .	10
1.7    Transponierte Systeme zu Aufgabe 1.6 . . . . .	10
1.8    Allgemeine Fragen zu FPGAs . . . . .	10
<b>2 LTF: LTI-Systeme im Frequenzbereich</b>	<b>13</b>
Theorie . . . . .	14
Spezielle Python-Funktionen . . . . .	15
Notebooks und Labs . . . . .	16
2.1    Filterung des Sensorsignals im Frequenzbereich . . . . .	16
2.2    Frequenzgang einfacher FIR-Filter . . . . .	17
2.3    Systemfunktion aus Impulsantwort . . . . .	17
2.4    Systemeigenschaften aus Übertragungsfunktion . . . . .	17
2.5    Systemeigenschaften aus Pol-/Nullstellenplan . . . . .	18
2.6    Frequenzgang aus Signalflussdiagramm . . . . .	18
2.7    Einfaches Moving Average Filter . . . . .	19
2.8    Linearphasiges FIR-Filter . . . . .	19
2.9    Allgemeine IIR-Struktur . . . . .	20
2.10    IIR-Filter zweiter Ordnung . . . . .	21
2.11    Allgemeines Moving Average Filter / rect-Fenster . . . . .	22

2.12 Notchfilter . . . . .	22
<b>3 DFT: Diskrete Fouriertransformation und FFT</b>	<b>23</b>
Theorie . . . . .	24
Spezielle Python-Funktionen . . . . .	26
Notebooks und Labs . . . . .	27
3.1 Allgemeine Fragen zu DFT und FFT . . . . .	29
3.2 # Fourierreihe und kohärente DFT . . . . .	29
3.3 Frequenzgang eines MA-Filters . . . . .	30
3.4 Frequenzauflösung der DFT . . . . .	30
3.5 Rechenaufwand für DFT und FFT . . . . .	31
3.6 DFT periodischer Signale mit Python . . . . .	31
3.7 + Kohärente DFT einer Rechteckschwingung . . . . .	32
<b>4 WIN: Fensterung periodischer und stationärer Signale</b>	<b>33</b>
Theorie . . . . .	35
Spezielle Python-Funktionen . . . . .	39
Notebooks und Labs . . . . .	40
4.1 Allgemeine Fragen . . . . .	41
4.2 Fourier-Analyse mit Rechteck-Fensterung . . . . .	41
4.3 DFT mit Hann-Fenster . . . . .	41
4.4 DFT mit Bartlett-Fenster . . . . .	42
<b>5 FIL: Einfache digitale Filter und FIR-Filterentwurf</b>	<b>43</b>
Theorie . . . . .	44
Spezielle Python-Funktionen . . . . .	48
Notebooks und Labs . . . . .	48
5.0 Filterentwurf für Sensorsignal . . . . .	50
5.1 Filterspezifikationen . . . . .	50
5.2 Amplitudengang linearphasiger Filter . . . . .	50
5.3 Maximal-, minimal- und linearphasige Filter . . . . .	51
5.4 FIR Halbbandfilter . . . . .	52
5.5 Filterentwurf mit Python . . . . .	53
5.6 Filtertransformationen . . . . .	53
5.7 Filterimplementierung auf FPGA . . . . .	54
5.8 Verständnisfragen zu Filtern . . . . .	54
<b>6 FIX: Wortlängeneffekte und Fixpoint-Systeme im Zeitbereich</b>	<b>57</b>
Theorie . . . . .	58
Spezielle Python-Funktionen . . . . .	59
Notebooks und Labs . . . . .	60
6.1 Zahlendarstellung im Zweierkomplementformat . . . . .	60
6.2 Analog-Digital-Wandlung eines Drucksensors . . . . .	60
6.3 Moving Average Filter mit endlicher Wortbreite . . . . .	61
6.4 FIR Filter mit Quantisierung . . . . .	62
6.5 + FIR Filter mit Skalierung . . . . .	63
6.6 + Integrator mit endlicher Wortbreite . . . . .	63
6.7 Verständnisfragen zu Filtern mit Quantisierung . . . . .	64
<b>7 NOI: Wortlängeneffekte im Frequenzbereich: Quantisierungsrauschen</b>	<b>67</b>
Spezielle Python-Funktionen . . . . .	71

Notebooks und Labs . . . . .	72
7.1 Quantisierungsrauschen bei Analog-Digital-Wandlung . . . . .	74
7.2 <i>SQNR</i> eines quantisierten Signals . . . . .	74
7.3 Quantisierungsrauschen im FIR-Filter . . . . .	75
7.4 DFT von Breitbandsignalen . . . . .	76
<b>8 SMP: Abtastung und Downsampling</b> . . . . .	<b>77</b>
Spezielle Python-Befehle . . . . .	78
Notebooks und Labs . . . . .	79
8.1 Spektren abgetasteter Signale . . . . .	79
8.2 Analog-Digital-Wandlung mit Oversampling . . . . .	79
8.3 Analog-Digital-Wandlung mit Oversampling und nachfolgender Dezimation . . . . .	79
8.4 Downsampling um den Faktor 2 nach Averaging . . . . .	80
8.5 Effizientes Averaging und Downsampling um den Faktor $N$ . . . . .	81
8.6 Zweistufige Dezimation . . . . .	81
8.7 Multiraten-Tiefpassfilter . . . . .	82
<b>9 INP: Upsampling, Interpolation und Digital-Analog Wandlung</b> . . . . .	<b>85</b>
Theorie . . . . .	86
Spezielle Python-Befehle . . . . .	86
Notebooks und Labs . . . . .	87
9.1 Wiederholspektren und Images . . . . .	88
9.2 Verschiedene Arten der Digital-Analog-Wandlung . . . . .	88
9.3 Ideales Interpolationsfilter . . . . .	91
9.4 Oversampling DAC . . . . .	92
9.5 + Zweistufige Interpolation . . . . .	93
<b>10 SRC: Abtastratenwandlung</b> . . . . .	<b>95</b>
Spezielle Python-Befehle . . . . .	96
Notebooks und Labs . . . . .	98
10.1 Einfache Abtastratenwandlung . . . . .	98
10.2 Ideale Abtastratenwandlung . . . . .	98
10.3 Abtastratenwandlung mit Aliasing im Übergangsbereich . . . . .	100
10.4 Abtastratenwandlung HiRes Audio . . . . .	100
10.5 Abtastratenwandlung CD Audio . . . . .	101
<b>II Musterlösungen</b> . . . . .	<b>103</b>
<b>M1 LTI-Systeme im Zeitbereich</b> . . . . .	<b>105</b>
M1.1 Filterung abgetasteter Signale . . . . .	105
M1.2 Faltung mit einfachem FIR-Filter . . . . .	110
M1.3 Systemeigenschaften aus Impulsantwort . . . . .	112
M1.4 Aufstellung der Differenzengleichung für Biquads aus SFG . . . . .	113
M1.5 Kritischer Pfad und Ressourcenverbrauch 1 . . . . .	114
M1.6 Kritischer Pfad und Ressourcenverbrauch 2 . . . . .	116
M1.7 Transponierte Systeme zu Aufgabe 1.6 . . . . .	117
M1.8 Allgemeine Fragen zu FPGAs . . . . .	117
<b>M2 LTF: LTI-Systeme im Frequenzbereich</b> . . . . .	<b>121</b>
M2.1 Filterung des Sensorsignals im Frequenzbereich . . . . .	121
M2.2 Frequenzgang einfacher FIR-Filter . . . . .	127

M2.3	Systemfunktion aus Impulsantwort . . . . .	130
M2.4	Systemeigenschaften aus Übertragungsfunktion . . . . .	130
M2.5	Systemeigenschaften aus Pol-/Nullstellenplan . . . . .	133
M2.6	Systemeigenschaften aus Signalflussdiagramm . . . . .	133
M2.7	Einfaches Moving Average Filter . . . . .	135
M2.8	Linearphasiges FIR-Filter . . . . .	139
M2.9	Allgemeine IIR-Struktur . . . . .	140
M2.10	IIR-Filter zweiter Ordnung . . . . .	147
M2.11	Moving Average Filter / rect-Fenster . . . . .	150
M2.12	Notchfilter . . . . .	151
<b>M3</b>	<b>DFT: Diskrete Fouriertransformation und FFT</b>	<b>153</b>
M3.1	Allgemeine Fragen zu DFT und FFT . . . . .	153
M3.2	Fourierreihe und kohärente DFT . . . . .	154
M3.3	Frequenzgang eines MA-Filters . . . . .	158
M3.4	Frequenzauflösung der DFT . . . . .	160
M3.5	Rechenaufwand für DFT und FFT . . . . .	160
M3.6	DFT periodischer Signale mit Python . . . . .	161
M3.7	+ Kohärente DFT einer Rechteckschwingung . . . . .	164
<b>M4</b>	<b>WIN: Fensterung periodischer und stationärer Signale</b>	<b>167</b>
M4.1	Allgemeine Fragen . . . . .	167
M4.2	Fourier-Analyse mit Rechteck-Fensterung . . . . .	167
M4.3	DFT mit Hann-Fenster . . . . .	170
M4.4	DFT mit Bartlett-Fenster . . . . .	170
<b>M5</b>	<b>FIL: Einfache digitale Filter und FIR-Filterentwurf</b>	<b>171</b>
M5.1	Filterspezifikationen . . . . .	171
M5.2	Amplitudengang linearphasiger Filter . . . . .	174
M5.3	Maximal-, minimal- und linearphasige Filter . . . . .	176
M5.4	FIR Halbbandfilter . . . . .	183
M5.5	Filterentwurf mit Python . . . . .	186
M5.6	Filtertransformationen . . . . .	186
M5.7	Filterimplementierung auf FPGAs . . . . .	188
M5.8	Verständnisfragen zu Filtern . . . . .	190
<b>M6</b>	<b>FIX: Wortlängeneffekte und Fixpoint-Systeme im Zeitbereich</b>	<b>191</b>
M6.1	Zahlendarstellung im Zweierkomplementformat . . . . .	191
M6.2	Analog-Digital-Wandlung eines Drucksensors . . . . .	193
M6.3	Moving Average Filter mit endlicher Wortbreite . . . . .	194
M6.4	FIR Filter mit Quantisierung . . . . .	195
M6.5	+ FIR Filter mit Skalierung . . . . .	197
M6.6	+ Integrator mit endlicher Wortbreite . . . . .	198
M6.7	Verständnisfragen zu Filtern mit Quantisierung . . . . .	200
<b>M7</b>	<b>NOI: Wortlängeneffekte im Frequenzbereich: Quantisierungsrauschen</b>	<b>203</b>
M7.1	Quantisierungsrauschen Analog-Digital-Wandlung . . . . .	203
M7.2	<i>SQNR</i> eines quantisierten Signals . . . . .	204
M7.3	Quantisierungsrauschen im FIR-Filter . . . . .	204
M7.4	DFT von Breitbandsignalen . . . . .	207
<b>M8</b>	<b>SMP: Abtastung und Downsampling</b>	<b>209</b>

M8.1	Spektren abgetasteter Signale . . . . .	209
M8.2	Analog-Digital-Wandlung mit Oversampling . . . . .	210
M8.3	Analog-Digital-Wandlung mit Oversampling und nachfolgender Dezimation . . . . .	211
M8.4	Downsampling um den Faktor 2 nach Averaging . . . . .	211
M8.5	Effizientes Averaging und Downsampling um den Faktor $N$ . . . . .	212
M8.6	Zweistufige Dezimation . . . . .	213
M8.7	Multiraten-Tiefpassfilter . . . . .	214
<b>M9</b>	<b>INP: Upsampling, Interpolation und Digital-Analog Wandlung</b>	<b>219</b>
M9.1	Wiederholspektren und Images . . . . .	219
M9.2	Verschiedene Arten der Digital-Analog-Wandlung . . . . .	219
M9.3	Ideales Interpolationsfilter . . . . .	222
M9.4	Oversampling DAC . . . . .	223
M9.5	+ Zweistufige Interpolation . . . . .	225
<b>M10</b>	<b>SRC: Abtastratenwandlung</b>	<b>227</b>
M10.1	Einfache Abtastratenwandlung . . . . .	227
M10.2	Ideale Abtastratenwandlung . . . . .	228
M10.3	Abtastratenwandlung mit Aliasing im Übergangsbereich . . . . .	234
M10.4	Abtastratenwandlung HiRes Audio . . . . .	234
M10.5	Abtastratenwandlung CD Audio . . . . .	234
<b>III</b>	<b>Anhang</b>	<b>237</b>
<b>A</b>	<b>Signalflussgraphen (SFG)</b>	<b>239</b>
A.1	Transformationen von SFGs . . . . .	240
A.1.1	Zusammenfassen von seriellen bzw. parallelen Zweigen . . . . .	240
A.1.2	Distributivität („Schieben über Knoten“) . . . . .	241
A.1.3	Transponieren . . . . .	241
<b>B</b>	<b>Wichtige Formeln und Zusammenhänge</b>	<b>243</b>
B.1	Komplexe Ebene . . . . .	243
B.2	Trigonometrische und Eulersche Identitäten . . . . .	243
B.2.1	Trigonometrische Identitäten . . . . .	244
B.2.2	Taylorreihenentwicklung für Sinus und Cosinus Funktionen . . . . .	245
B.3	Exponentialfunktion und Logarithmus . . . . .	245
B.3.1	Allgemeines . . . . .	245
B.3.2	Reihenentwicklung . . . . .	245
B.4	Quadratische Gleichung . . . . .	245
B.5	Summenformeln . . . . .	246
B.5.1	Unendliche geometrische Reihe . . . . .	246
B.5.2	Endliche geometrische Reihe . . . . .	246
B.5.3	Dirichlet-Kern . . . . .	247
B.5.4	Verwandte Reihen . . . . .	247
B.5.5	Einfache Reihen . . . . .	248
B.6	Reihenentwicklung . . . . .	248
B.7	Ein paar Integrale . . . . .	248
<b>C</b>	<b>Fourier-Transformation</b>	<b>249</b>
C.1	Fourier-Transformation kontinuierlicher Signale . . . . .	249
C.2	Fourier-Transformation diskreter Signale mit unendlicher Länge (DTFT) . . . . .	251

---

C.2.1	Definition . . . . .	251
C.2.2	Eigenschaften . . . . .	251
C.3	Fourier-Transformation diskreter Signale der Länge $N$ (DFT) . . . . .	253
C.3.1	Definition . . . . .	253
C.3.2	Eigenschaften . . . . .	253
<b>D</b>	<b><math>z</math>-Transformation</b>	<b>255</b>
	<b>Abbildungsverzeichnis</b>	<b>256</b>
	<b>Tabellenverzeichnis</b>	<b>261</b>
	<b>Listings</b>	<b>263</b>
	<b>Literaturverzeichnis</b>	<b>263</b>

## Abkürzungs- und Begriffsverzeichnis

---

Begriff	Erklärung
<b>ADC</b>	Analog-to-Digital-Converter
<b>Aliasing</b>	Bei der Abtastung eines Signals mit $f_S$ werden Spektralanteile oberhalb $f_S/2$ ins → Basisband abgebildet, diesen Vorgang nennt man A. Die hierdurch entstandenen Verfälschungen lassen sich im Allgemeinen nicht rückgängig machen; vor Abtastung oder Reduktion der Abtastrate muss daher die Bandbreite des Signals mit einem <i>Anti-Aliasingfilter</i> auf $f_S/2$ begrenzt werden, damit das $\rightarrow$ Nyquistkriterium erfüllt ist.
<b>Basisband</b>	Mit $f_S$ abgetastete Signale haben ein Spektrum, das periodisch mit $f_S$ ist (-> Wiederholspektren). Der Frequenzbereich $-f_S/2 \dots f_S/2$ (oder, seltener, $0 \dots f_S$ ) wird Basisband genannt. Das Spektrum reellwertiger Signale ist symmetrisch zu $f = 0$ , so dass hier bereits das positive Basisband $0 \dots f_S/2$ die vollständige Information enthält.
<b>BP(-Filter)</b>	Bandpass(-Filter)
<b>BS</b>	Bandsperre
<b>CIC</b>	Cascaded Integrator Comb (Filter)
<b>DAC</b>	Digital-to-Analog Converter
<b>Dezimation</b>	Verringerung der Abtastrate durch Wegwerfen von Samples (→ Downsampling) mit vorheriger Filterung zur Vermeidung von → Aliasing. In der Literatur werden beide Begriffe oft synonym verwendet.
<b>Downsampling</b>	Verringerung der Abtastrate, indem $R - 1$ von $R$ Abtastwerten weggeworfen und nur jedes $R$ -te Sample übernommen wird (→ Dezimation).
<b>DSP</b>	Digital Signal Processing oder Processor
<b>DF</b>	→ Direktform [Filter]
<b>DFT</b>	Diskrete Fourier Transformation
<b>Direktform-Filter</b>	Eine Filterstruktur, deren Koeffizienten <i>direkt</i> aus der Übertragungsfunktion $H(z)$ oder der Differenzengleichung abgeleitet werden können.
<b>DGL</b>	Differenzialgleichung
<b>DZGL</b>	Differenzengleichung
<b>DTFT</b>	Discrete-Time Fourier Transform; Fouriertransformierte von unendlich ausgedehnten, abgetasteten Zeitsignalen. Aufgrund der unendlichen Anzahl von Abtastpunkten hat die DTFT nur theoretische Bedeutung, auf Computern kann nur eine DFT oder FFT mit einer endlichen Punktzahl durchgeführt werden.

---

Begriff	Erklärung
<b>ENBW</b>	Equivalent Noise Bandwidth: Die Bandbreite eines hypothetischen idealen Tiefpassfilters, das die gleiche Rauschleistung am Ausgang liefert wie das betrachtete System. Die ENBW von gängigen Filtern und Fenstern sind in entsprechenden Fachbüchern tabelliert.
<b>FFT</b>	Fast Fourier Transform: Verschiedene Algorithmen zur effizienten Berechnung der → DFT. Die effizientesten Algorithmen basieren auf der Zerlegung der DFT in möglichst kleine Teilstücke und arbeiten daher am besten wenn die FFT eine Länge von $N_{FFT} = 2^m$ (Radix-2) bzw. $N_{FFT} = 4^m$ (Radix-4) hat.
<b>FIR-Filter</b>	Finite Impulse Response-Filter; ein Filter mit endlicher Impulsantwort, d.h. in der Regel ein nicht-rekursives Filter (Ausnahme: CIC-Filter mit rekursiver Struktur aber endlicher Impulsantwort)
<b>FPGA</b>	Field Programmable Gate Array: Integrierte Bausteine mit programmierbaren Logikverknüpfungen. Im Gegensatz zu CPLDs (Complex Programmable Logic Devices) erlauben FPGAs die Realisierung wesentlich umfangreicherer und flexiblerer Systeme zur Datenverarbeitung.
<b>GPIO</b>	General Purpose I/Os; I/O Pins meist für geringere Datenraten (im Gegensatz zu High-Speed Memory Interfaces etc.)
<b>HDL</b>	Hardware Description Language; Oberbegriff für Sprachen wie VHDL und Verilog, mit denen man das Verhalten von digitaler Hardware beschreiben kann.
<b>HP(-Filter)</b>	Hochpass(-Filter)
<b>Interpolation</b>	Allgemein Abschätzung von Werten zwischen Datenpunkten (z.B. Messwerte), wird hier verwendet für Ermittlung von zusätzlichen Abtastwerten bei Erhöhung der Abtastrate.
<b>IIR-Filter</b>	Infinite Impulse Response Filter; ein Filter mit unendlicher Impulsantwort. Das ist nur möglich mit einem rekursiven Filter.
<b>Image</b>	Wenn die Abtastrate durch Nullenstopfen von $f_S$ auf $1/f_S$ erhöht wird, liegen im neuen Basisband $-f_S/2 \dots f_S/2$ Kopien des alten Basisbands $-f_S/2 \dots f_S/2$ , die sich im Gegensatz zu -> Wiederholspektren unabhängig beeinflussen lassen.
<b>Kritischer Pfad</b>	In einem synchron getakteten (Teil-)System (= alle Register bekommen den gleichen Takt) müssen beim Eintreffen der nächsten Taktflanke die Daten an allen Registereingängen stabil anliegen. Der <i>kritische Pfad</i> mit der längsten Verzögerungszeit $\tau_{max}$ von einem Registerausgang zu einem Registereingang bestimmt daher die maximal zulässige Taktfrequenz des gesamten Systems, $f_{S,max} = 1/\tau_{max}$ . Die Verzögerungszeit setzt sich zusammen aus allen arithmetischen Operationen (Additionen, Multiplikationen, ...) im jeweiligen Pfad. Achtung: Die Werte am Eingang $x[n]$ werden von einem Registerausgang außerhalb des betrachteten Teilsystems geliefert, ebenso werden die Daten am Ausgang $y[n]$ von einem weiteren Registereingang erwartet. Auch diese Pfade müssen daher berücksichtigt werden!

---

Begriff	Erklärung
<b>Leistung</b>	Bei nachrichtentechnischen Betrachtungen werden Leistungen oft in $V^2$ angegeben, was physikalisch natürlich nicht korrekt ist. Diese schlampige Nomenklatur kann man wie folgt rechtfertigen: Stellt man sich vor, dass das Signal z.B. an einem $50 \Omega$ Abschlusswiderstand abfällt, erhält man die zugehörige „physikalische Leistung“. Meist geht es bei nachrichtentechnischen Betrachtungen aber sowieso um Verstärkungs- und Signal-Rauschleistungsverhältnisse, dann fallen (gleiche) Impedanzen aus der Rechnung heraus.
<b>LHP</b>	Left Half-Plane; linke Halbebene der $s$ -Ebene
<b>LTI-System</b>	Linear Time-Invariant System; Linear: System, bei dem eine Skalierung der Eingangsgrößen zur gleichen Skalierung der Ausgangsgrößen führt (Gegenbeispiele: Quadrierer, Logarithmierer, Begrenzer). Zeitinvariant: System, bei dem eine Verschiebung / Verzögerung der Eingangsgröße zu genau der gleichen Verschiebung des Ausgangssystems führt. Gegenbeispiel: Downampler.
<b>MIO</b>	Multipurpose IO, vielseitig rekonfigurierbare IO-Pins
<b>MIMO</b>	Multiple-Input, Multiple-Output [System]: System mit mehreren Eingangs- und / oder Ausgangsgrößen. Beispiele: Mehrgrößenregelung, WLAN-System mit mehreren Antennen.
<b>NENBW</b>	Normalized Equivalent Noise Bandwidth, -> ENBW eines Fensters, relativ zu der eines Rechteckfensters.
<b>Notch-Filter</b>	Sehr schmalbandige Bandsperre (notch: engl. für Kerbe), um typischerweise eine bestimmte Störfrequenz (z.B. 50 Hz) zu unterdrücken.
<b>Nutz-band(breite)</b>	Der Frequenzbereich des Signals, in dem die Information enthalten ist.
<b>Nyquist-Frequenz</b>	Halbe Abtastfrequenz
<b>Nyquist-Kriterium</b>	Wenn die Bandbreite eines Signals kleiner ist als die halbe Abtastfrequenz, kann das Signal aus der abgetasteten Darstellung fehlerfrei rekonstruiert werden. Das Nyquist-Kriterium ist dann erfüllt, andernfalls tritt → Aliasing auf.
<b>Nyquist-Wandler</b>	→ ADC oder → DAC, dessen Nutzband im Gegensatz zu einem -> Oversampling-Wandler nur bis zur → Nyquist-Frequenz geht.
<b>Nyquist-Zone</b>	Ein Teil des → Wiederholspektrums von $\pm[k f_S/2 \dots (k+1)f_S/2]$
<b>Rekursivfilter</b>	Filter mit Rückkopplung (im Gegensatz zum -> Transversalfilter). Dadurch haben R. meist eine unendlich ausgedehnte Impulsantwort und werden auch → IIR-Filter genannt.
<b>Out-of-Band Noise</b>	Rauschen außerhalb des Nutzbands (aber innerhalb des Basisbands, das gibt es nur bei → Oversampling). Out-of-Band Noise kann herausgefiltert werden, ohne das Nutzband zu beeinflussen.
<b>Oversampling</b>	Überabtastung; ein Signal mit Bandbreite $f_N$ , das mit $2f_S > f_N$ abgetastet wurde. Durch digitale Filterung kann das → Out-of-Band Noise entfernt und so das -> SNR des Signals verbessert werden.

---

Begriff	Erklärung
<b>Oversampling Wandler</b>	-> ADC oder -> DAC, der mit einer (meist deutlich) höheren Abtastfrequenz arbeitet, als vom -> Nyquist-Kriterium gefordert.
<b>PL</b>	→ Programmable Logic
<b>Processing Subsystem</b>	Der Teil des ZYNQ-SOCs, der den bzw. die ARM-Cores enthält und die konfigurierbaren IOs.
<b>PS</b>	→ Processing Subsystem
<b>Programmable Logic</b>	Der Teil des ZYNQ-SOCs mit dem FPGA Fabric, also der programmierbaren Logik
<b>pyFDA</b>	Python Filter Design and Analysis, interaktives Tool zum Design zeitdiskreter Filter und für kleine Experimente zur digitalen Signalverarbeitung ( <a href="https://github.com/chipmuenk/pyfda">github.com/chipmuenk/pyfda</a> ).
<b>Quantisierung</b>	Umwandlung eines wertkontinuierlichen in ein werdiskretes (= quantisiertes) Signal. Dieser Vorgang ist nichtlinear und mit einem Informationsverlust verbunden. Wird die Wortbreite eines quantisierten Signals reduziert, spricht man von -> Requantisierung.
<b>Requantisierung</b>	Reduktion der Wortbreite eines quantisierten Signals. Wie bei der -> Quantisierung ist dieser Vorgang nichtlinear und mit einem Informationsverlust verbunden.
<b>RHP</b>	Right Half-Plane; rechte Halbebene der $s$ -Ebene
<b>SFDR</b>	Spurious-Free Dynamic Range; der Dynamikbereich zwischen stärkster Störlinie und Nutzsignal im Spektrum
<b>SFG</b>	→ Signalflussgraph oder Signal Flow Graph
<b>SISO</b>	Single-Input, Single-Output [System]
<b>SNR</b>	Signal-To-Noise Ratio; Verhältnis von Signalleistung und gesamter Störleistung
<b>SOS</b>	Second-Order Section; Teilfilter zweiter Ordnung. Durch die Aufteilung von IIR-Filters höherer Ordnung in Teilfilter zweiter Ordnung werden die Filter deutlich robuster gegen Rundungsfehler.
<b>Signalflussgraph</b>	(SFG); Graphische Darstellung des Informationsflusses in einem System. Bei digitalen Systemen (im Gegensatz zu analogen Schaltungen) entspricht der SFG genau dem Blockschaltbild.
<b>TP(-Filter)</b>	Tiefpass(-Filter)
<b>transponieren</b>	Transformation eines -> SFGs in einen anderen SFG mit gleicher Übertragungsfunktion durch Vertauschen von Ein- und Ausgang, Umdrehen sämtlicher Signalflussrichtungen und Vertauschen von Summations- und Abzweigungsknoten.
<b>Transversalfilter</b>	Filter ohne -> rekursiven Anteil. Die Impulsantwort eines T. hat somit eine endliche Länge, es wird daher auch -> FIR-Filter genannt.
<b>WDF</b>	Wave Digital oder Wellen-Digitalfilter

---

Begriff	Erklärung
<b>Wiederhol-</b> <b>spektren</b>	Mit $f_S$ abgetastete Signale haben Spektren, die periodisch mit $f_S$ sind, d.h. das -> Basisbandspektrum wiederholt sich im Abstand von $f_S$ . Die Wiederholungen lassen sich im Gegensatz zu -> Images nicht unabhängig von einander beeinflussen!
<b>Zero Packing</b>	Nullenstopfen; Auffüllen der fehlenden Zwischenwerte mit Nullen bei Erhöhung der Abtastrate
<b>Zero Padding</b>	Anhängen von Nullen an eine Datensequenz um z.B. auf $2^m$ Werte zu kommen und damit eine schnelle Radix-2 FFT zu ermöglichen. Außerdem erhält man mit einer längeren DFT auch eine feiner aufgelöste Darstellung (!) des Spektrums.
<b>ZOH</b>	Zero-Order-Hold; Bei Abtastung wird der letzte Wert gehalten (zeitkontinuierliches Eingangssignal) bzw. wiederholt (zeitdiskretes Eingangssignal). Bei analogen Schaltungen auch als Sample & Hold bekannt.

---

Formelzeichen	Bedeutung	Einheit
<b>A</b>	Verstärkung; Dämpfung	1
<b>A<sub>DB</sub></b>	log. Verstärkung oder Welligkeit im Durchlassband	1
<b>A<sub>SB</sub></b>	minimale Dämpfung im Sperrband	1
<b>B</b>	Signalbandbreite	Hz
<b>B<sub>N</sub></b>	Rauschbandbreite	Hz
<b>E{·}</b>	Statistischer Erwartungswert	— <sup>1</sup>
<b>F</b>	Normierte Frequenz, $F = fT_S$	1
<b>F<sub>DB</sub></b>	Normierte Eckfrequenz des Durchlassbands	1
<b>F<sub>SB</sub></b>	Normierte Eckfrequenz des Sperrbands	1
<b>I</b>	Interpolationsfaktor, d.h. Faktor, um den die Abtastrate erhöht wird	1
<b>L</b>	Loop gain, Schleifenverstärkung	1
<b>M</b>	Anzahl (Koeffizienten, Taps); Ordnung	1
<b>N</b>	Anzahl (Nullstellen); Ordnung	1
<b>N</b>	Rauschleistung	W bzw. V <sup>2</sup>
<b>N<sub>FFT</sub></b>	Anzahl der FFT bzw. DFT-Stützstellen	1
<b>N<sub>Q</sub></b>	Quantisierungsrauschleistung	W, V <sup>2</sup> , etc. <sup>1</sup>
<b>N'<sub>Q</sub></b>	Quantisierungsrauschleistungsdichte	W/Hz etc. (s.o.)
<b>P</b>	(Nachrichtentechnische) Leistung	W bzw. V <sup>2</sup>
<b>R</b>	Dezimationsfaktor; Faktor, um den die Abtastrate reduziert wird	1
<b>S</b>	Signalleistung	W bzw. V <sup>2</sup>
<b>T<sub>S</sub></b>	Abtastperiode, $T_S = 1/f_S$	s
<b>W</b>	Wortbreite	(bits)
<b>WF</b>	Fractional Part der Wortbreite	s.o.
<b>WI</b>	Integer Part der Wortbreite	s.o.
<b>c<sub>Q</sub></b>	Codewert eines Quantisierers	1
<b>di<sub>N</sub>(x)</b>	Dirichlet-Kernel (periodische si-Funktion); $di_N(x) := \sin(Nx/2)/N \sin(x/2)$	1
<b>e<sub>N</sub></b>	Rauschleistung	W bzw. V <sup>2</sup>
<b>f</b>	Frequenz	Hz
<b>f<sub>S</sub></b>	Abtastfrequenz, $f_S = 1/T_S$	Hz

Formelzeichen	Bedeutung	Einheit
$f_{DB}$	Eckfrequenz des Durchlassbands	Hz
$f_{SB}$	Eckfrequenz des Sperrbands	Hz
$q$	LSB, Quantisierungsschritt	$\text{--}^1$
$s$	Komplexe Frequenz; $s = \sigma + j\omega$	rad/s
$\text{si}(x)$	si- oder sinc-Funktion; $\text{si}(x) := \sin(x)/x$	1
$x$	Eingangsgröße, Signal	$\text{--}^1$
$x_Q$	quantisierte Eingangsgröße; $x_Q = qc_Q$	$\text{--}^1$
$z$	Komplexe Frequenz; $z = e^{sT_S}$	rad/s
$z_0$	Nullstelle	rad/s
$z_\infty$	Polstelle	rad/s
$\Delta$	Differenz; Determinante oder charakteristische Gleichung eines Graphen/Systems	$\text{--}$
$\epsilon[n]$	Quantisierungsfehler; $\epsilon[n] = x_Q[n] - x[n]$	$\text{--}^1$
$\Omega$	Normierte Kreisfrequenz; $\Omega = 2\pi f T_S$	1
$\omega$	Kreisfrequenz; $\omega = 2\pi f$	$\text{s}^{-1}$

<sup>1</sup>Abhängig von der Dimension des Signals



# Unterlagen und Hinweise zu diesem Kurs

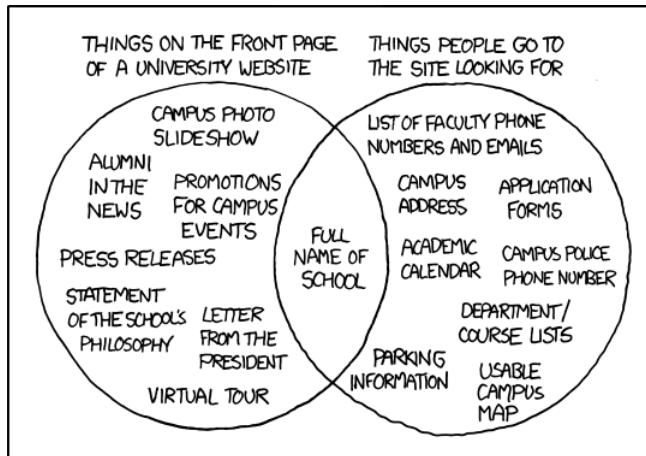


Abb. 1: Was Sie in diesen Unterlagen finden [<https://xkcd.com/773/>]

[www.pyfda.org](http://www.pyfda.org): Links zu den meisten hier erwähnten Ressourcen.

Da die Unterlagen recht umfangreich sind (manche sagen furchteinflößend . . . ), möchte ich Ihnen kurz den Aufbau beschreiben.

Dieser Kurs lebt von Ihrem eigenen Engagement: Zu jeder Lehreinheit gibt es unterschiedliches Lernmaterial (Übung, Text, Screencast, interaktives Notebook . . . ), die Sie zu Hause vorbereiten. An der Hochschule gibt es einen Impulsvortrag, wir besprechen gemeinsam Ihre Fragen, rechnen Übungsaufgaben und arbeiten mit Jupyter Notebooks und dem FPGA-Board, das sie erhalten.

## 1 Schriftlich (diese Unterlagen)

**Skript** im eigentlichen Sinne gibt es nicht, am Anfang der meisten Kapitel finden Sie aber eine knappe Zusammenfassung der wichtigsten Formeln und Zusammenhänge. Außerdem gibt es zu jedem Kapitel Jupyter Notebooks (s.u.), die die wichtigsten Inhalte des Kapitels aufgreifen und um Simulationen ergänzen.

**Themen:** Die ersten drei Kapitel sind überwiegend eine Wiederholung der Grundlagen der digitalen Signalverarbeitung, ergänzt um ein paar FPGA-Themen:

**Kap. 1 LTI:** LTI-Systeme im Zeitbereich (Signalflussgraph, Impulsantwort  $h[n]$ , Differenzengleichung und Systemfunktion  $H(z)$ ). Außerdem geht es um kritischen Pfad und Hardwareimplementierung von einfachen Systemen.

**Kap. 2 LTF:** LTI-Systeme im Frequenzbereich: Pole, Nullstellen, Betrags- und Phasengang, Gruppenlaufzeit und Stabilität, Kausalität und Linearphasigkeit.

**Kap. 3 DFT:** Diskrete Fourier-Transformation. Mit der DFT eines periodischen, abgetasteten Signals über eine ganzzahlige Anzahl Perioden kann man die Fourierreihe eines zeitkontinuierlichen Signals mit Hilfe eines Computers bestimmen (innerhalb der Nyquist-Bandbreite).

Der nächste Abschnitt beschäftigt sich mit Kernthemen der digitalen Signalverarbeitung: Es geht zum einen um die Bestimmung / Abschätzung des Spektrums von nicht-kohärent abgetasteten Signalen und um den Entwurf von zeiddiskreten Filtern. Mit dem pyFDA-Tool können Sie eigene Experimente zu beiden Themen machen.

**Kap. 4 WIN:** Fensterung und Zero-Padding zur Verringerung des Leckeffekts bei möglichst guter Frequenzauflösung

**Kap. 5 FIL:** Einfache digitale Filter und Entwurf von FIR-Filters

Die Effekte von begrenzter Wortlänge und Quantisierung werden in den folgenden beiden Kapiteln behandelt:

**Kap. 6 FIX:** Wortlängeneffekte und Fixpoint-Systeme im *Zeitbereich* (Quantisierung und Requantisierung, Fixpointformate und -arithmetik, Filter mit quantisierten Koeffizienten)

**Kap. 7 NOI:** Wortlängeneffekte im *Frequenzbereich* (Quantisierungsrauschen, *SQNR*)

Die nächsten drei Kapitel zeigen, welche Möglichkeiten die Änderung der Abtastrate bringt und die daraus resultierenden Effekte:

**Kap. 8 SMP:** Abtastung und Downsampling, also die Verringerung der Abtastrate

**Kap. 9 INP:** Upsampling, Interpolation und Digital-Analog-Wandlung, also die Erhöhung der Abtastrate

**Kap. 10 SRC:** Synchrone Abtastratenwandlung (Sample-Rate Conversion); die Änderung der Abtastrate um nicht-ganzzahlige (aber niedrig-rationale) Verhältnisse.

**Übungen:** Im Teil I werden zunächst für jedes Kapitel ein paar grundlegende Begriffe und Formeln sowie Python - Kommandos erläutert. Übungen, die mit (#) gekennzeichnet sind, gehen detailliert auf die Grundlagen ein, werden aber in dieser Form nicht in der Prüfung abgefragt. Übungen mit einem Plus (+) sind für Interessierte.

Die Musterlösungen in Teil II sind sehr ausführlich und dienen gleichzeitig als „Skriptersatz“.

**Anhang:** Hier (Teil III) finden Sie u.a. Tabellen zu Fourier-, Laplace und z-Transformation - diese Tabellen brauchen Sie nur für einzelne Aufgaben, aber nicht in der Prüfung.

Fehler habe ich absichtlich in den Unterlagen versteckt, damit Sie wachsam bleiben und mir nicht alles glauben ; - )

**Wenn Ihnen Fehler auffallen (auch Rechtschreibfehler, Inkonsistenzen, fehlende Bezüge ...), sagen Sie mir bitte Bescheid - ich bezahle in Schokolade!**

## 2 InterAKTIV

Nutzen Sie die Möglichkeiten, selbst aktiv zu werden - Sie bekommen ein besseres Verständnis für DSV, mehr Likes auf Instagram und ein glänzenderes Fell:

**Jupyter Notebooks:** Ohne Software-Installation arbeiten und simulieren Sie im Browser mit interaktiven Jupyter Notebooks (basierend auf Python), von die Sie von [https://github.com/chipmuenk/dsp\\_fpga](https://github.com/chipmuenk/dsp_fpga) bekommen und können sofort loslegen.

**PYNQ-Board:** Jede/r bekommt ein PYNQ FPGA Evaluationsboard, mit dem wir gemeinsam arbeiten.

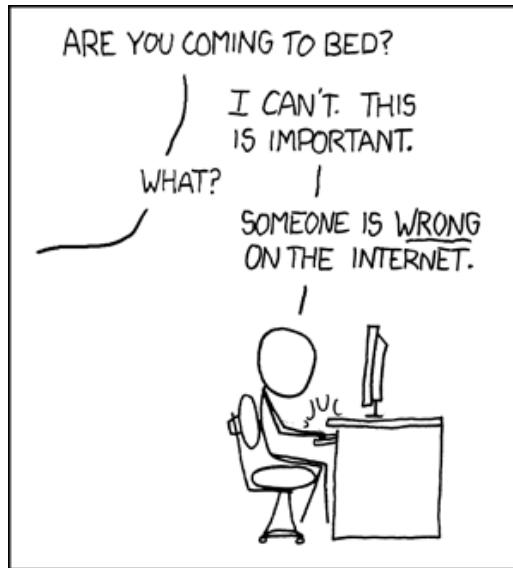


Abb. 2: XKCD: Fehler finden und ganz für sich behalten ist voll egoistisch! [<https://xkcd.com/386/>]

**Vivado:** Mit der Xilinx FPGA-Entwicklungsumgebung entwerfen, simulieren und implementieren Sie FPGA-Designs, die Sie auf Ihr PYNQ-Board laden und dort testen.

**Screencasts** zum Kurs finden Sie in der Playlist **DSP auf FPGAs** auf meinem YouTube Kanal <http://www.youtube.com/c/ChristianMünker>.

**pyFDA:** Mit meiner Software **pyFDA** können Sie bequem zeitdiskrete Systeme im Zeit- und Frequenzbereich analysieren und Filter entwerfen.

## 2.1 Jupyter Notebooks

Notebooks sind interaktive Dokumente, die Text, Formeln, Grafiken, Medien und ausführbaren Code enthalten können (Abb. 3). Detaillierte Erklärungen zum Umgang mit Python, Notebooks und PYNQ in diesem Kurs finden Sie im separaten PDF *Handreichungen zur Aufzucht und Pflege von Python in außergewöhnlichen Lebensräumen: PYNQ und Jupyter Notebooks* (Moodle und Magenta Cloud).

Notebooks werden im Browser bearbeitet, je nach Arbeitsweise müssen Sie nicht einmal etwas auf Ihrem Notebook / Tablet installieren:

**Microsoft Azure Notebooks:** Microsofts kostenloser Notebookserver in der Cloud (<https://notebooks.azure.com/>) genügt für unsere Bedürfnisse. Sie können sich mit Ihrem Microsoft Account anmelden, meine Notebooks von GitHub ([https://github.com/chipmuenk/dsp\\_fpga](https://github.com/chipmuenk/dsp_fpga)) clonen und sofort loslegen.

**Ihr PC:** Laden Sie die Notebooks per git auf Ihren lokalen Rechner herunter und führen Sie sie dort aus (Python - Installation erforderlich).

**PYNQ-Board:** Leihen Sie sich ein PYNQ-Evaluationsboard aus (Abb. 4), auf dem die Notebooks bereits vorinstalliert sind und verbinden Sie es per LAN mit Ihrem Rechner. Auf dem PYNQ-Board sind zusätzliche Notebooks installiert, die mit dem FPGA-Teil des Prozessors kommunizieren.

**Github:** Schließlich können Sie die Notebooks statisch betrachten auf GitHub ([https://github.com/chipmuenk/dsp\\_fpga](https://github.com/chipmuenk/dsp_fpga)).

**HM-Server:** Gibt es mangels Ressourcen momentan nicht.

Dieses Notebook demonstriert den grundsätzlichen Aufbau von Jupyter (IPython) Notebooks mit interaktiven Widgets am Beispiel der Fourierreihe einer Rechteckschwingung. Das Notebook besteht aus sog. Zellen, die Text (formatiert als Markdown oder Latex) oder ausführbaren Python-Code enthalten.

**Hinweise:**

- Mit Cell -> Run All werden alle Zellen nacheinander initialisiert bzw. ausgeführt.
- <SHIFT>-<RETURN> in einer einzelnen Zelle stellt die Zelle dar bzw. führt sie aus, <RETURN> fügt einen Zeilenumbruch ein.

Die Fourierreihe für eine symmetrische Rechteckschwingung mit Amplitude 1 und Periode  $T$  ist

$$s(t) = \frac{4}{\pi} \sum_{k=1}^{\infty} \frac{\sin((2k-1)2\pi t/T)}{2k-1}$$

Im Folgenden wird die Reihe mit einer endlichen Anzahl  $N$  Termen dargestellt.

```
In [4]: # Hier importierte Module und Funktionen können in allen folgenden Zellen verwendet werden
%matplotlib inline
import matplotlib.pyplot as plt
from numpy import pi,sin,linspace,zeros,arange,cos
from ipywidgets import interact

def square_wave(N=3, T=5, n_points=200):
    """Berechne und plotte die Fourierreihe mit N Termen für eine Rechteckschwingung mit
    Periode T und Amplitude A = 1."""
    t = linspace(-4*pi, 4*pi, n_points)
    s = zeros(n_points)
    for k in arange(1, N+1, 2): # start, stop (exklusive letztes Element), step
        s = s + (1./k * sin(2*pi*k*t/T))
    s = s * 4.0 / pi
    plt.plot(t, s)
```

```
In [5]: square_wave(N=5, T=10)
```

Abb. 3: Screenshot eines Jupyter Notebooks

## 2.2 PYNQ-Board

Mit Ihrem PYNQ Evaluationsboard (jeder bekommt eins) bearbeiten Sie ebenfalls Notebooks, programmieren das FPGA, lassen LEDs blinken und Ihren Kopfhörer quietschen.

Das PYNQ-Board (Abb. 4) ist ein Evaluationsboard für Xilinx „ZYNQ“ SOC; auf dem Chip befinden sich neben FPGA-Fabric zwei ARM9 Cores und zahlreiche Interfaces. Das Besondere an dem Board außer der knallpinken Farbe ist der Jupyter Notebook Server als bequemes Interface zum FPGA. Dank Audiocodec auf dem Board sind auch Audioexperimente leicht möglich. Und LEDs für „Blinky, die LED“ (das „Hello World“ für FPGAs) gibt es auch.

## 2.3 Screencasts

Ursprünglich für einen kompletten Inverted Classroom Kurs konzipiert, brauchen Sie die Screencasts jetzt nur noch für einzelne Themen (angekündigt) oder wenn Sie sich bestimmte Themen noch einmal anschauen möchten.

In der Cloud <https://www.magentacloud.de/share/6vdn6n5x0x> finden Sie neben den Videos im mp4-Format auch diese Unterlagen sowie die Folien zu den Videos im PDF und LibreOffice-Format.

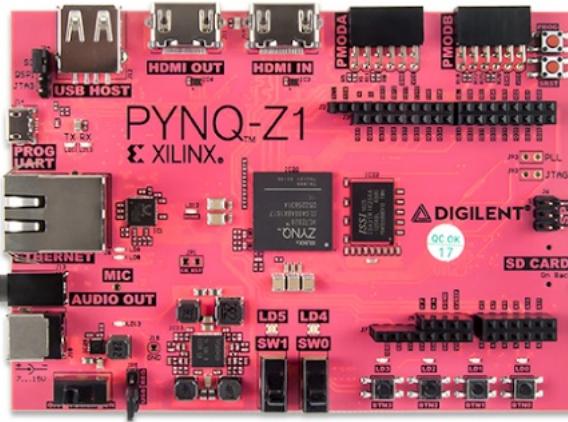


Abb. 4: PYNQ Board (c) Digilent

## 2.4 pyFDA

pyFDA (python Filter Design and Analysis) ist eine von mir in Python geschriebene Open-Source Software zur Analyse zeitdiskreter Systeme und zum Filterentwurf (Abb. 5) für Windows, OS X und Linux. Momentan benötigen Sie eine Python-Installation auf Ihrem Rechner, Binaries sind in Arbeit. Mehr Infos zur Installation etc. (und die Software selbst) finden Sie unter [github.com/chipmuenk/pyfda](https://github.com/chipmuenk/pyfda).

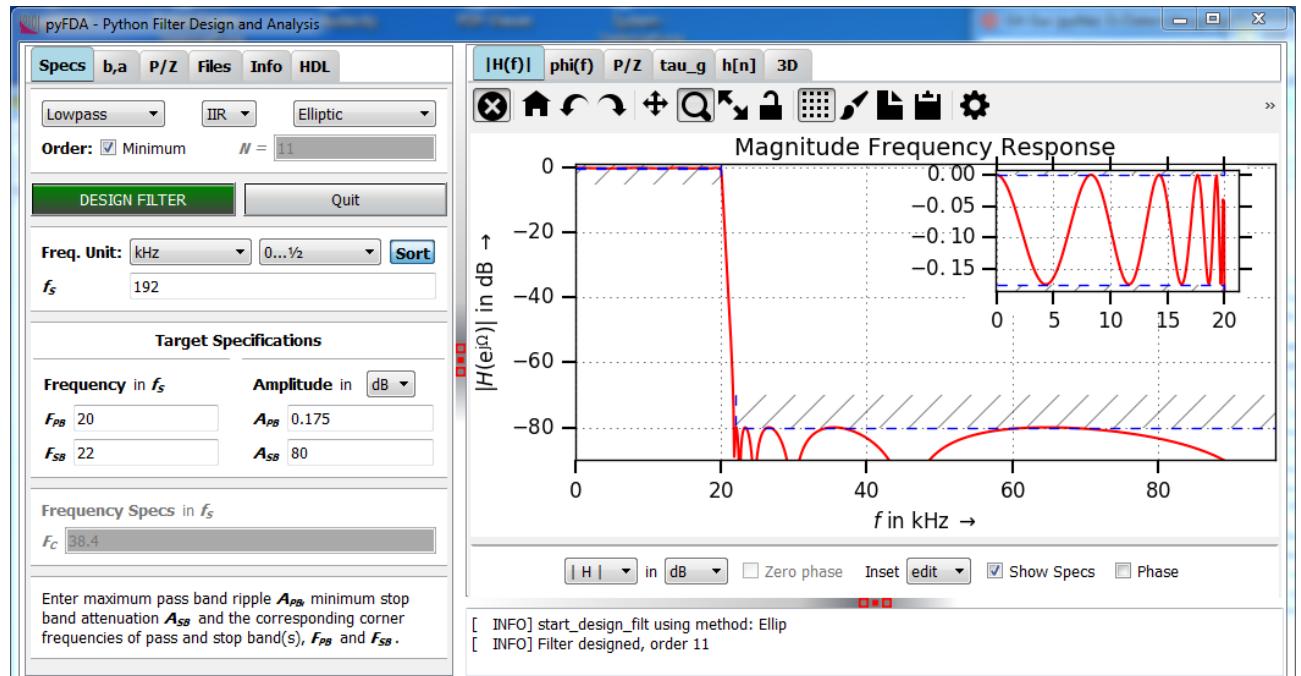


Abb. 5: pyFDA Screenshot

## 3 Noch mehr!?

Zusätzliche Literatur benötigen Sie eigentlich nur, wenn Sie mehr zu bestimmten Themen wissen wollen oder Ihnen die Folien nicht ausreichen. Im folgenden Abschnitt und den einzelnen Übungskapiteln gibt es kommentierte Literaturhinweise. Z.T. habe ich noch Videotipps für diejenigen, die laufende Bilder brauchen.

Wenn Sie mit meinem Vorlesungsstil nicht gut zurecht kommen, versuchen Sie doch mal den edX-Kurs (MIT) ELEC301x „Discrete Time Signals and Systems“ (<https://www.edx.org/course/rice/rice-elec301x-discrete-time-signals-1032>) oder den Coursera Kurs „Digital Signal Processing“ (<https://www.coursera.org/course/dsp>)

... und jetzt viel Spaß mit dem Kurs:



Abb. 6: Darf's etwas mehr sein? [<http://xkcd.com/864/>]

## Grundlagen - knapp & deutschsprachig

**Prof. N. Geng, [Gen11a] / [Gen11b]** Aufgabensammlung zu Signale und Systeme Teil 2: Zeitdiskrete Signale und Systeme, Hochschule München, Feb. 2011, 128 S. und Skriptum zu Signale und Systeme Teil 2: Zeitdiskrete Signale und Systeme, Hochschule München, Feb. 2011, 128 S.

Prof. Geng hat freundlicherweise Skript und Aufgabensammlung zur Vorlesung „Signale und Systeme“ für diesen Kurs zur Verfügung gestellt, Sie können die Dokumente bei Moodle herunterladen. Der Inhalt wird als bekannt vorausgesetzt!

**Lehrstuhl für Nachrichtentechnik an der TUM [Leh]**, Ein Lerntutorial für Nachrichtentechnik im world wide web, <http://www.lntwww.de/>.

Online-Tutorials, Videos und Übungen zu verschiedenen Themen („Bücher“) der Nachrichtentechnik. Für die Vorbereitung auf diese Vorlesung besonders empfohlen:

- Signaldarstellung (ganzes Buch, vor allem Kap. „Zeit- und frequenzdiskrete Signaldarstellung“)
- Stochastische Signaltheorie (nur Kap. „Kontinuierliche Zufallsgrößen“)

**G. Doblinger [Ger07]**, Zeitdiskrete Signale und Systeme - Eine Einführung in die grundlegenden Methoden der digitalen Signalverarbeitung.

Das frei erhältliche PDF (<https://www.nt.tuwien.ac.at/about-us/staff/gerhard-doblinger/>) → Teaching → Read More ist die aktualisierte Neuauflage des nur noch antiquarisch erhältlichen Buchs, Übungsaufgaben und Matlab-Beispiele können separat heruntergeladen werden. Es erklärt knapp, aber gut mit vielen Übungen und enthält auch eine Einführung in Multiratensysteme. Wortlängeneffekte fehlen.

**Martin Werner [Wer08]**, Signale und Systeme: Lehr- und Arbeitsbuch mit MATLAB®-Übungen und Lösungen.

Knappe Zusammenfassung der Grundlagen der analogen *und* digitalen Signalverarbeitung, das Buch geht auch kurz auf Abtastung, Interpolation und Resampling ein, auch Quantisierungsrauschen und allgemein stochastische Prozesse werden kurz behandelt. Gut geeignet, um „verschüttetes“ DSV und Nachrichtentechnikwissen wieder aufzufrischen, auch begleitend zur Vorlesung. Da der Stoff hier recht knapp abgehandelt wird, wird eher das vorige Buch zum Selbststudium empfohlen.

**Martin Werner [Wer11]**, *Digitale Signalverarbeitung mit MATLAB: Grundkurs mit 16 ausführlichen Versuchen.*

Knapper aber anschaulicher Grundlagenkurs, auch zum Selbststudium geeignet. Ohne Multiratensysteme, MATLAB-Programme zum Download. Achtung - der zweite Band heißt sehr ähnlich - „Digitale Signalverarbeitung mit MATLAB-Praktikum“, beschäftigt sich aber mit spezielleren Themen (u.a. Zustandsraumdarstellung digitaler Systeme, Skalierung und Koeffizientenquantisierung, adaptive Systeme)!

## Grundlagen - ausführlich & international

**S. J. Orfanidis [Orf10]**, *Introduction to Signal Processing.*

Ein Klassiker der Signalverarbeitung, der inzwischen auch als PDF (<http://www.ece.rutgers.edu/~orfanidi/intro2sp>) kostenlos erhältlich. Die Themen werden in eher ungewöhnlicher Reihenfolge behandelt; wenn Sie mit anderen Büchern nicht zureckkommen, versuchen Sie mal dieses hier. Es werden auch Multiratensysteme behandelt, Wortlängeneffekte werden nur angezissen. Viele Übungsaufgaben mit Lösungen (extra File)!

**S. W. Smith [Smi99]**, *The scientist and engineer's guide to digital signal processing.*

Sehr gute und ausführliche Erklärungen, vor allem zu DFT und Filtern! Als Buch oder online kostenlos unter <http://www.DSPguide.com> erhältlich.

**A. V. Oppenheim und R. W. Schafer [OS10]**, *Zeitdiskrete Signalverarbeitung / Discrete-time signal processing.*

Didaktisch exzellente Einführung über die digitale Signalverarbeitung (über 35 Jahre „gereift“) mit vielen Übungsaufgaben. Auch für das Selbststudium gut geeignet, auch auf Deutsch (2. Auflage) und in der Hochschulbibliothek erhältlich. Begleitender Kurs 6.341 mit gleichem Titel bei MIT OpenCourseWare ([ocw.mit.edu](http://ocw.mit.edu)).

**K.-D. Kammerer und K. Kroschel [KK18]**, *Digitale Signalverarbeitung: Filterung und Spektralanalyse mit MATLAB-Übungen*

Sehr gute und in die Tiefe gehende aber manchmal etwas knappe Darstellung der Grundlagen der DSP (ohne Multiratensysteme). Sehr ausführliche Darstellung der Verfahren zur Spektralschätzung (geht weit über Vorlesung hinaus). Umfangreicher MATLAB-Übungsteil.

## Spezielle und weiterführende Literatur

**U. Meyer-Baese [Mey07]**, *Digital Signal Processing with Field Programmable Gate Arrays.*

Das Standardwerk für schnelle digitale Signalverarbeitung auf FPGAs.

**U. Zölzer [ZöI05]**, *Digitale Audiosignalverarbeitung.*

Das Standardwerk zur digitalen Audiosignalverarbeitung. Ausführliche Beschreibung der Vor- und Nachteile verschiedener rekursiver Filterstufen.

**U. Zölzer und X. Amatriain [ZA11], DAFX: digital audio effects.**

Auf letzterem Buch aufbauend, ein weiteres Standardwerk, diesmal zu digitalen Audioeffekten.

**A. Mertins [Mer10], Signaltheorie: Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung.**

Signal**theorie** verschiedener Transformationen (Fourier-, Kurzzeitfourier-, Wavelet- ...), nicht-lineare Transformationen, Filterbänke. Für alle, die es immer schon etwas genauer wissen wollten  
...

# **Teil I**

## **Übungsaufgaben**



# 1 LTI: LTI-Systeme im Zeitbereich

## Lernziele

Dieses Kapitel beschäftigt sich mit zeitdiskreten, linearen zeit-invarianten (*Linear Time-Invariant, LTI*) Systemen im *Zeitbereich* und deren Darstellung als Signalflussgraph (SFG, siehe auch Anhang A). Es wiederholt Grundlagen der digitalen Signalverarbeitung, die als bekannt vorausgesetzt werden. Zum Auffrischen empfehle ich die Unterlagen von Prof. Geng (→ Moodle) oder ein anderes der im Abschnitt 3 empfohlenen einführenden Bücher.

Nach dem Durcharbeiten dieses Kapitel sollten Sie die folgenden Eigenschaften dieser Systeme verstanden haben und in einander überführen können:

- Sie bestimmen **Impulsantwort  $h[n]$**  und **Differenzengleichung** aus **SFG** und **Systemfunktion  $H(z)$**  und umgekehrt.
- Sie finden den **kritischen Pfad** in einem SFG und berechnen damit die **maximale Taktrate**.
- Sie bestimmen den **Hardware-Ressourcenverbrauch** (FlipFlops, Addierer, Multiplizierer, ...) aus dem SFG.
- Sie konstruieren **Direktformfilter** Typ 1 und 2 und **transponierte Filter** aus Systemfunktion und umgekehrt.

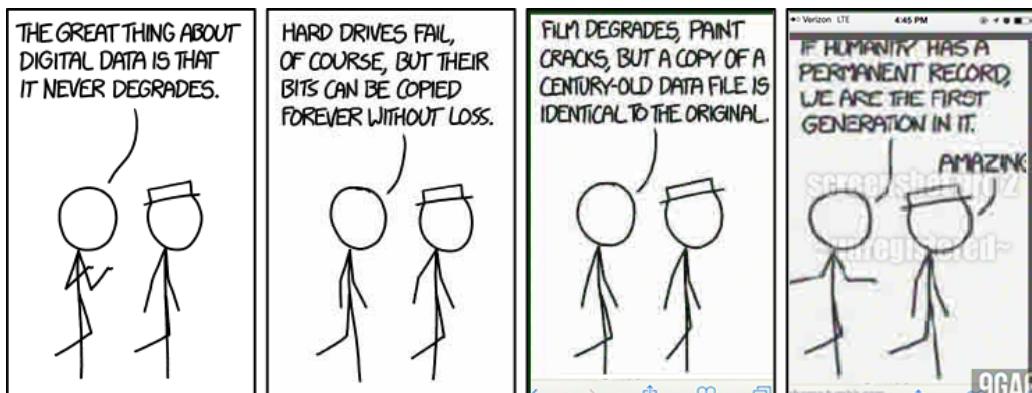


Abb. 1.1: Digital Data [<https://xkcd.com/1683>]

## Theorie

### Direktform-Systeme

Abb. 1.5 zeigt Systeme zweiter Ordnung in Direktform 1 (DF1) und 2 (DF2) - Struktur mit der gleichen Systemfunktion

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}}.$$

Die Koeffizienten für eine Realisierung in DF1- oder DF2 Struktur können direkt<sup>1</sup> aus einer gegebenen Übertragungsfunktion abgelesen werden. Versuchen Sie zu Übungszwecken aus den SFGs die Systemfunktion abzuleiten.

### Kritischer Pfad

In einem synchron getakteten (Teil-)System (= alle Register bekommen den gleichen Takt) müssen beim Eintreffen der nächsten Taktflanke die Daten an allen Registereingängen stabil anliegen. Der *kritische Pfad* mit der längsten Verzögerungszeit  $\tau_{max}$  von einem Registerausgang zu einem Registereingang bestimmt daher die maximal zulässige Taktfrequenz des gesamten Systems,  $f_{S,max} = 1/\tau_{max}$ . Die Verzögerungszeit setzt sich zusammen aus allen arithmetischen Operationen (Additionen, Multiplikationen, ...) im jeweiligen Pfad. Der kritische Pfad des Systems Abb. 1.2 hat z.B. eine Verzögerungszeit von  $\tau_{krit} = 3\tau_{add} + 2\tau_{mult}$ . Beachten Sie dabei:

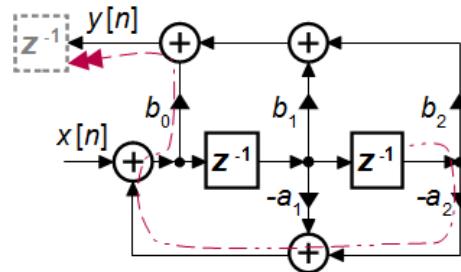


Abb. 1.2: IIR-Filter zweiter Ordnung in Direktform mit eingezeichnetem kritischen Pfad

- Werte am Eingang  $x[n]$  werden von einem Registerausgang außerhalb des betrachteten Teilsystems geliefert, ebenso werden die Daten am Ausgang  $y[n]$  von einem weiteren Registereingang des nachfolgenden Funktionsblocks erwartet (siehe z.B. Abb. 1.2). Auch diese Pfade müssen u.U. bei der Bestimmung des kritischen Pfads berücksichtigt werden!
- Ein Addierer mit drei Eingängen besteht eigentlich aus zwei Addierern mit je zwei Eingängen, die in unterschiedlicher Reihenfolge kombiniert werden können:  $(a + b) + c = a + (b + c)$  - daraus können sich abhängig von der Implementierung unterschiedliche kritische Pfade ergeben.

Die Länge des kritischen Pfade und damit die maximale Taktrate können sich für verschiedene Implementierungen der gleichen Übertragungsfunktion deutlich unterscheiden (siehe z.B. Aufgabe 1.5). Das gleiche gilt auch für den Verbrauch an Hardware-Ressourcen (Register, Addierer, Multiplizierer).

<sup>1</sup>Daher der Name ...

## Spezielle Python-Funktionen

Funktion	Python	Matlab	Beschreibung
Faltung	<code>y = np.convolve(x,h)</code>	<code>y = conv(x,h)</code>	Falte Vektoren $x$ und $h$ miteinander.
Nullstellen	<code>r = np.roots(p)</code>	<code>r = roots(p)</code>	Finde die Wurzeln $r$ eines Polynoms mit den Koeffizienten $p$ .
Polynom	<code>p = np.poly(r)</code>  <code>y1 = np.polyval(p, x1)</code>	<code>p = poly(r)</code>  <code>y1 = polyval(p, x1)</code>	Finde die Koeffizienten $p$ eines Polynoms mit den Wurzeln $r$ .  Bestimme den Wert $y1$ des Polynoms mit den Koeffizienten $p$ an der Stelle $x1$
<b>Grafik</b>			
Plotfenster	<code>fig = plt.figure(1)</code>	<code>figure(1)</code>	Öffne ein Plotfenster
Subplots	<code>ax = fig.add_subplot(211)</code>  <code>fig,ax = plt.subplots(nrows=1, ncols=2)</code>  <code>fig,(ax0, ax1) = plt.subplots(1, 2)</code>  <code>fig,ax=plt.subplots()</code>	<code>subfigure(211)</code>	Öffne Subplot (1,1) von 2  Öffne Plot- und Subplotfenster. $ax$ ist ein Array, referenziere $ax[0]$ und $ax[1]$ .  Alternative Schreibweise.  Alternativ für einen Subplot
Liniенplot	<code>ax.plot(x,y)</code>	<code>plot(x,y)</code>	Zeichne Liniенplot
Stemplot	<code>ax.stem(x,y)</code>	<code>stem(x,y)</code>	Zeichne Impulse als „Stengel“

Tab. 1.1: Spezielle Befehle zu Kap. 1

### Anmerkungen zu Tab. 1.1:

**Allgemein:** In Python finden sich die meisten Funktionen zur Signalverarbeitung im Numpy Paket (abgekürzt mit `np`) oder im Signal Processing Modul von Scipy (`scipy.signal`), in Matlab in der Signal Processing Toolbox.

**convolve** Optional: mode=“same“ oder „valid“ um Randeffekte zu unterdrücken.

## Notebooks und Labs

Filename	Beschreibung
<b>Notebooks</b>	
<code>LTI_faltung</code>	Einfaches Code-Beispiel zur Demonstration von Faltung
<code>LTI_Basics</code>	Einfache Filter im Zeitbereich
<code>LTI-Animate_Convolution</code>	Interaktive und Animierte Darstellung der Faltung

Filename	Beschreibung
LTI_IIR	Definition eines IIR-Filters über seine Koeffizienten, Impulsantwort und Faltung mit einer Eingangssequenz
LTI_sensor_DL	Musterlösung zu Aufgabe 1.1c.
LTI_convolve_audio	Faltung eines Audiosignals mit der Raumimpulsantwort (von Allen Downey, <a href="http://think-dsp.com">http://think-dsp.com</a> )
<b>Labs</b>	
VLab 0	Vivado-Installation
VLab 1	Ein erstes Design

Tab. 1.2: Jupyter Notebooks und Labs zu Kap. 1

Der wichtigste Python-Befehl für dieses Kapitel ist die **Faltung**, `np.convolve(x, h)` im Notebook **LTI\_faltung**. Das Mini-Programmchen zeigt außerdem, wie man Signale definiert und plottet.

Ein paar weitere nützliche Tricks und Anweisungen zeigt das Notebook **LTI\_IIR**, u.a. wie man Subplots erzeugt, im Zeitbereich filtert und die Impulsantwort auch von rekursiven Systemen bestimmt.

## 1.1 Filterung abgetasteter Signale → M1.1

Das Signal eines Temperatursensors ändert sich so langsam, dass es für die Dauer der Messung als Gleichsignal betrachtet werden darf,  $x_{Temp}(t) = 1,5$  V. Das Sensorsignal ist mit einer Brummstörung überlagert,  $x_n(t) = 0,5 \cos(2\pi \cdot 50 \text{ Hz} \cdot t)$ . Das Signal  $x(t) = x_{Temp}(t) + x_n(t)$  wird durch Abtastung mit  $f_{S1} = 25$  Hz,  $f_{S2} = 200$  Hz,  $f_{S3} = 240$  Hz und  $f_{S4} = 50 \cdot \pi$  Hz in zeitdiskrete Folgen  $x_i[k]$  gewandelt. In einem nachgeschalteten digitalen Filter soll die Brummstörung möglichst gut unterdrückt werden.

### a) Abgetastete Signale

Geben Sie für die Abtastraten  $f_{S1} \dots f_{S4}$  das zeitdiskrete Signal  $x_i[k] = x_{Temp,i}[k] + x_{n,i}[k]$  an und tragen Sie die ersten 8 Werte in Tab. 1.3 ein! Ist  $x_i[k]$  periodisch? Wenn ja, nach wievielen Abtastwerten  $N$  wiederholt sich  $x_i[k]$ ? Wieviele Perioden  $L$  von  $x_n(t)$  sind darin enthalten? Skizzieren Sie die abgetasteten Signale, erklären Sie das Ergebnis für  $f_{S1} = 25$  Hz!

### b) Filterung

Bestimmen Sie jeweils die folgenden Eigenschaften der beiden Moving Average Systeme und des verlustbehafteten Integrators ( $\alpha = 0,5$ ) in Abb. 1.3:

**Impulsantwort:** Skizzieren Sie die Impulsantwort  $h[k]$  oder schreiben Sie sie auf!

**Skalierung:** Wie muss die Amplitude des Ausgangssignals skaliert werden, damit der Gleichanteil des Eingangssignals korrekt wiedergegeben wird?

**Einschwingen:** Ab welchem Sample ist das Ausgangssignal eingeschwungen?

**Ausgangssignal:** Berechnen Sie für die Eingangsfolge  $x_2[k]$  ( $f_{S2} = 200$  Hz) das Ausgangssignal  $y[k]$  für  $k = 0 \dots 7$  und tragen Sie die Werte in Tab. 1.3 ein.

$f_s$	$n$	0	1	2	3	4	5	6	7
25 Hz	$x_1[n]$								
200 Hz	$x_2[n]$								
240 Hz	$x_3[n]$								
$50\pi$ Hz	$x_4[n]$								
	$y_{MA,4}[n]$								
200 Hz	$y_{MA,3}[n]$								
	$y_{cascl}[n]$								
	$y_{Int}[n]$								

Tab. 1.3: Ergebnisse zu Aufgabe 1.1

**Brummunterdrückung:** Versuchen Sie aus den diskreten Werten  $y[k]$  eine geschlossene Beschreibung für den eingeschwungenen Zustand zu finden und daraus die Brummunterdrückung abzuschätzen.

**Ordnung** (nur System Abb. 1.3a): Die Ordnung des Filters wird jetzt auf  $N = 3$  reduziert. Passen Sie die Skalierung an, bestimmen Sie das Ausgangssignal  $y_{b,3}[k]$  und tragen Sie die Werte für  $k = 0 \dots 7$  in Tab. 1.3 ein. Erklären Sie das Ergebnis!

**Kaskadiertes MA Filter:** Durch welches nicht-kaskadierte Filter man das System in Abb. 1.3b ersetzen? Welchen Nachteil hat die nicht-kaskadierte Form?

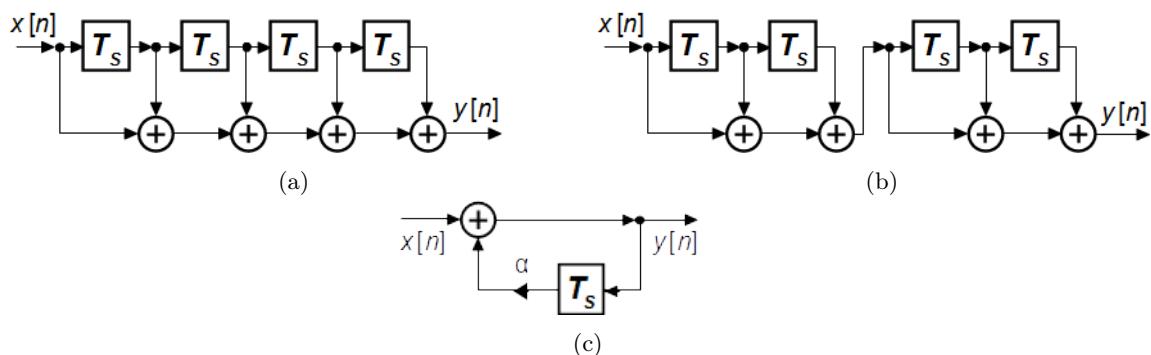


Abb. 1.3: Moving Average (MA) (a), kaskadiertes MA - Filter (b) und verlustbehafteter (gedämpfpter) Integrator (c)

- c) **Simulieren** Sie abgetastete Eingangssignale, Impulsantwort und die Ausgangssignale der Systeme mit Python, hierzu gibt es ein Notebook.

- d) Welchen **kritischen Pfad** haben die Filterimplementierungen und welche maximale Taktfrequenzen resultieren daraus? Eine Addition benötigt 2 ns, eine Multiplikation 10 ns.
- e) Wieviele **Taktzyklen pro Sample** benötigt ein uC, auf dem die Filter jeweils implementiert werden sollen? Pro Taktzyklus kann der uC nur eine Rechenoperation (Add, Multiply, ...) durchführen. Operationen zur Speicherverwaltung (Schieberegister, Ringbuffer etc.) sollen vernachlässigt werden.
- f) Wie könnte eine **effiziente Implementierung des Moving Average Filters** in Software oder Hardware aussehen? Schauen Sie sich dazu an, wie sich das Ausgangssignal mit jedem Sample ändert.
- g) Wie könnte ein System zur **Unterdrückung des Gleichanteils** (z.B. Offset) von Wechselsignalen aussehen?

## 1.2 Faltung mit einfachem FIR-Filter → M1.2

- a) Ermitteln Sie aus der Struktur des FIR-Filters in Abb. 1.4a die **Impulsantwort  $h[n]$**  und skizzieren Sie diese.

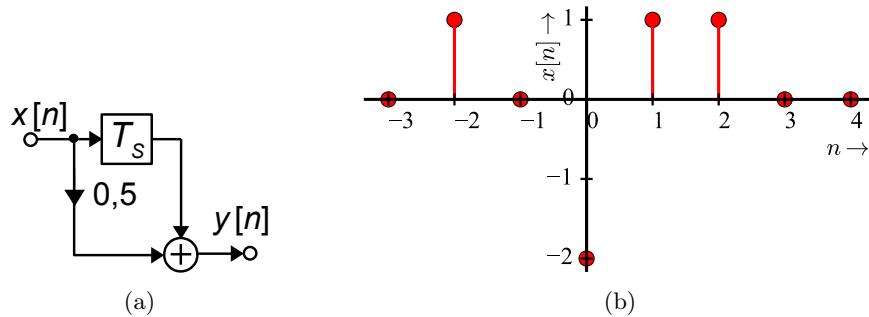


Abb. 1.4: FIR-Filter (a) mit Eingangssignal  $x[n]$  (b) zu Aufgabe 1.2 und 2.2

- b) Geben Sie die **Differenzengleichung** für das Filter an.
- c) Ermitteln Sie das **Ausgangssignal  $y[n]$**  für das Eingangssignal  $x[n] = \{1; 0; -2; 1; 1\}$  mit  $n = -2, \dots, 2$  (Abb. 1.4b) durch diskrete Faltung  $x[n] * h[n]$  oder  $h[n] * x[n]$ . Skizzieren Sie das Ausgangssignal  $y[n]$ .
- d) Welche **Reihenfolge** für die Berechnung der Faltung ist bei FIR-Filtern im Allgemeinen besser geeignet?

## 1.3 Systemeigenschaften aus Impulsantwort → M1.3

Die folgenden Systeme werden durch ihre Impulsantwort  $h[n]$  mit  $n = 0, 1, \dots$  beschrieben.

- a)  $h_A[n] = \{0,25; 0,5; 0,25\}$   
 b)  $h_B[n] = \{0,25; -0,5; 0,25\}$   
 c)  $h_C[n] = \{0,25; 0,5; 0,75\}$

Sind die Systeme kausal und stabil? Erwarten Sie eher Hochpass- oder Tiefpassverhalten aufgrund der Impulsantwort? Zeichnen Sie eine Hardware-Implementierung auf. Gibt es mehr als eine Hardware-Realisierung?

## 1.4 Aufstellung der Differenzengleichung für Biquads aus SFG → M1.4

In Abb. 1.5 sind die SFGs für 6 verschiedene Realisierungen der Systemfunktion (1.1) gezeigt. Da die Systemfunktion quadratisch in Zähler und Nenner ist, werden die zugehörigen Systeme auch „Biquads“ genannt. Sie sind wichtige Grundbausteine für komplexere Filter, werden aber auch für sich verwendet.

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2}}{1 + a_1 z^{-1} + a_2 z^{-2}} \quad (1.1)$$

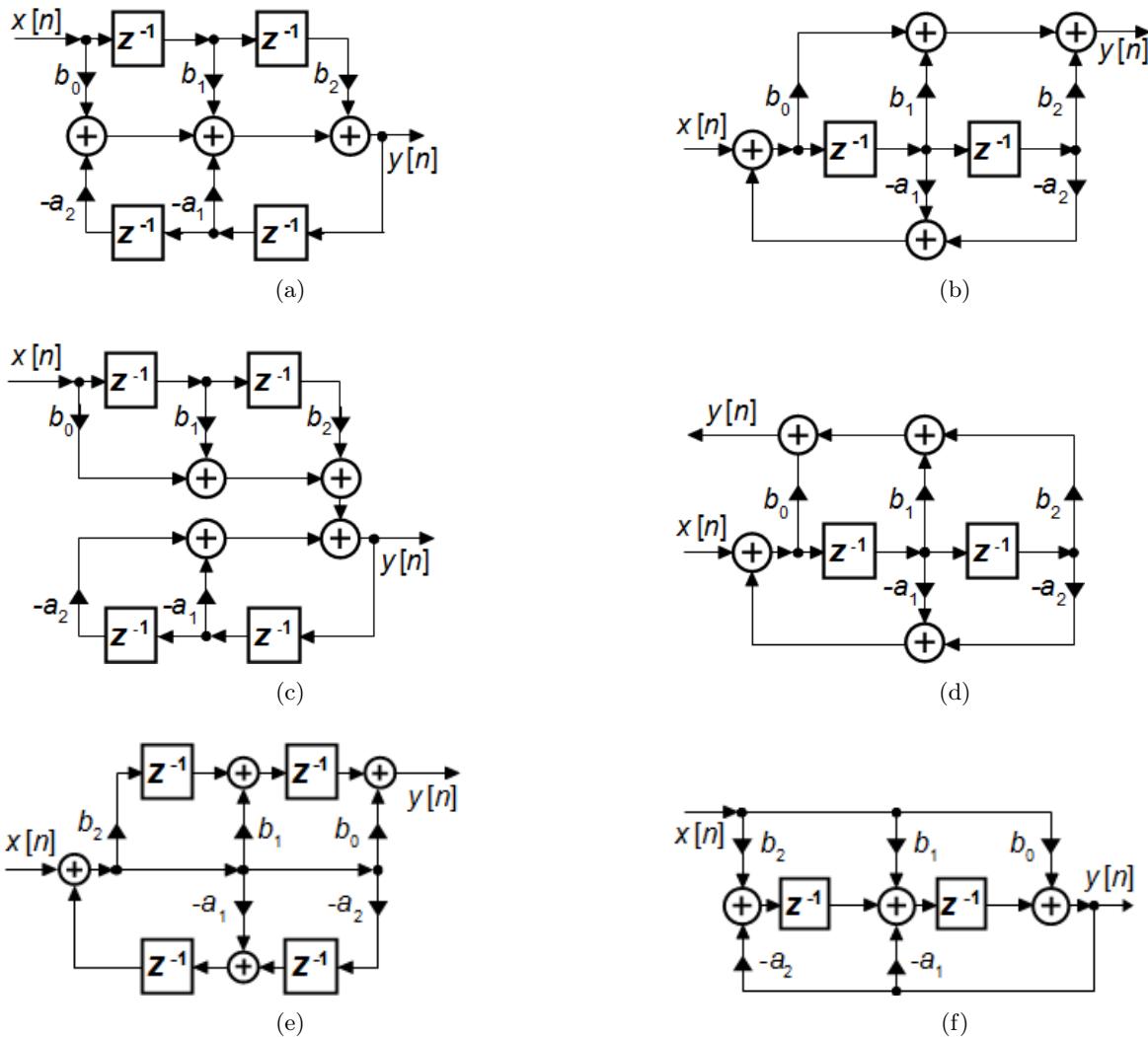


Abb. 1.5: IIR-Systeme mit identischer Systemfunktion zweiter Ordnung in Direktform 1 (a), (c) und (e), transponiert) und Direktform 2 (b), (d) und (f, transponiert) zu Aufgabe 1.4, 1.5 und 1.6

Informationen zu SFGs finden Sie auch im Anhang A.

- Versuchen Sie, durch **grafische Umformungen** die Systeme Abb. 1.5b - 1.5f in das DF1 - System Abb. 1.5a zu überführen.
- Stellen Sie die **Differenzengleichungen und Systemfunktionen** der Systeme auf. Manchmal ist es einfacher, einen Zwischenknoten zu definieren und die DZGL mit dessen Hilfe aufzustellen, oft ist es einfacher erst die Systemfunktion zu bestimmen und daraus die DZGL abzuleiten.

## 1.5 Kritischer Pfad und Ressourcenverbrauch 1 → M1.5

In Abb. 1.5 sind Systeme zweiter Ordnung in verschiedenen Arten von Direktform dargestellt. Bestimmen Sie Ressourcenverbrauch (= Anzahl Addierer, Multiplizierer, Register) der Systeme, den kritischen Pfad und die dadurch begrenzte Taktfrequenz sowie die Anzahl der Multiplikationen pro Input Sample (MPIS). Eine Multiplikation soll  $\tau_{mul} = 12$  ns und eine Addition oder Subtraktion  $\tau_{add} = 3$  ns dauern. Beachten Sie, dass ein Addierer mit drei Eingängen mit zwei Addierern mit je zwei Eingängen implementiert wird.

Bestimmen Sie die Verzögerung des kritischen Pfads der Systeme für die Ordnung  $N$ . Berechnen Sie daraus die max. Taktfrequenz für  $N = 8$  für schnellstes und langsamstes System.

## 1.6 Kritischer Pfad und Ressourcenverbrauch 2 → M1.6

Bestimmen Sie die für die Systeme in Abb. 1.6 den Ressourcenverbrauch (Addierer, Multiplizierer, Register) und die Verzögerungszeit  $\tau_{krit}$  des kritischen Pfads sowie die maximale Taktfrequenz. Eine Multiplikation soll  $\tau_{mul} = 12$  ns und eine Addition oder Subtraktion  $\tau_{add} = 3$  ns dauern.

Vergleichen Sie Ressourcenverbrauch und kritischen Pfad der Systeme zweiter Ordnung Abb. 1.6b) bis c) mit dem eines Systems in Direktform 2 (Abb. 1.5d).

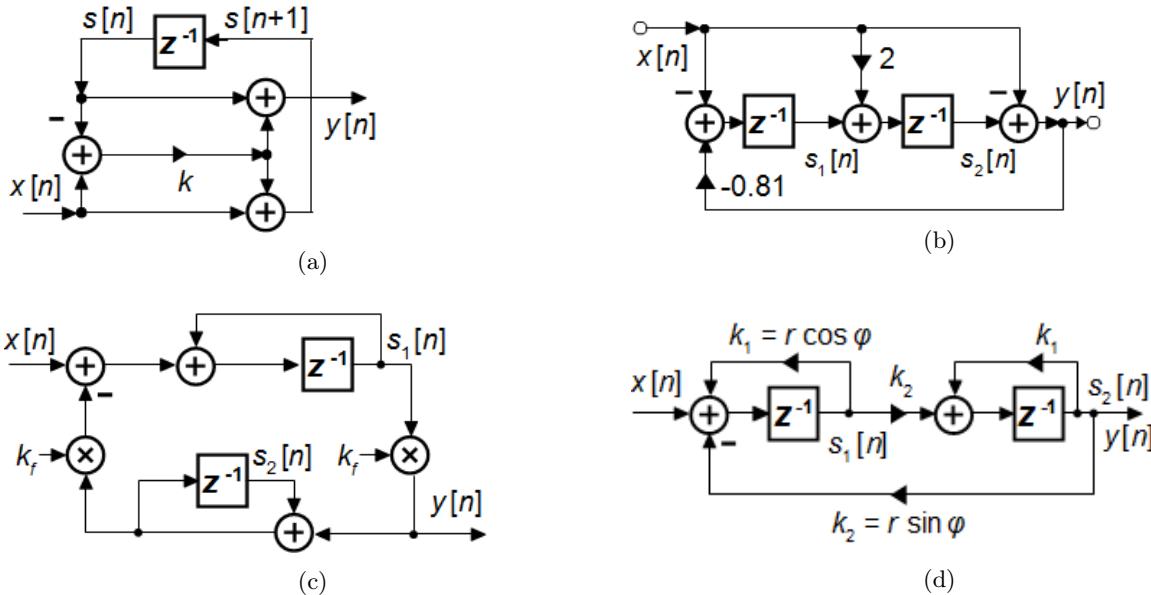


Abb. 1.6: Filterstrukturen zu Aufgabe 1.6, 1.7 und 2.6

## 1.7 Transponierte Systeme zu Aufgabe 1.6 → M1.7

Transponieren Sie die Systeme aus Abb. 1.6 und bestimmen Sie wiederum Ressourcenverbrauch und kritischen Pfad / max. Taktfrequenz (Angaben siehe Aufgabe 1.6).

## 1.8 Allgemeine Fragen zu FPGAs → M1.8

- Nennen Sie Vor- und Nachteile von FPGAs im Vergleich zu ASICs, DSPs und uCs.
- Warum werden Multiplizierer und Speicher oft als optimierte Hardmakros auf FPGAs implementiert? Welchen Nachteil haben Hardmakros?

- c) Welche Technologien werden zur Herstellung von FPGAs heutzutage überwiegend verwendet?  
Welche Vor- und Nachteile haben diese?



## 2 LTF: LTI-Systeme im Frequenzbereich

### Lernziele

Dieses Kapitel beschäftigt sich mit LTI-Systemen im *Frequenzbereich*. Nach dem Durcharbeiten dieses Kapitel sollten Sie die folgenden Eigenschaften von zeitdiskreten LTI-Systemen verstanden haben und diese u.a. aus der Systemfunktion  $H(z)$  bestimmen können:

- Ausgehend von einem **Pol-/Nullstellendiagramm** in der  $z$ -Ebene skizzieren Sie den prinzipiellen Verlauf des Betragsgangs und begründen ob das System stabil und kausal ist.
- Sie berechnen den **komplexen Frequenzgang**  $H(e^{j\Omega})$  aus der Systemfunktion  $H(z)$
- Sie bestimmen **Betrags- und Phasengang**  $|H(e^{j\Omega})|$  und  $\angle H(e^{j\Omega})$  aus der Systemfunktion. Hierzu nutzen Sie Symmetrien der Systemfunktion aus und fassen komplexe Terme zu Sinus- oder Cosinusfunktionen zusammen.
- Sie bestimmen die **Gruppenlaufzeit**  $\tau_g(e^{j\Omega})$  aus der Systemfunktion. Sie bestimmen die Gruppenlaufzeit aus der linearen Phase und umgekehrt.
- Sie skizzieren ein **Moving Average Filter** der Ordnung  $N$  und dessen P/N Diagramm und berechnen dessen Betragsgang.

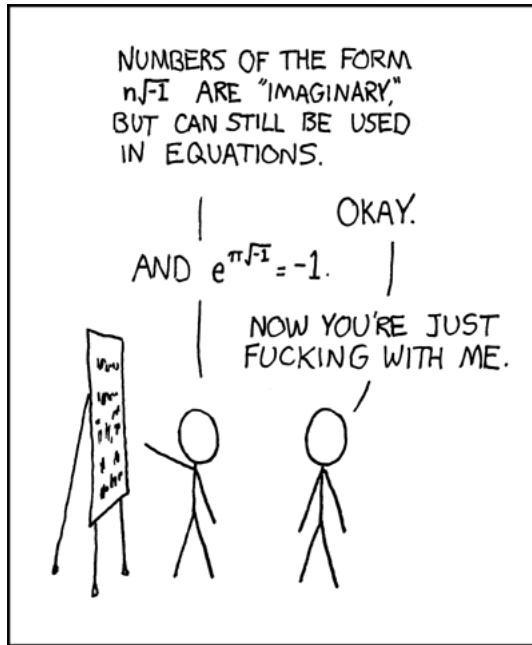


Abb. 2.1: Imaginäre Erklärungen [<http://xkcd.com/179/>]

## Theorie

### Definition der Systemfunktion

Aus der Systemfunktion  $H(z)$  können die meisten Eigenschaften des Systems abgeleitet (Impulsantwort, komplexer Frequenzgang, P/N-Diagramm,...) werden. Daher werden Systeme in Python / Matlab meist ausgehend von der Systemfunktion definiert, z.B.:

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

**b, a** = [2, 1]; **a** = [1, 0, 0.81]      **N** = [0.5]; **P** = [-0.9j, 0.9j]; **k** = 2

**b, a** sind die Koeffizienten von Zähler- und Nennerpolynom der Systemfunktion  $H(z)$ . Ausgehend von der Polynomform mit *positiven* Exponenten werden die Koeffizienten in *fallender Ordnung* angegeben. Koeffizienten mit Wert Null dürfen Sie nicht weglassen!

$$H(z) = \frac{b_0 z^N + b_1 z^{N-1} + \dots + b_N}{a_0 z^P + a_1 z^{P-1} + \dots + a_P} = \frac{b_0 + b_1 z^{-1} + \dots + b_N z^{-N}}{a_0 + a_1 z^{-1} + \dots + a_P z^{-P}} z^{N-P}$$

**b** = [b0, b1, ..., bN]; **a** = [a0, a1, ..., aP]

Wenn  $N = P$  ist, erhält man identische Koeffizienten ausgehend von der negativen und positiven Polynomform. Wenn  $P \geq N$  ist, sind die Koeffizienten „verschoben“ aufgrund der zusätzlichen Polstellen.

Bei FIR-Systemen muss **a** nicht angegeben werden, die notwendige Anzahl von Polen im Ursprung wird automatisch ergänzt. Wenn Sie ein Nennerpolynom angeben, muss seine Ordnung mindestens gleich groß der des Zählerpolynoms sein, da Sie ansonsten ein akausales System erhalten. Bei einem All-Pole System muss **b** = 1 angegeben werden.

**N, P** sind Null- und Polstellen der Systemfunktion in Produktform, ggf. muss ein Vorfaktor **k** angegeben werden. Bei FIR-Systemen muss **P** nicht angegeben werden (Default: **P** = 0).

**Achtung:** P/N müssen in Matlab als *Spaltenvektoren* angegeben werden, also mit Semikolons als Trenner, damit **zplane** weiß was gemeint ist:

**N** = [0.5]; **P** = [-0.9\*j; 0.9\*j]; **k** = 2;    N/P als *Spaltenvektor* (**Matlab**)

In Python wird nicht unterschieden (ein Semikolon als Elementtrenner ist ein Syntaxfehler):

**N** = [0.5]; **P** = [-0.9\*j, 0.9\*j]; **k** = 2;    (**Python**)

Die meisten Funktionen entscheiden hier selbstständig welche Form für einen eindimensionalen Vektor sinnvoll ist und transponieren ihn bei Bedarf.

**Umrechnung** zwischen beiden Formen ist möglich mit den aus dem letzten Kapitel bekannten Befehlen

**N** = **roots(b)**; **P** = **roots(a)**; und in die andere Richtung  
**b** = **k \* poly(P)**; **a** = **poly(N)**;

## Spezielle Python-Funktionen

Funktion	Python	Matlab	Beschreibung
Impulsantwort	<code>h, t = dsp_fpga.impz(b,a)</code>	<code>[h,t] = impz(b,a)</code>	Impulsantwort des durch <code>b,a</code> definierten Systems.
Filterung	<code>y = sig.lfilter(b,a,x)</code>	<code>y = filter(b,a,x)</code>	Filtere das Signal <code>x</code> mit dem durch <code>b,a</code> definierten System.
Anfangsbedingungen	<code>zi = sig.lfilter_zi(b,a)</code>  <code>zi = sig.lfiltic(b, a, y, x=None)</code>	<code>zi = filtic(b,a,x [,y])</code>	Berechne Ausgangszustand <code>zi</code> , der eingeschwungenem Zustand des Filters entspricht  Berechne eingeschwungenen Zustand des Filters
Zero-Phase Filterung	<code>y = sig.filtfilt(b,a,x)</code>	<code>y = filtfilt(b,a,x)</code>	Filterung ohne Gruppenlaufzeit mit dem durch <code>b,a</code> definierten System.
Frequenzgang (komplex)	<code>W,H = sig.freqz(b,a)</code>  <code>W,H =sig.freqz_zpk(N,P,k)</code>	<code>H,W = freqz(b,a)</code>  ... aus den Null- und Polstellen	Berechne den komplexen Frequenzgang entlang des Einheitskreises.  ... aus den Null- und Polstellen
Pole und Nullstellen	<code>dsp_fpga.zplane(b,a)</code>  <code>dsp_fpga.zplane(N,P, zpk=True)</code>  <code>b,a = sig.zpk2tf(N,P,k)</code>  <code>N,P,k = sig.tf2zpk(b,a)</code>	<code>zplane(b,a)</code>  <code>zplane(N,P)</code>  <code>b,a = zp2tf(N,P,k)</code>  <code>N,P,k = tf2zpk(b,a)</code>	Plotte P/N - Plan ausgehend von Koeffizienten <code>b,a</code> .  Plotte P/N - Plan ausgehend von Null/Polstellen <code>N,P</code> .  Umrechnung <code>N,P,k → b,a</code>  Umrechnung <code>b,a → N,P,k</code>
Groupdelay	<code>w,gd = grpdelay(b,a)</code>	<code>gd,w=grpdelay(b,a,n)</code>	Gruppenlaufzeit des durch <code>b,a</code> definierten Systems

Tab. 2.1: Spezielle Befehle zu Kap. 2

### Anmerkungen zu Tab. 2.1:

**impz** gibt es in Python von Haus aus bislang nicht, Sie finden die Funktion in meiner **dsp\_fpga** - Lib unter [https://github.com/chipmuenk/dsp\\_fpga/tree/master/notebooks/dsp\\_fpga\\_lib.py](https://github.com/chipmuenk/dsp_fpga/tree/master/notebooks/dsp_fpga_lib.py)  
Optionale Parameter: Anzahl der Punkte, Abtastfrequenz.

**filtfilt** Vorwärts/Rückwärtsfilterung eines Datenblocks mit dem gleichen Filter eliminiert den Phasengang (und damit auch die Gruppenlaufzeit), das funktioniert natürlich nur offline mit Datenblöcken, da man die Zeit nicht rückwärts laufen lassen kann!

**(l)filter** In Python ist **filter()** eine Funktion aus der Standardbibliothek, die Elemente aus einer Liste herausfiltert. Die Funktion zum linearen Filtern von Signalen heißt in Python daher **sig.lfilter(b,a,x,zi)**, optional können Anfangsbedingungen übergeben werden (genauso in Matlab).

**freqz** gibt den *komplexen* Frequenzgang  $H(e^{j\Omega})$  des Systems in Abhängigkeit von der normierten Kreisfrequenz zurück. Daraus können mit `abs(H)` bzw. `angle(H)` leicht Betrag bzw. Phase abgeleitet werden, also z.B.  
`[H,W] = freqz(b,a); plot(W / (2 * np.pi), abs(H))`.

Optionale Parameter:

**Python:** freqz(b, a=1, worN=None, whole = False)  
 worN=1024 nimmt 1024 Datenpunkte  
 whole = True: Plotten über den gesamten Frequenzbereich  $\Omega = 0 \dots 2\pi$

**Matlab:** freqz(b,a,n=1024, fs=1000)  
 fs=1000: Angabe der Abtastfrequenz zur Skalierung des zurückgegebenen Frequenzvektors in physikalischen Frequenzen

**zplane** gibt es in Python on Haus aus bislang nicht, Sie finden die Funktion in meiner **dsp-fpga** - Lib (s. o.).

**zp2tf** ist in Matlab speziell für zeitdiskrete Systeme; hier werden die Koeffizienten in der korrekten Reihenfolge für die Polynomform mit negativen Exponenten erwartet. Für zeitkontinuierliche Systeme gibt es in Matlab die Funktion **zp2tf**.

Nein, ich weiß auch nicht, warum die Entwickler von scipy sich entschieden haben die Reihenfolge der berechneten Arrays umzudrehen ...

## Notebooks und Labs

Filename	Beschreibung
<b>Notebooks</b>	
LTF-Basics	Darstellung von Signalen im Frequenzbereich (Betragsgang, P/N-Diagramm)
LTF-Animate_PZ	Animierte Darstellung des Zusammenhangs zwischen Pol-/Nullstellen und Betragsgang
LTF-Sensor	Code zu Aufgabe 2.1 „Filterung des Sensorsignals im Frequenzbereich“
LTF-IIR-allgemein	Code zu Aufgabe 2.9 „Allgemeine IIR-Struktur“
LTF-Integrator	Simulation eines Integrators (Akkumulators) mit interaktiv variierbarer Dämpfung
LTF-Notchfilter	Code zu Aufgabe 2.12 „Notchfilter“

Tab. 2.2: Jupyter Notebooks zu Kap. 2

## Experimente und Lesestoff zu Kap. 2

**Zero-Phase Filtering** <http://www.embedded.com/design/configurable-systems/4008847/DSP-Tricks-Doing-Zero-phase-filtering>

**Room Impulse Response:** App zur Bestimmung und Aufzeichnung der RaumImpulsantwort (RIR) :  
<https://play.google.com/store/apps/details?id=com.julian.apps.ImpulseResponse>

### 2.1 Filterung des Sensorsignals im Frequenzbereich → M2.1

Die Systeme aus Aufgabe 1.1 sollen nun im Frequenzbereich untersucht werden. Bestimmen Sie für die Systeme aus Abb. 1.3 die folgenden Eigenschaften und überprüfen Sie die Rechnungen durch Simulationen in Python. Beim verlustbehafteten Integrator soll  $\alpha = 0,9$  gelten.

a) Komplexer Frequenzgang  $H(e^{j\Omega})$  und Betragsgang  $|H(e^{j\Omega})|$

- b) DC-Verstärkung  $H(\Omega = 0)$
- c) Unterdrückung des Störsignals bei  $f = 50$  Hz für  $f_{S,2} = 200$  Hz und  $f_{S,3} = 240$  Hz
- d) Pol/Nullstellen-Diagramm
- e) Phasengang  $\varphi(e^{j\Omega})$  und Gruppenlaufzeit  
 $\tau_g(e^{j\Omega})$  Welche Werte erhält man bei  $f = 50$  Hz und  $f_{S,2} = 200$  Hz?

## 2.2 Frequenzgang einfacher FIR-Filter → M2.2

Das FIR Filter aus Aufgabe 1.2 mit  $h[n] = \{0,5; 1\}$  soll jetzt in der Frequenzebene untersucht werden.

- a) Bestimmen Sie die Übertragungsfunktion  $H(z)$  und ihre Nullstellen und Pole. Zeichnen Sie diese in der komplexen  $z$ -Ebene ein.
- b) Bestimmen Sie den komplexen Frequenzgang  $H(e^{j2\pi F})$  sowie seinen Betrag und seine Phase. Berechnen Sie diese Werte für  $f = 0, 0,25f_S$  und  $0,5f_S$ .

## 2.3 Systemfunktion aus Impulsantwort → M2.3

Ermitteln Sie für die Systeme aus Aufgabe 1.3 die Übertragungsfunktionen  $H(z)$ , bestimmen Sie Pole / Nullstellen und zeichnen Sie sie auf.

Welches Frequenzverhalten erwarten Sie aufgrund der Lage der Pol- und Nullstellen? Überprüfen Sie Ihre Antwort mit einer Matlab / Python Simulation, nutzen Sie dazu den Befehl `freqz`.

## 2.4 Systemeigenschaften aus Übertragungsfunktion → M2.4

Die folgenden Systeme sind durch ihre Systemfunktion  $H(z)$  definiert:

- a)  $H_a(z) = \frac{(z - 1)(z + 1)}{z^2}$
- b)  $H_b(z) = z^{-1} + z^{-3}$
- c)  $H_c(z) = (z - 1)^3$
- d)  $H_d(z) = \frac{1}{z^2 + 0,64}$

**Bestimmen Sie jeweils**

- Kausalität und Stabilität: Wie kann man ggf. nichtkausale Systeme in kausale Systeme umwandeln?
- Impulsantwort
- Pole und Nullstellen: Wandeln Sie hierzu die Übertragungsfunktion in ihre Produktform um
- Betrags- und Phasengang: Versuchen Sie, einen Faktor  $e^{jx}$  so auszuklammern, dass sich die restlichen Terme zu Sinus- und Cosinusfunktionen zusammenfassen lassen.

**und zeichnen Sie:**

- Pol- / Nullstellenplan
- Betragsgang: Nutzen Sie hierfür auch den P/N-Plan und berechnen Sie  $H(f = 0)$ ,  $H(f = f_S/4)$  und  $H(f = f_S/2)$

- Eine Hardware-Implementierung

Überprüfen Sie die Ergebnisse auch mit Matlab / Python - Simulationen.

## 2.5 Systemeigenschaften aus Pol-/Nullstellenplan → M2.5

Schätzen Sie folgende Eigenschaften der in Abb. 2.2 durch ihre Pol/Nullstellenpläne gegebenen Systeme ab:

- FIR oder IIR - Filter
- Kausalität
- Stabilität
- Frequenzverhalten

Wandeln Sie instabile Systeme durch Streichen von möglichst wenigen Pol- und/oder Nullstellen in stabile Systeme um! Wandeln Sie nicht-kausale Systeme durch Streichen oder Ergänzen von Pol- oder Nullstellen in kausale Systeme um. Die Frequenzcharakteristik soll sich dabei nicht ändern.

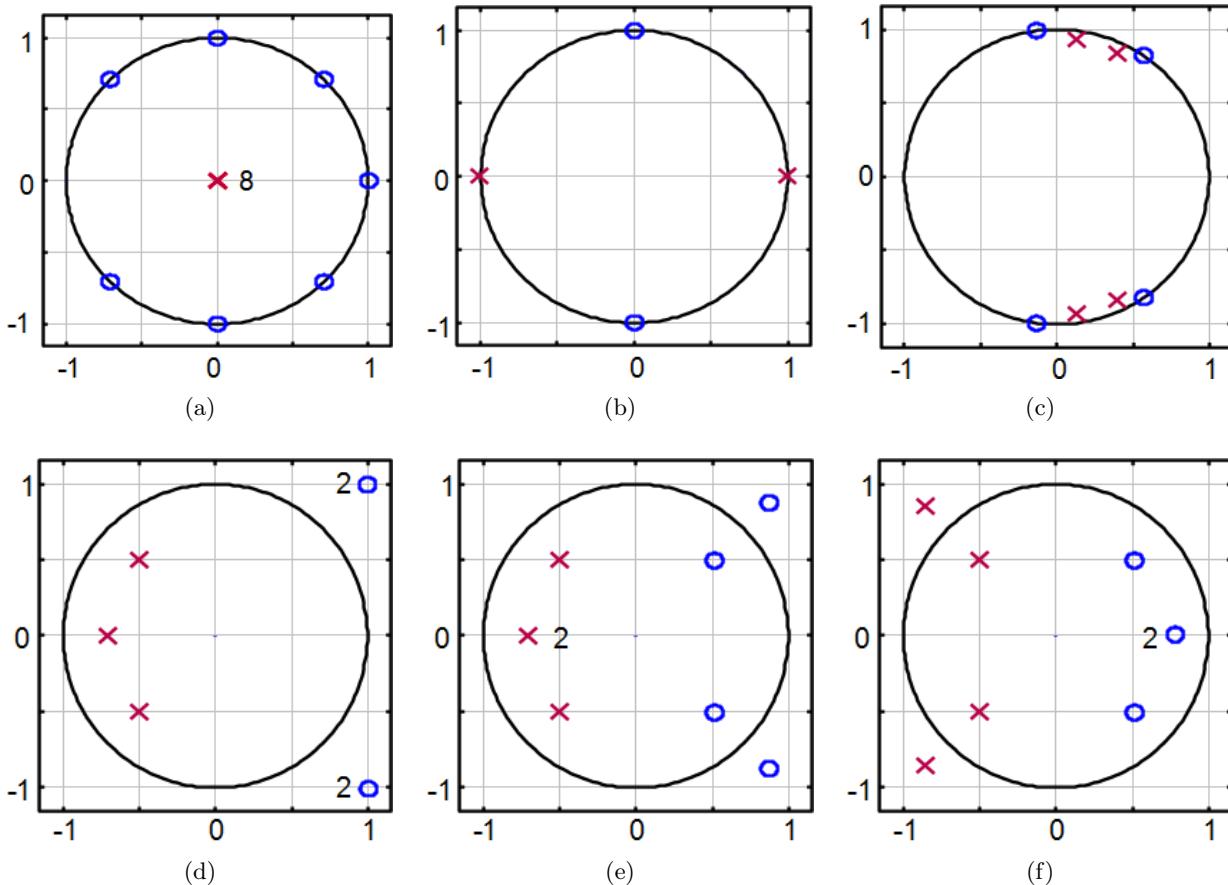


Abb. 2.2: Pol-/Nullstellenpläne zu Aufgabe 2.5

## 2.6 Frequenzgang aus Signalflussdiagramm → M2.6

Bestimmen Sie die für die Systeme in Abb. 1.6:

- Systemfunktion  $H(z)$  und Pol-/Nullstellen

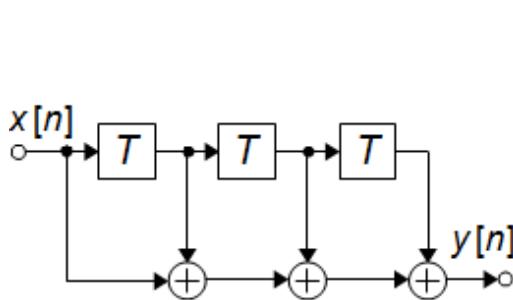
**2.** Charakteristik im Frequenzbereich (Hochpass, Tiefpass, ...)

Wenn Sie sich noch unsicher sind im Aufstellen der Systemfunktion aus einem SFG, können Sie als zusätzliche Übungsaufgaben versuchen, die Systemfunktion der transponierten Systeme (Abb. M1.10) zu bestimmen (die natürlich identisch sein muss mit der ursprünglichen Systeme).

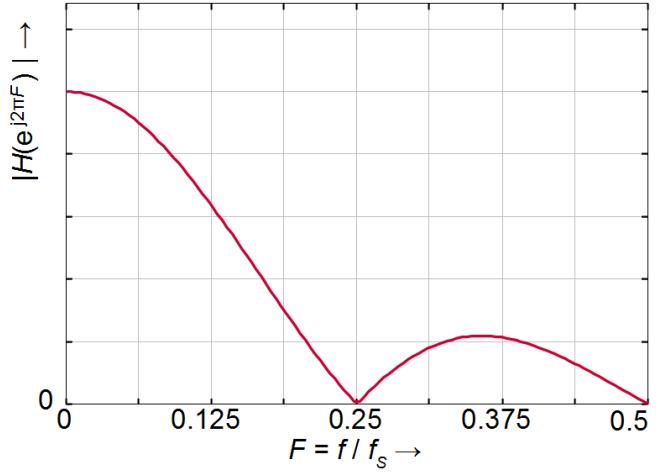
## 2.7 Einfaches Moving Average Filter → M2.7

Gegeben ist die im Abb. 2.3a gezeigte FIR-Filterstruktur.

- Ermitteln Sie Impulsantwort  $h[n]$  und Systemfunktion  $H(z)$  des Filters.
- Die Simulation ergibt den Amplitudengang  $|H(e^{j\Omega})|$  in Abb. 2.3b. Berechnen Sie  $H(f = 0)$  sowie die Nullstellen von  $H(z)$ .
- Ermitteln Sie den komplexen Frequenzgang  $H(e^{j\Omega})$  und den Amplitudengang  $|H(e^{j\Omega})|$
- Ermitteln Sie den Phasengang  $\varphi(e^{j\Omega})$  sowie die Gruppenlaufzeit  $\tau_g(e^{j\Omega})$ .
- Bestimmen Sie die Systemfunktion und den komplexen Frequenzgang für den Fall, dass die gegebene Filterstruktur auf  $L$  Taps (Ordnung  $N = L - 1$ ) erweitert wird. Der Vorfaktor wird dabei so gewählt, dass  $H(f = 0) = 1$  ist. Welchen Amplitudengang nähert man mit  $L \rightarrow \infty$  immer mehr an? Wozu könnte dieses Filter dienen?
- Schreiben Sie Python / Matlab Code zur Darstellung des Frequenzgangs eines MA-Filter mit  $N = 32$



(a)



(b)

Abb. 2.3: FIR-Filterstruktur (a) und Amplitudengang (b) zu Aufgabe 2.7

## 2.8 Linearphasiges FIR-Filter → M2.8

Ein FIR-Filter  $F$  wird als Kaskade zweier FIR-Filter  $F_1$  und  $F_2$  realisiert (Abb. 2.4).

- Haben die beiden Teil-Filter jeweils für sich allein einen **linearen Phasengang**?
- Geben Sie die **Impulsantworten**  $h_1[n]$  und  $h_2[n]$  sowie die **Übertragungsfunktionen**  $H_1(z)$  und  $H_2(z)$  der Teilträger an.

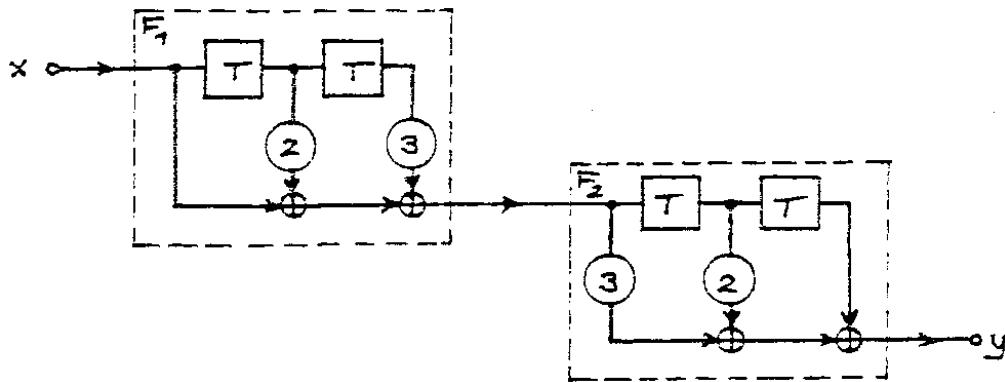
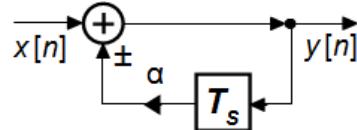


Abb. 2.4: FIR-Filter zu Aufgabe 2.8

- c) Bestimmen Sie die Nullstellen der Übertragungsfunktionen  $H_1(z)$  und  $H_2(z)$  und tragen Sie sie in einem PN-Diagramm ein. Wie groß ist  $H(f = 0)$ ?
- d) Bestimmen Sie die Impulsantwort  $h[n]$  des Gesamtfilters mit Hilfe der diskreten Faltung.
- e) Hat das Gesamtfilter  $F$  einen linearen Phasengang? Falls ja, wie groß ist die Gruppenlaufzeit  $\tau_g$  von  $F$ ?
- f) Berechnen Sie den Phasengang aus der Gruppenlaufzeit und skizzieren Sie im Bereich  $0 < f < f_S/2$  grob den Betragsgang des Filters mit Hilfe der Punkte  $H(f = 0)$ ,  $H(f = f_S/4)$  und  $H(f = f_S/2)$ .

## 2.9 Allgemeine IIR-Struktur → M2.9

Ausgehend von der nebenstehenden allgemeinen IIR-Struktur sollen die Eigenschaften der folgenden Varianten untersucht werden:



- a) Addierer, ein Verzögerungsglied  $T_S$
- b) Subtrahierer, ein Verzögerungsglied  $T_S$
- c) Addierer, zwei Verzögerungsglieder  $T_S$
- d) Subtrahierer, zwei Verzögerungsglieder  $T_S$

Die Eigenschaften sollen für alle vier Varianten jeweils für die Koeffizientenwerte  $\alpha = \alpha_1 = 0,9$  und  $\alpha = \alpha_2 = 0,5$  bestimmt werden.

1. Bestimmen Sie zunächst die Übertragungsfunktion  $H(z)$  der vier Varianten.
2. Geben Sie für jede Variante an bzw. skizzieren Sie:
  - den Filtertyp (TP, HP, BP, BS)
  - das Pol-Nullstellen-Diagramm
  - Minimum und Maximum der Betragsfunktion  $|H(e^{j2\pi F})|_{min}$  und  $|H(e^{j2\pi F})|_{max}$  und bei welcher Frequenz diese auftreten.
3. Skizzieren Sie den Verlauf der Betragsfrequenzgänge  $|H(e^{j2\pi F})|$  im Basisband  $0 \leq F \leq 0,5$  ohne  $H(e^{j2\pi F})$  formell abzuleiten

4. Leiten Sie die Betragsfrequenzgänge formelmäßig her
5. Skizzieren Sie die Impulsantworten, versuchen Sie die Impulsantworten zu berechnen [Systeme c) und d) nicht prüfungsrelevant]
6. Schreiben Sie ein Python / Matlab-Skript, um P/N-Diagramm, Frequenzgang und Impulsantwort darzustellen. Nutzen Sie hierzu die Befehle `zplane`, `freqz` und `impz`.

## 2.10 IIR-Filter zweiter Ordnung → M2.10

Gegeben ist das IIR-Filter in Abb. 2.5a.

- a) Geben Sie die **Differenzengleichung des IIR-Filters** an.

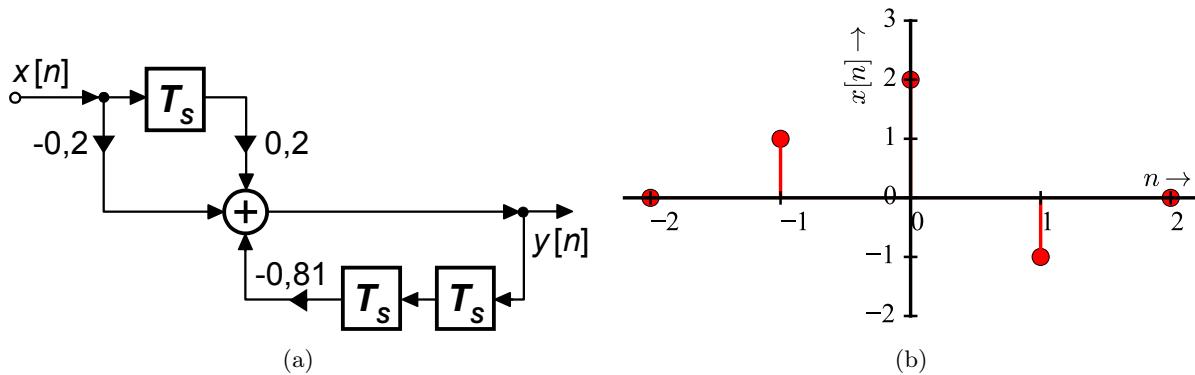


Abb. 2.5: IIR-Filter (a) mit Eingangssignal (b) zu Aufgabe 2.10

- b) Ermitteln Sie aus der Differenzengleichung die **Impulsantwort  $h[n]$**  für  $n = 0, \dots, 6$ .
- c) Ermitteln Sie für das in Abb. 2.5b skizzierte Eingangssignal  $x[n]$  das **Ausgangssignal  $y[n]$**  für  $n = 1, \dots, 5$  als Summe von bewerteten Impulsantworten und skizzieren Sie es.

Im Weiteren gelte eine Abtastperiode von  $T_S = 10 \mu\text{s}$ :

- d) Ermitteln Sie die **Übertragungsfunktion  $H(z)$  in Polynomform** und ihre Nullstellen und Pole. Zeichnen Sie diese in der komplexen z-Ebene ein. Um welche Art von Filter handelt es sich (TP, HP, BP, BS)?
- e) Ermitteln Sie **Betrag und Phase des komplexen Frequenzgangs  $H(e^{j2\pi f/f_s})$**  bei der Frequenz  $f = 25 \text{ kHz}$  direkt aus der Polynomform (nicht ausmultiplizieren!).
- f) Stellen Sie  **$H(z)$  in der Produktform** dar und berechnen Sie nochmals die im vorigen Punkt gefragten Größen.
- g) Wie müssen Sie die Filterkoeffizienten ändern, damit die Verstärkung bei 25 kHz den Betrag 1 hat, ohne dass sich dadurch der Verlauf des Betraggangs ändert?
- h) **Welche Phasenverschiebung und welche Gruppenlaufzeit** hat das System bei  $f = 25 \text{ kHz}$ ? Überlegen Sie theoretisch und simulieren Sie mit pyFDA, Matlab oder Python.

## 2.11 Allgemeines Moving Average Filter / rect-Fenster → M2.11

Es sollen die Eigenschaften von Moving Average Filtern mit der Länge  $L = 32$  und  $1024$  verglichen werden, die mit einer Taktrate von  $f_S = 10,24$  MHz betrieben werden. Betrachten Sie zunächst den allgemeinen Fall (Ordnung  $N = L - 1$ ) und nutzen Sie für die Rechnung die endliche geometrische Reihe (B.42) und den Dirichlet-Kern (B.43):

Bestimmen Sie jeweils:

- die Betragsantwort des Frequenzgangs  $|H(f)|$  und die Verstärkung  $|H(f = 0)|$
- die Frequenz, bei der der Frequenzgang die erste Nullstelle hat
- die Amplitude des größten Nebenzipfels relativ zum Wert bei  $f = 0$  (näherungsweise).

## 2.12 Notchfilter → M2.12

Ein mit  $f_S = 10$  kHz abgetastetes Audio-Signal enthält einen starken Störton bei  $f_1 = 3$  kHz, der vollständig unterdrückt werden soll. Bestimmen Sie für die folgenden drei Ansätze jeweils Übertragungsfunktion  $H(z)$ , Betragsgang  $|H(e^{j\Omega})|$  und dessen Größe bei den Frequenzen  $f = 0$ ,  $f = f_S/2$  und  $f = f_1 \pm 100$  Hz. überprüfen Sie die Rechnung jeweils mit Python.

- a) Entwerfen Sie durch Platzieren von Nullstellen ein einfaches reellwertiges FIR-Filter, das diese Frequenz vollständig unterdrückt. Welche minimale Ordnung hat das Filter? Welchen Nachteil hat dieses einfache Filter?
- b) Ersetzen Sie die einfachen Nullstellen durch doppelte Nullstellen. Werden die Nachteile von  $H_1$  dadurch vermieden?
- c) Durch Hinzufügen von Polen bei dem gleichen Winkel wie die Nullstellen lässt sich das Verhalten von  $H_1$  deutlich verbessern. Entwerfen Sie ein Notchfilter  $H_3$  mit zwei Null- und zwei Polstellen mit dem Polradius  $r = 0,95$ .

*An approximate answer to the right problem is worth a good deal more than an exact answer to an approximate problem.*

*John Tukey*

### 3 DFT: Diskrete Fouriertransformation und FFT

#### Lernziele

In diesem Kapitel wird die diskrete Fouriertransformation betrachtet von zeitdiskreten *Energie-signalen* (endliche Dauer und Energie) und *periodischen Signalen*. Abtastfrequenz  $f_S$  und Anzahl  $N_{DFT}$  der DFT-Punkte werden hier so gewählt, dass bei Energiesignalen alle Datenpunkte aufgenommen bzw. bei periodischen Signalen die erfassten Datenpunkte fehlerfrei periodisch fortgesetzt werden können (kohärente Abtastung). Falls die Folge durch Abtastung eines zeitkontinuierlichen Signals (begrenzt auf die Nyquist-Bandbreite) entstanden ist, kann man dessen Spektrum aus den Werten der DFT im Frequenzbereich bestimmen.

- Sie kennen den Unterschied zwischen Fourierreihe bzw. Continuous Fourier Transform (**CFT**), Discrete Time Fourier Transform (**DTFT**), Discrete Fourier Transform (**DFT**) und Fast Fourier Transform (**FFT**) und nennen Beispiele für deren Anwendung.
- Sie wissen, welche der obigen Transformationen **numerisch berechnet** werden kann.
- Sie berechnen die **Frequenzauflösung**  $\Delta f$  der DFT abhängig von Abtastfrequenz  $f_S$  und Anzahl der Datenpunkte  $N_{DFT}$  und **skalieren** die DFT, um korrekte Frequenz- und Amplitudenwerte für das Spektrum des ursprünglichen zeitkontinuierlichen Signals zu erhalten.

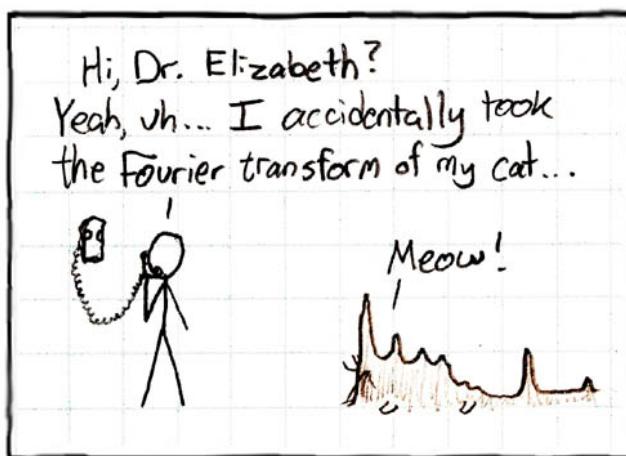


Abb. 3.1: „That cat has some serious periodic components“. [<http://xkcd.com/26/>]

## Theorie

### Fouriertransformationen zeitkontinuierlicher Signale und deren Energie bzw. Leistung

Wenn die Fouriertransformation (und ihre Verwandten) zur Messung und Analyse physikalischer Vorgänge verwendet wird, muss man die Grenzen der Methoden beachten, um keine Fehler zu machen. Oft ist man an der Energie bzw. Leistung eines Signals interessiert, dann ist es sinnvoll so zu skalieren, dass sie im Zeit- und Frequenzbereich gleich sind.

**Transiente Signale** (mit endlicher Zeitspanne) haben endliche Energie<sup>1</sup>, sie werden daher Energiesignale genannt. Das Fourierintegral beschreibt das komplexe Spektrum eines Energiesignals:

$$Y(f) = \int_{-\infty}^{\infty} y(t) e^{-j2\pi f t} dt \quad \bullet \circ \quad y(t) = \int_{-\infty}^{\infty} Y(f) e^{j2\pi f t} df \quad (3.1)$$

Die *Energy Spectral Density* (ESD) beschreibt die Energie pro Bandbreite des Signals:

$$S_{yy}(f) = Y(f) * Y^*(f) = |Y(f)|^2 \quad (3.2)$$

Daraus folgt die Äquivalenz der *Energie* im Zeit- und Frequenzbereich (Parsevalsche Gesetz):

$$E = \int_{-\infty}^{\infty} |y(t)|^2 dt = \int_{-\infty}^{\infty} |Y(f)|^2 df \quad (3.3)$$

**Periodische Signale** sind unendlich ausgedehnt und haben daher unendliche Energie. Bei solchen Signalen „stanzt“ man einen Teil des Signals aus<sup>2</sup>,  $y_T(t) = y(t)$  für  $|t| < T_1/2$ , 0 sonst, und rechnet weiter mit dem ausgeschnittenen Signal („Truncated“). Wenn die Länge  $T_1$  ein ganzzahliges Vielfaches der Periodendauer ist, beschreibt das Fourierintegral exakt das Signalspektrum, ein diskretes Linienspektrum mit komplexen Koeffizienten  $c_k \equiv Y[kf_1]$ :

$$Y[kf_1] \equiv c_k = \frac{1}{T_1} \int_{-T_1/2}^{T_1/2} y(t) e^{-j2k\pi f_1 t} dt; \quad k \in \mathbb{Z} \quad \bullet \circ \quad y(t) = \sum_{k=-\infty}^{\infty} Y[kf_1] e^{j2\pi k f_1 t} \quad (3.4)$$

Die Äquivalenz der *Leistung* im Zeit- und Frequenzbereich wird durch den Satz von Plancherel beschrieben:

$$P = \frac{1}{T_1} \int_{-\infty}^{\infty} |y_T(t)|^2 dt = \frac{1}{T_1} \int_{-\infty}^{\infty} |\mathcal{F}(y_T(f))|^2 df \quad (3.5)$$

Wenn der Signalausschnitt keine ganze Zahl Perioden erfasst und/oder Anfang und Ende mit einer „weicherem“ Fensterfunktion ausgeblendet werden, liefert die Fouriertransformation nur eine Schätzung des Spektrums und der Leistung. Das ist Thema von Kap. 4.

Für **statistische Prozesse** ist die Fouriertransformation nicht definiert, hier betrachtet man stattdessen die Fouriertransformierte der Autokorrelationfunktion  $\gamma(\tau)$  des Prozesses (Wiener-Kinchine-Theorem). Dieses Thema wird in Kap. 4 behandelt [https://en.wikipedia.org/wiki/Spectral\\_density](https://en.wikipedia.org/wiki/Spectral_density).

$$S_{yy}(f) = \int_{-\infty}^{\infty} \gamma(\tau) e^{-j2\pi f \tau} d\tau \quad \bullet \circ \quad \gamma(\tau) = \int_{-\infty}^{\infty} S_{yy}(f) e^{j2\pi f \tau} df \quad (3.6)$$

<sup>1</sup>jedenfalls „real world“ Signale

<sup>2</sup>Technischer ausgedrückt: Man fenstert das Signal mit einem Rechteckfenster.

## Rechenaufwand für DFT und FFT

Aus der Definitionsgleichung der DFT (C.41) bzw. deren SFG (Abb. 3.2) kann man sofort ablesen, dass die Berechnung von  $N_{DFT}$  Frequenzpunkten aus  $N_{DFT}$  Zeitpunkten im allgemeinen Fall  $N_{DFT}^2$  komplexe oder  $4N_{DFT}^2$  reellwertige Multiplikationen und Additionen (= MAC) erfordert<sup>3</sup>. Bei reellwertigen Eingangssignalen sind nur die Hälfte der Rechenoperationen erforderlich. Nutzt man die Symmetrie des Spektrums aus, muss man nur  $N/2$  Frequenzpunkte berechnen, so dass nur noch

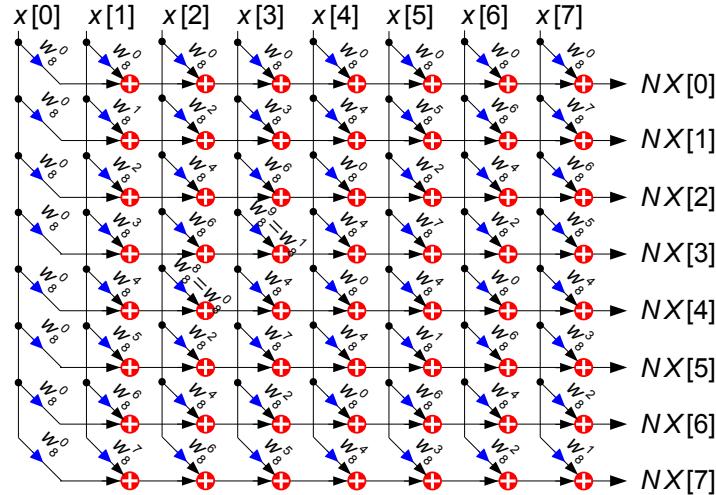


Abb. 3.2: Signalflossgraph einer DFT mit 8 Punkten

$$N_{MAC,DFT} \approx N_{DFT}^2 \text{ reellwertige Rechenoperationen} \quad (3.7)$$

erforderlich sind. In dieser Abschätzung ist nicht berücksichtigt, dass ein Teil der Rechenoperationen trivial ist (z.B. Multiplikation mit  $\pm 1$ ).

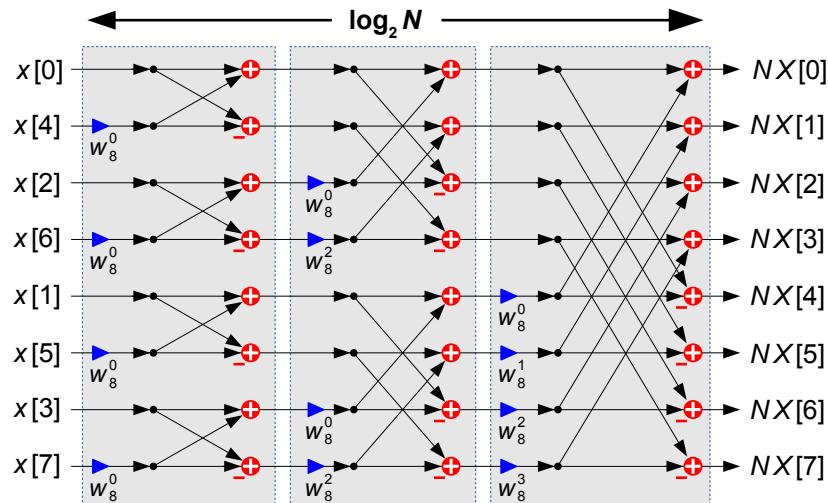


Abb. 3.3: Signalflossgraph einer FFT mit 8 Punkten

Wenn die Anzahl der Datenpunkte z.B. eine Zweierpotenz ist,  $N_{FFT} = 2^m$ , kann die DFT in kurze Teil-DFTs zerlegt und sehr effizient als sog. *Radix-2 FFT* (Fast-Fourier Transform) berechnet werden, die nur noch ca.

$$N_{MAC,FFT} \approx N_{FFT} \log_2 N_{FFT} \text{ reellwertige Multiplikationen} \quad (3.8)$$

<sup>3</sup>Multiplikation und Addition werden als eine Operation gerechnet, da DSPs, FPGAs und die meisten uCs Rechenwerke haben, die Multiplikation und Addition in einem Schritt durchführen können.

benötigt. Die genaue Anzahl der Multiplikationen hängt vom Algorithmus ab (Radix-4, Split Radix, Winograd, Chirp-z,...) und davon, wie gut sich  $N_{FFT}$  in kleine Primfaktoren zerlegen lässt. Die Anzahl der Additionen wurde der Einfachheit halber hier vernachlässigt, obwohl i.A. 3 ... 4 mal so viele wie Multiplikationen berechnet werden müssen.

Der SFG in Abb. 3.3 lässt sich sehr gut mit Pipeliningregistern zwischen den Stufen implementieren, so dass ein sehr hoher Datendurchsatz erreicht werden kann. Die Beschränkung auf reellwertige Eingangssignale spart hier allerdings kaum Rechenoperationen ein, da die zu verarbeitenden Zwischen signale überwiegend komplex sind. Immerhin müssen nur halb so viele Frequenzpunkte abgespeichert werden. In modernen Rechnerarchitekturen dominieren inzwischen oft die Zeiten für das Verschieben und Abspeichern der Daten.

Eine ausführlichere Darstellung von verschiedenen FFT Algorithmen finden Sie z.B. in <http://www.cmlab.csie.ntu.edu.tw/cml/dsp/training/coding/transform/fft.html>

## Spezielle Python-Funktionen

Funktion	Python	Matlab	Beschreibung
<b>FFT</b>	<code>X = fft(x, NFFT)</code>	<code>X = fft(x, NFFT)</code>	Berechne DFT X mit <b>NFFT</b> komplexen Werten je Dimension zu <b>NFFT</b> reellen oder komplexen Werten x (ein- oder mehrdimensional)
<b>IFFT</b>	<code>x = ifft(X, NFFT)</code>	<code>x = fft(X, NFFT)</code>	Berechne inverse DFT x mit <b>NFFT</b> komplexen Werten je Dimension zu <b>NFFT</b> Werten X (ein- oder mehrdimensional)
	<code>X = fftr(x, NFFT)</code>	- - -	Berechne DFT zu <i>reeller</i> Zeitfolge x (nur numpy)
<b>Zweiseitiges Spektrum</b>	<code>XS = fftshift(X)</code>	<code>XS = fftshift(X)</code>	Verschiebe DFT so, dass der Spektralwert zu $f = 0$ in der Mitte liegt.
	<code>X = ifftshift(XS)</code>	<code>X = ifftshift(XS)</code>	Inverse Operation zu <code>fftshift</code> .
<b>Frequenzvektor</b>	<code>f = fftfreq(NFFT, TS)</code>	- - -	Array mit <b>NFFT</b> Frequenzpunkten passend zu den Punkten der FFT (zweiseitiges Spektrum)
	<code>f = rfftfreq(NFFT, TS)</code>	- - -	Array mit <b>NFFT</b> /2 Frequenzpunkten passend zu den Punkten FFT (reellwertige Sequenz, einseitiges Spektrum)

Tab. 3.1: Spezielle Befehle zu Kap. 3

### Anmerkungen zu Tab. 3.1:

„FFT“ ist in vielen Programmiersprachen (so auch in Python und Matlab) ein Platzhalter für eine Reihe von Algorithmen, aus denen je nach Randbedingungen (Anzahl der Punkte, reell/komplex etc.) der am besten geeignete ausgewählt wird. Der Befehl kann daher mit einer beliebigen Anzahl Punkte (nicht nur Zweierpotenzen) verwendet werden, auch wenn es für  $N_{FFT} = 2^m$  besonders effiziente Algorithmen gibt.

„FFT“ wird oft gleichbedeutend mit „DFT“ verwendet, auch wenn das nicht ganz korrekt ist: DFT beschreibt die mathematische Methode / Transformation, FFT ist eine effizientere Algorithmus zur Berechnung der DFT.

**Python** Funktionen rund um DFT/FFT finden sich in den `fft` bzw. `fftpack` Submodulen von Numpy und Scipy, der Befehl `xxx` wird daher mit `import numpy.fft.xxx` bzw. `import scipy.fftpack.xxx` importiert. Aus Kompatibilitätsgründen werden beide Bibliotheken gepflegt. Anscheinend nutzt Scipy die Original-Fortran Routinen der FFTPACK Bibliothek, Numpy verwendet eine C Portierung. Scipy prüft ob die Daten reellwertig sind und nutzt dann automatisch die effizientere `fftr` ([Stack Overflow Frage 6363154](#)). Das FFT Submodul von Scipy ist umfangreicher als das von Numpy und auch besser dokumentiert, ich empfehle daher die Scipy - Routinen zu verwenden.

**Frequenzvektor:**  $f = [0, \dots, fs/2[, [-fs/2, \dots, 0]$ . Die  $N_{FFT}$  Werte des Frequenzvektors werden also beginnend bei  $f = 0$  ausgegeben. Für ein einseitiges Spektrum zwischen  $[0, \dots, fs/2[$  nimmt man lediglich die erste Hälfte der Punkte, also  $S1 = S[0:N_{FFT}/2 -1]$ . Für eine zweiseitige Darstellung symmetrisch zu  $f = 0$  muss die zweite Hälfte des Vektors mit der ersten vertauscht werden. Dies erledigt der Befehl `fftshift`.

Matlab hat keine Funktion `fftfreq`, hier nimmt man einfach `f = Fs/2*linspace(0,1,NFFT/2+1)` . (Das geht natürlich in Python auch.)

## Notebooks und Labs

Filename	Beschreibung
<b>Notebooks</b>	
DFT-Coherent	Kohärente Abtastung (Periodizität) und DFT
DFT-Skalierung	Wie interpretiert und skaliert man die DFT korrekt?
DFT-basics_ML	Musterlösung zu Aufgabe 3.2 „Fourierreihe und kohärente DFT“
DFT-MA	Code zu Aufgabe 3.3 „Frequenzgang eines Moving Average (MA) - Filters“
DFT-MA-ML	Musterlösung zu Aufgabe 3.3 „Frequenzgang eines MA - Filters“
DFT-Periodische_Signale	Code zu Aufgabe 3.6 „DFT periodischer Signale mit Python“

Tab. 3.2: Jupyter Notebooks zu Kap. 3

## Zusätzliche Literatur

Bitte schauen Sie sich bei Bedarf die folgenden Videos (deutsch) von Jörg Lovisach auf Youtube an, der sehr schön die Grundlagen von Fourier-Transformation, DFT und FFT erklärt:

**01.04.1 Fourier-Reihe:** <https://www.youtube.com/watch?v=g5Y25jsliFk>

**01.04.2 Fourier-Transformation:** <https://www.youtube.com/watch?v=ug9Co75tQuo>

**19.03.1 Diskrete Fouriertransformation:** <https://www.youtube.com/watch?v=ls5MfqLZVbo>

### 3.1 Allgemeine Fragen zu DFT und FFT → M3.1

- a) Nennen Sie die **Unterschiede** zwischen Fourierreihe, Fouriertransformation, DTFT, DFT und FFT?
- b) Welcher **Frequenz  $f_k$  entspricht der Frequenzpunkt  $k$ ?** Wie groß ist der Abstand  $\Delta f$  zwischen zwei Frequenzpunkten einer DFT mit  $N$  Punkten und der Abtastfrequenz  $f_s$ ? Welcher Frequenz entspricht der höchste Frequenzpunkt?
- c) Warum wird die DFT / FFT meist nur in der **positiven Hälfte des Nyquistbereichs** ( $0 \leq f < f_s/2$ ) ausgewertet?
- d) Warum benötigt man *immer* eine Fensterung vor Berechnung einer DFT / FFT?

### 3.2 # Fourierreihe und synchrone DFT → M3.2

Diese Aufgabe behandelt eigentlich Stoff des Grundstudiums; Sie sollten Sie dann lösen wenn Sie Verständnis- und Rechenschwierigkeiten mit der Fouriertransformation haben.

Das periodische zeitkontinuierliche Signal  $s(t) = 1 \text{ V} \cdot (1 + \cos(2\pi 1000 \text{ Hz } t + \pi/4))$  wird dreimal pro Periode abgetastet (das Abtasttheorem ist also erfüllt). Die erste Abtastung erfolgt bei  $t = 0$ .

- a) Berechnen und skizzieren Sie zunächst **Betrag und Phase der Fourierkoeffizienten  $c_k$**  des zeitkontinuierlichen Signals  $s(t)$  (Continuous Fourier Transform, CFT).
- b) Berechnen Sie jetzt die **ersten drei Werte des abgetasteten Signals  $s[n]$** . Welchen Wert hat die Abtastperiode  $T_S$ ?
- c) Berechnen und skizzieren Sie **Betrag  $|S[k]|$  und Phase  $\Phi_S[k]$  der DFT** dieser drei Abtastwerte  $s[n]$ , vergleichen Sie das Ergebnis mit der CFT.
- d) **Wieviele Elemente** hat die DFT? Wie ordnet man den **Index  $k$**  zu Frequenzwerten zu?
- e) Was ändert sich, wenn Sie die **DFT der ersten 6 Abtastwerte** berechnen (gleiche Abtastfrequenz wie vorher)?
- f) Wie sieht die DFT aus, wenn Sie stattdessen **6 Abtastwerte pro Periode** nehmen?
- g) Wie sieht die DFT aus, wenn Sie stattdessen die ersten drei Abtastwerte durch **Anhängen von drei Nullen** („zeropadding“) auf 6 Abtastwerte verlängern? Wie müssen Sie den Skalierungsfaktor für die DFT wählen?
- h) **Wieviele Rechenoperationen** sind allgemein bei  $N$  reellen Abtastwerten zur Berechnung der DFT notwendig?
- i) Verifizieren Sie Ihre Rechnungen durch Simulation in **Python / Matlab**, nutzen Sie hierfür den Befehl `fft(x, N_FFT)`.

---

```

1 h = [1,1,1,1]
2 n = np.arange(len(h))
3 plt.figure(1);
4 plt.subplot(211)
5 plt.stem(n,h)
6 plt.ylabel(r"$h[n] \rightarrow$")
7 plt.xlabel(r"$n \rightarrow$")
8 plt.subplot(212)
9 H = np.fft.fft(h,4)
10 k = np.arange(len(H))
11 plt.stem(k,abs(H))
12 plt.xlabel(r"$k \rightarrow$")
13 plt.ylabel(r"$|H[k]| \rightarrow$")
14 plt.tight_layout()
15 plt.show()

```

---

Lst. 3.1: Python Skript zu Aufgabe 3.3 (erster Ansatz, `DFT_MA_Filt.py`)

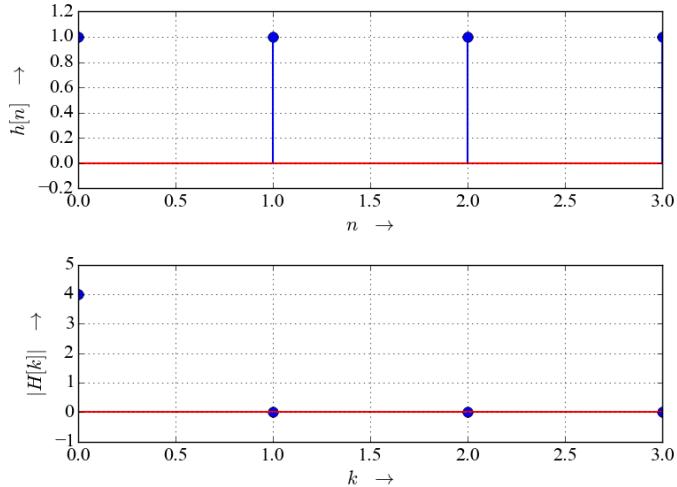


Abb. 3.4: Python Plot zu Listing 3.1

### 3.3 Frequenzgang eines Moving Average (MA) - Filters → M3.3

Zu dieser Aufgabe gibt es die Jupyter Notebooks `DFT-MA.ipynb` und `DFT-MA-ML.ipynb`.

Sie wollen aus der Impulsantwort  $h_{MA,3} = \{1; 1; 1; 1\}$  eines MA-Filters der Ordnung  $N = 3$  dessen Frequenzgang mit Hilfe der DFT bestimmen. Welches Ergebnis erwarten Sie? Sie versuchen dies mit dem Python-Skript in Listing 3.1 und erhalten das dort gezeigte Ergebnis:

Zur Verbesserung der Darstellung versuchen Sie:

- `h = [1,1,1,1,0,0,0,0]; fft(h,8)`
- `fft(h,8)`
- `fft(h,256)`

Interpretieren Sie die Ergebnisse! Entspricht eines davon Ihrer Erwartung?

### 3.4 Frequenzauflösung der DFT → M3.4

- Was versteht man unter **Frequenzauflösung**?
- Ein Audiosignal soll mit einer FFT mit  $N = 2048$  Punkten und einer Frequenzauflösung von  $\Delta f = 1$  Hz gemessen werden. Wie groß muss die **Abtastfrequenz**  $f_S$  gewählt werden? Wie groß ist die **höchste Frequenz**, die noch dargestellt werden kann?
- In Abb. 3.5 ist das Seismogramm eines Erdbebens dargestellt, das mit einer Radix-2 FFT spektral analysiert werden soll. Die FFT-Länge kann zwischen  $2^8$  und  $2^{15}$  gewählt werden, die Abtastrate binär gestuft zwischen  $f_S = 0,1 \dots 25,6$  Hz (mit jeweils passendem Anti-Aliasfilter). Interessierende Spektralkomponenten liegen im Bereich  $0 \dots 1$  Hz.

Welche Kombination aus FFT-Länge und Abtastrate liefert Ihnen die „besten“ **spektralen Informationen**, welche Kombination ist am effizientesten unter den angegebenen Randbedingungen?

- Welche **Frequenz hat ein Störer**, der bei einer FFT-Länge  $N = 2^{11}$  und einer Abtastrate  $f_S = 1,6$  Hz eine starke Störlinie bei Frequenzpunkt  $k = 620$  liefert?

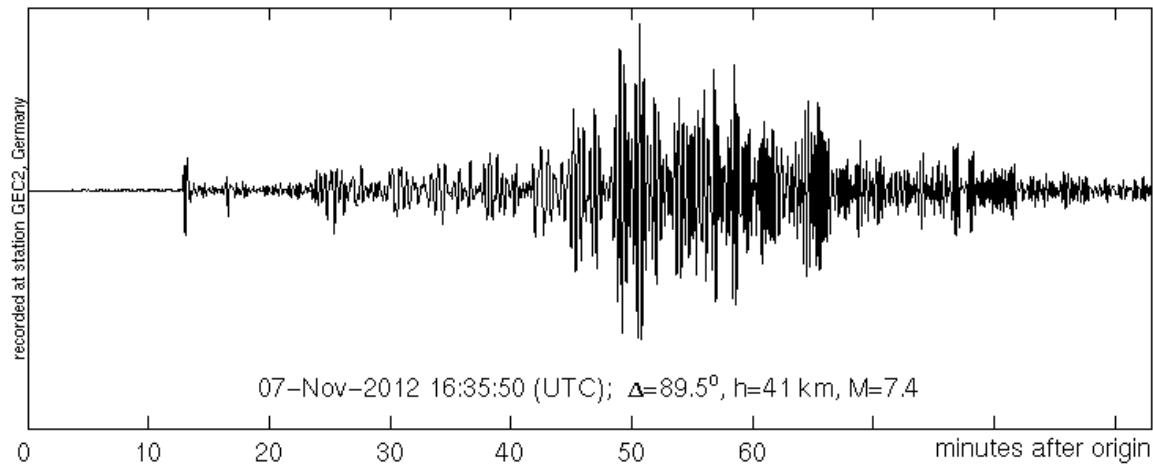


Abb. 3.5: Seismogramm eines Erdbebens am 7.11.2012 in Guatemala. Mit freundlicher Genehmigung der Bundesanstalt für Geowissenschaften und Rohstoffe  
[\[http://www.seismologie.bgr.de/sdac/erdbeben/big\\_quakes/guatemala\\_121107\\_deu.html\]](http://www.seismologie.bgr.de/sdac/erdbeben/big_quakes/guatemala_121107_deu.html)

### 3.5 Rechenaufwand für DFT und FFT → M3.5

Die 9. Symphonie von Beethoven wurde für eine CD mit der Laufzeit  $T_0 = 74$  min und der Abtastrate  $f_S = 44,1$  kHz aufgenommen. Verwenden Sie im folgenden (3.7) und (3.8) zur Berechnung der Anzahl der Rechenoperationen bei DFT bzw. FFT:

- Welche DFT Länge  $N_{DFT}$  benötigt man, um die Aufnahme vollständig als DFT zu kodieren und welche Frequenzauflösung  $\Delta f$  erhält man dabei?
- Wie viele reellwertige Rechenoperationen  $N_{MAC}$  wären zur Berechnung der DFT, wieviele zur Berechnung der FFT notwendig? Bedenken Sie dabei, dass die CD in Stereo aufgenommen wurde.
- Welche Rechenzeit  $T_{DFT}$  bzw.  $T_{FFT}$  erfordern die Berechnungen auf einem FPGA mit 512 Hardware - MAC - Units mit jeweils 500 MMAC/s? Die Zeit zum Speichern und Laden der Daten soll vernachlässigt werden.
- Welche Rechenleistung  $R_{MAC}$  in MAC/s ist mindestens erforderlich für eine Real-Time FFT der CD? Nehmen Sie hierfür Fensterlängen von  $N_{FFT,1} = 512$  und  $N_{FFT,2} = 8192$  an.

### 3.6 DFT periodischer Signale mit Python → M3.6

In Python soll mit einer DFT das Betragsspektrum  $|S_y(f)|$  eines Signals  $y(t)$  mit

$$y(t) = 1 + 0,5 \sin(2\pi t \cdot 1 \text{ kHz}) + 0,2 \cos(2\pi t \cdot 1,1 \text{ kHz})$$

näherungsweise bestimmt werden.

---

```

1 f_S = 51e3; T_S = 1. / f_S
2 N_FFT = 510; t_max = N_FFT * T_S
3 f_a = 1e3; f_b = 1.1e3
4 t = np.arange(0, t_max, T_S)
5 y = 1 + 0.5 * np.sin(2*np.pi*t*f_a) + 0.2 * np.cos(2*np.pi*t*f_b)
6 Sy = np.fft.fft(y,N_FFT)/N_FFT
7 f = np.arange(N_FFT)
8 plt.figure(1)
9 plt.stem(t, y)

```

---

---

```
10 plt.figure(2)
11 plt.stem(f, np.abs(Sy)); plt.show()
```

---

- a) Schreiben Sie das **erwartete Betragsspektrum**  $|S_y(f)|$  auf (ausführliche Berechnung der CFT ist nicht notwendig - genaues Hinschauen reicht!) und skizzieren Sie es.
- b) **Simulieren Sie das Betragsspektrum**  $|S_y(f)|$  mit Hilfe von Listing 3.2 nachdem Sie die Parameter an diese Aufgabe angepasst und ggf. Fehler beseitigt haben.
- c) Was muss gegeben sein, damit das Spektrum **keinen Leckeffekt** zeigt, d.h. damit die erwarteten Spektrallinien sauber dargestellt werden? Wie müssen Sie dazu die Parameter der DFT anpassen? Passen Sie die Frequenzachse so an, dass das Spektrum nicht über den Indices, sondern über den passenden physikalischen Frequenzen dargestellt wird.
- d) Wie können Sie die **Frequenzauflösung der DFT** verbessern? Ändern Sie die Parameter der DFT so, dass Sie die Spektrallinien bei  $f_a$  und  $f_b$  klar voneinander unterscheiden können.
- e) **Skalieren Sie DFT** jeweils so, dass die Amplituden für ein- und zweiseitiges Spektrum korrekt dargestellt werden. Überprüfen Sie die Skalierung durch Vergleich (analytische Rechnung und Simulation) der Energie bzw. Leistung im Zeitbereich und im Frequenzbereich. Welche Gesamtleistung hat das Signal?
- f) Wie müssen die Parameter geändert werden, damit eine **FFT mit schnellem Radix-2 Algorithmus** mit  $N_{FFT} = 2^m$  Punkten verwendet werden kann?

### 3.7 + Kohärente DFT einer Rechteckschwingung → M3.7

Gegeben ist eine zeitkontinuierliche (CT), mittelwertfreie symmetrische Rechteckschwingung mit Grundfrequenz  $f_1 = 1/T_1 = 1 \text{ kHz}$  und Amplitude  $A = 1 \text{ V}$ :

$$u(t) = \begin{cases} -A & \text{für } -T_1/2 \leq t < 0 \\ +A & \text{für } 0 \leq t < T_1/2 \end{cases} \quad \text{mit } u(t + T_1) = u(t)$$

Die Fourierkoeffizienten der CT-Funktion lauten

$$c_{2k+1} = \frac{2A}{\pi} \cdot \frac{1}{2k+1} \quad \text{und } c_{2k} = 0 \text{ mit } k = 0, 1, 2, \dots \quad \text{für } f_k = \frac{k}{T_1} = kf_1$$

- a) Wie lauten die **ersten 4 Fourierkoeffizienten**  $c_k$  der CFT?
- b) Leiten Sie die Gleichung für die Fourierkoeffizienten der Rechteckschwingung mittels (3.4) selbst her [optional].
- c) Bestimmen Sie die **Koeffizienten der DFT**, wenn die CT Rechteckfunktion mit 4 kHz und einer Fensterlänge von 1 ms abgetastet wird.
- d) Was könnte der Grund für die **Abweichungen zu den Koeffizienten der Fourierreihe** des CT-Signals sein? Simulieren Sie die DFT für  $f_S = 100 \text{ kHz}$  und  $N = 100$ .

## 4 WIN: Fensterung periodischer und stationärer Signale

### Lernziele

Aufbauend auf Kap. 3 beschäftigt sich dieses Kapitel mit der Fouriertransformation zeitdiskreter periodischer und stationärer Signale. Im Gegensatz zu letztem Kapitel ist die Länge des Fensters hier nicht auf ganzzahlige Vielfache der Periodendauer beschränkt.

- Sie erklären warum **nicht-kohärente Abtastung** eines periodischen Signals (Fensterlänge ist eine nicht-ganzzahlige Anzahl Perioden) Fehler erzeugt im Vergleich zur Fourierreihe.
- Sie erklären, was der **Frequenzfehler** und der **Leckeffekt** sind und unter welchen Bedingungen sie auftreten.
- Sie erklären, unter welchen Bedingungen der „**Lattenzauneffekt**“ oder **Amplitudenfehler** auftritt.
- Sie wissen, wie man durch Wahl eines geeigneten **Fensters** die oben genannten Effekte verringert. Sie kennen Eigenschaften und Kennzahlen verschiedener Fenster.
- Sie passen durch **Zeropadding** die Zahl der Datenpunkte für eine FFT so an, dass sich die Rechenzeit verringert. Sie skalieren eine Sequenz mit angehängten Null so, dass das Bezugsspektrum korrekt skaliert ist. Durch die höhere Anzahl Punkte werden Zwischenwerte im Spektrum berechnet (Interpolation in der Frequenzebene) und so der Amplitudenfehler verringert.

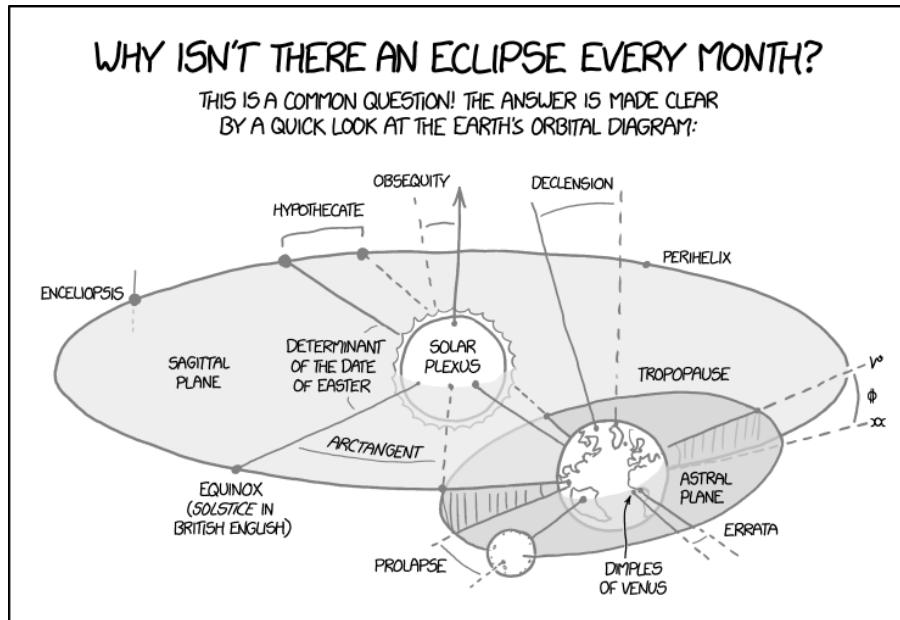


Abb. 4.1: Warum gibt es so viele verschiedene Fenster? [<https://xkcd.com/1878/>]

## Zusätzliche Literatur

Bitte schauen Sie sich bei Bedarf die Videos (deutsch) von Jörg Lovisach auf Youtube an, der sehr schön die Grundlagen der Fensterung erklärt:

**DFT mit Fensterung (19.03.2):** <https://www.youtube.com/watch?v=wd6hAPS4ILc>

**FFT in MATLAB(R) und Fensterung (17.3 ):** [https://www.youtube.com/watch?v=7bi5\\_KC08Kg](https://www.youtube.com/watch?v=7bi5_KC08Kg)

Detailliert befassen sich die folgenden Bücher und Veröffentlichungen mit Fensterung für die Spektralanalyse:

### Wikipedia

- *Fensterfunktionen*, <https://de.wikipedia.org/wiki/Fensterfunktion>. Unfangreicher Vergleich, man sieht sehr schön welchen großen Einfluss minimale Unterschiede in der Zeitfunktion auf die spektrale Performance haben.
- *Spectral leakage*, [https://en.wikipedia.org/wiki/Spectral\\_leakage](https://en.wikipedia.org/wiki/Spectral_leakage). Sehr informativer Eintrag für den Einstieg.

**Kammeyer / Kroschel [KK18]**, *Digitale Signalverarbeitung - Filterung und Spektralanalyse*

**G. Heinzel, A. Rüdiger and R. Schilling [HRS02]**, *Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows*, Feb. 15, 2002, <http://edoc.mpg.de/395068>. Sehr umfassender und gut erklärter Überblick über Fensterfunktionen.

**Fred Harris [Har08]**, *WINDOWS: Finite Aperture Effects and Applications in Signal Processing*. Das klassische Paper zur Fensterung, das verschiedene Einsatzgebiete von Fensterfunktionen beleuchtet.

**Svend Gade and Henrik Herlufsen, Brüel & Kjær**:

- *Use of Weighting Functions in DFT/FFT Analysis (Part I)*, Technical review No. 3 - 1987, S. 1 ff, <https://www.bksv.com/media/doc/bv0031.pdf>.  
*Signals and Units*, ibid., S. 29 ff.
- *Use of Weighting Functions in DFT/FFT Analysis (Part II)*, Technical review No. 4 - 1987, S. 1 ff, <https://www.bksv.com/media/doc/bv0032.pdf>

**H. Konstantin-Hansen, J. Wismer, N. Thrane, S. Gade, Brüel & Kjær**, *Choose your Units!*, Application Note BO438, 1997, <https://www.bksv.com/media/doc/bo0438.pdf>. Kurze Zusammenfassung der obigen B & K Artikel über die Einheiten Power, Power Spectral Density und Energy Spectral Density

**Gaussianwaves**, Überblick über Fensterfunktionen mit Matlab Code, <https://www.gaussianwaves.com/2011/02/window-functions-an-analysis/>

**Julius O. Smith [Smi11]**, *Spectral Audio Signal Processing*, [https://www.dsprelated.com/freebooks/sasp/.\]](https://www.dsprelated.com/freebooks/sasp/.)

Interessante Apps rund um das Thema spektrale Analyse und Synthese:

**Spectroid**: Real-Time Spektralanalyse mit wählbaren Fenstern etc.

**AudioTool**: Audioanalyse im Zeit- und Frequenzbereich, <https://play.google.com/store/apps/details?id=com.julian.apps.AudioTool>

**Phonopaper**, eine App zur Umwandlung eines Spektrogramms zu Musik (!): <http://warmplace.ru/soft/phonopaper/> <https://www.youtube.com/watch?v=d7AdeDyBOZY>

**Virtual ANS Synthesizer**, Eine Software Simulation eines russischen mikrotonal-spektralen Synthesizers  
<http://www.warmplace.ru/soft/ans/> mit Video-Demos

## Theorie

### Fensterfunktionen

Wie bereits in Kap. 3 beschrieben, „stanzen“ wir vor der DFT  $N_{DFT}$  Samples durch Fensterung mit  $w[n]$  aus einer unendlichen Folge aus (Rechteckfenster):

$$w_{Rect,N}[n] = \begin{cases} 1 & \text{für } 0 \leq n < N \\ 0 & \text{sonst.} \end{cases} \quad \text{mit } n = 0 \dots M \text{ und } M = N - 1$$

Alle anderen Fensterfunktionen sind ebenfalls abschnittsweise definiert oder multiplizieren eine periodische Funktion mit dem Rechteckfenster, hier läuft der Index  $n$  von  $0 \dots M$  mit der Abkürzung  $M := N_{DFT} - 1$ .

Fensterfunktionen werden so skaliert, dass im Zeitbereich das Maximum 1 ist. Die gängigsten Fensterfunktionen sind durch einfache mathematische Formeln definiert, damit man sie leicht berechnen und die Eigenschaften im Frequenzbereich vorhersagen kann. Die prinzipiellen Eigenschaften der Fenster ändern sich nicht durch eine Erhöhung der Punktzahl, das Fenster wird im Frequenzbereich nur stärker zusammengeschoben (bessere Frequenzauflösung).

Im folgenden wird nur der Teil der Fensterfunktion angegeben, der nicht Null ist.

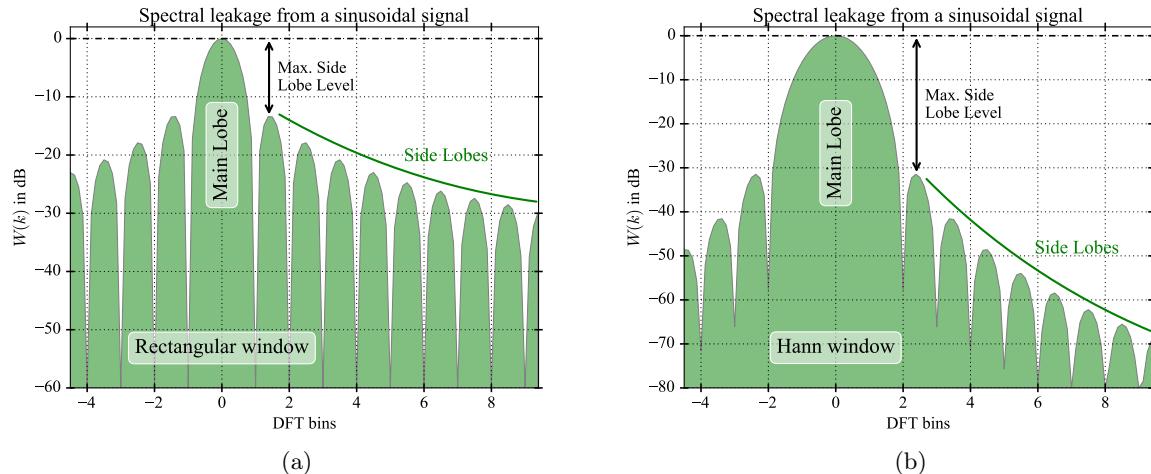


Abb. 4.2: Leckeffekt bei Rechteck- (a) und Hann-Fensterung (b)

$-3 \text{ dB BW}$	$SL_{max}$	$\Delta A_{max}$	$CGain$	$NENBW$
<b>Rechteckfenster</b>				
0,88	-13,3 dB	-3,8 dB	1	1

Das Rechteckfenster hat die geringste Bandbreite des Hauptmaximums und die schlechteste side lobe Dämpfung. Es sollte nur bei synchroner Fensterung (kein Leckeffekt) verwendet werden. Es hat die beste  $NENBW$  und wird daher auch verwendet bei schwachen Sinussignalen mit starken Rauschstörungen.

<b><math>-3 \text{ dB } BW</math></b>	<b><math>SL_{max}</math></b>	<b><math>\Delta A_{max}</math></b>	<b><math>CGain</math></b>	<b><math>NENBW</math></b>
<b>Bartlett- oder Dreiecksfenster</b>				
<b>1,27</b>	<b>-26,5 dB</b>	<b>-1,9 dB</b>	<b>0,5</b>	<b>4/3</b>

Das Dreiecksfenster entsteht durch Faltung von zwei Rechteckfenstern im Zeitbereich (Multiplikation im Frequenzbereich), daher ist der Main Lobe breiter und der max. Side Lobe bei -26 dB anstatt bei -13 dB. Es wird vor allem im Zusammenhang mit Periodogrammen verwendet.

<b>Hann- oder Hanning- oder Raised Cosine-Fenster</b>				
<b>1,44</b>	<b>-31 dB</b>	<b>-1,5 dB</b>	<b>0,5</b>	<b>1,5</b>
$w_{Hann}[n] = 0,5 - \cos \frac{2\pi n}{M_{DFT}}$				

Das Hann-Fenster ist die einfachste Variante der „klassischen“ Cosinus-Fensterfunktionen, definiert durch eine halbe Periode einer Cosinus-Funktion, die so „angehoben“ wurde dass die Endpunkte genau Null erreichen. Die Side Lobes fallen steil mit -18 dB/Okt. oder -60 dB/Dek. ab. Es ist ein Allroundfenster, das sehr gut geeignet ist bei starken Weitab-Störern.

<b>Hamming-Fenster</b>				
<b>1,34</b>	<b>-41 dB</b>	<b>-1,6 dB</b>	<b>0,54</b>	<b>1,36</b>
$w_{Hamming}[n] = 0,54 - 0,46 \cos \frac{2\pi n}{M_{DFT}}$				

Das Hamming - Fenster hat ebenfalls eine Cosinus-Fensterfunktion; es erreicht einen geringeren max. Sidelobe-Level als das Hann-Fenster hat, dieser fällt allerdings über den gesamten Frequenzbereich nur wenig ab.

<b>Blackman-Fenster</b>				
<b>1,66</b>	<b>-58 dB</b>	<b>-1,1 dB</b>		<b>1,75</b>
$w_{Blackman}[n, \alpha] = \frac{1 - \alpha}{2} - 0,5 \cos \frac{2\pi n}{M_{DFT}} + \frac{\alpha}{2} \cos \frac{4\pi n}{M_{DFT}}$				

Blackman - Fenster sind eine Familie von Cosinus-Fensterfunktionen mit drei Termen und Parameter  $\alpha$ , das „klassische“ Blackman-Fenster hat  $\alpha = 0,16$ . Sie erreichen eine besonders hohe Sperrdämpfung und einen Abfall der Side Lobes mit -60 dB/Dek. auf Kosten einer höheren Main Lobe Breite.

<b><math>-3 \text{ dB } BW</math></b>	<b><math>SL_{max}</math></b>	<b><math>\Delta A_{max}</math></b>	<b><math>CGain</math></b>	<b><math>NENBW</math></b>
<b>Kaiser-oder Kaiser-Bessel-Fenster</b>				
<b>2,1553</b>	<b>-119,8 dB</b>	<b>-0,64 dB</b>		<b>2,283</b>
$w_{Kaiser}[n] = \frac{I_0\left(\beta\sqrt{1 - ((n - \alpha)/\alpha^2)}\right)}{I_0(\beta)}$ mit $\alpha := \frac{M_{DFT}}{2}$				
<p>Das Kaiser-Fenster ist parametrisierbar mit dem Parameter <math>\beta</math>, hohe Werte für <math>\beta</math> ergeben hohe Side Lobe Unterdrückung und Main Lobe Breite und umgekehrt. Dabei ist <math>I_0(x)</math> die modifizierte Bessel-Funktion erster Art der Ordnung 0. Die obigen Angaben gelten für ein Kaiser-Fenster mit einem typischen Wert <math>\beta = 5</math>.</p>				
<b>Flattop-Fenster</b>				
<b>3,3451</b>	<b>-66,5 dB</b>	<b>-0,007 dB</b>	<b>0,2419</b>	<b>3,3868</b>
<p>Flattop-Fenster (es gibt viele, leicht unterschiedliche Varianten) basieren auf Cosinus-Funktionen mit 3 ... 5 Termen. Sie haben einen sehr geringen Amplitudenfehler und gute Side Lobe Dämpfung und werden daher u.a. in Spektrumanalysatoren eingesetzt. Der Preis dafür ist die große Breite des Hauptmaximums (schlechte Frequenzauflösung). Abhängig von der genauen Zielstellung erhält man unterschiedliche Optimierungen. Die obigen Angaben gelten für ein SFT4M Fenster [HRS02].</p>				

Tab. 4.1: Eigenschaften verschiedener Fensterfunktionen

Erläuterungen zu Tab. 4.1:

- **$-3 \text{ dB } BW$** : -3 dB Breite des Main Lobes in Bins ( $\hat{\equiv} f_S/N_{DFT}$ )
- **$SL_{max}$** : Höhe des größten Side Lobes
- **$\Delta A_{max}$** : Maximaler Amplitudenfehler
- **$CGain$** : Correlated Gain (s.u.); Verstärkung für (quasi)periodische Signale
- **$NENBW$** : Rauschbandbreite bezogen auf das Rechteckfenster

### Bewertungskriterien

Im Zusammenhang mit Spektralschätzung<sup>1</sup> beziehen sich die üblichen Bewertungskriterien für Fensterfunktionen auf deren Fouriertransformierte, also die Übertragungsfunktion im Frequenzbereich:

#### Breite des Hauptmaximums (Main Lobe, Hauptzipfel)

Eine Verbreiterung des Hauptmaximums führt zu einem schnelleren Abfall der Nebenmaxima (Nebenzipfel), erhöht die Dynamik der Fensterfunktion und verringert den Leck-Effekt. Allerdings wird dabei die Frequenzselektivität verringert. Fensterfunktionen mit breitem Hauptmaximum werden deshalb auch als nichtselektive, dynamische Fenster bezeichnet, und solche mit schmalem Hauptmaximum als selektive, nichtdynamische Fenster.

<sup>1</sup>Fensterfunktionen werden auch für den Entwurf zeitdiskreter Filter (Kap. 5) benötigt.

Die Breite des Hauptmaximums wird oft angegeben als Frequenz der ersten Nullstelle (z.B.  $F_S = 1/N_{DFT}$  oder  $\Omega_S = 2\pi/N_{DFT}$ ). Praxisgerechter ist die Angabe der 3-dB-Grenzfrequenz, bei der die Amplitude des Hauptmaximums um 3 dB abgefallen ist.

### **Relative Amplitude des Nebenmaximums (Side Lobe, Nebenzipfel)**

Starke Nebenmaxima einer Fensterfunktion erhöhen den Leck-Effekt bei der Frequenzanalyse und deuten auf eine geringe Dynamik der Fensterfunktion hin.

Als Bewertungskriterium wird das Verhältnis zwischen der Amplitude des Hauptmaximums und der Amplitude des höchsten Nebenmaximums verwendet.

### **Abfall der Nebenmaxima (Side Lobe fall off)**

Die Höhe der Nebenmaxima fällt bei den meisten Fenstern mit -20 dB/ Dekade ab. Je stärker der Abfall ist (z.B. -60 dB/ Dekade beim Hann-Fenster), desto weniger Einfluss haben weit entfernte Störsignale.

### **Maximaler Amplituden- oder Abtastfehler**

Der maximale Amplitudenfehler tritt auf bei Frequenzen, die genau zwischen zwei Frequenzbins liegen, er beschreibt den maximalen Messfehler für sinusförmige Signale.

### **Formfaktor (shape factor)**

Der Formfaktor ist definiert als das Verhältnis zwischen -3 dB Frequenz und -60 dB Frequenz. Der Formfaktor wird manchmal auch „window selectivity“ genannt.

### **Correlated Gain**

Die „Verstärkung“ (immer  $\leq 1$ ) für (pseudo-)periodische Signale (also Linienspektren im Frequenzbereich) entspricht dem Mittelwert des Fensters:

$$CGain = \frac{\sum_{n=0}^{N-1} w[n]}{N} \quad (4.1)$$

Für eine korrekte Amplitudendarstellung von Linienspektren muss das Amplitudenspektrum durch diesen Faktor dividiert werden.

## Normalized Equivalent Noise Bandwidth ( $NENBW$ )

Die normalisierte Rauschbandbreite des Fensters wird normalisiert bezogen auf die Rauschbandbreite eines Rechteckfensters. Mit Parcevals Theorem lässt sich das bequem in der Zeitebene berechnen:

$$NENBW = N \frac{\sum_{n=0}^{N-1} w[n]^2}{\left( \sum_{n=0}^{N-1} w[n] \right)^2} \text{ Bins} \quad (4.2)$$

Für eine korrekte Darstellung der Rauschleistung muss das Leistungsspektrum durch diesen Faktor dividiert werden.

## Spezielle Python-Funktionen

Funktion	Python	Matlab	Beschreibung
<b>Liste von Fenstern</b>	<code>w = get_window(win_name, N)</code>	<code>w = window(...)</code>	Erzeuge ein Fenster vom Typ <code>win</code> mit <code>N</code> Punkten, je nach Fenster müssen zusätzliche Parameter angegeben werden.
<b>Blackman-Fenster</b>	<code>w = blackman(N)</code>	<code>w = blackman(N)</code>	Erzeuge Blackman-Fenster mit <code>N</code> Punkten
<b>Fensterln</b>	<code>spectrum.Window</code>	<code>wttool</code>	Visualisierung von Fenstern im Zeit- und Frequenzbereich

Tab. 4.2: Spezielle Befehle zu Kap. 4

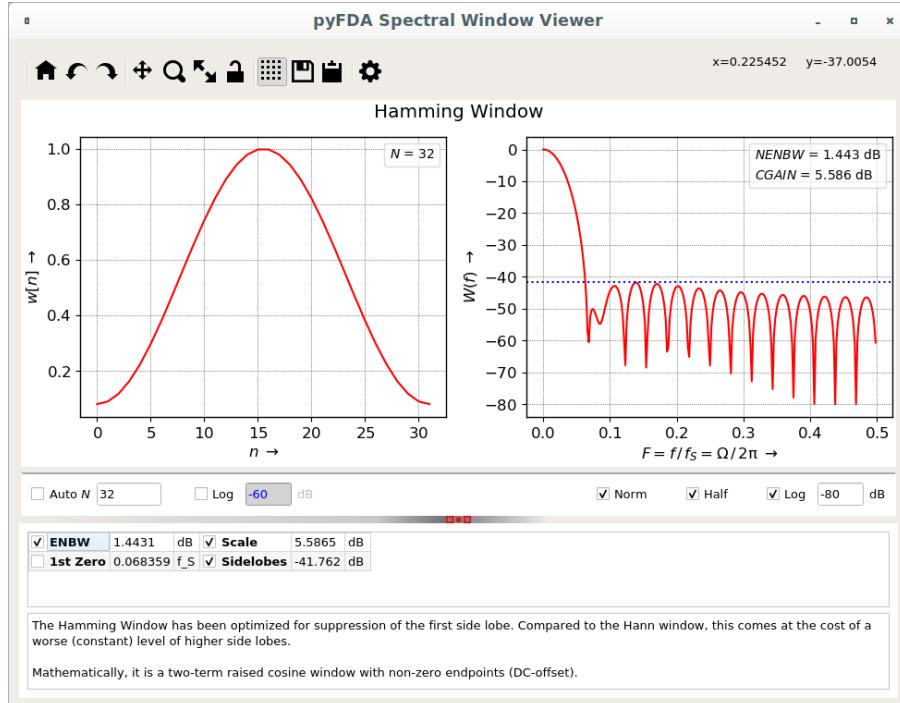


Abb. 4.3: Screenshot des pyFIDA Windowviewers

**Anmerkungen zu Tab. 4.2:**

**Allgemein:** Siehe auch Tab. 3.1 und die Hinweise dazu!

**Fenster:** U.a. sind die folgenden Fenster in `scipy.signal` und in Matlab (z.T. leicht abweichende Namen) implementiert: 'Barthann', 'Bartlett', 'Blackman', 'Blackmanharris', 'Bohman', 'Boxcar', 'Chebwin', 'Cosine', 'Flattop', 'General\_Gaussian', 'Gaussian', 'Hamming', 'Hann', 'Kaiser', 'Nuttall', 'Parzen', 'Slepian', 'Triang'.

Bei Matlab sind die Fenster in der Signal Processing Toolbox enthalten, bei Python im Submodul `scipy.signal`, also z.B. `from scipy.signal import kaiser`. Ein Teil der Fenster (z.B. Kaiser und Chebwin) sind parametrierbar.

**Schau, ein Fenster:** In Matlab können Sie sich mit `wvtool` und `wintool` Fenster im Zeit- und Frequenzbereich anzeigen lassen.

In Python können Sie hierfür in pyFDA ab Version 0.30 verwenden (Abb. 4.3). Drücken Sie hierzu den `WIN_VIEW` Button im `h[n]` Tab (Frequenzbereich). Alternativ gibt es das Modul `spectrum` (s.u.)

**Spectrum:** <http://thomas-cokelaer.info/software/spectrum/html/contents.html> ist ein leistungsfähiges Modul für Spektralschätzungen.

**Notebooks und Labs**


---

Filename	Beschreibung
<b>Notebooks</b>	
<code>WIN-Basics</code>	Fensterung mit kohärenter und nicht kohärenter Abtastung mit Rechteckfenster - von der DTFT zur DFT.
<code>WIN-Basic_2</code>	Einfluss der Fensterfunktion auf das Spektrum

---

Tab. 4.3: Jupyter Notebooks zu Kap. 4

## 4.1 Allgemeine Fragen → M4.1

Nehmen Sie ein mit  $f_S$  abgetastetes, unendlich ausgedehntes Signal  $x[n]$  an.

- Welche Eigenschaften hat das **Spektrum des ursprünglichen Signals**, welches Spektrum ist hier gemeint (kontinuierlich, diskret, (a)periodisch, DFT, DTFT, ...)?
- Welche Auswirkung hat die **Multiplikation des Signals mit einer Fensterfunktion der Länge  $N$**  auf das Spektrum?
- Wie sieht das Spektrum aus, wenn jetzt nur noch die  $N$  gefensterten Werte behandelt werden? Welches Spektrum ist jetzt gemeint?

## 4.2 Fourier-Analyse mit Rechteck-Fensterung → M4.2

Aus dem periodischen zeitkontinuierlichen Signal  $s(t) = A \cos(2\pi f_0 t)$  wird durch Multiplikation mit einem Rechteckfenster  $w(t)$  ein aperiodisches Signal  $x(t) = w(t)s(t)$  erzeugt. Dabei ist

$$w(t) = \begin{cases} 1 & \text{für } 0 \leq t \leq T_1 \\ 0 & \text{sonst.} \end{cases} \quad (4.3)$$

- Gesucht ist ein **analytischer Ausdruck für das Spektrum  $X(f)$**  (kontinuierliche Fouriertransformation). Es empfiehlt sich, zunächst die Spektren  $S(f)$  und  $W(f)$  formell anzuzeigen und dann den Faltungssatz anzuwenden.
- Skizzieren Sie den prinzipiellen Verlauf von  $|X(f)|$**  im Bereich  $-2 \text{ kHz} \leq f \leq 2 \text{ kHz}$ . Wählen Sie dazu die Zahlenwerte  $f_0 = 1 \text{ kHz}$ ,  $A = 1$ ,  $T_1 = 2T_0 = 2/f_0$  und überprüfen Sie das Ergebnis mit Python / MATLAB.
- Wie können Sie  $|X(f)|$  näherungsweise mit Hilfe einer **Radix-2 FFT** bestimmen?
- Können Sie das Spektrum von  $s(t)$  mit einer Radix-2 FFT ohne störende Effekte bekommen?

## 4.3 DFT mit Hann-Fenster → M4.3

ML fehlt noch!

Als einfaches Beispiel für Cosinus-Fenster wird in dieser Aufgabe das *Hann-Fenster* (Raised Cosine) betrachtet für  $M = N_{DFT} - 1$  Punkte:

$$w_{Hann,M}[n] = 0,5 \left( 1 - \cos \frac{2\pi n}{M} \right) = \sin^2 \frac{2\pi n}{M} \quad \text{mit } n = 0 \dots M$$

Im zweiten Schritt werden verschiedene periodische Signale  $s_i[n]$  mit dem Hann-Fenster gewichtet und mit einer DFT mit  $N_{DFT} = 32$  Punkten transformiert.

- Bestimmen Sie die **Spektralfunktion  $W_{Hann,M}(e^{j\Omega})$**  des Fensters.
- $s_1[n] = \cos 2\pi n/8$ : Wieviele Perioden passen in das Fenster, wie viele Frequenzpunkte erhalten Sie, die ungleich Null sind? Wie müssen Sie die Amplitude skalieren?
- $s_2[n] = \cos 2\pi n/8 + \cos 2\pi n/10$ : Welche Frequenzauflösung erhalten Sie?
- Plotten Sie Fenster und gefensterte Signale im Zeit- und Frequenzbereich.

#### 4.4 DFT mit Bartlett-Fenster → M4.4

**ML fehlt noch!**

In dieser Aufgabe sollen die Eigenschaften des Bartlett- oder Dreieck-Fensters der Länge  $N$  untersucht werden, gegeben durch

$$w[n] = 1 - \left| \frac{n - \frac{N-1}{2}}{\frac{N-1}{2}} \right| \quad (4.4)$$

- a) Bestimmen Sie die Verstärkung des Fensters für Schmalband- und Breitbandsignale (Correlated und Noise Gain). Nutzen Sie hierzu (4.1), (4.2) und die Summenformeln aus B.5.5.
- b) Schätzen Sie das Spektrum des Fensters ab. Hinweis: Sie erhalten ein Dreiecksfenster der Länge  $N$  durch Faltung von zwei Rechteckfenstern der halben Länge!

## 5 FIL: Einfache digitale Filter und FIR-Filterentwurf

### Lernziele

Dieses Kapitel beschäftigt sich mit einfachen allgemeinen Filtern und mit der Definition und dem Entwurf digitaler FIR Filter.

- Sie können Filter spezifizieren und kennen die Bedeutung der **Filterspezifikationen** in z.B. Filterentwurfstools.
- Sie nennen die Vor- und Nachteile von **FIR und IIR - Filtern** und erkennen sie im P/N Diagramm und der Systemfunktion  $H(z)$ .
- Sie kennen die grundlegenden **Filterentwurfsverfahren** und die resultierenden Filtereigenschaften.
- Sie erkennen **linearphasige Filter** im Zeit- und Frequenzbereich und kennen deren Vorteile. Sie konstruieren effiziente linearphasige Filter unter Ausnutzung der Symmetrieeigenschaften.
- Sie erkennen **Halbbandfilter** im Zeit- und Frequenzbereich und nennen deren spezielle Eigenschaften und Einsatzfelder.

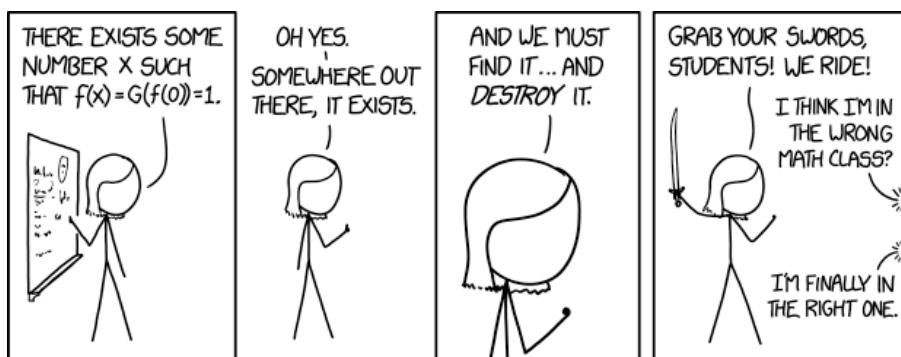


Abb. 5.1: Irgendwo da draußen ist das perfekte Filter [<https://xkcd.com/1856/>]

### Zusätzliche Literatur

J. O. Smith, *Introduction to digital filters*, <http://ccrma.stanford.edu/~jos/filters05/>, Sep. 2007, 480 S., 46 \$.

Sehr gut erklärte Grundlagen zu digitalen Filtern, auch zu Filtertopologien, mit Matlab-Beispielen. Online unter obiger URL kostenlos (im HTML-Format). Auch die anderen Bücher vom gleichen Autor ([https://ccrma.stanford.edu/~jos/filters05/Book\\_Series\\_Overview.html](https://ccrma.stanford.edu/~jos/filters05/Book_Series_Overview.html)) zu FFT und Audiotechnik sind sehr gut.

## Theorie

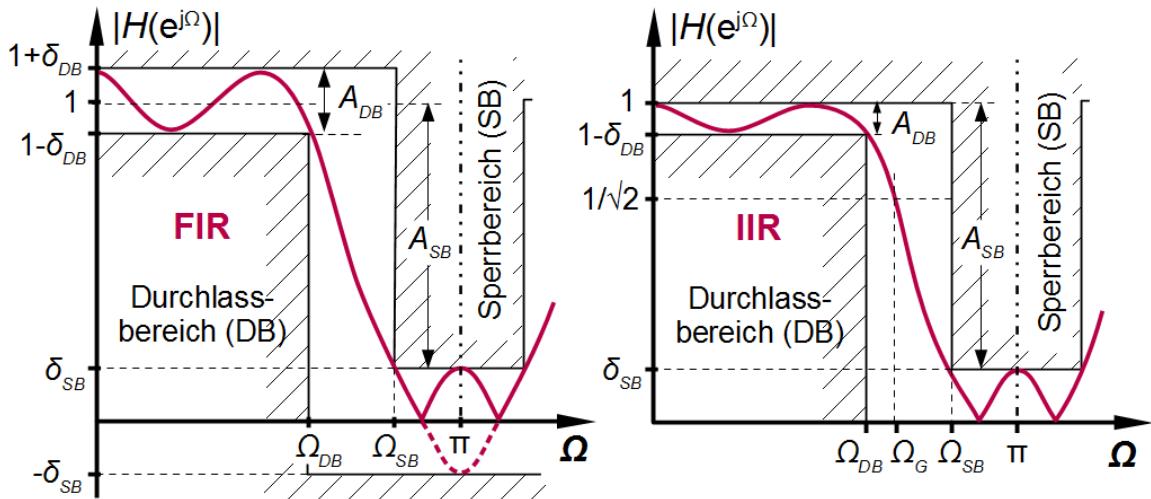


Abb. 5.2: Filterspezifikationen für FIR und IIR-Filter

## Spezifikationen

Abb. 5.2 zeigt die typische Bedeutung der Parameter für FIR- und IIR-Filter: IIR-Filter wurden früher ausschließlich von analogen passiven Filtern abgeleitet (s.u.), deren Durchlassband über die maximale Einfügedämpfung spezifiziert wird und eine maximale Verstärkung von 1 hat. Zu FIR-Filtern gibt es kein analoges Äquivalent, hier wird das DB spezifiziert über die maximale positive und negative Abweichung (Ripple)  $\delta_{DB}$  von der nominalen Verstärkung 1. Wenn die nominale Verstärkung  $A_0 \neq 1$  ist, muss der lineare Ripple mit  $A_0$  skaliert werden.

Die minimale Amplitudendämpfung im Sperrband in linearer und logarithmischer Schreibweise lässt sich für IIR und FIR-Filter ineinander umrechnen über:

$$A_{SB} = -20 \log_{10} \delta_{SB} \Leftrightarrow \delta_{SB} = 10^{-A_{SB}/20} \quad (5.1)$$

**Achtung:**  $A_{SB}$  bezeichnet die *Dämpfung*,  $\delta_{SB} = 0,001 \hat{=} A_{SB} = +60$  dB.

Im Durchlassband muss zwischen IIR und FIR-Filtern unterschieden werden:

$$\text{IIR: } A_{DB} = -20 \log_{10}(1 - \delta_{DB}) \Leftrightarrow \delta_{DB} = 1 - 10^{-A_{DB}/20} \quad (5.2)$$

$$\begin{aligned} \text{FIR: } A_{DB} &= 20 \log_{10}(1 + \delta_{DB}) - 20 \log_{10}(1 - \delta_{DB}) \\ &= 20 \log_{10} \frac{1 + \delta_{DB}}{1 - \delta_{DB}} \Leftrightarrow \delta_{DB} = \frac{10^{A_{DB}/20} - 1}{10^{A_{DB}/20} + 1} \\ &\approx 20 \log_{10}(1 + \delta_{DB})^2 = 40 \log_{10}(1 + \delta_{DB}) \Leftrightarrow \delta_{DB} \approx 1 - 10^{-A_{DB}/40} \end{aligned} \quad (5.3)$$

$A_{DB}$  bezeichnet den logarithmierten Ripple und ist ebenfalls positiv.

**Achtung:** Betrachtet man Leistungen (quadrierte Signale) anstelle von Amplituden, muss wie üblich mit  $10 \log_{10} P$  gerechnet werden!

Generell hängt es vom Entwurfverfahren ab, mit welchen Parametern der Frequenzgang spezifiziert wird. So wird z.B. beim Windowed FIR-Entwurfsverfahren die **-6** dB Eckfrequenz als Parameter angegeben, bei den klassischen IIR-Entwürfen dagegen die **-3** dB Eckfrequenz. Bei manchen Entwurfverfahren werden Filter zu gegebenen Randbedingungen (Eckfrequenzen von DB und SB und Ripple in den Bändern) gesucht.

In diesem Kurs werden die folgenden Symbole für Filterspezifikationen verwendet:

$f_s$	Abtastfrequenz
$f_{DB}$ ( $F_{DB}$ )	(Normierte) Eckfrequenz des Durchlassbands (DB)
$f_{SB}$ ( $F_{SB}$ )	(Normierte) Eckfrequenz des Sperrbands (SB)
$\delta_{DB}$	Ripple (= Welligkeit) im DB
$\delta_{SB}$	Ripple (= Welligkeit) im SB
$A_{DB}$	Log. Ripple im DB
$A_{SB}$	Log. Dämpfung im SB

Es wird hier immer von einer Grundverstärkung des Filters im Durchlassband von 1 ( $\hat{=} 0$  dB) ausgegangen.

**Für einen möglichst entspannten Filterentwurf müssen die Eckfrequenzen von Durchlass- und Stoppband möglichst weit auseinanderliegen (breiter Übergangsbereich)!**

### Halbbandfilter

Halbbandfilter sind Filter, deren Frequenzgang punktsymmetrisch zu  $(\pi/2, H(\Omega = \pi/2))$  verläuft. Es muss also allgemein gelten

$$H(\Omega) + H(\pi - \Omega) = 2H(\pi/2)$$

bzw.  $H(\pi/2 - \Omega) + H(\pi/2 + \Omega) = 2H(\pi/2)$

Damit müssen auch die Eckfrequenzen und der Ripple von Pass- und Stopband symmetrisch zu einander sein. Meistens (Filter mit normalisiertem Amplitudengang) ist  $H(\pi/2) = 0,5$ , aber das ist keine Voraussetzung für ein Halbbandfilter.

**Achtung:**  $H(\Omega)$  ist der komplexe Frequenzgang, nicht der Betragsgang! Da Halbband FIR-Filter aber linearphasig sein müssen, kann man die Rechnung deutlich vereinfachen, indem man die *akausale* Systemfunktion  $H_{ak}(z)$  bestimmt: Man eliminiert die lineare Phase aufgrund der mittleren Verzögerung  $z^{-N/2}$ :

$$H_{ak}(z) = H(z)z^{N/2}$$

$$\Rightarrow H_{ak}(e^{j\Omega}) = \underline{H}(e^{j\Omega})e^{j\Omega N/2}$$

Der so erhaltene Amplitudengang  $H_{ak}(e^{j\Omega})$  ist reellwertig, kann aber das Vorzeichen wechseln.

Ein FIR-Filter mit einem solcherart eingeschränkten Amplitudengang hat auch eine spezielle Impulsantwort / Koeffizienten: Halbband-FIR-Filter haben eine symmetrische Impulsantwort (sind also automatisch linearphasig), außer dem mittleren ist jeder zweite Koeffizient Null. Damit das möglich ist, muss die Anzahl  $L$  der Koeffizienten ungerade sein mit  $L = 4k + 3$  mit  $k = 1, 2, \dots$ . Die Ordnung  $N = L - 1$  muss dementsprechend gerade sein.

## Abschätzen der Ordnung

Zur schnellen Abschätzung der Ordnung verschiedener FIR- und IIR-Filterentwürfe können folgende Formeln verwendet werden:

$$N_{FIR} \approx \left\lceil \frac{k_{FIR} f_S}{f_{SB} - f_{DB}} \right\rceil = \left\lceil \frac{k_{FIR}}{F_{SB} - F_{DB}} \right\rceil \text{ mit } k_{FIR} \approx 3 \quad (5.4)$$

$$N_{FIR} \approx \left\lceil \frac{-20 \log_{10} \sqrt{\delta_{SB} \delta_{DB}} - 13}{14,6(F_{SB} - F_{DB})} \right\rceil \quad (5.5)$$

$$N_{IIR} \geq \left\lceil \frac{\sqrt{f_{SB} f_{DB}}}{f_{SB} - f_{DB}} + \frac{a_{SB}}{20 \text{ dB}} \right\rceil \quad (5.6)$$

Die grobe Abschätzung (5.4) ist entnommen aus [Smi99, S. 289] für Windowed Sinc Filter. Der Wert für  $k_{FIR}$  hängt leicht vom gewählten Fenster, der gewünschten Sperrdämpfung und dem DB-Ripple ab. Die Näherung passt auch grob für Equiripple-Filter bei einer Sperrdämpfung von  $\delta_{SB} = 10^{-4} \hat{=} -80 \text{ dB}$  und einem Ripple von  $\delta_{DB} = 0,01 \hat{=} 20 \log_{10}(1 + \delta_{DB}) = 0,086 \text{ dB}$ .

(5.5), entwickelt von Kaiser, ist eine Abschätzung für FIR-Filter, die mit Kaiser - Fenster entworfen wurden. Hier gehen auch die gewünschte Sperrdämpfung  $\delta_{SB}$  und der zulässige Ripple des Durchlassbands  $\delta_{DB}$  ein [J.F. Kaiser, *Nonrecursive Digital Filter Design Using  $I_0 - \sinh$  Window function*, Proc. IEEE Int. Symp. Circuits and Systems, 20-23, April 1974].

Bei FIR-Filtern kann der Frequenzgang nur über die Nullen beeinflusst werden, deren Anzahl gleich der Ordnung  $N_{FIR}$  ist. Die Dämpfung im Sperrbereich wird durch Verteilung von Nullstellen auf dem EK realisiert. Ist die Abtastfrequenz groß im Vergleich zur Eckfrequenz des Sperrbands, werden für das dann breite Sperrband viele Nullstellen benötigt. Die Ordnung ist daher in etwa proportional zur Abtastfrequenz (bei ansonsten unveränderten Parametern).

Der Abschätzung (5.6) liegt zugrunde, dass bei rekursiven Filtern vor allem die Platzierung der Pole (im Durchlassband) den Frequenzgang beeinflusst. Die Ordnung  $N_{IIR}$  ist gleich der Anzahl der Pole; im Sperrbereich, weit entfernt von den Polfrequenzen, fällt der Amplitudengang mit 20 dB/Dek. je Pol ab. Bei IIR - Filtern gibt es daher nur eine schwache Abhängigkeit von der Abtastfrequenz. Für elliptische Filter gilt näherungsweise das Gleichheitszeichen, bei anderen Filtertypen (Chebychev, Butterworth, ...) ist  $N_{IIR}$  um einen Faktor 1,5 ... 6 größer und hängt stärker von der relativen Bandbreite ab.

Vor allem bei IIR - Filtern, in extremen Fällen auch bei FIR - Filtern, führt eine sehr geringe relative Bandbreite außerdem zu numerischen Problemen. Die Quantisierung der Koeffizienten und der Arithmetik kann daher größere Abweichungen des Frequenzgangs und ein reduziertes SNR zur Folge haben, IIR - Filter können sogar instabil werden. Wenn geringe relative Bandbreiten benötigt werden, ist daher fast immer ein Multiratenfilter (→ Kap. 8) die beste Lösung.

## Entwurfsverfahren

Das Ergebnis von Filterentwürfen mit Python / Matlab sind generell Vektoren mit Zähler- und Nennerkoeffizienten (nur bei IIR-Filtern) von  $H(z)$  in Polynomschreibweise mit positiven Exponenten. Funktionen zum Filterentwurf sind in Python im Modul `scipy.signal` und in Matlab in der Signal Processing Toolbox enthalten.

Im Notebook **FIL-Intro** können Sie die verschiedenen Entwurfsverfahren ausprobieren und sich die Performance im Frequenzbereich anschauen. Noch bequemer geht das natürlich mit dem Filterentwurfstool pyFDA.

## Least-Square / Windowed Design / Fourier-Approximation

Dieses Entwurfsverfahren bestimmt die Koeffizienten eines FIR-Filters, indem die unendlich ausgedehnte Impulsantwort eines idealen Filters (rechteckiger Frequenzgang -> si-Zeitfunktion!) approximiert wird, d.h. durch Fensterung auf eine endliche Anzahl Koeffizienten begrenzt wird. Im einfachsten Fall bricht man die Impulsantwort einfach ab (= Rechteckfensterung, „boxcar“). Auf diese Art erhält man für eine gegebene Ordnung den minimalen Fehlerleistung (Least-Square) - das heißt aber nicht minimale Abweichung vom idealen Frequenzgang: Aufgrund der Rechteckfensterung treten Überschwinger an der Bandkante auf! Bei diesem Entwurfsverfahren kann man als einzigen Parameter die -6dB-Grenzfrequenz vorgeben.

**Matlab:** `firls()`, **Python:** `scipy.signal.firwin()` mit 'boxcar' - Fenster.

In Python findet man alle Filterentwurfsverfahren im Untermodul `scipy.signal`, die Fensterfunktionen unter `scipy.signal.windows` (im Folgenden weggelassen).

Durch eine „weichere“ Fensterfunktion im Zeitbereich erhält man einen Frequenzgang mit geringerem Überschwingen auf Kosten eines breiteren Übergangsbereichs. Vorteilhaft im Gegensatz zum Equirippleverfahren (s.u.) ist, dass die Filterdämpfung bei wachsendem Abstand von der Eckfrequenz je nach Fenster rasch abnimmt, nachteilig ist dass i.a. eine höhere Filterordnung benötigt wird.

**Matlab:** `fir1()`, **Python:** `firwin()`, Default-Fenster ist jeweils 'hamming'.

**Fenster in Python:** 'bartlett' (= Dreieck), 'blackman', 'blackmanharris', 'boxcar' (= Rechteck), 'flat-top', 'gaussian', 'hamming', 'hann' (oder 'hanning'), 'kaiser', 'nuttall', ...

**Fenster in Matlab:** 'bartlett' = 'triang', 'blackman', 'blackmanharris', 'boxcar' = 'rectwin', 'flattopwin', 'gausswin', 'hamming', 'hann', 'kaiser', 'nuttall', ...

Alle Fensterfunktionen haben als Parameter die Länge des Fenster und die optionale Angabe ob das Fenster symmetrisch (Default) oder periodisch sein soll.

## Frequency Sampling

Wenn man von Filterspezifikationen in der Frequenzebene ausgeht, ist Frequency Sampling vermutlich das am einfachsten verständliche und direkteste Filterentwurfverfahren: Der Wunschkoeffizienten wird in gleichmäßigen Abständen (äquidistant) abgetastet, durch inverse Fouriertransformation erhält man sofort die Impulsantwort, die den Filterkoeffizienten des FIR-Filters entsprechen. Auf diese Art können nahezu beliebige Verläufe des Frequenzgangs erzielt werden (auch Multiband, Differenzierer, Hilbert-Filter, ...). Allerdings wird der gewünschte Frequenzgang exakt nur an den Abtastpunkten erreicht - dazwischen können z.T. wilde Überschwinger des Frequenzverlaufs auftreten. Daher muss man unbedingt das Ergebnis des Entwurfs überprüfen, indem man eine „DTFT“ (d.h. eine DFT mit ausreichendem Zero Padding, die zwischen den Frequency Sampling Punkten interpoliert) der Filterkoeffizienten berechnet und mit dem Wunschkoeffizienten vergleicht.

**Python:** `firwin2()`, **Matlab:** `fir2()`:

## Equiripple / Remez / Parks-McClellan / Minimax

Dieses Entwurfsverfahren minimiert den maximalen Fehler (-> **Minimax**) im Durchlass- und Sperrband und schafft so einen gleichmäßigen Ripple (-> **Equiripple**), der die vorgegebenen Spezifikationen voll ausschöpft. Benutzt wird hierfür der Algorithmus von **Parks-McClellan** und das Austauschverfahren von **Remez** - das Verfahren ist in der Literatur daher unter allen vier Namen zu finden. Hier werden im einfachsten Fall die Eckfrequenzen von DB und SB mit den zulässigen Abweichungen

(Ripple) vorgegeben. Im Sperrband entspricht der Ripple der minimalen Dämpfung. Durch Angabe von Frequenz / Amplitudenpaaren können aber auch hier relativ beliebige Frequenzgänge vorgegeben werden. Optional können hier die DB und SB Spezifikationen unterschiedlich gewichtet werden.

**Matlab:** `firpm()`, **Python:** `remez()`.

Achtung: Bei Python sind die relativen Frequenzen hier auf  $f_S$ , nicht auf  $f_S/2$  bezogen (daher wird hier die Abtastfrequenz mit `Hz = 2` spezifiziert). Wie bei den zuvor besprochenen Verfahren, kann die Filterordnung manuell angegeben werden. Es gibt aber auch Algorithmen, mit denen man (mehr oder weniger korrekt) die Filterordnung und die Gewichtungen für DB und SB abschätzen kann. Mit den so gewonnenen Angaben startet man das eigentliche Filterdesign. Bei Python ist die Funktion `remezord()` nicht in Scipy enthalten, Sie finden Sie in der dsp - Library auf Moodle.

## Spezielle Python-Funktionen

Funktion	Python	Matlab	Beschreibung
Filterung	<code>h,t = sig.lfilter(b,a)</code>	<code>[h,t] = filter(b,a)</code>	Lineare Filterung mit dem durch <code>b</code> , <code>a</code> definierten System.
<b>FIR-Filter</b>			
Minimax-Filter	<code>sig.remez()</code>	<code>firpm()</code>	Parks-McClellan - FIR Filter

Tab. 5.1: Spezielle Befehle zu Kap. 5

### Anmerkungen zu Tab. 5.1:

**Allgemein** Siehe auch die Anmerkungen zu Tab. 2.1

## Notebooks und Labs

Filename	Beschreibung
<b>Notebooks</b>	
FIL_Intro	Implementierung verschiedener Filterentwurfsverfahren in Python (siehe oben)
FIL_BPSK	Bit Error Rate (BER) bei Störung eines BPSK-kodierten Signals durch additives weißes Rauschen. Es wird gezeigt, welchen Einfluss Grenzfrequenz und Phasenverzerrungen von verschiedenen Tiefpassen auf die BER haben.
FIL-Linphase_Filter	Einfluss von einfachen und doppelten Nullstellen auf den Amplitudengang (zu Aufgabe 5.2)
FIL-Linphase-Filter_annotation	wie voriges Programm, aber mit zusätzlichen Textausgaben im Plot

---

Filename	Beschreibung
FIL-Halffband_design	Entwurf von Halbbandfiltern mit unterschiedlichen Methoden, Vergleich im Zeit- und Frequenzbereich (Aufgabe 5.4)

---

Tab. 5.2: Jupyter Notebooks zu Kap. 5

## 5.0 Filterentwurf für Sensorsignal

Brummstörungen bei 50 Hz auf einem Sensorsignal sollen mit einem digitalen Filter um mindestens 60 dB unterdrückt werden. Das Sensorsignal wird mit einem ADC mit  $f_S = 400$  Hz abgetastet und hat eine Nutzbandbreite von 0 ... 40 Hz. Das Filter soll einen Amplitudenfehler von maximal  $\pm 1$  dB verursachen.

Entwerfen Sie verschiedene FIR Filter mit dem grafischen Entwurfstool pyFDA oder Mathworks fdatool. Alternativ können Sie auf dem Notebook **FIL\_intro** aufbauen und damit versuchen, die gestellten Anforderungen zu erfüllen. Die abgedruckten Listings zeigen nur Ausschnitte des Files, nehmen Sie daher die Files in elektronischer Form und passen Sie sie an die Aufgabenstellung an. Kommentieren Sie für die Simulationen jeweils alle Entwurfsverfahren bis auf eins aus.

Vergleichen Sie die Systeme bezüglich Resourcenverbrauch (Ordnung) und schauen Sie sich die unterschiedlichen Pol/Nullstellenpläne und Betrags- und Phasenverlauf an! Welche Filter haben eine gute Unterdrückung von Störungen bei höheren Frequenzen (z.B. 150 Hz)?

### 5.1 Filterspezifikationen → M5.1

- Ein Signal soll mit einem FIR- und einem IIR-Filter so gefiltert werden, dass die Ausgangsamplitude im Durchlassbereich um maximal  $\pm 10\%$  variiert. Welche **Spezifikationen**  $\delta_{DB}, A_{DB}$  muss man dazu in einem Filterentwurfstool angeben?
- Ein IIR- und ein FIR-Filter sind spezifiziert mit  $A_{DB} = 2$  dB und  $A_{SB} = 20$  dB. Die Grenzfrequenzen betragen jeweils  $F_{DB} = 0,1$  und  $F_{SB} = 0,13$ . Wie groß ist jeweils der **Ripple im Durchlassband und Sperrband**  $\delta_{DB}, \delta_{SB}$ ? Skizzieren Sie den prinzipiellen Verlauf beider Filter für ein Equiripple- und ein Chebychev (Typ 2, Ripple im Sperrband)-Design im linearen und logarithmischen Maßstab!
- Die Filter sollen jetzt einem (als ideal angenommenen) Audioverstärker mit einer maximalen Sinusleistung von  $P_{nenn} = 80$  W an  $8 \Omega$  vorgeschaltet werden. In welchem Bereich können die **Ausgangsleistung und Ausgangsamplitude im Durchlassband** variieren? Welche maximale Ausgangsleistung und -amplitude kann man im **Sperrband** erwarten?
- Einem Sensorsignal im Frequenzbereich  $f_{sig} = 0 \dots 20$  Hz mit Amplitude  $\hat{u}_{sig} = 1 \dots 50$  mV ist ein Brummen mit  $f_{br} = 50$  Hz und  $\hat{u}_{br} = 100$  mV überlagert. Die Abtastfrequenz beträgt  $f_S = 1$  kHz. Zur Dämpfung der Störkomponente soll jetzt ein Tiefpassfilter entworfen werden.

Der Übergangsbereich zwischen Durchlass- und Sperrband des Filters soll möglichst breit sein. Geben Sie die (**normierten**) **Eckfrequenzen an für Durchlass- und Sperrband**,  $f_{DB}, f_{SB}, F_{DB}, F_{SB}, \Omega_{DB}$  und  $\Omega_{SB}$ .

Bestimmen Sie die logarithmischen **Spezifikationen für das Filter**, wenn die Brummamplitude am Ausgang des Filters höchstens ein Tausendstel der Signalamplitude betragen darf und die Signalamplitude im Durchlassband maximal um 1% verfälscht werden darf.

Nutzen Sie zur Lösung der Aufgabe die Abbildungen und Formeln auf S. 44f.

### 5.2 Amplitudengang linearphasiger Filter → M5.2

Gegeben ist die Impulsantwort  $h[n] = \{-1/8, 0, 5/8, 1, 5/8, 0, -1/8\}$  des FIR-Filters F. Die nachstehenden Bilder Abb. 5.3a und Abb. 5.3b zeigen den Amplitudengang  $|H(e^{j2\pi F})|$  im linearen und logarithmischem Maßstab.

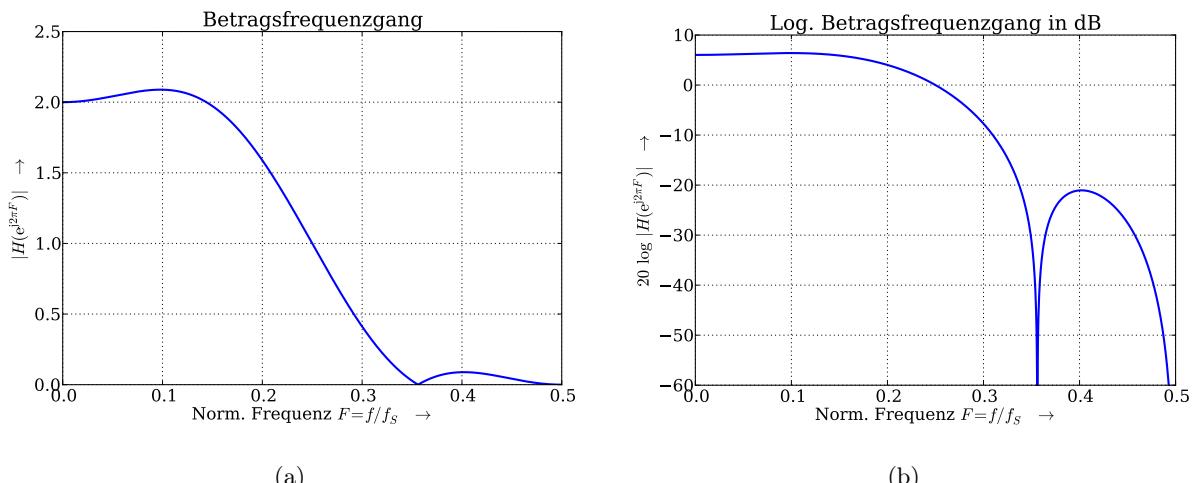


Abb. 5.3: Amplitudengang des Filters  $F$  im (a) linearen und (b) logarithmischen Maßstab

- a) Ist das Filter linearphasig? Ist es ein Halbbandfilter? Begründen Sie Ihre Aussage!
  - b) Beweisen Sie, dass  $H(z)$  bei  $f_S/2$  eine doppelte Nullstelle hat.
  - c) Schätzen Sie die Lage von zwei weiteren Nullstellen auf dem Einheitskreis der  $z$ -Ebene ab.
  - d) Wie viele Nullstellen sind noch übrig? Wo könnten sie liegen und was bewirken sie? Plotten Sie mit Hilfe eines Matlab / Python-Programms die Betragsfunktion für ein System *ohne* diese Nullstellen.
  - e) Skizzieren Sie das ganze Pol/Nullstellendiagramm in der  $z$ -Ebene. Schreiben Sie ein Python / Matlab - Programm, mit dem Sie das P/N - Diagramm darstellen können

## 5.3 Maximal-, minimal- und linearphasige Filter → M5.3

In dieser Aufgabe wird untersucht, wie sich die Platzierung einer Nullstelle auf Minimal- / Maximal- und Linearphasigkeit eines Filters erster bzw. zweiter Ordnung auswirkt. Es wird gezeigt, wie ein komplexwertiges Filter implementiert werden kann.

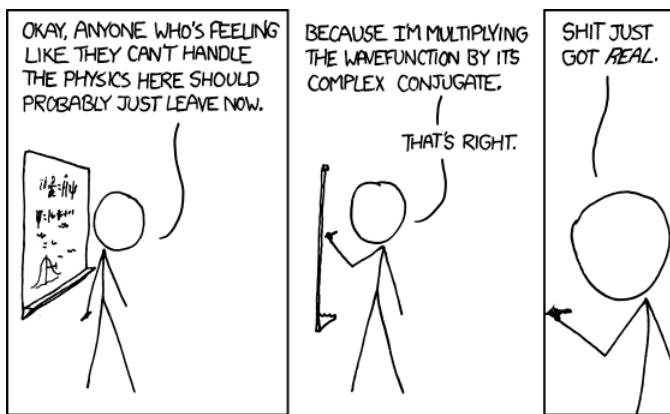


Abb. 5.4: Konjugiert und komplex [<http://xkcd.com/849/>]

- a) Bestimmen Sie allgemein die Systemfunktion  $H_1(z)$ , Impulsantwort  $h_1[n]$ , komplexen Frequenzgang  $H_1(e^{j\Omega})$ , Betragsgang  $|H_1(e^{j\Omega})|$  und Phasengang  $\angle H_1(e^{j\Omega})$  eines Filters  $F_1$  mit der Nullstelle  $z_{0,1} = r_1 e^{j\Omega_1}$  und einem Pol bei  $z = 0$ .

Ist das Filter  $F_1$  kausal, reellwertig, minimalphasig, linearphasig?

- b) Skizzieren Sie  $|H_1(e^{j\Omega})|$  und  $\angle H_1(e^{j\Omega})$  für die Zahlenwerte  $\Omega_1 = \pi/3$  und  $r_1 = 2$  im Bereich  $0 \leq \Omega \leq 2\pi$  (d.h.  $0 \leq f \leq f_S$ ).

Berechnen Sie hierzu zunächst Hilfswerte für  $|H_1(e^{j\Omega})|$  und  $\angle H_1(e^{j\Omega})$  bei den normierten Frequenzen  $\Omega = 0$ ,  $\Omega = \pi/3$  und  $\Omega = \pi$ .

- c) Leiten Sie aus  $F_1$  durch Spiegeln der Nullstelle  $z_{0,1}$  am EK das Filter  $F_2$  ab. Nehmen Sie weiterhin an, dass der Radius der ursprünglichen Nullstelle  $r_1 > 1$ . Ist das Filter  $F_2$  minimalphasig?

Skalieren Sie  $H_2(z)$  so, dass  $F_1$  und  $F_2$  bei  $\Omega = \Omega_1$  den gleichen Betrag der Verstärkung haben,  $|H_1(e^{j\Omega_1})| = |H_2(e^{j\Omega_1})|$ .

Können Sie zeigen, dass  $|H_1(e^{j\Omega})| = |H_2(e^{j\Omega})|$  bei allen Frequenzen gilt?

- d) Skizzieren Sie  $|H_2(e^{j\Omega})|$  und  $\angle H_2(e^{j\Omega})$  für die Zahlenwerte aus Unterpunkt b), berechnen Sie auch hier zunächst Hilfswerte.

- e) Konstruieren Sie das Filter F mit  $H(z) = H_1(z)H_2(z)$ . Untersuchen Sie, ob F linearphasig ist. Berechnen Sie die den Phasengang  $\angle H(e^{j\Omega})$  und die Gruppenlaufzeit  $\tau_{g,H}(e^{j\Omega})$  dieses Filters.

- f) Zeichnen Sie das Blockschaltbild für eine mögliche Realisierung des Filters F. Das Eingangssignal  $x[n]$  sei reell. Wie würden Sie dieses Filter in Hardware implementieren?

## 5.4 FIR Halbbandfilter → M5.4

- a) Ausgehend von folgenden Spezifikationen soll ein Halbbandfilter entworfen werden:

**Abtastfrequenz:**  $f_S = 24$  kHz

**Eckfrequenz des DB:**  $f_{DB} = 1$  kHz

**Ripple des DB:**  $A_{DB} = 0,1$  dB

**Eckfrequenz des SB:**  $f_{SB} = 10$  kHz

**Dämpfung des SB:**  $A_{SB} = 60$  dB

Passen Sie die obigen Spezifikationen so an, dass Sie ein Halbbandfilter erhalten. Die ursprünglichen Spezifikationen dürfen dabei nicht verletzt, aber verschärft werden. Entwerfen Sie dann das Filter mit Python / Matlab!

- b) Bestimmen Sie den Amplitudengang aus der Impulsantwort des folgenden Filters und zeigen Sie, dass es einen zu  $F = 0,25$  symmetrischen Betragsgang hat und somit ein Halbbandfilter ist:

$$h[n] = \{0,1; 0; 0,2; 0,3; 0,2; 0; 0,1\}$$

Welchen Wert hat der Betragsgang bei  $F = 0$ ,  $F = 0,25$  und  $F = 0,5$ ? Wie lässt sich die Dämpfung der Frequenzkomponente bei  $f_S/2$  verbessern?

Ermitteln Sie hierzu zunächst  $H(z)$  und daraus  $H(e^{j\Omega})$ . Formen Sie  $H(e^{j\Omega})$  dann so um, dass man die gefragte Symmetrie erkennen kann. Die Rechnung ist einfacher, wenn Sie die akausale Systemfunktion  $H_{ak}(z)$  ermitteln!

- c) Wie müssen Sie die Koeffizienten skalieren, damit gilt  $|H(F = 0,25)| = 0,5$ ?
- d) Zeigen Sie, dass **jeder zweite Koeffizient eines Halbbandfilters Null ist** (außer dem mittleren). Gehen Sie hierzu von der Übertragungsfunktion eines idealen, acausalen TP-Filters aus, mit  $\Omega_g = \pi/2$  mit Hilfe der inversen DTFT in den Zeitbereich. Nutzen Sie dazu folgende Formel:

$$X(\Omega) = \begin{cases} 1 & \text{für } 0 \leq |\Omega| < \Omega_g \\ 0 & \text{sonst} \end{cases}$$

•→  $x[n] = \frac{1}{2\pi} \int_{-\Omega_g}^{\Omega_g} e^{j\Omega n} = \frac{\Omega_g}{\pi} \frac{\sin(\Omega_g n)}{\Omega_g n} = \frac{\Omega_g}{\pi} \sin \Omega_g n \quad \text{für } n = 0, 1, 2, \dots$

- e) Welche Art von Filter erhalten Sie, wenn der **mittlere Koeffizient ebenfalls Null** ist?
- f) Konstruieren Sie aus dem (verbesserten) Filter aus b) ein **Halbband-Hochpassfilter**. Hierzu können Sie die TP -> HP - Transformation verwenden oder überlegen, wie Sie die Koeffizienten variieren müssen.
- g) Es gibt auch **IIR-Halbbandfilter**<sup>1</sup>. Überlegen Sie, welche der Ihnen bekannten IIR-Entwurfsverfahren geeignet ist und wie das P/N-Diagramm eines solchen Filters aussehen könnte!

## 5.5 Filterentwurf mit Python → M5.5

Entwerfen Sie mit Python / pyFDA (oder Mathworks fdatool) ein FIR-Filter nach der Equiripple-Methode und eins nach der Kaiser-Fenster-Methode mit folgenden Spezifikationen:

Das Durchlassband verläuft im Bereich  $F = 0,05 \dots 0,5$  und das Stopband zwischen  $F = 0 \dots 0,025$  (jeweils normalisiert auf  $f_S$ ). Die Dämpfung im Sperrband soll mindestens 50 dB betragen, der Ripple im Nutzband höchstens 0,1 dB.

## 5.6 Filtertransformationen → M5.6

Ein FIR-Filter F1 mit der Impulsantwort  $h_1[n] = \{1; 7/4; 1/2; -1/4\}$  hat den Betragsfrequenzgang in Abb. 5.5.

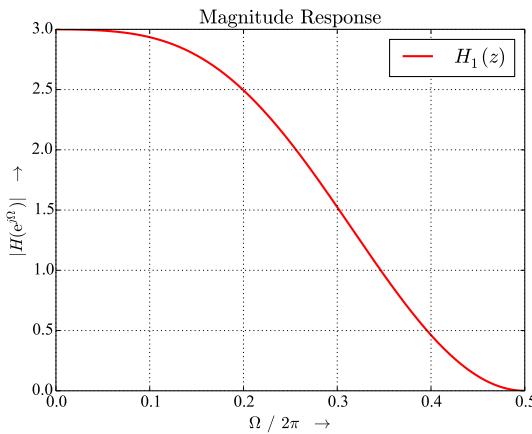


Abb. 5.5: Betragsfrequenzgang zu Aufgabe 5.6

<sup>1</sup>Auch wenn die Symmetriebedingung hier etwas anders definiert ist und natürlich keine Linearphasigkeit vorausgesetzt wird.

- a) Berechnen Sie  $H_1(f = 0)$  und  $H_1(f = f_S/2)$  und beweisen Sie, dass  $H_1(f)$  zwei Nullstellen bei  $f = f_S/2$  hat. Gibt es weitere Nullstellen? Wieviele, welche?
- b) Aus F1 wird ein neues Filter F2 abgeleitet, indem jedes Verzögerungsglied verdoppelt wird. Geben Sie  $h_2[n]$  und  $H_2(z)$  an. Wie wirkt sich die Transformation  $z^{-1} \rightarrow z^{-2}$  auf den Betragsfrequenzgang aus? Skizzieren Sie den Betragsfrequenzgang von F2.
- c) Erzeugen Sie mittels TP-HP-Transformation  $z \rightarrow -z$  aus Filter F1 ein Filter F3. Geben Sie  $h_3[n]$  an und skizzieren Sie den zugehörigen Betragsfrequenzgang.

## 5.7 Filterimplementierung auf FPGA → M5.7

Ein FIR-Filter 20. Ordnung soll auf einem FPGA mit MAC-Cores implementiert werden, die jeweils  $2 \cdot 10^8$  Multiplikationen je Sekunde durchführen können.

- a) Welcher maximale Datendurchsatz in Samples/s lässt sich für ein beliebiges FIR-Filter 20. Ordnung erzielen, wenn nur ein MAC-Core für das Filter eingesetzt werden darf?
- b) Welcher Durchsatz lässt sich bei einem linearphasigen FIR-Filter unter Ausnutzung der Symmetrie erzielen? Welcher Durchsatz ist bei einem Halbbandfilter möglich?
- c) Auf welchen Wert lässt sich der Durchsatz steigern, wenn zwei MAC-Cores verwendet werden dürfen?

## 5.8 Verständnisfragen zu Filtern → M5.8

Kreuzen Sie in Tab. 5.3 an, welche Aussage auf welche Filter zutrifft. Kann je nach Systementwurf „Ja“ oder „Nein“ zutreffen, kreuzen Sie beides an.

Aussage	FIR-Filter		IIR-Filter	
	ja	nein	ja	nein
Rekursiv:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Endliche Impulsantwort:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Instabil:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Linearphasig:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Minimalphasig:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Minimalphasig und linearphasig:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pole außerhalb des Ursprungs:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pole außerhalb des Einheitskreises:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nullstellen außerhalb des Ursprungs:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nullstellen außerhalb des Einheitskreises:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Halbbandfilter:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Konstante Gruppenlaufzeit:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Transponierte Form:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entwurf über bilineare Transformation:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Transversale Struktur:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Abgeleitet von analogen Butterworth-Filters:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tab. 5.3: Verständnisfragen zu Filtern (Aufgabe 5.8)



## 6 FIX: Wortlängeneffekte und Fixpoint-Systeme im Zeitbereich

### Lernziele

Dieses Kapitel beschäftigt sich mit der Repräsentation von Zahlen im Binärsystem und den Auswirkungen einer begrenzten Amplitudenauflösung (Quantisierung) im *Zeitbereich*. Fixpoint Systeme und Fixpoint-Arithmetik lassen sich mit deutlich geringeren Anforderungen an die Hardware implementieren als Floating Point Systeme. Nachteilig sind der deutlich geringere Dynamikbereich bei gleicher Wortlänge und die schlechtere Portierbarkeit auf Systeme mit anderen Fixpointformaten.

### Grundlagen

**Binäre Zahlendarstellung:** Sie stellen Integer- und rationale Zahlen mit und ohne Vorzeichen im Binärsystem dar. Sie kennen den Zusammenhang zwischen begrenzter Wortlänge und dem maximal darstellbaren Wertebereich (Full Scale Range, *FSR*) bzw. der maximalen Amplitude eines symmetrischen Sinussignals und können beides in einander umrechnen.

**Quantisierung:** Sie wissen, dass die meisten rationalen Zahlen in Fixpointsystemen nur näherungsweise repräsentiert werden können und berechnen den entstehenden Fehler.

### Neue Themen

**Filter mit quantisierten Koeffizienten:** Sie wissen wie die unvermeidliche Quantisierung der Koeffizienten von Fixpoint-Filters deren Performance beeinflusst (Eckfrequenzen, Amplitudengang, Welligkeit, Stabilität). Sie kennen robuste IIR-Filter-Topologien (kaskadierte Filter, LDI-Filter), die weniger empfindlich auf Variation der Koeffizienten sind als Direktformfilter.

**Fixpoint-Arithmetik:** Sie können Fixpoint-Systeme so auslegen, dass trotz Wortlängenwachstum bei Addition und Multiplikation keine Überläufe entstehen und kennen die Vor- und Nachteile von Sättigungslogik.

**Requantisierung:** Sie kennen die verschiedenen Methoden (Abschneiden, Runden, Betragsschneiden) zur Requantisierung (Wortlängenänderung) von Signalen und wissen welche Methoden DC-Bias

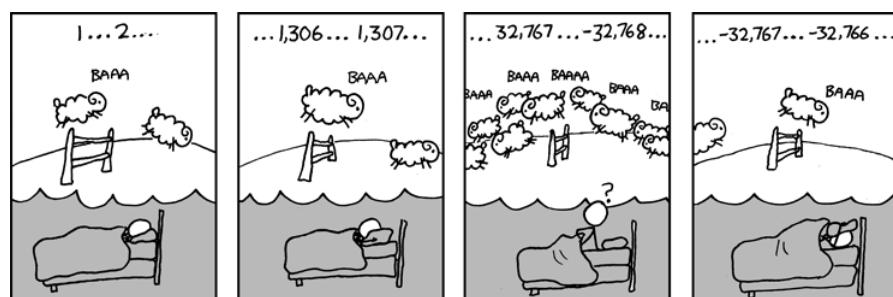


Abb. 6.1: Schlauflose Programmierer [<http://xkcd.com/571/>]

## Theorie

### Quantisierungseffekte und Wortlängenwachstum

Bei der Fixed-Point Zahlendarstellung kann man die quantisierte (d.h. auch nicht-lineare) Natur im Gegensatz zum Floating Point Zahlenformat<sup>1</sup> nicht mehr vernachlässigen.

**Wertebereich:** Der maximal darstellbare Zahlenbereich ist durch Wortlänge  $W$  vorgegeben. Achtung:  
Bei mehrstufigen Filtern können Resonanzüberhöhungen zu Überläufen im Filter führen!

**Koeffizientenquantisierung:** Die Quantisierung der Filterkoeffizienten ändert die Übertragungsfunktion und kann bei rekursiven Filtern sogar zu Instabilitäten führen. Durch Abschätzungen und Simulationen kann die Anzahl der Fractionalbits so angepasst werden, dass das gewünschte Toleranzschema des Filters erreicht wird.

**Requantisierung:** Da bei Multiplikationen die Anzahl der Nachkommastellen  $WF$  wächst und bei Additionen die Anzahl der Vorkommastellen  $WI$ , müssen vor allem bei rekursiven Systemen die Wortlängen im System durch Requantisierung angepasst werden.

**Wortlängenwachstum:** Bei der Addition von 2 Fixpointzahlen muss man für beliebige Werte *ein* zusätzliches Integerbit vorsehen,  $\Delta WI = 1$ . Addiert man  $N$  Zahlen benötigt man im worst case

$$\Delta WI = \lceil \log_2 N \rceil$$

zusätzliche Integerbits.

**Tipp:** Sie können den Logarithmus zur Basis 2 mit dem Taschenrechner berechnen über  $\log_2(x) = \log_{10}(x) / \log_{10}(2)$ .

**Koeffizientenfläche** wird die Summe des Betrags aller Koeffizienten eines FIR-Filters genannt:

$$A_b = \sum |b_i| \quad \Rightarrow \quad \Delta WI = \lceil \log_2 A_b \rceil$$

Mit Hilfe der Koeffizientenfläche kann man das Wortlängenwachstum präzise vorhersagen und damit die benötigte Wortbreite des Akkumulators. Kleinere Koeffizienten führen zu kleineren Teilprodukten und damit zu geringerem „bit growth“ des Akkumulators.

### (Re-)Quantisierungsmethoden

In Fixpointfiltern muss die Wortlänge zwischendurch fast immer requantisiert werden, um das Wortlängenwachstum zu begrenzen. Hierfür gibt es verschiedene Methoden:

**Wert- oder Abschneiden** (truncation) benötigt den geringsten Hardwareaufwand, führt aber zu einem systematischen DC-Fehler (DC-Bias), außerdem kann der Betrag durch Abschneiden größer werden und so zu Grenzzyklen führen.

**Runden** (rounding) vermeidet den systematischen DC-Fehler, aber auch durch Runden kann der Betrag ansteigen.

**Betragsschneiden** (round towards zero) garantiert, dass der Betrag ist immer kleiner ist als der ursprüngliche Wert. Nachteilig ist der größere Hardware-Aufwand und der größere Quantisierungsfehler.

---

<sup>1</sup>Floating-Point Zahlen sind zwar auch quantisiert, haben aber eine so große Dynamik, dass Quantisierungseffekte nur selten wahrnehmbar sind.

In Abb. 6.2 ist die Wahrscheinlichkeitsdichtefunktion (probability density function, PDF) des Quantisierungsfehlers dargestellt, der bei den drei Quantisierungsverfahren jeweils mit Quantisierungsstufe  $q$  entsteht.

Das Maximum der Wahrscheinlichkeitsdichtefunktion  $p$  in Abb. 6.2 ergibt sich daraus, dass die graue Fläche unter der PDF 1 sein muss (= der Quantisierungsfehler ist immer im irgendwo im skizzierten Bereich der x-Achse).

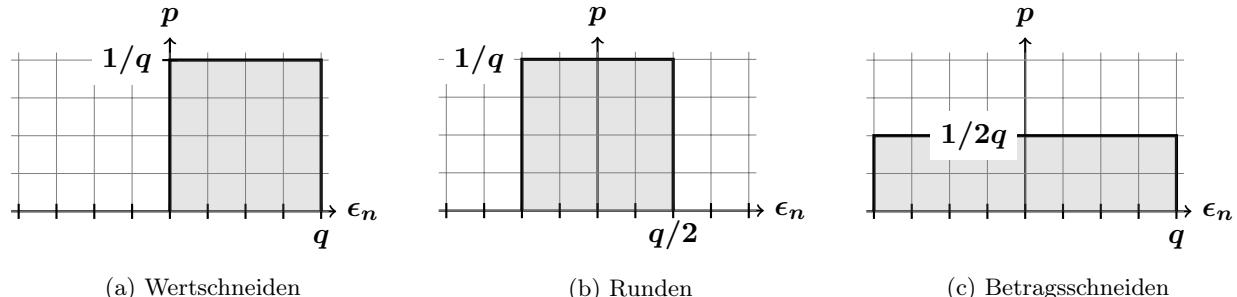


Abb. 6.2: Verteilungsdichtefunktionen des Quantisierungsfehlers  $\epsilon_n$  für verschiedene Requantisierungen

Quantisieren kann zu einem systematischen Fehler  $\mu$  (DC-Bias) führen:

$$\mu = E \langle \epsilon_n \rangle = \int_{-\infty}^{\infty} \epsilon_n p(\epsilon_n) d\epsilon_n \quad (6.1)$$

Bei Wertschneiden (Truncation) erhält man

$$\mu = \int_0^q \epsilon_n \frac{1}{q} d\epsilon_n = \frac{\epsilon^2}{2q} \Big|_0^q = \frac{q}{2}, \quad (6.2)$$

die beiden anderen Verfahren sind mittelwertfrei.

Die Varianz erhält man durch Integrieren des quadratischen Fehlers:

$$\sigma^2 = E \langle (\epsilon_n - m)^2 \rangle = \int_{-\infty}^{\infty} \epsilon_n p^2(\epsilon_n) d\epsilon_n \quad (6.3)$$

## Spezielle Python-Funktionen

In Python und Matlab ist der Umgang mit binären Fixed-Point Objekten eher exotisch und muss nachgerüstet werden. In Matlab gibt es hierzu die Fixed-Point Toolbox, in Python habe ich selbst die Klasse **Fixed()** geschrieben.

Das Notebook **FIX\_intro** zeigt, wie man die Python-Klasse **Fixed** anwendet: Zunächst wird ein Dictionary mit den Quantisierungsoptionen definiert (Anzahl der Integer- und Fractionalbits, Quantisierungsmethode und das Verhalten bei Überlauf).

Beim Instanziieren der Klasse wird das Dictionary als Parameter übergeben, danach wird z.B. der Methode **fix()** der zu quantisierende Wert oder Vektor übergeben. Zurückgegeben werden der quantisierte Wert oder Vektor. Die Anzahl der aufgetretenen Überläufe ist im Attribut **N\_over** gespeichert.

Die wichtigsten Quantisierungsmethoden sind dabei '**floor**' (Wertschneiden), '**round**' (Runden) und '**fix**' (Betragsschneiden).

Optionen für das Verhalten beim Verlassen des Wertebereichs sind '`'none'`' (keine Überläufe; Anzahl der Integerbits wird ignoriert), '`'wrap'`' (Überläufe mit Zweierkomplementverhalten - „nach ganz groß kommt ganz klein“) und '`'sat'`' (Begrenzung auf Minimal- bzw. Maximalwert).

## Notebooks und Labs

Filename	Beschreibung
<b>Schnipsel</b>	
<code>FIX_pyaudio_basics</code>	Grundgerüst für Echtzeit-Audiosignalverarbeitung in Python
<code>FIX_pyaudio-quantization</code>	Audiodemonstration von Quantisierungs- und Sättigungseffekten
<code>FIX_pyaudio-limit_cycles</code>	Audiodemonstration von Grenzzyklen in rekursiven Filtern (Code ist noch zu langsam für Echtzeit)
<b>Notebooks</b>	
<code>FIX_intro</code>	Quantisierungskennlinie und Wellenform bei verschiedenen Arten der Quantisierung und Sättigung
<code>FIX_FIR_quant_ML</code>	Notebook mit Musterlösung zu Aufgabe 6.4
<code>FIX_limit_cycles</code>	Grenzzyklen in IIR-Filtern

Tab. 6.1: Codeschnipsel und Jupyter Notebooks zu Kap. 6

## 6.1 Zahlendarstellung im Zweierkomplementformat → M6.1

- a) Welchen **dezimalen Wert** stellt **10101** dar für die Fälle: Signed / Unsigned Integer und Signed Fractional mit 2 Vorkommastellen ( $WI = 2$ )?
- b) Stellen Sie die Zahlen 2, 2,5,  $-3/4$ ,  $+1/3$  im **Q(2.6) Format** dar!
- c) Die Werte  $7/16$ ,  $-9/16$  und  $-11/16$  sollen mit einer Wortlänge von 5 Bit im Normalformat dargestellt und durch Wertschneiden, Runden und Betragsschneiden auf eine Wortlänge von 3 bit **requantisiert** (verkürzt) werden. Wie groß ist jeweils der Quantisierungsfehler? Nimmt der Betrag beim Requantisieren zu?

## 6.2 Analog-Digital-Wandlung eines Drucksensors → M6.2

Der Beschleunigungssensor ADXL 327 kann Beschleunigungen im Bereich  $a = \pm 2$  g messen und hat eine Empfindlichkeit von  $S_{Sens} = 400$  mV/g. Bei  $a = 0$  ist die Ausgangsspannung  $U_{Sens} = 1,5$  V. Die Ausgangsspannung des Sensors wird digitalisiert mit einem nachgeschalteten 8 Bit ADC mit Eingangsspannungsbereich (Full Scale Range)  $FSR = 0 \dots 5$  V.

- a) Bestimmen Sie die **Auflösung des Wandlers**  $q_U$  in V, d.h. die Änderung der Eingangsspannung, die einem LSB entspricht. Mit welcher Unsicherheit repräsentiert das Digitalwort einen analogen Wert?
- b) Welche **Auflösung bezogen auf die Messgröße**  $q_a$  in g erhält man, wenn der Wandler direkt an den Sensor angeschlossen wird?

- c) Bei  $a = 0$  soll der digitale Ausgangswert des ADCs 0000 0000 sein (im ZK-Format). Wie muss das Ausgangswort dazu **digital korrigiert** werden?
- d) Welchen **Digitalwert** (nach der Korrektur) liefert der ADC bei einer Beschleunigung von  $\pm 1,2 \text{ g}$ ? Wie werden diese Werte binär dargestellt, welchen Werten entsprechen sie im *normierten Zweierkomplement-Fraktional-Format* (nZKF-Format, Q0.7)?
- e) Welche **maximale Amplitude** für analoges Signal bzw. Beschleunigung ist digital darstellbar?
- f) Welcher **Beschleunigung** entspricht das binäre Wort 1000 0001?
- g) Das Sensorsignal soll jetzt **analog vorverstärkt** werden, damit es optimal an den Eingangsspannungsbereich des Wandlers angepasst ist. Welche Auflösung  $q_a$  in g erhalten Sie dadurch?
- h) Wieviele **Codeschritte des Wandlers** überstreichen Sie jeweils mit dem Spannungsbereich des Sensors bei beiden Lösungen? Welcher äquivalenten Wortlänge (Effective Number of Bits, *ENOB*) entspricht das jeweils?
- i) Zur **digitalen Filterung** des Sensorsignals wird ein MA-Filter der Ordnung 15 ohne Skalierung verwendet. Welche Wortlänge und welches Zahlenformat ist am Ausgang des Filters erforderlich, damit keine Überläufe auftreten können? Bei welcher normierten Frequenz liegt die erste Nullstelle des Betragsgangs?

### 6.3 Moving Average Filter mit endlicher Wortbreite → M6.3

Ein Moving Average Filter mit der Ordnung 8 (= 8 Register) soll mit einer Eingangswortbreite von 10 bit in Direktform und in transponierter Form (Abb. 6.3) aufgebaut werden, ohne dass im Filter Bits abgeschnitten werden.

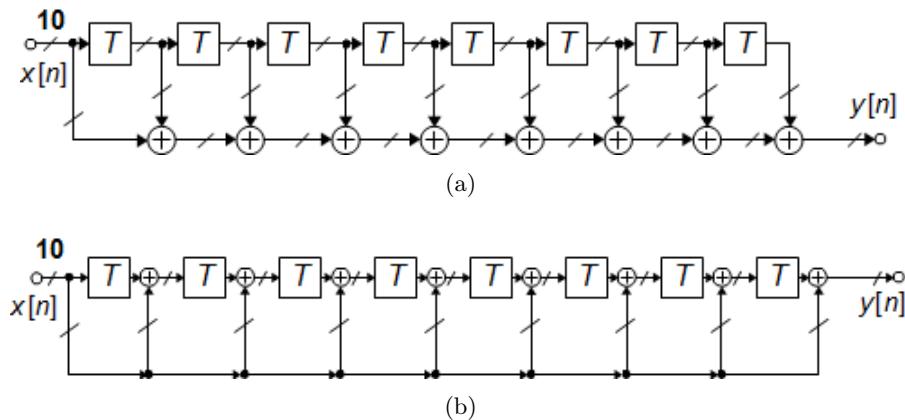


Abb. 6.3: Fixpoint Moving Average Filter zu Aufgabe 6.3, (a) Direktform und (b) transponierte Direktform

- a) Geben Sie die unquantisierte **Systemfunktion**  $H(z)$  und den **Amplitudenfrequenzgang**  $H(e^{j\Omega})$  an. Bei welcher Frequenz liegt die erste Nullstelle im Frequenzgang?
- b) Zeichnen Sie die **Wortbreiten** in Abb. 6.3 ein. Wie groß ist jeweils die Wortbreite am Ausgang?
- c) Welche Verzögerung hat jeweils der **kritische Pfad**? Mit welcher Taktrate kann das Filter maximal arbeiten? Nehmen Sie an, dass alle Addierer die gleiche Verzögerung  $\tau_{add} = 2 \text{ ns}$  haben und die Setup / Hold-Zeiten der Register vernachlässigt werden können.
- d) Wie viele **Flipflops** (= Register x Wortbreite) und **Volladdierer** (= Addierer x Wortbreite) benötigen die beiden Realisierungen?

## 6.4 FIR Filter mit Quantisierung → M6.4

Der Entwurf eines FIR-Filters hat die folgenden Koeffizienten geliefert:

$$\mathbf{b} = [0, 0.01623, 0, -0.06871, 0, 0.30399, 0.5, 0.30399, 0, -0.06871, 0, 0.01623]$$

Dieses Filter soll in Direktform auf einem FPGA implementiert werden (Abb. 6.4), das hardware-optimierte  $18 \times 18$  Bit Multiplizierer enthält. Die Daten am Eingang  $x[n]$  und am Ausgang  $y[n]$  in normalisierter Zweierkomplementform haben eine Wortlänge von 16 Bit,  $Q_X 0.15 = Q_Y 0.15$  (vorzeichenbehaftet, 15 fractional Bits, Wertebereich  $-1 \leq x, y < 1$ ).

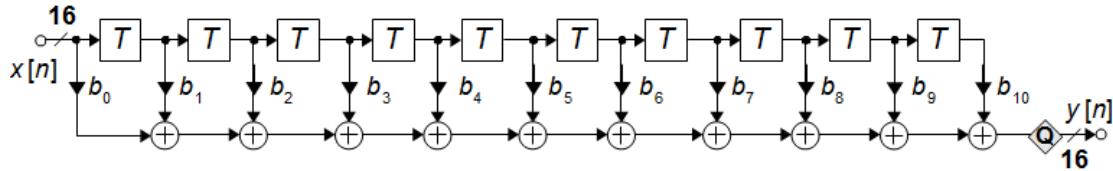


Abb. 6.4: FIR-Filter mit Quantisierung zu Aufgabe 6.4

**Hinweis:** Die folgenden Fragen lassen sich mit wenigen Ausnahmen unabhängig voneinander lösen.

- a) **Allgemeine Fragen:** Welche Ordnung hat das Filter? Ist das Filter linearphasig? Ist es ein Halbbandfilter? Wenn ja, werden diese Eigenschaften durch Koeffizientenquantisierung gestört?
- b) Wieviele „echte“ **Multiplizierer** benötigt das Filter? Es sollen Symmetrieffekte und offensichtliche Vereinfachungen ausgenutzt werden. Zeichnen Sie die optimierte Struktur auf!
- c) Welche Verzögerung hat der **kritische Pfad** dieser Implementierung? Mit welcher Taktrate kann das Filter maximal arbeiten? Nehmen Sie an, dass alle „echten“ Multiplizierer die gleiche Verzögerung  $\tau_{mult} = 10$  ns und alle Addierer die gleiche Verzögerung  $\tau_{add} = 2$  ns haben. Die Setup / Hold-Zeiten der Register können vernachlässigt werden. Vergleichen Sie diese Werte mit denen eines allgemeinen FIR Filters 10. Ordnung in Direktform (Abb. 6.4).
- d) Welchen Wert hat der **Amplitudenfrequenzgang  $H(f)$**  bei  $f = 0$ ,  $f = f_S/4$  und  $f = f_S/2$ ?
- e) Was ist das **optimale Datenformat  $Bx.y$**  (= minimaler Quantisierungsfehler) zur Darstellung der Koeffizienten  $b$  bei einer Gesamtwortlänge  $W = 18$  Bit? Alle Koeffizienten sollen das gleiche Format haben. Wie groß ist ein LSB in der gewählten Darstellung?
- f) **Quantisieren Sie die Koeffizienten** durch Wertschneiden (**floor**) und ermitteln Sie jeweils den Quantisierungsfehler  $\epsilon(0, QF)$ . Welche Werte ergeben sich für eine Wortbreite der Koeffizienten von 8 Bit? Hinweis: Die Bestimmung mit Hilfe eines Python / Matlab-Skripts ist deutlich einfacher als eine Handrechnung ... Simulieren Sie für diese Darstellung den Frequenzgang des Filters.
- g) Welche **Koeffizientenfläche  $A_i$**  hat das Filter?
- h) Wie muss das **Eingangswort (16 bit)** skaliert werden vor der Multiplikation ( $18 \times 18$  bits)? Welche Bits des Produkts sind (für beliebige Eingangsworte und Koeffizienten) relevant?
- i) Welche **Wortbreite muss der Akkumulator** haben wenn das Ergebnis zur Minimierung der Rundungsverluste erst nach Aufsummierung aller Teilprodukte auf 16 Bit am Ausgang requantisiert wird? Bestimmen Sie die Wortbreite zunächst für beliebige Koeffizienten und danach unter Berücksichtigung der Koeffizientenfläche.

- j) Wie muss das **Akkumulatorwort skaliert** werden (d.h. welcher Teil wird benutzt, welche Bits werden abgeschnitten), um den Quantisierungsfehler zu minimieren? Es darf dabei kein Überlauf bei beliebigen Eingangswerten  $x[n]$  auftreten.

## 6.5 + FIR Filter mit Skalierung → M6.5

Das FIR-Filter F1 aus Aufgabe 5.6 mit der Impulsantwort  $h_1[n] = \{1; 7/4; 1/2; -1/4\}$  und dem Betragsfrequenzgang in Abb. 6.5 soll jetzt in Fixpoint-Arithmetik implementiert werden mit einem Wertebereich  $-1 < x, y < 1$  am Eingang und Ausgang, das Filter ist in Direktform aufgebaut.

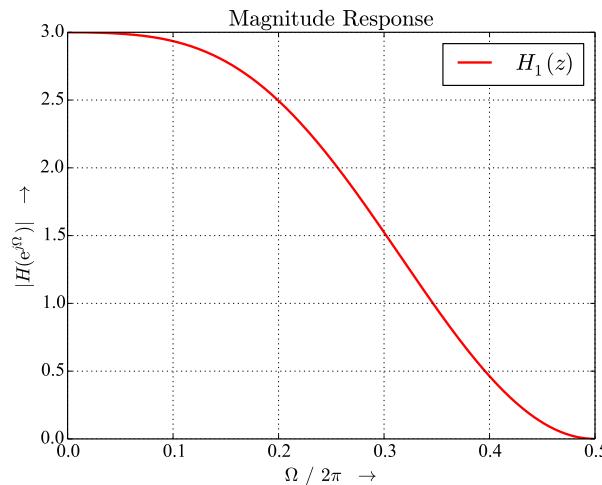


Abb. 6.5: Betragsfrequenzgang zu Aufgabe 6.5

- Bestimmen Sie die **DC-Verstärkung** des Filters.
- Skizzieren den **Signalfußgraph** (die Struktur) des Filters. An welcher Stelle kann im Filter ein Überlauf auftreten?
- Bestimmen Sie den **Skalierungskoeffizienten  $k_{E_\infty}$**  so, dass für beliebige sinusförmige Eingangssignale oder Gleichspannung ( $f = 0$ ) kein Überlauf entsteht (sog.  $L^\infty$ -Skalierung). Mit welcher Eingangsfolge  $x_\infty[n]$  wird bei dieser Skalierung  $y[n]_{max} = 1$  erreicht?
- Bestimmen Sie den **Skalierungskoeffizienten  $k_{E1}$**  so, dass für ein *beliebiges Eingangssignal* kein Überlauf entsteht (sog.  $L^1$ -Skalierung). Überlegen Sie dazu zuerst, welche Eingangsfolge  $x_1[n]$  der worst case für das System ist, also bei welcher Eingangsfolge das maximale Ausgangssignal entsteht.

## 6.6 + Integrator mit endlicher Wortbreite → M6.6

In dieser Aufgabe soll bestimmt werden, ob und wann die Integratoren in Abb. 6.6 mit einer Wortlänge von 8 bit überlaufen, wenn am Eingang ein Einheitsimpuls  $\delta[n]$  angelegt wird. Zum Zeitpunkt  $n = 0$  haben alle Integratoren den Wert 0.

Skizzieren Sie außerdem Pol-Nullstellendiagramm und den Frequenzgang (für Integratoren ohne Wortlängenbegrenzungen). Bei welcher Frequenz wird  $|H(e^{j\Omega})| = 1$ ?

Vor allem die Integratoren b) und c) lassen sich am einfachsten mit Hilfe der z-Transformation analysieren. Hierfür sind die folgenden Transformationspaare nützlich:

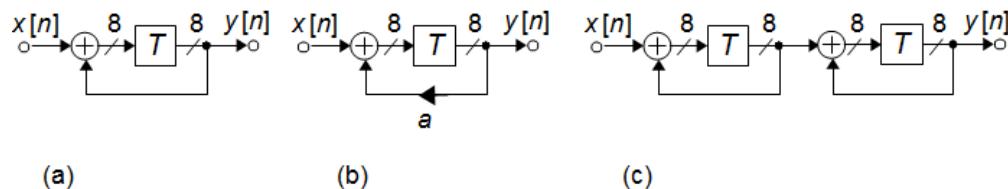


Abb. 6.6: Integratoren mit endlicher Wortbreite zu Aufgabe 6.6, (a) erster Ordnung, (b) erster Ordnung, verlustbehaftet, (c) zweiter Ordnung

$$\begin{aligned}
 u[n] & \circlearrowleft \bullet \quad \frac{z}{z-1} = \frac{1}{1-z^{-1}} \\
 nu[n] & \circlearrowleft \bullet \quad \frac{z}{(z-1)^2} = \frac{z^{-1}}{(1-z^{-1})^2} \\
 n^2u[n] & \circlearrowleft \bullet \quad \frac{z(z+1)}{(z-1)^3} = \frac{z^{-1}(1+z^{-1})}{(1-z^{-1})^3} \\
 a^n u[n] & \circlearrowleft \bullet \quad \frac{z}{z-a} = \frac{1}{1-az^{-1}}
 \end{aligned}$$

Untersuchen Sie auch die Sprungantwort von Integrator a). Die Sprungantwort von Integrator b) und c) lässt sich über Partialbruchzerlegung berechnen (komplizierter, ohne Musterlösung).

## 6.7 Verständnisfragen zu Filtern mit Quantisierung → M6.7

Im Folgenden sollen Filter betrachtet werden, bei denen Koeffizienten und Arithmetik quantisiert sind. Kreuzen Sie in Tab. 6.2 an, welche Aussage auf welche Filter zutrifft. Kann je nach Systementwurf „Ja“ oder „Nein“ zutreffen, kreuzen Sie beides an.

Aussage	FIR-Filter		IIR-Filter	
	ja	nein	ja	nein
Unendliche Impulsantwort:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Stabil:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Linearphasig:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Minimalphasig:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pole außerhalb des Ursprungs:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Pole außerhalb des Einheitskreises:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nullstellen außerhalb des Ursprungs:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Nullstellen außerhalb des Einheitskreises:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Halbbandfilter:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Konstante Gruppenlaufzeit:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Transponierte Form:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Entwurf über bilineare Transformation:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Transversale Struktur:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Abgeleitet von analogen Butterworth-Filters:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Kaskadierte Form ist unempfindlicher gegen Quantisierungseffekte als direkte Form:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Grenzzyklen:	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Tab. 6.2: Verständnisfragen zu Filtern mit Quantisierung (Aufgabe 6.7)



## 7 NOI: Wortlängeneffekte im Frequenzbereich: Quantisierungsrauschen

### Lernziele

Dieses Kapitel beschäftigt sich mit der Beschreibung der Signalquantisierung über *statistische Größen* und im *Frequenzbereich* als *Quantisierungsrauschen*.

- Sie beschreiben statistisch die Effekte der **Analog-Digital-Wandlung (Quantisierung)** eines analogen (= unendlich fein aufgelösten) Signals als Rauschen und können dessen Leistung berechnen.
- Sie kennen die Auswirkungen der **Requantisierung** (Wortbreitenreduktion) eines bereits quantisierten Signals und berechnen die Verringerung des *SNR*.
- Sie berechnen **Rauschleistung  $N$**  und **Rauschleistungsdichte  $N'(f)$**  eines quantisierten Signals aus dem Quantisierungsschritt  $q$  (LSB) und der Bandbreite.
- Sie bestimmen das **Signal- zu Quantisierungsrauschleistungsverhältnis  $SQNR$**  in Abhängigkeit von Aussteuerung und dem Quantisierungsschritt  $q$ . Sie können den Unterschied erklären zwischen dem *maximalen SQNR* eines quantisierten Systems und dem tatsächlich erreichten *SNR*.
- Sie erklären das Konzept der **Effective Number of Bits  $ENOB$**  und berechnen den Wert aus dem *SQNR*.

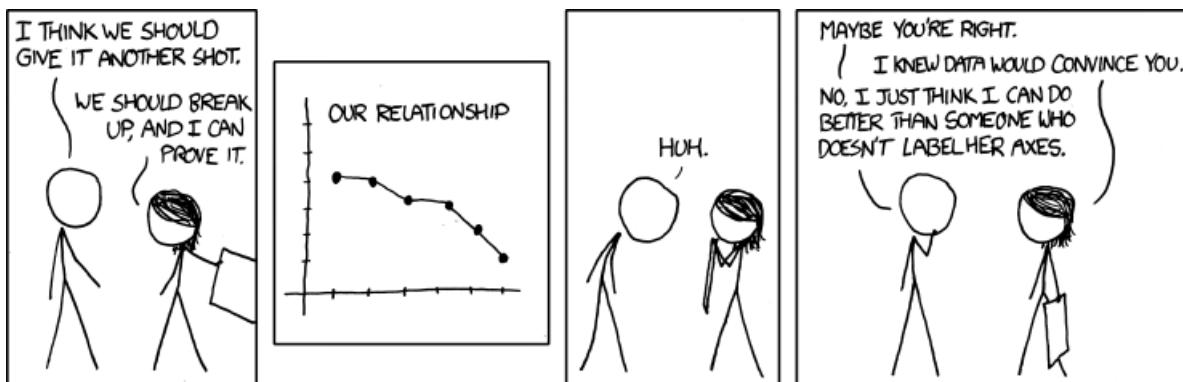


Abb. 7.1: „And if you labeled your axes, I could tell you exactly how MUCH better“ [<http://xkcd.com/833/>]

## Theorie

### Quantisierungsrauschen

Im Kap. 6 wurde untersucht, welche *Quantisierungsfehler*  $\epsilon[n] = x[n] - x_q[n]$  auftreten, wenn man den unendlich fein aufgelösten Wertebereich eines analogen Signals *diskretisiert* oder *quantisiert*, also auf eine endliche Anzahl  $M = 2^W$  von Werten abbildet. Reduziert man die Wortlänge  $W$  eines bereits quantisierten Signals, verringert man also die Anzahl der zur Verfügung stehenden Werte, entstehen ebenfalls Quantisierungsfehler. Diesen Prozess nennt man *Requantisierung*. Beide Vorgänge können mit den gleichen mathematischen Methoden untersucht werden. Wichtig ist hier vor allem der Wertebereich oder *Full Scale Range (FSR)* und die Höhe  $q$  der Quantisierungsstufen:

$$FSR = (M - 1)q = (2^W - 1)q \approx 2^W q \quad \Leftrightarrow \quad q = \frac{FSR}{2^W - 1} \approx \frac{FSR}{2^W} \quad \text{für } 2^W \gg 1 \quad (7.1)$$

Die ohne Übersteuerung maximal darstellbare Amplitude eines symmetrischen Wechselsignals ist

$$A_{max} = FSR/2 \approx 2^{W-1}q. \quad (7.2)$$

Die maximale Signalleistung eines *sinusförmigen* Signals ist daher

$$S_{max} = A_{max}^2/2 = FSR^2/8. \quad (7.3)$$

Im Kap. 6 wurden diese Vorgänge im Zeitbereich (z.B. Grenzzyklen) betrachtet, und es wurde untersucht wie man Systeme skalieren muss, damit kein Überlauf bzw. Sättigung auftritt. Außerdem wurde betrachtet, welchen Einfluss die Quantisierung der Filterkoeffizienten auf den Frequenzgang des Filters hat. Die Quantisierung der *Koeffizienten* eines linearen Systems  $H(z)$  führt zu einem neuen linearen<sup>1</sup> System  $H_Q(z)$ , das mit den bisherigen Methoden der Signalverarbeitung beschrieben werden kann. Die

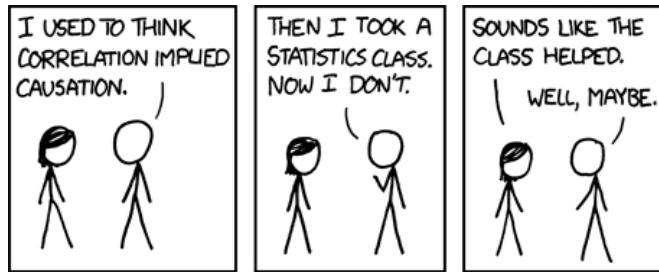


Abb. 7.2: Korrelation [<http://xkcd.com/552/>]

Quantisierung des *Signals* führt dagegen zu einem nicht-linearen System, das mit den üblichen Methoden der Signaltheorie nicht beschrieben werden kann. Daher beschreibt man Quantisierungsfehler als statistischen Prozess und kann so das System trotz der offensichtlichen Nichtlinearität der Quantisierung als linear betrachten und darf die mathematischen Methoden für LTI-Systeme ( $z$ -Transformation, Fourier-Transformation etc.) weiterhin verwenden. Der Quantisierungsfehler wird in diesem Kapitel als stationärer ergodischer<sup>2</sup> Zufallsprozess („Quantisierungsrauschen“) betrachtet, der durch seinen Mittelwert  $m$  und die Varianz  $\sigma^2$  beschrieben wird. Die AC-Leistung des Fehlers ist gleich der Varianz,  $N_Q = \sigma^2$ . Für die gängigen Quantisierungsmethoden *Runden* und *Wertschneiden* (truncation) gilt  $\sigma^2 = q^2/12$  (Herleitung z.B. [OS10], [KK18]):

$$q = 2^{-W} FSR \quad \Rightarrow \quad N_Q = \sigma^2 = \frac{q^2}{12} = \frac{FSR^2}{12 \cdot 2^{2W}} \quad (7.4)$$

<sup>1</sup>sofern es stabil ist

<sup>2</sup>„stationär“: unabhängig vom Zeitpunkt, „ergodisch“: Zeit- und Scharmittelwerte sind gleich.

## SQNR

Bei einer vorzeichenbehafteten Repräsentation  $X(WI.WF)$  benötigt man ein Bit für das Vorzeichen („sign“, S). Bei  $N = 2^W$  Quantisierungsstufen gilt daher  $WI+1+WF = W$ . Der darstellbare Zahlenbereich ist  $-2^{WI} \leq x \leq 2^{WI} - 2^{-WF} \approx 2^{WI}$ . Ein Sinussignal, das den darstellbaren Bereich voll ausnutzt, hat daher eine Amplitude von  $A_{max} = 2^{WI}$  und eine Leistung von  $S_{max} = A_{max}^2/2 = 2^{2WI}/2$ .

Die Rauschleistung eines quantisierten Signals wird bestimmt durch die Größe der Quantisierungsstufe  $q = 2^{-WF} = 1$  LSB - je feiner die Quantisierung, desto geringer der Quantisierungsfehler und damit im Mittel das Quantisierungrauschen. Wird das quantisierte Signal durch Runden oder Wertschneiden aus dem ursprünglichen Signal gewonnen, beträgt die Quantisierungrauschleistung  $N_q = q^2/12 = 2^{-2WF}/12 = 2^{-2WF-2}/3$ .

Das Verhältnis zwischen Signal- und Quantisierungrauschleistung nennt man *Signal-to-Quantization Noise Ratio (SQNR)*:

$$SQNR = \frac{S}{N_Q} \quad \text{oder} \quad SQNR[\text{dB}] = 10 \log_{10} \frac{S}{N_Q} \text{ dB} = S[\text{dB}] - N_Q[\text{dB}] \quad (7.5)$$

Das maximal<sup>3</sup> erreichbare *SQNR* für *sinusförmige* Signale, das  $SQNR_{max}$  in dB ist:

$$\begin{aligned} SQNR_{max} &= \frac{S_{max}}{N_Q} = \frac{FSR^2}{8} \frac{12 \cdot 2^{2W}}{FSR^2} = \frac{3}{2} \cdot 2^{2W} \\ &\cong 10 \log 1,5 \text{ dB} + 10 \log 2^{2W} \text{ dB} = (1,76 + 6,02W) \text{ dB} \end{aligned} \quad (7.6)$$

Manchmal werden bei logarithmischen Rechnungen die Pseudo-Einheiten dBV und dBW verwendet, sie sind definiert über  $a \text{ dBV} = 20 \log_{10}(A/1 \text{ V})$  bzw.  $p \text{ dBW} = 10 \log_{10}(P/1 \text{ W})$ .

**Beispiel:** ADC mit  $W = 18$  bit Auflösung und Eingangsbereich von  $\pm 1 \text{ V}$ :

$$\begin{aligned} &\Rightarrow FSR = 2 \text{ V}, A_{max} = 1 \text{ V}, S_{max} = 0,5 \text{ V}^2 \cong -3,01 \text{ dBV} \\ &\Rightarrow q = 2^{-18} \cdot FSR = 7,63 \mu\text{V} \cong -102,4 \text{ dBV}, N_Q = 4,85 \cdot 10^{-12} \text{ V}^2 \cong -113,1 \text{ dBW} \\ &\Rightarrow SQNR_{max} = 1,031 \cdot 10^{-11} \cong 110,1 \text{ dB}. \end{aligned}$$

**Anmerkung:** Im physikalischen Sinne ist  $\text{V}^2$  natürlich keine Leistung. In der Nachrichtentechnik ist es aber üblich quadrierte Größen als Leistung zu bezeichnen, da durch Division durch einen (Abschluss-)Widerstand jederzeit die physikalische Leistung berechnet werden kann. Meist werden sowieso Leistungsverhältnisse betrachtet (*SNR*, Verstärkung), aus denen der Widerstand herausfällt.

**Noch eine Anmerkung:** Wenn man von anderen Signalformen ausgeht, ergeben sich auch bei Vollaussteuerung leicht unterschiedliche *SQNR*-Werte: Ein dreieckförmiges Signal hat eine Leistung von  $A^2/3$ , damit ergibt sich  $SQNR_{max} = W \cdot 6,02 \text{ dB}$ .

**Quantisierung am DAC:** Der DAC an sich quantisiert das Signal nicht (und fügt daher auch kein Quantisierungrauschen hinzu!), sofern das digitale Signal die gleiche Wortlänge hat wie der DAC. Das ist aber meist nicht der Fall; in der digitalen Signalverarbeitung wird man meist ein paar Reservebits zur Verringerung von Rundungsverlusten vorhalten, während beim DAC als analoger Komponente jedes zusätzliche Bit die Kosten drastisch erhöht. Vor dem DAC ist daher in den allermeisten Fällen eine *Requantisierung* notwendig, die Quantisierungrauschen erzeugt - siehe nächster Abschnitt.

---

<sup>3</sup>d.h. für Vollaussteuerung

## Requantisierung

Bei einem ADC definiert die analoge Seite den *FSR* und die digitale Seite die Wortlänge *W*. Daraus ergibt sich sofort die Höhe der kleinsten darstellbaren Stufe *q* woraus wiederum das Quantisierungsrauschen folgt.

Signal- und Rauschleistung von digitalen Signalen können nicht eindeutig angegeben werden, da die Signalwerte keine direkte physikalische Bedeutung oder Einheit haben und die Position des Binärpunkts Interpretationssache ist. Das maximale *SQNR* eines requantisierten Signals kann man trotzdem mit (7.6) bestimmen, da hier nur das Verhältnis beider Größen eingeht; *W* ist dabei die Wortlänge *nach* der (Re-)quantisierung.

Wenn man zu Vergleichszwecken trotzdem *S* und *N<sub>Q</sub>* angeben möchte, empfiehlt es sich die Signalleistung wieder abhängig vom *FSR* anzugeben. Bei einer Darstellung mit *WI.WF*, also *WI* Integerbits, *WF* Fractionalbits und einem Vorzeichenbit und insgesamt *WI + 1 + WF = W* bit gilt:

$$\begin{aligned} \mathbf{FSR} &= 2^{WF} - 2^{-WF} - (-2^{WF}) = 2^{WF+1} - 2^{-WF} \approx \mathbf{2^{WF+1}} \text{ für } 2^{-WF} \ll 2^{WF} \\ \Rightarrow A_{\max} &\approx \mathbf{2^{WF}} \quad \text{und} \quad S_{\max} = \frac{A_{\max}^2}{2} \approx \frac{\mathbf{2^{2WF}}}{2} \end{aligned} \quad (7.7)$$

Das maximal erreichbare *SQNR* bei der Requantisierung erhält man immer aus der *Gesamtanzahl W = WF + WF* der Bits *nach* der Requantisierung!

Die von einer (Re-)Quantisierungsstufe erzeugte Quantisierungsrausleistung wird auch hier bestimmt durch den Quantisierungsfehler *q*, also durch die kleinste darstellbare Stufe, die wiederum von der Wortlänge *W* und dem Aussteuerbereich *FSR* abhängt:

$$q = \frac{FSR}{2^W} = \frac{2^{WF+1}}{2^{WF+1+WF}} = 2^{-WF} \Rightarrow N_Q = \frac{q^2}{12} = \frac{\mathbf{2^{-2WF}}}{12} \quad (7.8)$$

Wer möchte, kann sich vorstellen, dass die Anzahl der Integerbits den Aussteuerbereich und die Anzahl der Fractionalbits den Quantisierungsfehler bestimmen, auch wenn die Position des „Kommas“ bei Fixpointzahlen willkürlich ist. Dividiert man (7.7) und (7.8) kommt man jedenfalls zum gleichen Ergebnis wie (7.6).

**Achtung:** An unterschiedlichen Stellen im System (Eingang, Ausgang, nach Multiplikationen) werden Signale meist durch unterschiedliche Quantisierungen repräsentiert.

## Effective Number of Bits (*ENOB*)

*N<sub>Q</sub>* ist unabhängig von *A* - eine Verringerung der Signalamplitude führt daher zwangsläufig zu einem schlechteren *S(Q)NR*! Diese Verschlechterung gegenüber *SQNR<sub>max</sub>* (auch durch andere Nichtidealitäten des ADCs wie Nichtlinearitäten, thermisches Rauschen, gestörte Spannungsversorgung, Clockjitter, ...) lässt sich als verringerte effektive Wortlänge *Effective Number of Bits (ENOB)* ausdrücken:

$$ENOB = \frac{SNR [\text{dB}] - 1,76 \text{ dB}}{6,02 \text{ dB}} \text{ bit} \quad (7.9)$$

Mit dem *ENOB* kann man verschiedene ADCs oder Signalverarbeitungssysteme mit gleicher (nominaler) Wortbreite vergleichen und abschätzen, wie gut der theoretisch mögliche Signal-Rauschabstand erreicht wird.

## Rauschleistungsdichte

Da der Quantisierungsfehler als stationärer ergodischer Zufallsprozess angesehen wird (und dies in Praxis meist auch mit hinreichender Genauigkeit zutrifft), ist das Spektrum des Quantisierungsrauschen weiß<sup>4</sup> (d.h. die Rauschleistung pro Bandbreite  $\delta f$  ist für alle Frequenzen gleich). Die gesamte Quantisierungsrausleistung  $N_Q$  verteilt sich daher gleichmäßig auf das Basisband  $0 \dots f_S$  bzw.  $-f_S/2 \dots f_S/2$ , die *Rauschleistungsdichte* ist  $N'_Q = N_Q/f_S$  und wird in W / Hz angegeben. Aus Rauschleistungsdichte und Bandbreite kann man die Rauschleistung berechnen, z.B. wenn durch ein Filter Teile des Quantisierungsrauschen ausgefiltert wurden.

**Anmerkung:** Bei reellwertigen Signalen (gerades Betragsspektrum) kann man die Rauschleistungsdichte auch für ein einseitiges ( $f = 0 \dots f_S/2$ ) Spektrum angeben. Der Wert für  $N'_Q$  ist dann doppelt so groß, man darf dann aber auch nur mit der einseitigen Rauschbandbreite  $0 \dots +B_N$  rechnen und kommt insgesamt wieder auf die gleiche Rauschleistung.

Vor allem in der Hochfrequenztechnik werden beide Definitionen für die Rauschleistungsdichte / Power Spectral Density verwendet, man muss daher bei der Interpretation von Messergebnissen wissen, was gemeint ist.

## Rauschleistungsdichte und DFT

Rauschen ist ein breitbandiges Signal, dessen Leistung sich auf alle Bins einer DFT verteilt. Ändert man die Zahl der Bins  $N_{DFT}$ , ändert sich auch die *angezeigte* Leistung je Bin. Signale mit Linienspektren wie der Sinuston fallen dagegen idealerweise nur in ein Bin - hier gibt es keine Abhängigkeit von  $N_{DFT}$ . Bei Erhöhung der DFT-Länge sinkt das angezeigte Rauschen („Noise Floor“) ab und diskrete Spektrallinien treten klarer hervor. Bei konstanter Abtastfrequenz bedeutet das natürlich auch eine immer längere Messzeit.

Bei Spektralanalyse mit einer  $N_{DFT}$  Punkt DFT fällt eine Rauschleistung  $N_{Q,bin} = N_Q/N_{DFT}$  in jedes Bin ( $2N_Q/N_{DFT}$  für einseitige DFT) - dieser angezeigte Wert kann in die Rauschleistungsdichte  $N'_Q$  umgerechnet werden:

$$N'_Q = \frac{N_Q}{f_S} = \frac{N_{Q,bin}}{\Delta f_{bin}} = \frac{N_{Q,bin}}{f_S/N_{DFT}} \quad (7.10)$$

Die Bandbreite eines Bins ist  $\Delta f = f_S/N_{DFT}$ . Um die *Rauschleistungsdichte* aus der DFT direkt abzulesen, muss der angezeigte Wert daher durch diese Bandbreite geteilt werden.

## Spezielle Python-Funktionen

Spezielle Python/Matlab-Befehle werden für dieses Kapitel nicht benötigt: Die Quantisierung wurde bereits im Kap. 6 gezeigt, die Darstellung im Frequenzbereich erfolgt mit Hilfe der DFT (Kap. 3). Aufbauend auf dem Notebook **FIX\_intro** zeigt das Notebook **NOI-intro** wie man Quantisierungsrauschen, Rauschleistungsdichte und Rauschleistung in Python simuliert. Ein paar Dinge muss man dabei beachten:

**Kohärente DFT:** Wenn man periodische Signale quantisiert, sollte man die DFT unbedingt von einer ganzen Anzahl von Perioden bestimmen, ausgeschnitten mit einem Rechteckfenster. Die Anzahl der Perioden sollte eine Primzahl sein; dadurch werden Periodizitäten des Quantisierungsfehlers minimiert und das Rauschspektrum ist gleichmäßiger verteilt.

---

<sup>4</sup>in Analogie zu weißem Licht

**Amplituden und Leistungen:** Die DFT gibt die komplexe *Amplitude*  $\underline{A}_k$  in jedem Frequenzband zurück. Möchte man die *Leistung* im Frequenzband  $k$  wissen, muss man  $P_k = \underline{A}_k \underline{A}_k^*/2 = |\underline{A}_k|^2/2$  berechnen!  $20 \log_{10}(|\underline{A}_k|)$  ist 3 dB größer als die erwartete Leistung in dB. Man kann die DFT durch  $\sqrt{2}$  geteilt<sup>5</sup> und erhält dann Effektivwerte je Band anstatt der Amplituden, aus denen man direkt die Leistung berechnen kann.

**Ein- oder zweiseitiges Spektrum:** Möchte man nur physikalische (positive) Frequenzen darstellen, muss man die DFT-Amplituden (Beträge!) bei negativen Indices zu denen bei positiven Indices addieren, um auch im Frequenzbereich korrekte Leistungswerte zu erhalten. Bei reellwertigen Zeitsignalen kann man einfach alle Werte der DFT verdoppeln - bis auf den DC - Wert (Index 0)!

**Mittlere Rauschleistungsdichte:** Theoretisch sollte die Quantisierungsrausleistung bei allen Frequenzen gleich sein, praktisch gibt es bei der Quantisierung von sinusförmigen Signalen Korrelationseffekte. Daher mittelt man die Rauschleistung aller Frequenzbänder - *nach der Leistungsberechnung (Quadrieren<sup>6</sup>) und vor dem Logarithmieren!* Die Amplitude des unquantisierten Signals muss man vor der Berechnung im entsprechenden Frequenzband subtrahieren, da man ja nur die Rauschleistung berücksichtigen möchte. Den Index erhält man über  $k = f_{sig}/\Delta f = N f_{sig}/f_S$ .

**Rauschleistung:** Hat man die mittlere Rauschleistungsdichte bestimmt, muss man sie nur noch mit  $N_{FFT}/2$  multiplizieren - *vor dem Logarithmieren!* - um die gesamte Rauschleistung zu erhalten.

Details und Codebeispiele finden Sie im Notebook **NOI-intro**.

## Notebooks und Labs

Filename	Beschreibung
<b>Notebooks</b>	
NOI-intro	Auswirkung der Quantisierung im Frequenzbereich
NOI-Wahrscheinlichkeitsdichte	Quantisierung und Wahrscheinlichkeitsdichtefunktion
NOI_DFT	Skalierung von Signal (Linienspektrum) und Quantisierungsrauschen (Breitbandspektrum) im Frequenzbereich, SQNR, ENOB etc.
NOI-DFT-wideband	Code zu Aufgabe 7.4: DFT von Schmal- und Breitbandsignalen
NOI-DFT-wideband-ML	Musterlösung zu Aufgabe 7.4
NOI_Random_processes	Verschiedene Arten von Zufallsprozessen im Zeit- und Frequenzbereich

Tab. 7.1: Jupyter Notebooks zu Kap. 7

## Experimente zu Kap. 7

Spielen Sie mit dem Skript herum, versuchen Sie den Code mit Hilfe der obigen Hinweise nachzuvollziehen und versuchen Sie folgende Fragen zu beantworten (keine Musterlösung, wird in der Vorlesung besprochen):

<sup>5</sup>Achtung: Mit Ausnahme des DC-Werts!!!

<sup>6</sup>Im Gegensatz zu Matlab bewirkt  $\mathbf{A}\mathbf{Q} * \mathbf{A}\mathbf{Q}$  in Python eine *elementweise* Quadrierung des Vektors.

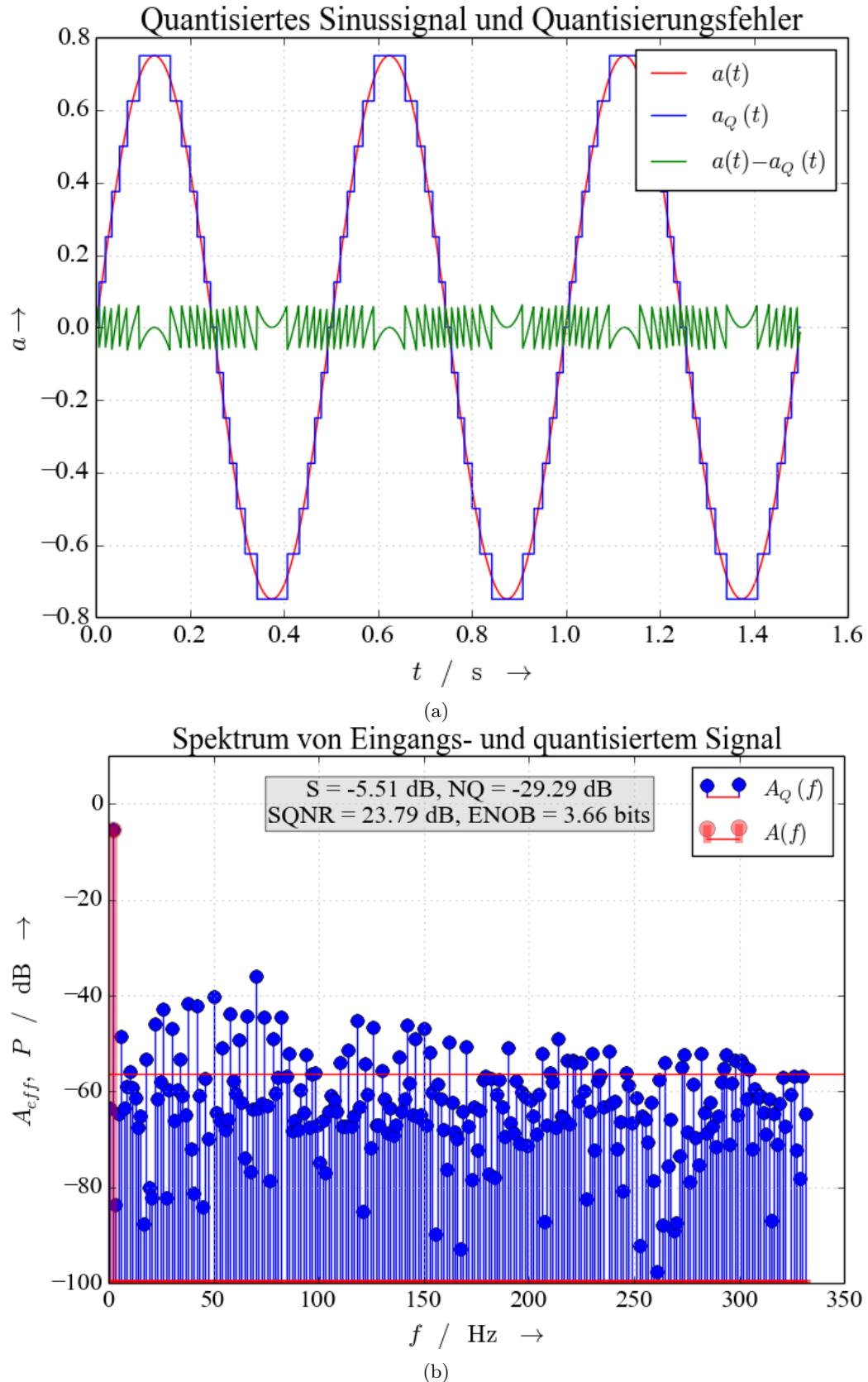


Abb. 7.3: Auswirkung der Signalquantisierung (a) im Zeit- und (b) im Frequenzbereich (Notebook **NOI-intro**)

- Können Sie aus Abb. 7.3b erkennen, welche **Art der Requantisierung und ob Sättigungslogik** verwendet wurde?
- Welches **maximale SQNR** erwarten Sie bei den Quantisierungsparametern im Notebook **NOI-intro**? Warum ergibt die Simulation einen schlechteren Wert? Welchen maximalen Wert können Sie erreichen?
- Legen Sie ein Sinussignal mit der halben maximalen Amplitude an den Quantisierer. Welche Signalleistung erwarten Sie, welches *SQNR* und welche *ENOB*?
- Was passiert, wenn Sie den Quantisierer übersteuern? Vergleichen Sie die Optionen '**sat**' und '**wrap**'.
- Welchen Einfluss auf das Quantisierungsrauschen haben die Quantisierungsmethoden, also z.B. '**floor**', '**round**' und '**fix**'? Schauen Sie sich dabei auch den DC-Wert des Spektrums an.
- Erklären Sie, welchen Einfluss die Anzahl der dargestellten Signalperioden **N\_per** im Code auf das dargestellte Spektrum hat.

## 7.1 Quantisierungsrauschen bei Analog-Digital-Wandlung → M7.1

Ein ideal sinusförmiges Signal mit  $f_1 < f_S/2$  wird mit einem 10 bit Analog-Digital Wandler mit einem Aussteuerbereich von  $FSR = 2 \text{ V}$  (Full Scale Range) quantisiert. Eine anschließende Spektralanalyse der digitalen Daten ergibt einen *SNR* von 53 dB.

- Wie groß ist das **maximal erreichbare SNR** für sinusförmige Signale mit einem perfekten 10 bit ADC? Welcher Effective Number of Bits (*ENOB*) entspricht das *SNR* von 53 dB?
- Welche Gründe könnte es geben für die **Abweichung vom maximal erreichbaren SQNR**? Nehmen Sie an, dass das Rauschen durch die Quantisierung dominiert wird und schätzen Sie die Amplitude  $A$  des Sinussignals ab aus der Abweichung zwischen maximalem und tatsächlichen S(Q)NR.
- Wie groß sind die **Rauschleistung  $N_Q$**  und die **Rauschleistungsdichte  $N'_Q$**  des voll ausgesteuerten Wandlers ( $A = 1 \text{ V}$ ) wenn mit  $f_{S1} = 10 \text{ kHz}$  bzw.  $f_{S2} = 1 \text{ MHz}$  abgetastet wird?
- Eine 2048-Punkt FFT mit realem Wandler und sinusförmigem Eingangssignal ergibt das Spektrum in Abb. 7.4. Wie groß sind **SNR und Amplitude des Signals**? Wie viele effektive Bits (**ENOB, effective number of bits**) hat der Wandler?

## 7.2 SQNR eines quantisierten Signals → M7.2

- Wie groß ist das **Signal-to-Quantization Noise Ratio (SQNR)** eines sinusförmigen **Eingangssignals**, das mit 16 bit quantisiert wurde und das den Wertebereich voll ausnutzt?
- Wie groß ist das *SQNR*, wenn das **Signal den Aussteuerbereich nur zu einem Zehntel ausnutzt**?

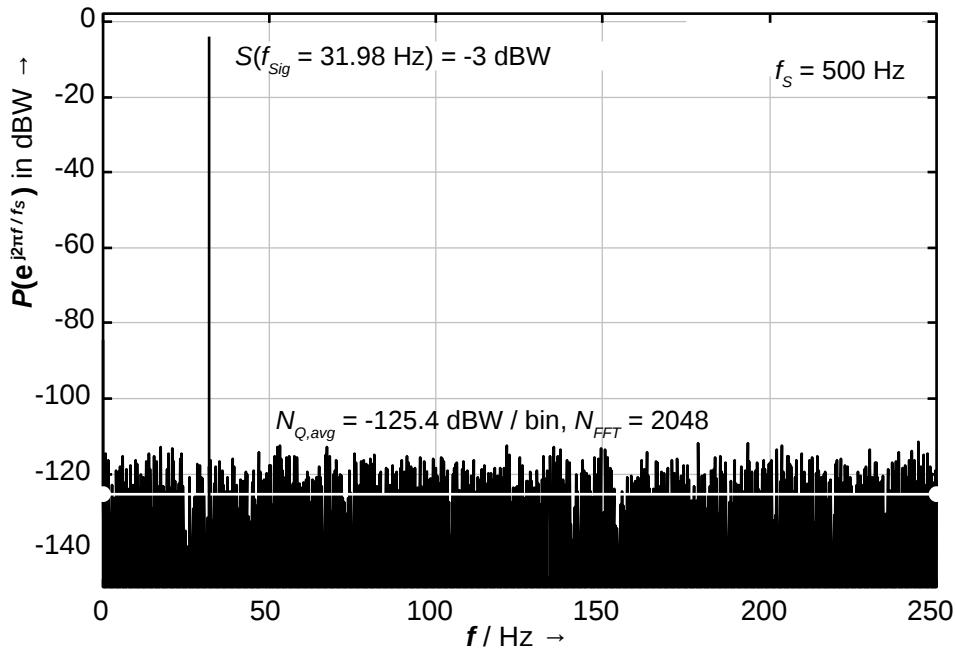


Abb. 7.4: Spektrale Leistungsdichte am Ausgang eines ADCs (Aufgabe 7.1d)

### 7.3 Quantisierungsrauschen im FIR-Filter → M7.3

Es soll das Quantisierungsrauschen am Ausgang des Filters aus Aufgabe 6.4 untersucht werden. Die (Re-)Quantisierung erfolgt an allen Stellen durch Runden, der Ein- und Ausgangsspannungsbereich sei  $\pm 1 \text{ V}$ . Das Signal habe zunächst eine niedrige Frequenz  $F_1$ , bei der gilt  $H(F_1) = 1$  und eine Amplitude  $A_{in} \approx A_{out} \approx 1$  (es tritt gerade eben noch keine Übersteuerung auf). Vernachlässigen Sie bei der Rechnung alle Rauschquellen außer den jeweils angegebenen. **Bestimmen Sie Rauschleistung  $N_{Q,Out}$ , SNR und ENOB am Ausgang für**

- a)  $Q_{DAC}(0.15)$ : Requantisierung am Ausgang auf 16 bit vor dem DAC
- b)  $Q_{ADC}(0.14)$ : Quantisierung am Eingang durch 15 bit ADC
- c)  $Q_{ADC}(0.14)$  und  $Q_{DAC}(0.15)$ : Quantisierung am Eingang durch 15 bit ADC und Requantisierung vor dem 16 bit DAC
- d)  $Q_{Mul}(0.15)$ : Requantisierung nach allen Multiplizierern auf 16 bit. Wie viele Multiplizierer werden bei diesem Filter wirklich verwendet?

Das Filter wird jetzt wie bei Unterpunkt c) mit einem 15 bit ADC  $Q_{ADC}(0.14)$  am Eingang und einem 16 bit DAC  $Q_{DAC}(0.15)$  am Ausgang betrieben; mit dieser Konfiguration wurde eine **ENOB = 15,1 bits** erzielt.

- e) Auf wieviele bits  $Q_{Mul}$  darf nach den Multiplikationen minimal gerundet werden, damit das hierdurch hervorgerufene Quantisierungsrauschen kleiner ist als von ADC und DAC? Welches **SNR** und welche **ENOB** ergibt sich in diesem Fall durch die Wirkung aller (Re-)Quantisierer am Ausgang?
- f) Die Requantisierung nach der Multiplikation soll die **ENOB** um maximal 0,1 bit verschlechtern. Wieviel Bits sind hierfür notwendig?
- g) Welches **SNR** erhält man für ein sinusförmiges Eingangssignal mit maximaler Amplitude und der Signalfrequenz  $F_1 = 1/4$ ?

## 7.4 DFT von Breitbandsignalen → M7.4

In dieser Aufgabe soll ein durch breitbandiges Rauschen gestörtes Signal mit dem Notebook **NOI-DFT-wideband** analysiert werden.

- a) Welche **Nutzleistungen**  $S_a$  und  $S_b$  haben die Frequenzkomponenten des Nutzsignals?
- b) Wie groß ist die **Gesamtleistung**  $S$ ?
- c) Welche **Rauschleistung**  $N$  hat das Rauschsignal im Notebook **NOI-DFT-wideband**? Wie groß ist das **Signal-to-Noise Ratio**  $SNR$ ? Finden Sie ggf. heraus, wie der gaußverteilte Zufallsprozess `randn(N)` definiert ist.
- d) Wie groß ist die **Rauschleistungsdichte**  $N'(f)$ ?
- e) Welche **mittlere Rauschleistung** zeigt die DFT in der Simulation an? Überlegen Sie dazu, welcher Teil der Gesamtrauschleistung in jedes DFT-Bin fällt.
- f) **Verbessern Sie den Code** so, dass die Spektren korrekt skaliert dargestellt werden.

## 8 SMP: Abtastung und Downampling

### Lernziele

Dieses Kapitel beschäftigt sich mit der Abtastung (Sampling) zeitkontinuierlicher Signale und dem Downampling (Reduktion der Abtastrate, d.h. erneute Abtastung) zeitdiskreter Signale. Da man zeitkontinuierliche Signal betrachten kann wie ein Signal, das unendlich schnell abgetastet wurde, werden beide Vorgänge mit den gleichen Methoden behandelt.

**Aliasing:** Sie kennen den Mechanismus, durch den Spektralkomponenten oberhalb der Nyquistfrequenz (= halbe Abtastfrequenz) auf das neue Basisband oder in andere Nyquistzonen abgebildet werden und können deren Frequenzen vorhersagen.

**Anti-Aliasfilter:** Sie können ein Anti-Aliasfilter so auslegen, dass Aliasing vermieden wird.

**Oversampling:** Sie kennen die Vorteile von „zu schneller“ Abtastung eines zeitkontinuierlichen Signals.

**Dezimation:** Sie können die Abtastrate eines Systems reduzieren, kennen die Vorteile und wissen was Sie dabei beachten müssen.

**Multiratenfilter:** Sie können Downampling für den Entwurf effizienter Filter nutzen und wissen wie man die Teilkomponenten auslegen muss.

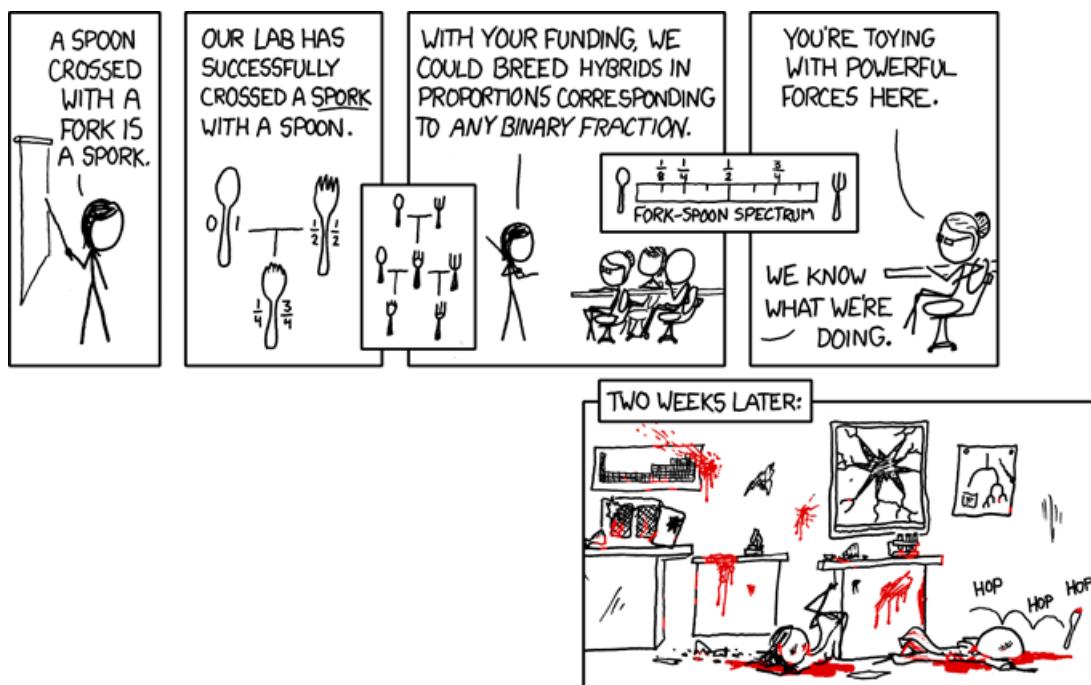


Abb. 8.1: Abtasten ist gefährlich [<http://xkcd.com/419/>]

## Zusätzliche Literatur

Siehe Literaturempfehlungen zu Kap. 10.

### Mehr!

Nicht besprochen werden in dieser Vorlesung die verwandten Themen „Subsampling“ und „Compressed Sensing“. Ersteres nutzt gezieltes Aliasing, um ein Signal aus einer höheren in eine niedrigere Nyquistzone oder ins Basisband zu bringen und dabei die Samplingrate zu reduzieren. Fehlerfrei funktioniert das nur für Bandpasssignale, die auf eine Nyquistzone beschränkt sind.

„Compressed Sensing“ (auch *compressive sensing*, *compressive sampling* oder *sparse sampling*) ist ein Verfahren zur Erfassung und Rekonstruktion von dünnbesetzten (englisch sparse) Signalen oder Informationsquellen. Solche lassen sich aufgrund ihrer Redundanz ohne wesentlichen Informationsverlust komprimieren. Dies wird bei der Abtastung der Signale effizient genutzt, um die Abtastrate im Vergleich zu konventionellen Verfahren deutlich zu reduzieren. [Wikipedia.de, [https://de.wikipedia.org/wiki/Compressed\\_Sensing](https://de.wikipedia.org/wiki/Compressed_Sensing)]

<http://ieeexplore.ieee.org/abstract/document/7080526/>

## Spezielle Python-Befehle

Für dieses Kapitel benötigt man Befehle, um jeden  $L$ -ten Wert eines Arrays zu entnehmen (*Downsampling*), mit Hilfe der Slicing Syntax von Python und Matlab geht das sehr kompakt und elegant. Optional kann die Startphase (also das erste Sample)  $n_0$  spezifiziert werden.

Bei diesem Prozess kann Aliasing auftreten, daher wird üblicherweise vorher tiefpassgefiltert, den gesamten Prozess nennt man dann *Dezimation*.

Funktion	Python	Matlab	Beschreibung
<b>Downsample</b>	<code>y = x[::L]</code>	<code>y = x(1:L:N)</code>	Nimm nur jedes $L$ -te Sample von $x$ , beginnend mit Sample $n_0$
	<code>y = x[n0::L]</code>	<code>y = x(n0:L:N)</code>	
<b>Dezimation</b>	<code>y = sig.decimate(x, L, n=None, ftype='iir')</code>	<code>y = decimate(x, L [,N,'iir'])</code>	Dezimiere $x$ um den Faktor $L$

Tab. 8.1: Spezielle Befehle zu Kap. 8

### Anmerkungen zu Tab. 8.1:

**downsample** Achtung: Unterschiedliche Reihenfolge der Slicing-Parameter bei Python und Matlab; das erste Element eines Arrays hat in Python den Index 0, in Matlab den Index 1!

**decimate** verbindet Anti-Alias-Filterung und Downsampling miteinander. Die Angabe der Filterordnung  $N$  und von 'iir' bzw. `ftype = 'iir'` ist optional. Defaultmäßig wird ein Chebychev-Filter Typ I (IIR) achter Ordnung verwendet. Hier wird die Sequenz vorwärts und rückwärts gefiltert (siehe `filtfilt` bzw. `sigfiltfilt()`) und so eine lineare Phase (und die doppelte Ordnung) erzielt. Da die komplette Sequenz hierfür im Speicher stehen muss, ist diese Methode für lange Sequenzen schlecht geeignet.

Wählt man mit 'fir' bzw. `ftype = 'fir'` ein FIR-Filter, wird defaultmäßig ein Filter 30. Ordnung mit Hamming-Fenster verwendet. Dieses Filter ist linearphasig und wird nur in einer Richtung durchlaufen, die

Sequenz muss sich daher nicht komplett im Speicher befinden und kann beliebig lang sein.

Achtung: Die Filter haben Ripple im Durchlassbereich, daher können bei einzelnen Frequenzen Überhöhungen auftreten!

## Notebooks und Labs

Filename	Beschreibung
<b>Notebooks</b>	
SMP_Audio-decimation	Code-Beispiel zur Dezimation (mit / ohne Filterung) von Audio-Signalen
SMP_sampled_sine	Abtastung sinusförmiger Signale mit und ohne Aliasing, animiertes Beispiel

Tab. 8.2: Jupyter Notebooks zu Kap. 8

## 8.1 Spektren abgetasteter Signale → M8.1

Auf der linken Seite von Abb. 8.2 sind die Betrags- und Phasenpektrum der drei zeitkontinuierlichen Signale

$$s_i(t) = \cos(2\pi f_i t + \varphi_i) \text{ mit } f_1 = 500 \text{ Hz}, f_2 = 1000 \text{ Hz}, f_3 = 2500 \text{ Hz}, \varphi_1 = -45^\circ, \varphi_2 = \varphi_3 = 45^\circ$$

aufgetragen. Diese Signale werden jetzt mit  **$f_S = 1500 \text{ Hz}$**  abgetastet.

Skizzieren Sie auf der rechten Seite von Abb. 8.2 die Spektren der resultierenden zeitdiskreten Signale. Beschränken Sie sich dabei auf die Komponenten, die durch  $\pm 2f_S$ ,  $\pm f_S$  und  $f = 0$  entstehen und markieren Sie jeweiligen Spektrallinien farbig!

## 8.2 Analog-Digital-Wandlung mit Oversampling → M8.2

Ein sinusförmiges Signal wird mit 8 bit Auflösung und der Abtastrate  $f_S = 1 \text{ MHz}$  abgetastet, es werde der volle Aussteuerbereich ausgenutzt.

- a) Wie groß ist das **maximale SQNR** bei Nyquist-Wandlung (maximale Nutzbandbreite  $f_N$ )?
- b) Wie groß ist das **maximale SQNR** bei Oversampling mit OSR = 10 und bei OSR = 50 und idealer Tiefpassfilterung? Welcher Effective Number of Bits (ENOB) entsprechen die SQNR-Werte?
- c) Wie groß ist das **maximale SQNR** bei Sigma-Delta-Wandlung (ein Bit) erster oder zweiter Ordnung, OSR = 10 und OSR = 50 und idealer Tiefpassfilterung?

## 8.3 Analog-Digital-Wandlung mit Oversampling und nachfolgender Dezimation → M8.3

In einem System A soll ein analoges Signal mit  $B = 12 \text{ kHz}$  Bandbreite mit einem idealen ADC mit  $f_{S1} = 100 \text{ kHz}$  Abtastrate gewandelt werden.

Alternativ soll das System B verwendet werden mit einem idealen ADC mit  $f_{S2} = 1 \text{ MHz}$  Abtastrate und nachfolgender Dezimation um den Faktor  $R = 10$ . Dabei werden immer 9 aufeinander folgende

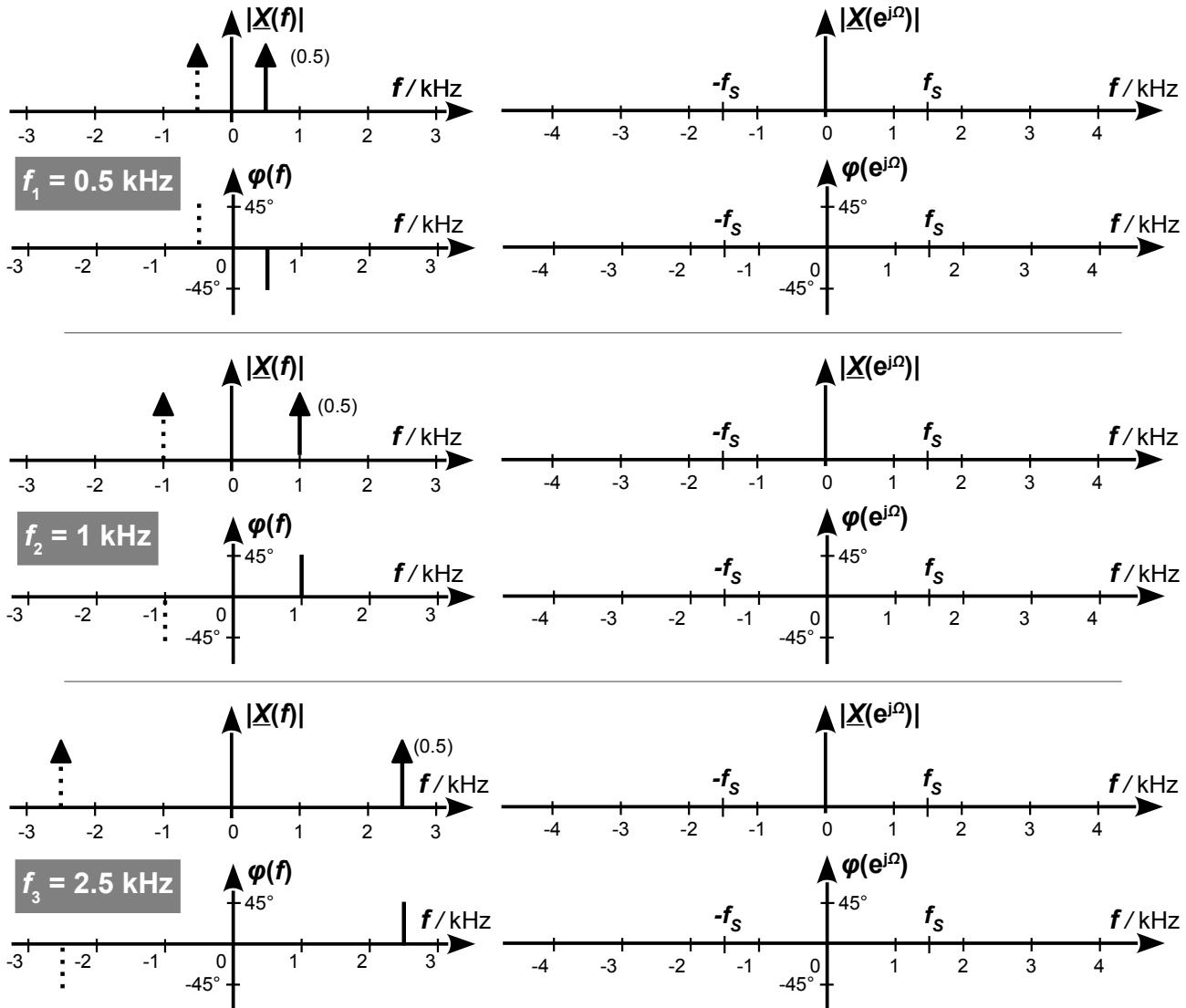


Abb. 8.2: Amplituden- und Phasenspektren von zeitkontinuierlichen und abgetasteten Signalen zur Aufgabe 8.1.

Samples verworfen. Zwischen ADC und Dezimation wird kein digitales Anti-Antialias / Dezimationsfilter verwendet.

- Sind die Ausgangssignale von System A und B identisch? Wird das Spektrum des Signals in System B auf diese Art verfälscht?
- Welcher maximale ganzzahlige Dezimationsfaktor  $R_{max}$  kann in System B gewählt werden, ohne das Spektrum zu verfälschen?
- Warum ist im Allgemeinen ein digitales Filter vor der Dezimationsstufe erforderlich?
- Welche Vorteile hat die Kombination von Überabtastung und nachfolgender Dezimation?

## 8.4 Downsampling um den Faktor 2 nach Averaging → M8.4

Die Abtastrate eines Signals mit Bandbreite  $B = 80 \text{ kHz}$  soll von  $f_{S1} = 384 \text{ kHz}$  auf  $f_{S2} = 192 \text{ kHz}$  reduziert werden,

- a) Wie muss das Anti-Aliasing Filter für dieses **bandbegrenzte Signal** ausgelegt werden?

Jetzt sei dem Signal weißes Rauschen überlagert, die Nutzbandbreite ist immer noch  $B = 80 \text{ kHz}$ .

- b) **Filterung mit überlagertem Rauschen:** Welche Anforderungen muss das Filter erfüllen, damit kein Aliasing des Rauschens ins Nutzband auftritt?

Das Downsampling soll jetzt mit minimalem Rechenaufwand durchgeführt werden, indem der Mittelwert aus jeweils zwei aufeinanderfolgenden Samples bestimmt und an Stelle der beiden Samples verwendet wird.

- c) Welchem **FIR - Filter** entspricht die Mittelung von jeweils zwei aufeinander folgenden Samples?
- d) Welchen **Betragsgang**  $|H(e^{j\Omega})|$  hat das Filter? Welche maximale Dämpfung hat es im Durchlassband, welche minimale Dämpfung im Sperrband?

## 8.5 Effizientes Averaging und Downsampling um den Faktor N → M8.5

In dem System aus Aufgabe 8.4 soll die Abtastrate ausgehend von  $f_{S1} = 384 \text{ kHz}$  um den Faktor  $R = L$  reduziert werden nach Mittelung von  $L$  aufeinander folgenden Samples (Moving Average Filter). Die Signalbandbreite betrage  $B = 10 \text{ kHz}$ .

- a) Mit welcher **maximalen Dezimationsrate**  $R_{max}$  kann das Signal fehlerfrei dezimiert werden?
- b) Welchen **Betragsgang** hat das Mittelwertfilter für  $L = R_{max}$ ? Nutzen Sie die Ergebnisse von Aufgabe 2.11.
- c) Wie könnte eine **effiziente Kombination aus Mittelwertberechnung und Downsampling** aussehen? Gehen Sie dafür von der effizienten Implementierung von Moving Average Filtern in Aufgabe M1.1 aus und versuchen Sie Samples, die beim Downsampling verworfen werden, erst gar nicht zu berechnen. [ML unvollständig]

## 8.6 Zweistufige Dezimation → M8.6

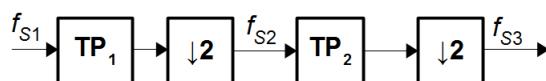


Abb. 8.3: Zweistufige Dezimation zu Aufgabe 8.6

Ein Analogsignal wird mit vierfacher Überabtastung digitalisiert. Die Abtastfrequenz soll in zwei Stufen um den Faktor 4 reduziert werden (Dezimation). Die Eckfrequenz des Durchlassbands des gesamten Systems in Abb. 8.3 soll ein Drittel der Abtastfrequenz am Ausgang  $f_{S3}$  betragen, dies ist auch die Nutzbandbreite  $f_N = f_{S3}/3 = f_{S1}/12$  des Eingangssignals. Das Spektrum des Eingangssignals kann als weiß betrachtet werden, daher ist bei der Dezimation auf ausreichende Anti-Alias Filterung zu achten.

- a) Beide Dezimationsfilter (= Anti-Aliasfilter) sollen zunächst als (dezimierende) Halbbandfilter realisiert werden. Bei beiden Dezimationsstufen ist **Aliasing ins Basisband** (aber nicht ins Nutzband!) gestattet.
- b) Bei der zweiten Dezimation ist jetzt **kein Aliasing ins Basisband** mehr erlaubt, in der ersten Dezimationsstufe schon (aber nicht ins Nutzband). Das erste Dezimationsfilter soll wieder als Halbbandfilter, das zweite als allgemeines FIR-Filter realisiert werden. Gehen Sie hier am besten vom Ausgang des Filters aus und überlegen Sie, welche Bedingung die Teilfilter jeweils erfüllen müssen, damit keine Aliasingkomponenten zum Ausgang gelangen.

Berechnen Sie für beide Fälle die Eckfrequenzen von Durchlass- und Sperrband der Teilfilter!

**Hinweis:**

Halbbandfilter sind Filter, deren Frequenzgang punktsymmetrisch zu  $|H(f = f_S/4)| = 0,5 \hat{=} -6 \text{ dB}$  ist. Die Eckfrequenzen von Durchlass- und Sperrband können daher nicht unabhängig voneinander gewählt werden,  $f_{DB} = f_S/2 - f_{SB}$ . Dezimierende Halbbandfilter verbinden Filterung und Downsampling um den Faktor 2 und nutzen bei der Implementierung aus, dass jeder zweite Koeffizient (bis auf den mittleren) gleich Null ist, sie werden daher gerne in Dezimierungsfiltern genutzt, die mit hoher Taktfrequenz arbeiten müssen.

## 8.7 Multiraten-Tiefpassfilter → M8.7

In Abb. 8.4 ist ein zweistufiges Multiraten-Tiefpassfilter dargestellt mit den idealisierten<sup>1</sup> Frequenzgängen der Tiefpässe  $TP_1$  und  $TP_2$ . Die Abtastfrequenz am Eingang beträgt  $f_{S1} = 48 \text{ kHz}$ . Das Gesamtfilter soll Eckfrequenzen bei  $f_{DB} = 2 \text{ kHz}$  und  $f_{SB} = 4 \text{ kHz}$  haben, die Sperrdämpfung soll  $A_{SB} = 80 \text{ dB}$  betragen. Untersuchen Sie, ob das Filter die Spezifikationen erfüllt.

Beachten Sie die unterschiedlichen Abtastfrequenzen und die endliche Sperrdämpfung der Tiefpässe!

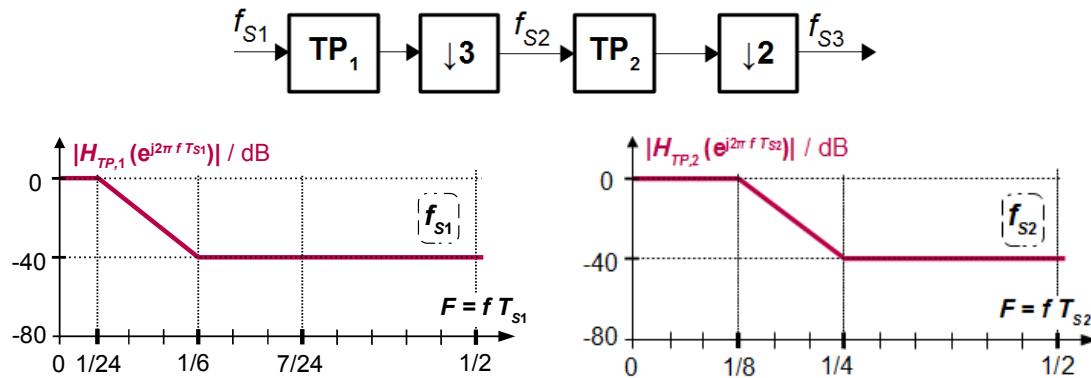


Abb. 8.4: Multiraten-Tiefpassfilter und Betragsgang der Teilfilter zu Aufgabe 8.7

- a) Skizzieren Sie den **Amplitudengang der gesamten Filterkaskade** zunächst im Bereich  $f = 0 \dots f_{S3}/2$ . Bestimmen Sie die Dämpfung der gesamten Kaskade bei der Eckfrequenz von Durchlass- und Sperrband und tragen Sie  $A_{DB}$  und  $A_{SB}$  in Tab. 8.3 ein.

Am Eingang werden jetzt **sinusförmige Testsignale** mit den Frequenzen  $f_1$ ,  $f_2$  und  $f_3$  angelegt (siehe Tab. 8.3).

- b) Berechnen Sie die **Aliasfrequenzen**  $f_{out,i}$ , bei denen die Testsignale am Ausgang der Filterkaskade erscheinen ...
- c) ... sowie deren **Gesamtdämpfung**  $A_{al,i}$  und tragen Sie die Daten in Tab. 8.3 ein.
- d) **Verbessern Sie die Spezifikationen** von  $TP_1$  und / oder  $TP_2$  so, dass die die gestellten Anforderungen für das Gesamtfilter erfüllt werden.
- e) **Entwerfen Sie ein Filter** mit Hilfe eines Filterentwurfstools, das die Spezifikationen für  $TP_1$  aus dem letzten Aufgabenpunkt erfüllt. Welche minimale Filterordnung benötigt ein Equiripple-Entwurf, welche Ordnung ein gefensterter Entwurf mit Hann-Fenster?

<sup>1</sup>d.h. Durchlass-/Übergangs-/Sperrbereich sind nur schematisch dargestellt

$f_{in}$	$f_{DB}$	$f_{SB}$	$f_1$	$f_2$	$f_3$
	2 kHz	4 kHz	8 kHz	14 kHz	16 kHz
$f_{in}/f_{S1}$	<b>1/24</b>				
$f_{in}/f_{S2}$					
$f_{in}/f_{S3}$					
$f_{out}$ bei $f_{S2}$	2 kHz	4 kHz			
$f_{out}$ bei $f_{S3}$	2 kHz	4 kHz			
	$A_{DB}$	$A_{SB}$	$A_{al,i}$		
Soll (dB)	0,1	80	80	80	80
Ist (dB)					

Tab. 8.3: Eigenschaften der Filterkaskade aus Aufgabe 8.7



## 9 INP: Upsampling, Interpolation und Digital-Analog Wandlung

### Lernziele

Dieses Kapitel beschäftigt sich mit der Erhöhung der Abtastrate bei zeitdiskreten Systemen und mit der D/A Wandlung. Beide Prozesse kann man mit den gleichen Methoden behandeln, da man ein analoges Signal als unendlich schnell abgetastet betrachten kann ( $T_S \rightarrow 0$ ):

**Upsampling:** Sie skizzieren Systeme zur Erhöhung der Abtastrate durch **Nullenstopfen** und durch Wiederholen des letzten Wertes (Zero-Order Hold). Sie skizzieren Zeitsignale und Spektren und begründen, warum man fehlende Werte meistens nicht durch Wiederholung des letzten Wertes ergänzt.

**Images:** Sie kennen den Unterschied zwischen Wiederholspektren von abgetasteten Signalen und Images, die bei Erhöhung der Abtastrate durch Nullenstopfen von  $f_S$  auf  $I f_S$  entstehen.

**Interpolation:** Sie skizzieren ein System zur Erhöhung der Abtastrate *ohne* dass Images entstehen. Sie wissen was **ideale Interpolation** ist und legen ein Interpolationsfilter entsprechend aus.

**Rekonstruktion:** Sie skizzieren ein System zur Rekonstruktion der analogen Signalform aus dem abgetasteten Signal und begründen warum der Digital-Analog-Wandler (DAC) immer als Zero-Order-Hold realisiert ist.

**Oversampling:** Sie skizzieren ein System mit Oversampling DAC, nennen die Vor- und Nachteile dieser Architektur und dimensionieren die Filter.

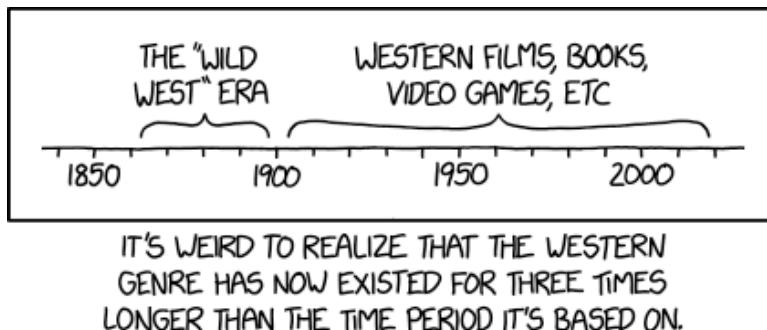


Abb. 9.1: I = 4: Basisband und Images [<https://xkcd.com/2152>]

## Zusätzliche Literatur

Siehe Literaturempfehlungen zu Kap. 10

## Theorie

### Zero-Stuffing

#### Zero-Order Hold DAC

Ein hypothetischer idealer DAC würde analoge Diracstöße produzieren mit einem Gewicht (= Fläche)  $s[n]\delta(t - nT_S)$ , proportional zum zu wandelnden Digitalwert  $s[n]$ . Das Ausgangsspektrum eines solchen idealen DACs sähe genauso aus wie das Wiederholspektrum des digitalen Signals, wiederholt bis zu unendlich hohen Frequenzen. Da Diracstöße eine zeitliche Ausdehnung von 0 und eine unendlich große Amplitude aufweisen, ist diese Art der Wandlung aber technisch nicht realisierbar.

Stattdessen wird in der Praxis eine endliche analoge Spannung generiert, die proportional zum Digitalwert ist, und für die Dauer einer Abtastperiode  $T_S$  gehalten (Zero-Order Hold, ZOH). Diese Rechteckfläche kann z.B. den gleichen Wert haben wie die Fläche des Diracstößes, aber das Spektrum ist unterschiedlich: Die Zero-Order Hold (ZOH) Charakteristik des DACs dämpft das Wiederholspektrum mit einer si-Funktion, deren erste Nullstelle bei  $f_{0,min} = f_S$  liegt. Die maximale Dämpfung  $a_N$  im Nutzband hat man daher an der oberen Kante des Nutzbands bei  $f_N$ .

## Spezielle Python-Befehle

Für dieses Kapitel benötigt man vor allem einen Python- bzw. Matlab-Befehl, um zwischen jeden Wert eines Arrays  $I$  Nullen zu „stopfen“ oder jeden Wert  $I$  mal zu wiederholen (Zero-Order Hold). Das geht mit Hilfe der Slicing-Schreibweise (siehe Python Kurzanleitung) sehr kompakt und elegant:

Funktion	Python	Matlab	Beschreibung
si-Funktion	<code>y = np.sinc(x)</code>	<code>y = sinc(x)</code>	$\sin(x)/x$ Funktion
di-Funktion	<code>y = scipy.special.diric(x,N)</code>	<code>y = diric(x,N)</code>	Dirichlet-Kernel
Upsampling (Zero-Stuffing)	(siehe Listing 9.1)	<code>y = upsample(x,I [,n0])</code>	Füge $I-1$ Nullen zwischen Samples ein
Upsampling (ZOH)	<code>y = np.repeat(x, I)</code>	<code>y = kron(x, ones(1,I))</code>	Wiederhole jedes Sample $I-1$ mal
Interpolation	siehe <code>scipy.interpolate</code>	<code>y = interp(x,r[,n,a])</code>	

Tab. 9.1: Spezielle Befehle zu Kap. 9

### Anmerkungen zu Tab. 9.1:

`upsample` kann auch in Python effizient mit `np.insert` durchgeführt werden oder durch „Missbrauch“ von `sig.upfirdn()` (siehe nächstes Kapitel).

`kron` ist das Kronecker-Produkt; in Python könnte man genauso schreiben `y = np.kron(x, np.ones(I))`, das ist aber deutlich langsamer als `repeat` oder `tile`.

---

```

1 import numpy as np
2 a_in = np.arange(1,4) # initial array
3 I = 3; n_0 = 0 # Upsampling factor & start phase
4 # create array of zeros with size a_in * I:
5 a_up = np.zeros(I * a.size, dtype=a.dtype)
6 # fill in elements of a_in every Ith position:
7 a_up[n_0::I] = a_in
8 # Alternativ:
9 a_up = np.insert(a_in, np.arange(len(a_in)), np.zeros(I-1))
10 # Upsampling with ZOH:
11 a_ZOH = np.repeat(a_in, I)
12
13 >>> a_in
14 array([1, 2, 3])
15
16 >>> a_up
17 array([1, 0, 0, 2, 0, 0, 3, 0, 0])
18
19 >>> a_ZOH
20 array([1, 1, 1, 2, 2, 2, 3, 3, 3])

```

---

Lst. 9.1: Upsampling mit Python

---

```

%
a_in = [1:3];
I = 3; n_0 = 1
%
a_up = zeros(1, I * length(a_in));
%
a_up(n_0:I:length(a_up)) = a_in;
% alternativ:
a_up = upsample(a_in, I, n_0)
% Zero-Order Hold:
% allocate memory for speed-up:
a_ZOH = zeros(1, I*length(a_in));
a_ZOH = kron(a_in, ones(1,I));
%
%
%
%
%
%
%
%
%
%
%
```

---

Lst. 9.2: Upsampling mit Matlab

Soll nicht nur die Abtastrate erhöht werden (Upsampling), sondern auch gleich interpoliert (= Zwischenwerte berechnet bzw. Images entfernt) werden, kann man `a_ip = interp(a_in, I)` (Matlab) verwenden. In Python gibt es hierfür `interp1d()` aus `textscipy.interpolate` (siehe <http://docs.scipy.org/doc/scipy/reference/tutorial/interpolate.html>). `np.interp()` kann nur linear interpolieren und ist daher schlecht geeignet.

In Matlab entwirft man mit dem Befehl `b = intfilt(I, p, alpha)` ein linearphasiges Tiefpassfilter für die ideale bandbegrenzte Interpolation einer Sequenz mit Zerostuffing. Alternativ entwirft `intfilt(I, N, 'Lagrange')` ein Filter, das Lagrange-Interpolation der Ordnung `N` durchführt.

## Notebooks und Labs

---

Filename	Beschreibung
<b>Notebooks</b>	
<b>INP_Upsampling</b>	Erhöhung der Abtastrate durch Nullenstopfen und die Auswirkungen im Frequenzbereich. Mit Audiobeispielen?
<b>INP_ZOH</b>	Erhöhung der Abtastrate durch Wiederholen des letzten Wertes (Zero-Order Hold) im Zeit- und Frequenzbereich
	Darstellung und Simulation der si-Interpolation
<b>INP_si-Interpolation</b>	

---

Tab. 9.2: Jupyter Notebooks zu Kap. 9

## 9.1 Wiederholspektren und Images → M9.1

Ein digitales Audiosignal  $s[n]$  mit Abtastrate  $f_{S1} = 48$  kHz und Nutzbandbreite  $f_N = 12$  kHz hat das Basisbandspektrum  $S(e^{j2\pi f T_{S1}})$  in Abb. 9.2a).

- a) Ergänzen Sie die **Wiederholspektren** in Abb. 9.2a und kennzeichnen Sie den Frequenzbereich des **Basisbands**.
- b) Die Abtastrate von  $s[n]$  wird jetzt durch Nullenstopfen um den Faktor  $I = 3$  erhöht. Ergänzen Sie **Wiederholspektren und Images** in Abb. 9.2b, kennzeichnen und bezeichnen Sie beide durch eine eindeutige Schraffur. Tragen Sie die Grenzen des Basisbands ein.

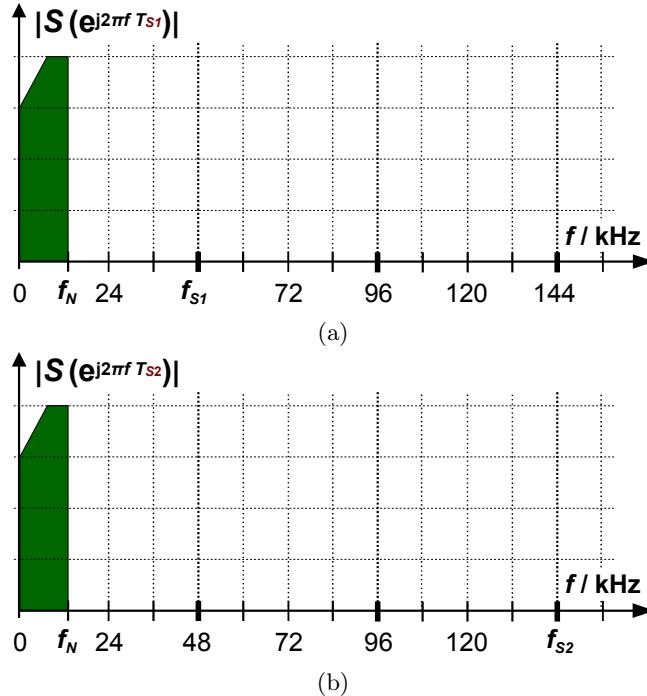


Abb. 9.2: Spektrum von  $s[n]$ , (a) vor und (b) nach Abtastratenerhöhung um  $I = 3$

## 9.2 Verschiedene Arten der Digital-Analog-Wandlung → M9.2

Ein digitales Audiosignal  $s[n]$  mit Abtastrate  $f_{S1} = 48$  kHz und Nutzbandbreite  $f_N = 12$  kHz (Basisbandspektrum siehe Abb. 9.2a) soll auf verschiedene Arten analog rekonstruiert werden. Die eigentliche D/A Wandlung erfolgt dabei immer mit einfachem analogen Halteglied (Zero-Order Hold):

- a) **Direkte D/A-Wandlung** des digitalen Audiosignals bei der Abtastfrequenz  $f_{S1}$ .
- b) **Überabtastung um den Faktor  $I = 3$  ohne digitale Interpolation**  
Das digitale Audiosignal wird zunächst um den Faktor 3 überabgetastet, zwischen den Signalwerten werden dabei Nullen eingefügt (Nullenstopfen). Die hierdurch reduzierte Leistung im Basisband wird durch Multiplikation um den Faktor 3 ausgeglichen, es findet aber keine digitale Interpolation statt. Die D/A Wandlung findet bei der Abtastfrequenz  $f_{S2} = 3f_{S1}$  statt.
- c) **Überabtastung um den Faktor  $I = 3$  mit idealer digitaler Interpolation**  
Nach Überabtastung um den Faktor  $I = 3$  und Nullenstopfen ist jetzt ein ideales Interpolationsfilter mit Amplitudenkorrektur geschaltet. Die D/A Wandlung findet bei der Abtastfrequenz  $f_{S2} = 3f_{S1}$  statt. Was heißt ideale Interpolation? Bestimmen Sie die Eckfrequenzen des Durchlass- und des Sperrbandes für das Interpolationsfilter (möglichst breiter Übergangsbereich).

d) **Überabtastung um den Faktor  $I = 3$  mit digitalem Zero-Order Hold**

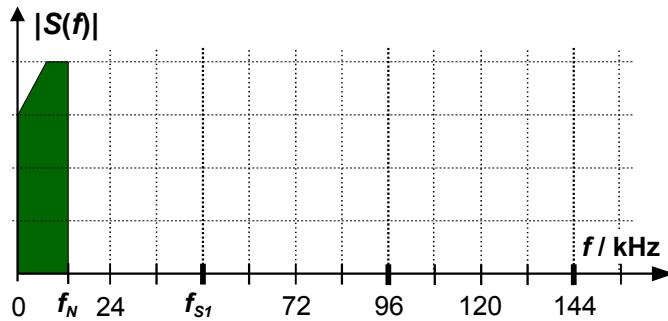
Nach Überabtastung um den Faktor  $I = 3$  und Nullenstopfen wird ( $I-1$ ) mal der letzte Wert wiederholt (digitales Zero-Order Hold). Die D/A Wandlung findet bei der Abtastfrequenz  $f_{S2} = 3f_{S1}$  statt. Wie kann man die Wiederholung des letzten Wertes durch ein einfaches Filter beschreiben?

e) **Vergleichen Sie die Spektren aus a) - d)** und erläutern Sie Unterschiede und Gemeinsamkeiten.

Skizzieren Sie für jeden Fall das Blockschaltbild mit Abtastfrequenzen und ggf. Eckfrequenzen des Filters. Tragen Sie in die Diagramme und Tabellen auf der nächsten Seite ein:

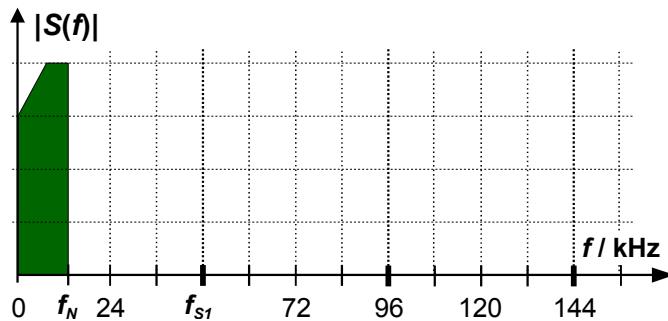
- das **Spektrum** des Audiosignals nach der D/A-Wandlung mittels schraffierter Flächen
- die **maximale Dämpfung  $a_N$**  in dB im Nutzband (bei welcher Frequenz tritt sie auf? warum?)
- die **niedrigste Imagefrequenz  $f_{im}$**  und die **Dämpfung  $a_{im}(f_{im})$**  in dB bei dieser Frequenz.
- für Fall d) zusätzlich die Dämpfung des digitalen Interpolationsfilters im Nutzband  $a_{N,I}$  und bei der niedrigsten Imagefrequenz  $a_{im,I}$

a) **Direkte D/A Wandlung**



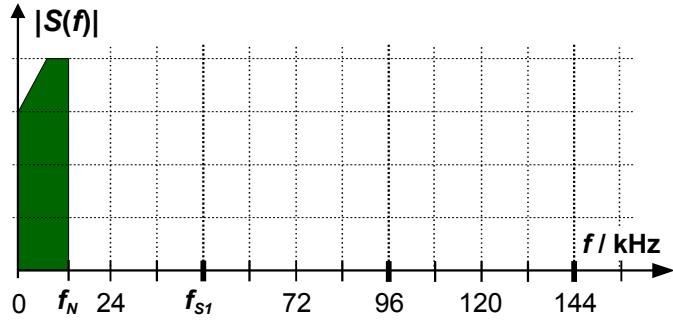
$a_N$	_____
$f_{im}$	_____
$a_{im}$	_____

b) **Überabtastung um den Faktor  $I = 3$  ohne Interpolation**

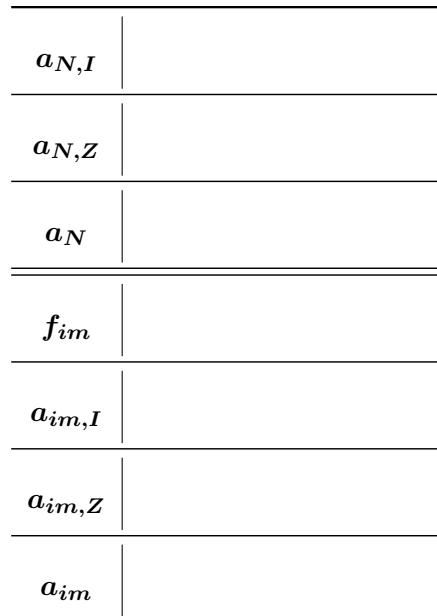
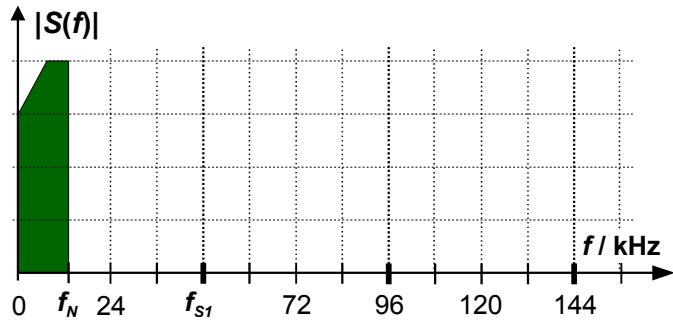


$a_N$	_____
$f_{im}$	_____
$a_{im}$	_____

c) Überabtastung um den Faktor  $I = 3$  mit idealer Interpolation



d) Überabtastung um den Faktor  $I = 3$  mit digitaler ZOH Interpolation



### 9.3 Ideales Interpolationsfilter → M9.3

Wahr oder falsch? Ein ideales Interpolationsfilter ( $iI$ ) ...

- a) filtert alle Wiederholspektren und Images aus dem Basisband
- b) filtert alle Wiederholspektren und Images aus dem Nutzband
- c) hat möglichst weit auseinanderliegende Eckfrequenzen von Sperr- und Durchlassband
- d) entfernt alle Images
- e) lässt das Nutzband unverändert
- f) hat keinen Einfluss auf Wiederholspektren
- g) lässt Images unverändert
- h) verhindert Rückfaltungen ins Basisband
- i) verhindert Rückfaltungen ins Nutzband
- j) kann im Durchlassband eine beliebige Verstärkung haben

## 9.4 Oversampling DAC → Aufgabe M9.4

Ein mit  $f_{S1} = 8 \text{ kHz}$  abgetastetes Sprachsignal im Telefonnetz (Nutzband 0...3,3 kHz) soll mit einem Zero-Order-Hold (ZOH) - DAC und analogem Tiefpassfilter  $H_{RKTP}$  rekonstruiert werden. Als Testsignal wird ein ideal abgetastetes Sinussignal mit  $f_{sig} = 3,3 \text{ kHz}$  und  $\hat{u} = 5 \text{ V}$  verwendet. Eine Realisierung ohne Oversampling (Abb. 9.3a) und eine mit (Abb. 9.3b) sollen hinsichtlich der si-Verzerrungen des Testsignals verglichen werden, Imagekomponenten von  $y_s$  sollen Amplituden unter 50 mV haben.

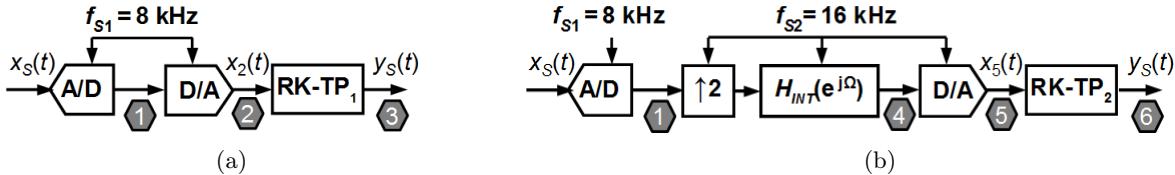


Abb. 9.3: DAC ohne (a) und mit (b) Oversampling zu Aufgabe 9.4

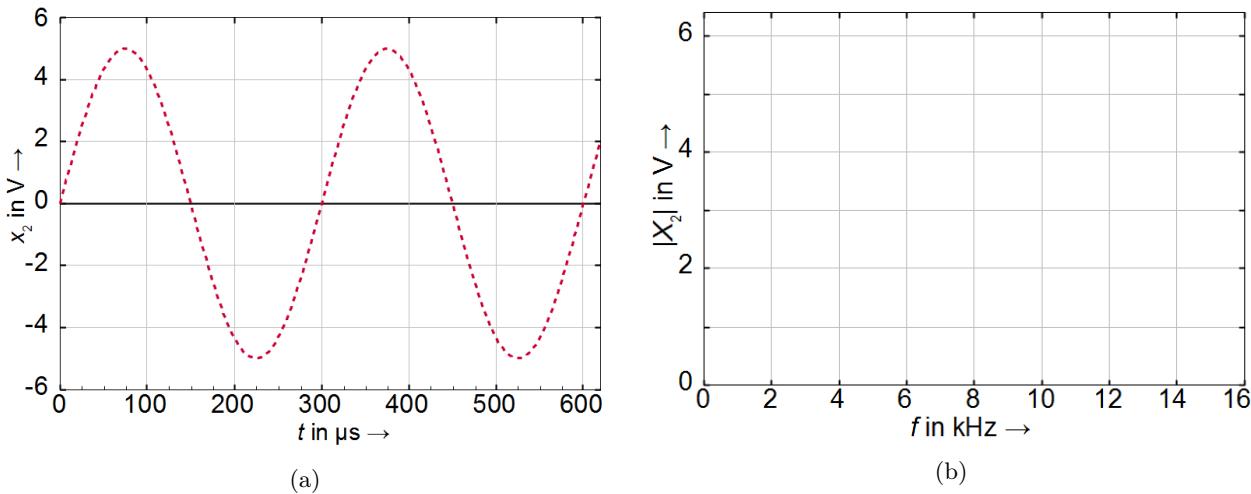


Abb. 9.4: ZOH-Signal  $x_2(t)$  und Amplitudenspektrum  $|X_2(f)|$  zu Aufgabe 9.4a

- D/A-Wandlung ohne Oversampling:** Skizzieren Sie in Abb. 9.4a) das Signal  $x_2(t)$  nach dem ZOH-DAC und in Abb. 9.4b) das dazugehörige Amplitudenspektrum  $X_2(f)$ . Bei welcher Frequenz tritt das erste Image auf? Wie stark werden das Nutzsignal und das erste Image durch die si-Charakteristik der ZOH-Stufe gedämpft?
- Interpolations-Tiefpass:** Skizzieren Sie in Abb. 9.5a) die Spezifikationen für den Interpolations-Tiefpass aus Abb. 9.3b), der als Halbband-FIR-Filter entworfen werden soll. Welche Vorteile bietet eine Implementierung als Halbband-Filter?
- D/A-Wandlung mit Oversampling:** Skizzieren Sie in Abb. 9.6a) das Signal  $x_5(t)$  nach dem ZOH-DAC, vernachlässigen Sie dabei die Gruppenlaufzeit des Interpolationsfilters. Skizzieren Sie in Abb. 9.6b) das dazugehörige Amplitudenspektrum  $X_5(f)$  bis  $f_{S2} = 2f_{S1}$ . Bei welcher Frequenz tritt das erste Image auf? Wie stark werden das Nutzsignal und das erste Image durch die si-Charakteristik der ZOH-Stufe gedämpft?
- Skizzieren Sie in Abb. 9.5b) die **Spezifikationen für die analogen Rekonstruktions-Tiefpässe**  $H_{RKTP,1}$  und  $H_{RKTP,2}$  aus Abb. 9.3a) und b).

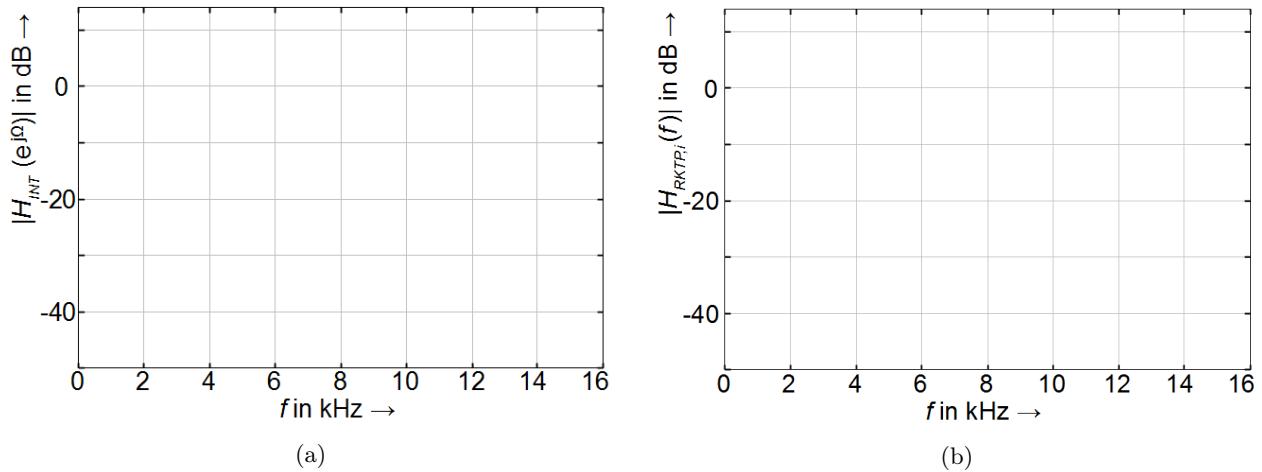


Abb. 9.5: Spezifikationen für Interpolationsfilter  $H_{int}$  (a) und Rekonstruktionstiefpässe  $H_{RKTP,i}$  (b) zu Aufgabe 9.4b und d)

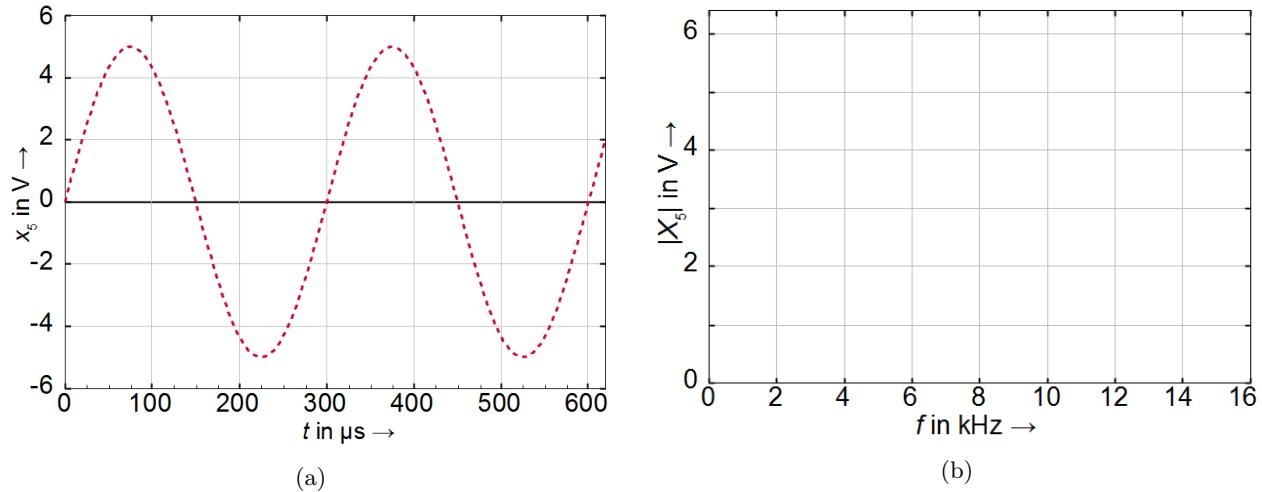


Abb. 9.6: ZOH-Signal  $x_5(t)$  und Amplitudenspektrum  $X_5(f)$  zu Aufgabe 9.4c

## 9.5 + Zweistufige Interpolation → M9.5

Abb. 9.7 zeigt die zu Aufgabe 8.6 transponierte zweistufige Interpolation, die Tiefpassfilter sind identisch zu denen aus Aufgabe 8.6.

- Skizzieren Sie den **Amplitudengang der Kaskade** im logarithmischen Maßstab.
- Am Eingang wird eine Sinusschwingung der Frequenz  $f_{S3}/4$  und der Amplitude 10 dBV (d.h. bezogen auf eine Referenzspannung von  $1 \text{ V}_{\text{eff}} \hat{=} 0 \text{ dBV}$ ) angelegt. Berechnen Sie die Frequenzen aller **Komponenten am Ausgang der Kaskade** im Basisband ( $0 \dots f_{S1}/2$ ) und deren Dämpfung bezogen auf den Eingangspegel.

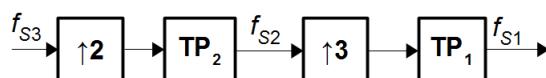


Abb. 9.7: Zweistufige Interpolation zu Aufgabe 9.5



# 10 SRC: Abtastratenwandlung

## Lernziele

Dieses Kapitel wirft die Ergebnisse der letzten beiden Kapitel in einen Topf, es geht darum die Abtastrate eines Signals um einen rationalen Faktor wie z.B.  $7/8$  zu ändern. Wichtiges Thema dabei ist die effiziente Filterung, um Aliasing und unerwünschte Images zu unterdrücken.

### Multiratensystem

### Synchrone Abtastratenwandlung

### Polyphaseninterpolator

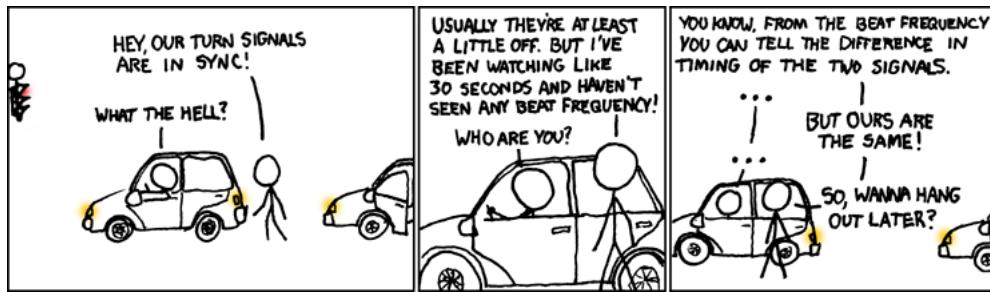


Abb. 10.1: Wenn Ingenieure blinken [<http://xkcd.com/165/>]

### Hinweise:

- In Multiratensystemen muss bei normierten Frequenzen immer gekennzeichnet werden, auf welche Abtastfrequenz sie bezogen sind!
- Sofern nicht anders angegeben, soll Interpolation durch Nullenstopfen erfolgen. Die Amplitudengenauigkeit um den Interpolationsfaktor  $I$  soll im nachfolgenden Tiefpass erfolgen.
- Die Eckfrequenz des Sperrbands soll so *hoch* gewählt werden, dass das Signal gerade noch nicht durch Aliasing oder Images verfälscht wird.
- Die Eckfrequenz des Durchlassbands soll so *niedrig* gewählt werden, dass die Nutzsignalbandbreite gerade noch nicht beschnitten wird.
- Auch mit einer niedrigeren Eckfrequenz des Sperrbandes oder mit einer höheren Eckfrequenz des Durchlassbands würde das System einwandfrei funktionieren, aber:

Für einen möglichst entspannten Tiefpassentwurf müssen die Eckfrequenzen von Durchlass- und Stoppband möglichst weit auseinanderliegen (breiter Übergangsbereich)!

**Anmerkung:** Ein noch entspannterer Filterentwurf ist möglich, wenn Aliasing im Übergangsbereich (außerhalb des Nutzbands) zulässig ist - dann kann die Eckfrequenz des Sperrbandes noch etwas größer gewählt werden.

## Zusätzliche Literatur

**F. J. Harris**, *Multirate signal processing for communication systems*, Prentice Hall PTR, Upper Saddle River, NJ, USA, 2004, 478 S., ca. 90 €.

Sehr praxisnahes Buch speziell zu Multiratensystemen und effizienten Filtern.

**Norbert Fliege**, *Multiraten-Signalverarbeitung: Theorie und Anwendungen*, Teubner Verlag, 1993, 405 S., ca. 45 €

Gut verständliches Buch, nur noch antiquarisch erhältlich.

**R. E. Crochiere and L. R. Rabiner**, *Interpolation and decimation of digital signals - a tutorial review*, Proc. of the IEEE Vol. 69 (1981), 300–331.

Sehr gute (und sehr tiefgehende) Einführung zu Dezimierung und Interpolation

**R. Crochiere and L. Rabiner**, *Optimum FIR digital filter implementations for decimation, interpolation, and narrow-band filtering*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 23 (1975), no. 5, S. 444 – 456.

Grundlegendes Paper mit Theorie und Implementierung von Multistage-Dezimatoren und Interpolatoren.

**L. Rabiner and R. Crochiere**, *A novel implementation for narrow-band FIR digital filters*, IEEE Transactions on Acoustics, Speech and Signal Processing, Vol. 23 (1975), no. 5, S. 457 – 464.

Begleitendes Paper zu letztem mit detaillierter Beschreibung von schmalbandigen Filtern in Multiraten-Implementierung.

**Wolfram Alpha**, *Multirate Signal Processing: Upsampling*, <http://demonstrations.wolfram.com/MultirateSignalProcessingUpsampling/>

Interaktive Demonstration, die den Effekt von Upsampling, gefolgt von Filterung im Zeit- und Frequenzbereich zeigt.

## Spezielle Python-Befehle

Funktion	Python	Matlab	Beschreibung
<code>upfirdn</code>	<code>y = sig.upfirdn(h,x,p,q)</code>	<code>y = upfirdn(x,h,p,q)</code>	Upsampling von $x$ um $p$ , Filtern mit FIR-Filter $h$ , Downsampling um $q$
<code>resample</code> (Zeitber.)	<code>y = sig.resample_poly(x, p,q, window='kaiser',5)</code>	<code>y = resample(x, p, q [, N, beta])</code>	Wie <code>upfirdn</code> , mit effizienter Polyphasenimplementierung.
<code>resample</code> (Freq.ber.)	<code>y = sig.resample(x, num, t=None, window=None)</code>	--	Resampling auch von nicht äquidistanten Daten mit Hilfe der FFT.

Tab. 10.1: Spezielle Befehle zu Kap. 10

### Anmerkungen zu Tab. 10.1:

**UpFIRDown** arbeitet im Zeitbereich, es erhöht die Anzahl der Samples von  $x$  durch Nullenstopfen um den Faktor  $p$  (Upsampling), filtert mit der Impulsantwort  $h$  eines FIR-Filters und reduziert die Anzahl der Samples danach um den Faktor  $q$  (Downsampling). Mit  $h=[1]$  deaktiviert man das FIR-Filter, z.B. `y = upfirdn([1], x, 3, 1)`.

**Resample (Polyphase):**  $x$  wird um den Faktor  $p/q$  mit Polyphasenfilterung resampled. Während des Resamplingprozesses wird defaultmäßig ein FIR Anti-Aliasfilter verwendet, das mit einem Kaiserfenster (**beta=5**) entworfen wurde. Die Länge des resultierenden Arrays ist  $\text{ceil}(\text{length}(x)*p/q)$ . Matlab kann im Gegensatz zur Python-Variante auch mit nicht-äquidistanten Sequenzen umgehen, optional kann die Anzahl **N** der Werte spezifiziert werden, die vor und nach dem aktuellen Sample  $x(k)$  beim Resampling berücksichtigt werden (Default **N = 10**), um Randeffekte abzumildern.

**Resample (Fourier):**  $x$  wird auf **num** Abtastwerte mit Hilfe der Fourier-Methode entlang der gewählten Achse neu abgetastet. Bei langen Sequenzen sollte man darauf achten, eine Zweierpotenz zu wählen, um die Berechnung der FFT zu beschleunigen! Die Methode ist am besten geeignet für periodische Signale - der letzte und der erste Wert der neu abgetasteten Sequenz werden „passend gemacht“ und müssen ggf. abgeschnitten werden.

Optional können angegeben werden: **t** ist ein optionales Array mit den Zeitpunkten bzw. Positionen von  $x$  (muss gleiche Länge haben und äquidistant sein, dann wird ein Tuple zurückgegeben mit den neuen Zeitwerten). Außerdem kann spezifiziert werden, mit welchem Fenster die Daten im Frequenzbereich vor dem Zero-Padding ausgeblendet werden.

**Signal Processing Library** (<http://mubeta06.github.io/python/sp/multirate.html>) rüstet ein paar Multirate-Befehle in Python nach, die in **scipy** fehlen, und zwar **downsample**, **upsample**, **upfirdn**, **interp** und **resample**.

## Python

Flexibler sind die Interpolatoren aus **scipy.interpolate**: Im Code-Schnipsel Listing 10.1 werden durch kubische Spline-Interpolation nicht-äquidistante Wertepaare der Funktion  $y = f(x)$  in äquidistante, dichter abgetastete Wertepaare  $y_{\text{new}} = f(x_{\text{new}})$  umgewandelt:

---

```

1 import numpy as np
2 from scipy import interpolate
3 import matplotlib.pyplot as plt
4 # Definiere y = f(x) mit 13 nicht-äquidistanten Wertepaaren
5 x = np.array([0,1,2,3,4,5,6,6.5,7,7.5,8,8.5,9])
6 y = np.cos(-x**2/4.0)
7 f = interpolate.interp1d(x, y, kind = 'quadratic')
8
9 xnew = np.linspace(x[0], x[-1], 50) # 50 äquidistante x-Werte
10 ynew = f(xnew) # Berechne interpolierte Werte bei xnew
11
12 plt.plot(x,y, 'bo-')
13 plt.plot(xnew,ynew, 'g.-')
14 plt.show()
```

---

Lst. 10.1: Interpolation / Resampling mit Python

Natürlich kann man genauso äquidistante in (anders) äquidistante Wertepaare umrechnen („normale“ Abtastratenwandlung), es lassen sich verschiedene Arten der Interpolation (ZOH, linear, Spline k-ter Ordnung, ...) wählen. **y** und **ynew** können auch mehrdimensionale Arrays (z.B. Stereosignal) sein, dann werden mehrere Werte zu den gleichen **xnew** Punkten berechnet. Bei langen Sequenzen spart man Sortierzeit, wenn die Wertepaare zeitlich monoton sortiert und die Option **assume\_sorted = True** gesetzt sind.

Eine weitere Alternative ist **pandas.DataFrame.resample** aus dem **pandas** Modul (noch nicht selbst ausprobiert).

## Notebooks und Labs

Filename	Beschreibung
<b>Notebooks</b>	
SRC-Intro	Grundlegende Demonstration von Resampling

Tab. 10.2: Jupyter Notebooks und Labs zu Kap. 10

### 10.1 Einfache Abtastratenwandlung → M10.1

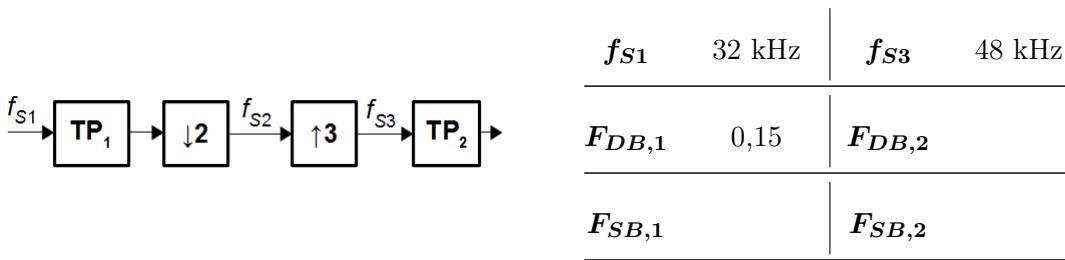


Abb. 10.2: Synchrone Abtastratenwandlung zu Aufgabe 10.1

In Abb. 10.2 ist eine Kaskade aus Dezimator ( $R = 2$ ) und Interpolator ( $I = 3$ ) zur synchronen Wandlung der Abtastrate von  $f_{S1} = 32$  kHz auf  $f_{S3} = 48$  kHz dargestellt. Die Durchlassgrenze von TP<sub>1</sub> ist  $F_{DB,1} = f_{DB,1}/f_{S1} = 0,15$ , die Tiefpässe sperren zunächst ideal.

- a) Welche **Aufgaben** haben die Filter TP<sub>1</sub> und TP<sub>2</sub>?
- b) Berechnen Sie die **Eckfrequenz des Sperrbands von TP<sub>1</sub>**,  $F_{SB,1} = f_{SB,1}/f_{S1}$  so, dass keinerlei Aliasing auftritt.
- c) Berechnen Sie die **Eckfrequenz des Sperrbands von TP<sub>2</sub>**,  $F_{SB,2} = f_{SB,2}/f_{S3}$  so, dass alle Images unterdrückt werden. Berechnen Sie außerdem die Eckfrequenz des Durchlassbands  $F_{DB,2} = f_{DB,2}/f_{S3}$ .

Die Tiefpässe haben jetzt eine Sperrdämpfung von nur 20 dB, daher treten Aliasingeffekte auf und Images werden nicht mehr vollständig unterdrückt. Als Testsignale werden nacheinander Sinustöne eingespeist mit den Frequenzen  $F_{0,\alpha} = f_{0,\alpha}/f_{S1} = 0,1$  und  $F_{0,\beta} = f_{0,\beta}/f_{S1} = 0,4$ .

- d) Welche **Frequenzkomponenten**  $f/f_{S2}$  treten am Ausgang des *Dezimators*, welche Frequenzkomponenten  $f/f_{S3}$  am Ausgang des *Interpolators* im jeweiligen Basisband  $0 \dots f_{S,i}/2$  auf?
- e) **Entwerfen Sie mit Python / Matlab** linearphasige Dezimations- und Interpolationsfilter so, dass das Durchlassband des Gesamtsystems keine Welligkeit hat. Die Sperrdämpfung soll größer als 80 dB sein, der Ripple im Durchlassband kleiner als 0,1% sein.

### 10.2 Ideale Abtastratenwandlung → M10.2

In Abb. 10.3 ist eine Kaskade aus Dezimator ( $R = 5$ ) und Interpolator ( $I = 6$ ) zur Wandlung der Abtastrate dargestellt.  $s_1[n]$  ist das Ausgangssignal eines 8-Bit Oversampling ADCs, der mit der Abtastrate  $f_{S1} = 100$  kHz arbeitet, die Nutzbandbreite ist  $f_N = 8$  kHz (Abb. 10.4).

Es wird zunächst angenommen, dass die Tiefpässe ideal sperren.

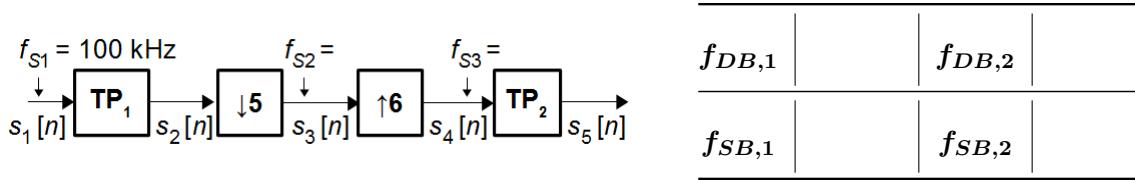


Abb. 10.3: Abtastratenwandlung zu Aufgabe 10.2 und 10.3

- Geben Sie das **Verhältnis der Abtastraten**  $f_{S1}$ ,  $f_{S2}$  und  $f_{S3}$  zueinander an.
- Welche Aufgabe hat TP<sub>1</sub>? Berechnen Sie die **Eckfrequenzen von TP<sub>1</sub>** (Durchlass- und Sperrband  $F_{DB1}$  und  $F_{SB1}$ ) so, dass auch im Übergangsbereich kein Aliasing auftritt. Das Nutzsignal darf nicht beschnitten werden. Skizzieren den Frequenzgang  $|H_{TP1}(e^{j2\pi F})|$  von TP<sub>1</sub> sowie die Spektren  $|S_2(e^{j2\pi F})|$  und  $|S_3(e^{j2\pi F})|$  in Abb. 10.4.
- Welche Aufgabe hat TP<sub>2</sub>? Bestimmen Sie die **Eckfrequenzen von TP<sub>2</sub>** (Durchlass- und Sperrband  $F_{DB2}$  und  $F_{SB2}$ ) so, dass bei maximaler Bandbreite alle Images unterdrückt werden. Skizzieren Sie den Frequenzgang  $|H_{TP2}(e^{j2\pi F})|$  von TP<sub>2</sub> sowie die Spektren  $|S_4(e^{j2\pi F})|$  und  $|S_5(e^{j2\pi F})|$  in Abb. 10.4.

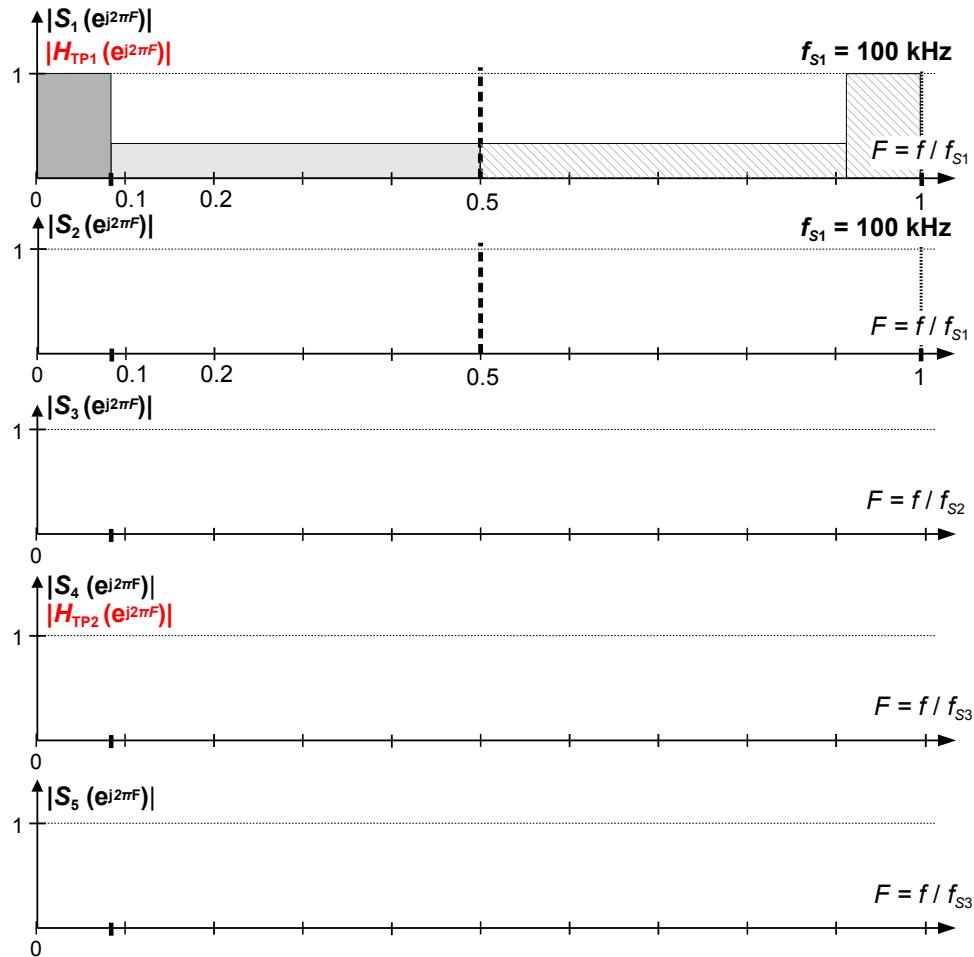


Abb. 10.4: Spektren und Frequenzgänge zu Aufgabe 10.2

Dezimations- und Interpolationsstufe sollen jetzt vertauscht werden.

- Skizzieren Sie das **Blockschaltbild** dieser „vertauschten“ Kaskade und tragen Sie im gleichen

Bild die Eckfrequenzen der Tiefpassfilter ein. Achten Sie dabei auf die richtige Anordnung der Tiefpassfilter für Interpolation und Dezimation - welche Aufgabe hat welches Tiefpassfilter? Kann man das System vereinfachen? Optional: Skizzieren Sie die Spektren und Frequenzgänge (ohne ML).

- e) Bestimmen Sie ***SQNR*** und ***ENOB*** für die Signale  $s_1[n]$  bis  $s_5[n]$  in beiden Systemen. Gehen Sie dazu von einem sinusförmigen Eingangssignal mit  $f_1 < f_N$  aus, das den ADC voll aussteuert und betrachten Sie die Rauschbandbreite an den verschiedenen Punkten im System. Wie ändern sich Rauschleistung und Rauschleistungsdichte bei Erhöhung oder Erniedrigung der Abtastrate, was geschieht in TP<sub>1</sub> und TP<sub>2</sub>?
- f) Welche **Vor- und Nachteile** haben die beiden Systeme? Vergleichen Sie dabei die maximal mögliche Bandbreite  $B_{max}$  und die höchste Abtastfrequenz  $f_{S,max}$  im System. Die Tiefpassfilter sollen als FIR-Filter realisiert werden. Bestimmen Sie die benötigte Ordnung für die beiden Filter mit der Näherungsformel (5.5) und einem Filterentwurfstool und tragen Sie alle Werte in Tab. 10.3 ein. Verwenden Sie Equiripple-Filter mit einer minimalen Sperrdämpfung von  $\delta_{SB} = 10^{-3} \hat{=} 60$  dB und einem maximalen Ripple von  $\delta_{DB} = 10^{-2}$  im Durchlassband.
- g) Geben Sie die **Spezifikationen für analoge Anti-Alias und Rekonstruktionsfilter** für beide Fälle an.

System	$B_{max}$	$f_{S,max}$	$N_{FIR,1}$	$N_{FIR,2}$
			$\approx$	sim
a)				
d) (2 TP)				
d) (1 TP)				—

Tab. 10.3: Lösungen zu Aufgabe 10.2

### 10.3 Abtastratenwandlung mit Aliasing im Übergangsbereich → M10.3

Im System Abb. 10.3 soll jetzt für einen entspannteren Filterentwurf Aliasing in den Übergangsbereich erlaubt sein.

- a) Bestimmen Sie die hierfür **zulässigen Eckfrequenzen** von TP<sub>1</sub>.
- b) Welche **Spezifikationen** werden jetzt für TP<sub>2</sub> benötigt?
- c) Schätzen Sie die **Ordnungen**  $N_{FIR,1}$  und  $N_{FIR,2}$  für die Filter ab.

### 10.4 Abtastratenwandlung HiRes Audio → M10.4

Es sollen Konverter für High Resolution Audiostreams mit Bandbreite  $B = 20$  kHz entworfen werden. Die Streams haben 24 Bit Auflösung und Abtastrate  $f_{S1} = 192$  kHz bzw. 16 Bit Auflösung und Abtastrate  $f_{S2} = 48$  kHz.

- a) Skizzieren Sie das System zur **Reduktion der Abtastrate und Auflösung**  $f_{S1} \rightarrow f_{S3}$
- b) Skizzieren Sie das System zur **Erhöhung der Abtastrate und Auflösung**  $f_{S3} \rightarrow f_{S1}$

## 10.5 Abtastratenwandlung CD Audio → M10.5

Ein digitaler Audiostream mit 16 Bit Auflösung und Abtastrate  $f_{S1} = 44,1 \text{ kHz}$  (CD-Audioformat) soll jetzt auf die Abtastrate  $f_{S3} = 48 \text{ kHz}$  (professional Audio) umgesetzt werden (gleiche Wortbreite).

- a) Finden Sie die **kleinsten Up- und Downsamplingfaktoren** für diese Sampleratenkonvertierung und skizzieren sie das System.
- b) Geben Sie die **Filterspezifikationen** an.



## **Teil II**

# **Musterlösungen**



# M1 LTI-Systeme im Zeitbereich

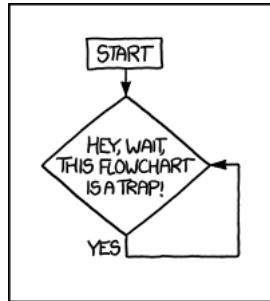


Abb. M1.1: <http://xkcd.com/1195/>

## M1.1 Filterung abgetasteter Signale → A1.1

### a) Abgetastete Signale / Periodizität

Nach der Abtastung eines Signals mit der Frequenz  $f_S = 1/T_S$  sind Funktionswerte nur noch bei den Abtastzeitpunkten  $kT_S$  mit  $k = 0, 1, 2, \dots$  definiert. Daher wird aus

$$\begin{aligned} x(t) &= 1,5 \text{ V} + 0,5 \text{ V} \cos \underbrace{(2\pi \cdot 50 \text{ Hz} \cdot t)}_{\varphi(t)} \\ \Rightarrow x(kT_S) &= x[k] = 1,5 \text{ V} + 0,5 \text{ V} \cos \underbrace{(2\pi \cdot 50 \text{ Hz} \cdot kT_S)}_{\varphi[k]} \end{aligned}$$

Setzt man die Abtastfrequenzen  $f_{S,i}$  ein, erhält man die zeitdiskreten Folgen  $x_i[k]$  (Abb. M1.2a - d):

$$\begin{aligned} x_1[k] &= 1,5 \text{ V} + 0,5 \text{ V} \cos(2\pi k \cdot 50 \text{ Hz}/25 \text{ Hz}) = 1,5 \text{ V} + 0,5 \text{ V} \cos(4\pi k) = 2 \text{ V} \\ x_2[k] &= 1,5 \text{ V} + 0,5 \text{ V} \cos(2\pi k \cdot 50 \text{ Hz}/200 \text{ Hz}) = 1,5 \text{ V} + 0,5 \text{ V} \cos(\pi k/2) \\ x_3[k] &= 1,5 \text{ V} + 0,5 \text{ V} \cos(2\pi k \cdot 50 \text{ Hz}/240 \text{ Hz}) = 1,5 \text{ V} + 0,5 \text{ V} \cos(5\pi k/12) \\ x_4[k] &= 1,5 \text{ V} + 0,5 \text{ V} \cos(2\pi k \cdot 50 \text{ Hz}/(50 \cdot \pi \text{ Hz})) = 1,5 \text{ V} + 0,5 \text{ V} \cos(2k) \end{aligned}$$

Man sieht, dass das Abtastsignal für  $f_{S,1} = 25 \text{ Hz}$  nicht mehr dem ursprünglichen Signal ähnelt; der Wechselanteil ist verschwunden und der Gleichanteil hat den falschen Wert. Dies ist eine Folge der zu geringen Abtastfrequenz; das Nyquistkriterium wird verletzt und es tritt *Aliasing* auf (→ Kap. 8).

Die cos-Funktion ist mit  $2\pi$  periodisch, daher ist die kontinuierliche Cosinusfunktion  $s(t)$  immer periodisch mit dem Kehrwert der Frequenz  $T_1 = 1/f_1$ :

$$\begin{aligned} s(t) &= \cos(2\pi f_1 t) = \cos(2\pi f_1 t + 2\pi L) = \cos(2\pi f_1(t + LT_1)) = s(t + LT_1) \quad \text{mit } L \in \mathbb{Z} \\ \Leftrightarrow \varphi(t) &= 2\pi f_1 t = 2\pi f_1 t + 2\pi L - 2\pi L = \varphi(t + LT_1) - 2\pi L \end{aligned}$$

Die Phase  $\phi(t)$  steigt linear mit  $t$  an (lineare Phase).

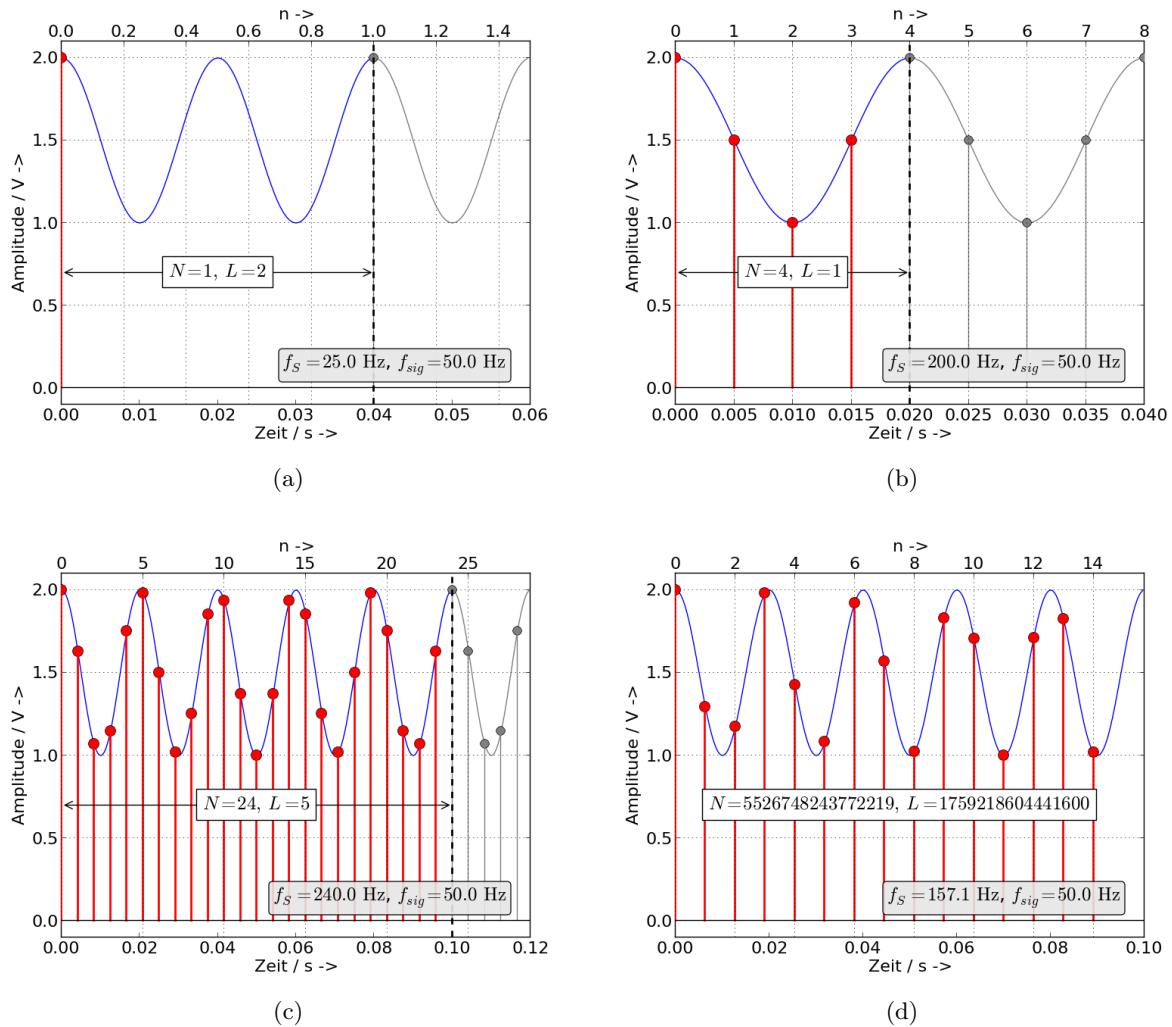


Abb. M1.2: Eingangssignal  $x(t) = 1,5 \text{ V} + 0,5 \text{ V} \cos(2\pi \cdot 50 \text{ Hz} \cdot t)$ , abgetastet mit verschiedenen Samplingraten  $f_S$

Die zeitdiskrete cos-Funktion  $s[k]$  ist nur bei  $t = kT_S$  definiert; sie ist nur dann periodisch wenn  $N$  Abtastwerte  $L$  Perioden von  $s(t)$  ergeben bzw. wenn  $\varphi[N] \bmod 2\pi \equiv 0$ :

$$\begin{aligned} \cos(2\pi f_1 k T_S) &\stackrel{!}{=} \cos(2\pi f_1 (k + N) T_S - 2\pi L) \quad \text{mit } L, N \in \mathbb{N}_0 \\ \Rightarrow \varphi(k T_S) &= 2\pi f_1 k T_S \stackrel{!}{=} 2\pi f_1 (k + N) T_S - 2\pi L = \varphi((k + N) T_S) - 2\pi L \end{aligned}$$

In Kurzschreibweise  $\varphi[k] := \varphi(k T_s)$ :

$$\begin{aligned} \varphi[k] &\stackrel{!}{=} \varphi[k + N] - 2\pi L \\ \Leftrightarrow 2\pi k T_S / T_1 &\stackrel{!}{=} 2\pi (k + N) T_S / T_1 - 2\pi L \\ \Leftrightarrow k T_S / T_1 &\stackrel{!}{=} (k + N) T_S / T_1 - L \end{aligned}$$

Diese Gleichung muss für jeden Zeitpunkt  $k$  erfüllt sein (LTI-System), also auch für  $k = 0$ :

$$0 = N T_S / T_1 - L \Rightarrow N T_S = L T_1$$

Damit die obige Gleichung für ganzzahlige Werte von  $L, N$  erfüllbar ist <sup>1</sup>, müssen Abtastperiode  $T_S$  und Signalperiode  $T_1$  (bzw. Abtastfrequenz  $f_S$  und Signalfrequenz  $f_1$ ) ganzzahlige Vielfache

<sup>1</sup>sog. diophantische Gleichung

einer gemeinsamen Zeiteinheit  $T_c$  (bzw. Frequenz  $f_c$ ) sein. Sie müssen „kommensurabel“ (lat.: zusammen messbar) sein:

$$NT_S = LT_1 = NLT_c$$

$$\Rightarrow T_c = \text{gcf}(T_S, T_1) \Rightarrow L = T_S/T_c \quad \text{und} \quad N = T_1/T_c$$

oder  $f_c = \text{gcf}(f_1, f_S) \Rightarrow L = f_1/f_c \quad \text{und} \quad N = f_S/f_c$

**gcf** ist der „greatest common factor“, also der größte gemeinsame Faktor oder Teiler von  $T_1$  und  $T_S$  (im Allgemeinen nicht ganzzahlig). Damit erhält man die kleinste Lösung für  $N, L$ ; Vielfache davon  $mL = mNf_1/f_S$  mit  $m = 2, 3, \dots$  sind natürlich immer möglich.

Mathematisch ausgedrückt:

$$\exists T_c \in \mathbb{R} \text{ so, dass } T_1 = NT_c \wedge T_S = LT_c \text{ mit } T_1, T_S \in \mathbb{R} \text{ und } N, L \in \mathbb{Z}.$$

$$\Rightarrow \frac{T_1}{T_S} = \frac{f_S}{f_1} = \frac{N}{L} = c \in \mathbb{Q}.$$

Listing M1.1 zeigt eine rekursive Python-Implementierung der gcf - Funktion. Der Variablen b wird dabei in so lange der Rest der ganzzahligen Division a % b zugewiesen (aus numerischen Gründen auf 6 Stellen gerundet) bis dieser Null ist.

$f_S/\text{Hz}$	$n$	0	1	2	3	4	5	6	7	8	9
25	$x_1[n]$	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00	2,00
200	$x_2[n]$	2,00	1,50	1,00	1,50	2,00	1,50	1,00	1,50	2,00	1,50
240	$x_3[n]$	2,00	1,63	1,07	1,15	1,75	1,98	1,50	1,02	1,25	1,85
50π	$x_4[n]$	2,00	1,29	1,17	1,98	1,43	1,08	1,92	1,57	1,02	1,83
200	$y_{MA,4}[n]$	0,40	0,70	0,90	1,20	<b>1,60</b>	1,50	1,40	1,50	1,60	1,50
	$y_{MA,3}[n]$	0,50	0,88	1,12	<b>1,50</b>	1,50	1,50	1,50	1,50	1,50	1,50
	$y_{casc}[n]$	0,22	0,61	1,11	1,33	<b>1,44</b>	1,50	1,56	1,50	1,44	1,50
	$y_{Int}[n]$	1,00	1,25	1,12	1,31	1,66	1,58	1,29	1,39	1,70	1,60

Tab. M1.1: Ergebnisse zu Aufgabe 1.1

## b) Filterung

### i. Moving Average Filter

Das System in Abb. 1.3a hat die Ordnung  $N = 4$  (d.h. 5 Taps); es bestimmt den gleitenden Mittelwert (*Moving Average*, MA) des Eingangssignals der jeweils letzten 5 Samples. Es hat die Impulsantwort

$$h_{MA4}[n] = \{1; 1; 1; 1; 1\}$$

Wird ein Gleichsignal mit der Amplitude 1 an den Eingang gelegt, ist das Ausgangssignal konstant 5. Daher muss das Signal am Ausgang des Filters durch 5 dividiert werden, damit es korrekt skaliert ist.

Mit dem fünften Eingangsample ( $n = 4$ ) sind alle Register (= Verzögerungsglieder) belegt, das Filter ist damit eingeschwungen. Die berechneten Werten für  $y_{MA4}[n]$  passen z.B. zu folgender Gleichung:

$$y_{MA4}[n] = 1,5 \text{ V} + 0,1 \text{ V} \cos(\pi(n - 4)/2) \quad \text{für } n \geq 4$$

Hieraus liest man eine Amplitude von nur noch 100 mV für das überlagerte Brummen ab, das somit somit um den Faktor 5 gedämpft wird ( $\hat{\equiv} 20 \log_{10} 1/5 = -14 \text{ dB}$ ). Der abgelesene Wert ist nur eine Schätzung, für verlässlichere Werte müsste man das Signal interpolieren und so sicherstellen, dass man auch wirklich Maxima / Minima der Funktion getroffen hat. Einfacher bekommt man das gewünschte Ergebnis im Frequenzbereich (Aufgabe 2.1).

Bei einer Ordnung von 3 lautet die Impulsantwort

$$h_{MA3}[n] = \{1; 1; 1; 1\},$$

das Ausgangssignal muss für korrekte Skalierung durch vier geteilt werden. Das Brummen im Ausgangssignal ist jetzt komplett verschwunden. Das liegt daran, dass bei einer Abtastrate von  $f_{S,2} = 200 \text{ Hz}$  genau eine Periode des 50 Hz Störsignals in das MA-Filter „passt“, positive und negative Halbwelle heben sich daher immer auf, das Filter hat eine perfekte Brummunterdrückung für diese Signalfrequenz.

## ii. Kaskadiertes Moving Average Filter

Das System in Abb. 1.3b besteht aus zwei kaskadierten MA-Filtern der Ordnung  $N = 2$ , es hat daher ebenfalls die Ordnung  $N = 4$ . Die Impulsantwort

$$h_{kask}[n] = \{1; 2; 3; 2; 1\}.$$

kann man entweder durch Ausprobieren ermitteln oder indem man die Impulsantworten der beiden Teilfilter  $h_1[n] = h_2[n] = \{1; 1; 1\}$  mit einander faltet. Wird ein Gleichsignal mit der Amplitude 1 an den Eingang gelegt, ist das Ausgangssignal konstant  $1 + 2 + 3 + 2 + 1 = 9$ . Daher muss das Signal am Ausgang des Filters durch 9 dividiert werden, damit es korrekt skaliert ist.

Mit dem fünften Eingangsample sind auch hier alle Register belegt, das Filter ist damit eingeschwungen. Hier liest man eine Amplitude von 60 mV ab und erhält

$$y_{kask}[k] = 1,5 \text{ V} - 0,06 \text{ V} \cos(\pi(k - 4)/2) \quad \text{für } k \geq 4.$$

Die Dämpfung des Brummens ist damit ca.  $60/500 = 0,12 \hat{\equiv} -18,4 \text{ dB}$ .

Das kaskadierte Filter kann durch ein einfaches FIR-Filter ersetzt werden mit den Koeffizienten  $h_i = \{1; 2; 3; 2; 1\}$ . Ein Nachteil der nicht-kaskadierten Form ist, dass Multiplizierer (wenn auch nur einfache) benötigt werden.

## iii. Gedämpfter Integrator

Das gedämpfte Integrator in Abb. 1.3b ist ein rekursives System, das eine unendliche Impulsantwort hat. Das macht die manuelle Berechnung aufwändiger bis unmöglich, man kann sich behelfen indem man die Impulsantwort abbricht, also z.B.  $h_{Int}[n] = 0,5^n \approx \{1; 0,5; 0,25; 0,125; 0,0625\}$ .

Aufgrund der unendlichen Impulsantwort ist das Filter strenggenommen niemals eingeschwungen, man sieht aber aus Simulation oder Rechnung, dass das Ausgangssignal nach einer Weile auf die periodische Sequenz  $\{1,7; 1,6; 1,3; 1,4; \dots\}$  zusteuernt, aus der man eine Amplitude von 200 mV abschätzen kann.

c) **Simulation mit Matlab / Python**

Hier zu gibt es das Notebook `LTI_sensor_ML`.

d) **Kritischer Pfad**

Der kritische Pfad erstreckt sich beim einfache MA und beim kaskadierten MA-Filter über 4 Additionen, entsprechend  $\tau_{krit} = 8$  ns. Die maximale Taktfrequenz ist damit  $f_{S,max} = 1/\tau_{krit} = 125$  MHz.

Beim kaskadierten MA-Filter könnte der kritische Pfad allerdings leicht auf 2 Additionen verkürzt werden, wenn man ein zusätzliches Register zwischen beide Hälften einsetzt. Damit erhält man eine maximale Taktfrequenz von 250 MHz, auf Kosten einer zusätzlichen Latenz.

Beim gedämpften Integrator sind ein Multiplizierer und ein Addierer im kritischen Pfad, entsprechend  $\tau_{krit} = 12$  ns und  $f_{S,max} = 1/\tau_{krit} = 83,3$  MHz.

e) **Taktzyklen pro Sample**

Beim MA und beim kaskadierten MA-Filter müssen pro Sample 4 Additionen durchgeführt und die Samples in einem Ringbuffer neu indiziert werden. Der Aufwand für die Speicherverwaltung soll vernachlässigt werden, daher müssen hier pro Eingangssample 4 Additionen berechnet werden. Beim gedämpften Integrator werden eine Addition und eine Multiplikation benötigt.

f) **Effiziente Implementierung**

Beim einfachen MA-Filter mit  $L$  Taps (Ordnung  $N = L - 1$ ) unterscheidet sich der aktuelle Ausgangswert  $y[k]$  vom vorherigen Wert  $y[k - 1]$  nur durch den neuen Eingangswert  $x[k]$  und Wegfall des letzten Speicherwerts  $x[k - (N + 1)]$ :

$$\begin{aligned} y_{MA}[k] &= \sum_{i=0}^N x[k - i] = x[k] - x[k - (N + 1)] + \sum_{i=0}^N x[k - 1 - i] \\ &= x[k] - x[k - (N + 1)] + y_{MA}[k - 1] \end{aligned}$$

Diese Gleichung lässt sich durch das Netzwerk in Abb. M1.3b implementieren, einer Kettenschaltung aus einem sog. Kammfilter und einem Integrator. Aus numerischen Gründen ( $\rightarrow$  Kap. 6 und Kap. 10) werden allerdings meist beide Teilfilter vertauscht, die resultierende Struktur wird *Cascaded Integrator Comb* (CIC) - Filter genannt. Sie benötigt nur zwei Addierer, die auch den kritischen Pfad ausmachen. Ein „normales“ MA-Filter der Ordnung  $N$  benötigt im Vergleich  $N$  Addierer (Abb. M1.3a).

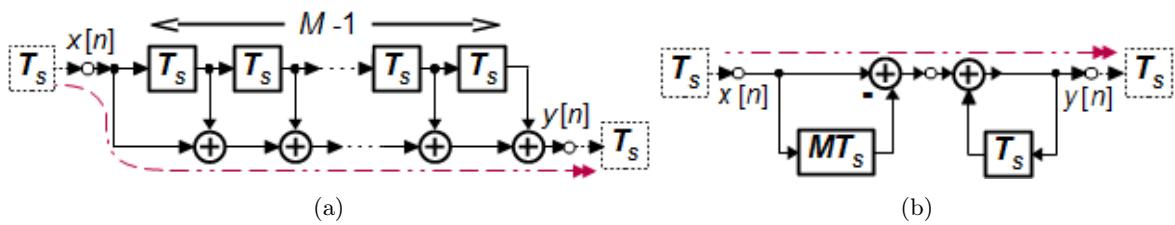


Abb. M1.3: MA-Filter der Ordnung  $M - 1$  und effiziente Implementierung mit kritischen Pfaden

g) **Hochpassfilter**

Ein Hochpassfilter lässt sich z.B. realisieren, indem man die Koeffizienten eines FIR-Filters wählt als  $h[n] = \{1; -1; 1; -1\}$ . Wenn die Summe der Koeffizienten Null ist, werden Gleichsignale unterdrückt. Warum ist das so? Berechnen Sie hierzu allgemein das Ausgangssignal eines FIR-Filters, das mit einem Gleichsignal eingeschwungen ist.

---

```

1 def gcf(a, b):
2 # Greatest Common Factor function for non-integer arguments
3     while b != 0:
4         a, b = b, round((a % b),6) # das heißt: a = b; danach b = round(a % b)
5     return a
6 #
7 T_c = gcf(T_S, T_1)

```

---

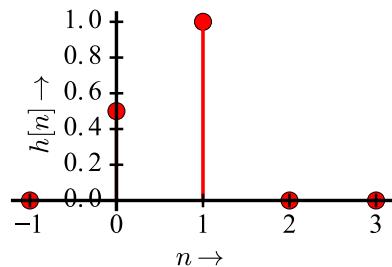
Lst. M1.1: Python-Implementierung der gcf - Funktion zu Aufgabe M1.1

## M1.2 Faltung mit einfachem FIR-Filter → A1.2

### a) Die Impulsantwort $h[n]$ ...

... erhält man als Antwort auf das Eingangssignal  $x[n] = \delta[n] = 1; 0; 0; \dots$  durch „genaues Hinschauen“:  
 $h[n] = \{0,5; 1\}$ .

Bei FIR-Filtern ist die Impulsantwort identisch mit den Koeffizienten des Filters:  $b_0 = 0,5; b_1 = 1$ .

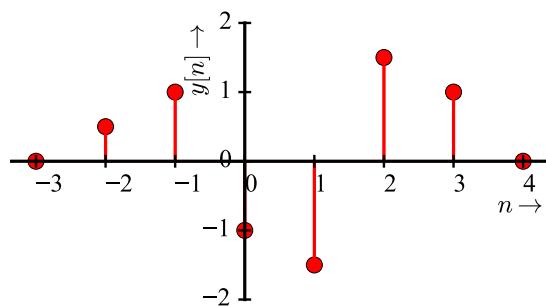


### b) Die Differenzengleichung...

... erhält man ebenfalls durch genaues Hinschauen oder mit Hilfe des Faltungsoperators:

$$y[n] = h[n] * x[n] = \sum_{i=-\infty}^{\infty} h[i]x[n-i] = \sum_{i=0}^{N=1} h[i]x[n-i] = 0,5x[n] + x[n-1]$$

Da die Impulsantwort endlich ist, muss die Faltungssumme nicht von  $n = -\infty \dots \infty$  berechnet werden, sondern nur für die Indices  $i$  bei denen  $h[i] \neq 0$ .

Abb. M1.4: Ausgangssignal  $y[n]$  des FIR-Filters zum Eingangssignal aus Abb. 1.4

### c) Das Ausgangssignal $y[n]$ auf ein gegebenes Eingangssignal $x[n]$ ...

... erhält man auf verschiedene Weisen:

**DZGL:** Direkt aus der DZGL durch Einsetzen von  $x[n]$ :

$$\begin{aligned}
 y[n] &= 0,5x[n] + x[n-1] \\
 &= 0,5(\delta[n+2] - 2\delta[n] + \delta[n-1] + \delta[n-2]) \\
 &\quad + (\delta[n+1] - 2\delta[n-1] + \delta[n-2] + \delta[n-3]) \\
 &= 0,5\delta[n+2] + \delta[n+1] - \delta[n] - 1,5\delta[n-1] + 1,5\delta[n-2] + \delta[n-3]
 \end{aligned}$$

**Faltung:** Das Ausgangssignal ist die Faltung des Eingangssignals mit der Impulsantwort, also die Summe der mit der Höhe der Eingangsimpulse bewerteten und verschobenen Impulsantworten und lässt sich durch die folgenden Schritte durchführen:

- „Umklappen“ von  $h[n]$  oder  $x[n]$  (d.h. Bildung von  $h[-n]$  oder  $x[-n]$ )
- Verschieben und Bewerten
- Aufsummieren der Teilprodukte

$$\begin{aligned}
 y[n] &= h[n] * x[n] = \sum_{i=-\infty}^{\infty} h[i]x[n-i] = \sum_{i=0}^{N=1} h[i]x[n-i] = 0,5x[n] + x[n-1] \\
 &= x[n] * h[n] = \sum_{i=-\infty}^{\infty} x[i]h[n-i] = \sum_{i=-2}^{N=2} x[i]h[n-i] \\
 &= h[n+2] + 0h[n+1] - 2h[n] + h[n-1] + h[n-2]
 \end{aligned}$$

$n$	-2	-1	0	1	2	3	4
$x[n]$	1	0	-2	1	1	0	0
$0,5x[n]$	0,5	0	-1	0,5	0,5	0	
$+ x[n + 1]$		1	0	-2	1	1	0
$y[n] = \sum h[i] \cdot x[n - i] =$	0,5	1	-1	-1,5	1,5	1	0

Tab. M1.2: Zeitdiskrete Faltung  $y[n] = h[n] * x[n]$  zu Aufgabe 1.2

Wie schon in Unterpunkt b) beschrieben, ist es am einfachsten, wenn man die Eingangssequenz  $x[n]$  mit der Impulsantwort  $h[n]$  bewertet und verschiebt, da  $h[n]$  kurz ist (Tab. M1.2). Filtert man eine kurze Eingangssequenz mit einem IIR-Filter (unendliche Impulsantwort) wie in Aufgabe 2.10, ist die Rechnung leichter in umgekehrter Reihenfolge (Tab. M1.3). Die Ergebnisse sind aber identisch, da die Faltung kommutativ ist.

$i$	-3	-2	-1	0	1	2	3	4
$x[i]$	0	1	0	-2	1	1	0	0
$n$	$h[n - i]$	$\cdot \downarrow$	$\cdot \downarrow$					
-2	$h[-2 - i]$	1	0,5					
-1	$h[-1 - i]$		1	0,5				
0	$h[0 - i]$			1	0,5			
1	$h[1 - i]$				1	0,5		
2	$h[2 - i]$					1	0,5	
3	$h[3 - i]$						1	0,5
4	$h[4 - i]$							1

Tab. M1.3: Zeitdiskrete Faltung  $y[n] = x[n] * h[n]$  zu Aufgabe 1.2

- d) Für die Berechnung der DZGL oder des Ausgangssignals von FIR-Filtern ist daher im Allgemeinen die Schreibweise  $h[n] * x[n]$  besser geeignet.

### M1.3 Systemeigenschaften aus Impulsantwort → A1.3

Da die Systeme alle durch eine endliche Impulsantwort beschrieben werden, die zum Zeitpunkt  $n = 0$  beginnt, sind sie sowohl stabil als auch kausal. Alle Systeme lassen sich durch ein transversales System in Direktform (Abb. M1.5a) implementieren, dessen Koeffizienten der Impulsantwort entsprechen,  $h[n] = \sum_i h[i]\delta[n - i]$ . Bei Systemen a) und b) kann die Symmetrie der Koeffizienten genutzt werden, um die Anzahl der Multiplizierer zu minimieren (Abb. M1.5b).

- a) Alle Koeffizienten haben positives Vorzeichen, daher kann das System nur TP-Charakteristik haben (gewichtete Mittelung über die letzten drei Eingangswerte). Dies wird bestätigt durch  $H(z = 1) = H(f = 0) = 1$  und  $H(z = -1) = H(f = f_S/2) = 0$ .

- b) Abwechselnd positive und negative Vorzeichen der Koeffizienten deuten oft auf HP-Charakteristik hin. Um sicher zu gehen, sollte man sich aber auf jeden Fall  $H(z = 1)$  und  $H(z = -1)$  anschauen - Gegenbeispiele:  $\mathbf{b} = [-0.1, 1, -0.1, 1, -0.1]$  (Bandsperre) oder  $\mathbf{b} = [-0.05, 1, 1, -0.05]$  (Tiefpass). In dieser Aufgabe bestätigen  $H(z = 1) = 0$  und  $H(z = -1) = 1$  die HP-Charakteristik.
- c) Wie beim ersten System, deuten rein positive Koeffizienten auf eine TP-Charakteristik hin,  $H_C(z = 1) = 1.5$  und  $H(z = -1) = -0.5$ . Hier ist allerdings eine genauere Analyse notwendig, da das System nicht linearphasig ist (keine Symmetrie der Koeffizienten).

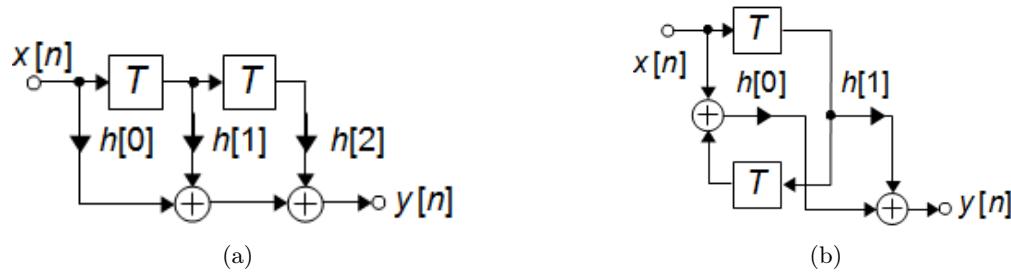


Abb. M1.5: FIR (transversale) Filter in Direktform (a) und in „gefalteter Form“ (b)

## M1.4 Aufstellung der Differenzengleichung für Biquads aus SFG → A1.4

### a) Grafische Umformung

Die Umformung zwischen DF1 und DF2 Systemen ist am einfachsten, wenn man von Abb. 1.5c ausgeht und daraus am Summierknoten zwei Teilsysteme  $H_1(z)$  und  $H_2(z)$  macht. Da LTI-Systeme kommutativ sind, kann man die Teilsysteme vertauschen. Da die Information in beiden Verzögerungsketten identisch ist, kann man sie durch eine ersetzen (Abb. M1.6).

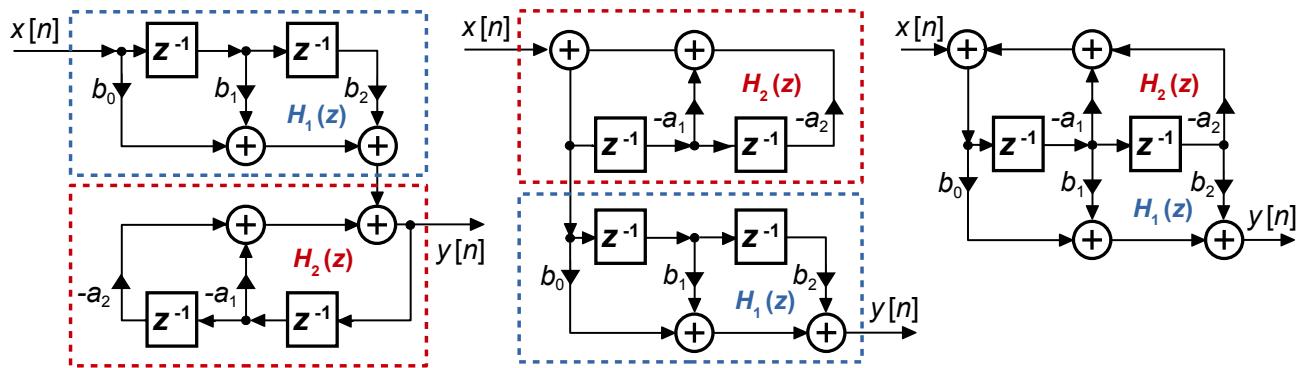


Abb. M1.6: Graphische Umwandlung eines Filters von DF1- zu DF2-Form

Man erhält die transponierten Strukturen Abb. 1.5e und 1.5f durch **Transponieren** der entsprechenden Strukturen, d.h. Vertauschen aller Signalrichtungen, Ersetzen von Addierern durch Verzweigungsknoten und umgekehrt sowie Vertauschen von Eingangs- und Ausgangssignal.

### b) Aufstellung DZGL und Systemfunktion

Für die **DF1 Systeme** Abb. 1.5a und 1.5c kann man tatsächlich „direkt“ die DZGL hinschreiben, wenn man sich klar macht, dass das Ausgangssignal  $y[n]$  am Eingang der unteren Verzögerungskette liegt:

$$y[n] = b_0x[n] + b_1x[n - 1] + b_2x[n - 2] - a_1y[n - 1] - a_2y[n - 2]$$

Aus dieser DZGL erhält man durch Umstellen und  $z$ -Transformation sofort die Systemfunktion aus der Aufgabenstellung.

Für die **DF2 Systeme Abb. 1.5b** und **1.5d** ist es etwas komplizierter. Hier führt man einen Zwischenknoten  $w[n]$  am Eingang der Verzögerungskette ein mit

$$\begin{aligned} w[n] &= x[n] - a_1w[n-1] - a_2w[n-2] \\ \circ\bullet W(z) &= X(z) - a_1W(z)z^{-1} - a_2W(z)z^{-2} = \frac{X(z)}{1 + a_1z^{-1} + a_2z^{-2}} \end{aligned}$$

und erhält

$$\begin{aligned} y[n] &= b_0w[n] + b_1w[n-1] + b_2w[n-2] \\ \circ\bullet Y(z) &= b_0W(z) + b_1W(z)z^{-1} + b_2W(z)z^{-2} = W(z)(b_0 + b_1z^{-1} + b_2z^{-2}) \\ &= X(z) \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \end{aligned}$$

Wenn man möchte, kann aus dieser Form einfach die DZGL ableiten.

Für das **TDF1 System Abb. 1.5e** führt man einen Zwischenknoten  $w[n]$  nach dem ersten Addierer ein und erhält wie beim DF2 System:

$$\begin{aligned} w[n] &= x[n] - a_1w[n-1] - a_2w[n-2] \\ \circ\bullet W(z) &= X(z) - a_1W(z)z^{-1} - a_2W(z)z^{-2} = \frac{X(z)}{1 + a_1z^{-1} + a_2z^{-2}} \end{aligned}$$

Die restliche Rechnung läuft genauso wie beim DF2-System ab.

Für das **TDF2 System Abb. 1.5f** erhält man DZGL/Systemfunktion wieder durch „genaues Hinschauen“, am einfachsten in der Frequenzebene:

$$\begin{aligned} Y(z) &= b_0X(z) + (b_1X(z) - a_1Y(z) + (b_2X(z) - a_2Y(z))z^{-1})z^{-1} \\ &= X(z) \frac{b_0 + b_1z^{-1} + b_2z^{-2}}{1 + a_1z^{-1} + a_2z^{-2}} \end{aligned}$$

## M1.5 Kritischer Pfad und Ressourcenverbrauch 1 → A1.5

Wie man an Abb. M1.7 sieht, können sich die Länge des kritischen Pfads und damit die maximale Taktrate für zwei Implementierungen der gleichen Übertragungsfunktion sowie der Ressourcenverbrauch deutlich unterscheiden:

Ein IIR-Filter zweiter Ordnung benötigt in DF1-Form 4 Register, in DF2-Form 2 Register. Beide Formen benötigen 5 Koeffizientenmultiplizierer und 4 Addierer. Addierer mit 3 Eingängen sind eigentlich zwei Addierer mit je zwei Eingängen und müssen daher doppelt gerechnet werden. Der Ressourcenverbrauch der transponierten Formen ist identisch zu den ursprünglichen Strukturen: Die Anzahl der Register und Multiplizierer ändert sich nicht durch das Transponieren. Bei den Strukturen in Abb. M1.7 sind Anzahl der Summierer und Verzweigungsknoten identisch<sup>2</sup>, so dass sich auch hier nichts durch das Transponieren ändert.

**Extra:** Ist das bei allen Strukturen so? Versuchen Sie, ein Gegenbeispiel zu konstruieren!

Ein IIR-Filter  $N$ -ter Ordnung hat im allgemeinen Fall (keine speziellen Koeffizienten wie 0, 1 oder  $1/2$ )  $N+1$  Koeffizientenmultiplizierer im nicht-rekursiven oder transversalen Pfad (Zähler). Im rekursiven

<sup>2</sup>Addierer und Knoten mit 3 Zweigen müssen doppelt gerechnet werden!

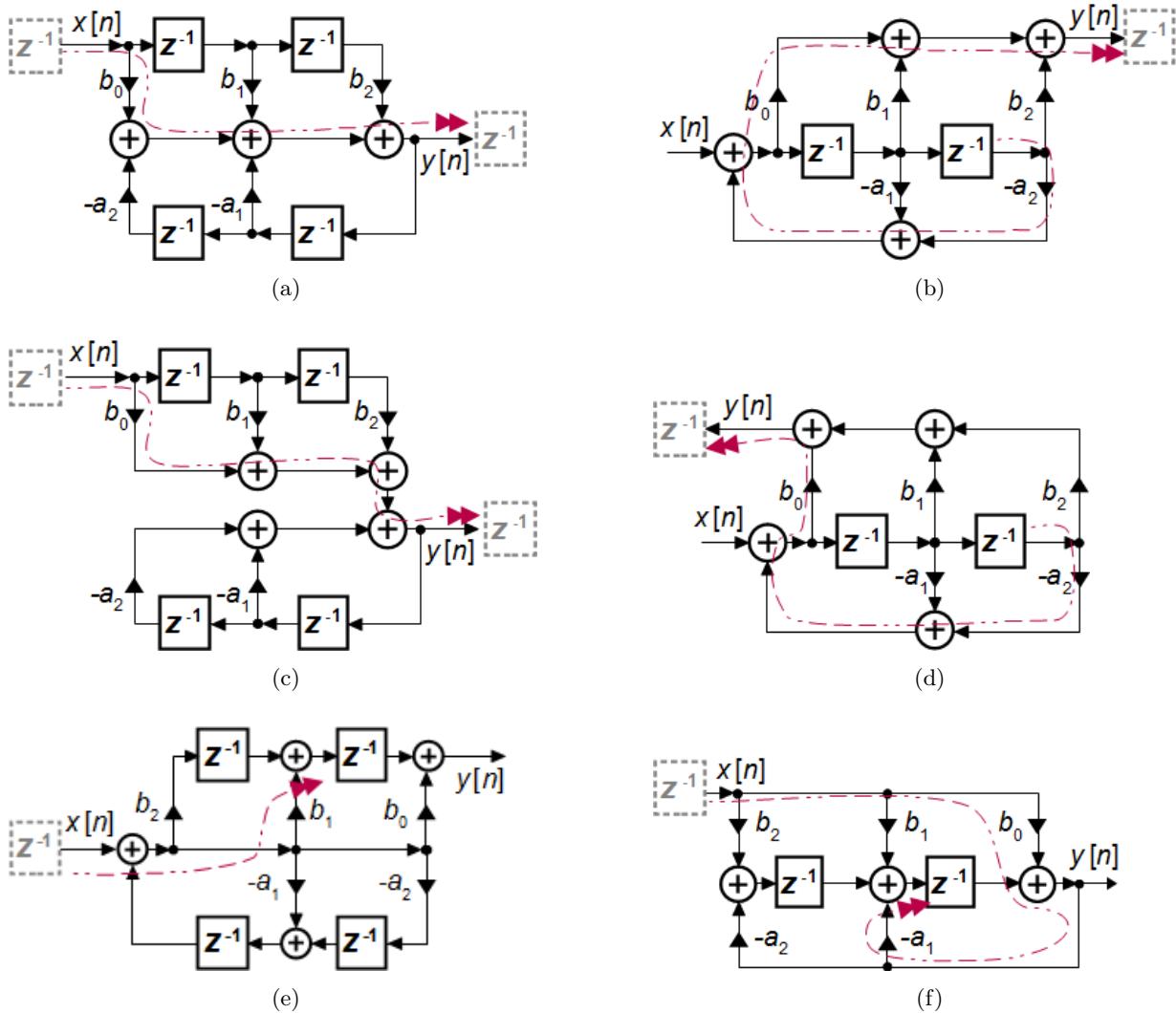


Abb. M1.7: Systeme zweiter Ordnung in Direktform 1 (a), (c) und (e, transponiert) und Direktform 2 (b), (d) und (f, transponiert) mit kritischen Pfaden

Pfad (Nenner) ist ein Koeffizient immer 1 (warum?), daher gibt es hier maximal  $N$  Multiplizierer. Insgesamt werden daher benötigt:

**(T)DF1:**  $2N$  Register,  $2N$  Addierer,  $2N + 1$  Koeffizientenmultiplizierer

**(T)DF2:**  $N$  Register,  $2N$  Addierer,  $2N + 1$  Koeffizientenmultiplizierer

Da die DF2-Struktur den minimalen Ressourcenverbrauch hat, wird sie auch *kanonische* (= minimale) Form genannt.

Der kritische Pfad von Abb. M1.7a (DF1) hängt davon ab, in welcher Reihenfolge der Addierer mit drei Eingängen aufgelöst wird (Abb. M1.8): Den kürzesten kritischen Pfad bekommt man, wenn man zuerst die Multiplikationsergebnisse von  $a_1$  und  $b_1$  addiert und das Zwischenergebnis mit dem Ergebnis der Multiplikationen mit  $a_2$  und  $b_0$  aufsummiert.

In Abb. M1.7c sind die Additionen bereits „baumartig“ zusammengefasst, der kritische Pfad lässt sich daher nicht weiter verkürzen. Der kritische Pfad von Abb. M1.7b ist länger als der von Abb. M1.7d, da hier alle Addierer im rekursiven *und* im transversalen Pfad durchlaufen werden müssen. Bei den transponierten Strukturen sind Addierer und Register verschachtelt, daher hängt die Länge des

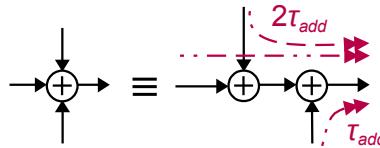


Abb. M1.8: Addierer mit drei Eingängen

kritischen Pfaden in Abb. M1.7e und M1.7f nicht von der Ordnung des Filters ab. Bei Filtern höherer Ordnung machen sich diese Unterschiede besonders stark bemerkbar (Tab. M1.4).

	$N$ allgemein		$N = 2$		$N = 8$	
Typ	$\tau_{mul}$	$\tau_{add}$	$\tau_{krit}$	$f_{S,max}$	$\tau_{krit}$	$f_{S,max}$
a) DF1	1	$N + 1$	21 ns	48 MHz	39 ns	25 MHz
b) DF2	2	$2N$	36 ns	28 MHz	72 ns	14 MHz
c) DF1	1	$N + 1$	21 ns	48 MHz	39 ns	25 MHz
d) DF2	2	$N + 1$	33 ns	30 MHz	51 ns	19 MHz
e) TDF1	1	2	18 ns	56 MHz	18 ns	56 MHz
f) TDF2	2	2	30 ns	33 MHz	30 ns	33 MHz

Tab. M1.4: Maximale Taktfrequenz für verschiedene Filterimplementierungen ( $\tau_{mul} = 12$  ns und  $\tau_{add} = 3$  ns)

Zugrunde liegt aber immer die gleiche Übertragungsfunktion mit identischen Pol- und Nullstellen und Frequenzgang (-> Kap. 2)!!

Bei allen Strukturen müssen 5 Multiplikationen und 4 Additionen pro Inputsample berechnet werden. Unterschiede zwischen verschiedenen Implementierungen können sich dadurch ergeben, dass Symmetrie oder spezielle Koeffizientenwerte ausgenutzt werden (linearphasige Filter, Halbbandfilter -> Kap. 2 und 5) oder durch Änderung der Samplerate im Filter (Kap. 8 - 10). In letzterem Fall können sich auch die Anzahl der Input- und Outputsamples pro Zeiteinheit unterscheiden.

## M1.6 Kritischer Pfad und Ressourcenverbrauch 2 → A1.6

- a) Die Implementierung in Abb. 1.6a benötigt 1 Register, 1 Multiplizierer und 3 Addierer. Die maximale Verzögerungszeit von zwei Additionen und einer Multiplikation tritt an zwei Pfaden auf und ist

$$\tau_{krit,a} = \tau_{mul} + 2\tau_{add} = 12 \text{ ns} + 6 \text{ ns} = 18 \text{ ns} \Rightarrow f_{S,max} = 56 \text{ MHz}.$$

- b) Die Implementierung des Systems in Abb. 1.6b ist ein rekursives System zweiter Ordnung, benötigt aber nur einen Multiplizierer, da die übrigen Koeffizienten Werte von 0, -1 oder 2 haben und sich ohne Multiplizierer implementieren lassen. Die maximale Verzögerungszeit von zwei Additionen und einer Multiplikation tritt im Rückkopplungspfad auf:

$$\tau_{krit,b} = \tau_{mul} + 2\tau_{add} = 12 \text{ ns} + 6 \text{ ns} = 18 \text{ ns} \Rightarrow f_{S,max} = 56 \text{ MHz}$$

Die Struktur in Abb. 1.6b erhält man durch Transponieren einer DF2-Struktur, sie wird daher *transponierte Direktform 2* (TDF2) genannt. Da hier Register und Addierer abwechseln, hat die TDF2 Struktur kurze kritische Pfade und ist dementsprechend schnell.

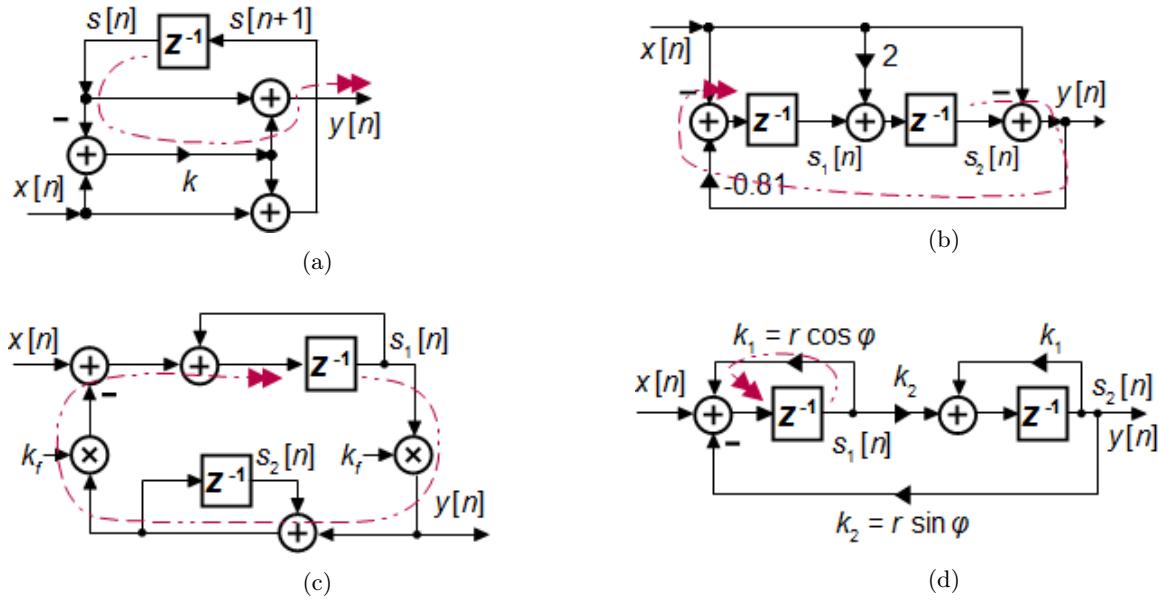


Abb. M1.9: Kritische Pfade zu Aufgabe M1.6

- c) Die Implementierung des Systems in Abb. 1.6c benötigt 2 Register, 2 Multiplizierer und 3 Addierer. Der kritische Pfad geht über von drei Additionen und zwei Multiplikationen:

$$\tau_{krit,c} = 2\tau_{mul} + 3\tau_{add} = 24 \text{ ns} + 9 \text{ ns} = 33 \text{ ns} \Rightarrow f_{S,max} = 30 \text{ MHz}$$

- d) Die Implementierung der Struktur Abb. 1.6d benötigt 2 Register, 4 Multiplizierer und 3 Addierer. Die maximale Verzögerungszeit von zwei Additionen und einer Multiplikation tritt im Rückkopplungspfad auf. Der Addierer mit den drei Eingängen muss dabei als zwei normale Addierer gerechnet werden:

$$\tau_{krit,d} = 1\tau_{mul} + 2\tau_{add} = 12 \text{ ns} + 6 \text{ ns} = 18 \text{ ns} \Rightarrow f_{S,max} = 56 \text{ MHz}$$

## M1.7 Transponierte Systeme zu Aufgabe 1.6 → A1.7

Die transponierten Formen der Systeme aus Abb. 1.6 sind in Abb. M1.10 dargestellt. Die „Spiegelung“ der Darstellung um den gewohnten Signalfluss von links nach rechts zu erhalten, ist z.B. in der Prüfung nicht erforderlich!

Der kritische Pfad des transponierten Systems unterscheidet sich meist von dem des ursprünglichen Systems, Impulsantwort, Übertragungsfunktion, P/N-Diagramm sind bei idealer Arithmetik (keine Quantisierung) aber gleich. Auch die benötigten Hardwareressourcen sind identisch (siehe auch Hinweise zu Aufgabe M1.5). Auch bei dieser Aufgabe sind die Anzahl der Abzweigungen und der Addierer der transponierten und der ursprünglichen Strukturen gleich: Bei *einem* Eingang und *einem* Ausgang (single-input, single-output, SISO) ist das zwangsläufig so, da alle Abzweigungen wieder zu einem Ausgang zusammengeführt werden müssen. Hat das System eine unterschiedliche Zahl von Ein- und Ausgängen (multiple-in, multiple-out, MIMO), kann sich die Anzahl der Abzweigungen und der Addierer unterscheiden.

## M1.8 Allgemeine Fragen zu FPGAs → A1.8

- a) **Vor- und Nachteile von FPGAs:** Im Vergleich zu ASICs sind die Einmalkosten (Non-Recurring Expense oder Engineering, NRE) bei FPGAs deutlich günstiger, da die Maskenkosten

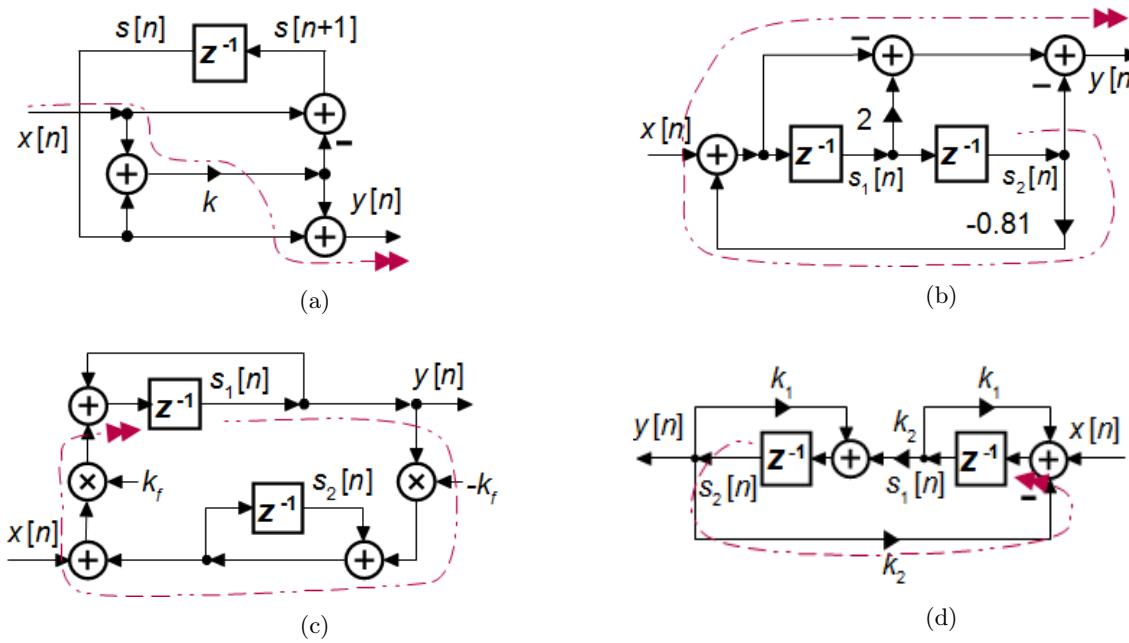


Abb. M1.10: Transponierte Filterstrukturen zu Abb. 1.6 mit kritischen Pfaden

entfallen und auch die Entwicklungskosten im Allgemeinen dank optimierter Software geringer sind. Aufgrund der noch höheren Flexibilität (optimierte Technologie und Hardware) lassen sich dafür mit ASICs geringere Chipfläche und Verlustleistung (bei gleicher Technologie) sowie noch höhere Performance erzielen. Bei Stückzahlen  $< 100.000$  sind dennoch FPGAs bei vergleichbarer Performance meist deutlich günstiger.

Im Vergleich zu DSPs und uCs sind FPGAs deutlich flexibler (beliebige Wortlängen, bei Bedarf Parallelisierung für extrem hohe Datenraten, verschiedene Taktraten auf dem gleichen Chip) und ermöglichen passgenaue Kompromisse zwischen Datenrate und Leistungsverbrauch. DSPs und uC sind für Standardanwendungen leichter zu entwickeln und im mittleren Performancesegment auch meist kostengünstiger.

- b) **Hardmakros:** Speziell Multiplizierer und Speicher benötigen viel Fläche wenn sie aus Standard FPGA-Blöcken synthetisiert werden. Da beide Komponenten für schnelle Signalverarbeitung immer benötigt werden, wird meist eine gewisse Anzahl layoutoptimiert als Hardmakros im FPGA-Fabric verteilt. Hardmakros sind zwar deutlich schneller, kompakter und verlustleistungsärmer als synthetisierte Softmakros, können aber nicht anderweitig genutzt werden und liegen brach, falls nicht alle Multiplizierer und Speicherblöcke benötigt werden.

#### c) FPGA-Technologien:

**Standard-CMOS:** Kann in der jeweils aktuellsten Technologie gefertigt werden, ist daher am schnellsten (oder am kostengünstigsten, wenn ältere Technologien verwendet werden). Nachteile: Das Programmierfile wird in SRAM-Zellen gespeichert, beim Abschalten der Stromversorgung verlieren Speicherzellen den Inhalt und müssen beim nächsten Power-Up aus externem PROM neu geladen werden -> Auslesen des Programmierfiles möglich, FPGA-Ports sind beim Einschalten u.U. in undefiniertem Zustand. Weiterer Nachteil: Leckströme in SRAM-Zellen erhöhen statische Verlustleistung.

**Flash:** Programmierinformationen werden in On-Chip Flash-Speicher abgelegt und stehen beim Einschalten sofort zur Verfügung. Nachteil: Komplexere Technologie erhöht Kosten, kleine-

re Strukturgrößen stehen immer erst ca. 1 1/2 Jahre später als bei Standard-CMOS zur Verfügung.

**Antifuse:** Radiation-hard (Programmierbits können durch ionisierende Strahlung nicht gekippt werden, Nachteile: In Technologien < 180 nm nicht möglich, keine Neuentwicklungen mehr, nur einmal programmierbar. Seit 2010 werden daher zunehmend rad-hard FPGAs in Flash-Technologie angeboten, die Ausfälle durch redundante Strukturen verhindern sollen. Hierzu werden z.B. Rechnungen in drei identischen Rechenwerken durchgeführt und das richtige Ergebnisse durch Mehrheitsentscheid ausgewählt.



## M2 LTF: LTI-Systeme im Frequenzbereich

### M2.1 Filterung des Sensorsignals im Frequenzbereich → A2.1

Simulationen und weitere Infos finden Sie im Notebook **LTF-Sensor**.

#### a) Komplexer Frequenzgang und Betragsgang

##### MA-Filter der Ordnung 3 und 4

Zur Bestimmung des Frequenzgangs muss  $H(z)$  entlang des Einheitskreises berechnet werden, man muss also  $z = e^{j\Omega}$  setzen:

$$H_{MA4}(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4}$$

$$\Rightarrow H_{MA4}(z = e^{j\Omega}) = 1 + e^{-j\Omega} + e^{-2j\Omega} + e^{-3j\Omega} + e^{-4j\Omega} \quad (\text{M2.1})$$

Diese Darstellung des Frequenzgangs ist zwar korrekt, lässt sich aber kaum interpretieren. Vor allem kann man den Betragsfrequenzgang  $|H(z = e^{j\Omega})|$  aus dieser Form nicht ohne weiteres bestimmen. Wenn wie in diesem Fall die Koeffizienten von  $z^{-i}$  symmetrisch zu einer gedachten Mittellinie liegen, gibt es einen einfachen Trick: Man klammert zunächst den Faktor  $z^{-N/2}$  aus  $H(z)$  aus (wobei  $N$  der Grad des Polynoms ist), so dass auch die Exponenten von  $z$  symmetrisch zur Mittellinie liegen („Spiegelpolynom“):

$$H_{MA4}(z) = z^{-2} \left( z^2 + z^1 + \overleftrightarrow{1} + z^{-1} + z^{-2} \right)$$

$$\Rightarrow H_{MA4}(e^{j\Omega}) = e^{-2j\Omega} \left( e^{2j\Omega} + e^{j\Omega} + \overleftrightarrow{1} + e^{-j\Omega} + e^{-2j\Omega} \right)$$

$$= e^{-2j\Omega} \left( \underbrace{e^{2j\Omega} + e^{-2j\Omega}}_{2 \cos 2\Omega} + \underbrace{e^{j\Omega} + e^{-j\Omega}}_{2 \cos \Omega} + 1 \right) \quad (\text{M2.2})$$

Die Terme werden dann paarweise mit (B.3) und (B.4) zu Sinus- oder Cosinusfunktionen (je nach Vorzeichen) zusammengefasst.

$$\Rightarrow H_{MA4}(e^{j\Omega}) = \underbrace{e^{-j2\Omega}}_{\text{lin. Phase}} \underbrace{(2 \cos 2\Omega + 2 \cos \Omega + 1)}_{\text{Amplitudengang + Vorzeichen}} \Rightarrow H(\Omega = 0) = 5 \quad (\text{M2.3})$$

Bei MA-Filtern (vor allem höherer Ordnung) kommt man mit Hilfe der endlichen geometrischen Reihe (B.44) zu einer übersichtlicheren Darstellung von (M2.1) bzw. (M2.3):

$$H_{MA4}(e^{j\Omega}) = \sum_{k=0}^4 e^{-jk\Omega} = \frac{1 - e^{-j5\Omega}}{1 - e^{-j\Omega}} = \frac{e^{-j5\Omega/2}}{e^{-j\Omega/2}} \frac{e^{j5\Omega/2} - e^{-j5\Omega/2}}{e^{j\Omega/2} - e^{-j\Omega/2}}$$

$$= e^{-j2\Omega} \frac{\sin \frac{5\Omega}{2}}{\sin \frac{\Omega}{2}} = 5e^{-j2\Omega} \operatorname{di}_5(\Omega)$$

$$\text{mit } \operatorname{di}_N(\Omega) = \frac{\sin N\Omega/2}{N \sin \Omega/2} \approx \frac{\sin N\Omega/2}{N\Omega/2} = \operatorname{si}(N\Omega/2) \quad (\text{M2.4})$$

$\operatorname{di}_N(\Omega)$  ist der sogenannte *Dirichlet-Kernel* (B.43), auch *periodische si-Funktion* genannt, der auch bei Spektren von zeitdiskreten Rechteck-Pulsen oder -Fenstern auftaucht. Damit ist er das

direkte Äquivalent zur si-Funktion, die das Spektrum von zeitkontinuierlichen Rechteckpulsen beschreibt (C.8).

Aus (M2.3) oder (M2.4) lässt sich jetzt der Betragsgang ableiten:

$$|H_{MA4}(e^{j\Omega})| = |(2 \cos 2\Omega + 2 \cos \Omega + 1)| = \left| \frac{\sin \frac{5\Omega}{2}}{\sin \frac{\Omega}{2}} \right| = \frac{|\sin \frac{5\Omega}{2}|}{|\sin \frac{\Omega}{2}|} = |5 \operatorname{di}_5(\Omega)|$$

Zu analogen Ergebnissen kommt man für das MA-Filter der Ordnung 3:

$$\begin{aligned} H_{MA3}(z) &= z^{-3/2} \left( z^{3/2} + z^{1/2} \xrightarrow{+} z^{-1/2} + z^{-3/2} \right) \\ \Rightarrow H_{MA3}(e^{j\Omega}) &= e^{-3j\Omega/2} \left( e^{3j\Omega/2} + e^{j\Omega/2} \xrightarrow{+} e^{-j\Omega/2} + e^{-3j\Omega/2} \right) \\ &= 2e^{-3j\Omega/2} \left( \cos \frac{3\Omega}{2} + \cos \frac{\Omega}{2} \right) \\ &= \sum_{k=0}^3 e^{-jk\Omega} = e^{-j3\Omega/2} \frac{\sin 2\Omega}{\sin \frac{\Omega}{2}} = 4e^{-j3\Omega/2} \operatorname{di}_4(\Omega) \end{aligned}$$

Auch hier lässt sich leicht die Betragsfunktion ableiten.

### Kaskadiertes MA-Filter

Das kaskadierte MA-Filter kann man ausgehend von der äquivalenten Direktform untersuchen

$$\begin{aligned} H_{casc}(z) &= 1 + 2z^{-1} + 3z^{-2} + 2z^{-3} + z^{-4} \\ \Rightarrow H_{casc}(z = e^{j\Omega}) &= 1 + 2e^{-j\Omega} + 3e^{-2j\Omega} + 2e^{-3j\Omega} + e^{-4j\Omega} \\ &= e^{-j2\Omega} (2 \cos 2\Omega + 4 \cos \Omega + 3), \end{aligned}$$

für manche Analysen ist allerdings die kaskadierte Struktur günstiger:

$$\begin{aligned} H_{casc}(z) &= (1 + z^{-1} + z^{-2})^2 \\ \Rightarrow H_{casc}(z = e^{j\Omega}) &= (1 + e^{-j\Omega} + e^{-2j\Omega})^2 \\ &= (e^{-j\Omega})^2 (e^{j\Omega} + 1 + e^{-j\Omega})^2 = e^{-j2\Omega} (2 \cos \Omega + 1)^2 \end{aligned}$$

Von beiden Darstellungen lässt sich leicht die Betragsfunktion ableiten. Um zu zeigen, dass beide Frequenzgänge identisch sind, benötigt man das Theorem  $\cos 2x = 2 \cos^2 x - 1$ .

### Verlustbehafteter Integrator

Beim verlustbehafteten Integrator kann man den Spiegelpolynomtrick leider nicht anwenden, da das Nennerpolynom nicht symmetrisch ist. Hier kann man die komplexen Exponentialfunktionen nur mit Hilfe der Eulerschen Identität (B.2) auflösen:

$$\begin{aligned} H_{int}(z) &= \frac{1}{1 - \alpha z^{-1}} \Rightarrow H_{int}(z = e^{j\Omega}) = \frac{1}{1 - \alpha e^{-j\Omega}} \\ &= \frac{1}{1 - \alpha \cos \Omega + j\alpha \sin \Omega} \end{aligned}$$

Die Betragsfunktion muss man hier „händisch“ ausrechnen:

$$\begin{aligned} |H_{int}(z = e^{j\Omega})| &= \left| \frac{1}{1 - \alpha \cos \Omega + j\alpha \sin \Omega} \right| = \frac{1}{|1 - \alpha \cos \Omega + j\alpha \sin \Omega|} \\ &= \frac{1}{\sqrt{(1 - \alpha \cos \Omega)^2 + (\alpha \sin \Omega)^2}} \\ &= \frac{1}{\sqrt{1 + \alpha^2 - 2\alpha \cos \Omega}} = \frac{1}{\sqrt{1,81 - 1,8 \cos \Omega}} \quad \text{für } \alpha = 0,9 \end{aligned}$$

wobei die Identität  $\cos^2 \Omega + \sin^2 \Omega = 1$  ausgenutzt wurde. Man sieht, dass die Betragsfunktion monoton abnimmt wenn  $\Omega$  von  $0 \rightarrow \pi$  geht.

Die DC-Verstärkung für  $\alpha = 0,9$  kann man mit der unendlichen Reihe (B.41)

$\sum_{i=0}^{\infty} 0,9^i = \frac{1}{1-0,9} = 10$  ermitteln; notfalls kann man auch Terme aufsummieren und nach einer endlichen Zahl Summanden abbrechen.

### b) DC-Verstärkung

Die DC-Verstärkung eines Systems lässt sich sowohl direkt aus  $H(e^{j\Omega})$  als auch aus  $H(z)$  bestimmen. Allerdings werden bei der Dirichlet-Form von MA-Systemen bei  $\Omega = 0$  Zähler und Nenner 0, der Funktionswert muss dann über die Regel von L'Hospital berechnet werden:

$$H_{MA4}(\Omega = 0) = \lim_{\Omega \rightarrow 0} \frac{\sin 5\Omega/2}{\sin \Omega/2} = \left. \frac{(\sin 5\Omega/2)'}{(\sin \Omega/2)'} \right|_{\Omega=0} = \left. \frac{\frac{5\Omega}{2} \cos \frac{5\Omega}{2}}{\frac{\Omega}{2} \cos \frac{\Omega}{2}} \right|_{\Omega=0} = 5$$

Analog dazu berechnet man  $H_{MA3}(\Omega = 0) = 4$ ,  $H_{casc}(\Omega = 0) = 9$ ,  $H_{int}(\Omega = 0) = 10$ .

**Wichtiger Trick:** Zu den gleichen Ergebnissen kommt man auch schneller, wenn man weiß, dass  $H(\Omega = 0) = H(z = 1)$  ist und einfach  $z = 1$  in die jeweilige Systemfunktion einsetzt.

### c) Dämpfung bei $f = 50$ Hz

Die Dämpfung des Systems bei einer bestimmten Frequenz  $\Omega_1$  könnte man wieder direkt aus  $H(e^{j\Omega_1})$  bzw.  $H(z)$  bestimmen. Allerdings ist hier  $z = e^{j\Omega_1}$  im Allgemeinen komplex, was die Rechnung erschwert.

Auf jeden Fall muss aus der Signalfrequenz  $f_1$  (die hier die Frequenz des Störsignals ist) die normierte Frequenz  $F_1 = f_1/f_S$  bzw. die normierte Kreisfrequenz  $\Omega_1 = 2\pi F_1 = 2\pi f_1/f_S$  berechnet werden:

Hier:  $f = 50$  Hz,  $f_S = 200$  Hz  $\Rightarrow F_1 = 50/200 = 0,25$  und  $\Omega_1 = 2\pi F_1 = \pi/2$ .

Mit den Ergebnissen von Unterpunkt a) erhält man:

$$\begin{aligned} |H_{MA4}(\Omega_1)| &= |2 \cos 2\Omega_1 + 2 \cos \Omega_1 + 1| \\ &= |2 \cos \pi + 2 \cos \pi/2 + 1| = |2 \cdot (-1) + 2 \cdot 0 + 1| = 1 \\ \Rightarrow |H_{MA4}(\Omega_1)| / |H_{MA4}(\Omega = 0)| &= 0,2 \end{aligned}$$

Analog dazu:

$$\begin{aligned} |H_{MA3}(\Omega_1)| &= 2 \left| \cos \frac{3\Omega_1}{2} + \cos \frac{\Omega_1}{2} \right| = 2 \left| -\frac{1}{\sqrt{2}} + \frac{1}{\sqrt{2}} \right| = 0 \\ |H_{casc}(\Omega_1)| &= (2 \cos \Omega_1 + 1)^2 = 1 \\ |H_{int}(\Omega_1)| &= \frac{1}{\sqrt{1,81 - 1,8 \cos \Omega_1}} = 0,743 \end{aligned}$$

### d) Pol- und Nullstellenplan

Zur Bestimmung des P/N-Plans müssen (nicht weiter überraschend) zunächst die Pole und Nullstellen der Systemfunktionen bestimmt werden. Analytisch ist die Zerlegung eines Polynoms  $N$ -ter Ordnung in seine Linearfaktoren im Allgemeinen nicht möglich; die Beispiele dieser Aufgabe lassen sich aber noch geschlossen lösen. Zur Bestimmung der Pole und Nullstellen wandelt man am  $H(z)$  in die Form mit positiven Exponenten um. Wenn man eine oder mehrere Nullstellen kennt oder erraten kann, kann man die Ordnung des Polynoms durch Polynomdivision

$f_S(\text{Hz})$	25	200	240	$50\pi$
$\mathbf{F}_1$	0	1/4	5/24	1/ $\pi$
$\Omega_1$	0	$\pi/2$	$5\pi/12$	2
$ H_{MA,4} (\Omega_1)$	—	0,2	0,043	—
$ H_{MA,3} (\Omega_1)$	—	0	0,21	—
$ H_{casc} (\Omega_1)$	—	0,11	0,26	—
$ H_{int} (\Omega_1)$	—	0,074	0,086	—

Tab. M2.1: Dämpfungen der verschiedenen Filter aus Aufgabe M2.1 bei  $f_1 = 50$  Hz bezogen auf  $|H(\Omega = 0)|$  bei unterschiedlichen Abtastfrequenzen

reduzieren. MA-Filter kann man durch Umformung der endlichen geometrischen Reihe (B.42) in eine Form bringen, aus der man sofort Null- und Polstellen erkennen kann.

### MA-Filter der Ordnung 4

Bei diesem Filter kann man Nullstellen nur schwer erraten, dafür lässt es sich bequem als endliche geometrische Reihe darstellen und umformen:

$$H_{MA4}(z) = 1 + z^{-1} + z^{-2} + z^{-3} + z^{-4} = \sum_{k=0}^{N=4} z^{-k} = \frac{1 - z^{-5}}{1 - z^{-1}} = \frac{z^5 - 1}{z^4(z - 1)}$$

$$\Rightarrow z_{0,k} = e^{2\pi k/5} \text{ für } k = 0 \dots 4 \text{ und } z_{\infty,0} = 1 \text{ sowie } z_{\infty,1\dots 4} = 0$$

Die Nullstelle  $z_{0,0} = 1$  und die Polstelle  $z_{\infty,0} = 1$  heben sich gegenseitig auf.

Für die Auflösung des Zählers wurde benutzt  $z^N = 1 = e^{2k\pi} \Leftrightarrow z = e^{j2k\pi/N}$ .

### MA-Filter der Ordnung 3

Auch diese Systemfunktion lässt sich über die endliche geometrische Reihe umformen wie im letzten Fall (versuchen Sie es!), hier kann alternativ die Nullstelle bei  $z = -1$  erraten,  $H_{MA3}(z = -1) = 0$ . Durch Polynomdivision durch  $(z + 1)$  erhält man ein Polynom zweiter Ordnung, das man leicht in Linearfaktoren zerlegen kann:

$$\begin{aligned} H_{MA3}(z) &= 1 + z^{-1} + z^{-2} + z^{-3} && \text{mit} \\ &= \frac{z^3 + z^2 + z^1 + 1}{z^3} && \left( \begin{array}{r} z^3 + z^2 + z + 1 \\ - z^3 - z^2 \\ \hline z + 1 \end{array} \right) \div (z + 1) = z^2 + 1 \\ &= \frac{(z^2 + 1)(z + 1)}{z^3} && \left( \begin{array}{r} - z - 1 \\ \hline 0 \end{array} \right) \\ \Rightarrow z_{0,0} &= -1; z_{0,1} = +j; z_{0,2} = -j; && \\ z_{\infty,0\dots 2} &= 0 && \end{aligned}$$

### Kaskadiertes MA-Filter

Wie bei der Ermittlung des Betragsgangs geht man auch für die Bestimmung der Pol- und Nullstellen am einfachsten von der kaskadierten Form aus und löst die quadratischen Terme mit

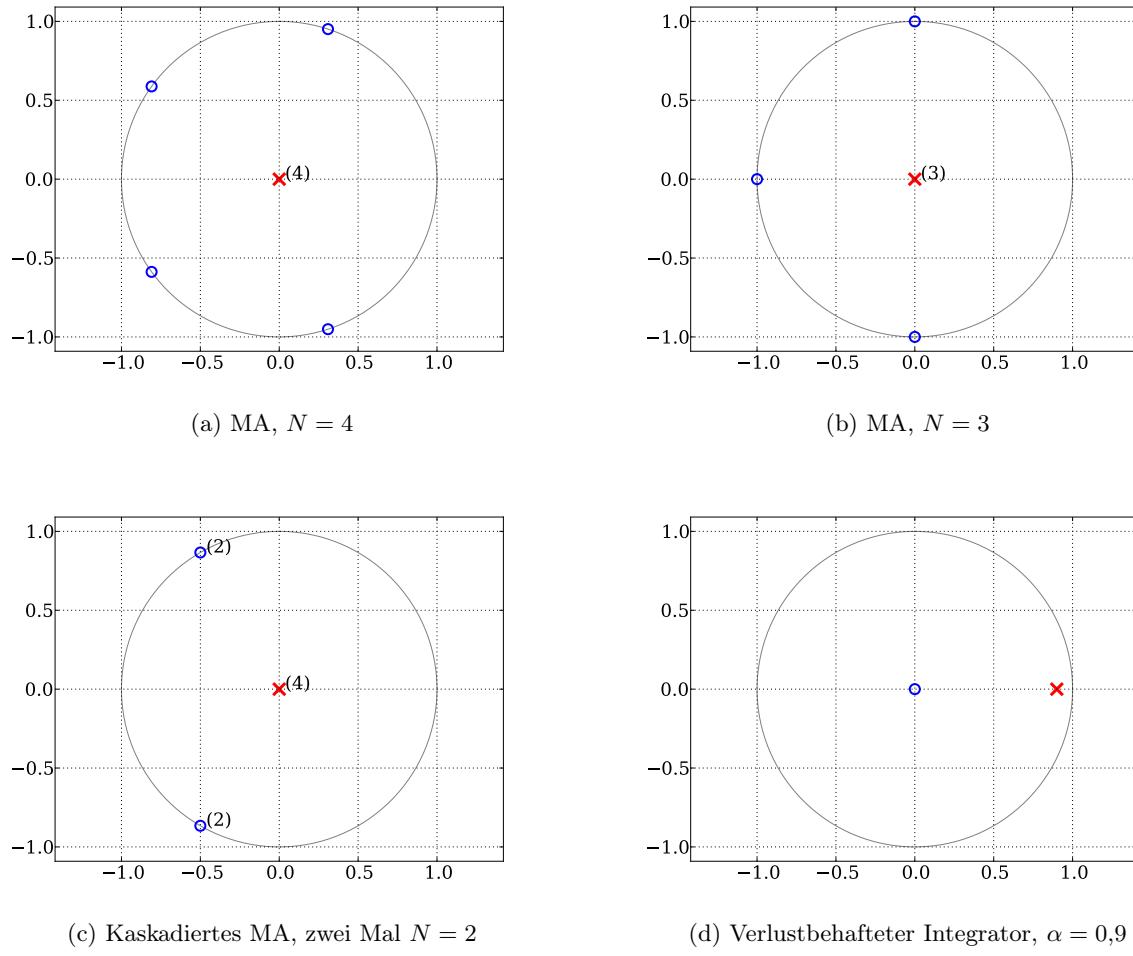


Abb. M2.1: PN-Diagramme zu Aufgabe 2.1

(B.39) auf:

$$\begin{aligned}
 H_{casc}(z) &= (1 + z^{-1} + z^{-2})^2 = \frac{(z^2 + z + 1)^2}{z^4} \\
 &= \frac{(z + 0,5 \pm j\sqrt{3}/2)^2}{z^4} \\
 \Rightarrow z_{0,0,1} &= -0,5 + j\sqrt{3}/2; \quad z_{0,2,3} = -0,5 - j\sqrt{3}/2; \quad z_{\infty,0...3} = 0
 \end{aligned}$$

**Integrator mit Dämpfung**

Den Pol- / Nullstellenplan ermittelt man durch „genaues Hinschauen“ auf

$$H_{int}(z) = \frac{1}{1 - \alpha z^{-1}} = \frac{z}{z - \alpha} \Rightarrow z_0 = 0, \quad z_{\infty} = \alpha = 0,9$$

### e) Phasengang und Gruppenlaufzeit

Aus (M2.4) lässt sich auch der Phasengang des **MA-Filters der Ordnung 4** ableiten:

$$\begin{aligned}\varphi_{MA4}(e^{j\Omega}) &= \angle e^{-j2\Omega} \frac{\sin \frac{5\Omega}{2}}{\sin \frac{\Omega}{2}} = -2\Omega + \angle \sin \frac{5\Omega}{2} - \angle \sin \frac{\Omega}{2} \\ &= \underbrace{-2\Omega}_{\text{lin. Phase}} + \underbrace{\frac{\pi}{2} \left( \left( 1 - \text{sgn} \sin \frac{5\Omega}{2} \right) - \left( 1 - \text{sgn} \sin \frac{\Omega}{2} \right) \right)}_{\text{Vorzeichen des Amplitudengangs}} \\ &= \begin{cases} -2\Omega & \text{für } 0 \leq \Omega < 2\pi/5 & \text{linearphasig mit} \\ \pi - 2\Omega & \text{für } 2\pi/5 < \Omega \leq 4\pi/5 & \text{Phasensprüngen um } \pi \\ 2\pi - 2\Omega & \text{für } 4\pi/5 < \Omega \leq 6\pi/5 & \text{bei } \Omega = 2\pi/5, \\ 3\pi - 2\Omega & \text{für } 6\pi/5 < \Omega \leq 8\pi/5 & \Omega = 4\pi/5, 6\pi/5, \\ 4\pi - 2\Omega & \text{für } 8\pi/5 < \Omega \leq 2\pi & \text{und bei } \Omega = 8\pi/5 \\ \dots & & \end{cases} \end{aligned}$$

Da die Phase der reellwertigen Sinusfunktion nur 0 oder  $\pi$  sein kann, kann man sie mit Hilfe der Signum- oder Vorzeichenfunktion angeben:

$$\text{sgn}(x) = \begin{cases} -1 & \text{für } x < 0, \\ 0 & \text{für } x = 0, \\ +1 & \text{für } x > 0 \end{cases} \quad (\text{M2.5})$$

Allerdings ist diese Beschreibung relativ umständlich zu lesen, übersichtlicher ist die abschnittsweise Angabe des Phasengangs. Dabei ist es Geschmackssache, ob man die Phasensprünge wie hier aufaddiert oder ob man z.B. zwischen 0 und  $\pi$  abwechselt - 0,  $2\pi$  und  $4\pi$  lassen sich auf dem Einheitskreis schließlich nicht unterscheiden.

Um die Phasenverschiebung bei  $f_1 = 50$  Hz und  $f_{S,2} = 200$  Hz herauszubekommen, muss man nur  $\Omega_1 = 2\pi f_1 / f_{S,2} = \pi/2$  in obige Gleichung einsetzen:

$$\varphi_{MA4}(\Omega_1 = \pi/2) = \pi - 2\Omega_1 = 0$$

Bei dieser Frequenz hat das Ausgangssignal also die gleiche Phasenlage wie das Eingangssignal. Bei vier Frequenzen wechselt das Vorzeichen des Amplitudenterms (Nullstelle auf dem EK), bei diesen Frequenzen ändert sich die Phasendrehung um  $\pm\pi$ . Ob man  $+\pi$  oder  $-\pi$  annimmt, ist egal, da Unterschiede von  $2\pi$  auf dem EK nicht sichtbar sind. Da die Phase an dieser Stelle springt, drängt sich die Frage auf: Welche Phase hat man *genau* an dieser Stelle? Die Antwort lautet: egal, da die Amplitude an dieser Stelle Null ist ...

Bei  $\Omega = 0$  gibt es keinen Phasensprung - dort wechseln gleichzeitig Zähler- und Nennerfunktion ihr Vorzeichen (Nullstelle und Polstelle heben sich auf).

Die **Gruppenlaufzeit** ist die Ableitung der Phase nach  $\partial\omega$  bzw.  $\partial\Omega$ . Vereinfacht kann man sich vorstellen, dass die Gruppenlaufzeit beschreibt, um welche Zeit ein Pulspaket im Filter verzögert wird. Bei linearphasigen Systemen ist die Gruppenlaufzeit unabhängig von der Frequenz (= wie bei einem Verzögerungsglied); sie ändern nicht die grundsätzliche Impulsform (= keine Dispersion) und sind daher gut geeignet für die Filterung von amplituden- und phasenmodulierten Signalen. Linearphasige Filter sind immer symmetrisch zur Filtermitte aufgebaut, hier entspricht die Gruppenlaufzeit immer der Laufzeit bis zur (gedachten) Filtermitte. Bei einem linearphasigen FIR-Filter der Ordnung  $N$  ( $\hat{N}$  Verzögerungen) ist die Gruppenlaufzeit daher immer  $\tau_g =$

**NT<sub>S</sub>/2.** An den Stellen, an denen die Phase springt, bereitet die Berechnung der Gruppenlaufzeit oft numerische Probleme (Ausreißer im Plot).

$$\begin{aligned}\tau_g(e^{j\Omega}) &= -\frac{\partial \varphi_{MA4}(e^{j\Omega})}{\partial \omega} = -T_S \frac{\partial \varphi}{\partial \Omega} = -\frac{1}{2\pi} \frac{\partial \varphi}{\partial f} \\ \Rightarrow \tau_{g,MA4}(e^{j\Omega}) &= 2T_S = \frac{2}{200 \text{ Hz}} = 10 \text{ ms}\end{aligned}$$

Zu ähnlichen Ergebnissen kommt man für den **MA-Filter der Ordnung 3**:

$$\varphi_{MA3}(e^{j\Omega}) = \begin{cases} -\frac{3}{2}\Omega & \text{für } 0 \leq \Omega < \pi/2 \quad \text{linearphasig mit} \\ \pi - \frac{3}{2}\Omega & \text{für } \pi/2 < \Omega \leq \pi \quad \text{Phasensprung um } \pi \text{ bei } \Omega = \pi/2 \\ \dots \end{cases}$$

Bei  $f_1 = 50 \text{ Hz} \hat{=} \Omega_1 = \pi/2$  ist der Betragsgang 0 und die Phase nicht definiert. Bei geringfügig kleineren Frequenzen beträgt die Phasendrehung  $\varphi_{MA3}(\Omega_{1-}) = -3\Omega_{1-}/2 = -3\pi/4$ , bei geringfügig höheren Frequenzen  $\varphi_{MA3}(\Omega_{1+}) = \pi - 3\Omega_{1+}/2 = \pi/4$ .

$$\Rightarrow \tau_{g,MA3}(e^{j\Omega}) = \frac{3}{2}T_S = 7,5 \text{ ms}$$

Beim **kaskadierten MA-Filter** ist die Berechnung der Phase noch einfacher, da hier das Vorzeichen der Amplitudenfunktion nicht wechselt (doppelte Nullstellen):

$$\varphi_{kask}(e^{j\Omega}) = -2\Omega \Rightarrow \tau_{g,kask}(e^{j\Omega}) = 2T_S = 10 \text{ ms}$$

Achtung: In Abb. M2.2c springt die Phase nur scheinbar um  $2\pi$  - das liegt nur an der Phasendarstellung des Plot-Programms, das die Phase auf einen Bereich von  $\pm\pi$  umbrikt (wrapped).

Beim **verlustbehafteten Integrator** ist die Rechnung komplizierter, da das System nicht linearphasig ist:

$$\begin{aligned}\varphi_{int}(e^{j\Omega}) &= -\angle \frac{1}{1 - \alpha \cos \Omega + j\alpha \sin \Omega} = \angle 1 - \angle(1 - \alpha \cos \Omega + j\alpha \sin \Omega) \\ &= 0 - \text{atan}2 \frac{\alpha \sin \Omega}{1 - \alpha \cos \Omega} = -\arctan \frac{\alpha \sin \Omega}{1 - \alpha \cos \Omega}\end{aligned}$$

$$\Rightarrow \varphi_{int}(\Omega_1 = \pi/2) = -\arctan \alpha$$

$$\begin{aligned}\Rightarrow \tau_{g,int}(e^{j\Omega}) &= -\frac{\partial \varphi_{int}(\Omega)}{\partial \Omega} = \partial \left( \arctan \frac{\alpha \sin \Omega}{1 - \alpha \cos \Omega} \right) / \partial \Omega \\ &= -\frac{-\left( \frac{\alpha \sin \Omega}{1 - \alpha \cos \Omega} \right)^2 + \frac{\alpha \cos \Omega}{1 - \alpha \cos \Omega}}{\left( \frac{\alpha \sin \Omega}{1 - \alpha \cos \Omega} \right)^2 + 1}\end{aligned}$$

## M2.2 Frequenzgang einfacher FIR-Filter → A2.2

a) Null- und Polstellen ermittelt man am besten aus der Systemfunktion mit positiven Exponenten:

$$H(z) = 0,5 + z^{-1} = \frac{0,5z + 1}{z} = 0,5 \frac{z + 2}{z} \Rightarrow z_{0,1} = -2, z_{\infty,1} = 0$$

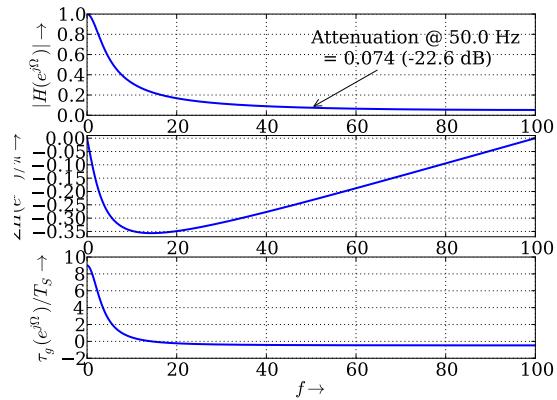
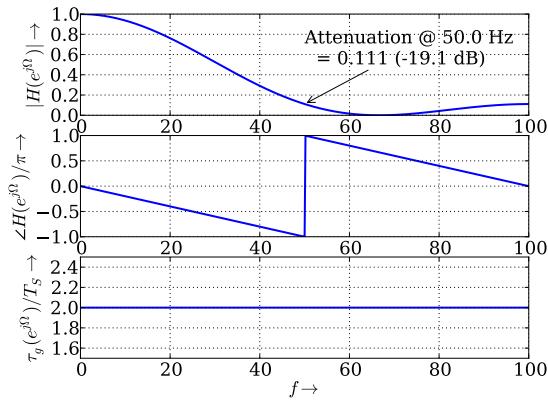
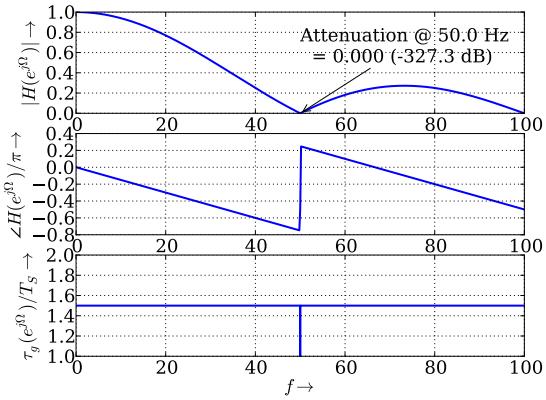
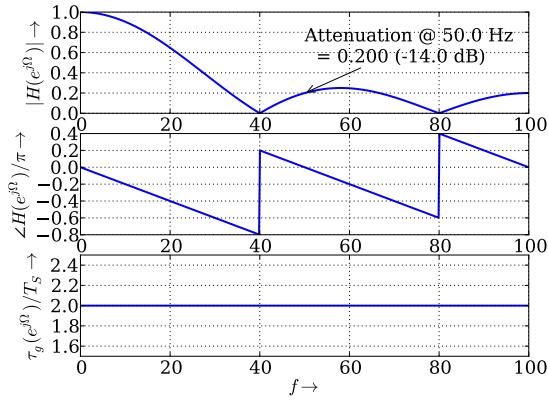


Abb. M2.2: Betrags- und Phasengang sowie Gruppenlaufzeit zu Aufgabe 2.1

Mit Null- und Polstellen kann man sofort die Systemfunktion in Produktform angeben

$$H(z) = 0,5 \frac{z - (-2)}{z - 0}$$

obwohl das bei einem System erster Ordnung wie hier nicht besonders viel neue Erkenntnisse liefert. Der P/N-Plan ist dargestellt in Abb. M2.3.

- b) Den Frequenzgang bei physikalischen Frequenzen erhält man entlang des Einheitskreises, d.h. man wertet die Systemfunktion nur aus bei  $z = e^{j\Omega} = e^{j2\pi f/f_s}$ . Hier startet am einfachsten mit der Systemfunktion mit negativen Exponenten (möglichst wenige Terme) und erhält den *komplexen* Frequenzgang<sup>1</sup> durch einfache Ersetzung:

$$\begin{aligned} H(z) &= 0,5 + z^{-1} \\ \Rightarrow H(z = e^{j2\pi f}) &= 0,5 + e^{-j2\pi f/f_s} = 0,5 + \cos 2\pi f/f_s - j \sin 2\pi f/f_s \end{aligned}$$

Hierfür wurde die Eulersche Identität benutzt,

$$e^{\pm j\Omega} = \cos \Omega \pm j \sin \Omega$$

<sup>1</sup>Wie fast immer in dieser Vorlesung, werden komplexe Größen nicht gesondert gekennzeichnet, da im Frequenzbereich die meisten Resultate komplex sind.

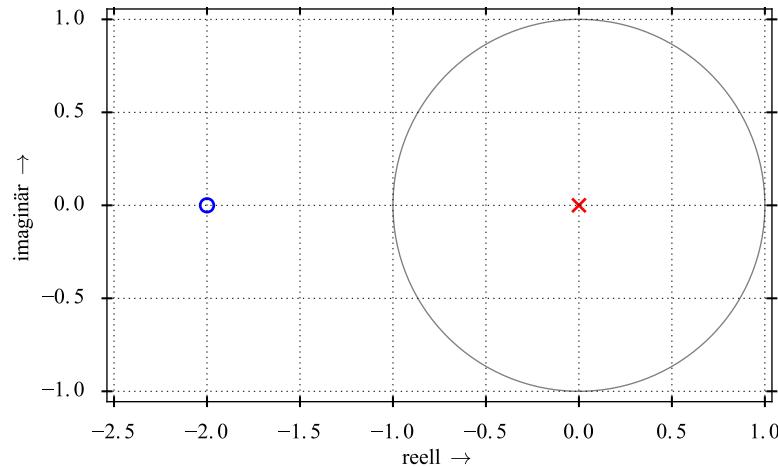


Abb. M2.3: Pol/Nullstellenplan zu Aufgabe 2.2

Allerdings ist der komplexe Frequenzgang für Humanoide nicht besonders hilfreich, Betrags- und Phasengang kann man sich viel leichter vorstellen und auch darstellen:

$$\begin{aligned} |H(z = e^{j2\pi f})| &= \sqrt{\Re^2 \{H(e^{j2\pi f})\} + \Im^2 \{H(e^{j2\pi f})\}} \\ &= \sqrt{(0,5 + \cos 2\pi f/f_s)^2 + (-\sin 2\pi f/f_s)^2} \\ &= \sqrt{0,25 + \cos(\cdot) + \cos^2(\cdot) + \sin^2(\cdot)} = \sqrt{1,25 + \cos 2\pi f/f_s} \end{aligned}$$

Für die Phasenberechnung reicht die arctan - Funktion nicht aus, da sie nur für Argumente zwischen  $0 \dots \pi$  eindeutige Ergebnisse liefert, hier muss die atan2 Funktion (siehe Abb. B.1) verwendet werden!

$$\angle H(e^{j2\pi f}) = \text{atan2} \frac{\Im \{H(e^{j2\pi f})\}}{\Re \{H(e^{j2\pi f})\}} = \text{atan2} \frac{-\sin 2\pi f/f_s}{0,5 + \cos 2\pi f/f_s}$$

Man sieht, dass der Phasengang hier nicht-linear ist (siehe auch Abb. M2.4).

Um Betrags- und Phasengang an einzelnen Frequenzen zu berechnen, muss man die Frequenz nur in die obigen Gleichungen einsetzen:

$$\begin{aligned} f = 0 : \quad |H(f = 0)| &= \sqrt{1,25 + 1} = 1,5 & \angle H(f = 0) &= \text{atan2} \frac{0}{+0,5} = 0^\circ \\ f = \frac{f_s}{4} : \quad \left|H\left(f = \frac{f_s}{4}\right)\right| &= \sqrt{1,25 + 0} = 1,118 & \angle H\left(f = \frac{f_s}{4}\right) &= \text{atan2} \frac{-1}{0,5} = 63,43^\circ \\ f = \frac{f_s}{2} : \quad \left|H\left(f = \frac{f_s}{2}\right)\right| &= \sqrt{1,25 - 1} = 0,5 & \angle H\left(f = \frac{f_s}{2}\right) &= \text{atan2} \frac{0}{-0,5} = -180^\circ \end{aligned}$$

Für die speziellen Frequenzen in dieser Aufgabe ist es aber einfacher direkt die entsprechenden Werte für  $z$  in  $H(z)$  einzusetzen:

$$\begin{aligned} f = 0 &\hat{=} z = 0 & \Rightarrow & H(f = 0) = H(z = 1) = 1,5 \\ f = \frac{f_s}{4} &\hat{=} z = j & \Rightarrow & H\left(f = \frac{f_s}{4}\right) = H(z = j) = 0,5 - j \\ f = \frac{f_s}{2} &\hat{=} z = -1 & \Rightarrow & H\left(f = \frac{f_s}{2}\right) = H(z = -1) = -0,5 \end{aligned}$$

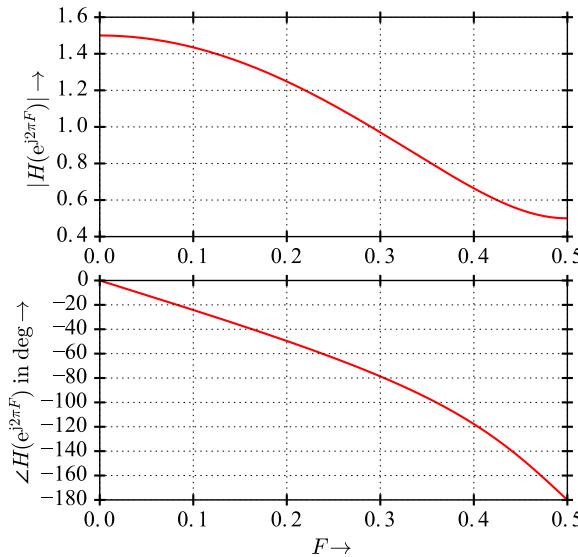


Abb. M2.4: Betrags- und Phasengang zu Aufgabe 2.2

### M2.3 Systemfunktion aus Impulsantwort → A2.3

Alle Systeme dieser Aufgabe lassen sich durch ein transversales System in Direktform implementieren, dessen Koeffizienten auch die Systemfunktion liefern,  $H(z) = \sum h[i]z^{-i}$ .

- a) Das System hat TP-Charakteristik: Dies wird bestätigt durch  $H(z = 1) = H(f = 0) = 1$  und  $H(z = -1) = H(f = f_S/2) = 0$ .
- b) Das System hat HP-Charakteristik:  $H(z = 1) = H(f = 0) = 0$  und  $H(z = -1) = H(f = f_S/2) = 1$ .
- c) Wie beim ersten System ergibt sich eher eine TP-Charakteristik:  $H_C(z = 1) = H_C(f = 0) = 1,5$  und  $H(z = -1) = H(f = f_S/2) = 0,5$ . Hier ist allerdings eine genauere Analyse notwendig.

### M2.4 Systemeigenschaften aus Übertragungsfunktion → A2.4

**System A** ist ein rein transversales System (keine Rückkopplung, Pole nur im Ursprung) und damit stabil. Es ist kausal, da es nur Verzögerungen in der Impulsantwort gibt (bzw. Anzahl der Nullstellen gleich der Anzahl der Polstelle ist,  $N = P$ ). Die Systemfunktion wird für die Bestimmung der Impulsantwort in die Polynomform mit negativen Exponenten umgewandelt:

$$H_A(z) = \frac{(z-1)(z+1)}{z^2} = 1 - z^{-2} \quad \Rightarrow \quad h_b[n] = \delta[n] - \delta[n-2]$$

Aus der Impulsantwort oder der Systemfunktion in Polynomform mit negativen Exponenten können *direkt* die Koeffizienten eines FIR-Filters in *Direktform* abgelesen werden.

Die Nullstellen bei  $z_{0;1,2} = \pm 1$  und die Polstellen bei  $z_{\infty;1,2} = 0$  kann man leicht aus der ursprünglichen Form von  $H_A(z)$  ablesen, sie führen zu Bandpass-Verhalten.

$$\begin{aligned} H_A(z = 1) &= H_A(f = 0) = 0, \quad H_A(z = j) = H_A(f = f_S/4) = 2 \\ H_A(z = -1) &= H_A(f = f_S/2) = 0. \end{aligned}$$

Der komplexe Frequenzgang wird entlang des Einheitskreises bestimmt, indem man  $z = e^{j\Omega}$  setzt:

$$\begin{aligned} H_A(z = e^{j\Omega}) &= 1 - e^{-2j\Omega} = e^{-j\Omega}(e^{j\Omega} - e^{-j\Omega}) = 2je^{-j\Omega} \sin \Omega \\ \Rightarrow |H_A(z = e^{j\Omega})| &= |2 \sin \Omega| \\ \Rightarrow \angle H_A(z = e^{j\Omega}) &= \pi/2 - \Omega \end{aligned}$$

Durch Ausklammern von  $e^{-2j\Omega}$  entsteht hier ein symmetrisches Polynom („Spiegelpolygone“), das sich mit (B.3) zu einer Sinusfunktion zusammenfassen lässt. Die Bestimmung des Phasengangs ist hier besonders einfach, da die Sinusfunktion im Intervall  $\Omega = 0 \dots \pi$  das Vorzeichen nicht ändert, so dass dort die Phase nur durch den Vorfaktor  $je^{-j\Omega} = e^{j(\pi/2-\Omega)}$  bestimmt wird. Die Phase hat einen konstanten Anteil  $\pi/2$  und ist ansonsten proportional zur Frequenz  $\Omega$ , solche Systeme nennt man *linearphasig*. Nur bei Systemen mit symmetrischen Koeffizienten lassen sich alle Exponentialterme zu Sinus- und Kosinusfunktionen zusammenfassen lassen, darum können nur solche Systeme linearphasig sein!

**System B** ist stabil (rein transversal, Pole nur im Ursprung) und kausal (nur Verzögerungen in Impulsantwort bzw. Anzahl der Nullstellen kleiner als Anzahl der Polstellen,  $N < P$ ). Die Impulsantwort kann direkt aus der gegebenen Systemfunktion abgelesen werden:

$$H_B(z) = z^{-1} + z^{-3} \Rightarrow h_a[n] = \delta[n-1] + \delta[n-3]$$

Auch hier kann man aus den Koeffizienten bzw. der Impulsantwort  $\{1; 0; -1\}$  ein Filter in Direktform ableiten

Aus der Impulsantwort oder der Systemfunktion in Polynomform mit negativen Exponenten können *direkt* die Koeffizienten eines FIR-Filters in *Direktform* abgelesen werden.

Das System hat eine zusätzliche Verzögerung (Latenz) von einer Sampleperiode, die entfernt werden könnte, ohne den Betragsgang zu beeinflussen.

Bestimmung von Pol- und Nullstellen aus der Polynomform mit positiven Exponenten (Erweitern mit  $z^3$ ):

$$H_B(z) = \frac{z^2 + 1}{z^3} \Rightarrow z_{\infty;1,2,3} = 0; z_{0;1,2} = \pm j$$

Nullstellen bei  $z = \pm j$  erzeugen Bandsperre bei  $f = f_S/4$ .

$$\begin{aligned} H_B(z=1) &= H_B(f=0) = 2 \\ H_B(z=j) &= H_B(f=f_S/4) = 0 \\ H_B(z=-1) &= H_B(f=f_S/2) = -2. \end{aligned}$$

Der komplexe Frequenzgang wird wie bei System A bestimmt:

$$\begin{aligned} H_B(z = e^{j\Omega}) &= e^{-j\Omega} + e^{-3j\Omega} = e^{-2j\Omega}(e^{j\Omega} + e^{-j\Omega}) = 2e^{-2j\Omega} \cos \Omega \\ \Rightarrow |H_B(z = e^{j\Omega})| &= |2 \cos \Omega| \\ \Rightarrow \angle H_B(z = e^{j\Omega}) &= \begin{cases} -2\Omega & \text{für } 0 \leq \Omega < \pi/2 & \text{linearphasig mit} \\ \pi - 2\Omega & \text{für } \pi/2 < \Omega \leq \pi & \text{Phasensprüngen um } \pi \end{cases} \end{aligned}$$

Der Betragsgang lässt sich hier bequem bestimmen, indem man  $e^{-2j\Omega}$  ausklammert. Dadurch entsteht ein symmetrisches Polynom, dessen Terme sich zu einer Kosinusfunktion zusammenfassen lassen. Bei der Bestimmung des Phasengangs muss man beachten, dass die Kosinusfunktion bei  $\Omega = \pi/2$  das Vorzeichen wechselt, daher springt die Phase dort um  $\pi$ .

**System C** ist nicht kausal (nur „Verfrühungen“, keine Verzögerungen bzw.  $N = 3 > P = 0$ ), kann aber durch Verzögerung um drei Samples (Multiplikation mit  $z^{-3}$ ) in ein kausales System  $H'_C$  umgewandelt werden:

$$\begin{aligned} H_C(z) &= z^3 - 3z^2 + 3z - 1 \Rightarrow h_c[n] = \delta[n+3] - 3\delta[n+2] + 3\delta[n+1] - \delta[n] \\ H'_C(z) &= H_C(z)z^{-3} = (1 - z^{-1})^3 = 1 - 3z^{-1} + 3z^{-2} - z^{-3} \\ \Rightarrow h'_c[n] &= \delta[n] - 3\delta[n-1] + 3\delta[n-2] - \delta[n-3] \end{aligned}$$

Das System ist stabil (rein transversal, Pole nur im Ursprung) und hat eine dreifache Nullstelle bei  $z = 1$ , es unterdrückt daher sehr stark Gleichanteile (Hochpass). Die Hardwareimplementierung ergibt sich aus der Impulsantwort des verzögerten Systems,  $h'_C = \{1; -3; 3; -1\}$ . Den Frequenzgang erhält man genau wie bei den vorigen Systemen:

$$\begin{aligned} H'_C(z=1) &= H'_C(f=0) = 0 \\ H'_C(z=j) &= H'_C(f=f_S/4) = 1 + 3j - 3 - j = -2(1+j) = 2\sqrt{2}e^{j3\pi/4} \\ H'_C(z=-1) &= H'_C(f=f_S/2) = 8 \\ H'_C(z=e^{j\Omega}) &= 1 - 3e^{-j\Omega} + 3e^{-2j\Omega} - e^{-3j\Omega} \\ &= e^{-3/2j\Omega} \left( e^{j3\Omega/2} - 3e^{j\Omega/2} + 3e^{-j\Omega/2} - e^{-j3\Omega/2} \right) \\ &= je^{-3/2j\Omega} (2\sin 3\Omega/2 - 6\sin \Omega/2) \\ \Rightarrow |H'_C(z=e^{j\Omega})| &= |2\sin 3\Omega/2 - 6\sin \Omega/2| \\ \Rightarrow \angle H'_C(z=e^{j\Omega}) &= -\pi/2 - 3/2\Omega \quad \text{für } 0 < \Omega \leq \pi \end{aligned}$$

Die Bestimmung des Phasengangs ist hier nicht trivial, man muss wissen, dass  $|6\sin \Omega/2| \geq |2\sin 3\Omega/2|$  im ganzen betrachteten Intervall gilt und sich daher das Vorzeichen nicht ändert.

**System D** ist rekursiv, daher muss zur Untersuchung der Stabilität zunächst die Lage der Polstellen bestimmt werden:

$$H_D(z) = \frac{1}{z^2 + 0,64} = \frac{1}{(z + 0,8j)(z - 0,8j)}$$

Die Polstellen bei  $\pm 0,8j$  liegen innerhalb des Einheitskreises, das System ist daher stabil. Die Polstellen bewirken eine Bandpasscharakteristik (Maximum bei  $f = f_S/4$ ). Wegen  $P = 2 < N = 0$  (die Nullstellen im Unendlichen werden ignoriert) ist das System kausal.

Die Hardwareimplementierung ergibt sich aus der Polynomform mit negativen Exponenten

$$H_D(z) = \frac{z^{-2}}{1 + 0,64z^{-2}}$$

als rein rekursives System mit zwei Verzögerungsgliedern und einem Koeffizienten mit dem Wert -0,64.

Da das System rekursiv ist, kann man die Impulsantwort in geschlossener Form nur über die inverse  $z$ -Transformation berechnen. Iterativ erhält man  $h_D[n] = \{1; 0; -0,64; 0; 0,64^2; 0; -0,64^3; \dots\}$  z.B. durch Betrachten der Hardwareimplementierung.

Der Betragsgang kann hier nur über die Eulersche Identität (B.2) bestimmt werden, für den Phasengang ergibt sich eine komplizierte, nichtlineare Lösung, die hier nicht aufgeführt

wird.

$$\begin{aligned}
 H_D(z = e^{j\Omega}) &= \frac{1}{e^{j2\Omega} + 0,64} = \frac{1}{\cos 2\Omega + j \sin 2\Omega + 0,64} \\
 \Rightarrow |H_D(z = e^{j\Omega})| &= \frac{1}{\sqrt{\cos^2 2\Omega + 0,64^2 + 1,28 \cos 2\Omega + \sin^2 2\Omega}} \\
 &= \frac{1}{\sqrt{1,410 + 1,28 \cos 2\Omega}} \\
 \Rightarrow \angle H_D(z = e^{j\Omega}) &= ...
 \end{aligned}$$

## M2.5 Systemeigenschaften aus Pol-/Nullstellenplan → A2.5

- a) FIR (nur Polstellen im Ursprung), daher stabil, kausal ( $N \leq P$ ). System hat HP-Verhalten und vier Nullstellen im Frequenzband, auch bei  $f = 0$ .
- b) IIR, instabil (Polstellen auf EK), kausal. Um ein stabiles System daraus zu erzeugen, müssen beide Polstellen entfernt oder ein wenig in den EK hineingeschoben werden. Sie erhalten dann in beiden Fällen ein BS-Verhalten.
- c) IIR, stabil (keine Polstellen außerhalb EK) und kausal, Bandpass.
- d) IIR, stabil und acausal. Ein zusätzlicher Pol im Ursprung (Verzögerung um eine Samplingperiode) macht das System kausal ohne den Frequenzgang zu verändern. Das System hat HP-Charakteristik. Da keine Nullstelle auf dem EK liegt, wird der Frequenzgang nirgendwo Null.
- e) IIR, stabil und kausal, ansonsten wie beim vorigen System.
- f) IIR, instabil und kausal. Zwei Pole außerhalb des EK müssen entfernt werden. Da das System dann acausal wird, müssen zwei Pole im Ursprung ergänzt werden. Das resultierende System hat dann HP-Verhalten.

## M2.6 Systemeigenschaften aus Signalflussdiagramm → A2.6

- a) Zum Ermitteln der Übertragungsfunktion  $H_a(z)$  des Systems Abb. 1.6a stellt man am einfachsten zunächst die Gleichungen für  $s[n+1]$  und  $y[n]$  bzw. für  $zS(z)$  und  $y(z)$  auf:

$$\begin{aligned}
 s[n+1] &= -ks[n] + (1+k)x[n] \quad \circlearrowleft \bullet \quad S(z)z^{+1} = -kS(z) + (1+k)X(z) \\
 y[n] &= (1-k)s[n] + kx[n] \quad \circlearrowleft \bullet \quad Y(z) = (1-k)S(z) + kX(z)
 \end{aligned} \tag{M2.6}$$

Man erhält daraus die Übertragungsfunktion des Systems (und kann daraus auch direkt die Größen des Zustandsraumsystems ablesen).

$$\begin{aligned}
 S(z)z + kS(z) &= (1+k)X(z) \quad \Rightarrow \quad S(z) = X(z) \frac{1+k}{z+k} \\
 \Rightarrow Y(z) &= X(z)(1-k) \frac{1+k}{z+k} + kX(z) = X(z) \frac{1-k^2 + kz + k^2}{z+k} \\
 \Rightarrow H_a(z) &= \frac{1+kz}{z+k} = \frac{z^{-1} + k}{kz^{-1} + 1} \Rightarrow z_\infty = -k; z_0 = -\frac{1}{k}
 \end{aligned}$$

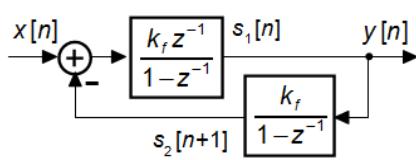
Das System hat eine Pol- und eine Nullstelle, die symmetrisch zum Einheitskreis sind ( $z_0 = z_\infty^{-1}$ ) und damit eine *Allpass*-Charakteristik.

- b) Die Übertragungsfunktion des Systems in Abb. 1.6b lautet:

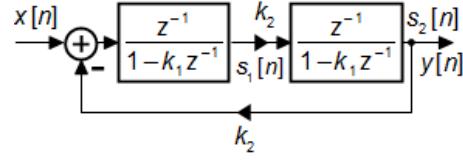
$$H_b(z) = \frac{-z^{-2} + 2z^{-1} - 1}{0,81z^{-2} + 1} = -\frac{1 - 2z + z^2}{0,81 + z^2} \Rightarrow z_{\infty,1,2} = \pm 0,9j; z_{0,1,2} = 1$$

Das System hat eine doppelte Nullstelle bei  $z = 1$  ( $f = 0$ ) und je eine Polstelle in der Nähe von  $z = \pm j$  ( $f = \pm f_S/4$ ). Aufgrund der konjugiert komplexen Polstellen ( $z_{\infty,1} = z_{\infty,2}^*$ ) ist das System reellwertig mit Hochpass-Charakteristik (doppelte Nullstelle bei DC).

- c) Die Übertragungsfunktion des Systems in Abb. 1.6c lässt sich am leichtesten bestimmen, wenn man zunächst die Übertragungsfunktionen der beiden Integratoren / Akkumulatoren mit Multiplikator bestimmt,  $H_{I,1}(z) = k_f z^{-1} / (1 - z^{-1}) = k_f / (z - 1)$  und  $H_{I,2}(z) = k_f / (1 - z^{-1})$ , und daraus dann die gesamte Übertragungsfunktion:



(a) System c



(b) System d

$$\begin{aligned} Y(z) &= X(z)H_{I,1}(z) - Y(z)H_{I,2}(z)H_{I,1}(z) \\ &= X(z) \frac{H_{I,1}(z)}{1 + H_{I,1}(z)H_{I,2}(z)} = \frac{\frac{k_f z^{-1}}{1 - z^{-1}}}{1 + \frac{k_f z^{-1}}{1 - z^{-1}} \frac{k_f}{1 - z^{-1}}} \\ \Rightarrow H_c(z) &= \frac{Y(z)}{X(z)} = \frac{k_f z^{-1} (1 - z^{-1})}{1 - \left(2 - k_f^2\right) z^{-1} + z^{-2}} \end{aligned}$$

Die Pole von  $H_c$  sind:

$$\begin{aligned} z_{\infty 1,2} &= 1 - \frac{k_f^2}{2} \pm \frac{\sqrt{\left(2 - k_f^2\right)^2 - 4}}{2} = 1 - \frac{k_f^2}{2} \pm j\sqrt{k_f^2 - k_f^4/4} \\ |z_{\infty 1,2}| &= \sqrt{\left(1 - \frac{k_f^2}{2}\right)^2 + k_f^2 - k_f^4/4} = 1 \\ \phi_{\infty 1,2} &= \arg(z_{\infty 1,2}) = \text{atan} 2 \frac{\Im\{z_{\infty 1,2}\}}{\Re\{z_{\infty 1,2}\}} = \pm \arctan \frac{\sqrt{k_f^2 - k_f^4/4}}{1 - \frac{k_f^2}{2}} \\ &= \pm \arctan \frac{k_f \sqrt{1 - k_f^2/4}}{1 - \frac{k_f^2}{2}} \approx \pm \arctan k_f \text{ für } |k_f| \ll 1 \end{aligned}$$

Der Betrag der Pole ist genau 1 (auf dem Einheitskreis), unabhängig von  $k_f$ . Das System lässt sich also prinzipiell als Oszillatoren verwenden, dessen Frequenz über  $k_f$  eingestellt werden kann. In der Praxis führt das unvermeidliche Abschneiden oder Runden des Multiplikationsergebnisses (→ Kap. 6) dazu dass der Pol nicht genau auf dem EK bleibt; die Oszillation stirbt ab oder wird instabil. Daher sind zusätzliche Maßnahmen zur Stabilisierung der Schwingung notwendig (z.B. Integration und Rückführung des Quantisierungsfehlers). Die endliche Wortlänge des Koeffizienten  $k_f$  (Koeffizientenquantisierung) ist bei dieser Struktur übrigens unkritisch: Abweichungen vom idealen Wert führen hier nur zu einer leichten Verschiebung der Resonanzfrequenz.

- d) Das System in Abb. 1.6d findet man in der Literatur unter der Bezeichnung **Gold-Rader-Biquad**. Auch hier empfiehlt es sich zunächst die Teilübertragungsfunktion  $H_{I,d}(z)$  der beiden gedämpften Integratoren und danach die Gesamtübertragungsfunktion mit Hilfe der Blockdarstellung Abb. M2.5b aufzustellen:

$$\begin{aligned}
 H_{I,d}(z) &= \frac{z^{-1}}{1 - k_1 z^{-1}} = \frac{1}{z - k_1} \\
 Y(z) &= (X(z) - k_2 Y(z)) k_2 H_{I,d}^2(z) = X(z) \frac{k_2 H_{I,d}^2(z)}{1 + k_2^2 H_{I,d}^2(z)} \\
 \Rightarrow H_d(z) &= \frac{Y(z)}{X(z)} = \frac{k_2}{H_{I,d}^{-2}(z) + k_2^2} = \frac{k_2}{(z - k_1)^2 + k_2^2} \\
 &= \frac{k_2}{z^2 - 2k_1 z + k_1^2 + k_2^2} = \frac{k_2}{z^2 - 2r \cos \phi z + r^2} = \frac{k_2}{(z - re^{j\phi})(z - re^{-j\phi})} \\
 \Rightarrow z_{\infty 1,2} &= re^{\pm j\phi} = r(\cos \phi \pm j \sin \phi) = k_1 \pm j k_2
 \end{aligned}$$

Tiefergehende Informationen zu verschiedenen Implementierungen von IIR-Systemen zweiter Ordnung (Biquad) finden Sie u.a. in [Zöl05].

## M2.7 Einfaches Moving Average Filter → A2.7

### a) Impulsantwort und Systemfunktion

Durch „Hingucken“ erhält man:

$$h_{MA3}[n] = \{1; 1; 1; 1\} \quad \circlearrowleft \bullet \quad H_{MA3}(z) = 1 + z^{-1} + z^{-2} + z^{-3} = \frac{z^3 + z^2 + z + 1}{z^3}$$

Die Systemfunktion eines Moving Average (MA) - Filters der Ordnung  $N$  (hier = 3) ist eine endliche geometrische Reihe (B.42)  $\sum_k^N a^k$  mit  $a = z^{-1}$  setzt und lässt sich genauso umformen und damit übersichtlicher darstellen, vor allem für MA-Filter höherer Ordnung:

$$\begin{aligned}
 H(z) &= \sum_{k=0}^N z^{-k} = \frac{1 - z^{-(N+1)}}{1 - z^{-1}} = \frac{z^{-(N+1)} - 1}{z^{-1} - 1} \quad \text{für } z \neq 1 \\
 \Rightarrow H_{MA3}(z) &= 1 + z^{-1} + z^{-2} + z^{-3} = \sum_{k=0}^3 z^{-k} = \frac{z^{-4} - 1}{z^{-1} - 1} = \frac{z^4 - 1}{z - 1}
 \end{aligned}$$

### b) DC-Verstärkung und Nullstellen

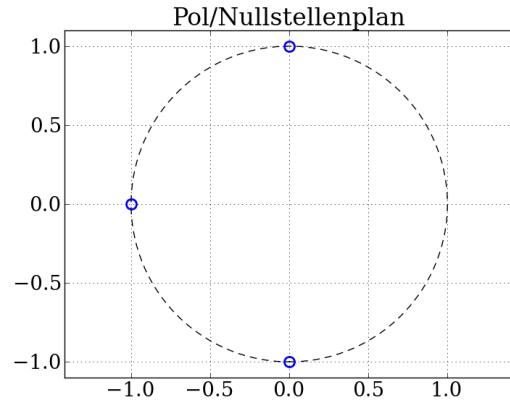
Die DC-Verstärkung erhält man direkt aus der Systemfunktion, indem man setzt

$$H_{MA3}(f = 0) = H(z = 1) = 4.$$

In Abb. 2.3b erkennt man eine Nullstelle bei  $F = 0,5$  bzw.  $f = f_S/2$  oder  $z = -1$ . Testweises Einsetzen in  $H(z)$  bestätigt  $H(z = -1) = 0$ . Die übrigen Nullstellen ermittelt man durch Polynomdivision:

$$\begin{array}{r} (z^3 + z^2 + z + 1) \div (z + 1) = z^2 + 1 \\ \hline -z^3 - z^2 \\ \hline z + 1 \\ -z - 1 \\ \hline 0 \end{array}$$

$$\Rightarrow z_{01} = -1; z_{02} = +j; z_{03} = -j;$$



Natürlich kann man bei dieser Aufgabe die Nullstelle bei  $F = 0,25$  bzw.  $z = j$  auch ohne Rechnung erkennen. Aber Polynomdivision zu üben ist auch nicht schlecht, oder?

### c) Frequenzgang

Zur Bestimmung des Frequenzgangs muss „nur“  $H(z)$  entlang des Einheitskreises berechnet werden, man muss also  $z = e^{j\Omega}$  setzen:

$$H(z) = 1 + z^{-1} + z^{-2} + z^{-3} \Rightarrow H(z = e^{j\Omega}) = 1 + e^{-j\Omega} + e^{-2j\Omega} + e^{-3j\Omega} \quad (\text{M2.7})$$

Diese Darstellung des Frequenzgangs ist zwar korrekt, lässt sich aber kaum interpretieren. Vor allem kann man den Betragsfrequenzgang  $|H(z = e^{j\Omega})|$  aus dieser Form nicht ohne weiteres bestimmen. Wenn wie in diesem Fall die Koeffizienten von  $z^{-i}$  symmetrisch zu einer gedachten Mittellinie liegen, gibt es einen einfachen Trick: Man klammert zunächst aus  $H(z)$  den Faktor  $z^{-N/2}$  aus (wobei  $N$  der Grad des Polynoms ist), so dass auch die Exponenten von  $z$  symmetrisch zur Mittellinie liegen („Spiegelpolynom“):

$$\begin{aligned} H(z) &= z^{-\frac{3}{2}} \left( z^{\frac{3}{2}} + z^{\frac{1}{2}} \xrightarrow{\leftrightarrow} z^{-\frac{1}{2}} + z^{-\frac{3}{2}} \right) \\ \Rightarrow H(e^{j\Omega}) &= e^{-\frac{3}{2}j\Omega} \left( e^{\frac{3}{2}j\Omega} + e^{\frac{1}{2}j\Omega} \xrightarrow{\leftrightarrow} e^{-\frac{1}{2}j\Omega} + e^{-\frac{3}{2}j\Omega} \right) \\ &= e^{-\frac{3}{2}j\Omega} \left( \underbrace{e^{\frac{3}{2}j\Omega} + e^{-\frac{3}{2}j\Omega}}_{2 \cos \frac{3}{2}\Omega} + \underbrace{e^{\frac{1}{2}j\Omega} + e^{-\frac{1}{2}j\Omega}}_{2 \cos \frac{\Omega}{2}} \right) \end{aligned} \quad (\text{M2.8})$$

Die Terme werden dann paarweise mit (B.3) und (B.4) zu Sinus- oder Cosinusfunktionen (je nach Vorzeichen) zusammengefasst.

$$\Rightarrow H(e^{j\Omega}) = 2 \underbrace{e^{-j\frac{3}{2}\Omega}}_{\text{lineare Phase}} \underbrace{\left( \cos \frac{3}{2}\Omega + \cos \frac{1}{2}\Omega \right)}_{\text{Amplitudengang + Vorzeichen}} \Rightarrow H(\Omega = 0) = 4 \quad (\text{M2.9})$$

Aus (M2.9) kann man jetzt den Frequenzgang durch Summation der verschiedenen Sinus- bzw. Cosinusfunktionen konstruieren.

Bei MA-Filtern höherer Ordnung kommt man wieder mit Hilfe der endlichen geometrischen Reihe

(B.42) zu einer übersichtlicheren Darstellung von (M2.7):

$$\begin{aligned} H_{MA3}(e^{j\Omega}) &= \sum_{k=0}^{N=3} e^{-jk\Omega} = \frac{1 - e^{-j4\Omega}}{1 - e^{-j\Omega}} = \frac{e^{-j4\Omega/2}}{e^{-j\Omega/2}} \frac{e^{j2\Omega} - e^{-j2\Omega}}{e^{j\Omega/2} - e^{-j\Omega/2}} \\ &= e^{-j3\Omega/2} \frac{\sin 2\Omega}{\sin \frac{\Omega}{2}} = 4e^{-j3\Omega/2} \operatorname{di}_4(\Omega) \\ \text{mit } \operatorname{di}_L(\Omega) &= \frac{\sin L\Omega/2}{L \sin \Omega/2} \approx \frac{\sin L\Omega/2}{\Omega/2} = \frac{\operatorname{si}(L\Omega/2)}{L} \end{aligned} \quad (\text{M2.10})$$

$\operatorname{di}_L(\Omega)$  ist der sogenannte *Dirichlet-Kernel* (B.43), auch *periodische si-Funktion* genannt, der immer auftaucht, wenn es um das Spektrum von zeit *diskreten* Rechteck-Pulsen oder -Fenstern geht. Damit ist er das direkte Äquivalent zur si-Funktion, die das Spektrum von zeit *kontinuierlichen* Rechteckpulsen beschreibt (C.8). Warum kann die si-Funktion nicht das Spektrum von zeitdiskreten Rechteckpulsen repräsentieren?<sup>2</sup>

Mit (M2.10) lässt sich auch leicht der Betragsgang in einer übersichtlichen Form aufschreiben:

$$|H_{MA3}(e^{j\Omega})| = \left| e^{-j3\Omega/2} \frac{\sin 2\Omega}{\sin \frac{\Omega}{2}} \right| = \left| \frac{\sin 2\Omega}{\sin \frac{\Omega}{2}} \right| = 4 |\operatorname{di}_4(\Omega)|$$

Bei  $\Omega = 0$  werden Zähler und Nenner 0, der Funktionswert an dieser Stelle wird daher über die Regel von L'Hospital berechnet:

$$H_{MA3}(\Omega = 0) = \lim_{\Omega \rightarrow 0} \frac{\sin 2\Omega}{\sin \Omega/2} = \frac{(\sin 2\Omega)'}{(\sin \Omega/2)'} \Big|_{\Omega=0} = \frac{2\Omega \cos 2\Omega}{\frac{\Omega}{2} \cos \frac{\Omega}{2}} \Big|_{\Omega=0} = 4$$

#### d) Phasengang und Gruppenlaufzeit

Aus (M2.9) kann man leicht den Phasengang ermitteln:

$$\varphi(e^{j\Omega}) = \begin{cases} -\frac{3}{2}\Omega & \text{für } 0 \leq \Omega < \pi/2 & \text{linearphasig mit} \\ \pi - \frac{3}{2}\Omega & \text{für } \pi/2 < \Omega \leq \pi & \text{Phasensprung um } \pi \text{ bei } \Omega = \pi/2 \end{cases}$$

Bei  $\Omega = \pi/2$  (Nullstelle auf dem EK) wechselt das Vorzeichen des Sinus-Terms, bei Frequenzen  $\pi/2 < \Omega \leq \pi$  hat man daher eine zusätzliche Phasendrehung von  $\pi$ . Da die Phase bei  $\Omega = \pi/2$  Stelle springt, drängt sich die Frage auf: Welche Phase hat man an dieser Stelle? Die Antwort lautet: eigentlich egal, da die Amplitude an dieser Stelle eh Null ist ...

Die Gruppenlaufzeit ist die Ableitung der Phase nach  $\partial\omega$ :

$$\tau_g(e^{j\Omega}) = -\frac{\partial \varphi}{\partial \omega} = -\frac{-\partial \frac{3}{2}\Omega}{\partial \omega} = \frac{\partial \frac{3}{2}\omega T_S}{\partial \omega} = \frac{3}{2}T_S$$

Bei linearphasigen Filtern wie hier entspricht die Gruppenlaufzeit der Laufzeit bis zur (gedachten) Filtermitte.

#### e) Erweiterung auf $L$ Taps mit $H(f=0) = 1$

Die Systemfunktion eines Moving Average Filters mit  $L$  Taps, also der Ordnung  $N = L - 1$ , und

<sup>2</sup>Spektren zeitdiskreter Signale sind immer periodisch, aber die si-Funktion ist es nicht!

Vorfaktor  $1/L$  lässt sich einfach aufstellen als:

$$\begin{aligned} H(z) &= \frac{1}{L} \sum_{i=0}^{L-1} z^{-i} = \frac{1}{L} \frac{1 - z^{-L}}{1 - z^{-1}} \\ \Rightarrow H(e^{j\Omega}) &= \frac{1}{L} \frac{1 - e^{-jL\Omega}}{1 - e^{-j\Omega}} = \frac{e^{-jL\Omega/2}}{Le^{-j\Omega/2}} \frac{e^{jL\Omega/2} - e^{-jL\Omega/2}}{e^{j\Omega/2} - e^{-j\Omega/2}} = \frac{e^{-j(L-1)\Omega/2}}{L} \frac{\sin L\Omega/2}{\sin \Omega/2} \\ \Rightarrow |H(e^{j\Omega})| &= \frac{1}{L} \left| \frac{\sin L\Omega/2}{\sin \Omega/2} \right| = |\text{di}_L(\Omega)| \end{aligned}$$

Für  $L \rightarrow \infty$  nähert sich  $H(e^{j\Omega})$  immer mehr einer Dirac-Schar an (B.46), was man in Abb. M2.6 für  $L = 32$  bereits gut erkennen kann. Je länger das MA-Filter wird, desto besser filtert es den Gleichanteil des zeitdiskreten Signals heraus. Allerdings steigt gleichermaßen die Gruppenlaufzeit an, das Filter wird immer träger. Für Anwendungen, bei denen ein Signal gemittelt oder geglättet (Suchbegriff: „averaging“ oder „smoothing“) werden soll, gibt es daher eine Vielzahl von Filtern mit unterschiedlichen Kompromissen bzgl. Verzögerungszeit und Frequenzgang.

#### f) Python / Matlab Code für MA-Filter mit $L = 32$

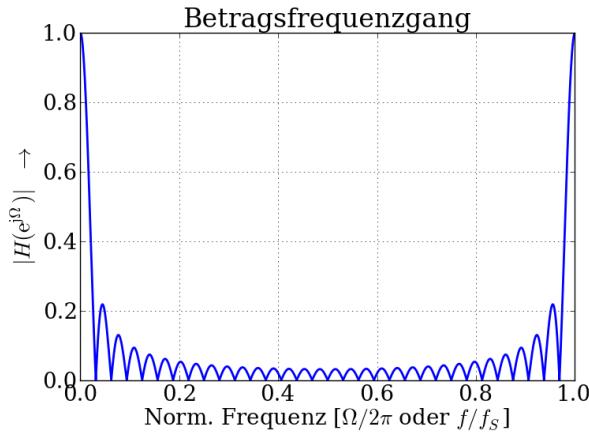
---

```

1 # ... imports wie üblich
2 L = 32                      # Anzahl Taps
3 b = np.ones(1, L)  # (alle Nennerkoeff. = 1)
4 a = L # Zählerkoeff.; hier nur Faktor
5 [w,H] = sig.freqz(b, a, whole=True)
6 # Frequenzgang von F = 0 ... 1 (whole)
7 plt.plot(w /(2*pi), abs(H))
8 grid(True); plt.show()
%
```

---

Lst. M2.1: Python Listing zu M2.7f



```

L = 32;
b = ones(1,L);
a = L;
[w,H]=freqz(b,a);
%
plot(w/(2*pi), abs(H));
grid on;
```

Lst. M2.2: Matlab Listing zu M2.7f

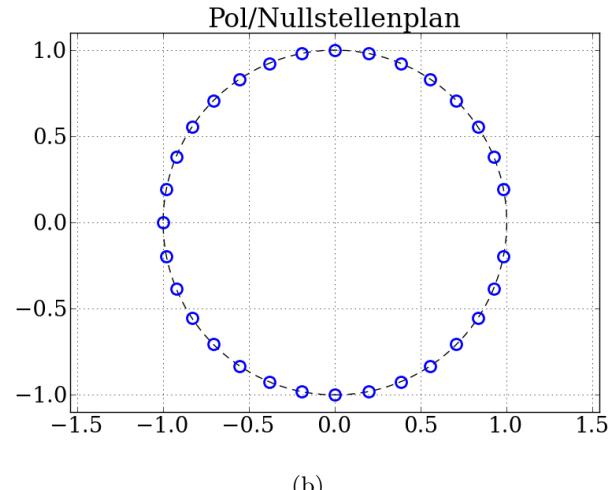


Abb. M2.6: MA-Filter mit  $N = 32$  Taps, (a) Amplitudengang und (b) P/N-Diagramm

## M2.8 Lineарphasiges FIR-Filter → A2.8

- a) Die Filter  $F_1$  und  $F_2$  haben **keine linearen Phasengänge**, da sie weder symmetrische noch antisymmetrische Koeffizienten haben.
- b) **Impulsantworten und Systemfunktionen** erhält man durch Hinschauen:

$$\begin{array}{lll} h_1[n] = \{1; 2; 3\} & \circlearrowleft \bullet & H_1(z) = 1 + 2z^{-1} + 3z^{-2} \\ h_2[n] = \{3; 2; 1\} & \circlearrowleft \bullet & H_2(z) = 3 + 2z^{-1} + z^{-2} \end{array}$$

- c) Die **Nullstellen der Übertragungsfunktionen** erhält man durch Nullsetzen der Systemfunktionen, Erweitern mit  $z^2$  und Lösen der quadratischen Gleichungen (B.4):

$$\begin{aligned} H_1(z): \quad 0 &\stackrel{!}{=} z^2 + 2z + 3 \quad \Rightarrow \quad z_{1,2} = -1 \pm \sqrt{1-3} = -1 \pm \sqrt{2}\mathbf{j} \\ H_2(z): \quad 0 &\stackrel{!}{=} 3z^2 + 2z + 1 \\ &= z^2 + \frac{2}{3}z + \frac{1}{3} \quad \Rightarrow \quad z_{3,4} = -\frac{1}{3} \pm \frac{\sqrt{2}}{3}\mathbf{j} \end{aligned}$$

Abb. M2.7 zeigt die Nullstellen von  $F_1$  und  $F_2$ .

Man sieht, dass die Nullstellen von  $H_1(z)$  und  $H_2(z)$  gleiche Verhältnisse von Real- und Imaginärteil und damit gleiche Winkel haben. Die Beträge sind reziprok zu einander,  $r_{1,2} = 3 = r_{3,4}^{-1}$ .

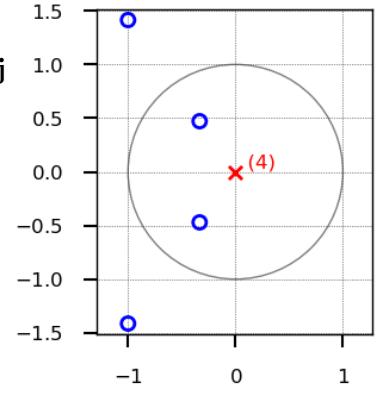


Abb. M2.7

- d) Die **Gesamtimpulsantwort** erhält man mit Hilfe der diskreten Faltung:

$$h[n] = h_1[n] * h_2[n] = \sum h_1[i] \cdot h_2[n-i]$$

In der Frequenzebene ist die Rechnung deutlich einfacher:

$$\begin{aligned} H(z) &= H_1(z) \cdot H_2(z) \\ &= 3 + 8z^{-1} + 14z^{-2} + 8z^{-3} + 3z^{-4} \\ &\bullet \rightsquigarrow \circ \end{aligned}$$

$$\begin{aligned} h[n] &= 3\delta[n] + 8\delta[n-1] + 14\delta[n-2] \\ &\quad + 8\delta[n-3] + 3\delta[n-4] \end{aligned}$$

$n$	0	1	2	3	4
$i$	$h_1[i]$	$h_1[i] \cdot h_2[n-i]$			
0	1	3	2	1	
1	2		6	4	2
2	3			9	6
					3
	$h[n]$	3	8	14	8
					3

- e) Das Gesamtfilter hat eine symmetrische Impulsantwort und ist daher **lineарphasig**. Es hat die Gesamtordnung  $N = 4$ , die **Gruppenlaufzeit** ist daher  $\tau_g = N/2T_S = 2T_S$  (Mitte des Filters).
- f) Da der **Phasengang** linear ist, erhält man ihn leicht über die (hier konstante) Gruppenlaufzeit:

$$\tau_g = -\frac{d\varphi}{d\omega} = -\frac{1}{2\pi} \frac{d\varphi}{df} = 2T_S \Leftrightarrow \varphi = -4\pi f T_S = -4\pi F$$

Im Bereich  $0 \dots f_S/2$  fällt  $\varphi$  daher von 0 auf  $-2\pi$ .

Zum Skizzieren des **Betragsgangs** kann man drei Punkte aus der Systemfunktion leicht berechnen:

$$\begin{aligned} H(f=0) &= H(z=1) = 3 + 8 + 14 + 8 + 3 = \mathbf{36} \\ H\left(f=\frac{f_s}{4}\right) &= H(z=j) = 3 + \frac{8}{j} - 14 - \frac{8}{j} + 3 = -8 \\ H\left(f=\frac{f_s}{2}\right) &= H(z=-1) = 3 - 8 + 14 + 8 + 3 = \mathbf{4} \end{aligned}$$

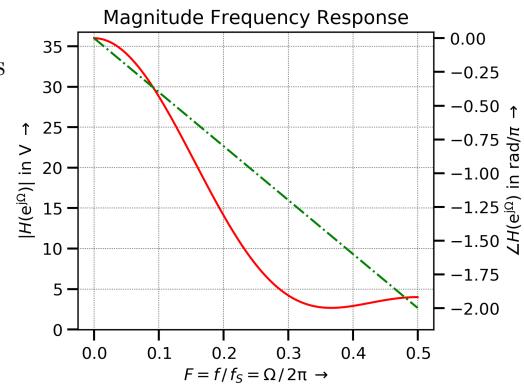


Abb. M2.8: Betrags- und Phasengang

Man sieht, dass die Kombination von zwei nicht-linearphasigen Teiltern insgesamt ein linearphasiges Filter ergeben kann. Das funktioniert immer dann, wenn die Impulsantwort des zweiten Filters die identisch ist mit der des ersten, nur „rückwärts abgespielt“. Darauf beruht auch der **filtfilt** Algorithmus, bei dem ein Filter vorwärts und rückwärts gerechnet wird (Tab. 2.1).

## M2.9 Allgemeine IIR-Struktur → A2.9

Mit Hilfe des Notebooks **LTF-IIR-allgemein** können Sie die Ergebnisse dieser Aufgabe reproduzieren und eigene Experimente anstellen.

### a) Addierer, eine Verzögerung

Aus dem SFG Abb. M2.9 des Systems bestimmt man zunächst  $H_a(z)$ , daraus durch Setzen von

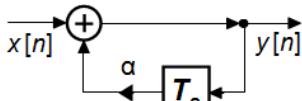


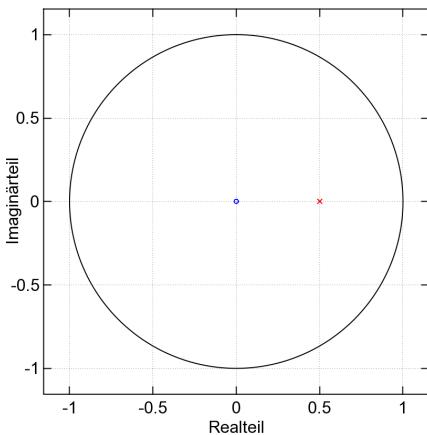
Abb. M2.9: System a)

$$\begin{aligned} H_a(z) &= \frac{1}{1 - \alpha z^{-1}} = \frac{z}{z - \alpha} \Rightarrow z_0 = 0, z_\infty = \alpha \\ \Rightarrow H_a(e^{j\Omega}) &= \frac{1}{1 - \alpha e^{-j\Omega}} = \frac{1}{1 - \alpha \cos \Omega + \alpha j \sin \Omega} \\ \Rightarrow |H_a(e^{j\Omega})| &= \frac{1}{\sqrt{1 + \alpha^2 \cos^2 \Omega - 2\alpha \cos \Omega + \alpha^2 \sin^2 \Omega}} \\ &= \frac{1}{\sqrt{1 + \alpha^2 - 2\alpha \cos \Omega}} \end{aligned}$$

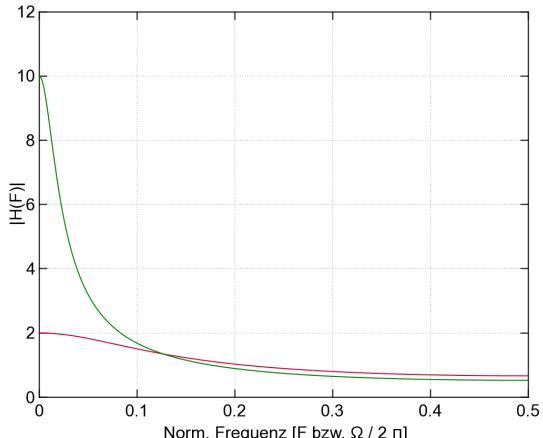
$z = e^{j\Omega}$  den komplexen Frequenzgang  $H_a(e^{j\Omega})$  und daraus schließlich den Amplitudenbetragsgang  $|H_a(e^{j\Omega})|$ . Aus dem P/N Diagramm Abb. M2.10a oder aus  $|H_a(e^{j\Omega})|$  sieht man, dass das Maximum des Amplitudenbetragsgangs bei  $\Omega = 0$  und das Minimum bei  $\Omega = \pi$  auftritt, das System ist ein Tiefpass. Man sieht aus Abb. M2.10, dass für  $\alpha \rightarrow 1$  die Überhöhung bei  $f = 0$  immer stärker wird, gleichzeitig sinkt die Bandbreite. Maximum und Minimum von  $|H_a(e^{j\Omega})|$  lassen sich entweder durch Einsetzen von  $\Omega = 0$  bzw.  $\Omega = \pi$  oder direkt aus  $H_a(z)$  bestimmen:

$$\begin{aligned} |H_a(e^{j\Omega})|_{max} &= |H_a(\Omega = 0)| = |H_a(z = 1)| = \frac{1}{1 - \alpha} = \begin{cases} 2 & \text{für } \alpha = 0,5 \\ 10 & \text{für } \alpha = 0,9 \end{cases} \\ |H_a(e^{j\Omega})|_{min} &= |H_a(\Omega = \pi)| = |H_a(z = -1)| = \frac{1}{1 + \alpha} = \begin{cases} 0,67 & \text{für } \alpha = 0,5 \\ 0,53 & \text{für } \alpha = 0,9 \end{cases} \end{aligned}$$

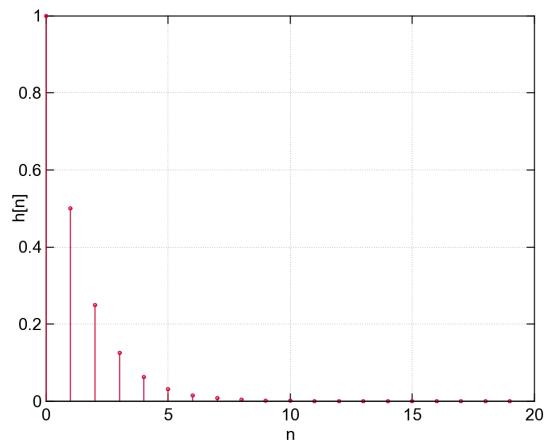
Man sieht aus Abb. M2.11, dass für  $\alpha \rightarrow 1$  die Impulsantwort immer länger andauert und



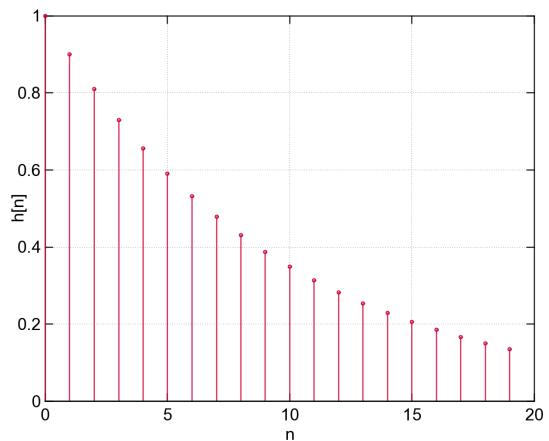
(a)



(b)

Abb. M2.10: Pol-Nullstellen-Plan für  $\alpha = 0,5$  (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9a

(a)



(b)

Abb. M2.11: Impulsantwort für  $\alpha = 0,5$  (a) und  $\alpha = 0,9$  (b) zu Aufgabe 2.9a

ungedämpfter wird. Berechnen lässt sich die Impulsantwort entweder durch „genaues Hinschauen“ direkt aus dem System Abb. M2.9 oder durch inverse  $z$ -Transformation:

$$H_a(z) = \frac{z}{z - \alpha} \quad \bullet \sim \circ \quad h_a[n] = \alpha^n u[n]$$

### b) Subtrahierer, eine Verzögerung

Der Lösungsansatz ist der Gleiche wie bei Unterpunkt a), das resultierende System ist ein

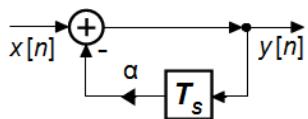
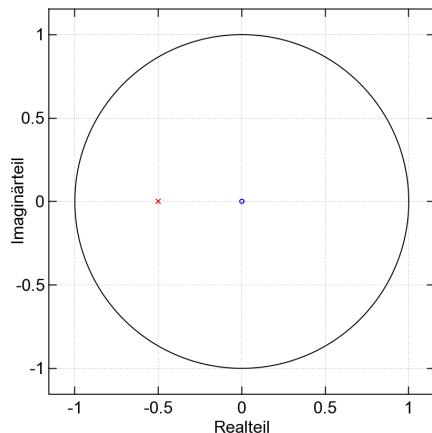


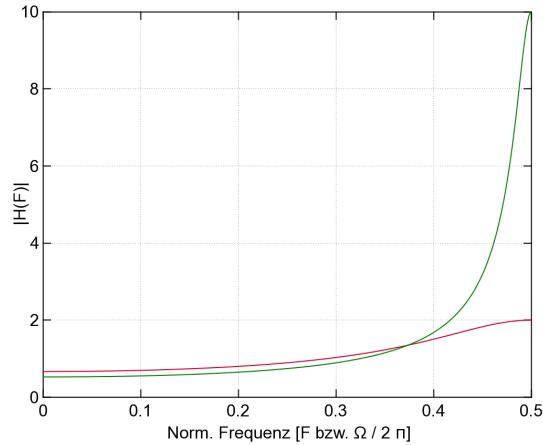
Abb. M2.12: System b)

$$\begin{aligned} H_b(z) &= \frac{1}{1 + \alpha z^{-1}} = \frac{z}{z + \alpha} \Rightarrow z_0 = 0, z_\infty = -\alpha \\ H_b(e^{j\Omega}) &= \frac{1}{1 + \alpha e^{-j\Omega}} = \frac{1}{1 + \alpha \cos \Omega - \alpha j \sin \Omega} \\ |H_b(e^{j\Omega})| &= \frac{1}{\sqrt{1 + \alpha^2 + 2\alpha \cos \Omega}} \end{aligned}$$

Hochpass mit Maximum von  $|H_b(e^{j\Omega})|$  bei  $\Omega = \pi$  und Minimum bei  $\Omega = 0$ .



(a)

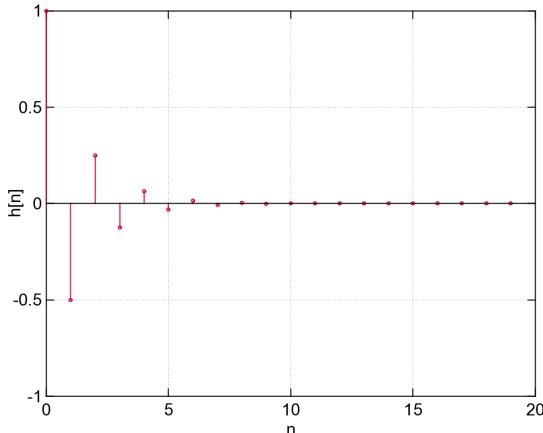


(b)

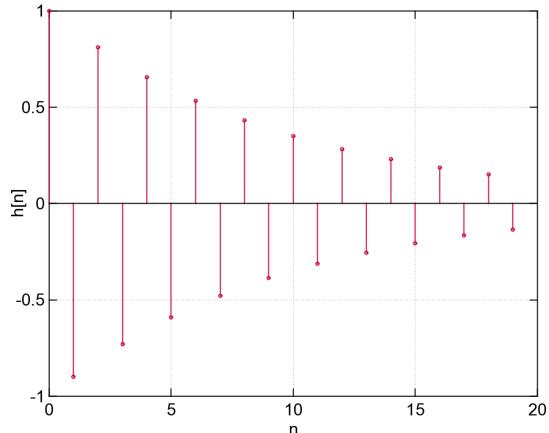
Abb. M2.13: Pol-Nullstellen-Plan (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9b

$$\begin{aligned} |H_b(e^{j\Omega})|_{min} &= |H_b(\Omega = 0)| = |H_b(z = 1)| = \frac{1}{1 + \alpha} = \begin{cases} 0,67 & \text{für } \alpha = 0,5 \\ 0,53 & \text{für } \alpha = 0,9 \end{cases} \\ |H_b(e^{j\Omega})|_{max} &= |H_b(\Omega = \pi)| = |H_b(z = -1)| = \frac{1}{1 - \alpha} = \begin{cases} 2 & \text{für } \alpha = 0,5 \\ 10 & \text{für } \alpha = 0,9 \end{cases} \end{aligned}$$

Im System Abb. M2.12 liegt das Maximum des Frequenzgangs bei  $f_S/2$ , dementsprechend zeigt



(a)



(b)

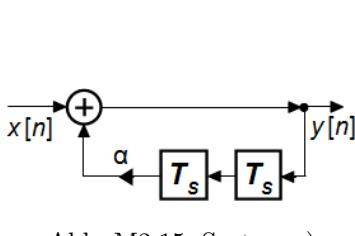
Abb. M2.14: Impulsantwort für  $\alpha = 0,5$  (a) und  $\alpha = 0,9$  (b) zu Aufgabe 2.9b

auch die Impulsantwort eine gedämpfte Oszillation mit  $f_S/2$ . Berechnung wie bei Unterpunkt a):

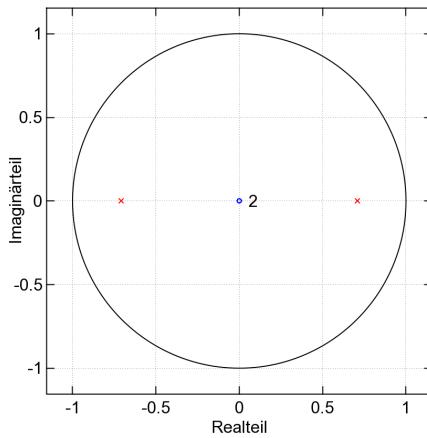
$$H_b(z) = \frac{z}{z + \alpha} \quad \bullet \rightsquigarrow \quad h_b[n] = (-\alpha)^n u[n]$$

### c) Addierer, zwei Verzögerungen

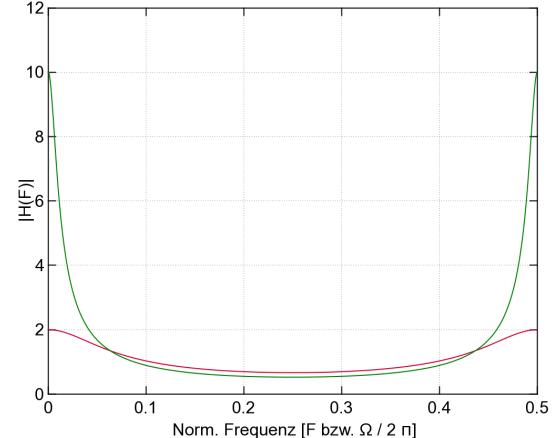
Die Ersetzung von  $T_S$  durch  $2T_S$  (bzw. von  $z^{-1}$  durch  $z^{-2}$ ) wirkt sich im Frequenzgang Abb. M2.16b aus als Transformation  $H(\Omega) \rightarrow H(2\Omega)$ , der Frequenzgang wird also im Vergleich zu Abb. M2.10b gestaucht. Das resultierende System ist eine Bandsperre mit Maxima bei  $\Omega = 0$



$$\begin{aligned}
 H_c(z) &= \frac{1}{1 - \alpha z^{-2}} = \frac{z^2}{z^2 - \alpha} \Rightarrow z_{0,1,2} = 0, z_{\infty,1,2} = \pm\sqrt{\alpha} \\
 H_c(e^{j\Omega}) &= \frac{1}{1 - \alpha e^{-j2\Omega}} = \frac{1}{1 - \alpha \cos 2\Omega + \alpha j \sin 2\Omega} \\
 |H_c(e^{j\Omega})| &= \frac{1}{\sqrt{1 + \alpha^2 \cos^2 2\Omega - 2\alpha \cos 2\Omega + \alpha^2 \sin^2 2\Omega}} \\
 &= \frac{1}{\sqrt{1 + \alpha^2 - 2\alpha \cos 2\Omega}}
 \end{aligned}$$



(a)



(b)

Abb. M2.16: Pol-Nullstellen-Plan für  $\alpha = 0,5$  (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9c

und  $\Omega = \pi$  und einem Minimum bei  $\Omega = \pi/2$ . Maximum und Minimum von  $|H_c(e^{j\Omega})|$  lassen sich bestimmen wie bei a) und b):

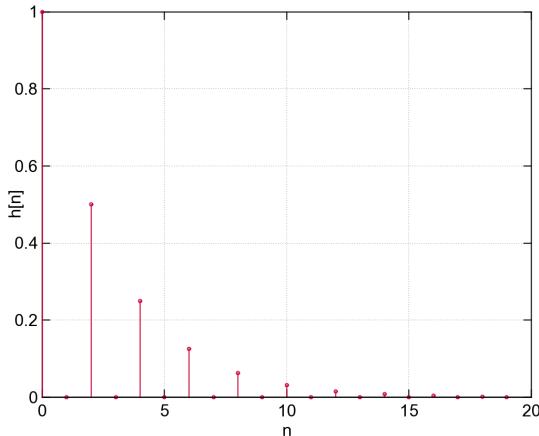
$$\begin{aligned}
 |H_c(e^{j\Omega})|_{max} &= |H_c(\Omega = 0 \text{ bzw. } \pi)| = |H_c(z = \pm 1)| = \frac{1}{1 - \alpha} = \begin{cases} 2 & \text{für } \alpha = 0,5 \\ 10 & \text{für } \alpha = 0,9 \end{cases} \\
 |H_c(e^{j\Omega})|_{min} &= |H_c(\Omega = \pi/2)| = |H_c(z = j)| = \frac{1}{1 + \alpha} = \begin{cases} 0,67 & \text{für } \alpha = 0,5 \\ 0,53 & \text{für } \alpha = 0,9 \end{cases}
 \end{aligned}$$

In der Impulsantwort Abb. M2.17 erzeugt die zweite Verzögerung (Transformation  $z^{-1} \rightarrow z^{-2}$ ) zusätzliche Samples mit dem Wert Null im Vergleich zu Abb. M2.11. Im Punkt e) wird die Impulsantwort berechnet (nicht prüfungsrelevant).

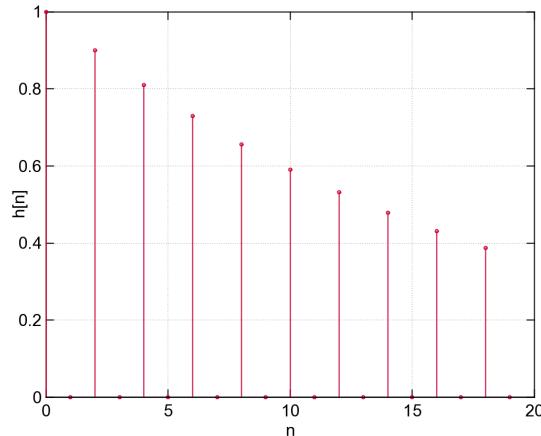
#### d) Subtrahierer, zwei Verzögerungen

Auch hier wird der Frequenzgang gestaucht, das resultierende System ist ein Bandpass mit Maximum von  $|H_d(e^{j\Omega})|$  bei  $\Omega = \pi/2$  und Minima  $\Omega = 0$  und  $\Omega = \pi$ .

$$\begin{aligned}
 |H_d(e^{j\Omega})|_{max} &= |H_d(\Omega = \pi/2)| = |H_d(z = j)| = \frac{1}{1 - \alpha} = \begin{cases} 2 & \text{für } \alpha = 0,5 \\ 10 & \text{für } \alpha = 0,9 \end{cases} \\
 |H_d(e^{j\Omega})|_{min} &= |H_d(\Omega = 0 \text{ bzw. } \pi)| = |H_d(z = \pm 1)| = \frac{1}{1 + \alpha} = \begin{cases} 0,67 & \text{für } \alpha = 0,5 \\ 0,53 & \text{für } \alpha = 0,9 \end{cases}
 \end{aligned}$$



(a)



(b)

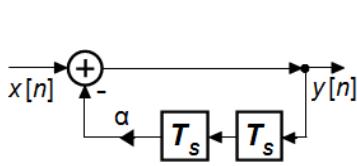
Abb. M2.17: Impulsantwort für  $\alpha = 0,5$  (a) und  $\alpha = 0,9$  (b) zu Aufgabe 2.9c

Abb. M2.18: System d)

$$\begin{aligned}
 H_d(z) &= \frac{1}{1 + \alpha z^{-2}} = \frac{z^2}{z^2 + \alpha} \Rightarrow z_{0,1,2} = 0, z_{\infty,1,2} = \pm j\sqrt{\alpha} \\
 H_d(e^{j\Omega}) &= \frac{1}{1 + \alpha e^{-j2\Omega}} = \frac{1}{1 + \alpha \cos 2\Omega - \alpha j \sin 2\Omega} \\
 |H_d(e^{j\Omega})| &= \frac{1}{\sqrt{1 + \alpha^2 \cos^2 2\Omega + 2\alpha \cos 2\Omega + \alpha^2 \sin^2 2\Omega}} \\
 &= \frac{1}{\sqrt{1 + \alpha^2 + 2\alpha \cos 2\Omega}}
 \end{aligned}$$

**Zusatz: Berechnung der Impulsantwort für System c) und d)** (nicht prüfungsrelevant)

Die Impulsantwort von System c) (Abb. M2.15) und d) (Abb. M2.15) kann durch Partialbruchzerlegung bestimmt werden. Dabei werden nacheinander die folgenden Schritte durchgeführt:

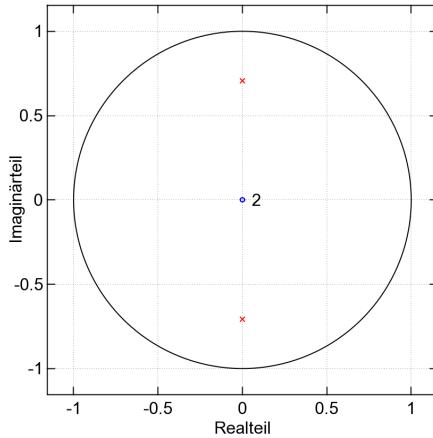
**Normalisierung:** Wenn der Zählergrad  $N$  größer oder gleich dem Nennergrad  $P$  ist<sup>3</sup>, muss der Bruch zunächst durch eine Polynomdivision in die normalisierte Form gebracht werden.

**Bestimmung der Pole:** Der Nenner muss in Produktform dargestellt werden.

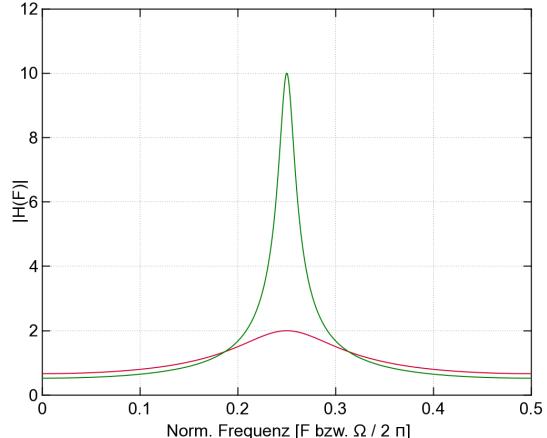
**Darstellung in Summenform:** Für jeden Pol wird ein Partialbruch mit zu bestimmendem Zählerkoeffizienten aufgestellt, danach werden die Partialbrüche auf einen gemeinsamen Nenner erweitert. Der Nenner ist jetzt wieder identisch zum ursprünglichen Nenner.

**Bestimmung der Koeffizienten:** Der umgeformte Zähler wird mit dem ursprünglichen gleichgesetzt, die Koeffizienten werden durch Koeffizientenvergleich ermittelt.

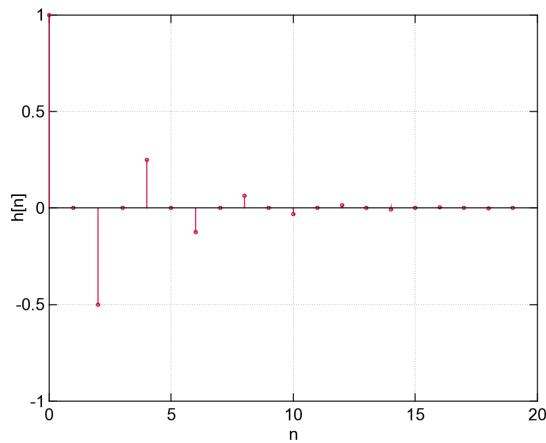
<sup>3</sup>Bei praktischen DSV-Systemen ist wie hier in der Aufgabe höchstens  $N = P$  möglich, sonst wird das System akausal.



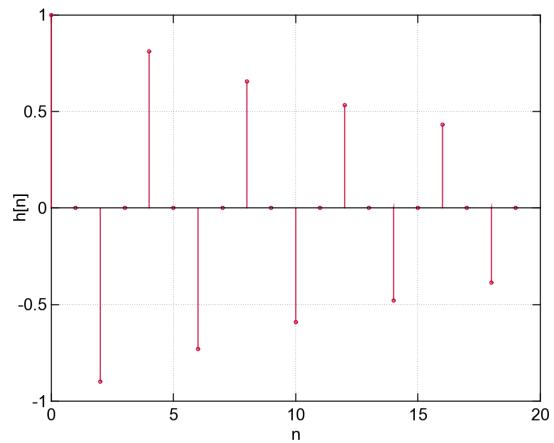
(a)



(b)

Abb. M2.19: Pol-Nullstellen-Plan für  $\alpha = 0,5$  (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9d

(a)



(b)

Abb. M2.20: Impulsantwort für  $\alpha = 0,5$  (a) und  $\alpha = 0,9$  (b) zu Aufgabe 2.9d

**System c):**

$$\begin{aligned}
 H_c(z) &= \frac{z^2}{z^2 - \alpha} = 1 + \frac{\alpha}{z^2 - \alpha} = 1 + \frac{\alpha}{(z - \sqrt{\alpha})(z + \sqrt{\alpha})} \stackrel{!}{=} 1 + \frac{b_1}{z - \sqrt{\alpha}} + \frac{b_2}{z + \sqrt{\alpha}} \\
 &= 1 + \frac{b_1(z + \sqrt{\alpha}) + b_2(z - \sqrt{\alpha})}{(z - \sqrt{\alpha})(z + \sqrt{\alpha})} \\
 \Rightarrow \quad \alpha &\stackrel{!}{=} b_1(z + \sqrt{\alpha}) + b_2(z - \sqrt{\alpha}) = (b_1 + b_2)z + (b_1 - b_2)\sqrt{\alpha} \\
 \Rightarrow \quad b_1 + b_2 &= 0 \text{ und } b_1 - b_2 = \sqrt{\alpha} \quad \Rightarrow \quad b_1 = \sqrt{\alpha}/2, \quad b_2 = -\sqrt{\alpha}/2 \\
 \Rightarrow \quad H_c(z) &= 1 + \frac{\sqrt{\alpha}}{2(z - \sqrt{\alpha})} - \frac{\sqrt{\alpha}}{2(z + \sqrt{\alpha})}
 \end{aligned}$$

•—○

$$\begin{aligned}
 h_c[n] &= \delta[n] + \sqrt{\alpha}/2 \left( (\sqrt{\alpha})^{n-1} - (-\sqrt{\alpha})^{n-1} \right) u[n] = \{1; 0; \alpha; 0; \alpha^2; \dots\} \text{ für } n = 0, 1, \dots \\
 h_c[2n] &= (\sqrt{\alpha})^{2n} u[2n]; \quad h_c[2n+1] = 0 \quad \text{für } n = 0, 1, \dots
 \end{aligned}$$

System d):

$$\begin{aligned}
 H_d(z) &= \frac{z^2}{z^2 + \alpha} = 1 - \frac{\alpha}{z^2 + \alpha} = 1 - \frac{\alpha}{(z - j\sqrt{\alpha})(z + j\sqrt{\alpha})} \stackrel{!}{=} 1 + \frac{b_1}{z - j\sqrt{\alpha}} + \frac{b_2}{z + j\sqrt{\alpha}} \\
 &= 1 + \frac{b_1(z + j\sqrt{\alpha}) + b_2(z - j\sqrt{\alpha})}{(z - j\sqrt{\alpha})(z + j\sqrt{\alpha})} \\
 \Rightarrow -\alpha &\stackrel{!}{=} b_1(z + j\sqrt{\alpha}) + b_2(z - j\sqrt{\alpha}) = (b_1 + b_2)z + (b_1 - b_2)j\sqrt{\alpha} \\
 \Rightarrow (b_1 + b_2)z &= 0 \quad \Rightarrow \quad b_1 = -b_2 \text{ und} \\
 (b_1 - b_2)j\sqrt{\alpha} &= -\alpha \Rightarrow b_1 - b_2 = j\sqrt{\alpha} \quad \Rightarrow \quad b_1 = j\sqrt{\alpha}/2, \quad b_2 = -j\sqrt{\alpha}/2 \\
 \Rightarrow H_d(z) &= 1 + \frac{j\sqrt{\alpha}/2}{z - j\sqrt{\alpha}} - \frac{j\sqrt{\alpha}/2}{z + j\sqrt{\alpha}}
 \end{aligned}$$

• ↵ ○

$$\begin{aligned}
 h_d[n] &= \delta[n] + j\sqrt{\alpha}/2 \left( (j\sqrt{\alpha})^{n-1} - (-j\sqrt{\alpha})^{n-1} \right) u[n] = \{1; 0; -\alpha; 0; \alpha^2; \dots\} \\
 &\text{für } n = 0, 1, \dots \\
 h_d[2n] &= (\sqrt{\alpha})^{2n} u[2n]; \quad h_d[2n+1] = 0 \quad \text{für } n = 0, 1, \dots
 \end{aligned}$$

**Anmerkung:**

Die Übertragungsfunktion in Summenform, die man nach der Partialbruchzerlegung erhält, lässt sich in Hardware umsetzen; man nennt dies die *Parallelform*.

## M2.10 IIR-Filter zweiter Ordnung → A2.10

### a) Differenzengleichung

Das Filter in Abb. 2.5a ist ein IIR-Filter zweiter Ordnung in Direktform 1 (Abb. M1.7a)

mit  $b_0 = -0,2$ ,  $b_1 = 0,2$ ,  $a_1 = 0$ ,  $a_2 = -0,81$ .

Die Differenzengleichung lautet:  $y[n] = -0,2x[n] + 0,2x[n-1] - 0,81y[n-2]$ .

### b) Impulsantwort aus Differenzengleichung

Zur Bestimmung der Impulsantwort wird der zeitdiskrete Dirakstoß als Eingangsfolge verwendet, alle Register haben den Wert 0 bei  $n \leq 0 \Rightarrow y[-1] = y[-2] = 0$ :

$$x[n] = \delta[n] = \begin{cases} 1 & \text{für } n = 0 \\ 0 & \text{sonst} \end{cases}$$

$$\Rightarrow h[n] = y[n] = -0,2\delta[n] + 0,2\delta[n-1] - 0,81h[n-2] \Rightarrow$$

$$\begin{aligned} n = 0 : \quad h[0] &= -0,2\delta[0] + 0,2\delta[-1] - 0,81h[-2] &= -0,2 \cdot 1 + 0 + 0 &= \mathbf{-0,2} \\ n = 1 : \quad h[1] &= -0,2\delta[-1] + 0,2\delta[0] - 0,81h[-1] &= 0 + 0,2 \cdot 1 + 0 &= \mathbf{+0,2} \\ n = 2 : \quad h[2] &= -0,2\delta[-2] + 0,2\delta[-1] - 0,81h[0] &= 0 + 0 - 0,81 \cdot (-0,2) &= \mathbf{+0,162} \\ n = 3 : \quad h[3] &= \dots - 0,81h[1] &= 0 + 0 - 0,81 \cdot 0,2 &= \mathbf{-0,162} \\ n = 4 : \quad h[4] &= \dots - 0,81h[2] &= 0 + 0 - 0,81 \cdot 0,162 &= \mathbf{-0,131} \\ n = 5 : \quad h[5] &= \dots - 0,81h[3] &= 0 + 0 - 0,81 \cdot (-0,162) &= \mathbf{+0,131} \\ n = 6 : \quad h[6] &= \dots - 0,81h[4] &= 0 + 0 - 0,81 \cdot (-0,131) &= \mathbf{+0,106} \\ &&&\vdots \end{aligned}$$

### c) Ausgangssignal über diskrete Faltung

Das Ausgangssignal  $y[n]$  eines LTI-Systems erhält man allgemein durch diskrete Faltung von Impulsantwort  $h[n]$  und Eingangssignal  $x[n]$ :

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

Allerdings hat das System in diesem Fall eine unendlich ausgedehnte Impulsantwort (IIR-Filter), die außerdem nicht in geschlossener Form bestimmt wurde, so dass man auch  $y[n]$  nicht in geschlossener Form bestimmen kann. Da das Eingangssignal  $x[n]$  nur eine Länge von drei hat, kann  $y[n]$  aus den gewichteten und verzögerten Impulsantworten bestimmt werden:

$k$	$x[k]$	$n =$	-1	0	1	2	3	4	5
-1	1	$x[-1]h[n+1]$	-0,2	0,2	0,162	-0,162	-0,131	0,131	0,106
0	2	$x[0]h[n]$		-0,4	0,4	0,324	-0,324	-0,262	0,262
1	-1	$x[1]h[n-1]$			0,2	-0,2	-0,162	0,162	0,131
$y[n] = \sum x[k]h[n-k] =$			-0,2	-0,2	0,762	-0,038	-0,617	0,031	0,489

#### d) Übertragungsfunktion

Aus der Differenzengleichung in a) oder direkt aus Abb. 2.5a (durch „genaues Hinschauen“) lässt sich leicht die Übertragungsfunktion konstruieren:

$$\begin{aligned} y[n] &= -0,2x[n] + 0,2x[n-1] - 0,81y[n-2] \\ \circlearrowleft \bullet Y(z) &= -0,2X(z) + 0,2X(z)z^{-1} - 0,81Y(z)z^{-2} \\ \Rightarrow H(z) &= \frac{Y(z)}{X(z)} = \frac{-0,2 + 0,2z^{-1}}{1 + 0,81z^{-2}} = \frac{-0,2z^2 + 0,2z^1}{z^2 + 0,81} = -0,2 \frac{z^2 - z^1}{z^2 + 0,81} \end{aligned}$$

Die letzte Form von  $H(z)$  ist besonders gut geeignet, um die Pol- und Nullstellen zu ermitteln (positive Exponenten, Koeffizient 1 der höchsten Potenzen in Zähler und Nenner):

$$\begin{aligned} \text{Nullstellen: } z^2 - z &= z(z-1) \stackrel{!}{=} 0 \quad \Rightarrow z_{01} = 0; z_{02} = 1 \\ \text{Polstellen: } z^2 + 0,81 &\stackrel{!}{=} 0 \Rightarrow z^2 = -0,81 \Rightarrow z_{\infty 1,2} = \pm 0,9j \end{aligned}$$

Es handelt sich um einen Bandpass (BP); das Maximum von  $|H(e^{j\Omega})|$  im Basisband liegt in der Nähe von  $\Omega = \pi/2$  ( $f = f_s/4$ ), da dort die Polstelle dem Einheitskreis am nächsten kommt.

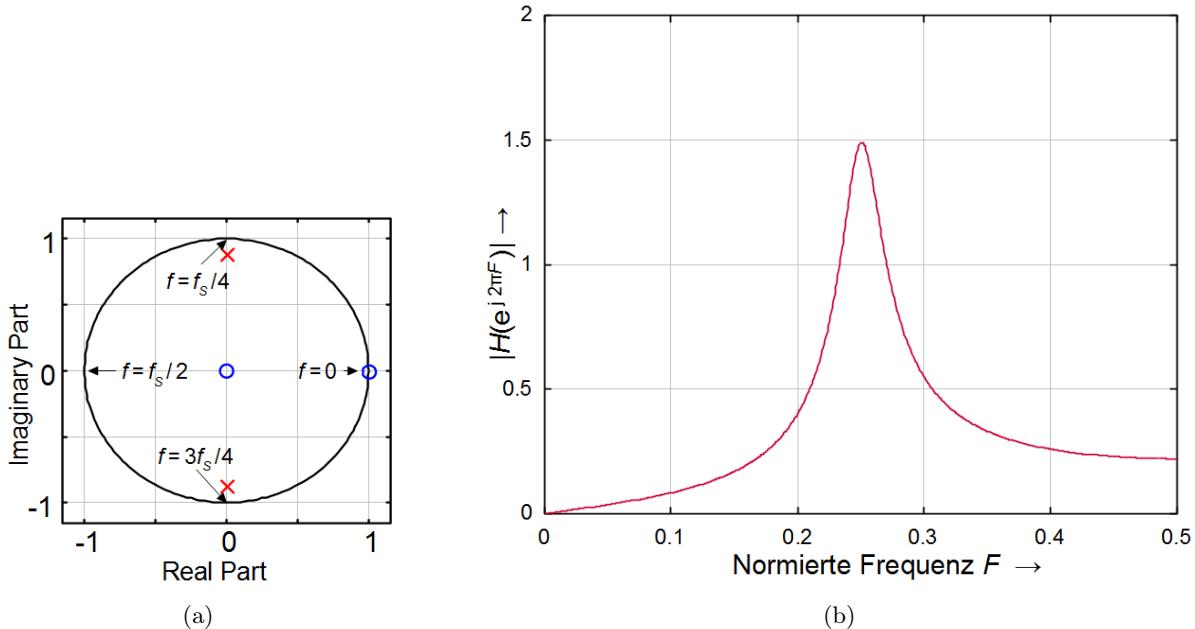


Abb. M2.21: Pol-Nullstellen-Plan (a) und Amplitudenfrequenzgang (b) zu Abb. 2.5a

#### e) Betrag und Phase bei $f_1 = 25$ kHz

Eine Abtastperiode von  $T_S = 10 \mu s$  entspricht einer Abtastfrequenz von  $f_S = 1/T_S = 100$  kHz. Damit erhält man die normierte Frequenz:

$$\begin{aligned} F_1 &= f_1/f_S = 1/4; \Omega_1 = 2\pi F_1 = \pi/2 \Rightarrow z_1 = e^{j\Omega_1} = e^{j\pi/2} = j \\ \Rightarrow H(f)|_{f=25 \text{ kHz}} &= H(z)|_{z=j} = -0,2 \frac{j^2 - 0,2j}{j^2 + 0,81} = -0,2 \frac{-1 - j}{-1 + 0,81} = 1,0526 (-1 - j) \end{aligned}$$

$H(z = j)$  hat also einen negativen Real- und einen negativen Imaginärteil. Daher muss der Phasenwinkel im dritten Quadranten liegen, also  $-90^\circ > \angle H(z = j) > -180^\circ$  sein. Diese Betrachtung ist wichtig, da

$$\arctan \frac{\Im\{H(z = j)\}}{\Re\{H(z = j)\}} = \arctan \frac{-1,0526}{-1,0526} = \arctan \frac{1,0526}{1,0526} = 45^\circ \hat{=} \pi/4$$

nicht dem Phasenwinkel entspricht - die Arcustangens-Funktion kann nicht zwischen erstem und dritten Quadranten unterscheiden (oder zwischen zweitem und viertem)! Daher gibt es in vielen Programmiersprachen die `atan2`-Funktion, die für den dritten und vierten Quadranten den Arcustangens-Wert um  $\pi$  (bzw.  $180^\circ$ ) korrigiert:

$$\angle H(z = j) = \text{atan2} \frac{\Im\{H(z = j)\}}{\Re\{H(z = j)\}} = \arctan \frac{\Im\{H(z = j)\}}{\Re\{H(z = j)\}} + 180^\circ = 45^\circ + 180^\circ = 225^\circ \hat{=} \frac{5\pi}{4}$$

Eine Korrektur um  $-\pi$  bzw.  $-180^\circ$  wäre ebenfalls richtig und würde das äquivalente Ergebnis liefern  $\angle H(z = j) = -135^\circ \hat{=} -3\pi/4$ .

Für die Betragsberechnung ist diese Fallunterscheidung natürlich nicht nötig:

$$|H(z = j)| = \sqrt{\Im^2\{H(z = j)\} + \Re^2\{H(z = j)\}} = 1,0526\sqrt{2} = 1,489$$

### f) Übertragungsfunktion in Produktform

Pole und Nullstellen wurden ja bereits in Punkt d) bestimmt, damit kann man  $H(z)$  in Produktform sofort aufstellen:

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

Wie im vorigen Unterpunkt sind  $f_1 = 25$  kHz und  $f_S = 100$  kHz, also gilt wieder  $F_1 = 1/4$  und  $z_1 = j$ :

$$H(F = F_1) = H(z = j) = \frac{-0,2j(j - 1)}{(j - 0,9j)(j + 0,9j)} = \frac{-0,2(j - 1)}{0,19}$$

Der Gesamtbetrag ergibt sich aus dem Produkt der Teilbeträge, die hier besonders leicht zu bestimmen sind:

$$|H(z = j)| = \frac{|-0,2| |j| |j - 1|}{|j - 0,9j| |j + 0,9j|} = \frac{0,2 \cdot 1 \cdot \sqrt{2}}{0,1 \cdot 1,9} = 1,0526\sqrt{2} = 1,489$$

Die Gesamtphase erhält man durch Aufsummieren der Teilphasen, Beiträge im Nenner müssen subtrahiert werden:

$$\begin{aligned} \angle H(z = j) &= \angle(-0,2) + \underbrace{\angle(j) + \angle(j - 1)}_{\text{Zähler}} - \underbrace{(\angle(j - 0,9j) + \angle(j + 0,9j))}_{\text{Nenner}} \\ &= 180^\circ + 90^\circ + 135^\circ - (90^\circ + 90^\circ) = 225^\circ \hat{=} 5\pi/4 \end{aligned}$$

**Anmerkung:** Die Berechnungen in diesem Unterpunkt entsprechen der grafischen Abschätzung von Amplituden- und Phasengang aus der Lage von Polen und Nullstellen im P/N-Diagramm. Auch hier werden die Entfernungen (= Beträge) und Winkel der Pole und Nullstellen zum Aufpunkt auf dem Einheitskreis zusammengefasst.

### g) Änderung der Filterkoeffizienten

Die Koeffizienten des Zählers dürfen mit einem (einheitlichen!) konstanten Faktor skaliert werden, ohne dass sich die Lage der Nullstellen ändert. Im Nenner wäre dies rechnerisch genauso möglich, ohne dass sich die Lage der Polstellen verschiebt. Allerdings wird dann  $a_0 \neq 1$  und die zugehörige Differenzengleichung lässt sich nicht mehr direkt in eine Hardwarestruktur umwandeln. Versuchen Sie es selbst einmal.

Daher kann nur der Zähler, nicht aber der Nenner skaliert werden!

Es wurde bereits berechnet, dass  $|H(f)|_{f=25 \text{ kHz}} = |H(z = j)| = 1,489$  ist.

Soll  $|H(z = j)| = 1$  gelten, müssen die Zählerkoeffizienten durch 1,489 dividiert werden:

$$b'_0 = b_0/1,489 = -0,2/1,489 = -0,1344 \quad \text{und} \quad b'_1 = b_1/1,489 = 0,2/1,489 = 0,1344.$$

### h) Phasenverschiebung und Gruppenlaufzeit bei $f = 25$ kHz

Die Phasenverschiebung wurde bereits in Unterpunkt f) zu  $\Delta\phi = 5\pi/4$  berechnet.

## M2.11 Moving Average Filter / rect-Fenster → A2.11

Die Stoßantwort und Übertragungsfunktion von Moving Average Filtern der Ordnung  $N - 1$  (d.h. mit  $N - 1$  Verzögerungsgliedern) lauten:

$$h[n] = \sum_{k=0}^{N-1} \delta[n - k] \Rightarrow H(z) = \sum_{k=0}^{N-1} z^{-k} = \frac{1 - z^{-N}}{1 - z^{-1}}$$

Die Summe in der  $z$ -Ebene wurde mit Hilfe der Formel für die endliche geometrische Reihe (B.42) umgeformt. Aus der Systemfunktion  $H(z)$  erhält man die Frequenzantwort auf dem Einheitskreis durch Setzen von  $z = e^{j\Omega} = e^{j\omega T_S}$ . Im nächsten Schritt werden aus Zähler und Nenner Exponentialterme so ausgeklammert, dass ein symmetrisches Polynom („Spiegelpolynom“) entsteht, das mit den Eulerschen Identitäten in Sinus bzw. Cosinus-Funktionen umgewandelt werden kann.

$$\begin{aligned} H(e^{j\Omega}) &= \sum_{k=0}^{N-1} e^{-jk\Omega} = \frac{1 - e^{-jN\Omega}}{1 - e^{-j\Omega}} = \frac{e^{-jN\Omega/2}}{e^{-j\Omega/2}} \frac{e^{+jN\Omega/2} - e^{-jN\Omega/2}}{e^{+j\Omega/2} - e^{-j\Omega/2}} \\ &= e^{-j(N-1)\Omega/2} \frac{\sin(N\Omega/2)}{\sin(\Omega/2)} = \underbrace{e^{-j(N-1)\Omega/2}}_{\text{lineare Phase}} \cdot \underbrace{N}_{H(\Omega=0)} \cdot \underbrace{\text{di}_N(\Omega)}_{\text{Amplitudengang}} \end{aligned}$$

Der ausgeklammerte Exponentialterm hat einen Betrag von 1 und beschreibt eine lineare Phase von  $(N - 1)\Omega/2$ . Damit lautet der Betrag der Frequenzantwort:

$$|H(f)| = \left| \frac{\sin(N\Omega/2)}{\sin(\Omega/2)} \right| = |N \text{di}_N(\Omega)| \text{ mit } H(0) = N$$

Für den Einsatz als Moving Average Filter muss also der Ausgangswert noch mit  $1/N$  skaliert werden.

Die erste Nullstelle  $f_0$  der Frequenzantwort liegt bei  $N\omega T_S/2 = \pi \Rightarrow f_0 = f_S/N$  bzw.  $F_0 = 1/N$  bzw.  $\Omega = 2\pi/N$ .

Das Maximum des ersten Nebenzipfels liegt in der Mitte<sup>4</sup> zwischen erster und zweiter Nullstelle bei  $f_{NZ} = 3f_S/2N$ .

$$\frac{|H(f_{NZ})|}{H(0)} \approx \frac{1}{N} \left| \frac{\sin(3\pi/2)}{\sin(3\pi/2N)} \right| \approx \frac{1}{N} \frac{1}{3\pi/2N} = \frac{2}{3\pi} \text{ mit } \sin x \approx x \text{ für } x \ll 1$$

Damit ist die Höhe des ersten Nebenzipfels nahezu unabhängig von der Länge des MA - Filters.

	Allgemein	$N = 32$	$N = 1024$
$H_0$	$N$	32	1024
$f_{01}$	$f_S/N$	320 kHz	10 kHz
$H(f_{NZ})$	$2/(3\pi)$	$0,21 \hat{=} -13,5$ dB	$0,21 \hat{=} -13,5$ dB

Tab. M2.2: Daten zu Aufgabe 2.11

<sup>4</sup>Gilt in guter Näherung, da sich die Sinus-Funktion im Nenner zwischen zwei Zählnullstellen nur wenig ändert.

**Anmerkungen:**

- Eine Verbesserung der Nebenzipfelunterdrückung kann einfach erreicht werden, indem man mehrere MA-Filter kaskadiert bzw. die Daten das gleiche Filter mehrfach durchlaufen lässt.
- Eine direkte Implementierung eines MA-Filters mit  $N - 1 = 1023$  Addierern wäre sehr ineffizient, in der Praxis werden MA-Filter daher fast immer als Cascaded Integrator-Comb (CIC) Filter realisiert (Kap. 6, 8 und 9)
- Die Ergebnisse beschreiben auch das Verhalten eines Rechteckfensters mit  $N = 32$  bzw. 1024 Punkten!

**M2.12 Notchfilter → A2.12**

Simulationen finden Sie im Notebook **LTF-Notchfilter**.

- a) Am einfachsten löscht man den Störton aus, indem man eine Nullstelle genau bei der Frequenz  $F_1 = f_1/f_S = 0,3$  auf dem Einheitskreis platziert,  $z_{0,1} = e^{j2\pi F_1}$ . Um ein reellwertiges Filter zu erhalten, muss eine zweite konjugiert-komplexe Nullstelle  $z_{0,2} = z_{0,1}^* = e^{-j2\pi F_1}$  hinzugefügt werden. Daher hat das Filter mindestens die Ordnung zwei.

Die Übertragungsfunktion des einfachen Notchfilters lautet

$$H_1(z) = z^{-2}(z - e^{j2\pi F_1})(z - e^{-j2\pi F_1}) = z^{-2}(z^2 - 2z \Re\{e^{j2\pi F_1}\} + 1) = 1 - 2z^{-1} \cos 2\pi F_1 + z^{-2}.$$

Frequenz- und Betragsgang sind:

$$\begin{aligned} H_1(e^{j\Omega}) &= 1 - 2e^{-j\Omega} \cos 2\pi F_1 + e^{-2j\Omega} = e^{-j\Omega} (e^{j\Omega} - 2 \cos 2\pi F_1 + e^{-j\Omega}) \\ &= 2e^{-j\Omega} (\cos \Omega - \cos 2\pi F_1) \\ \Rightarrow |H_1(e^{j\Omega})| &= 2 |\cos 2\pi F - \cos 2\pi F_1| = 2 |\cos 2\pi F + 0,309| \end{aligned}$$

Man sieht, dass zwar bei  $F = F_1$  die gewünschte Auslöschung stattfindet, dass aber der Frequenzgang unsymmetrisch zu  $F_1$  ist.

- b) Die doppelten Nullstellen bei  $F_1$  führen zu der Übertragungsfunktion  $H_2(z) = H_1^2(z)$ :

$$\begin{aligned} H_2(z) &= z^{-4}(z - e^{j2\pi F_1})^2(z - e^{-j2\pi F_1})^2 = H_1^2(z) \\ \Rightarrow |H_2(e^{j\Omega})| &= |H_1(e^{j\Omega})|^2 = 4 (\cos 2\pi F - \cos 2\pi F_1)^2 \end{aligned}$$

Der Betragsgang von  $H_2$  weist daher eine stärkere Dämpfung in der Nähe der Notchfrequenz, aber auch eine noch stärkere Asymmetrie auf als der von  $H_1$ .

- c) Die Polstellen sollen den gleichen Winkel aufweisen wie die Nullstellen,  $z_{\infty,1} = r_1 e^{j2\pi F_1}$  und  $z_{\infty,2} = r_1 e^{-j2\pi F_1}$ . Die Übertragungsfunktion lautet damit:

$$\begin{aligned} H_3(z) &= \frac{(z - e^{j2\pi F_1})(z - e^{-j2\pi F_1})}{(z - r_1 e^{j2\pi F_1})(z - r_1 e^{-j2\pi F_1})} = \frac{z^2 - 2z \Re\{e^{j2\pi F_1}\} + 1}{z^2 - 2r_1 z \Re\{e^{j2\pi F_1}\} + r_1^2} \\ &= \frac{z^2 - 2z \cos 2\pi F_1 + 1}{z^2 - 2r_1 z \cos 2\pi F_1 + r_1^2} \end{aligned}$$

Frequenz- und Betragsgang sind:

$$\begin{aligned}
 H_3(e^{j\Omega}) &= \frac{e^{j\Omega} (e^{j\Omega} - 2 \cos \Omega_1 + e^{-j\Omega})}{e^{j\Omega} (e^{j\Omega} - 2r_1 \cos \Omega_1 + r_1^2 e^{-j\Omega})} = \frac{e^{j\Omega} - 2 \cos \Omega_1 + e^{-j\Omega}}{e^{j\Omega} - 2r_1 \cos \Omega_1 + r_1^2 e^{-j\Omega}} \\
 &= \frac{2 (\cos \Omega - \cos \Omega_1)}{\cos \Omega + j \sin \Omega - 2r_1 \cos \Omega_1 + r_1^2 (\cos \Omega - j \sin \Omega)} \\
 \Rightarrow |H_3(e^{j\Omega})| &= \frac{2 |\cos \Omega - \cos \Omega_1|}{\sqrt{((1 + r_1^2) \cos \Omega - 2r_1 \cos \Omega_1)^2 + ((1 - r_1^2) \sin \Omega)^2}}
 \end{aligned}$$

Wer es schafft,  $|H_3(e^{j\Omega})|$  übersichtlicher aufzulösen, wird in der nächsten Ausgabe namentlich erwähnt ...

	$H(F = 0)$	$H(F = 0,29)$	$H(F = 0,31)$	$H(F = 0,5)$
$ H_1(F) $	2,618 $\equiv$ 8,36 dB	0,121 $\equiv$ -18,37 dB	0,118 $\equiv$ -18,55 dB	1,382 $\equiv$ 2,81 dB
$ H_2(F) $	6,854 $\equiv$ 16,72 dB	0,015 $\equiv$ -36,74 dB	0,014 $\equiv$ -37,09 dB	1,910 $\equiv$ 5,62 dB
$ H_3(F) $	1,052 $\equiv$ 0,44 dB	0,815 $\equiv$ -1,78 dB	0,815 $\equiv$ -1,78 dB	1,051 $\equiv$ 0,43 dB

Tab. M2.3: Ergebnisse zu Aufgabe [M2.12](#)

# M3 DFT: Diskrete Fouriertransformation und FFT

## M3.1 Allgemeine Fragen zu DFT und FFT → A3.1

- a) **Fourierreihe:** Ein periodisches Signal kann durch eine unendliche Summe von Sinus- und Cosinusschwingungen dargestellt werden, deren Frequenzen ganzzahlige Vielfache der Grundschwingung sind. Die Amplituden der Sinus- und Cosinusschwingungen nennt man *Fourierkoeffizienten*, das Spektrum ist ein Linienspektrum.

**Fouriertransformation:** Ein unendlich ausgedehntes Signal mit endlicher Energie (z.B. ein pulsförmiges Signal das bis auf wenige Ausnahmen den Wert Null hat) kann durch ein unendlich ausgedehntes Integral von Sinus- und Cosinusschwingungen dargestellt werden. Die Gewichtungsfunktion für die Sinus- und Cosinusschwingungen nennt man *Fouriertransformierte*, es ist im Allgemeinen eine kontinuierliche Funktion. Hat das Eingangssignal unendliche Energie (z.B. ein Zufallsprozess), ist die Fouriertransformierte nicht definiert. Eine Aussage über die spektralen Eigenschaften dieses Prozesses lässt sich nur treffen, indem man die Fouriertransformierte der Autokorrelationsfunktion berechnet. Das Ergebnis ist dann die *Leistungsdichfunktion* mit der Einheit W/Hz.

**DTFT:** Discrete-Time Fourier Transform; wird vergleichbar mit der konventionellen Fourieranalyse und -synthese auf unendlich ausgedehnte zeitdiskrete Folgen angewendet. Aufgrund der zeitdiskreten Natur des Eingangssignals ist das Spektrum periodisch und im Allgemeinen kontinuierlich (Ausnahme: Das Eingangssignal ist periodisch). Da nur unendlich ausgedehnte Folgen analysiert werden können, ist die DTFT ungeeignet für computerbasierte Analysen.

**DFT:** Discrete Fourier Transform; aus einer unendlich ausgedehnten Folge wird ein Stück herausgeschnitten (Fensterung), für die DFT tut man dann so, als ob sich das herausgeschnittene Stück unendlich wiederholt. Aufgrund dieser „Periodizität“ ist das resultierende Spektrum ein Linienspektrum; im allgemeinen Fall entsprechen  $N$  Werte im Zeitbereich  $N$  Werten im Frequenzbereich. Die DFT ist daher eine Blocktransformation, die sehr gut für computerbasierte Berechnungen geeignet ist.

**FFT:** Fast Fourier Transform. Unter FFT versteht man verschiedene effiziente Algorithmen zur schnelleren Berechnung der DFT. Bei idealer Arithmetik liefern DFT und FFT identische Ergebnisse, allerdings sind für eine  $N$ -Punkt DFT  $N^2$  komplexe Multiplikationen notwendig. Das entspricht  $4N^2$  reellen Multiplikationen bei komplexwertigen und  $2N^2$  reellen Multiplikationen bei reellwertigen Eingangssignalen. Für eine Radix-2 FFT mit  $N$  Punkten sind nur ca.  $2N \log_2 N$  reelle Multiplikationen erforderlich (die genaue Zahl hängt vom verwendeten Algorithmus ab).

- b) **Frequenzen  $f_k$  der Spektrallinien:** Eine DFT mit  $N$  Punkten liefert  $N$  Frequenzpunkte im Frequenzbereich  $[0 \dots f_S]$ . Der Abstand der Spektrallinien ist  $\Delta f = f_S/N$ , die Linien treten auf bei  $f_k = k f_S/N$  mit  $k = 0, 1, \dots, N-1$  und die höchste Frequenz liegt dementsprechend bei  $f_S(N-1)/N$ . Bei reellwertigen Eingangssignalen sind die oberen  $N/2$  Frequenzpunkte redundant, da Amplituden- und Phasenspektrum symmetrisch zu  $f = 0$  und  $f = f_S/2$  sind.
- c) **Auswertung in der positiven Hälfte des Nyquistbereichs** Bei reellwertigen Signalen sind Real- und Imaginärteil bzw. Betrag und Phase der DFT symmetrisch zu  $f = 0$ , daher genügt es eine Hälfte der Koeffizienten zu kennen. Hierbei haben Realteil und Betrag eine gerade Symmetrie

und Imaginärteil und Phase eine ungerade Symmetrie. Bei komplexwertigen Eingangssignalen müssen beide Hälften ausgewertet werden.

- d) Eine **Fensterung vor Berechnung einer DFT / FFT** ist notwendig, da die DFT im Gegensatz zur DTFT eine Blocktransformation ist, die mit einer endlichen Anzahl Samples arbeitet (s.o.). Wenn von vorneherein nur eine endliche Anzahl Samples zur Verfügung steht, kann man sich vorstellen, dass die restlichen (unendlich vielen) Samples alle Null waren (DTFT) oder sich periodisch wiederholen und mit einem Rechteckfenster abgetrennt wurden.

## M3.2 Fourierreihe und kohärente DFT → A3.2

CT Signal:  $s(t) = 1 \text{ V} (1 + \cos(2\pi f_1 t + \pi/4))$  mit  $f_1 = 1 \text{ kHz}$

DT Signal:  $s[n] = 1 \text{ V} (1 + \cos(2\pi f_1 n T_S + \pi/4)) = 1 \text{ V} (1 + \cos(2\pi n F_1 + \pi/4))$

a) **Betragsspektrum und Phasenspektrum des zeitkontinuierlichen (CT) Signals**

erhält man im Allgemeinen über die kontinuierliche Fouriertransformation (C.1), bei periodischen Signalen (wie hier) über die Fourierreihe (C.3):

$$\begin{aligned} c_k &= \frac{1}{T_1} \int_0^{T_1} s(t) e^{-j2k\pi f_1 t} dt = \frac{1 \text{ V}}{T_1} \int_0^{T_1} (1 + \cos(2\pi f_1 t + \pi/4)) e^{-j2k\pi f_1 t} dt \\ &= \frac{1 \text{ V}}{T_1} \left( \int_0^{T_1} e^{-j2k\pi f_1 t} + \frac{1}{2} (\underbrace{e^{j(2\pi f_1 t + \pi/4)} + e^{-j(2\pi f_1 t + \pi/4)}}_{\text{DC}}) e^{-j2k\pi f_1 t} \right) dt \\ &= \frac{1 \text{ V}}{T_1} \left( \int_0^{T_1} e^{-j2k\pi f_1 t} + \frac{1}{2} (\underbrace{e^{j(2\pi f_1 t(1-k) + \pi/4)} + e^{-j(2\pi f_1 t(1+k) + \pi/4)}}_{\text{DC}}) \right) dt \\ &= \begin{cases} 1 \text{ V} & \text{für } k = 0 \quad (\text{DC}) \\ \frac{1}{2} e^{j\pi/4} \text{ V} & \text{für } k = +1 \quad (+1 \cdot f_1 = +1 \text{ kHz}) \\ -\frac{1}{2} e^{j\pi/4} \text{ V} & \text{für } k = -1 \quad (-1 \cdot f_1 = -1 \text{ kHz}) \\ 0 & \text{sonst} \end{cases} \end{aligned}$$

da die Integrale über die Exponentialfunktion mit Periodizität  $kT_1$  und dem Integrationsintervall von  $T_1$  nur einen Beitrag liefern wenn der Exponent Null ist. Natürlich lässt sich das Spektrum hier auch durch „genaues Hinschauen“ ermitteln, da im Integral nur zwei komplexe Drehzeiger  $0,5 \text{ V} \cdot e^{\pm j(2\pi f_1 t + \pi/4)}$  und ein DC-Wert stehen, deren Frequenzen / Beträge / Phasen man direkt ablesen kann.

b) **Abgetastetes Signal und Abtastperiode**

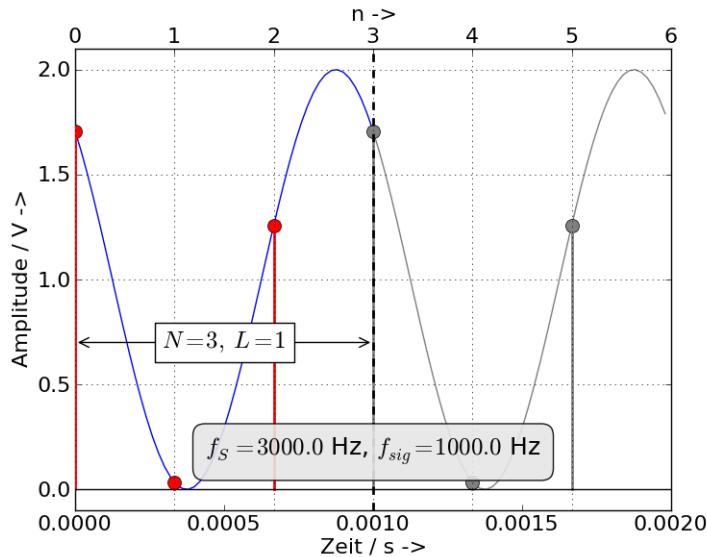
Bei einer Abtastfrequenz  $f_S$  und Abtastperiode  $T_S$  von

$$f_S = 3f_1 = 3 \text{ kHz} \Rightarrow T_S = (3f_1)^{-1} = 0,333 \text{ ms}$$

erhält man die Datenpunkte

$$\begin{aligned} \Rightarrow s[0] &= s(0 \cdot T_S) = 1 \text{ V} (1 + \cos(\pi/4)) = 1,7071 \text{ V} \\ s[1] &= 1 \text{ V} \cdot (1 + \cos(2\pi \cdot 1000/3000 + \pi/4)) = 0,0341 \text{ V} \\ s[2] &= 1 \text{ V} \cdot (1 + \cos(2\pi \cdot 2000/3000 + \pi/4)) = 1,2588 \text{ V} \\ s[3] &= 1 \text{ V} \cdot (1 + \cos(2\pi \cdot 3000/3000 + \pi/4)) = s[0] = 1,7071 \text{ V} \\ &\dots \end{aligned}$$

Das abgetastete Signal ist mit  $N = 3$  Abtastwerten periodisch, dabei wird eine ganzzahlige Anzahl  $K = 1$  Perioden des CT Signals erfasst. Die Anzahl beider Perioden steht in einem ganzzahligen Verhältnis, daher nennt man die Abtastung *kohärent*.

Abb. M3.1: Eingangssignal  $s(t)$  und abgetastetes Signal  $s[n]$  (zwei Perioden) zu Aufgabe 3.2

### c) Betrag und Phase der DFT der drei Abtastwerte

Die DFT (C.41) ist eine Block-Transformation, die zu  $N = 3$  Abtastwerten  $N = 3$  Frequenzpunkte berechnet:

$$S[k] = \frac{1}{N} \sum_{n=0}^{N-1} s[n] e^{-j2\pi kn/N} = \frac{1}{3} \sum_{n=0}^2 s[n] e^{-j2\pi kn/3}$$

$$\begin{aligned} S[0] &= \frac{1V}{3} (1,7071 + 0,0341 + 1,2588) = 1 \text{ V} \\ S[1] &= \frac{1V}{3} \left( 1,7071 + 0,0341e^{-j2\pi/3} + 1,2588e^{-j4\pi/3} \right) \\ &= \frac{1V}{3} (1,7071 + 0,0341(-0,5 - 0,8660 j) + 1,2589(-0,5 + 0,8660 j)) \text{ V} \\ &= (0,3536 + 0,3536 j) \text{ V} = 0,5e^{j\pi/4} \text{ V} \\ S[2] &= \frac{1V}{3} \left( 1,7071 + 0,0341e^{-j4\pi/3} + \underbrace{1,2588e^{-j8\pi/3}}_{e^{-j2\pi/3}} \right) \\ &= \frac{1V}{3} (1,7071 + 0,0341(-0,5 + 0,8660 j) + 1,2589(-0,5 - 0,8660 j)) \text{ V} \\ &= (0,3536 - 0,3536 j) \text{ V} = 0,5e^{-j\pi/4} \text{ V} = S[-1] = S^*[1] \end{aligned}$$

Die explizite Berechnung von  $S[2]$  wäre also nicht notwendig gewesen, da die DFT mit  $N = 3$  periodisch ist und bei reellen Zeitsignalen  $S[k] = S^*[-k]$  ist.

#### Skalierung der Frequenzpunkte:

Im Zeitbereich wird das Signal repräsentiert durch eine Periode der unendlich ausgedehnten Folge  $x[n]$  mit  $N$  Abtastwerten im Abstand  $T_S$  über  $L$  Perioden  $T_1$  des CT Signals. Insgesamt wird also eine Zeitspanne  $T = NT_S = LT_1$  abgedeckt.

Im Frequenzbereich treten periodische Spektren auf (zeitdiskretes Signal), hier wird das Signal repräsentiert durch die Folge  $X[k]$  mit  $N$  Frequenzpunkten im Intervall  $[0, f_S[$ . Der Index  $k$  entspricht daher der Frequenz  $f_S k/N$ ,  $k = 0 \dots N - 1$ , die maximale Frequenzauflösung ist  $\Delta f = f_S/N$ .

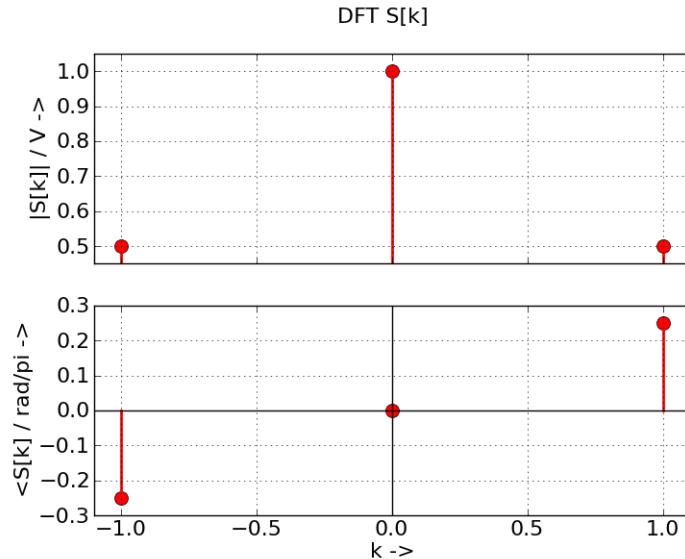


Abb. M3.2: Betrag und Phase der DFT  $S[k]$  für  $f_S = 3000$  Hz und  $N = 3$  zu Aufgabe 3.2c

#### d) Vergleich CFT mit DFT

Man sieht, dass die DFT für periodische, auf  $B < f_S/2$  bandbegrenzte CT Signale das gleiche Ergebnis liefern kann wie die CFT. Voraussetzung dazu ist, dass eine ganzzahlige Anzahl  $L$  Perioden des CT Signals  $s(t)$  so abgetastet wird, dass eine Folge  $s[n]$  mit Periode  $N$  entsteht. Hierfür müssen Abtastperiode  $T_S$  und Signalperiode  $T_1$  „kommensurabel“ (lat.: zusammen messbar) sein, d.h. ganzzahlige Vielfache einer gemeinsamen Zeiteinheit  $T_c$  sein mit

$$NT_S = LT_1 = NLT_c$$

Mathematisch ausgedrückt:

$$\exists T_c \in \mathbb{R} \text{ so, dass } T_1 = NT_c \wedge T_S = LT_c \text{ mit } T_1, T_S \in \mathbb{R} \text{ und } N, L \in \mathbb{Z}.$$

$$\Rightarrow \frac{T_1}{T_S} = \frac{f_S}{f_1} = \frac{N}{L} = c \in \mathbb{Q}.$$

Abtastfrequenz und Signalfrequenz müssen also in einem rationalen Verhältnis stehen. Dies ist in der Praxis nicht immer zu erreichen, da die maximale Anzahl der Abtastpunkte  $N$  aus Gründen der Messdauer, des Speicherplatzes und der Rechenzeit begrenzt ist. Besonders effiziente DFT Algorithmen (Fast Fourier Transform, FFT) erhält man für  $N = 2^r$ , was zusätzliche Einschränkungen mit sich bringt. Siehe hierzu auch Aufgabe 1.1 (Periodizität abgetasteter Signale).

**e) DFT mit 3 Abtastwerten pro Periode über zwei Perioden**

Berechnet man die DFT über 2 Perioden, wiederholen sich die Abtastwerte:

$$\begin{aligned} S[0] &= \frac{1V}{6} (1,7071 + 0,0341 + 1,2588 + 1,7071 + 0,0341 + 1,2588) = 1 \text{ V} \\ S[1] &= \frac{1V}{6} \left( 1,7071 + 0,0341e^{-j2\pi/6} + 1,2588e^{-j4\pi/6} \right. \\ &\quad \left. + 1,7071 e^{-j6\pi/6} + 0,0341 e^{-j8\pi/6} + 1,2588 e^{-j10\pi/6} \right. \\ &\quad \left. = -1 \quad = -e^{-j2\pi/6} \quad = -e^{-j4\pi/6} \right) = 0 \text{ V} \\ S[2] &= \frac{1V}{6} \left( 1,7071 + 0,0341e^{-j4\pi/6} + 1,2588e^{-j8\pi/6} \right. \\ &\quad \left. + 1,7071 e^{-j12\pi/6} + 0,0341 e^{-j16\pi/6} + 1,2588 e^{-j20\pi/6} \right. \\ &\quad \left. = +1 \quad = +e^{-j4\pi/6} \quad = +e^{-j8\pi/6} \right) = 0,5e^{-j\pi/4} \text{ V} \\ &\dots \end{aligned}$$

Jeder zweite Frequenzpunkt ist also Null. Das kann man leicht auch ohne Rechnung erkennen: Der Index  $k$  entspricht Frequenzen von  $k/Nf_S = k \cdot 500$  Hz, die Grundfrequenz ist aber 1 kHz. Sinnvoll kann eine solche synchrone Abtastung über mehrere Perioden trotzdem sein: Bei einer rotierenden Welle können so Frequenzkomponenten unterhalb der Grundfrequenz (Subharmonische) detektiert werden. Liefern Frequenzkomponenten unerwarteterweise einen Beitrag, kann das auch ein Hinweis auf eine nicht ganz synchrone Abtastung sein.

**f) Sechs Abtastwerte pro Periode**

Um sechs Abtastwerte je Periode zu erzielen, muss man hier die Abtastfrequenz auf  $f_{S2} = 6$  kHz erhöhen. Mit der Abtastfrequenz erhöht sich der abgedeckte Spektralbereich  $-f_{S2}/2 \dots +f_{S2}/2 = \pm 3$  kHz, aber nicht die Frequenzauflösung  $f_{S2}/N = 1$  kHz.

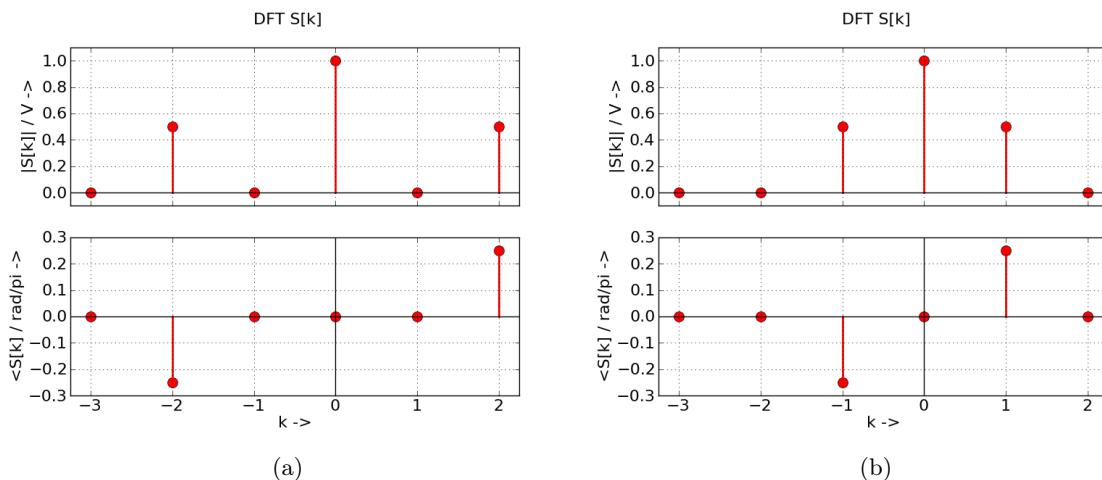


Abb. M3.3: Betrag und Phase der DFT  $S[k]$  für  $N = 6$  mit  $f_S = 3000$  Hz zu Aufgabe 3.2e (a) und mit  $f_S = 6000$  Hz zu Aufgabe 3.2f (b)

**g) Zeropadding mit drei Nullen auf insgesamt 6 Abtastwerte**

Setzt man die „normale“ Gleichung der DFT an, erhält man (die letzten drei Abtastwerte sind Null):

$$S[0] = \frac{1V}{6} (1,7071 + 0,0341 + 1,2588 + 0 + 0 + 0) = 0,5 \text{ V}$$

Um einen korrekten Skalierungsfaktor zu erhalten, dividiert man bei Zeropadding also nicht durch die Länge  $N$  der DFT, sondern durch die Anzahl  $L$  der relevanten Abtastwerte:

$$\begin{aligned} S[0] &= \frac{1V}{3} (1,7071 + 0,0341 + 1,2588 + 0 + 0 + 0) = 1 \text{ V} \\ S[1] &= \frac{1V}{3} (1,7071 + 0,0341e^{-j2\pi/6} + 1,2588e^{-j4\pi/6}) = 0,5220e^{-j0,2536\pi} \text{ V} \\ S[2] &= \frac{1V}{3} (1,7071 + 0,0341e^{-j4\pi/6} + 1,2588e^{-j8\pi/6}) = 0,5e^{-j\pi/4} \text{ V} \\ &\dots \end{aligned}$$

Die Werte für  $S[0]$  und  $S[2]$  sind identisch mit den Werten für  $N = 6$  aus Unterpunkt e), der Wert für  $S[1]$  ist ein Zwischenwert, der dem Wert der DTFT entspricht.

#### h) Anzahl der Rechenoperationen einer $N$ -Punkt DFT

Pro Frequenzpunkt müssen alle  $N$  reellen Abtastwerte mit komplexen Drehzeigern multipliziert und anschließend aufsummiert werden, es müssen daher jeweils  $2N$  reelle Multiplikationen und Additionen durchgeführt werden. Für die Berechnung der gesamten DFT sind also jeweils  $2N^2$  reelle Multiplikationen und Additionen notwendig. Aufgrund der Symmetrie  $S[k] = S^*[-k]$  bei reellen Zeitsignalen muss nur die Hälfte der Frequenzpunkte berechnet werden, insgesamt müssen daher  $N^2$  reelle Multiplikationen und Additionen berechnet werden.

#### i) Simulation in Python

Das Notebook **DFT-Basics\_ML** enthält die Lösung und Kommentare dazu.

### M3.3 Frequenzgang eines Moving Average (MA) - Filters → A3.3

Sie haben vermutlich  $|H(F = 0)| = 4$  und  $|H(F = 0,5)| = 0$  erwartet, vielleicht noch eine weitere Nullstelle bei  $F = 0,25$ . Das liefert die Simulation auch, aber der Verlauf entspricht nicht der erwarteten si-Funktion.

```

1 h = [1,1,1,1]
2 n = np.arange(len(h))
3 fig, (ax1, ax2) = plt.subplots(2,1)
4 ax1.stem(n,h)
5 ax1.set_xlabel(r"$n \rightarrow$")
6 ax1.set_ylabel(r"$h[n] \rightarrow$")
7 H = np.abs(np.fft.fft(h,256))
8 k = np.arange(len(H)/2)
9 ax2.stem(k,H[:len(k)])
10 ax2.set_xlabel(r"$k \rightarrow$")
11 ax2.set_ylabel(r"$|H[k]| \rightarrow$")
12 fig.tight_layout(True)
13 plt.show()

```

Lst. M3.1: Korrigierter Code aus Notebook **DFT-MA\_Filt\_ML** zu Aufgabe 3.3

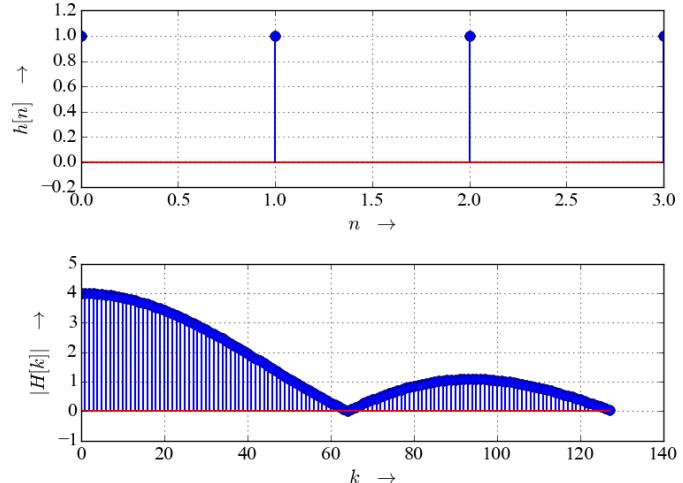


Abb. M3.4: Frequenzgang mit Zeropadding

Es gibt mehrere Fallen bei der Anwendung der DFT/FFT auf „echte“ Signale, in die am Anfang fast jeder stolpert:

**Die DFT/FFT ...**

**... lässt sich direkt nur auf periodische Signale anwenden.**

Die DFT ist eine Blocktransformation für  $N$  Samples zu Zeitpunkten  $t_n = nT_S$  mit  $n = 0 \dots N - 1$ , die  $N$  Frequenzpunkte  $f_k = kf_1 = kfs/N$  mit  $k = 0 \dots N - 1$  liefert. Ein zeitdiskretes Signal hat ein periodisches Spektrum (Wiederholspektrum), zu einem diskreten (Linien-)Spektrum gehört ein periodisches Zeitsignal! Ein zeitlich begrenztes Signal mit unendlich vielen angehängten Nullen ist ein Fall für die DTFT.

**... gibt das gesamte Basisband zurück.**

Bei reellwertigen Zeitsignalen ist die FFT im Bereich  $f_S/2 \dots fs$  gespiegelt zu  $0 \dots f_S/2$  und liefert keine neuen Erkenntnisse. Meist plottet man daher nur die ersten  $N/2$  Frequenzpunkte.

**... liefert im Allgemeinen komplexwertige Ergebnisse.**

Man muss daher vor dem Plotten Betrag / Phase oder Real- und Imaginärteil der Fouriertransformierten berechnen. Vergisst man das, plottet Python nur den Realteil, die Warnung „Casting complex values to real discards the imaginary part“ in der Konsole kann man leicht übersehen.

**... muss für periodische und Gleichsignale skaliert werden, um physikalisch korrekte Werte zu liefern.**

Die *Energie* eines Signals bzw. Signalausschnitts bleibt bei der DFT erhalten. Soll mit Hilfe der DFT näherungsweise die CFT bzw. DTFT eines *aperiodischen* Signals (= Energiesignal, da die *Energie* definiert ist) bestimmt werden (wie in dieser Aufgabe), darf *nicht* skaliert werden.

Bei einem periodischen oder einem Gleichsignal interessiert die *Leistung* (die zeitliche Ausdehnung und daher die Energie sind ja unendlich groß), die man erhält indem man die Energie pro Zeitabschnitt bestimmt.

Die Fouriertransformierte *periodischer* Signale muss daher bei Matlab und Python durch die Anzahl  $N$  der Samples (= die zeitliche Ausdehnung) dividiert werden, damit Amplituden / Leistungen im Zeit- und Frequenzbereich übereinstimmen.<sup>1</sup> Wurde das Signal durch Zeropadding auf die Gesamtlänge  $L$  verlängert, darf trotzdem nur mit der ursprünglichen Anzahl  $N$  skaliert werden - andernfalls würde die Amplitude der Fouriertransformierten des Signals ja von der Anzahl der angehängten Nullen abhängen.

Der Plot, der von Listing 3.4 erzeugt wird, ist nicht falsch; man kann ihn interpretieren als DFT

- eines Gleichsignals, die noch nicht mit der Anzahl der Datenpunkte skaliert wurde oder
- der Impulsantwort mit vier Abtastpunkten. Die vier Frequenzpunkte sind korrekt, allerdings wäre eine Interpolation zwischen den Punkten praktisch.

In Abb. M3.4 sind die o.g. Korrekturen integriert: Macht man die Anzahl der FFT-Punkte größer als die Anzahl der Datenpunkte (Zeile 10), ergänzen Python und Matlab die fehlenden Punkte automatisch durch Nullen (*Zeropadding*). Mit der `abs()` Funktion wird der Betrag aus Real- und Imaginärteil berechnet, in Zeile 11 wird der Vektor `k` über die Hälfte der Frequenzpunkte gebildet, in Zeile 12 wird die erste Hälfte von  $|H[k]|$  über diesen Bereich geplottet.

---

<sup>1</sup>Vorsicht: Bei anderen Softwarebibliotheken werden u.U. andere Skalierungen verwendet.

## M3.4 Frequenzauflösung der DFT → A3.4

### a) Frequenzauflösung der DFT

Die Frequenzauflösung  $\Delta f$  ist die Breite eines Frequenzintervalls der DFT; sie ist der Kehrwert der Länge des Zeitfensters  $T_0$  (Abtastperiode x Anzahl der Messpunkte),  $\Delta f = 1/T_0 = 1/(NT_S) = f_S/N$ .

Diese maximale Frequenzauflösung wird nur erreicht, wenn das Signal mit einem Rechteckfenster „ausstanzt“. Andere Fenster, die das Zeitsignal an den Rändern sanfter ausblenden, haben geringeren Amplitudenfehler und Leckeffekt, dafür aber eine höhere -3dB Bandbreite  $B = m\Delta f$  mit  $m = 0,9$  (Rechteckfenster) . . .  $m = 3,7$  (Flattop-Fenster).

### b) Abtastfrequenz

Eine Frequenzauflösung von  $\Delta f = 1$  Hz entspricht einem Zeitfenster  $T_0 = 1/\Delta f = 1$  s. Mit  $T_0 = N/T_S = N/f_S$  folgt  $f_S = N/T_0 = 2048$  Hz. Die höchste Frequenz, die von der DFT dargestellt wird, ist  $f_S/2 = 1024$  Hz. Das ist auch die höchste relevante Frequenzkomponente eines mit  $f_S$  abgetasteten reellwertigen Signals, oberhalb von  $f_S/2$  wiederholt sich das Spektrum.

### c) DFT eines Seismogramms

Die maximal erfassbare Frequenz ist  $f_S/2$ , die bestmögliche Frequenzauflösung erhält man mit maximaler Anzahl von Messpunkten  $N_{max} = 2^{15}$ , das resultierende Messfenster hat die Länge  $T_{mess} = N_{max}/f_S$ .

Relevante Daten wurden über einen Zeitraum von ca.  $T_{Data} = 80$  min = 4800 s aufgezeichnet, diesen Zeitraum muss also das Messfenster der DFT erfassen.

Bei  $N_{max} = 2^{15} = 32\,768$  und  $f_S = 25,6$  Hz ist das Messfenster nur  $T_0 = 32\,768/25,6$  Hz = 1280 s lang, erfasst also nur einen Bruchteil der Daten. Um das gesamte Messfenster zu erfassen, darf man höchstens mit

$$f_{S,max} = N_{max}/T_{Data} = 32\,768/4800 \text{ s} = 6,8 \text{ Hz}$$

abtasten. Die nächstkleinere einstellbare Abtastfrequenz ist dann  **$f_S = 6,4$  Hz**. Die Frequenzauflösung beträgt dann  $\Delta f = f_S/N = 0,2$  mHz, die maximal darstellbare Frequenz ist  $f_S/2 = 3,2$  Hz.

Da das gefilterte Seismogramm oberhalb von 1 Hz keine Frequenzkomponenten enthält, wäre eine effizientere Kombination  $f_{S2} = 3,2$  Hz, die maximal darstellbare Frequenz  $f_{S2}/2 = 1,6$  Hz und  $N_{min} \geq T_{Data}f_{S2} = 15360$ . Die nächstmögliche Anzahl von Abtastwerten wäre mit  $N = 2^{14} = 16384$  nur halb so groß wie vorher.

### d) Störfrequenz

Der Abstand der Frequenzpunkte beträgt  $\Delta f = f_S/N = 0,78$  mHz, der Frequenzindex  $k = 620$  entspricht daher einer Signalfrequenz von  $f_k = k\Delta f = kf_S/N = 0,48$  Hz. Solche Störfrequenzen werden von großen Maschinen wie z.B. von Sägewerken erzeugt.

## M3.5 Rechenaufwand für DFT und FFT → A3.5

### a) DFT-Länge der 9. Symphonie

Die erforderliche Länge der DFT ist identisch mit der Anzahl der Datenpunkte je Kanal,  $N_{DFT} = T_0 f_S = 4440 \text{ s} \cdot 44100 \text{ Hz} = 195\,804\,000$ . Die Frequenzauflösung hängt nur mit der Länge des Datenfensters zusammen,  $\Delta f = 1/T_0 = 1/4440 \text{ s} = 225 \mu\text{Hz}$ . Aus den Koeffizienten der DFT ließe sich theoretisch mit der inversen DFT das Zeitsignal perfekt rekonstruieren, da im Messfenster alle Datenpunkte enthalten sind (Energiesignal).

### b) Rechenaufwand DFT / FFT

Für jeden Kanal muss eine DFT mit  $N_{DFT}$  Punkten berechnet werden, insgesamt wären daher mit (3.7)  $N_{MAC,DFT} = 2N_{DFT}^2 = 7,67 \cdot 10^{16}$  reelle Multiplikationen, für eine FFT mit (3.8) immer noch  $N_{MAC,FFT} = 2N_{DFT} \log_2 N_{DFT} = 2,16 \cdot 10^{10}$  reelle Multiplikationen erforderlich.

Die Anzahl der Frequenzpunkte ist bei DFT und FFT identisch, FFTs benötigen aber aufgrund effizienterer Algorithmen wesentlich weniger Rechenoperationen!

### c) Rechenzeit auf FPGA

Das FPGA hat eine theoretische Gesamtrechenleistung von

$R_{MAC} = 512 \cdot 500 \text{ MMAC/s} = 256 \text{ GMAC/s}$ . Damit ließen sich DFT bzw. FFT theoretisch berechnen in

$$T_{DFT} = N_{MAC,DFT}/R_{MAC} = 3,0 \cdot 10^5 \text{ s} = 3,5 \text{ d}$$

$$T_{FFT} = N_{MAC,FFT}/R_{MAC} = 0,084 \text{ s}.$$

In der Realität würde die meiste Zeit für das Speichern und Laden der großen Datenmengen benötigt, hier sollte nur gezeigt werden, welche Rechenzeiteinsparung mit der FFT möglich ist.

### d) Rechenaufwand für Real-Time FFT

Für Real-Time Signalverarbeitung müssen im zeitlichen Mittel  $f_S$  Samples pro Sekunde und Kanal verarbeitet werden. Ein Fenster mit  $N_{FFT}$  Samples hat eine zeitliche Länge  $T_{Mess} = N_{FFT}/f_S$  und benötigt für die FFT  $N_{MAC} = N_{FFT} \log_2 N_{FFT}$  reellwertige Rechenoperationen (Multiply-Accumulate, MAC). Damit werden im Mittel

$$R_{MAC} = \frac{N_{MAC}}{T_{Mess}} = \frac{N_{FFT} \log_2 N_{FFT}}{N_{FFT}/f_S} = f_S \log_2 N_{FFT} \quad (\text{pro Kanal})$$

Rechenoperationen pro Sekunde benötigt.

	$N_{MAC}$	$T_{mess}$	$R_{MAC}$
$N_{FFT}$	$N_{FFT} \log_2 N_{FFT}$	$N_{FFT}/f_S$	$f_S \log_2 N_{FFT}$
<b>512</b>	4608	11,6 ms	$397 \cdot 10^3$
<b>8192</b>	106496	185,8 ms	$573 \cdot 10^3$

Tab. M3.1: Anzahl der Rechenoperationen für eine FFT in Abhängig von der Anzahl der Datenpunkte ( $f_S = 44,1 \text{ kHz}$ , Aufgabe 3.5)

Angeblich wurde die Laufzeit von CDs und der dafür erforderliche Durchmesser so festgelegt, dass die gesamte 9. Symphonie gerade noch auf eine CD passt. Vermutlich ist das aber nur eine gut erfundene Geschichte:

<http://gizmodo.com/5729864/why-the-cd-is-74-minutes-long>.

## M3.6 DFT periodischer Signale mit Python → A3.6

Benutzen Sie auch das Notebook **DFT-Periodische-Signale-ML**) mit der Musterlösung.

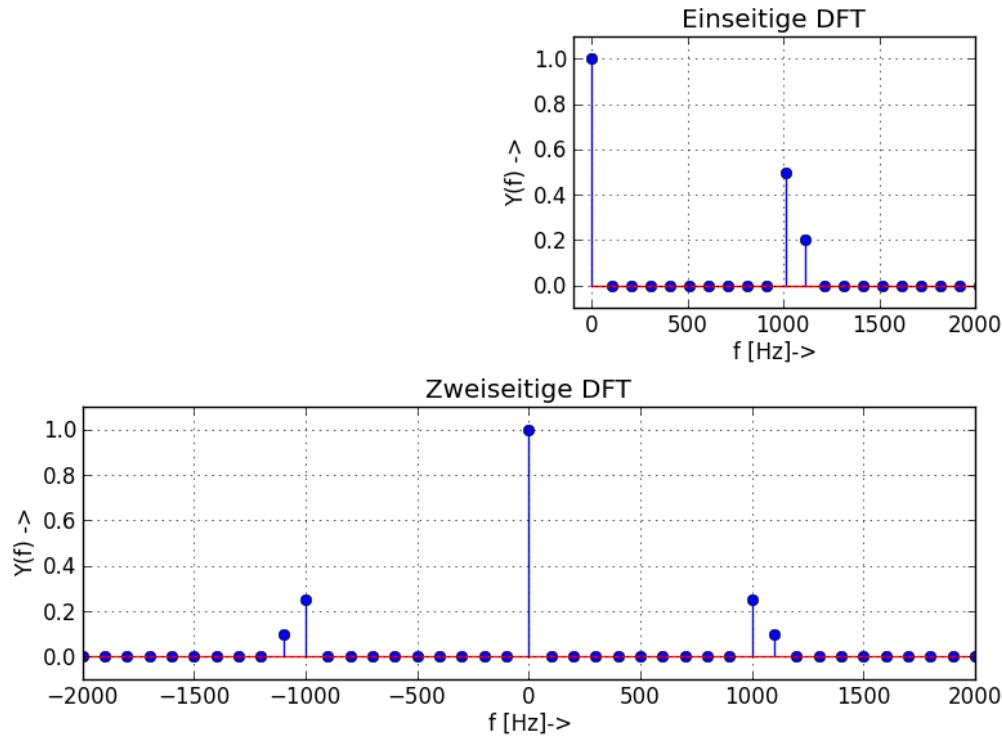


Abb. M3.5: Erwartetes ein- und zweiseitiges Spektrum zu Aufgabe 3.6

a) Das **erwartete Spektrum** des *periodischen zeitkontinuierlichen* Signals

$$y(t) = 1 + 0,5 \sin(2\pi t \cdot 1 \text{ kHz}) + 0,2 \cos(2\pi t \cdot 1,1 \text{ kHz})$$

ist ein *aperiodisches Linienspektrum* (= *diskrete Linien*). Die Koeffizienten  $a_k, b_k$  der *Fourierreihe* entsprechen der Amplitude der Cosinus- und Sinusfunktionen, aus denen sich leicht der Amplitudenbetrag  $A_k = \sqrt{a_k^2 + b_k^2}$  oder die Leistung  $P_k = A_k^2/2$  der  $k$ -ten Spektralkomponenten (Linie) berechnen lässt:

$$\begin{aligned} y(t) &= a_0/2 + \sum_{k=1}^{\infty} a_k \cos 2\pi k f_1 t + b_k \sin 2\pi k f_1 t = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k f_1 t} \\ \circ \bullet \quad Y(f) &= a_0/2 + \sum_{k=1}^{\infty} a_k (\delta(f - kf_1) + \delta(f + kf_1)) + j b_k (\delta(f - kf_1) - \delta(f + kf_1)) \end{aligned}$$

Die Grundfrequenz (größter gemeinsamer Teiler der beiden Frequenzen) ist hier  $f_1 = 100$  Hz, dementsprechend sind die Fourierkoeffizienten  $a_0 = 2$ ,  $b_{10} = 0,5$  und  $a_{11} = 0,2$ , der Index bezeichnet dabei die Nummer der Oberwelle.

Aufgetragen als einseitiges ( $f \geq 0$ ) Betragsspektrum  $|Y(f)|$  ergeben sich Linien bei  $f_0 = 0$  (DC),  $f_{10} = 1000$  Hz und  $f_{11} = 1100$  Hz mit den Amplitudenbeträgen  $A_0 = 1$ ,  $A_{10} = 0,5$  und  $A_{11} = 0,2$ .

### Anmerkung:

Bei reellen Eingangssignalen kann man die Darstellung als *komplexe Fourierreihe*,  $f(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk t}$ , leicht von der reellen Fourierreihe ableiten (nicht gefragt), da dann gilt

$$c_k = \frac{a_k - j b_k}{2} = c_{-k}^* \quad \text{und} \quad c_0 = \frac{a_0}{2}$$

Die negativen Koeffizienten ( $k < 0$ ) liefern also für reelle Eingangssignale keine zusätzlichen Informationen und werden daher meist nicht dargestellt. Um trotzdem eine korrekte Gesamtleistung bei einseitiger Darstellung ( $k \geq 0$ ) zu bekommen, müssen die Koeffizienten  $c_k$  für  $k > 0$  verdoppelt werden;  $c_0$  hat bereits den korrekten Wert.

- b) Die **Simulation** von Listing 3.2 mit angepassten Signalfrequenzen liefert zunächst ein verschmiertes Spektrum („Leckeffekt“), in dem man das erwartete Spektrum nur mit etwas Phantasie erkennt. Die Spektrallinien bei  $f_a$  und  $f_b$  lassen sich nicht unterscheiden.
- c) Der **Leckeffekt** kann durch *kohärente Abtastung* vermieden werden, d.h. indem man die Länge  $T_{mess}$  des Messfensters als ganzzahliges Vielfaches der Grundperiode (hier:  $T_1 = 1/f_1 = 10$  ms) wählt,  $T_{mess} = kT_1$ . Im einfachsten Fall führt man die FFT über *eine* Grundschwingung durch und wählt  $T_{mess} = T_1 = 10$  ms. Abtastperiode und  $N_{FFT}$  müssen passend dazu gewählt werden, damit  $T_S = T_{mess}/N_{FFT} = kT_1/T_{mess}$  erfüllt wird.

Ein Array mit passenden Punkten für die Frequenzachse erhält man in Matlab mit  $f = \text{linspace}(-f_S/2, f_S/2 - f_S/NFFT, NFFT)$ . In Python kann man das Gleiche mit der Hilfsfunktion `scipy.fft.freq(NFFT, T_S)` erreichen.

- d) Eine **Verbesserung der Frequenzauflösung** ist nur möglich durch Verlängerung des Messfensters mit  $\Delta f = 1/T_{mess} = 1/(N_{FFT}T_S)$ . Das ist möglich durch *mehr Abtastpunkte* bei gleichbleibender Abtastrate, also z.B. mit  $N_{FFT} = 200 \Rightarrow \Delta f = 1/(200T_S) =$ .

Erhöht man die Abtastrate bei gleichbleibender Anzahl von Abtastpunkten, z.B. mit  $T_S = 1e-5$ , verringert sich sogar die Frequenzauflösung, da das Messfenster dann kürzer wird.

- e) Für eine **kohärente FFT mit  $2^m$  Punkten** muss die Abtastrate so angepasst werden, dass das Messfenster wieder ein ganzzahliges Vielfaches der Periodenlänge ist, z.B. mit  $T_{mess} = 1e-2$ ;  $N_{FFT} = 128$ ;  $T_S = T_{mess} / N_{FFT}$ ;  $f_S = 1/T_S$ . Diese Bedingung ist nicht immer zu erreichen, da oft die Abtastrate durch die verwendete Hardware vorgegeben ist.

## f) Leistungsberechnung / Amplitudenskalierung der DFT

### Analytische Rechnung

Im Zeitbereich gilt:

$$\begin{aligned} P &= \frac{1}{T_1} \int_{t=0}^{T_1} y^2(\tau) d\tau = \frac{1}{T_1} \int_{t=0}^{T_1} (1 + 0,5 \sin(2\pi \cdot 10f_1\tau) + 0,2 \cos(2\pi \cdot 11f_1\tau))^2 d\tau \\ &= \frac{1}{T_1} \int_{t=0}^{T_1} 1^2 + 0,25 \sin^2(2\pi \cdot 10f_1\tau) + 0,04 \cos^2(2\pi \cdot 11f_1\tau) d\tau \\ &= 1 + 0,125 + 0,02 = A_0^2 + \frac{A_{10}^2}{2} + \frac{A_{11}^2}{2} = 1,145 \end{aligned}$$

Beim Integrieren über eine Periode  $T_1$  liefern alle Terme  $\sin 2\pi k f_1 t \cos 2\pi l f_1 t$  den Wert 0, für  $k \neq l$  auch die Terme  $\sin 2\pi k f_1 t \sin 2\pi l f_1 t$  und  $\cos 2\pi k f_1 t \cos 2\pi l f_1 t$ . Übrig bleiben daher nur die  $\sin^2$  und  $\cos^2$  Terme, deren Integral sich vergleichsweise leicht berechnen lässt mit

$$\int_{t=0}^{T_1} \cos^2(2\pi k f_1 \tau) d\tau = \frac{\tau}{2} + \frac{\sin(2\pi k f_1 \tau) \cos(2\pi k f_1 \tau)}{2 \cdot 2\pi k f_1} \Big|_{\tau=0}^{T_1} = T_1/2.$$

Für periodische Signale ist der Vergleich von Zeit- und Frequenzbereich einfach, da hier nur die Koeffizienten der Sinus- und Kosinusterme und der Betrag der Spektrallinien verglichen werden müssen.

$$P = \frac{a_0^2}{2} + \sum_{k=1}^{\infty} \frac{a_k^2 + b_k^2}{2} = A_0^2 + \sum_{k=1}^{\infty} \frac{A_k^2}{2}$$

Falls man übrigens vergessen hat, wie das mit dem Integrieren funktioniert, hilft einem das Python Modul **sympy** für symbolische Rechnungen auf die Sprünge:

---

```

1 from sympy import integrate, cos, pi
2 from sympy.abc import f, x
3 print integrate(cos(x*f)**2, x)
4 print integrate(cos(x*f)**2, x).expand(mul = True)
5 print integrate(cos(2*pi*10*f*x)**2, (x, 0, 1/f))

```

---

Lst. M3.2: Symbolische Integration mit SymPy

### Simulation

Im Zeitbereich nähert man das Integral durch Summation, also mit

$$P = \frac{1}{T_1} \int_{t=0}^{T_1} y^2(\tau) d\tau \approx \frac{1}{T_1} \sum_{n=0}^{T_1/T_S} y^2[n] T_S. \quad (\text{M3.1})$$

In Python / Matlab muss für die korrekte Skalierung das Amplitudenspektrum durch die Anzahl der DFT - Punkte  $N_{FFT}$  dividiert werden. Bei einseitigen Spektren über  $0 \dots f_S/2$  müssen mit Ausnahme des DC-Werts (Bin 0) außerdem die Amplituden verdoppelt werden (s.o.).

**Achtung:** Python beginnt Arrays beim Index 0 (im Gegensatz zu Matlab). Zum Ausgleich ist bei `a = arange(0,NFFT)` oder `a = x[0:NFFT]` der Wert NFFT bzw. das Element `x[NFFT]` *nicht* mit eingeschlossen.

In Python lässt sich die Leistung im Frequenzbereich am elegantesten über das Skalarprodukt aus **Sy** und des dazu konjugierten und transponierten Arrays bestimmen (siehe Notebook **DFT-Periodische-Signale-ML**). Die Matlab Syntax ist hier eleganter (wenn auch schwieriger zu verstehen): **Sy'** konjugiert und transponiert **Sy**, **Sy \* Sy'** berechnet das Skalarprodukt (da die Arrays transponiert zu einander sind). Aufgrund von Rundungsfehlern hat das Ergebnis einen minimalen imaginären Anteil, den man ggf. noch beseitigen muss, wenn das Ergebnis weiterverarbeitet werden soll.

Alternativ (langsamer) kann man die Leistung auch über `sum(abs(Sy)**2)` (Python) bzw. `sum(abs(Sy).^2)` (Matlab) bestimmen. Die Syntax bewirkt jeweils eine elementweise Quadrierung.

## M3.7 + Kohärente DFT einer Rechteckschwingung → A3.7

- a) Die ersten Fourierkoeffizienten der CT Funktion lauten:

$$c_0 = 0, \quad c_1 = 2A/\pi \approx 0,63A, \quad c_2 = 0, \quad c_3 = 2A/3\pi \approx 0,42A.$$

b) Fourierkoeffizienten der Rechteckschwingung (optionale Herleitung):

$$\begin{aligned}
 c_k &= \frac{1}{T_1} \int_0^{T_1} u(t) e^{-j2k\pi f_1 t} dt = \frac{A}{T_1} \left[ \int_0^{T_1/2} e^{-j2k\pi f_1 t} dt - \int_{-T_1/2}^0 e^{-j2k\pi f_1 t} dt \right] \\
 &= \frac{A}{T_1} \left[ \frac{e^{-j2k\pi f_1 t}}{-j2k\pi f_1} \Big|_0^{T_1/2} - \frac{e^{-j2k\pi f_1 t}}{-j2k\pi f_1} \Big|_{-T_1/2}^0 \right] \\
 &= -\frac{A}{j2k\pi} \left[ e^{-j2k\pi f_1 t} \Big|_{t=T_1/2} - 1 - 1 + e^{-j2k\pi f_1 t} \Big|_{t=-T_1/2} \right] = -\frac{A}{jk\pi} [e^{-jk\pi} - 1] \\
 &= \begin{cases} 0 & \text{für gerade } k \\ \frac{2A}{jk\pi} = -j\frac{2A}{k\pi} \approx 0,637 \frac{A}{k} e^{-j\pi/2} & \text{für ungerade } k \end{cases}
 \end{aligned}$$

c) Abgetastete Rechteckfunktion

Die abgetastete Funktion erhält man durch

$$u[n] = \begin{cases} -A & \text{für } -1/2 \leq nT_S/kT_1 < 0 \\ +A & \text{für } 0 \leq nT_S/kT_1 < 1/2 \end{cases}$$

Die Koeffizienten der DFT sind gegeben durch:

$$S[k] = \frac{1}{N} \sum_{n=0}^{N-1} u[n] e^{-j2\pi kn/N}$$

Bei einer Abtastung mit  $f_S = 4$  kHz benötigt man mindestens  $N = 4$  Samples für eine kohärente DFT:

$$\begin{aligned}
 S[0] &= \frac{A}{4} (e^0 + e^0 - e^0 - e^0) = 0 \\
 S[1] &= \frac{A}{4} (e^0 + e^{-j\pi/2} - e^{-j\pi} - e^{-j\pi 3/2}) = \frac{A}{4} (1 - j + 1 - j) = \frac{A}{\sqrt{2}} e^{-j\pi/4} = 0,707 A e^{-j\pi/4} \\
 S[2] &= \frac{A}{4} (e^0 + e^{-j\pi} - e^{-j2\pi} - e^{-j3\pi}) = 0 \\
 S[3] &= S[-1] = S^*[1] = \frac{A}{4} (e^0 + e^{-j3\pi/2} - e^{-3j\pi} - e^{-j\pi 9/2}) = \frac{A}{\sqrt{2}} e^{+j\pi/4}
 \end{aligned}$$

d) Die **Abweichungen in Amplitude und Phase** im Vergleich zur CFT sind eine Folge von **Aliasing** (Kap. 8): Bei einer Abtastung mit 4 kHz können keine Frequenzkomponenten  $\geq 2$  kHz dargestellt werden. Sind im ursprünglichen Signal höherfrequente Komponenten enthalten (wie in der Rechteckschwingung), werden diese auf niedrigerfrequente Komponenten abgebildet und verfälschen diese. Erhöht man die Abtastfrequenz, reduziert sich hier der Fehler durch Aliasing, da immer mehr Oberwellen korrekt erfasst werden und die Amplituden der „zurückgefalteten“ Aliaskomponenten mit  $1/k$  abnehmen. In Abb. M3.6b kann man gut erkennen, dass die Phase bei Grund- und niedrigen Oberschwingungen näher an den aus der CFT erwarteten  $\pi/2$  liegt. Die Abweichungen der Amplitude sind zu gering, um im Plot eindeutig eine Verbesserung erkennen zu können.

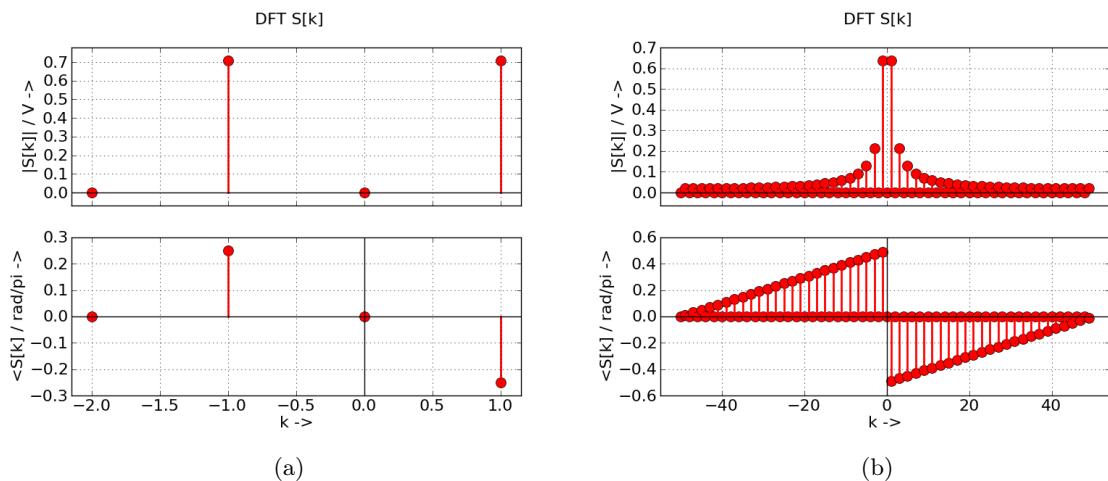


Abb. M3.6: Betrag und Phase der DFT  $S[k]$  für  $f_S = 4000$  Hz und  $N = 4$  (a) bzw.  $f_S = 100$  kHz und  $N = 100$  (b) zu A3.7b

## M4 WIN: Fensterung periodischer und stationärer Signale

### M4.1 Allgemeine Fragen → A4.1

- a) Das ursprüngliche abgetastete Signal  $x[k]$  hat unendlich viele Datenpunkte, sein Spektrum  $X(e^{j\Omega})$  erhält man daher über die DTFT. Es ist periodisch mit  $f_S$  und ist diskret oder kontinuierlich, je nachdem ob  $x[k]$  periodisch ist oder nicht.
- b) Nach der **Multiplikation mit der Fensterfunktion  $w[n]$**  (überall 0 außer für  $n \in [0, N - 1]$ ) hat das Signal  $x_w[k] = x[k] \cdot w[k]$  nur noch  $N$  relevante Datenpunkte (der Rest ist Null) und ist damit nicht mehr periodisch, die Anzahl der Punkte ist aber immer noch unendlich. Das Spektrum des resultierenden Signals  $X_w(e^{j\Omega})$  wird daher ebenfalls mit der DTFT bestimmt, es ist kontinuierlich und periodisch mit  $f_S$ . Im Frequenzbereich berechnet man es über die Faltung von  $X(e^{j\Omega})$  mit  $W(e^{j\Omega})$ ,  $X_w(e^{j\Omega}) = X(e^{j\Omega}) *_N W(e^{j\Omega})$ .

Bei einem rechteckförmigen Fenster ist die Transformierte der Fensterfunktion  $W(e^{j\Omega})$  eine periodische si-Funktion (Dirichlet-Kernel), deren Hauptzipfel eine Breite von im Optimalfall (periodisches Signal mit kohärenter Abtastung und Rechteckfenster) ist die Frequenzauflösung  $1/Nf_S$ , Diskrete Spektrallinien des ursprünglichen Signals werden also mit  $W(e^{j\Omega})$  „verschmiert“.

die Transformierten anderer Fensterfunktionen sehen ähnlich aus.

- c) Das **Herausschneiden von  $N$  Abtastwerten** aus der unendlich ausgedehnten Folge  $x[k]$  entspricht der Multiplikation mit einer Fensterfunktion  $w[k]$  (im einfachsten Fall: rechteckförmig) im Zeitbereich,  $x_N[k] = x[k] \cdot w[k]$ .  $w[k]$  ist überall 0 außer für  $k \in [0, N - 1]$ . Im Frequenzbereich muss das Spektrum  $X(e^{j\Omega})$  des ursprünglichen Signals  $x[k]$  mit der Transformierten der Fensterfunktion  $W(e^{j\Omega})$  gefaltet werden,  $X_N(e^{j\Omega}) = X(e^{j\Omega}) *_N W(e^{j\Omega})$ .

Beschränkt man sich jetzt auf  $N$  Datenpunkte, ist die Transformierte die DFT, die als Blocktransformation numerisch bestimmt werden kann und  $N$  Frequenzpunkte liefert,  $X_N[k] = X_w(\Omega = \frac{2k\pi}{N})$ , man muss also  $X_N(e^{j\Omega})$  in der Frequenzebene abtasten. als Näherung für das Spektrum von  $x[k]$  zu bestimmen,

### M4.2 Fourier-Analyse mit Rechteck-Fensterung → A4.2

- a) Spektrum des gefensterten Signals

$$s(t) = A \cos(2\pi f_i t) \quad \text{---} \bullet \quad S(f) = \frac{A}{2} [\delta(f - f_0) + \delta(f + f_0)]$$

oder, über Eulersche Identität und Verschiebungssatz hergeleitet:

$$s(t) = A \cos(2\pi f_i t) = \frac{A}{2} [e^{j2\pi f_0 t} + e^{-j2\pi f_0 t}] \quad \text{---} \bullet \quad S(f) = \frac{A}{2} [\delta(f - f_0) + \delta(f + f_0)]$$

Frequenzantwort der Fensterfunktion  $w(t)$ :

$$w(t) = \text{rect}\left(\frac{T - T_1/2}{T_1}\right) \quad \text{---} \bullet \quad W(f) = T_1 \text{ si}(\pi f T_1) \quad \underbrace{e^{-j2\pi f T_1 / 2}}_{\text{Verzögerung um } T_1/2}$$

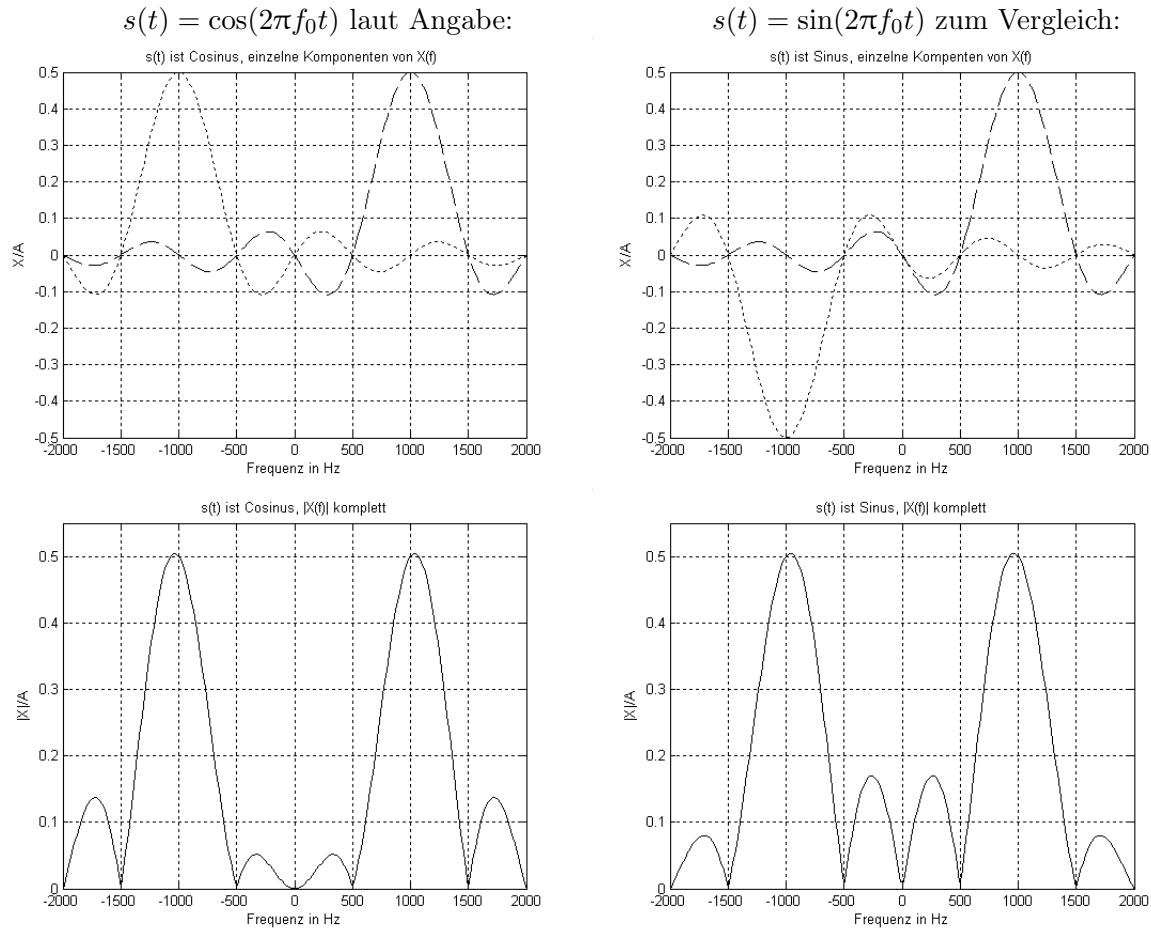
$$x(t) = w(t) \cdot s(t) \quad \text{---} \bullet \quad X(f) = W(f) * S(f)$$

$$\Rightarrow X(f) = \frac{T_1 A}{2} \left\{ e^{-j\pi(f-f_0)T_1} \operatorname{si}[\pi(f-f_0)T_1] + e^{-j\pi(f+f_0)T_1} \operatorname{si}[\pi(f+f_0)T_1] \right\}$$

**b) Darstellung des Betragsspektrums von  $|X(f)|$**

Die Bestimmung des Betragsspektrums fällt deutlich einfacher, wenn man sieht, dass die Verzögerung um  $T_1/2$  des Rechteckfensters im Spektrum  $W(f)$  lediglich eine lineare Phase  $e^{-j2\pi f T_1/2}$  bewirkt. Wählt man daher ein Rechteckfenster  $w'(t)$  gleicher Länge, das symmetrisch zur  $y$ -Achse ist (gerade Funktion) mit  $w'(t) = \operatorname{rect}(T/T_1)$ , erhält man eine rein reelle Frequenzantwort  $W'(f)$  mit  $|W'(f)| = |W(f)|$  und damit auch ein rein reelles Spektrum  $X'(f)$ . Da  $T_1 = 2T_0$  ist, schneiden sowohl  $w(t)$  als auch  $w'(t)$  zwei ganze Perioden von  $s(t)$  mit identischer Startphase aus, daher sind auch die Betragsspektren der aperiodischen Signale identisch:

$$|X(f)| = |X'(f)| = X'(f) = \frac{T_1 A}{2} \{ \operatorname{si}[\pi(f-f_0)T_1] + \operatorname{si}[\pi(f+f_0)T_1] \}$$



```

1 % Spektrum eines zeitlich begrenzten Cosinus-Signals mit
2 % Zeitfenster T1 in s und Signalfrequenz f0
3 T1=0.002; f0=1000;
4 % Vorzeichen V=1 für Cosinus, V=-1 für Sinus
5 V=1;
6 f=linspace(-2000,2000,200);
7 X1=0.5*(sin(pi*(f-f0)*T1+eps)./(pi*(f-f0)*T1+eps));
8 X2=V*0.5*(sin(pi*(f+f0)*T1+eps)./(pi*(f+f0)*T1+eps));
9 %
10 plot(f,X1,'k-{}-',f,X2,'k:');
11 axis([-2000,2000,-0.5,0.5]); grid;
12 xlabel('Frequenz in Hz'); ylabel('X/A');
13 %
14 figure;
15 plot(f,abs(X1+X2),'k');
16 axis([-2000,2000,0,0.55]); grid;
17 xlabel('Frequenz in Hz'); ylabel('|X|/A');

```

Lst. M4.1: Matlab-Code zu Aufgabe 4.2

c) Näherungsweise Bestimmung von  $|X(f)|$  mit Radix-2 FFT

$f_S$  groß genug wählen, um Leckeffekt zu minimieren, dann Zero-Padding um genügend Frequenzpunkte für  $X(f)$  zu erhalten (und um auf  $N = 2^r$  Punkte für eine effiziente Radix-2 FFT-Berechnung zu kommen)

d) Eine FFT von  $s(t)$  ohne Störeffekte

wäre nur möglich mit  $N \rightarrow \infty$  Punkten der FFT, da  $s(t)$  ein unendlich ausgedehntes Zeitsignal ist!  
Das entspricht dann einer DTFT mit einer unendlichen Punktzahl, die nicht numerisch bestimmt werden kann.

Durch Fensterung [  $s(t) \rightarrow x(t)$  ] und Abtastung einer ganzzahligen Anzahl von Perioden kann mit Hilfe der DFT bzw. FFT aus  $x(t)$  das Spektrum von  $s(t)$  bestimmt werden. Voraussetzung dafür ist, dass die periodische Fortsetzung des Signalausschnitts  $x(t)$  identisch mit  $s(t)$  ist.

### M4.3 DFT mit Hann-Fenster → A4.3

Das Hann-Fenster ist definiert über

$$w_{Hann,M}[n] = w_{rect,M+1}[n] \cdot \frac{1}{2} \left( 1 - \cos \frac{2\pi n}{M} \right) = \sin^2 \frac{2\pi n}{M} \text{ mit } n = 0 \dots M$$

- a) Die **Spektralfunktion** des Hann-Fensters erhält man durch Fouriertransformation (DTFT!) der Zeitfunktion mit (C.34):

$$w_{Hann,M}[n] \circledcirc W_{Hann,M}(e^{j\Omega}) = W_{rect,M+1}(e^{j\Omega}) * \frac{1}{2} \left( 1 - \frac{1}{2} (\delta(\Omega - \Omega_M) + \delta(\Omega_M - \Omega)) \right)$$

- b)  $s_1[n] = \cos 2\pi n / 8$ : Die Cosinus-Funktion ist periodisch mit 8 Punkten, bei einer Fensterlänge von  $N_{DFT} = 32$  passen dementsprechend genau 4 Perioden in das Fenster. Wie viele Frequenzpunkte erhalten Sie, die ungleich Null sind? Wie müssen Sie die Amplitude skalieren?
- c)  $s_2[n] = \cos 2\pi n / 8 + \cos 2\pi n / 10$ : Welche Frequenzauflösung erhalten Sie?
- d) Plotten Sie Fenster und gefensterte Signale im Zeit- und Frequenzbereich.

Anmerkung: Auch die Ableitungen der Hann-Fensterfunktion an den Endpunkten sind Null, bei einer Taylorreihenentwicklung ist daher erst der 3. Term ungleich Null. Damit kann man erklären, dass die Side Lobes steil mit -18 dB/Okt oder -60 dB/Dek abfallen.

### M4.4 DFT mit Bartlett-Fenster → A4.4

# M5 FIL: Einfache digitale Filter und FIR-Filterentwurf

## M5.1 Filterspezifikationen → A5.1

- a) Die **logarithmischen Spezifikationen** lassen sich leicht aus den linearen Spezifikationen berechnen: Eine Schwankung der Ausgangsamplitude im Durchlassbereich von maximal  $\pm 10\%$  entspricht beim FIR-Filter  $\pm \delta_{DB,FIR} = 0,1$  und beim IIR-Filter  $\delta_{DB,IIR} = 0,2$ .

$$\text{IIR: } A_{DB,IIR} = -20 \log_{10}(1 - \delta_{DB,IIR}) = 1,94 \text{ dB}$$

$$\begin{aligned} \text{FIR: } A_{DB,FIR} &= 20 \log_{10}(1 + \delta_{DB,FIR}) - 20 \log_{10}(1 - \delta_{DB,FIR}) \\ &= 0,828 \text{ dB} & + 0,915 \text{ dB} & = 1,74 \text{ dB} \end{aligned}$$

- b) Auch für die **Berechnung der linearen Spezifikationen** aus den logarithmischen Spezifikationen  $A_{DB} = 2 \text{ dB}$  und  $A_{SB} = 20 \text{ dB}$  müssen FIR und IIR-Filter separat betrachtet werden: Im Sperrband gilt für beide Filter

$$\delta_{SB} = 10^{-A_{SB}/20} = 10^{-1} = 0,1.$$

Im Durchlassband gilt für das IIR-Filter mit (5.2)

$$\delta_{DB,IIR} = 1 - 10^{-A_{DB}/20} = 1 - 10^{-0,1} = 0,206$$

und für das FIR-Filter mit (5.3) und der Näherung  $\frac{1}{1+x} \approx 1-x$

$$\begin{aligned} \delta_{DB,FIR} &= \frac{10^{A_{DB}/20} - 1}{10^{A_{DB}/20} + 1} = 0,115 \\ &\approx 10^{A_{DB}/40} - 1 = 0,122. \end{aligned}$$

Die Näherung für  $\delta_{DB,FIR}$  ist schon relativ gut, wird für kleinere Werten von  $A_{DB}$  aber noch besser.

Das Equiripple-Filter in Abb. M5.1 benötigt die Ordnung  $N = 26$ , das Chebychev-Filters (Typ 2) die Ordnung  $N = 5$  um die vorgegebenen Spezifikationen zu erfüllen (ermittelt mit Hilfe der Befehle `remezord()` bzw. `cheb2ord()` und durch Ausprobieren).

- c) **Ausgangsleistung bzw. -spannung im Durchlass- und Sperrband:**

Um die Schwankung der Ausgangsleistung bzw. -spannung zu berechnen, kann man zunächst wie im vorigen Punkt die linearen Abweichungen der Amplitude  $\delta_{DB[A]}$  und der Leistung  $\delta_{DB[P]}$  berechnen und mit den Nennwerten multiplizieren. Eine Nennleistung von  $P_{nenn} = 80 \text{ W}$  an  $R = 8 \Omega$  entspricht einer nominellen Ausgangsamplitude von

$$\hat{u}_{nenn} = \sqrt{2PR} = \sqrt{160 \text{ W} \cdot 8 \Omega} = 35,78 \text{ V.}$$

Wie üblich muss bei Leistungen (quadrierten Signalen) mit  $10 \log_{10} P$  und bei Amplituden mit  $20 \log_{10} A$  gerechnet werden:

$$\text{Amplitudenvariation: } \delta_{DB,IIR}[A] = 1 - 10^{-A_{DB}/20} = 1 - 10^{-2/20} = 0,206$$

$$\text{Leistungsvariation: } \delta_{DB,IIR}[P] = 1 - 10^{-A_{DB}/10} = 1 - 10^{-2/10} = 0,369$$

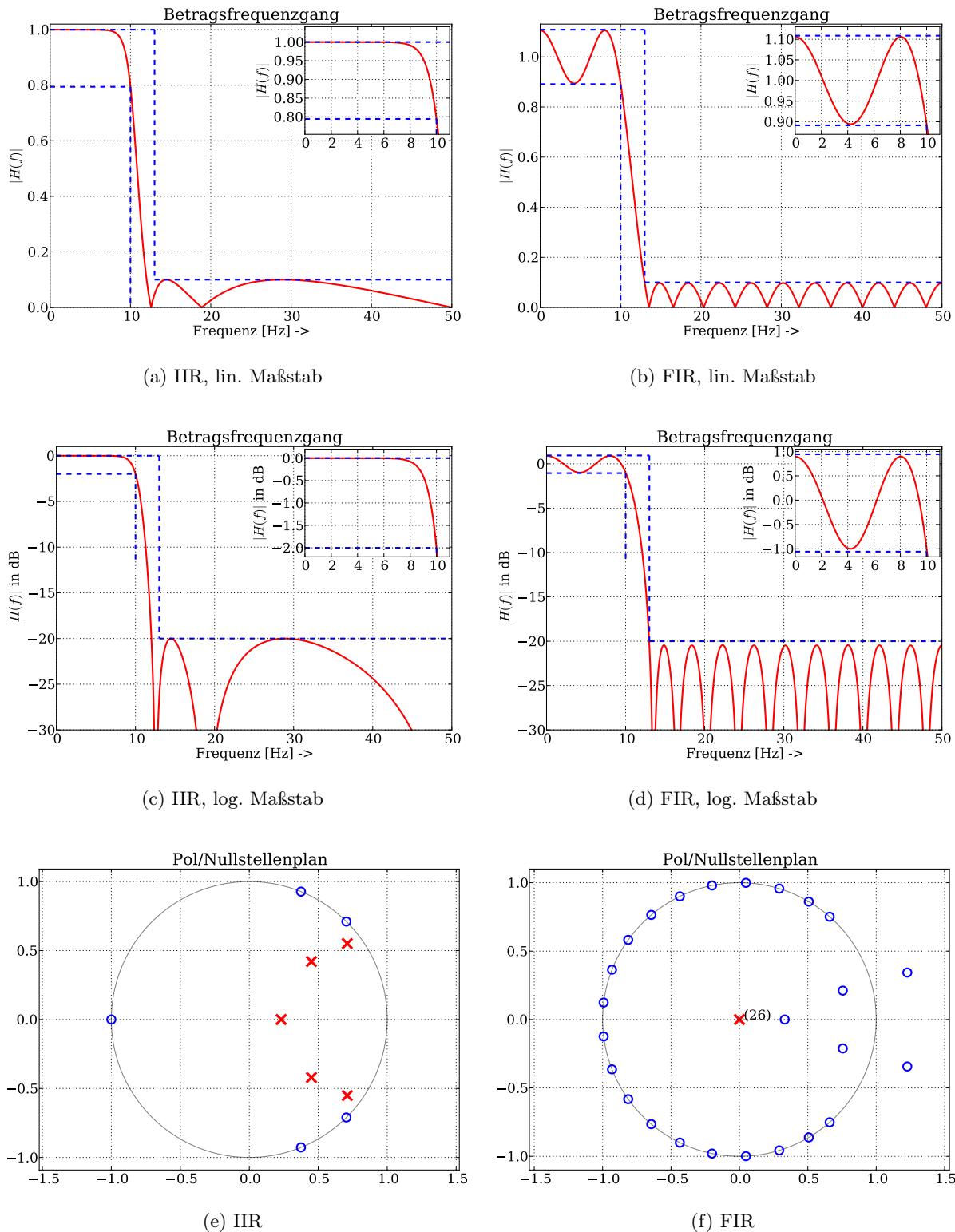


Abb. M5.1: Frequenzgänge und Spezifikationen im linearen und log. Maßstab sowie P/N-Diagramme für IIR- und FIR-Filter zu Aufgabe 5.1

Damit können Ausgangsamplitude und -leistung beim IIR-Filter im Durchlassband variieren zwischen:

$$\text{Amplitude: } \hat{u}_{IIR} = \hat{u}_{nenn} \dots \hat{u}_{nenn} (1 - \delta_{DB,IIR[A]}) = 35,78 \dots 28,41 \text{ V}$$

$$\text{Leistung: } P_{IIR} = P_{nenn} \dots P_{nenn} (1 - \delta_{DB,IIR[P]}) = 80 \dots 50,48 \text{ W}$$

Bei einem FIR-Filter gilt analog:

$$\text{Amplitudenvariation: } \pm \delta_{DB,FIR[A]} = \pm \frac{10^{A_{DB}/20} - 1}{10^{A_{DB}/20} + 1} = \pm 0,115$$

$$\text{Leistungsvariation: } \pm \delta_{DB,FIR[P]} = \pm \frac{10^{A_{DB}/10} - 1}{10^{A_{DB}/10} + 1} = \pm 0,226$$

und damit

$$\text{Amplitude: } \hat{u}_{FIR} = \hat{u}_{nenn} (1 \pm \delta_{DB,FIR[A]}) = 39,90 \dots 31,665 \text{ V}$$

$$\text{Leistung: } P_{FIR} = P_{nenn} (1 \pm \delta_{DB,FIR[P]}) = 98,08 \dots 61,92 \text{ W}$$

Wesentlich einfacher ist die Rechnung, wenn man in der logarithmischen Darstellung bleibt:

Bei einem FIR-Filter bedeutet  $A_{DB} = 2 \text{ dB}$  dass die Ausgangsleistung im Durchlassband ausgehend von  $0 \text{ dB} \cong 80 \text{ W}$  um ca.  $\pm 1 \text{ dB} \cong 10^{\pm 1/10}$  schwanken kann.

### Sperrband:

Im Sperrband sind die Spezifikationen für IIR- und FIR-Filter identisch definiert. Den Bereich der **Ausgangsspannungen bzw. -leistungen** berechnet man ansonsten genauso wie im Durchlassband ausgehend von  $P_{nenn} = 80 \text{ W}$  und  $\hat{u}_{nenn} = 35,78 \text{ V}$ :

$$\text{Max. Amplitude: } \delta_{DB[A]} = 10^{-A_{SB}/20} = 10^{-20/20} = 0,1$$

$$\text{Max. Leistung: } \delta_{DB[P]} = 10^{-A_{SB}/10} = 10^{-20/10} = 0,01$$

Damit können Ausgangsamplitude und -leistung bei beiden Filtern im Sperrband variieren zwischen:

$$\text{Amplitude: } \hat{u} = 0 \dots \hat{u}_{nenn} \delta_{SB[A]} = 0 \dots 3,58 \text{ V}$$

$$\text{Leistung: } P = 0 \dots P_{nenn} \delta_{SB[P]} = 0 \dots 0,8 \text{ W}$$

### d) Filterspezifikationen:

Die Kante des Durchlassbands entspricht der max. Signalfrequenz  $f_{sig,max} = 20 \text{ Hz}$ , die des Sperrbands der niedrigsten Störfrequenz  $f_{br} = 50 \text{ Hz}$ . Die Eckfrequenzen für den Filterentwurf sind dementsprechend:

	$f_{DB}$	$f_{SB}$	$F_{DB}$	$F_{SB}$	$\Omega_{DB}$	$\Omega_{SB}$
<b>Formel</b>	—	—	$f_{DB}/f_S$	$f_{SB}/f_S$	$2\pi f_{DB}/f_S$	$2\pi f_{SB}/f_S$
<b>Wert</b>	20 Hz	50 Hz	0,02	0,05	0,126	0,314

Tab. M5.1: Eckfrequenzen für den Filterentwurf zu Aufgabe 5.1d

Die normalisierte Frequenz  $F$  ist einheitenlos, die normalisierte Winkelfrequenz  $\Omega$  hat die Einheit rad / s / Sample.

Am Eingang des Filters ist die Amplitude des Störsignals im Worst Case 100 mal so groß wie die des Nutzsignals, am Ausgang soll sie nur noch ein Tausendstel so groß sein. Daher muss das Störsignal unterdrückt werden um  $A_{SB} = 10^2 \cdot 10^3 = 10^5 \hat{=} 100$  dB.

Das Nutzsignal soll um maximal  $\pm 1\%$  verändert werden, das entspricht

$$\delta_{DB,FIR} = 0,01 \text{ und } A_{DB,FIR} = 20 \log_{10} \frac{1 + \delta_{DB}}{1 - \delta_{DB}} = 0,174 \text{ dB}$$

$$\delta_{DB,IIR} = 0,02 \text{ und } A_{DB,IIR} = -20 \log_{10} 1 - \delta_{DB} = 0,175 \text{ dB}$$

Ein Equiripple FIR-Filter benötigt laut Filterentwurfstool mindestens eine Ordnung  $N = 143$ , ein elliptisches IIR Filter  $N = 7$  und Chebychev-Filter  $N = 9$  um die Anforderungen zu erfüllen.

## M5.2 Amplitudengang linearphasiger Filter → A5.2

### a) Filtereigenschaften

Das Filter ist linearphasig, weil die Koeffizienten symmetrisch sind (gerade Symmetrie, ungerade Anzahl Koeffizienten → Typ 1).

Das Filter ist ein Halbbandfilter, weil jeder zweite Koeffizient außer dem mittleren gleich Null ist.

### b) Nullstelle(n) bei $f_S/2$

Beweis für eine Nullstelle bei  $f_S/2$  durch Einsetzen von  $z = -1$  in  $H(z)$ :

$\Rightarrow H(z = -1) = 0$ . Um zu zeigen, dass bei  $z = z_0$  eine  $m$ -fache Nullstelle vorliegt, muss  $H(z)$  per Polynomdivision durch  $(z - z_0)^m$  dividiert werden, ohne dass ein Rest bleibt:

$$H(z) = \frac{-1 + 5z^{-2} + 8z^{-3} + 5z^{-4} - z^{-6}}{8} = \frac{-z^6 + 5z^4 + 8z^3 + 5z^2 - 1}{8z^6}$$

Hier soll auf eine doppelte Nullstelle bei  $z = -1$  untersucht werden, daher wird das Zählerpolynom durch  $(z + 1)^2 = z^2 + 2z + 1$  dividiert und untersucht, ob die Division ohne Rest aufgeht:

$$\begin{array}{r} (-z^6 + 5z^4 + 8z^3 + 5z^2 - 1) \div (z^2 + 2z + 1) = -z^4 + 2z^3 + 2z^2 + 2z - 1 \\ \hline z^6 + 2z^5 + z^4 \\ \hline 2z^5 + 6z^4 + 8z^3 \\ \hline -2z^5 - 4z^4 - 2z^3 \\ \hline 2z^4 + 6z^3 + 5z^2 \\ \hline -2z^4 - 4z^3 - 2z^2 \\ \hline 2z^3 + 3z^2 \\ \hline -2z^3 - 4z^2 - 2z \\ \hline -z^2 - 2z - 1 \\ \hline z^2 + 2z + 1 \\ \hline 0 \end{array}$$

Da die Division ohne Rest aufgeht, liegt bei  $z_{0,1} = z_{0,2} = -1$  eine doppelte Nullstelle vor!

c) Eine weitere Nullstelle auf dem Einheitskreis

kann man aus Abb. 5.3a ablesen. Sie liegt ungefähr bei  $f_0 = 0,36f_S$  also bei  $z_{0,3} = e^{j2 \cdot 0,36\pi}$ . Da das System nur reelle Koeffizienten hat, muss es reellwertig sein und daher eine zweite dazu konjugiert-komplexe Nullstelle (= symmetrisch zur  $x$ -Achse) haben bei  $f_0 = -0,36f_S$  ( $z_{0,4} = e^{-j2 \cdot 0,36\pi}$ ). Die Nullstellen sind einfach, da  $|H(e^{j\Omega})|$  an dieser Stelle einen Knick hat,  $H(e^{j2\pi F})$  also das Vorzeichen wechselt. Im Gegensatz dazu sieht man bei  $|H(e^{j2\pi F})|$  bei  $F = 0,5$  (doppelte Nullstelle) keinen Knick.

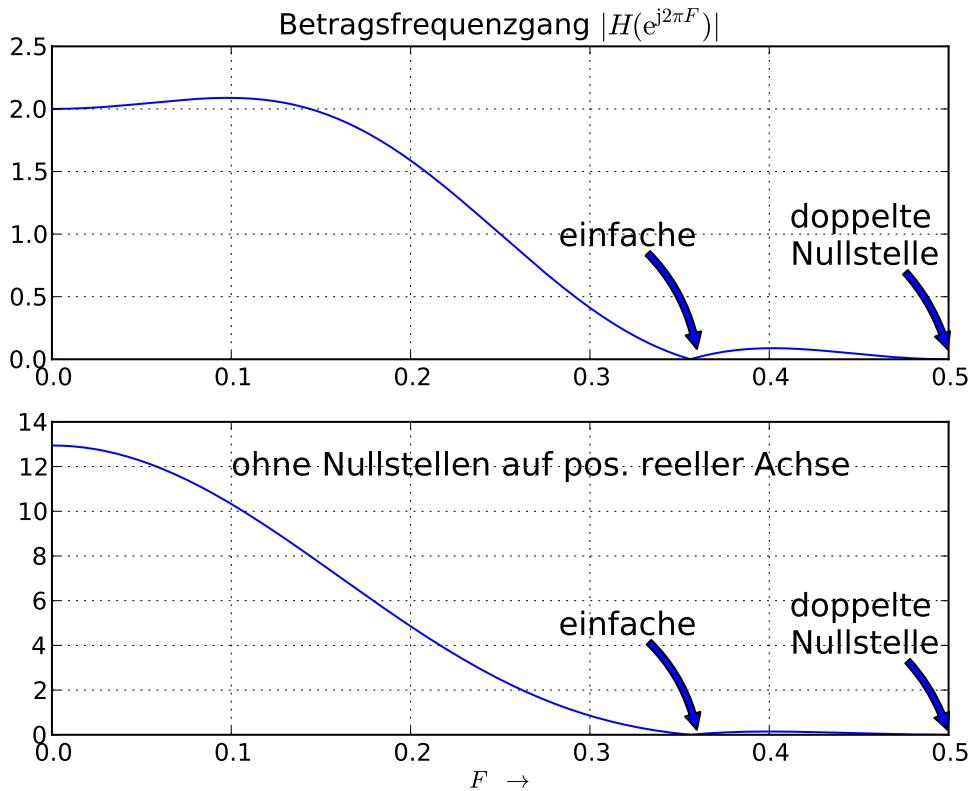


Abb. M5.2: Amplitudengang mit Kennzeichnung der Nullstellen auf dem Einheitskreis zu Aufgabe 5.2

d) Weitere Nullstellen:

Da das Filter  $M = 7$  Koeffizienten hat, hat es die Ordnung  $N = 6$  und muss daher auch 6 Nullstellen haben. Damit muss es noch zwei weitere Nullstellen geben, die in  $H(e^{j2\pi F})$  nicht zu erkennen sind. Da das Filter linearphasig und reellwertig ist, müssen Nullstellen immer konjugiert-komplex und symmetrisch zum Einheitskreis sein. Im Sonderfall  $z = -1$  wird das durch eine Nullstelle erreicht (die hier doppelt platziert ist), die Nullstellen auf dem Einheitskreis sind „symmetrisch zu sich selbst“ und treten daher als Paar auf. Die fehlenden zwei Nullstellen müssen daher auf der reellen Achse liegen („konjugiert-komplex zu sich selbst“) und zwar symmetrisch zum Einheitskreis mit  $z_{0,5} = r$  und  $z_{0,6} = 1/r$ . Sie liegen auf der *positiven* reellen Achse, um einen zu frühen Abfall des DBs zu verhindern (vgl. Abb. M5.2 oben und unten). Lässt man die beiden Nullstellen auf der reellen Achse weg, erhöht sich außerdem die DC-Verstärkung.

e) PN-Diagramm

Die Nullstellen in Tab. M5.2 wurden mit Hilfe des Notebooks **FIL-Linphase\_Filter** ermittelt.

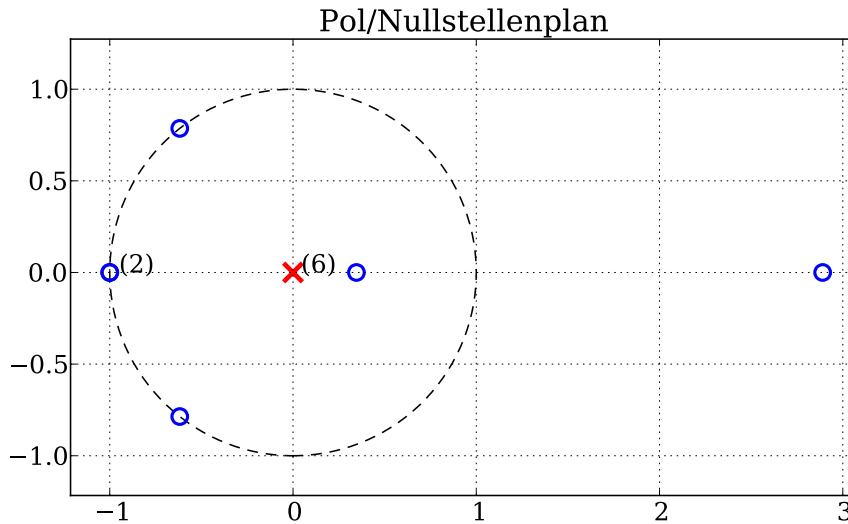


Abb. M5.3: P/N Diagramm zu Aufgabe 5.2

<i>i</i>	$z_{0,i}$
1	$-1,0000+0,0000j = e^{j\pi}$
2	$-1,0000+0,0000j = e^{j\pi}$
3	$-0,6180+0,7862j = e^{0,712j\pi}$
4	$-0,6180-0,7862j = e^{-0,712j\pi}$
5	$0,3460+0,0000j = 0,346 \cdot e^0$
6	$2,8901+0,0000j = 2,8901 \cdot e^0$

Tab. M5.2: Nullstellen des Zählerpolynoms

### M5.3 Maximal-, minimal- und linearphasige Filter → A5.3

#### a) Allgemeine Systemfunktion / Übertragungsfunktionen von $F_1$

Die *Systemfunktion*  $H_1(z)$  erhält man einfach durch Einsetzen von Pol- und Nullstelle:

$$H_1(z) = \frac{z - z_{0,1}}{z - z_{\infty,1}} = \frac{z - r_1 e^{j\Omega_1}}{z} = 1 - r_1 e^{j\Omega_1} z^{-1}$$

•→○  $h_1[n] = \delta[n] - r_1 e^{j\Omega_1} \delta[n-1] = \delta[n] - r_1 \cos \Omega_1 \delta[n-1] - j r_1 \sin \Omega_1 \delta[n-1]$

Man sieht, dass  $H_1(z)$  aufgrund des Pols bei  $z = 0$  **kausal** ist, es enthält also keine „Verfrühung“  $z^{+1}$ .

Offensichtlich ist  $F_1$  **nicht reellwertig**, da die Impulsantwort komplexwertig ist. Im Frequenzbereich kann man das erkennen, da es keine konjugiert-komplexe Nullstelle zu  $z_{0,1}$  gibt. Reellwertig wäre  $F_1$  nur für  $\Omega_1 = 0$  oder  $\Omega_1 = \pi$ , da dann die Nst. auf der reellen Achse liegt und man die Systemfunktion  $H_1(z) = 1 \pm r_1 z^{-1}$  erhält.

Das System hat nur eine einzige Nullstelle, die abhängig davon ob  $r_1$  größer, kleiner oder gleich Null ist, innerhalb oder außerhalb des EKs bzw. auf dem EK liegt. Damit hängt es von  $r_1$  ab, ob das Filter **maximalphasig** oder **minimalphasig** ist.

Da es zu der Nullstelle  $z_{0,1}$  keine am EK gespiegelte Nst. gibt, ist das Filter **nur für  $r = 1$  linearphasig**, da für diesen Spezialfall die Nst. auf dem EK liegt und  $H_1(z) = 1 \pm e^{j\Omega_1} z^{-1}$ . Im Zeitbereich kann man Linearphasigkeit erkennen, indem man untersucht, ob die Impulsantwort / Koeffizienten symmetrisch sind. Bei komplexen Koeffizienten muss gelten:  $b_i = +b_{N-i}^*$  oder  $b_i = -b_{N-i}^*$ , wobei  $N$  die Ordnung des Filters ist.

Den *komplexen Frequenzgang*  $H_1(e^{j\Omega})$  erhält man wie üblich durch Setzen von  $z = e^{j\Omega}$ :

$$H_1(e^{j\Omega}) = 1 - r_1 e^{j\Omega_1} e^{-j\Omega} = 1 - r_1 e^{j(\Omega_1 - \Omega)} = \underbrace{1 - r_1 \cos(\Omega_1 - \Omega)}_{\text{Realteil}} - r_1 \sin(\Omega_1 - \Omega) j \underbrace{j}_{\text{Imaginärteil}}$$

Den *Betragsgang* erhält man mit

$$\begin{aligned} |H_1(e^{j\Omega})| &= \sqrt{\Re\{H_1(e^{j\Omega})\}^2 + \Im\{H_1(e^{j\Omega})\}^2} \\ &= \sqrt{1 - 2r_1 \cos(\Omega_1 - \Omega) + r_1^2 \cos^2(\Omega_1 - \Omega) + r_1^2 \sin^2(\Omega_1 - \Omega)} \\ &= \sqrt{1 + r_1^2 - 2r_1 \cos(\Omega_1 - \Omega)} \end{aligned} \quad (\text{M5.1})$$

und den Phasengang (siehe auch (B.1) und Abb. B.1) mit

$$\angle H(e^{j\Omega}) = \begin{cases} \arctan \frac{\Im\{H(e^{j\Omega})\}}{\Re\{H(e^{j\Omega})\}} & \text{für } \Re\{H(e^{j\Omega})\} > 0 \\ \arctan \frac{\Im\{H(e^{j\Omega})\}}{\Re\{H(e^{j\Omega})\}} + \pi & \text{für } \Re\{H(e^{j\Omega})\} < 0 \end{cases}$$

**Vorsicht:** Die Funktion `atan(x)` in Python / Matlab ist die „normale“ Arcustangens-Funktion (keine eingebaute Fallunterscheidung); verwenden Sie zur Phasenberechnung in Python, Matlab, C, ... stattdessen `arctan2(z)` bzw. `atan2(z)` oder `angle(z)`. Die Fallunterscheidung ist notwendig, da der Wertebereich der Arcustangens-Funktion auf  $\pm\pi/2$  beschränkt ist und den gleichen Wert liefert, unabhängig davon ob Real- und Imaginärteil beide positiv oder negativ sind.

Damit hängt auch zusammen, dass die Phase an einer Nullstelle von  $|H_1(e^{j\Omega})|$  um  $\pi$  springt: Der Betrag von  $H(e^{j\Omega})$  wird nur Null, wenn Real- und Imaginärteil zu Null werden und damit beide das Vorzeichen wechseln (Ausnahme: doppelte Nullstelle).

**Achtung, nicht verwechseln:** Nullstellen von  $H(e^{j\Omega})$  liegen auf dem EK (physikalische Frequenzen), diese Frequenzen werden vom System vollständig unterdrückt. Nullstellen von  $H(z)$  können sich irgendwo in der komplexen Ebene befinden.

Bei dieser Aufgabe gilt dementsprechend:

$$\angle H_1(e^{j\Omega}) = \begin{cases} \arctan \frac{-r_1 \sin(\Omega_1 - \Omega)}{1 - r_1 \cos(\Omega_1 - \Omega)} & \text{für } \Re\{H_1(e^{j\Omega})\} = 1 - r_1 \cos(\Omega_1 - \Omega) > 0 \\ \arctan \frac{-r_1 \sin(\Omega_1 - \Omega)}{1 - r_1 \cos(\Omega_1 - \Omega)} + \pi & \text{für } \Re\{H_1(e^{j\Omega})\} = 1 - r_1 \cos(\Omega_1 - \Omega) < 0 \end{cases} \quad (\text{M5.2})$$

**b) Verlauf von Betrags- und Phasengang von  $F_1$  mit  $\Omega_1 = \pi/3$  und  $r_1 = 2$  (maximalphasiges System)**

Die Systemfunktion und der komplexe Frequenzgang von  $F_1$  lauten:

$$\begin{aligned} H_1(z) &= 1 - 2e^{j\pi/3} z^{-1} = 1 - (1 + j\sqrt{3}) z^{-1} \\ H_1(e^{j\Omega}) &= 1 - 2e^{j(\pi/3 - \Omega)} = 1 - 2 \cos(\pi/3 - \Omega) - j2 \sin(\pi/3 - \Omega) \end{aligned}$$

Da die einzige Nullstelle außerhalb des Einheitskreises liegt, ist das System **maximalphasig**.

Für die Zeichnung werden zunächst mit  $\cos \pi/3 = \frac{1}{2}$  und  $\sin \pi/3 = \sqrt{3}/2$  einige Hilfswerte berechnet (Tab. M5.3): Den Betragsgang berechnet man mit (M5.1) aus dem letzten Unterpunkt.

$\Omega$	$\Re\{H_1\}$	$\Im\{H_1\}$	$ H_1 $	$\angle H_1$
0	0	$-\sqrt{3}$	$\sqrt{3}$	$-90^\circ$
$\pi/3$	-1	0	1	$-180^\circ$
$\pi$	2	$\sqrt{3}$	$\sqrt{7}$	$41^\circ = -319^\circ$

Tab. M5.3: Hilfswerte für Filter F<sub>1</sub> (Aufgabe 5.3b)

Die Berechnung des Phasengangs mit (M5.2) ist ungünstig, da hier aufgrund  $r_1 > 1$  der Realteil über  $\Omega$  das Vorzeichen wechselt und eine Fallunterscheidung getroffen werden muss. Eine bessere Darstellung erhält man nach einer kleinen Umformung:

$$\begin{aligned} H_1(e^{j\Omega}) &= 1 - r_1 e^{j(\Omega_1 - \Omega)} = e^{j(\Omega_1 - \Omega)} \left[ e^{-j(\Omega_1 - \Omega)} - r_1 \right] \\ &= e^{j(\Omega_1 - \Omega)} \left[ \underbrace{\cos(\Omega_1 - \Omega) - r_1}_{\Re\{H_1\} < 0} - \underbrace{\sin(\Omega_1 - \Omega) j}_{\Im\{H_1\}} \right] \\ \Rightarrow \angle H_1(e^{j\Omega}) &= \Omega_1 - \Omega + \arctan \frac{\sin(\Omega_1 - \Omega)}{r_1 - \cos(\Omega_1 - \Omega)} + \pi \end{aligned} \quad (\text{M5.3})$$

In den Plots in Abb. M5.4 sieht man Folgendes:

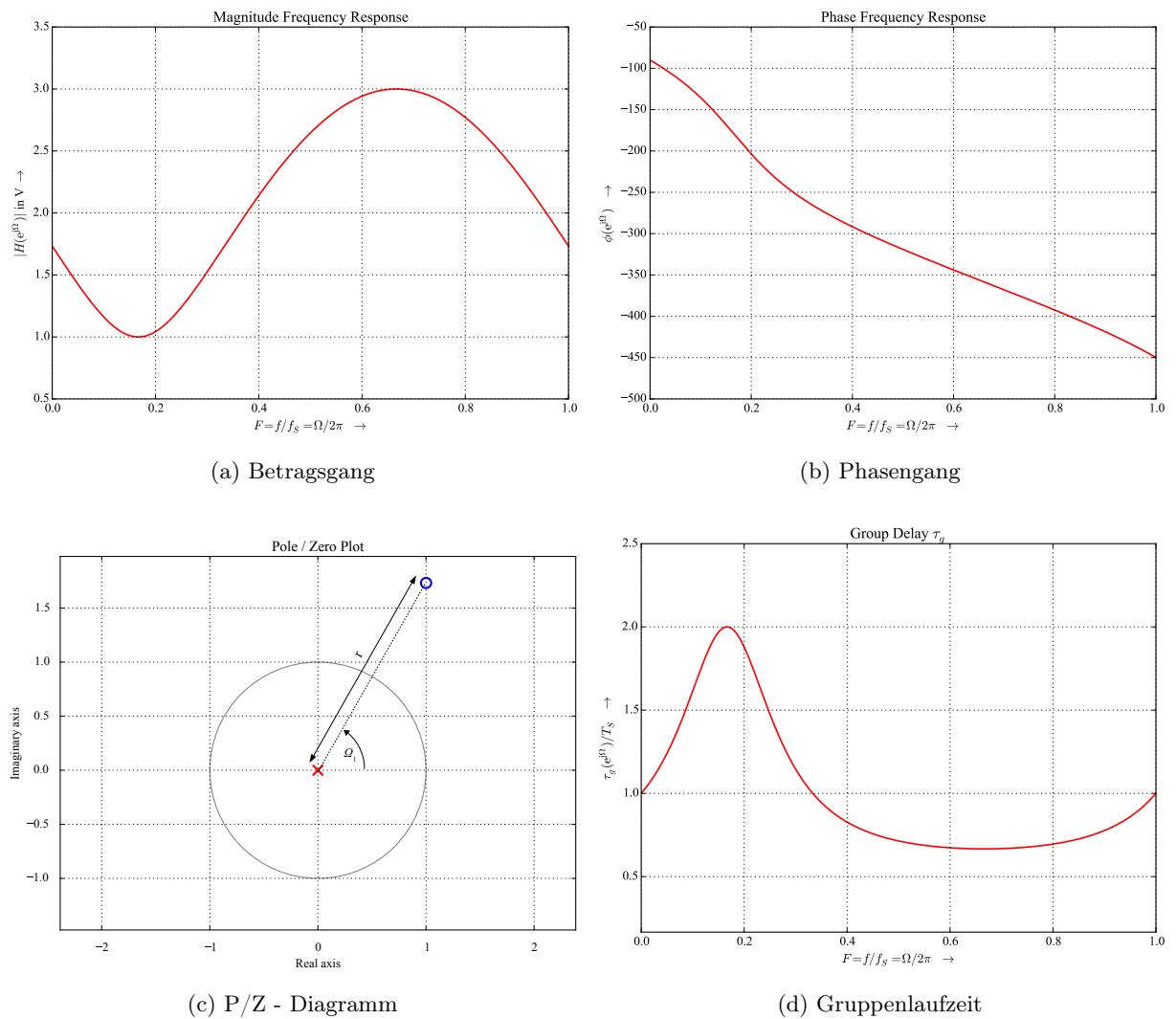
- Bei der Phase kann man nicht zwischen  $+41^\circ$  und  $-319^\circ$  unterscheiden. Da die Phase immer weiter abfällt (aufgrund der Verzögerung des Filters), ist es sinnvoll sie auch so darzustellen (Option `unwrapped` in Python / Matlab) anstatt sie immer umzubrechen, sobald  $\pm 180^\circ \hat{=} \pm \pi$  erreicht sind.
- Die Gruppenlaufzeit erreicht ihr Maximum von  $2T_S$  bei  $F \approx 1/6$  bzw.  $\Omega = \pi/3$ . Wird der Filter z.B. innerhalb einer Regelschleife betrieben, kann eine zu hohe Gruppenlaufzeit / Phasendrehung das System destabilisieren.

### c) Filter F<sub>2</sub> mit am EK gespiegelter Nullstelle (minimalphasiges System)

Die Spiegelung eines Punktes am EK wird mathematisch beschrieben durch  $\tilde{z}_1 = 1/z_1^*$ . Beim Spiegeln eines Punktes am EK bleibt der Winkel unverändert,  $\Omega_2 = \Omega_1$ , der neue Radius ist der Kehrwert des alten,  $r_2 = 1/r_1$ . Damit liegt die Nullstelle von F<sub>2</sub> bei  $z_{0,2} = r_2 e^{j\Omega_2} = r_1^{-1} e^{j\Omega_1}$ .

F<sub>2</sub> ist damit bis auf den Skalierungsfaktor  $c$  bestimmt:

$$\begin{aligned} |H_2(\Omega = \Omega_1)| &= \left| c \left( 1 - r_1^{-1} e^{j(\Omega_1 - \Omega)} \right) \right| \stackrel{!}{=} \left| 1 - r_1 e^{j(\Omega_1 - \Omega)} \right| = |H_1(\Omega = \Omega_1)| \text{ mit } r_1 > 1 \\ \Rightarrow |c| \left| 1 - r_1^{-1} \right| &\stackrel{!}{=} |r_1 - 1| \\ \Rightarrow |c| &= \left| \frac{r_1 - 1}{1 - r_1^{-1}} \right| = \left| \frac{r_1 - 1}{r_1^{-1}(r_1 - 1)} \right| = r_1 \Rightarrow c = \pm r_1 \end{aligned}$$

Abb. M5.4: Plots zu maximalphasigem Filter F<sub>1</sub> (Aufgabe 5.3b)

Wählt man  $c = +r_1$ , erhält man

$$\begin{aligned} H_2(z) &= r_1 (1 - r_1^{-1} e^{j\Omega_1} z^{-1}) = r_1 - e^{j\Omega_1} z^{-1} \\ \Rightarrow H_2(e^{j\Omega}) &= r_1 - e^{j(\Omega_1 - \Omega)} = r_1 - \cos(\Omega_1 - \Omega) - j \sin(\Omega_1 - \Omega) \\ \Rightarrow |H_2(e^{j\Omega})| &= \sqrt{1 + r_1^2 - 2r_1 \cos(\Omega_1 - \Omega)} \end{aligned}$$

Der Betragsgang von F<sub>2</sub> ist für alle Frequenzen gleich wie der von F<sub>1</sub> (vergleichen Sie mit Aufgabe Punkt a.).

Wegen  $r_1 > 1 \Rightarrow r_2 < 1$  liegt die einzige Nullstelle innerhalb des EK, F<sub>2</sub> ist **minimalphasig**. Aus M5.2 sieht man, dass der Realteil von  $H_2$  immer positiv ist und die Fallunterscheidung hier nicht notwendig ist:

$$\angle H_2(e^{j\Omega}) = \arctan \frac{-r_2 \sin(\Omega_1 - \Omega)}{1 - r_2 \cos(\Omega_1 - \Omega)} = -\arctan \frac{\sin(\Omega_1 - \Omega)}{r_1 - \cos(\Omega_1 - \Omega)} \text{ mit } r_2 = 1/r_1$$

d) **Verlauf von Betrags- und Phasengang von  $F_2$  mit  $\Omega_1 = \pi/3$  und  $r_2 = 1/r_1 = 0,5$**

Die Systemfunktion und der komplexe Frequenzgang von  $F_2$  lauten (Skalierungsfaktor  $c$  nicht vergessen):

$$H_2(z) = 2 - e^{j\pi/3}z^{-1} = 2 - \frac{1}{2}(1 + j\sqrt{3})z^{-1}$$

$$H_2(e^{j\Omega}) = 2 - \frac{1}{2}\cos(\pi/3 - \Omega) - \frac{j}{2}\sin(\pi/3 - \Omega)$$

Auch hier werden zunächst ein paar Hilfswerte mit  $\cos \pi/3 = \frac{1}{2}$  und  $\sin \pi/3 = \sqrt{3}/2$  berechnet: In den Plots in Abb. M5.5 sieht man Folgendes:

$\Omega$	$\Re\{H_2\}$	$\Im\{H_2\}$	$ H_2 $	$\angle H_2$
0	3/2	$-\sqrt{3}/2$	$\sqrt{3}$	-30°
$\pi/3$	1	0	1	0°
$\pi$	5/2	$\sqrt{3}/2$	$\sqrt{7}$	19,1°

Tab. M5.4: Hilfswerte für Filter  $F_2$  (Aufgabe 5.3c)

- Der Betragsgang ist der gleiche wie der von Filter  $F_1$  (wie schon im vorigen Unterpunkt bewiesen).
- Die Phasendrehung und deren Variation sind deutlich geringer als bei  $F_1$  (*minimalphasiges* System).
- Daher ist die Gruppenlaufzeit auch deutlich geringer (max.  $0,3 T_S$ ).

e) **Linearphasigkeit des zusammengesetzten Filters F**

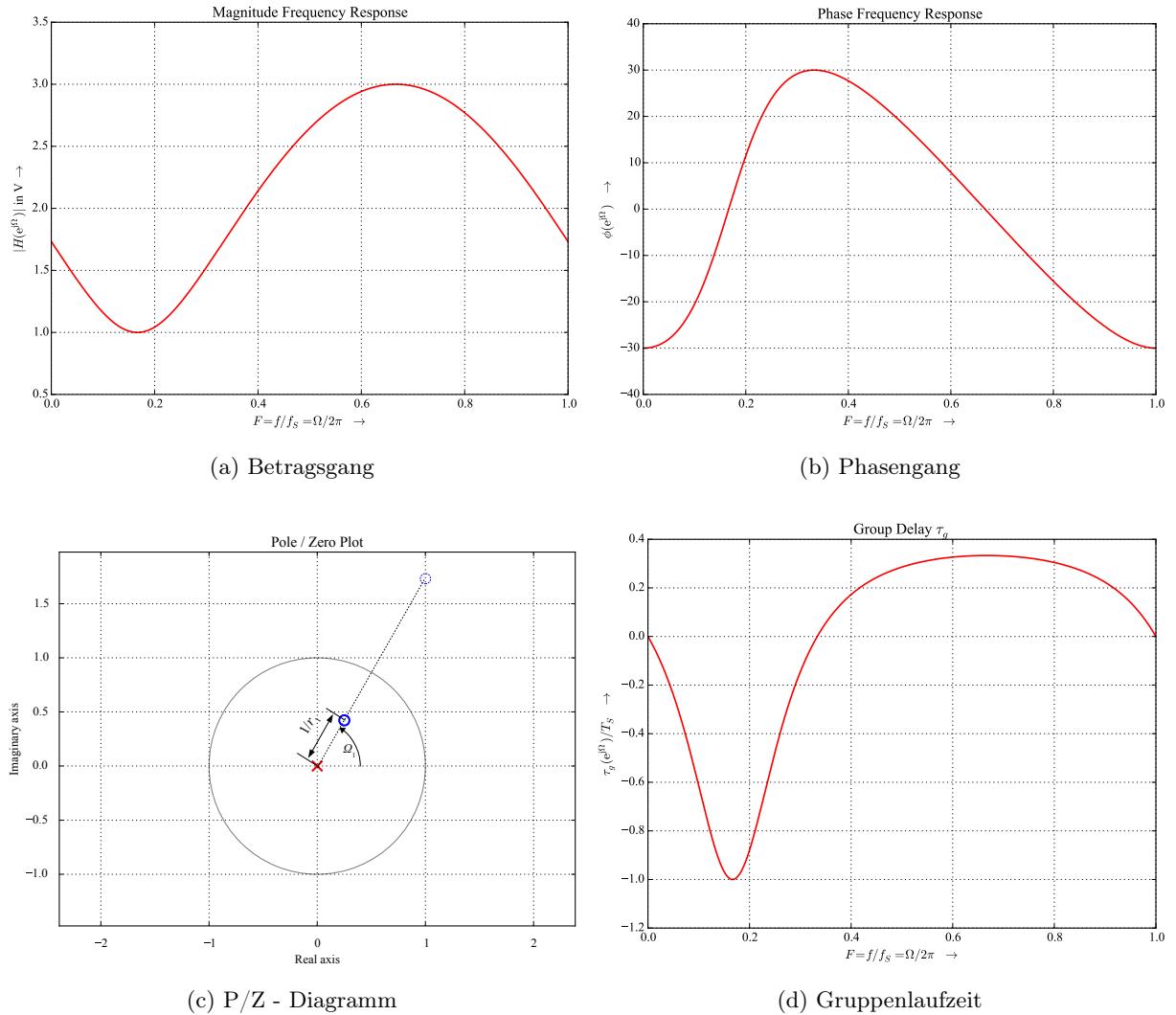
Wir stellen die Systemfunktion von  $H(z) = H_1(z)H_2(z)$  auf und versuchen, einen Vorfaktor so auszuklammern, dass ein symmetrisches Polynom („Spiegelpolynom“) übrigbleibt:

$$\begin{aligned} H(z) &= H_1(z)H_2(z) = (1 - r_1 e^{j\Omega_1} z^{-1})(r_1 + e^{j\Omega_1} z^{-1}) \\ &= r_1 - (r_1^2 + 1)e^{j\Omega_1} z^{-1} + r_1 e^{j2\Omega_1} z^{-2} \\ &= r_1 e^{j\Omega_1} z^{-1} \left[ \underbrace{e^{-j\Omega_1} z^{+1}}_{\tilde{b}_0} - \underbrace{\left( r_1 + \frac{1}{r_1} \right)}_{\tilde{b}_1} + \underbrace{e^{j\Omega_1} z^{-1}}_{\tilde{b}_2} \right] \end{aligned}$$

Diese Zerlegung gelingt,  $\tilde{b}_0 = \tilde{b}_2^*$ , daher kann man das Spiegelpolynom einfach darstellen als Summe aus reellwertigen Sinus<sup>1</sup>- und Cosinustermen:

$$\begin{aligned} H(e^{j\Omega}) &= r_1 e^{j(\Omega_1 - \Omega)} \left[ e^{-j(\Omega_1 - \Omega)} - \left( r_1 + \frac{1}{r_1} \right) + e^{j(\Omega_1 - \Omega)} \right] \\ &= r_1 e^{j(\Omega_1 - \Omega)} \underbrace{\left[ 2 \cos(\Omega_1 - \Omega) - \left( r_1 + \frac{1}{r_1} \right) \right]}_{\text{immer } > 2} \\ &= -r_1 e^{j(\Omega_1 - \Omega)} \left| 2 \cos(\Omega_1 - \Omega) - \left( r_1 + \frac{1}{r_1} \right) \right| \end{aligned}$$

<sup>1</sup>Hier = 0.

Abb. M5.5: Plots zu minimalphasigem Filter  $F_2$  (Aufgabe 5.3d)

(Den Beweis für  $|r + r^{-1}| \geq 2$  finden Sie am Ende der Aufgabe.) Damit lässt sich die Phase einfach ablesen (nicht das negative Vorzeichen vergessen!) als

$$\angle H = \varphi_H = \underbrace{\Omega_1 + \pi}_{\text{konstant}} - \underbrace{\Omega}_{\text{linear}},$$

man sieht sofort dass die Phase linear ist. Die Gruppenlaufzeit ist konstant:

$$\tau_{g,H} = -\frac{\partial \varphi_H}{\partial \omega} = -\frac{\partial \varphi_H}{\partial \Omega} \frac{\partial \Omega}{\partial \omega} = 1 \cdot \frac{1}{f_S} = T_S \text{ mit } \Omega = 2\pi \frac{f}{f_S}$$

Der Betragsgang lässt ebenfalls leicht aus der obigen Form von  $H(e^{j\Omega})$  berechnen, da ja bereits Amplituden- und Phasengang getrennt worden und der Amplitudengang nie das Vorzeichen wechselt:

$$\begin{aligned} |H(e^{j\Omega})| &= r_1 \left| 2 \cos(\Omega_1 - \Omega) - \left( r_1 - \frac{1}{r_1} \right) \right| = (r_1^2 - 1) - 2 \cos(\Omega_1 - \Omega) \\ &= |H_1(e^{j\Omega})| |H_2(e^{j\Omega})| = |H_1(e^{j\Omega})|^2 \end{aligned}$$

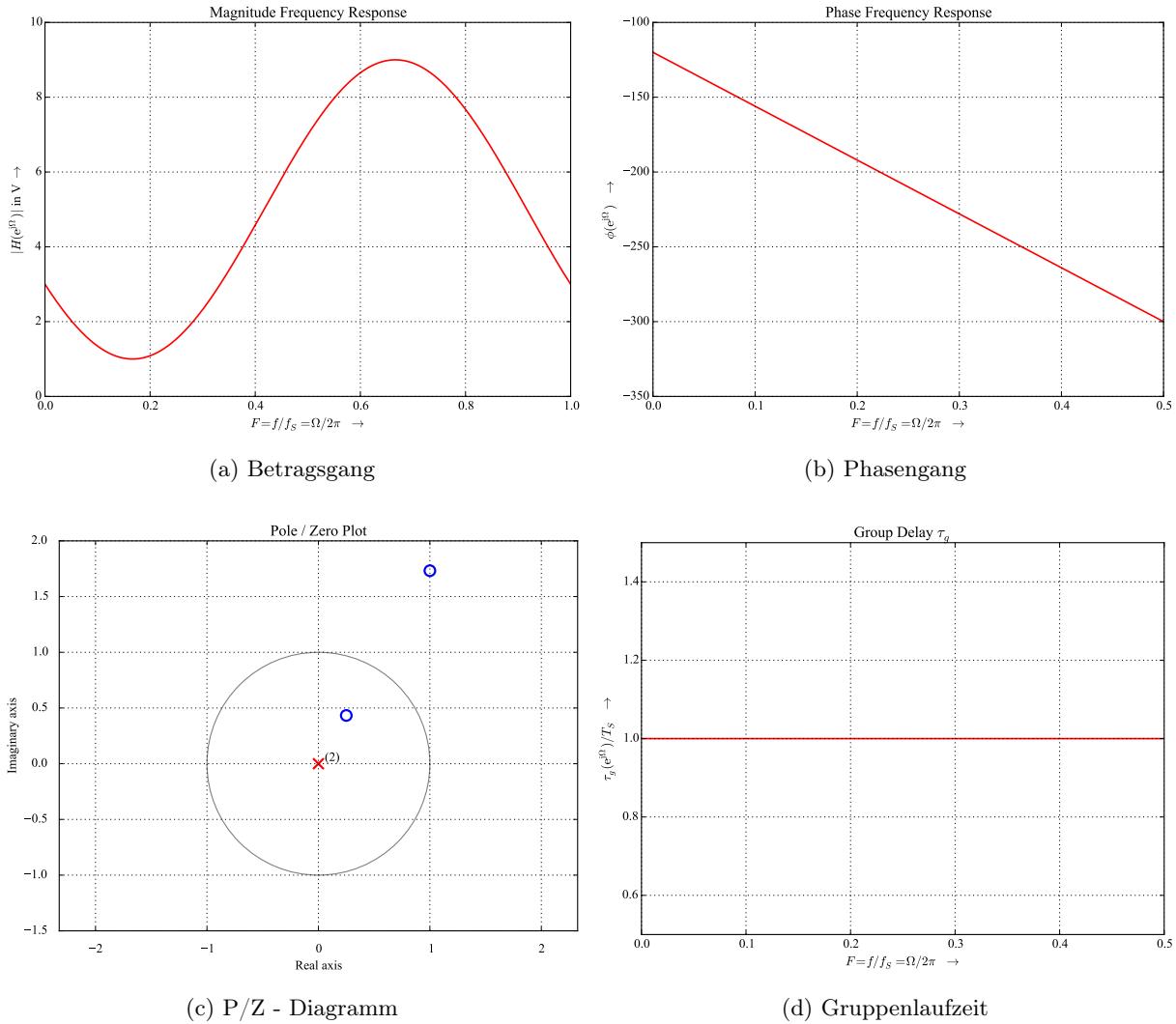


Abb. M5.6: Plots zu linearphasigem Filter F (Aufgabe 5.3e)

Mit den Zahlenwerten aus b)  $r_1 = 2$  und  $\Omega_1 = \pi/3$  und  $\cos \pi/3 = \frac{1}{2}$ ,  $\cos 2\pi/3 = -\frac{1}{2}$  und  $\sin \pi/3 = \sin 2\pi/3 = \sqrt{3}/2$  erhält man:

$$H(z) = 2 - 5e^{j\pi/3}z^{-1} + 2e^{j2\pi/3}z^{-2} = 2 - \frac{5}{2} \left( 1 + j\sqrt{3} \right) z^{-1} + \left( -1 + j\sqrt{3} \right) z^{-2}$$

Zusammenfassend gilt für das linearphasige Filter als Kombination der beiden Teilfilter:

- Der Betragsgang ist das Produkt der beiden Teilbetragsgänge,  $|H(e^{j\Omega})| = |H_1(e^{j\Omega})| |H_2(e^{j\Omega})| = |H_1(e^{j\Omega})|^2$ .
- Die Gesamtphase ist die Summe der Teilphasen,  $\angle H(e^{j\Omega}) = \angle H_1(e^{j\Omega}) + \angle H_2(e^{j\Omega}) = \Omega_1 + \pi - \Omega$ .
- Die Phase fällt linear ab, die Gruppenlaufzeit ist konstant. Würde man die Gruppenlaufzeit dieses linearphasigen mit den entsprechenden minimal- und maximalphasigen Filtern zweiter Ordnung vergleichen (doppelter Pol innerhalb bzw. außerhalb des EK), würde man sehen, dass gilt  $\tau_{g,min} < \tau_{g,lin} < \tau_{g,max}$ .

### f) Implementierung für Filter F

Für komplexwertige Systeme kann man einen Signalflussgraphen genau wie für reellwertige Systeme zeichnen; in Abb. M5.7a ist das Eingangssignal reellwertig; die Koeffizienten, das Ausgangssignal und ein Teil der Signale (fette Linien) komplexwertig.

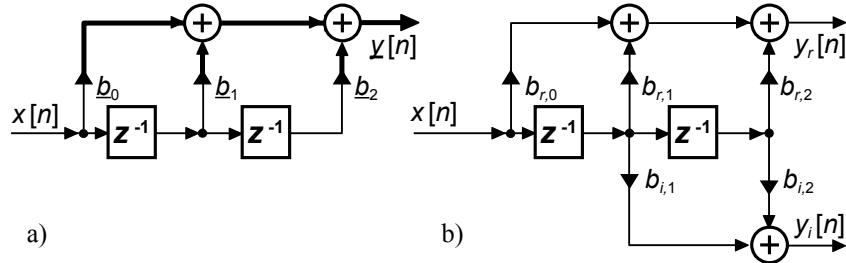


Abb. M5.7: Blockschaubild (a) und Implementierung (b) für Filter F (Aufgabe 5.3e)

Die Koeffizienten lauten:

$$\begin{aligned}\underline{b}_0 &= r \\ \underline{b}_1 &= -(r^2 + 1)e^{j\Omega_1} \\ \underline{b}_2 &= re^{j2\Omega_1}\end{aligned}$$

Bei einer Hardwareimplementierung werden Real- und Imaginärteil meist separat verarbeitet (Ausnahme: Floating-Point Unit mit komplexen MAC-Units), dann benötigt man für Real- und Imaginärteil separate Signalpfade, Koeffizienten, Multiplizierer und Addierer. Das Gesamt-Ausgangssignal ist dann  $\underline{y}[n] = \underline{y}_r[n] + j\underline{y}_i[n]$ . Die Koeffizienten sind hier:

$$\begin{array}{ll} b_{0,r} = \Re\{\underline{b}_0\} = r & b_{0,i} = \Im\{\underline{b}_0\} = 0 \\ b_{1,r} = \Re\{\underline{b}_1\} = -(r^2 + 1) \cos \Omega_1 & b_{1,i} = \Im\{\underline{b}_1\} = -(r^2 + 1) \sin \Omega_1 \\ b_{2,r} = \Re\{\underline{b}_2\} = r \cos 2\Omega_1 & b_{2,i} = \Im\{\underline{b}_2\} = r \sin 2\Omega_1 \end{array}$$

**Beweis** dass  $r + r^{-1} \geq 2$ :

$$\begin{aligned}(r - 1)^2 &\geq 0 \quad \text{für } r > 1 \\ \Rightarrow r^2 - 2r + 1 &\geq 0 \\ \Rightarrow r^2 + 1 &\geq 2r \quad | : r \\ \Rightarrow r + r^{-1} &\geq 2\end{aligned}$$

## M5.4 FIR Halbbandfilter → A5.4

### a) Spezifikationen:

Die Bandmitte liegt bei  $f_S/4 = 6$  kHz. Für ein Halbbandfilter muss die Eckfrequenzen von DB und SB symmetrisch zu  $F = 1/4$  ( $f = f_S/4$ ) liegen, es muss also gelten

$$f_S/4 - f_{DB} \stackrel{!}{=} f_{SB} - f_S/4 \quad \text{oder} \quad f_{DB} \stackrel{!}{=} f_S/2 - f_{SB}.$$

Um diese Bedingung zu erfüllen, muss das Durchlassband von  $f_{DB} = 1$  kHz auf  $f_{DB} = 2$  kHz erweitert werden. Würde man das Sperrband mit  $f_{SB} = 11$  kHz dem Durchlassband anpassen, erhielte man einen Filterentwurf, der schwächer als der ursprüngliche Entwurf ist.

Der vorgegebene Ripple im DB beträgt  $A_{DB} = 0,1$  dB oder  $\delta_{DB} = 10^{A_{DB}/40} - 1 = 5,7 \cdot 10^{-3}$ , der Ripple im SB beträgt  $A_{SB} = 60$  dB oder  $\delta_{SB} = 10^{-A_{DB}/20} = 10^{-3}$ .

Für ein Halbbandfilter müssen auch der Verlauf von  $H(e^{j\Omega})$  im DB und SB punktsymmetrisch zu ( $F = 0,25; H(F = 0,25)$ )einander sein. Das kann im Filterentwurf nur erreicht werden, wenn der zulässige Ripple im DB und im SB identisch sind. Falls das wie hier nicht der Fall ist, muss man einen der Parameter dem anderen anpassen, ohne die ursprüngliche Filterspezifikation zu verletzen. Daher muss hier der Ripple im DB dem im SB (härtere Bedingung) angepasst werden auf  $\delta_{DB} = \delta_{SB} = 10^{-3}$  oder  $A_{DB} = 0,017$  dB.

In Python / Matlab wählt man einen FIR Equiripple - Entwurf mit den obigen Spezifikationen. Überzeugen Sie sich selbst davon, dass jeder zweite Koeffizient (außer dem mittleren) Null ist und das der Betragsverlauf symmetrisch zu  $|H(F = 1/4)|$  ist.

### b) Frequenzgang aus Impulsantwort:

Aus der Impulsantwort lässt sich leicht die Systemfunktion  $H(z)$  ableiten und durch Ausklammern von  $z^{-3}$  in eine Form mit symmetrischen Koeffizienten und Exponenten bringen. Lässt man den Term  $z^{-3}$  weg, erhält man die *akausale* Systemfunktion  $H_{ak}(z)$ . Für diese Systemfunktion gibt es zwar keine Hardwarerealisierung, da die Impulsantwort symmetrisch zu  $n = 0$  ist und somit auch Anteile hat bei  $n < 0$ . Der *Betragsgang* von  $H_{ak}(z)$  ist aber identisch mit dem von  $H(z)$ , da  $z^{-3}$  als Verzögerung lediglich den Phasengang beeinflusst.

$$\begin{aligned} h[n] &= \{0,1; 0; 0,2; 0,3; 0,2; 0; 0,1\} \\ \circlearrowleft \bullet \quad H(z) &= 0,1 + 0,2z^{-2} + 0,3z^{-3} + 0,2z^{-4} + 0,1z^{-6} \\ &= z^{-3}(0,1z^3 + 0,2z + 0,3 + 0,2z^{-1} + 0,1z^{-3}) \\ \Rightarrow H_{ak}(z) &= 0,1z^3 + 0,2z + 0,3 + 0,2z^{-1} + 0,1z^{-3} \\ \Rightarrow H_{ak}(z = e^{j\Omega}) &= 0,1e^{j3\Omega} + 0,2e^{j\Omega} + 0,3 + 0,2e^{-j\Omega} + 0,1e^{-j3\Omega} \\ &= 0,4 \cos \Omega + 0,2 \cos 3\Omega + 0,3 \end{aligned}$$

Beide Cosinusfunktionen nehmen bei  $\Omega = 0$  den Wert 1, bei  $\Omega = \pi/2$  den Wert 0 und bei  $\Omega = \pi$  den Wert -1 an (multipliziert mit dem jeweiligen Koeffizienten). Beide Cosinus-Funktionen sind punktsymmetrisch zu ( $\Omega = \pi/2, 0$ ) und werden um den DC-Wert „angehoben“, der durch den mittleren Koeffizienten 0,3 gegeben ist. Es ist also automatisch die Halbbandbedingung erfüllt mit  $H(\pi/2 - \Omega) + H(\pi/2 + \Omega) = 2H(\pi/2) = 0,6$ .

Es ist also:

- $H_{ak}(\Omega = 0) = H_{ak}(z = 1) = \sum_i h_i$  (gilt immer)
- $H_{ak}(\Omega = \pi/2) = h_{N/2}$ , da alle Cosinusfunktionen, aus denen sich der acausal Amplitudengang zusammensetzt, hier Null werden, so dass nur der mittlere Koeffizient übrigbleibt.
- $H_{ak}(\Omega = \pi) = H_{ak}(z = -1) = -\sum_i h_i + 2h_{N/2}$ , da alle geraden Koeffizienten Null sind außer dem mittleren.

Insgesamt sind also  $H(\Omega = 0) = 0,2 + 0,4 + 0,3 = 0,9$ ,  $H(\Omega = \pi/2) = 0,3$  und  $H(\Omega = \pi) = -0,3$

Mit der Zusatzbedingung  $H(\Omega = \pi) \stackrel{!}{=} 0$  lässt sich ein verbessertes Tiefpassverhalten erreichen, da dann die Frequenzkomponente bei  $f_S/2$  komplett unterdrückt wird. Das erreicht man am einfachsten durch Anpassung des mittleren Koeffizienten:

$$H(\Omega = 0) + \underbrace{H(\Omega = \pi)}_{\stackrel{!}{=} 0} = 2H(\pi/2) \Rightarrow \sum_i h_i = 2h_{N/2}$$

Die Summe aller Koeffizienten *ohne*  $h_{N/2}$  muss also gleich  $h_{N/2}$  sein, hier im Beispiel  $2h_0 + 2h_2 = \mathbf{0,6} \stackrel{!}{=} h_3$

Allgemein:

$$\begin{aligned} h[n] &= \{h_0; 0; h_2; 0; \dots; 0; h_{N/2-1}; h_{N/2}; h_{N/2+1}; 0; \dots; 0; h_{N-2}; 0; h_N\} \\ &= \{h_0; 0; h_2; 0; \dots; 0; h_{N/2-1}; h_{N/2}; h_{N/2-1}; 0; \dots; 0; h_2; 0; h_0\} \\ \circlearrowleft \bullet \quad H_{ak}(z) &= h_{N/2} + \sum_{i=0}^{(N-2)/4} h_{2i} (z^{2i-N/2} + z^{N/2-2i}) \\ \Rightarrow H_{ak}(e^{j\Omega}) &= h_{N/2} + 2 \sum_{i=0}^{(N-2)/4} h_{2i} \cos(2i\Omega) \end{aligned}$$

- c) Skalierung der Koeffizienten so, dass  $|\tilde{H}(\Omega = \pi/2)| = 0,5$

Wie in der vorigen Aufgabe gezeigt, ist

$$\tilde{H}_{ak}(\Omega = \pi/2) = \tilde{h}_{N/2} = 0,5$$

Die Größen mit Tilde gehören zu der Zielübertragungsfunktion. Damit der Zielwert  $|\tilde{H}(\Omega = \pi/2)| = 0,5$  erreicht wird, ohne den Verlauf des Frequenzgangs zu ändern, müssen alle Koeffizienten skaliert werden (gerundet auf drei Stellen):

$$\begin{aligned} \tilde{h}_i &= \frac{\tilde{h}_{N/2}}{h_{N/2}} h_i = \frac{0,5}{0,3} h_i \\ &= \{0,167; 0; 0,333; 0,5; 0,333; 0; 0,167\} \\ \circlearrowleft \bullet \quad \tilde{H}(z) &= 0,167 + 0,333z^{-2} + 0,5z^{-3} + 0,333z^{-4} + 0,167z^{-6} \end{aligned}$$

- d) Wenn der **mittlere Koeffizient Null ist**, entfällt die DC-Komponente aus Aufgabenpunkt b), damit muss  $H(\Omega = 0) + H(\Omega = \pi) = 2H(\Omega = \pi/2) = 0$  sein und es gilt  $H(\Omega = 0) = -H(\Omega = \pi)$ . Solche Filter haben einen Betragsgang, der symmetrisch zur Nullstelle bei  $H(\Omega = \pi/2)$  sind, sie sind also Bandsperren oder Notchfilter (siehe auch Aufgabe 5.6b und Abb. M5.9).
- e) Dass jeder zweite Koeffizient (außer dem mittleren) eines Halbbandfilters immer Null ist, lässt sich mit Hilfe der inversen DTFT zeigen: Die Transformationsformel in der Aufgabenstellung mit  $\Omega_g = \pi/2$

$$X_{HB,id}(\Omega) = \begin{cases} 1 & \text{für } 0 \leq |\Omega| < \pi/2 \\ 0 & \text{sonst} \end{cases} \quad \bullet \rightsquigarrow x[n] = \frac{1}{2} \frac{\sin(n\pi/2)}{n\pi/2} \quad \text{für } n = 0, 1, 2, \dots$$

ergibt Null für alle geraden Werte von  $n$ . Vom idealen kommt man zu einem realen TP, indem man die unendlich lange Impulsantwort (= die Koeffizienten  $h_{HB,id}$ ) mit einer geeigneten Fensterfunktion  $w[n]$  multipliziert. Die Koeffizienten, die Null sind, bleiben bei dieser Operation Null. Wenn die Fensterfunktion symmetrisch ist, bleibt auch die Symmetrie der Koeffizienten und damit die Halbbandbedingungen erhalten!

- f) Einen **Hochpass-Halbbandfilter** erhält man mit der Transformation  $z \rightarrow -z$ :

$$\begin{aligned} H_{ak,TP}(z) &= 0,1z^3 + 0,2z + 0,6 + 0,2z^{-1} + 0,1z^{-3} \\ \Rightarrow H_{ak,HP}(z) &= -0,1z^3 - 0,2z + 0,6 - 0,2z^{-1} - 0,1z^{-3} \end{aligned}$$

Beim Halbbandfilter ist die Transformation besonders einfach durchzuführen, da es nur Terme mit ungeraden Exponenten gibt (außer dem mittleren) und daher alle Terme (außer dem mittleren : - ) das Vorzeichen wechseln.

- g) **IIR-Halbbandfilter** können nicht linearphasig sein, da sie Pole außerhalb des Ursprungs haben. Es gibt dennoch IIR-Halbbandfilter, allerdings ist hier die Halbbandbedingung anders definiert: Hier muss die *quadrierte* Betragsfunktion symmetrisch zu  $\Omega = \pi/2$  verlaufen.

Nicht alle Verfahren eignen sich zum Entwurf: Da Pass- und Stopband symmetrisch zu einander sein müssen, können z.B. keine Chebychev-Approximationen verwendet werden (Ripple im Pass- oder Stopband). Meist werden elliptische Filter verwendet, der resultierende Pol-Nullstellenplan hat Pole auf der imaginären Achse und Nullstellen entlang des Einheitskreises im Sperrband (bei den dazu symmetrischen Frequenzen im Durchlassband ist der Betragsgang 1).

Auch bei IIR Halbbandfiltern ist jeder zweite Koeffizient Null, so dass auch hier knapp die Hälfte der Multiplikationen eingespart werden kann.

## M5.5 Filterentwurf mit Python → A5.5

Entworfen werden sollte ein FIR-Filter nach der Equiripple-Methode und eins mit Kaiser Window jeweils mit folgenden Spezifikationen:

$$F_{DB} = 0,05, F_{SB} = 0,025, A_{DB} \leq 0,1 \text{ dB}, A_{SB} \geq 50 \text{ dB}$$

Den Spezifikationen entnimmt man, dass ein *Hochpassfilter* gesucht wird. Für Equiripple Filter gibt es einen Algorithmus, der mit guter Genauigkeit die Koeffizienten für ein Filter minimaler Ordnung ermitteln kann (Abb. M5.8a).

Für den gefensterten Entwurf (Abb. M5.8b). gibt es keinen so guten Algorithmus, hier ist mehr Handarbeit gefragt. Dabei kann man an folgenden Schrauben drehen:

**Grenzfrequenz  $F_C$ :** Die -6 dB Grenzfrequenz des Filters kann man im Übergangsbereich des Filters so verschieben, dass eher das Durchlass- oder das Sperrband „angekratzt“ werden.

**Filterordnung  $N$ :** Je höher die Filterordnung, desto steiler ist der Übergangsbereich. Die Höhe des Nebenmaximums bleibt dabei unverändert. Überschreitet die Höhe die geforderte Sperrdämpfung, muss man entweder die Ordnung so hoch wählen, dass das Nebenmaximum im Übergangsbereich auftritt oder man muss den Parameter  $\beta$  erhöhen.

**Parameter  $\beta$ :** Der Parameter  $\beta$  des Kaiser-Fensters beeinflusst den Kompromiss zwischen Main-Lobe Breite und Fläche der side lobes. Je größer man  $\beta$  wählt, desto breiter wird der main lobe (= breiterer Übergang im Durchlassband) und desto geringer wird die Fläche der side lobes (= bessere Dämpfung im Sperrband).

## M5.6 Filtertransformationen → A5.6

Die Systemfunktion  $H_1(z)$  erhält man durch  $z$ -Transformation von  $h_1[n]$ :

$$\begin{aligned} h_1[n] = \{1; 7/4; 1/2; -1/4\} \Rightarrow H_1(z) &= 1 + 7/4z^{-1} + 1/2z^{-2} - 1/4z^{-3} \\ &= z^{-3} (z^3 + 7/4z^2 + 1/2z - 1/4) \end{aligned}$$

### a) Nullstellen von $H_1(z)$

Eine weitere Nullstelle muss vorliegen, da das Filter dritter Ordnung ist; der Verlauf von  $H_1(e^{j2\pi F})$  lässt vermuten dass diese sich bei  $F = 0,5$  bzw.  $f = f_S/2$  befindet.

Polynomdivision durch  $(z + 1)^2 = z^2 + 2z + 1$  ergibt:

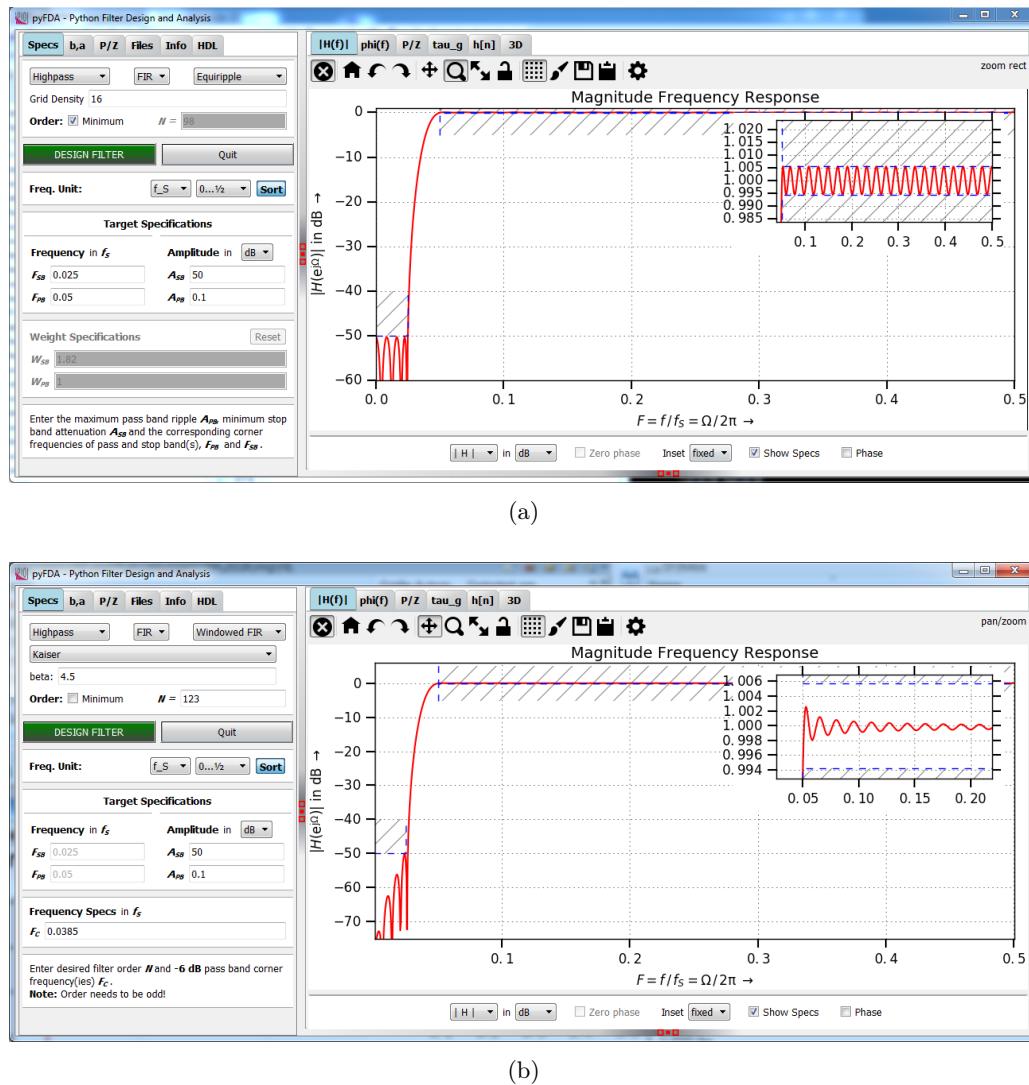


Abb. M5.8: Equiripple (a) und Kaiser Window (b) Hochpassentwurf mit pyFDA zu Aufgabe 5.5

$$\begin{array}{r}
 \left( \begin{array}{c} z^3 + \frac{7}{4}z^2 + \frac{1}{2}z - \frac{1}{4} \\ -z^3 - 2z^2 - z \end{array} \right) \div (z^2 + 2z + 1) = z - \frac{1}{4} \\
 \hline
 \begin{array}{c} -\frac{1}{4}z^2 - \frac{1}{2}z - \frac{1}{4} \\ \frac{1}{4}z^2 + \frac{1}{2}z + \frac{1}{4} \end{array} \\
 \hline
 0
 \end{array}$$

Die Polynomdivision zeigt, dass bei  $z = -1$  eine doppelte Nullstelle vorliegt und bei  $z = +1/4$  eine weitere einfache Nullstelle.

Beweis, dass *eine* (oder mehrere) Nullstellen bei  $f_S/2$  vorliegen durch Einsetzen:

$$H_1(f = f_S/2) = H_1(z = -1) = -1 + 7/4 - 1/2 - 1/4 = 0$$

$$H_1(f = 0) = H_1(z = 1) = 1 + 7/4 + 1/2 - 1/4 = 3$$

$$H_1(f = f_S/2) = H_1(z = -1) = 1 - 7/4 + 1/2 + 1/4 = 0$$

### b) Filter mit jeweils zwei Verzögerungsgliedern

$$h_2[n] = \{1; 0; 7/4; 0; 1/2; 0; -1/4\} \quad \circlearrowleft \bullet \quad H_2(z) = 1 + 7/4z^{-2} + 1/2z^{-4} - 1/4z^{-6}$$

Zusätzliche Verzögerungsglieder stauchen den Frequenzgang zusammen. Die Maxima finden sich jetzt bei  $f = 0$  und  $f = f_S/2$ ; das Minimum liegt jetzt bei  $f_S/4$  (siehe Abb. M5.9) (nicht gefragt).

### c) TP-HP-Transformation

durch  $H_3(z) = H_1(-z)$  ergibt  $H_3(z) = 1 - 7/4 z^{-1} + 1/2 z^{-2} + 1/4 z^{-3}$ . Durch die Transformation wird der Frequenzgang  $|H_3(f)|$  gegenüber  $|H_1(f)|$  um  $f_S/2$  verschoben, das Maximum liegt daher jetzt bei  $f_S/2$ , das Minimum bei  $f = 0$  (siehe Abb. M5.9):

$$H_3(f=0) = H_3(z=1) = 1 - 7/4 + 1/2 + 1/4 = 0$$

$$H_3(f=f_S/2) = H_3(z=-1) = 1 + 7/4 + 1/2 - 1/4 = 3$$

$$h_3[n] = \{1; -7/4; +1/2; +1/4\}$$

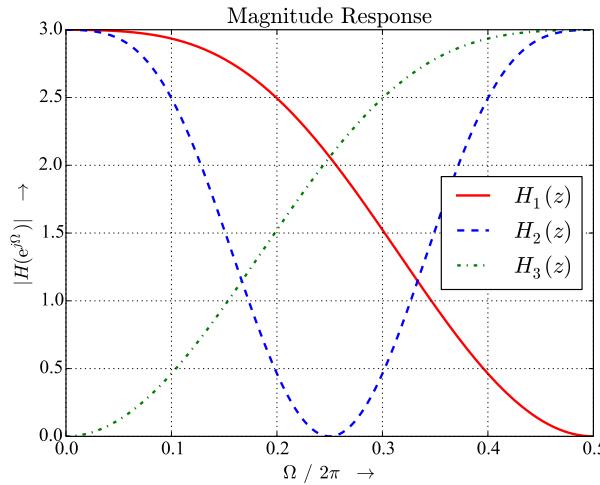


Abb. M5.9: Betragsfrequenzgänge zu Aufgabe 5.6

## M5.7 Filterimplementierung auf FPGAs → A5.7

Ein FIR-Filter 20. Ordnung benötigt im allgemeinen Fall (d.h. es kann keine Symmetrie ausgenutzt werden, um Multiplikationen einzusparen)  $N_{FIR} = 21$  Koeffizientenmultiplizierer, die hier mit MAC-Cores implementiert werden sollen.

### a) Maximaler Durchsatz mit einem MAC-Core:

Wenn der MAC-Core mit maximaler Frequenz getaktet wird, kann er  $f_{MAC} = 2 \cdot 10^8$  Samples je Sekunde mit einem Koeffizientenwert multiplizieren. In einem allgemeinen FIR-Filter 20. Ordnung müssen pro Sample  $N_{FIR} = 21$  Multiplikationen durchgeführt werden. Damit können maximal

$$R_a = f_{MAC}/N_{FIR} = 2 \cdot 10^8 \text{S/s}/21 = 9,52 \text{ MS/s}$$

vom Filter verarbeitet werden.

### b) Maximaler Durchsatz bei linearphasigen FIR-Filter:

Beim linearphasigen Filter sind die Koeffizientenwerte symmetrisch (gerade oder ungerade) zum mittleren Koeffizienten, die zugehörigen Samples können daher vor der Multiplikation paarweise durch Addition (gerade Symmetrie) oder Subtraktion (ungerade Symmetrie) zusammengefasst werden. In diesem Fall bleiben  $N_{FIR,lin} = 11$  Multiplikationen übrig, damit wird ein Durchsatz erzielt von

$$R_b = f_{MAC}/N_{FIR,lin} = 2 \cdot 10^8 \text{S/s}/11 = 18,18 \text{ MS/s}.$$

Bei einem Halbbandfilter ist zusätzlich jeder zweite Koeffizient außer dem mittleren Null, damit können 5 weitere Multiplikationen eingespart werden (s. Aufgabe 6.4) und es werden nur noch  $N_{FIR,HB} = 6$  Multiplikationen je Sample benötigt:

$$R_{b,HB} = f_{MAC}/N_{FIR,HB} = 2 \cdot 10^8 \text{S/s}/6 = 33,33 \text{ MS/s.}$$

c) **Maximaler Durchsatz mit zwei MAC-Cores:**

Die Multiplikationen je Eingangssample können hier zwischen beiden Multiplizierern aufgeteilt werden, der eine übernimmt 5 und der andere 6 Multiplikationen je Eingangssample und begrenzt damit den Durchsatz auf

$$R_c = f_{MAC}/\lceil N_{FIR,lin}/2 \rceil = 2 \cdot 10^8 \text{S/s}/6 = 33,33 \text{ MS/s.}$$

## M5.8 Verständnisfragen zu Filtern → A5.8

Aussage	FIR-Filter		IIR-Filter	
	ja	nein	ja	nein
Rekursiv:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Endliche Impulsantwort:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Instabil:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Linearphasig: (1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Minimalphasig: (1)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Minimalphasig und linearphasig: (1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Pole außerhalb des Ursprungs:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pole außerhalb des Einheitskreises:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Nullstellen außerhalb des Ursprungs:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Nullstellen außerhalb des Einheitskreises:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Halbbandfilter: (2)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Konstante Gruppenlaufzeit:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Transponieren möglich: (3)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Entwurf über bilineare Transformation:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Transversale Struktur:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Abgeleitet von analogen Butterworth-Filtern:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Tab. M5.5: Musterlösung zu Aufgabe 5.8

### Erklärungen zu Tab. M5.5

- (1) Linearphasige Filter setzen voraus, dass Nullstellenpaare (oder auch Polstellenpaare) symmetrisch zum Einheitskreis liegen, d.h. Radien von  $r_1$  und  $1/r_1$  aufweisen. Bei Polstellenpaaren würde immer eine Polstelle außerhalb des EK liegen, das resultierende Filter wäre instabil. Daher gibt es keine linearphasigen IIR-Filter.
- (2) Halbbandfilter können sowohl mit FIR- als auch mit IIR-Strukturen realisiert werden, allerdings sind letztere nicht linearphasig (siehe auch Aufgabe 5.4). Bei minimalphasigen Filtern müssen alle Null- und Polstellen innerhalb des EK liegen, daher gibt es keine Filter, die gleichzeitig linearphasig und minimalphasig sind.
- (3) Beliebige (lineare) Strukturen können transponiert werden.

# M6 FIX: Wortlängeneffekte und Fixpoint-Systeme im Zeitbereich

## M6.1 Zahlendarstellung im Zweierkomplementformat → A6.1

- a) Dezimaler Wert von **10101** in verschiedenen Zahlenformaten

	1	0	1	0	1	
Unsigned Integer	$2^4$	$2^3$	$2^2$	$2^1$	$2^0$	<b>21</b>
Signed Integer	$-2^4$	$2^3$	$2^2$	$2^1$	$2^0$	<b>-11</b>
<b>Q(2.2)</b>	$-2^2$	$2^1$	$2^0$	$2^{-1}$	$2^{-2}$	<b>-2,75</b>

Tab. M6.1: **10101** in verschiedenen Zahlenformaten (Aufgabe [6.1a](#))

- b) Darstellung verschiedener Zahlen im **Q(2.6)** Format

	<b>S</b>	<b><math>2^1</math></b>	<b><math>2^0</math></b>	.	<b><math>2^{-1}</math></b>	<b><math>2^{-2}</math></b>	<b><math>2^{-3}</math></b>	<b><math>2^{-4}</math></b>	<b><math>2^{-5}</math></b>	<b><math>2^{-6}</math></b>
<b>+2</b>	0	1	0	.	0	0	0	0	0	0
<b>+2,5</b>	0	1	0	.	1	0	0	0	0	0
<b>-3/4</b>	1	1	1	.	0	1	0	0	0	0
<b>+1/3</b>	0	0	0	.	0	1	0	1	0	1

Tab. M6.2: Verschiedene Zahlen im Q2.6 Format (Aufgabe [6.1b](#))

c) Requantisierung von  $7/16$ ,  $-9/16$  und  $-11/16$  mit verschiedenen Methoden

Die Ergebnisse finden Sie in Tab. M6.3, die letzte Spalte zeigt an, ob der Betrag nach der Requantisierung kleiner ist als vorher,  $|x_q| \leq |x|$ .

ZK	$S [-2^4]$	$2^3$	$2^2$	$2^1$	$2^0$		
NZKF	$S [-2^0]$	$2^{-1}$	$2^{-2}$	$2^{-3}$	$2^{-4}$		
<b>Ursprünglicher Wert</b>	0	0	1	1	1	<b>7/16</b>	
<b>Abschneiden</b>	0	0	1	-	-	<b>1/4</b>	✓
<b>Runden</b>	0	0	1	1	-		
			+	1	-		
	0	1	0	-	-	<b>1/2</b>	!
<b>Wertschneiden</b>	0	0	1	-	-		
		+	$S = 0$	-	-		
	0	0	1	-	-	<b>1/4</b>	✓
<b>Ursprünglicher Wert</b>	1	0	1	1	1	<b>-9/16</b>	
<b>Abschneiden</b>	1	0	1	-	-	<b>-3/4</b>	!
<b>Runden</b>	1	0	1	1	-		
			+	1	-		
	1	1	0	-	-	<b>-1/2</b>	✓
<b>Wertschneiden</b>	1	0	1	-	-		
		+	$S = 1$	-	-		
	1	1	0	-	-	<b>-1/2</b>	✓
<b>Ursprünglicher Wert</b>	1	0	1	0	1	<b>-11/16</b>	
<b>Abschneiden</b>	1	0	1	-	-	<b>-3/4</b>	!
<b>Runden</b>	1	0	1	0	-		
			+	1	-		
	1	0	1	-	-	<b>-3/4</b>	!
<b>Wertschneiden</b>	1	0	1	-	-		
		+	$S = 1$	-	-		
	1	1	0	-	-	<b>-1/2</b>	✓

Tab. M6.3: Zweierkomplement und Requantisierung (Aufgabe 6.1c)

## M6.2 Analog-Digital-Wandlung eines Drucksensors → A6.2

- a) Die **Auflösung des Wandlers** beträgt  $q_U = FSR/2^8 = 5 \text{ V}/256 = 19,53 \text{ mV}$ ; das ist auch die Unsicherheit  $q_U$ , die durch die Quantisierung hervorgerufen wird: Für den analogen Wert  $x$  gilt:  $x \in [x_q - q_U/2 \dots x_q + q_U/2]$ .
- b) Die **Auflösung des Wandlers bezogen auf die Messgröße** ergibt sich aus

$$q_a = q_U / S_{Sens} = 19,5 \text{ mV}/400 \text{ mV/g} = 5/(256 \cdot 0,4) \text{ g} = 48,83 \cdot 10^{-3} \text{ g}.$$

### c) Digitale Nullpunkt Korrektur

Bei  $a = 0$  liefert der Sensor 1,5 V, die vom ADC in den Code  $\tilde{c}_0 = \text{round}(U_0/q_U) = 77$  gewandelt werden. Durch Subtraktion dieses Wertes wird der Beschleunigung  $a = 0$  der Digitalwert  $c_0 = 0$  zugeordnet und so der analoge Offset von 1,5 V im Digitalen kompensiert.

- d) Den **Digitalwert bei  $a = \pm 1,2 \text{ g}$**  erhält man jetzt - da der Offset kompensiert ist - einfach durch

$$c(a = \pm 1,2 \text{ g}) = \lfloor a S_{Sens}/q_U \rfloor = \lfloor \pm 1,2 \text{ g} \cdot 400 \text{ mV/g} / 19,53 \text{ mV} \rfloor = \pm 24$$

Binär geschrieben erhält man  $+24 \hat{=} 0001\ 1000$  und  $-24 \hat{=} 1110\ 1000$  (ZK: Bits invertieren und 1 LSB addieren). Für das nZKF-Format stellt man sich den Fraktionalpunkt nach dem MSB vor bei **0,001 1000** bzw. **1,110 1000**, entsprechend einem Links-Shift um  $WF = 7$  Stellen. Ein LSB entspricht dann  $2^{-WF} = 2^{-7} = 1/128$ , im Dezimalsystem repräsentiert die Darstellung die Werte  $\pm 24/128 = \pm 0,1875$ .

Man kann jetzt einen Skalierungsfaktor zwischen der Sensorgröße  $a$  und der digitalen Repräsentation in nZKF Format bestimmen, um einfacher zwischen beiden Welten umrechnen zu können:

$$\frac{x_{nZKF}}{a} = \frac{2^{-WF} a/q_a}{a} = \frac{1}{2^{WF} q_a} = 0,16 \text{ g}^{-1}$$

- e) Die **maximale Eingangsamplitude** des Wandlers muss so gewählt werden, dass weder die minimale noch die maximal darstellbaren Zahlen überschritten werden. Aussteuerungsgrenze des ADCs überschritten wird. Sie entspricht daher ungefähr der Hälfte des  $FSR$ , also  $A_{max} = 2,5 \text{ V}$ .
- f) Das binäre Wort **1000 0001** entspricht  $128 + 1 = 129$  in vorzeichenloser („unsigned“) Interpretation bzw.  $-128 + 1 = -127$  (**-0111 1111**) in vorzeichenbehafteter („signed“) Interpretation. In nZKF Interpretation repräsentiert das Binärwort den Wert  $-127 / 128 = -0,9921875$ .
- g) Die **Empfindlichkeit und Auflösung des Sensors mit Vorverstärker** werden durch analoge Verstärkung und Offsetkorrektur maximiert, wenn bei maximaler bzw. minimaler Beschleunigung  $a_{max} = \pm 2000 \text{ mg}$  gerade die maximale bzw. minimale Eingangsspannung des ADCs  $U_{ADC,max} = 0$  bzw.  $5 \text{ V}$  erreicht wird. Hierfür ist eine Vorverstärkung von

$$A_{pre} = \frac{FSR}{\Delta U_{Sens,max}} = \frac{5 \text{ V}}{\Delta a_{max} S_{Sens}} = \frac{5 \text{ V}}{4g \cdot 0,4 \text{ V/g}} = 3,125 \quad \text{erforderlich.}$$

Damit erhöht sich die Empfindlichkeit des Sensors mit Vorverstärker auf

$$S_{Sens,Pre} = \frac{FSR}{\Delta a_{max}} = \frac{5 \text{ V}}{4g} = A_{pre} S_{Sens} = 3,125 \cdot 0,4 \text{ V/g} = 1,25 \text{ V/g}$$

und die Auflösung bezogen auf die Messgröße auf

$$q_{a,Pre} = q_U / S_{Sens,pre} = 19,53 \text{ mV}/1250 \text{ mV/g} = 15,62 \cdot 10^{-3} \text{ g}.$$

- h) Die **Anzahl der genutzten Codeschritte** des ADCs erhält man, indem man den Bereich der angelegten Spannung durch die Auflösung  $q_U$  des ADCs dividiert:

$$\begin{aligned} N_{dir} &= \Delta U_{Sens,max} / q_U = \Delta U_{Sens,max} 2^W / FSR = 1,6 \text{ V} \cdot 256 / 5 \text{ V} = \mathbf{81,9} \\ \Rightarrow ENOB_{dir} &= \log_2(N_{dir}) = \mathbf{6,36 \text{ Bits}} \\ N_{pre} &= \Delta U_{Sens,pre,max} / q_U = FSR / q_U = 2^8 = \mathbf{256} \\ \Rightarrow ENOB_{pre} &= \log_2(N_{pre}) = \mathbf{8 \text{ Bits}} \end{aligned}$$

Man gewinnt also mehr als 1,5 Bit effektive Auflösung, indem man die zu wandelnde Spannung richtig an den Eingangsbereich des Wandlers anpasst. Allerdings ist es heutzutage oft billiger, einen höher auflösenden ADC zu verwenden und den  $FSR$  nicht voll auszunutzen als einen einfacheren ADC mit zusätzlichem Vorverstärker einzusetzen.

- i) Beim **MA-Filter 15. Ordnung** werden insgesamt  $N = 16$  Werte aufaddiert, die Wortlänge erhöht sich daher um  $\lceil \log_2(16) \rceil = 4$  Bits (bit growth). Der Fall eines DC-Eingangssignals mit  $x[n] = 1$  kann nicht vorkommen, da der Wert 1 außerhalb des Wertebereichs liegt, daher erreicht das Ausgangssignal auch nicht ganz den Wert 16. Bei einem MA-Filter mit der Ordnung 16 (also mit 17 Taps) müsste man jedoch die Wortlänge bereits um 5 Bits erhöhen. Die zusätzlichen 4 Bits werden als Integerbits benötigt, um den vergrößerten Wertebereich aufzunehmen, es wird also das Format Q4.7 benötigt.

Ein MA-Filter der Ordnung  $N - 1$  hat  $N$  gleichverteilte Nullstellen auf dem EK, wobei die Nullstelle bei  $z = 1$  durch eine Polstelle kompensiert wird. Die erste Nullstelle liegt daher bei  $F_{0,1} = 1/N = 1/16$ . Der Betragsgang (nicht gefragt) wird beschrieben durch (siehe Aufgabe M2.1):

$$|H(f)| = \left| \frac{\sin(N\Omega/2)}{\sin(\Omega/2)} \right| = |N \operatorname{di}_N(\Omega)| \text{ mit } H(0) = N$$

### M6.3 Moving Average Filter mit endlicher Wortbreite → A6.3

- a) **Systemfunktion und Amplitudenfrequenzgang**

Die Systemfunktion lässt sich mit Hilfe der endlichen geometrischen Reihe (B.42) umformen zu

$$H(z) = \sum_{i=0}^8 z^{-i} = \frac{1 - z^{-9}}{1 - z^{-1}} = \frac{z^9 - 1}{(z - 1) z^8} \Rightarrow H(e^{j\Omega}) = \frac{\sin 9\Omega/2}{\sin \Omega/2} e^{-j4\Omega}$$

Das Filter ist 8. Ordnung und hat daher 8 Nullstellen, die beim MA-Filter alle auf dem EK liegen. Die Verteilung der Nullstellen lässt sich am einfachsten bestimmen, wenn man  $H(z)$  mit Hilfe der endlichen geometrischen Reihe umformt (siehe oben). Dann sieht man aus  $z^9 - 1 = 0$  sofort, dass 9 Nullstellen auf dem EK verteilt liegen bei  $\phi = 2k\pi/9; k = \dots, -2, -1, 0, 1, 2, \dots$ . Die Nullstelle bei  $z = 1$  wird durch den einzigen Pol kompensiert, so dass man wieder die erwarteten 8 Nullstellen erhält. Die erste Nullstelle liegt daher bei  $\Omega = 2\pi/9$  oder  $f = f_S/9$ .

Das gleiche Ergebnis erhält man natürlich aus  $H(e^{j\Omega})$ , indem man  $9\Omega/2 = \pi$  setzt. Aus  $H(e^{j\Omega})$  kann man außerdem die lineare Phase von  $\phi = -4\Omega$  ablesen, entsprechend einer Gruppenlaufzeit von  $\tau_g = 4T_S$  (bis zur Mitte des Filters gerechnet).

- b) **Wortlängen**

Das Wachstum  $\Delta N$  der benötigten Wortlänge („bit growth“) bei  $k$  Summanden ergibt sich aus  $\Delta N = \lceil \log_2 k \rceil$  (siehe Abb. M6.1). Die Wortlängen am Ausgang betragen in beiden Fällen 14 bits, siehe Tab. M6.4. In den Abb. M6.1 sind optionale Quantisierer am Ausgang eingezeichnet, die benötigt werden falls mit 10 bit weitergearbeitet werden soll.

$k$	1	2	3	4	5	6	7	8	9
$\log_2 k$	0	1	1,6	2	2,3	2,6	2,8	3	3,2
$\Delta N$	0	1	2	2	3	3	3	3	4

Tab. M6.4: Wortlängenwachstum zu Aufgabe M6.3

### c) Kritischer Pfad

Bei der Implementierung in Direktform liegen 8 Addierer im kritischen Pfad,  $T_{krit,DF} = 8\tau_{add} = 16$  ns und  $f_{max,DF} = 1/T_{krit,DF} = 62,5$  MHz. Bei der transponierten Direktform liegt nur ein Addierer im kritischen Pfad, theoretisch könnte das MA-Filter in dieser Implementierung also achtmal so schnell getaktet werden,  $f_{max,TDF} = 1/T_{krit} = 500$  MHz!

### d) Hardware: FlipFlops und Full-Adder

Direktform:  $8 \times 10 = 80$  FlipFlops,  $10 + 11 + 2 \times 12 + 4 \times 13 = 97$  FA.

Transponierte Direktform:  $10 + 11 + 2 \times 12 + 4 \times 13 = 97$  FlipFlops und FA.

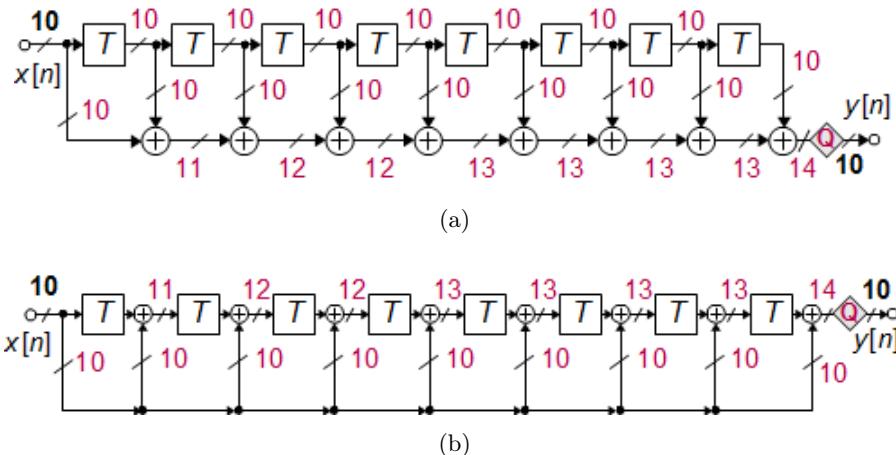


Abb. M6.1: Fixpoint Moving Average Filter mit Wortbreiten zu Aufgabe 6.3, (a) Direktform und (b) transponierte Direktform

## M6.4 FIR Filter mit Quantisierung → A6.4

### a) Allgemeine Filtereigenschaften

Das Filter hat 11 Koeffizienten und daher 10 Verzögerungsglieder  $\rightarrow$  zehnter Ordnung. Es ist linearphasig (Koeffizienten sind symmetrisch zur Mitte des Filters) und ein Halbbandfilter (jeder zweite Koeffizient außer dem mittleren ist Null). Da Koeffizientenquantisierung die Symmetrie nicht stört und die Nullen Null bleiben, ist auch das quantisierte Filter ein Halbbandfilter. Es wurde berechnet mit dem Kommando

```
b = fir1(10, [0 0,3 0,7 1], [1 1 0 0])
```

entsprechend der Eckfrequenzen des Durchlassbandes  $F_{DB} = 0,15$  und des Sperrbandes  $F_{SB} = 0,35$ , die symmetrisch zu  $F = 0,25$  gewählt wurden.

### b) Hardwareressourcen

In der optimierten Version (Abb. M6.2) werden 6 Addierer und nur drei echte Multiplizierer benötigt; Multiplizierer für identische Koeffizienten können zusammengefasst werden, indem vor

der Multiplikation die Datenwörter addiert werden:

$$y[n] = (x[n] + x[n - 10])b_0 + (x[n - 2] + x[n - 8])b_2 + (x[n - 4] + x[n - 6])b_4 + b_5x[n - 5]$$

Der Koeffizient  $b_5$  beträgt genau  $1/2$  und kann daher effizient durch Schieben der Bits um eine Position nach rechts realisiert werden.

Im Vergleich dazu werden in der nicht optimierten Version 6 Addierer und 6 Multiplizierer benötigt (auch hier wird der Koeffizient  $b_5$  durch Bitschieben realisiert).

Bei einem allgemeinen FIR-Filter 10. Ordnung werden 10 Addierer und 11 Multiplizierer benötigt.

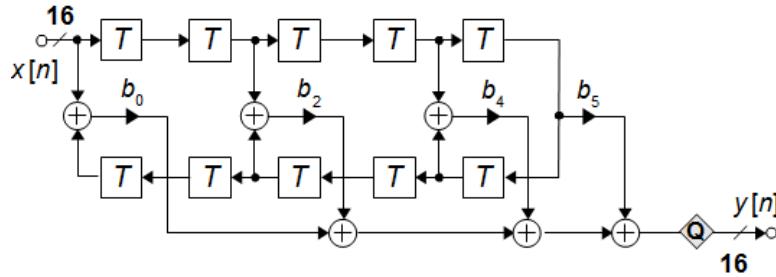


Abb. M6.2: FIR-Filter mit optimierter Struktur und Quantisierung zu [M6.4](#)

### c) Kritischer Pfad

Bei einem allgemeinen FIR-Filter 10. Ordnung (Abb. [6.4](#)) ist die Verzögerungszeit des kritischen Pfads  $T_{krit} = \tau_{mul} + 10\tau_{add} = 30$  ns, die maximale Taktrate ist daher  $f_{max} = 1/T_{krit} = 33,3$  MHz.

Wenn man die Bedingung ausnutzt, dass jeder zweite Koeffizient Null ist (Halbbandfilter), ist der kritische Pfad  $T_{krit} = \tau_{mul} + 6\tau_{add} = 22$  ns, die maximale Taktrate ist daher  $f_{max} = 1/T_{krit} = 45,4$  MHz.

Beim optimierten Filter (Abb. [M6.2](#)) ist die Verzögerungszeit des kritischen Pfads  $T_{krit,opt} = \tau_{mul} + 4\tau_{add} = 18$  ns, die maximale Taktrate ist daher  $f_{max,opt} = 1/T_{krit,opt} = 55,6$  MHz.

### d) Amplitudenfrequenzgang

$$H(f = 0) = H(z = 1) = \sum_i b_i = 1,003$$

$$H(f = f_S/2) = H(z = -1) = \sum_i (-1)^{-i} b_i = 0,003$$

$$H(f = f_S/4) = H(z = j) = \sum_i (j)^{-i} b_i = -0,5j$$

### e) Optimales Koeffizientenformat

Der betragsmäßig größte Koeffizient,  $b_5 = 0,5$ , muss im gewählten Zahlenformat gerade noch darstellbar sein. Dafür wird das Zahlenformat  $B(0,17)$  benötigt, ein LSB ist  $q = 2^{-17} = 7,6 \cdot 10^{-6}$ . Allgemein lässt sich die Anzahl der benötigten Vorkommastellen bestimmen über  $WI = \lceil \log_2(\max(b_i)) \rceil$

$b_i$	ideal	$B(0,17)$	$\epsilon(0,17)$	$B(0,7)$	$\epsilon(0,7)$
$b_0 = b_{10}$	0,01623	0,0162277	$2,28 \cdot 10^{-6}$	0,015625	$0,61 \cdot 10^{-3}$
$b_2 = b_8$	-0,06871	-0,0687103	$0,33 \cdot 10^{-6}$	-0,070312	$1,60 \cdot 10^{-3}$
$b_4 = b_6$	0,30399	0,303986	$4,40 \cdot 10^{-6}$	0,296875	$7,11 \cdot 10^{-3}$
$b_5$	0,5	0,5	0	0,5	0
<b>LSB</b>	—	—	$7,6 \cdot 10^{-6}$	—	$7,8 \cdot 10^{-3}$

Tab. M6.5: Filterkoeffizienten in verschiedenen Zahlenformaten zu [M6.4](#)

- f) **Quantisierte Koeffizienten:** Die quantisierten Koeffizienten in Tab. [M6.5](#) und deren Quantisierungsfehler  $\epsilon_i = b_{i,id} - b_{i,Q}$  wurden mit dem Computer berechnet. Der Fehler ist bei positiven Koeffizienten immer größer Null und bei negativen Koeffizienten kleiner Null, da die Koeffizienten durch Wertschneiden bestimmt wurden und er ist natürlich immer kleiner als ein LSB ( $2^{-17}$  bzw.  $2^{-7}$ ). Die Koeffizienten in Tab. [M6.5](#) sind auf 4 Nachkommastellen gerundet dargestellt, die  $\epsilon_i$ -Werte wurden aber natürlich aus den quantisierten Koeffizienten mit allen Nachkommastellen ermittelt.

Siehe hierzu das Notebook **FIX\_FIR\_quant\_ML**.

- g) Die **Koeffizientenfläche** ist die Summe des Betrags aller Koeffizienten:  $A_b = \sum |b_i| = 1,2779$ . Dieser Wert wird benötigt, um genauer abzuschätzen welche Wortbreite der Akkumulator für die Teilprodukte haben muss. Kleinere Koeffizienten führen zu kleineren Teilprodukten und damit zu geringerem „bit growth“ des Akkumulators.
- h) **Skalierung der Daten vor Multiplikation:** Es gibt zwei Möglichkeiten die Daten vor der Multiplikation zu skalieren: Das Datenwort kann um zwei Bits nach links geschoben werden, die beiden LSBs werden mit Nullen aufgefüllt. Die zwei LSBs des Produkts sind dann ebenfalls Null und können ohne Genauigkeitsverlust abgeschnitten werden. Alternativ können zwei MSBs ergänzt werden, hier muss allerdings das Vorzeichen beachtet werden und mit '1' oder '0' aufgefüllt werden. Jetzt sind die beiden MSBs des Produkts irrelevant und können weggelassen werden.

In beiden Fällen hat das Produkt  $18 + 16 = 34$  relevante Bits.

- i) Im **Akkumulator** müssen bei einem Filter 10. Ordnung im allgemeinen Fall 11 Teilprodukte aufsummiert werden. Das bedeutet einen Bit Growth von  $\lceil \log_2(11) \rceil = 4$  Bits, der Akkumulator muss also  $34 + 4 = 38$  Bits breit sein. Berücksichtigt man, dass nur 7 Koeffizienten ungleich Null sind, werden nur 3 zusätzliche Bits benötigt. Berücksichtigt man außerdem die Koeffizientenfläche  $A_b$ , also den Betrag aller Koeffizienten, benötigt man nur  $\lceil \log_2(A_b) \rceil = 1$  zusätzliches Bit.

#### j) Akkumulatorskalierung

Wenn der Akkumulator dimensioniert wurde wie im vorigen Aufgabenpunkt beschrieben, wird sein Wertebereich voll ausgenutzt. Es werden dann einfach die obersten 16 Bits benutzt (ggf. nach Rundung) und die unteren Bits abgeschnitten.

## M6.5 + FIR Filter mit Skalierung → A6.5

### a) Frequenzgang bei $f = 0$

$$H_1(f = 0) = H_1(z = 1) = 1 + 7/4 + 1/2 - 1/4 = 3$$

b) **Signalflussgraph und Überlauf**

Bei diesem einfachen FIR-Filter kann nur in den Summierern (bzw. im Akkumulator wenn die Summierer zusammengefasst werden) ein Überlauf auftreten.

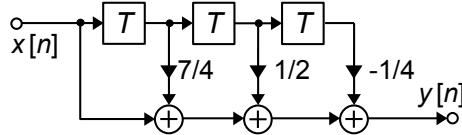


Abb. M6.3: SFG in Direktform zu Aufgabe 6.5

c) **Skalierungskoeffizient  $k_{E\infty}$  für  $L^\infty$ -Skalierung**

Es tritt an keinem Knoten ein Überlauf bei sinusförmigem Eingangssignal in eingeschwungenem Zustand auf. Aus Frequenzgang ist erkennbar, dass das Maximum bei  $H_1(f = 0) = L^\infty = H_1(z = 1) = 3$  auftritt, also bei der Eingangsfolge  $x[n] = \{1; 1; 1; 1\}$ .

Der Skalierungsfaktor lautet dementsprechend:  $k_{E\infty} = 1/L^\infty = 1/3$

d) **Skalierungskoeffizient  $k_{E1}$  für  $L^1$ -Skalierung**

Es tritt kein Überlauf bei beliebigem Eingangssignal auf; worst case ist eine Folge von Einheitsimpulsen, die Vorzeichen der Einheitspulse hängen von den Vorzeichen der Koeffizienten ab.  $L^1 = \sum |h_1[n]| = 3,5$ . Dieser Wert tritt auf bei  $x[n] = \{-1; 1; 1; 1\}$ . Der Skalierungsfaktor lautet:  $k_{E1} = 1/L^1 = 1/3,5$

## M6.6 + Integrator mit endlicher Wortbreite → A6.6

Der Wertebereich eines vorzeichenbehafteten 8 bit Worts beträgt -128 ... 127. Für ein positives Eingangssignal ist der maximale Ausgangswert des Integrators daher +127.

a) **Integrator erster Ordnung - Impulsantwort:**

$$y_{a,\delta}[n+1] = y_a[n] + \delta[n] = u[n] \Rightarrow y_{a,\delta}[n] = u[n-1] = \{0; 1; 1; 1; \dots\}$$

$z$ -Transformation:

$$Y_{a,\delta}(z)z = Y_{a,\delta}(z) + 1 \Rightarrow Y_{a,\delta}(z) = \frac{1}{z-1} = \frac{z^{-1}}{1-z^{-1}} \bullet \circlearrowleft y_{a,\delta}[n] = u[n-1]$$

Die  $z$ -Transformierte der Impulsantwort ist die Übertragungsfunktion,  $H_a(z) = Y_{a,\delta}(z) = 1/(z-1)$  mit einer Polstelle bei  $z = 1$  und einer Nullstelle im Unendlichen. Für Gleichspannungseingangssignale ist die Übertragungsfunktion daher unendlich,  $H(f=0) = H(z=1) \rightarrow \infty$ .  $H(f)$  fällt monoton ab zu  $H(f=f_s/2) = H(z=-1) = -1/2$ .

Berechnung der Frequenz, bei der der Betragsgang 1 wird:

$$\begin{aligned} |H(e^{j\Omega})| &= \left| \frac{1}{e^{j\Omega} - 1} \right| = \left| \frac{1}{\cos \Omega + j \sin \Omega - 1} \right| = \frac{1}{\sqrt{\cos^2 \Omega - 2 \cos \Omega + 1 + \sin^2 \Omega}} = 1 \\ &\Rightarrow 2 - 2 \cos \Omega = 1 \Rightarrow \cos \Omega = 1/2 \Rightarrow \Omega = \arccos 1/2 = \pi/3 \text{ (entspricht } f_s/6) \end{aligned}$$

**Sprungantwort (nicht gefragt; Lösung durch „genaues Hinschauen“):**

$$y_{a,u}[n+1] = y_{a,u}[n] + u[n] = (n+1)u[n] = \{1; 2; 3; \dots\} \Rightarrow y_{a,u}[n] = nu[n-1] = \{0; 1; 2; \dots\}$$

oder über  $z$ -Transformation:

$$zY_{a,u}(z) = Y_{a,u}(z) + \frac{z}{z-1} \Rightarrow Y_{a,u}(z) = \frac{z}{(z-1)^2} = \frac{z^{-1}}{(1-z^{-1})^2} \quad \bullet\rightsquigarrow \circ \quad y_{a,u}[n] = nu[n-1]$$

Der Integrator läuft beim Schritt  $n = 128$  über, das Ausgangswort wird dann vorzeichenbehaftet als -128 interpretiert.

**b) Verlustbehafteter Integrator - Impulsantwort:**

$$\begin{aligned} y_{b,\delta}[n+1] &= ay_{b,\delta}[n] + \delta[n] \\ \circ\rightsquigarrow \bullet \quad zY_{b,\delta}(z) &= aY_{b,\delta}(z) + 1 \Rightarrow Y_{b,\delta}(z) = \frac{1}{z-a} \\ \bullet\rightsquigarrow \circ \quad y_{b,\delta}[n] &= a^{n-1}u[n-1] = \{0; 1; a; a^2; \dots\} \end{aligned}$$

Für  $a < 1$  konvergiert die Folge gegen 0.

Die Übertragungsfunktion ist  $H_b(z) = Y_{b,\delta}(z) = 1/(z-a)$ , die Polstelle liegt bei  $z = a$ . Das Maximum von  $H(f)$  wird auch hier bei  $f = 0$  erreicht, hat aber einen endlichen Wert („verlustbehafteter Integrator“), das System ist stabil.

Ähnlich wie in Aufgabenteil a) lässt sich bestimmen, dass  $|H(e^{j\Omega})| = 1$  gilt für  $\Omega = \arccos a/2$ .

**c) Integrator zweiter Ordnung - Impulsantwort:**

$$y_{c,\delta}[n+1] = y_{c,\delta}[n] + y_{a,\delta}[n] = y_{c,\delta}[n] + u[n-1] = y_{a,u}[n-1] = (n-1)u[n-2]$$

oder über  $z$ -Transformation:

$$\begin{aligned} \circ\rightsquigarrow \bullet \quad zY_{c,\delta}(z) &= Y_{c,\delta}(z) + \frac{1}{z-1} \Rightarrow Y_{c,\delta}(z) = \frac{1}{(z-1)^2} = \frac{z^{-2}}{(1-z^{-1})^2} \\ \bullet\rightsquigarrow \circ \quad y_{a,u}[n] &= (n-1)u[n-2] = \{0; 0; 1; 2; 3; \dots\} \end{aligned}$$

Der Integrator läuft beim Schritt  $n = 129$  über, das Ausgangswort wird dann vorzeichenbehaftet als -128 interpretiert.

Die Übertragungsfunktion ist  $H_c(z) = Y_{c,\delta}(z) = 1/(z-1)^2$ , es gibt eine doppelte Nullstelle bei  $z = 1$ ,  $H(f=0) = H(z=1) \rightarrow \infty$ .  $H(f)$  fällt monoton ab zu  $H(f=f_S/2) = H(z=-1) = 1/4$ .

Genau wie in Aufgabenteil a) ist  $|H(e^{j\Omega})| = 1$  bei  $\Omega = \pi/3$ .

## M6.7 Verständnisfragen zu Filtern mit Quantisierung → A6.7

Aussage	FIR-Filter		IIR-Filter	
	ja	nein	ja	nein
Unendliche Impulsantwort: (1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Stabil: (2)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Linearphasig: (3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Minimalphasig: (4)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Pole außerhalb des Ursprungs: (5)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Pole außerhalb des Einheitskreises: (2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Nullstellen außerhalb des Ursprungs: (5)	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Nullstellen außerhalb des Einheitskreises: (2)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Halbbandfilter: (3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Konstante Gruppenlaufzeit: (3)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Transponieren möglich:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Entwurf über bilineare Transformation:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Transversale Struktur:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Abgeleitet von analogen Butterworth-Filters:	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Kaskadierte Form ist unempfindlicher gegen Quantisierungseffekte als direkte Form:	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Grenzzyklen: (1)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Tab. M6.6: Musterlösung zu Aufgabe 6.7

### Erklärungen zu Tab. M6.6

- a) Bei IIR Filtern führt die endliche Rechengenauigkeit der quantisierten Arithmetik dazu, dass entweder Grenzzyklen auftreten (→ unendliche Impulsantwort) oder dass die Ergebnisse aller Multiplikationen zu Null gerundet / abgeschnitten werden (→ endliche Impulsantwort). Bei FIR-Filters können keine Grenzzyklen auftreten, da sie keine Rückkopplung haben. Hier kann es höchstens vorkommen, dass Koeffizienten beim Quantisieren Null werden und dadurch die Impulsantwort kürzer wird.
- b) Durch Koeffizientenquantisierung verschiebt sich die Lage von Pol- und Nullstellen, damit werden auch  $H(f)$  und  $H(z)$  verfälscht. Da nur IIR-Filter Pole außerhalb des Ursprungs haben, können auch nur hier Polstellen aus dem EK herausgeschoben werden und dadurch das System destabilisieren. Nullstellen werden oft auch außerhalb des EK platziert (→ „Linearphasige Filter“); falls sie durch die Quantisierung aus dem EK herauswandern, beeinträchtigt das nicht die Stabilität.
- c) Linearphasige Filter und Halbbandfilter setzen Symmetrien der Koeffizienten voraus. Da die

Symmetrie bei der Quantisierung erhalten bleibt, werden auch Linearphasigkeit und/oder die Eigenschaft des Halbbandfilters nicht gestört. Das gleiche gilt für die konstante Gruppenlaufzeit, die alle linearphasigen Filter haben.

- d) Bei minimalphasigen Filtern müssen alle Null- und Polstellen innerhalb des EK liegen. Die Verschiebung der Pol- und Nullstellen durch Quantisierung kann diese Eigenschaft u.U. ändern.
- e) Pole und Nullstellen im Ursprung entsprechen zusätzlichen (positiven oder negativen) Verzögerungen, die durch die Quantisierung nicht beeinflusst werden.



# M7 NOI: Wortlängeneffekte im Frequenzbereich: Quantisierungsrauschen

## M7.1 Quantisierungsrauschen Analog-Digital-Wandlung → A7.1

- a) **Maximales  $SQNR$ :** Für einen Nyquist-Wandler ( $f_N = f_S/2$ ) mit  $W$  bit Auflösung ist das maximal erreichbare  $SQNR$  für sinusförmige Signale mit Vollaussteuerung

$$SQNR_{max} = (6,02W + 1,76) \text{ dB} = 61,96 \text{ dB für } W = 10.$$

Umgekehrt kann aus einem gegebenen  $S(Q)NR$  die effektive Anzahl von Bits berechnet werden:

$$ENOB = \frac{SNR - 1,76 \text{ dB}}{6,02 \text{ dB/bit}} = 8,51 \text{ bit für } SNR = 53 \text{ dB.}$$

- b) **Verringertes  $SNR$ :** Nichtidealitäten oder Übersteuerung des ADC können das  $SNR$  verringern, Signalkomponenten oberhalb der Nyquistfrequenz  $f_S/2$  fallen bei der Abtastung zurück ins Nutzband (Aliasing; hier in der Aufgabenstellung ausgeschlossen). Die wahrscheinlichste Ursache ist jedoch, dass der Eingangsspannungsbereich nicht ausgenutzt wird: Die Signalamplitude  $A$  bestimmt die Signalleistung  $S$ , während die Rauschleistung  $N_Q$  unabhängig von  $A$  ist.

Ein Aussteuerbereich von  $FSR = 2 \text{ V}$  entspricht einer maximalen Signalamplitude  $A_{max} = FSR/2 = 1 \text{ V}$ .

$$\begin{aligned} SQNR &= \frac{S}{N_Q} = \frac{S}{N_Q} \frac{S_{max}}{S_{max}} = SQNR_{max} \frac{S}{S_{max}} = SQNR_{max} \frac{A^2/2}{A_{max}^2/2} \\ \Rightarrow SQNR[\text{dB}] &= SQNR_{max}[\text{dB}] + 10 \log_{10} \frac{S}{S_{max}} = SQNR_{max}[\text{dB}] + 20 \log_{10} \frac{A}{A_{max}} \\ \Rightarrow A &= A_{max} 10^{(SNR-SQNR_{max})/20} = 1 \text{ V} \cdot 10^{8,78/20} = \mathbf{0,36 \text{ V}} \end{aligned}$$

Hat man bereits die  $ENOB$  des ADCs berechnet, könnte man auch daraus die Eingangsamplitude bestimmen, allerdings werden die  $ENOB$  normalerweise bei Vollaussteuerung bestimmt.

- c) Bei der (Re-)Quantisierung eines Signals auf eine LSB-Größe von  $q$  entsteht ein Fehler, der im zeitlichen Mittel eine **Rauschleistung**  $N_Q = q^2/12$  hat. Diese Rauschleistung ist unabhängig von der Abtastfrequenz, so wie die Leistung eines Sinussignals unabhängig von dessen Frequenz ist! Die Rauschleistung ist außerdem unabhängig von der Amplitude des zu quantisierenden Signals.<sup>1</sup>

Die Größe eines LSBs hängt ab vom Aussteuerbereich (Full Scale Range,  $FSR$ ) und der Anzahl der Bits:  $q = FSR/(2^W - 1) \approx FSR/2^W$  für  $2^W \gg 1$ . Hier ist  $FSR = 2 \text{ V}$  und  $W = 10$ :

$$N_Q = \frac{q^2}{12} = \frac{FSR^2}{12 \cdot 2^{2W}} = \frac{4V^2}{12 \cdot 2^{20}} = 3,179 \cdot 10^{-7} \text{ V}^2 \quad [\text{exakte Rechnung: } 3,185 \cdot 10^{-7} \text{ V}^2]$$

---

<sup>1</sup>Die Amplitude muss natürlich so groß sein, dass überhaupt ein paar Quantisierungsstufen ausgesteuert werden.

Das Quantisierungsrauschen soll auch hier als weiß betrachtet werden, so dass sich die Rauschleistung gleichmäßig über das gesamte (positive) Basisband  $f_S/2$  verteilt. Die **Rauschleistungsdichte**  $N'_Q$  ist

$$N'_Q(f_{S1}) = \frac{N_Q}{f_{S1}/2} = \frac{3,17 \cdot 10^{-7} \text{ W}}{5 \text{ kHz}} = 6,36 \cdot 10^{-11} \text{ W/Hz bzw.}$$

$$N'_Q(f_{S2}) = \frac{N_Q}{f_{S2}/2} = \frac{6,34 \cdot 10^{-7} \text{ W}}{500 \text{ kHz}} = 6,36 \cdot 10^{-13} \text{ W/Hz}$$

Aus der Rauschleistungsdichte  $N'_Q$  berechnet man die Rauschleistung  $N_Q$ , indem man  $N'_Q$  mit der Rauschbandbreite  $B_N$  multipliziert: Lässt man z.B. den ADC mit einer Abtastrate von 1 MHz laufen, will damit aber Signale nur bis zu 50 kHz wandeln, kann man das Quantisierungsrauschen zwischen 50 kHz und  $f_S/2$  mit einem digitalen Tiefpass der Bandbreite  $B = 50 \text{ kHz}$  eliminieren:

$$N_Q = B_N N'_Q(f_{S2}) = B_N \frac{N_Q}{f_{S2}/2} = \frac{50 \text{ kHz}}{500 \text{ kHz}} N_Q = \frac{N_Q}{10} = 3,18 \cdot 10^{-8} \text{ W}$$

Die gesamte Rauschleistung wird also auf ein Zehntel reduziert, was einer Verbesserung der *ENOB* um  $10/6 = 1,7$  bit entspricht.

- d) **FFT eines ADC-Signals:** Die Gesamtrauschleistung (Quantisierungsrauschen und thermisches Rauschen) verteilt sich auf die  $N_{FFT}/2 = 1024$  Bins in Abb. 7.4, im Mittel wurde je Bin gemessen  $N' = -125,4 \text{ dBW/Bin}$ . Die Gesamtrauschleistung ist daher

$$N = N' N_{FFT}/2 \hat{=} N'[\text{dBW}] + 10 \log_{10} (N_{FFT}/2) = -125,4 \text{ dBW} + 30,1 \text{ dB} = -95,3 \text{ dBW}$$

Die Signalleistung ist  $S = -3 \text{ dBW}$ , der Wandler ist also voll ausgesteuert (warum?<sup>2</sup>). Daraus ergeben sich

$$\text{SNR} = S - N = 92,3 \text{ dB und}$$

$$\text{ENOB} = (\text{SNR} - 1,76 \text{ dB})/6,02 = 15,0 \text{ bit.}$$

## M7.2 SQNR eines quantisierten Signals → A7.2

- a) **Das maximale Signal-to-Quantization Noise Ratio ( $SQNR_{max}$ ) bei sinusförmigem Signal, das den gesamten Aussteuerbereich ausnutzt, ist gegeben durch**

$$SQNR_{max} = 1,76 \text{ dB} + W \cdot 6,02 \text{ dB} = 98,1 \text{ dB für } W = 16$$

- b) **Signal-to-Quantization Noise Ratio  $SQNR$  bei reduzierter Signalamplitude**

Wenn  $x[n]$  nur ein Zehntel des Aussteuerbereichs ausnutzt, beträgt die Signalleistung nur noch ein Hundertstel, also -20 dB. Das  $SQNR$  reduziert sich damit auf 78,1 dB. Das entspricht einer *effective number of bits (ENOB)* von **12,7 bit**.

## M7.3 Quantisierungsrauschen im FIR-Filter → A7.3

Die Signalleistung am Ein- und Ausgang ist im Durchlassbereich des Filters gleich, da hier der Betragsgang 1 ist. Sie beträgt  $S = A^2/2 = FSR^2/8 = 0,5 \text{ W} \hat{=} \mathbf{-3,01 \text{ dBW}}$ .

Detaillierte Rechenergebnisse finden Sie in Tab. M7.1.

---

<sup>2</sup> $S = A^2/2 \hat{=} 20 \log_{10} A - 3 \text{ dB}$ . Also muss  $A = 0 \text{ dBVpk}$  und damit voll ausgesteuert sein.

a) Quantisierung am Ausgang auf 16 bit mit  $Q_{DAC}(0.15)$  vor DAC

$$N_{Q,DAC} = q^2/12 = 2^{-30}/12 \text{ W} = 7,76 \cdot 10^{-11} \text{ W} \hat{=} 10 \log_{10} (7,76 \cdot 10^{-11}) = -101,1 \text{ dBW}$$

$$\Rightarrow SQNR = S/N_{Q,DAC} = 98,1 \text{ dB}$$

Da dieser Quantisierer direkt am *Ausgang* liegt, gilt  $N_{Q,out} = N_{Q,DAC}$ , die *ENOB* ist ***ENOB* = (*SQNR* − 1,76)/6,02 = 16 bit** (nachrechnen!).

**Achtung:** Nicht der DAC selbst, sondern die *vorherige* Verringerung der Wortbreite auf 16 bit mit  $Q_{DAC}(0.15)$  verursacht das Quantisierungsrauschen!

b) Quantisierung am Eingang durch 15 bit ADC  $Q_{ADC}(0.14)$

Der 15 - Bit ADC erzeugt ein Quantisierungsrauschen von

$$N_{Q,ADC} = q^2/12 = 2^{-28}/12 \text{ W} = 3,10 \cdot 10^{-10} \text{ W} \hat{=} -95,1 \text{ dBW.}$$

Da diese Quantisierung am *Eingang* des Filters wirkt, wird das Quantisierungsrauschen mit dem Frequenzgang des Filters bewertet. Das hier betrachtete Filter ist ein Halbbandfilter, daher beträgt die Rauschbandbreite  $B_N \approx f_S/4$ . Für die ausführliche Rechnung wird zunächst die Rauschleistungsdichte bestimmt,  $N'_{Q,ADC} = N_{Q,ADC}/(f_S/2)$ .

Daraus erhält man die mit der Bandbreite gewichtete Rauschleistung am Ausgang:

$N_{Q,ADC,out} = B_N N'_{Q,ADC} = N_{Q,ADC}/2 = 1,55 \cdot 10^{-10} \text{ W} \hat{=} -98,1 \text{ dBW}$ . Mit etwas Übung erkennt man den Faktor 1/2 auch ohne Umweg über die Rauschbandbreite.

$$\Rightarrow SQNR = S - N_{Q,ADC,out} = 95,1 \text{ dB} \Rightarrow \textbf{ENOB} = 15,5 \text{ bit.}$$

Die *ENOB* wurde also durch die Filterung des Quantisierungsrauschen im Vergleich zum 15 Bit ADC um 1/2 bit verbessert.

**Anmerkung:** Die Simulation ergibt  $SQNR = 95,9 \text{ dB}$  und  $ENOB = 15,6 \text{ dB}$ , an der Abweichung ist die (zu) einfache Berechnung der Rauschbandbreite schuld:  $H(f_S/4) = 1/2$ , dementsprechend ist die Ausgangsleistung bei dieser Frequenz nur noch 1/4. Die Bandbreite für die Rauschleistung ist also kleiner als  $f_S/4$ , das *SQNR* daher etwas höher als berechnet.

c) Quantisierung durch  $Q_{ADC}(0.14)$  und durch  $Q_{DAC}(0.15)$

Die Rauschbeiträge von  $Q_{DAC}$  und ADC zum Gesamtrauschen am Ausgang wurden bereits berechnet, daher müssen nur noch beide Leistungen aufaddiert werden (es wird angenommen, dass beide Rauschbeiträge unkorreliert sind):

$$N_{Q,out} = N_{Q,ADC,out} + N_{Q,DAC} = 1,55 \cdot 10^{-10} \text{ W} + 7,76 \cdot 10^{-11} \text{ W} = 2,33 \cdot 10^{-10} \text{ W}$$

d) Quantisierung nach jeder Multiplikation mit 16 bit  $Q_{mul}(0.15)$

Die Quantisierung nach den Multiplizierern wirkt sich als Quantisierungsrauschen direkt am Ausgang aus,  $N_{Q,out} = 7N_{Q,Mul} = 7N_{Q(0.15)} = 7 \cdot 7,76 \cdot 10^{-11} \text{ W} = 5,43 \cdot 10^{-10} \text{ W}$ .

e) Rauschbeiträge von  $Q_{mul}$  sollen kleiner sein als die von DAC und ADC:

$$N_{Q,out,mul} = 7N_{Q,Mul} \leq N_{Q,ADC}/2 + N_{Q,DAC} = 2,33 \cdot 10^{-10} \text{ W}$$

$$\Rightarrow N_{Q,Mul} = 2^{-2WF_{mul}}/12 \leq 2,33 \cdot 10^{-10} \text{ W}/7 = 3,33 \cdot 10^{-11} \text{ W}$$

$$\Rightarrow WF_{mul} = \lceil -0,5 \log_2(12 \cdot 3,33 \cdot 10^{-11}) \rceil = 16 \text{ bit}$$

Nach den Multiplizierern muss also auf  $Q_{mul}(0.16)$  requantisiert werden, alle 7 Quantisierer erzeugen zusammen eine Rauschleistung von

$$N_{Q,out,mul} = 7N_{Q,Mul} = 7N_{Q(0.16)} = 7 \cdot 1,94 \cdot 10^{-11} \text{ W} = 1,36 \cdot 10^{-10} \text{ W.}$$

Das Gesamtrauschen am Ausgang in diesem Fall ist:

$$N_{Q,out} = 7N_{Q,Mul} + N_{Q,ADC}/2 + N_{Q,DAC} = (1,36 + 2,33) \cdot 10^{-10} \text{ W} = \mathbf{3,69 \cdot 10^{-10} \text{ W}} \\ \cong -94,33 \text{ dBW}$$

$$SQNR = -3,01 \text{ dB} - N_{Q,out} = \mathbf{91,31 \text{ dB}}$$

$$ENOB = (SQNR - 1,76 \text{ dB}) / 6,02 \text{ dB/bit} = \mathbf{14,9 \text{ bit}}$$

- f) Die **Rauschbeiträge von  $Q_{mul}$**  sollen die Gesamt- $ENOB$  aus Unterpunkt c) nur um 0,1 bit, also von  $ENOB = 15,2$  bit auf 15,1 bit verringern:

$$ENOB = (SNR \text{ (dB)} - 1,76 \text{ dB}) / 6,02 \text{ dB/bit} \stackrel{!}{=} 15,1 \text{ bit} \Rightarrow SNR = 92,66 \text{ dB} \\ \Rightarrow N_{Q,out,max} = -SNR - 3,01 \text{ dB} = -95,67 \text{ dB} \cong 2,71 \cdot 10^{-10} \\ = 7N_{Q,Mul,max} + N_{Q,ADC}/2 + N_{Q,DAC} \\ \Rightarrow N_{Q,Mul,max} = (2,71 \cdot 10^{-10} - 2,33 \cdot 10^{-10}) / 7 = 5,41 \cdot 10^{-12} 2^{-2WF_{mul}} / 12 \\ \Rightarrow WF_{mul} = \lceil -0,5 \log_2(12 \cdot 5,41 \cdot 10^{-12}) \rceil = \mathbf{17 \text{ bit}}$$

Nach den Multiplizierern muss also mindestens auf das Format  $Q_{mul}(0.17)$  requantisiert werden.

- g)  **$SQNR$  für Testsignal mit der Frequenz  $F_1 = 1/4$ :**

Da das Filter ein Halbbandfilter ist, ist die Signalamplitude am Ausgang des Filters nur noch 1/2, die Leistung dementsprechend nur noch 1/4  $\cong -6,02 \text{ dB}$ . Alle  $SQNR$ -Werte verringern sich also um 6,02 dB, die zugehörigen  $ENOB$ -Werte um 1 bit.

	(a) $Q_{DAC}$	(b) ADC	(c) $Q_{DAC} + ADC$	(d) $7Q_{Mul}$	(e) Alles
$N_{Q,Out} / \text{W}$	$7,76 \cdot 10^{-11}$	$1,55 \cdot 10^{-10}$	$2,33 \cdot 10^{-10}$	$5,43 \cdot 10^{-10}$	$3,69 \cdot 10^{-10}$
$N_{Q,Out} / \text{dBW}$	-101,10	-98,09	-96,33	-92,65	-94,33
$SQNR / \text{dB}$	98,09	95,08	93,32	89,64	91,31
$ENOB$	16,0	15,5	15,2	14,6	14,9

Tab. M7.1: Zusammenfassung der Ergebnisse aus Aufgabe M7.3

## M7.4 DFT von Breitbandsignalen → A7.4

Die Musterlösung finden Sie im Notebook **NOI-DFT\_wideband\_ML**.

### a) Signalleistungen

Die Nutzsignale sind Sinussignale, deren Leistung  $S = A^2/2$  bei kohärenter DFT auf genau ein Frequenzbin (einseitiges Spektrum) bzw. zwei Frequenzbins (zweiseitiges Spektrum) fällt:

$$\begin{aligned} A_a &= \frac{5}{\sqrt{2}} \text{ V}_{rms} & \Rightarrow S_a = A_a^2 &= 12,5 \text{ W} & \hat{=} & 10 \log_{10} \frac{12,5 \text{ W}}{1 \text{ W}} = 10,97 \text{ dBW} \\ A_b &= \frac{1}{\sqrt{2}} \text{ V}_{rms} & \Rightarrow S_b = A_b^2 &= 0,5 \text{ W} & \hat{=} & 10 \log_{10} \frac{0,5 \text{ W}}{1 \text{ W}} = -3,01 \text{ dBW} \end{aligned}$$

Die Gesamtleistung von sinusförmigen Signalen mit unterschiedlichen Frequenzen ist die Summe der Einzelleistungen (siehe Aufgabe [M3.6f](#)):

$$S = S_a + S_b = 13 \text{ W} \hat{=} 11,14 \text{ dBW}$$

Haben beide Signale die gleiche Frequenz, muss man vektoriell addieren, da konstruktive oder destruktive Interferenz auftreten kann.

Bei nicht-korrelierten Signalen (z.B. Rauschen) werden immer die Leistungen addiert.

### b) Rauschleistung und Signal-to-Noise Ratio

Der Rauschanteil im Notebook **NOI-DFT\_wideband** ist ein Gauß- oder normalverteilter Zufallsprozess mit Varianz (= AC-Leistung)  $\sigma^2 = 0,01 \text{ V}^2 \hat{=} 10 \text{ mW} \hat{=} -20 \text{ dBW}$ . Da der Prozess gleichanteilsfrei ist, ist das gleichzeitig die Gesamtleistung. Die Standardabweichung ist die Quadratwurzel der Varianz, sie beträgt  $\sigma = 0,1 \text{ V}$ .

Das Signal-to-Noise Ratio  $SNR$  erhält man aus dem Verhältnis der Gesamt signal- zur Rauschleistung:

$$SNR = \frac{S}{N} = \frac{13 \text{ W}}{0,01 \text{ W}} = 1300 \hat{=} 11,14 \text{ dBW} - (-20 \text{ dBW}) = 31,14 \text{ dB}$$

### c) Die Rauschleistungsdichte $N'$ ist die Rauschleistung pro Hz:

$$\begin{aligned} N' &= N/f_S & = 10 \text{ mW}/5 \text{ kHz} &= 2 \mu\text{W}/\text{Hz} \text{ (zweiseitiges Spektrum)} \\ \text{bzw. } N' &= N/(f_S/2) & = 10 \text{ mW}/2,5 \text{ kHz} &= 4 \mu\text{W}/\text{Hz} \text{ (einseitiges Spektrum)} \end{aligned}$$

### d) Die mittlere angezeigte Rauschleistung der DFT in der Simulation ist die Rauschleistung pro Frequenzbin:

$$\begin{aligned} N/\text{bin} &= N/N_{FFT} & = 10 \text{ mW}/2000 \text{ bins} &= 5 \mu\text{W}/\text{bin} \text{ (zweis. Spektrum)} \\ \text{bzw. } N/\text{bin} &= N/(N_{FFT}/2) & = 10 \text{ mW}/1000 \text{ bins} &= 10 \mu\text{W}/\text{bin} \text{ (eins. Spektrum)} \end{aligned}$$

### e) Siehe Notebook

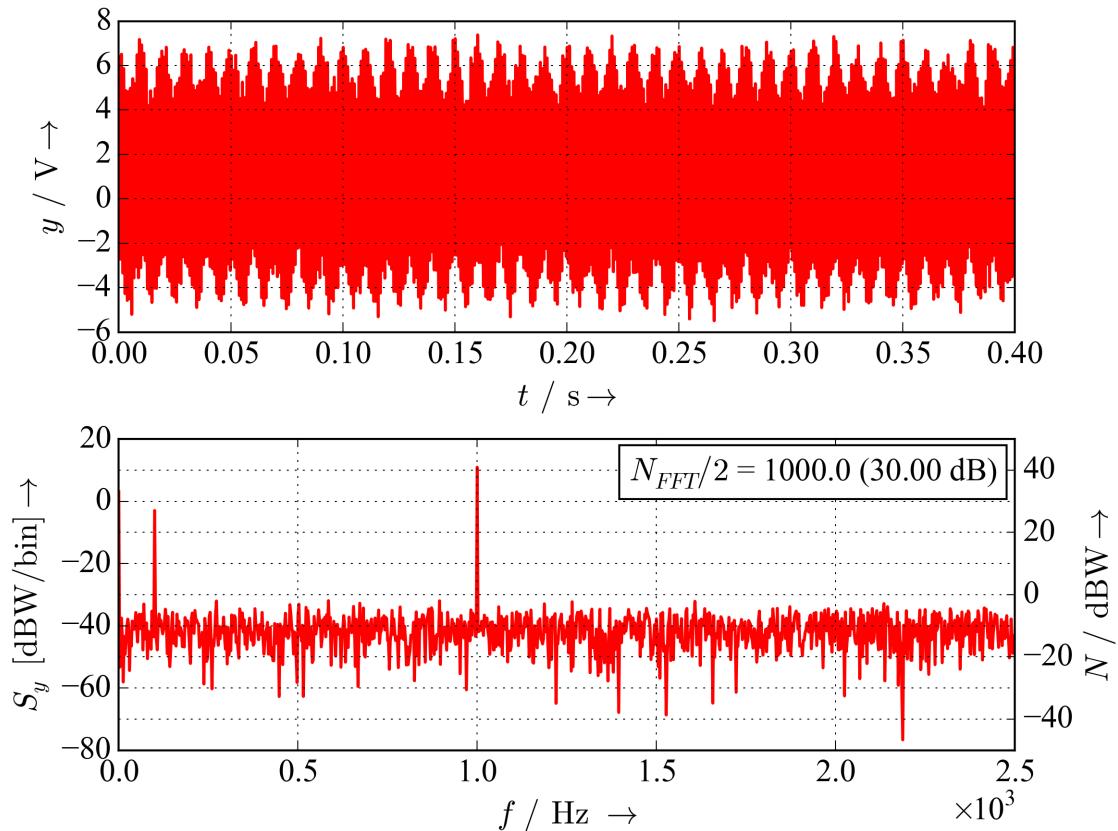


Abb. M7.1: Einseitiges Spektrum von Schmal- und Breitbandsignalen

## M8 SMP: Abtastung und Downsampling

### M8.1 Spektren abgetasteter Signale → A8.1

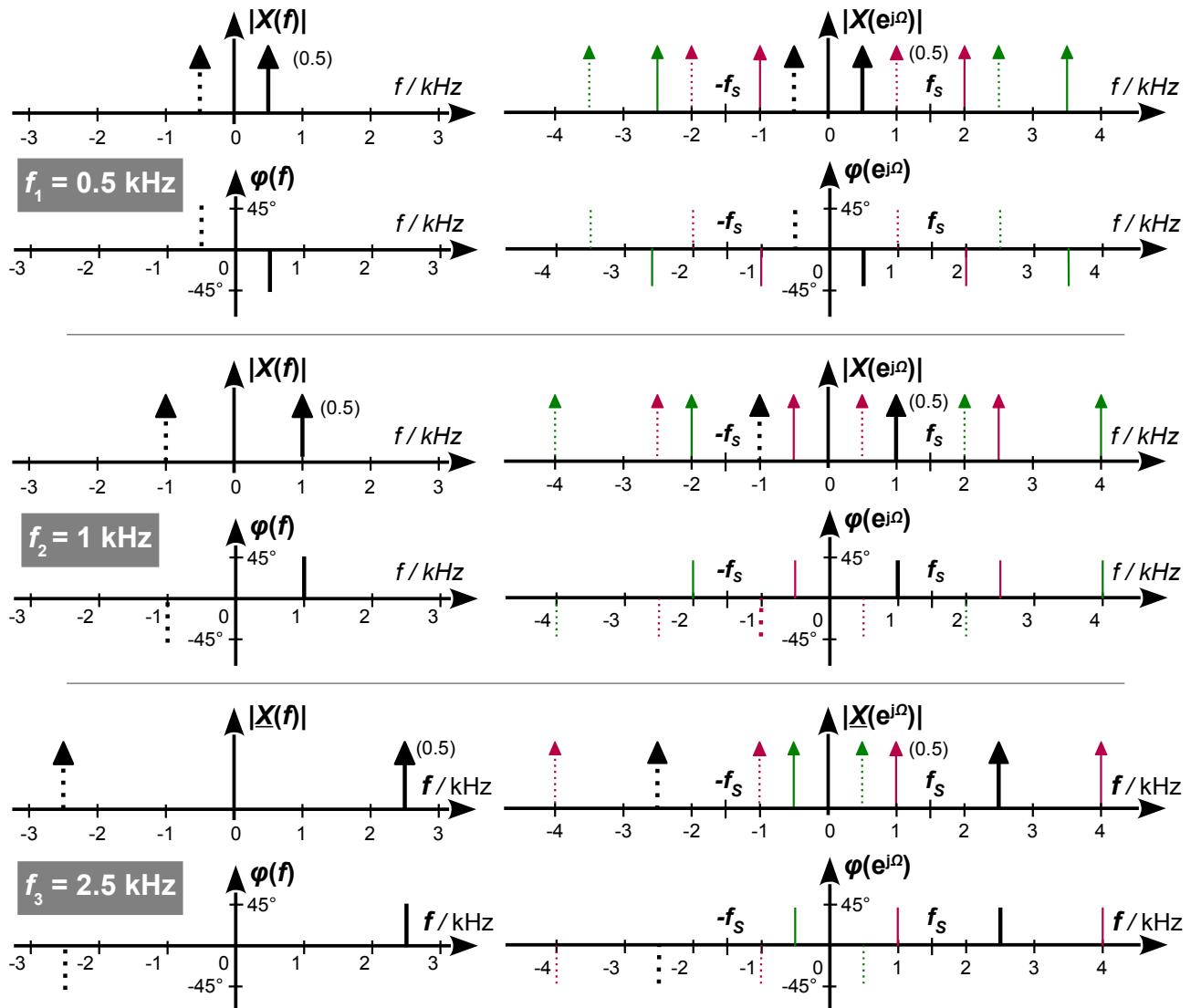


Abb. M8.1: Amplituden- und Phasenspektren von zeitkontinuierlichen (links) und mit  $f_S = 1500$  Hz abgetasteten (rechts) Signalen zu Aufgabe 8.1. Gezeichnet wurden nur die mit  $\pm 2f_S, \pm f_S$  und  $f = 0$  modulierten Produkte.

Die Spektren des abgetasteten Signals werden konstruiert, indem man jede Linie des Betrags- bzw. Phasenspektrums um  $k f_S$  mit  $k = \dots, -2, -1, 0, 1, 2, \dots$  verschiebt.

#### Anmerkung:

Frequenzen und Nullphasen der analogen Signale wurden hier so gewählt, dass sich gleiche Abtastwerte im Zeitbereich (Abb. M8.2) und daher auch gleiche Abtastspektren (Frequenzbereich) für alle drei

Signale ergeben. Bei den Signalen mit 1000 Hz und 2500 Hz findet Aliasing statt! „Fehlende“ Linien in Abb. M8.1 können durch zusätzliche Abtastprodukte aufgrund  $\pm 3f_S$  ergänzt werden.

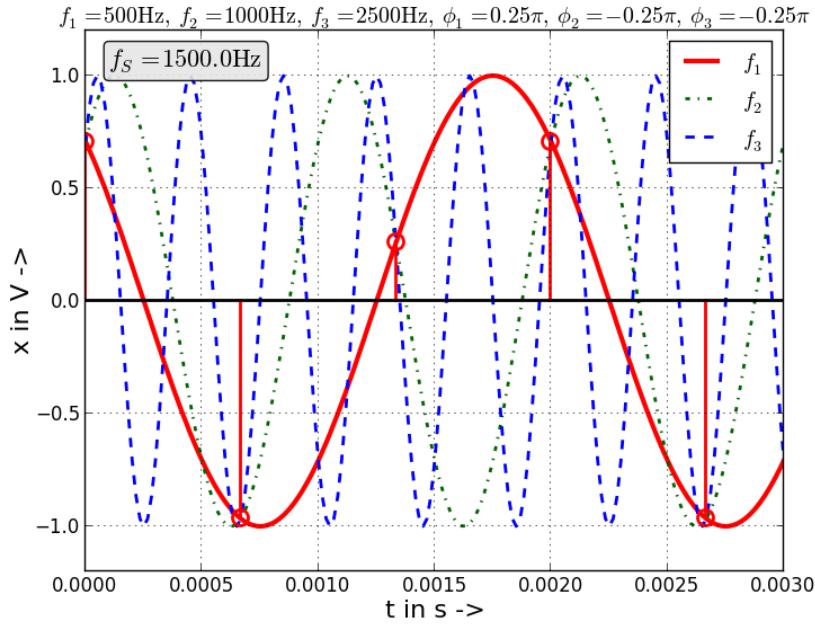


Abb. M8.2: Zeitkontinuierliche Signale mit identischen Abtastwerten zu Aufgabe 8.1.

## M8.2 Analog-Digital-Wandlung mit Oversampling → A8.2

- a) Das **maximale SQNR bei Nyquist-Wandlung** ist gegeben durch

$$SQNR_{max} = (6,02W + 1,76) \text{ dB} = 49,92 \text{ dB} \text{ für } W = 8.$$

- b) Das **maximale SQNR mit Oversampling** ist gegeben durch

$$\begin{aligned} SQNR_{max}(OSR) &= \frac{S}{f_N N'_Q} = \frac{S}{N_Q} \frac{f_S/2}{f_N} = SQNR_{max} \frac{f_S/2}{f_N} = SQNR_{max} OSR \\ &\Rightarrow SQNR[\text{dB}] = SQNR_{max}[\text{dB}] + 10 \log_{10} OSR \end{aligned}$$

$$\Rightarrow SQNR(OSR = 10) = 59,92 \text{ dB} \quad \hat{=} \quad ENOB = (59,9 - 1,8)/6,0 = 9,7 \text{ bit}$$

$$\Rightarrow SQNR(OSR = 50) = 66,91 \text{ dB} \quad \hat{=} \quad ENOB = (66,9 - 1,8)/6,0 = 10,9 \text{ bit}$$

- c) Das **maximale SQNR eines Sigma-Delta-Wandlers** erster Ordnung mit einem Bit ist gegeben durch

$$SQNR_{\Sigma\Delta,max} = -6,6 + 30 \log_{10} OSR = 23,4 \text{ dB } (OSR = 10) \text{ bzw. } 44,4 \text{ dB } (OSR = 50)$$

und für einen Wandler zweiter Ordnung durch

$$SQNR_{\Sigma\Delta,max} = -6,6 + 50 \log_{10} OSR = 43,4 \text{ dB } (OSR = 10) \text{ bzw. } 78,3 \text{ dB } (OSR = 50)$$

## M8.3 Analog-Digital-Wandlung mit Oversampling und nachfolgender Dezimation → A8.3

- a) Das Amplitudenspektrum des Signals wird nicht verfälscht bei der Reduktion der Abtastrate auf  $f_{S,2}/R$ , da das analoge Signal (und damit auch das Basisband des abgetasteten Signals) auf  $B = 12 \text{ kHz} < f_{S2}/(2R)$  bandbegrenzt ist.

Zeitsignal und Phasenspektrum beider Signale sind aber nur dann identisch, wenn bei der Dezimation das gleiche Sample verwendet wird wie bei der Abtastung mit der niedrigen Abtastfrequenz  $f_{S1}$ . Andernfalls tritt ein Zeitversatz zwischen den abgetasteten Sequenzen auf, der das Phasenspektrum beeinflusst.

- b) Maximaler Downsamplingfaktor  $R_{max}$ : Ein analoges AA-Filter muss die Bandbreite des analogen Signals vor der Abtastung auf  $f_{S1}/2$  begrenzen, damit das Signal nicht durch Aliasing bei der Abtastung verfälscht wird. Wird die Abtastrate des Signals anschließend um den Faktor  $R$  verringert (Resampling, Dezimation), entspricht das einer neuerlichen Abtastung mit  $f_{S2}/2 = f_{S1}/(2R)$ . Daher muss die Bandbreite vor der Dezimation geringer als  $f_{S2}/2$  sein:

$$\frac{f_{S1}}{2R_{max}} > B \Leftrightarrow R_{max} = \left\lfloor \frac{f_{S2}}{2B} \right\rfloor = 41$$

- c) Digitales Anti-Aliasfilter: Vor der Verringerung der Abtastrate auf  $f_{S,2} = f_{S,1}/R$  muss ggf. mit einem digitalen Filter die Bandbreite des abgetasteten Signals auf  $f_{S,2}/2 = f_{S,1}/(2R)$  begrenzt werden.
- d) Überabtastung um  $I$  mit nachfolgender Dezimation hat folgende Vorteile:

**Entspannter Entwurf des analogen AA-Filters:** Die Bandbreite des analogen Signals muss nur auf  $I f_{S}/2$  begrenzt werden, hierfür kann ein Filter mit deutlich breiterem Übergangsbereich und daher geringerer Ordnung verwendet werden. Vor der Verringerung der Abtastrate muss u.U. die Bandbreite des abgetasteten Signals reduziert werden, um Aliasing bei der Dezimation zu vermeiden.

**Verbessertes SQNR:** Das Quantisierungsrauschen  $q^2/12$  verteilt sich bei Abtastung mit erhöhter Abtastrate über einen breiteren Frequenzbereich, so dass die Rauschleistungsdichte sinkt. Durch Begrenzung der Bandbreite in einem digitalen Filter (unbedingt notwendig!) wird die Gesamtrauschleistung verringert und so das *SQNR* verbessert.

## M8.4 Downsampling um den Faktor 2 nach Averaging → A8.4

- a) Anti-Aliasfilter für bandbegrenztes Signal: Ohne überlagertes Rauschen ist kein Filter erforderlich, da die Bandbreite  $B$  des Signals geringer ist als die halbe Abtastfrequenz nach dem Downsampling,  $B < f_{S2}/2$ .
- b) Anti-Aliasfilter für verrauschtes Signal: Die Eckfrequenz des Durchlassbands muss  $f_{DB} \geq B = 80 \text{ kHz}$  sein, damit das Nutzband nicht beschnitten wird. In den Frequenzbereich  $0 \dots B$  dürfen keine Frequenzkomponenten zurückgefaltet werden, daher muss die Eckfrequenz des Sperrbands  $f_{SB} < f_{S2} - B = 112 \text{ kHz}$  sein.
- c) Das gesuchte **FIR-Filter** ist ein Moving Average (= gleitender Mittelwert) Filter der Länge zwei mit den Koeffizienten  $h[n] = \{0,5; 0,5\}$ .

- d) Der Betragsgang des Filters mit  $H(z) = 0,5 + 0,5z^{-1}$  ist

$$|H(e^{j\Omega})| = |0,5 + 0,5e^{-j\Omega}| = \left|0,5e^{-j\Omega/2} \left(e^{j\Omega/2} + e^{-j\Omega/2}\right)\right| = |\cos \Omega/2|$$

Mit

$$\Omega_{DB} = 2\pi f_{DB}/f_{S1} = 2\pi \cdot 80/384 \text{ und}$$

$$\Omega_{SB} = 2\pi f_{SB}/f_{S1} = 2\pi \cdot 112/384$$

erhält man die gesuchten Dämpfungen an den Kanten von Durchlass- und Sperrband:

$$|H(\Omega_{DB})| = |\cos(\Omega_{DB}/2)| = 0,793 \hat{=} -2,0 \text{ dB und}$$

$$|H(\Omega_{SB})| = |\cos(\Omega_{SB}/2)| = 0,609 \hat{=} -4,3 \text{ dB}$$

Man sieht, dass dieser Dezimations-Ansatz nur sehr geringe Hardwareressourcen benötigt. In vielen Fällen werden aber weder die relativ hohe Dämpfung im Nutzband noch die schwache Unterdrückung von Aliasing-Komponenten akzeptabel sein. Daher sind oft aufwändigere Filter erforderlich, vor allem bei einer hohen Bandbreite des Nutzsignals.

## M8.5 Effizientes Averaging und Downsampling um den Faktor $N$ → A8.5

- a) Bei der maximalen Dezimationsrate  $R_{max}$  muss gerade noch gelten

$$B = 10 \text{ kHz} < \frac{f_{S2}}{2} = \frac{f_{S1}}{2R_{max}} = \frac{384 \text{ kHz}}{2R_{max}}$$

$$\Rightarrow R_{max} = \left\lfloor \frac{f_{S1}}{2B} \right\rfloor = \left\lfloor \frac{384 \text{ kHz}}{2 \cdot 10 \text{ kHz}} \right\rfloor = 19$$

- b) Betragsgang für  $L = R_{max} = 19$ :

$$|H(f)| = \left| \frac{\sin(L\Omega/2)}{\sin(\Omega/2)} \right| = \left| \frac{\sin(19\pi F)}{\sin(\pi F)} \right| \text{ mit } H(0) = 19$$

- c) Effiziente Implementierung - Erklärung unvollständig!

In Aufgabe M1.1f wurde eine effiziente Implementierung von Moving Average Systemen gezeigt (Abb. M8.3).

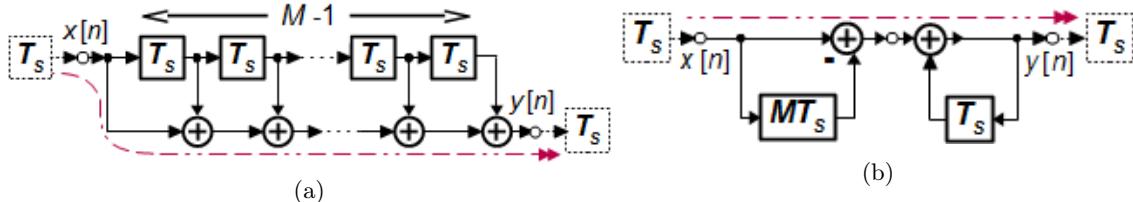


Abb. M8.3: Effiziente Implementierung eines MA-Filters der Ordnung  $N$

Ausgehend vom System Abb. M8.3b kann man die Dezimation um den Faktor  $N$  nach dem Filter auch vor das Filter ziehen und erhält dann ein sog. Hogenauerfilter.

## M8.6 Zweistufige Dezimation → A8.6

In dieser Aufgabe geht es darum, die Abtastrate eines zeitdiskreten Signals in zwei Stufen von  $f_{S1}$  auf  $f_{S3} = f_{S2}/2 = f_{S1}/4$  zu reduzieren. Die obere Grenze des Nutzbands ist hier  $f_N = f_{S3}/3 = f_{S2}/6 = f_{S1}/12$ , das ist auch die *minimale* Frequenz für das Durchlassband beider Tiefpassfilter,  $f_N \leq f_{DB}$ , da ansonsten das Nutzband vom Tiefpassfilter „angeschnitten“ wird. Der Übergangsbereich zwischen Durchlass- und Sperrband soll möglichst breit sein, damit man einen möglichst entspannten Tiefpassentwurf (= niedrige Ordnung) erhält. Die Eckfrequenz für das Durchlassband soll daher so niedrig wie möglich gewählt werden (im Extremfall  $f_N = f_{DB}$ ), die Eckfrequenz des Sperrbands möglichst hoch.

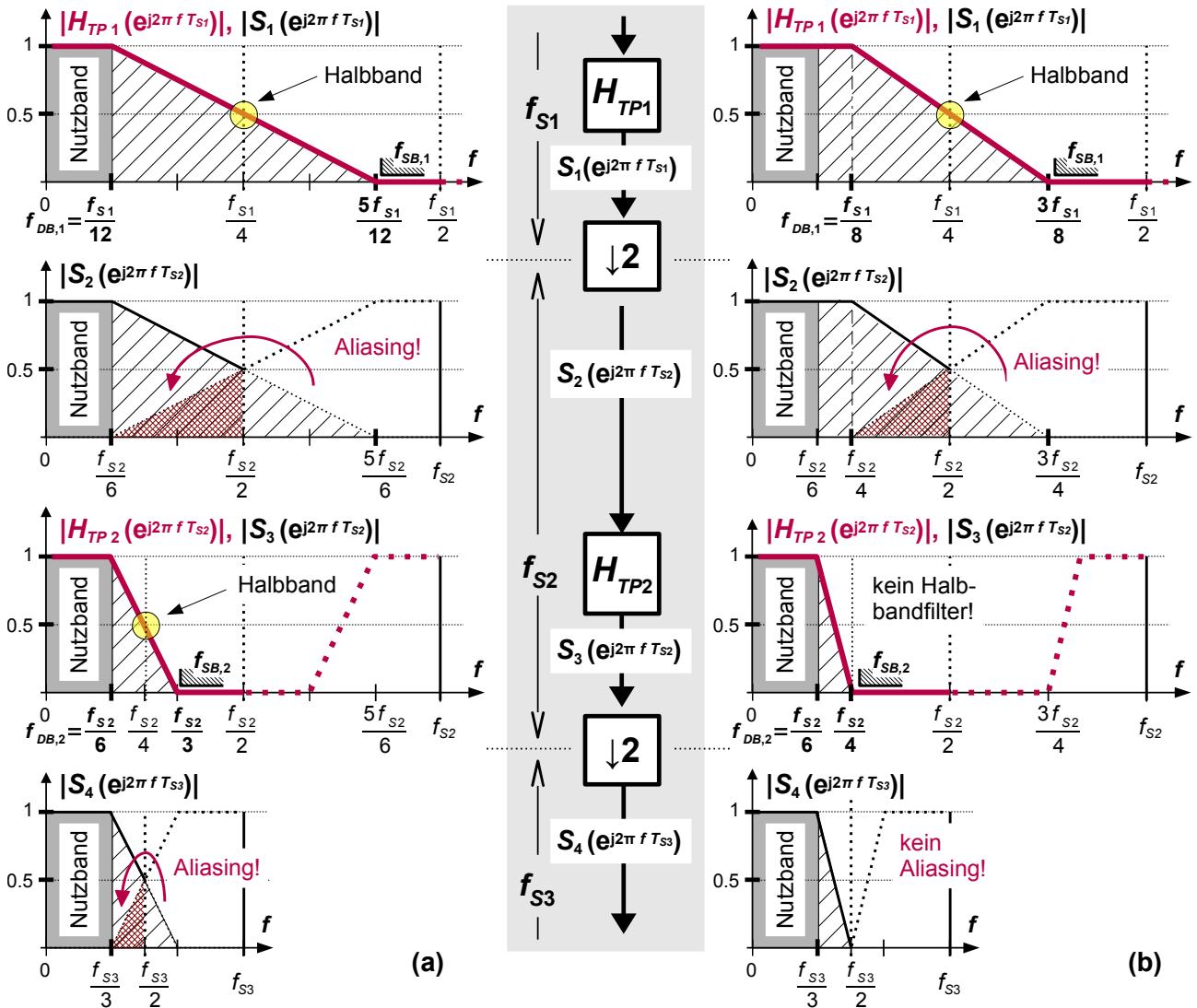


Abb. M8.4: Zweistufige Dezimation, mit (a) und ohne (b) Aliasing ins Basisband am Ausgang zu Aufgabe 8.6a)

### a) Realisierung mit zwei dezimierenden Halbbandfiltern (Abb. M8.4a)

Da es hier keine zusätzlichen Vorgaben gibt, soll die Eckfrequenz für das Durchlassband für beide Tiefpassfilter  $f_N = f_{DB} = f_{DB1} = f_{DB2}$  gewählt werden (also möglichst niedrig) damit der Übergangsbereich möglichst breit wird.

**TP<sub>1</sub>** arbeitet mit  $f_{S1}$ . Um Aliasing bei der nachfolgenden ersten Dezimation um  $R = 2$  vollständig zu vermeiden, müsste die Grenzfrequenz des Sperrbands bei  $f_{SB1} = f_{S1}/4 = f_{S2}/2$  liegen. Hier ist jedoch Aliasing ins Basisband bis an den Rand des Nutzbands gestattet, daher kann

die Grenzfrequenz des Sperrbands höher gewählt werden,  $f_{SB1} = f_{S1}/2 - f_{DB1} = 5/12f_{S1}$ . Mit dieser Wahl sind die Eckfrequenzen von DB und SB symmetrisch zu  $f_{S1}/2$  und man kann man ein Halbbandfilter verwenden. Frequenzkomponenten mit  $f_{S2}/2 < f < f_{SB1}$  werden dann bei der Dezimation  $f_{S2} - f > f_N$  abgebildet (eng schraffiert gekennzeichnet).

Entsprechendes gilt für **TP<sub>2</sub>**: Hier ist  $f_{SB2} = f_{S2}/2 - f_N = f_{S2}/3$ . Man sieht, dass Halbbandfilter gut geeignet sind als Anti-Aliasfilter für eine Dezimation um  $R = 2$ : Mit  $f_{DB} = f_N$  wird ja gleichzeitig  $f_{SB} = f_{S2}/2 - f_{DB}$  festgelegt, dies ist aber die maximale Grenzfrequenz, bei der gerade noch kein Aliasing ins Nutzband auftritt.

Nachteilig bei dieser Lösung ist, dass noch ein drittes Filter (hier nicht gezeichnet) benötigt wird, das die Aliasing-Komponenten im Frequenzbereich zwischen  $f_N = f_{S3}/3$  und  $f_{S3}$  „aufräumt“. Dieses Filter müsste extrem steilbandig sein, da ja  $f_{S3}/3$  auch die obere Grenze des Nutzbandes ist. In der Praxis wird man daher die Grenzfrequenzen der Sperrbänder beider Dezimationsfilter etwas niedriger (und damit automatisch die der Durchlassbänder etwas höher) legen müssen.

### b) Realisierung mit dezimierendem Halbbandfilter und FIR-Filter (Abb. M8.4b)

Die Nachteile der vorigen Lösung werden vermieden, wenn man Aliasing ins Basisband bei der zweiten Dezimation verbietet und dafür die Filterübergänge etwas steiler auslegt. Am besten beginnt man die Aufgabe vom Ausgang des Multiratenfilters her:

Wenn kein Aliasing ins Basisband bei der Dezimation auf  $f_{S3}$  auftreten darf (unterstes Bild von Abb. M8.4b), muss  $f_{SB,2} \leq f_{S3}/2 = f_{S2}/4$  sein, mit Hinblick auf minimale Steilheit des Filters also  $f_{SB,2} = f_{S2}/4$ . Die Eckfrequenz des DB soll aus dem gleichen Grund möglichst klein sein, also  $f_{DB,2} = f_N = f_{S2}/6$ . Der Frequenzbereich unterhalb  $f_{S2}/4$  wird in der zweiten Stufe nicht unterdrückt, daher darf in der ersten Stufe des Filters kein Aliasing in den Bereich  $0 \dots f_{S2}/4$  auftreten. Die Bedingung für die maximal zulässige Eckfrequenz des Sperrbandes  $f_{SB,1}$  lautet daher:

$$f_{S2} - f_{SB,1} = \frac{f_{S1}}{2} - f_{SB,1} = \frac{f_{S2}}{4} = \frac{f_{S1}}{8} \Leftrightarrow f_{SB,1} = \frac{3f_{S1}}{8}$$

Aufgrund der vorgegebenen Symmetrie des Halbbandfilters folgt daraus automatisch

$$f_{DB,1} = f_{SB,1} - \frac{f_{S1}}{4} = \frac{f_{S1}}{8}$$

## M8.7 Multiraten-Tiefpassfilter → A8.7

- a) In Abb. M8.5 ist der **Amplitudengang der Kaskade** aus TP<sub>1</sub> und TP<sub>2</sub> im Bereich  $0 \dots f_{S1}/2 = 3f_{S2}/2 = 3f_{S3}$  dargestellt. Hierfür wurden die Amplitudengänge von TP<sub>1</sub> und TP<sub>2</sub> im gleichen Frequenzmaßstab aufgetragen, aufgrund der geringeren Abtastfrequenz sind bei TP<sub>2</sub> im Bereich  $f_{S1}/6 \dots f_{S1}/2 (\cong f_{S2}/2 \dots 3f_{S2}/2)$  Wiederholspektren eingetragen. In diesem Bereich sollte daher der Begriff „Amplitudengang“ nur mit Vorsicht benutzt werden: Beim ersten Downsampling werden Frequenzkomponenten aus diesem Bereich in das Basisband  $0 \dots f_{S2}/2 = f_{S1}/6 = 8$  kHz abgebildet, nach der zweiten Downsamplingstufe ist das Basisband sogar auf  $0 \dots f_{S3}/2 = f_{S2}/4 = f_{S1}/12 = 4$  kHz begrenzt.

Man kann jedoch den „Amplitudengang“ auf folgende Art benutzen: Zunächst ermittelt man mit Hilfe des kaskadierten Frequenzgangs (Abb. M8.5d) die Dämpfung, die eine bestimmte Frequenzkomponente erfährt. Danach errechnet man (oder ermittelt mit dem „Faltungsschema“ Abb. M8.5f) auf welche Ausgangsfrequenz die Eingangsfrequenz abgebildet wird.

- b) Für die Berechnung der **Aliasfrequenzen** nach der ersten Dezimation ( $f_{S2}$ ) und am Ausgang der Kaskade ( $f_{S3}$ ) kann man entweder  $f_i + kf_S$  berechnen und die Frequenzen suchen, die im Basisband  $-f_S/2 \dots f_S/2$  landen. Alternativ rechnet man normalisiert  $f_i/f_S + k$  und sucht alle Ergebnisse im Bereich  $-\frac{1}{2} \dots \frac{1}{2}$ . Dabei ist jeweils  $k = \dots, -2, -1, 0, 1, 2, \dots$

- c) Für die Berechnung der **Gesamtdämpfung der Testtöne** bestimmt man zunächst deren Dämpfung durch TP1 bei der ursprünglichen Frequenz. Danach bestimmt man ob und auf welche Frequenz der Ton bei der ersten Dezimation abgebildet wird und bestimmt die Dämpfung durch TP2 bei dieser Frequenz. Beispiel: Der Testton bei  $f_2 = 7/24f_{S1} = 14$  kHz wird von TP1 um 40 dB gedämpft. Da die Frequenz außerhalb des Basisbands von  $f_{S2}$  liegt, wird sie bei der 1. Dezimation auf  $\tilde{f}_2 = f_{S2} - f_2 = 1/24f_{S1} = 2$  kHz abgebildet. Diese Frequenz liegt im DB von TP2 und wird daher um 0 dB gedämpft. Insgesamt wird bei dieser Frequenz Aliasing daher nur um 40 dB gedämpft (siehe Tab. M8.1).

$f_{in}$	$f_{DB}$	$f_{SB}$	$f_1$	$f_2$	$f_3$
	2 kHz	4 kHz	8 kHz	14 kHz	16 kHz
$f_{in}/f_{S1}$	1/24	1/12	1/6	7/24	1/3
$f_{in}/f_{S2}$	1/8	1/4	1/2	7/8	1
$f_{in}/f_{S3}$	1/4	1/2	1	7/4	2
$f_{out}$ bei $f_{S2}$	2 kHz	4 kHz	8 kHz	2 kHz	0
$f_{out}$ bei $f_{S3}$	2 kHz	4 kHz	0	2 kHz	0
	$A_{DB}$	$A_{SB}$		$A_{al,i}$	
Soll (dB)	0,1	80	80	80	80
Ist (dB)	0	53	80	40	40

Tab. M8.1: Eigenschaften der Filterkaskade aus Aufgabe 8.7

- d) Ein **verbesserter Filterentwurf** muss zwei Probleme des bestehenden Entwurfs beseitigen:

1. Die **Aliasdämpfung** ist nicht bei allen Eingangsfrequenzen ausreichend: Die Frequenz  $f_2 = 14$  kHz wird beim Downsampling auf  $\tilde{f}_2 = f_{S2} - f_2 = 2$  kHz, die Frequenz  $f_3 = 16$  kHz und damit ins Durchlassband von TP2 abgebildet. Die gewünschte Alias-Unterdrückung von 80 dB bei Eingangsfrequenzen  $f_{in} \geq f_{S2} - f_{DB} = 14$  kHz muss daher TP1 alleine erreichen. Daraus folgt als zusätzliche Spezifikation für TP1  $A_{SB,1a} = 80$  dB bei  $f_{SB,1a} = 14$  kHz.
2. Die **Gesamtdämpfung der Kaskade** bei  $f_{SB} = 4$  kHz ist ebenfalls zu gering. Das lässt sich auf verschiedene Arten verbessern:
  - i. Man kann die Eckfrequenz des Sperrbandes von TP1 auf  $f_{SB,1} = 4$  kHz ( $\hat{F}_{SB,1} = 1/12$ ) reduzieren, dort genügt weiterhin  $A_{SB,1} = 40$  dB.
  - ii. Alternativ kann die Sperrdämpfung von TP2 bei  $f_{SB,2} = 4$  kHz ( $\hat{F}_{SB,2} = 1/4$ ) auf  $A_{SB,2} = 67$  dB erhöht werden.

- e) Der **Filterentwurf** soll ausgehend von Variante i (Spezifikationen von TP2 bleiben unverändert) durchgeführt werden:

Bei einem Equiripple-Filter ist die Dämpfung von 80 dB über das gesamte Sperrband konstant, wählt man  $f_{SB} = 4 \text{ kHz}$  benötigt man  $N = 87$ . Wählt man  $f_{SB} = 4,3 \text{ kHz}$  (Ausprobieren), erreicht man bei 4 kHz immer noch eine Dämpfung > 40 dB und benötigt nur  $N = 72$ .

Beim Entwurf mit Hann-Fenster erreicht man die geforderten Spezifikationen mit  $f_C = 3,17 \text{ kHz}$  und  $N = 89$  (Ausprobieren im Filterdesign-Tool erforderlich, da es hier keinen Minimum-Order-Algorithmus gibt).

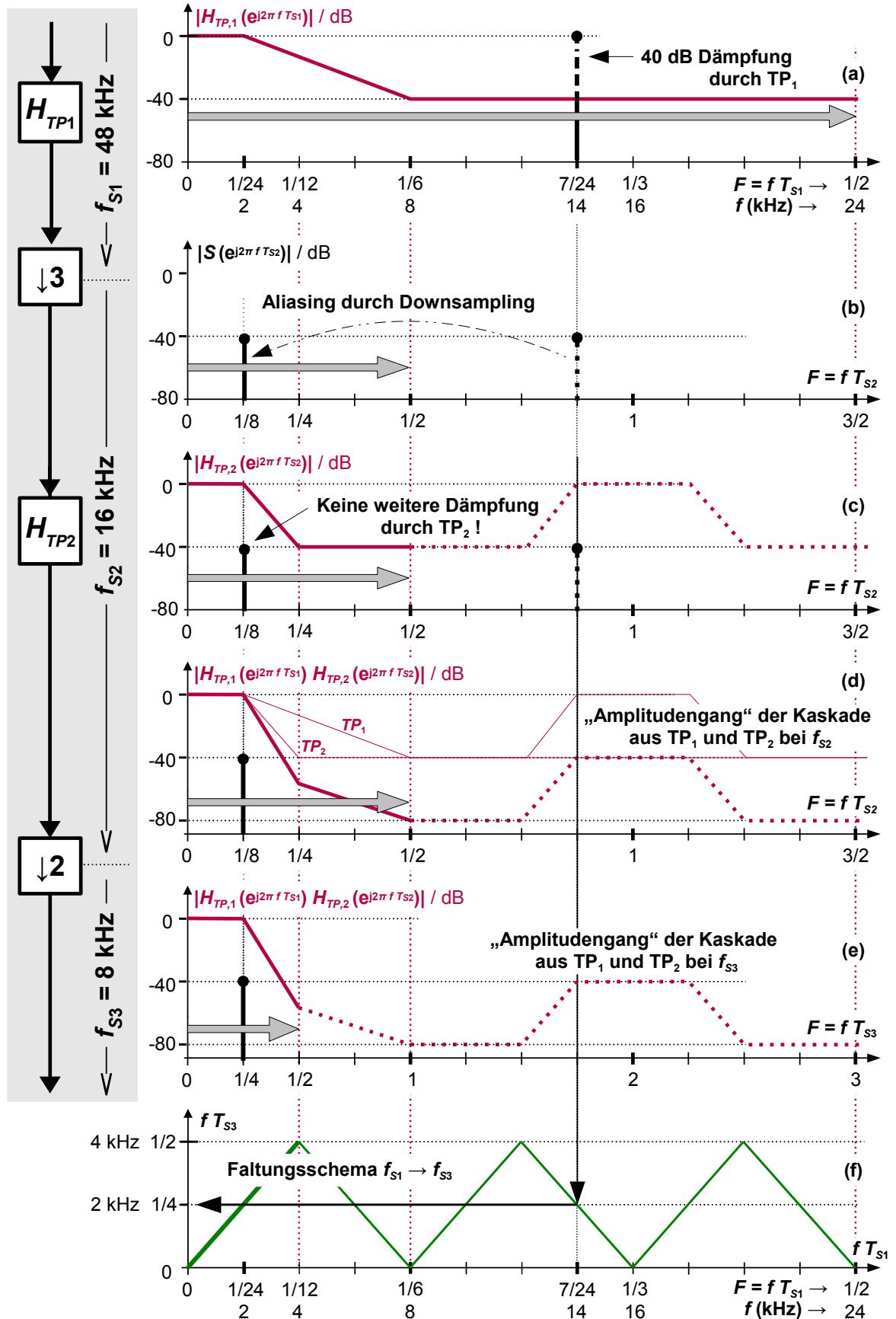


Abb. M8.5: Frequenzgänge der Multiraten-Filterkaskade zu Aufgabe 8.7. Graue Pfeile symbolisieren die jeweiligen Basisbänder.



# M9 INP: Upsampling, Interpolation und Digital-Analog Wandlung

## M9.1 Wiederholspektren und Images → A9.1

In Abb. 9.2 ist das Signalspektrum nur für die positive Hälfte des Basisbands  $0 \dots f_{S1}/2 = 24$  kHz eingezeichnet. Zur Konstruktion des Wiederholspektrums vor der Abtastratenerhöhung ergänzt man am besten in Gedanken die negative Hälfte des Signalspektrums und kopiert dann das Spektrum so oft man will, jeweils zentriert um  $k \cdot 48$  kHz (Abb. M9.1a). Abtastratenerhöhung durch

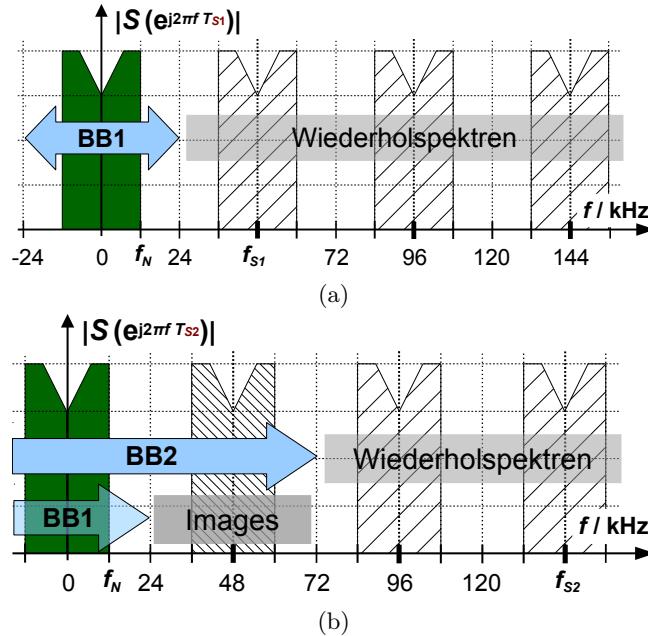


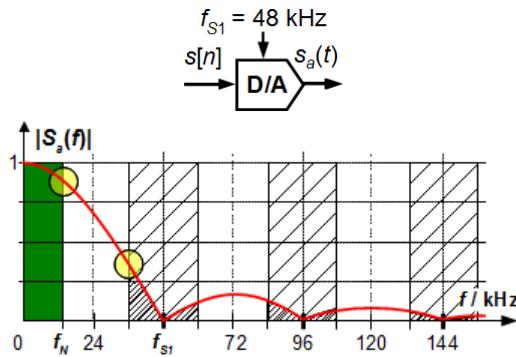
Abb. M9.1: ML: Spektrum von  $s[n]$ , (a) vor und (b) nach Abtastratenerhöhung um  $I = 3$

Nullenstopfen ändert nichts am prinzipiellen Aussehen des Spektrums, das daher genauso konstruiert wird in Punkt a). Alle Spektralanteile im neuen Basisband bis  $f_{S2}/2 = 72$  kHz sind jetzt aber „real“; alle Anteile außer dem ursprünglichen Signalspektrum nennt man „Images“ (Abb. M9.1b). Images können unabhängig vom ursprünglichen Signalspektrum verändert werden, die Wiederholspektren der Images ändern sich dann natürlich genauso.

## M9.2 Verschiedene Arten der Digital-Analog-Wandlung → A9.2

### a) Direkte D/A-Wandlung des digitalen Audiosignals mit Abtastfrequenz $f_{S1}$

Bei direkter D/A-Wandlung wird das analoge Ausgangssignal jeweils für die Periode  $T_{S1}$  gehalten, das Wiederholspektrum des digitalen Signals (weit schraffiert) wird dadurch mit der Funktion  $H_{DAC}(f) = \text{si}(\pi f T_{S1})$  gewichtet, die die erste Nullstelle bei  $f_{S1}$  hat. Im analogen Ausgangssignal bleiben die dicht schraffiert gezeigten Images übrig.



$$a_N \quad \text{si}(\pi f_N/f_{S1}) = \text{si}(\pi/4) = 0,90 \\ \equiv -0,91 \text{ dB}$$

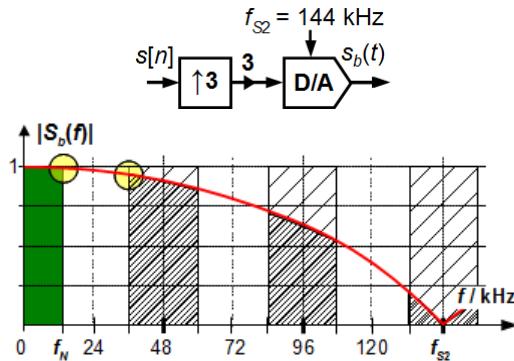
$$f_{im} \quad f_{S1} - f_N = 36 \text{ kHz}$$

$$a_{im} \quad \text{si}(\pi f_{im}/f_{S1}) = \text{si}(3\pi/4) = 0,30 \\ \equiv -10,5 \text{ dB}$$

### b) Überabtastung um den Faktor $I = 3$ ohne digitale Interpolation

Hier wird das digitale Audiosignal zunächst um den Faktor 3 überabgetastet, zwischen den Signalwerten werden dabei Nullen eingefügt (Nullenstopfen). Dadurch werden die Wiederholstrukturen bis zur neuen Nyquistfrequenz  $f_S/2$  zu Images, also „real“ und könnten separat gefiltert werden (was aber hier nicht getan wird). Außerdem wird das gesamte Spektrum durch das Nullenstopfen um den Faktor 3 gedämpft, dies muss durch ein Verstärkungsglied wieder ausgeglichen werden.

Die D/A Wandlung bei der dreifachen Abtastrate  $f_{S2} = 3f_{S1}$  setzt die digitalen Werte in dreimal schmalere ZOH-Rechteckpulse um, dementsprechend wird das Signalspektrum (mit Wiederholstrukturen) hier mit einer si-Funktion  $H_{DAC}(f) = \text{si}(\pi f T_{S2})$  gewichtet, die die erste Nullstelle bei  $f_{S2}$  hat.



$$a_N \quad \text{si}(\pi f_N/f_{S2}) = \text{si}(\pi/12) = 0,99 \\ \equiv -0,10 \text{ dB}$$

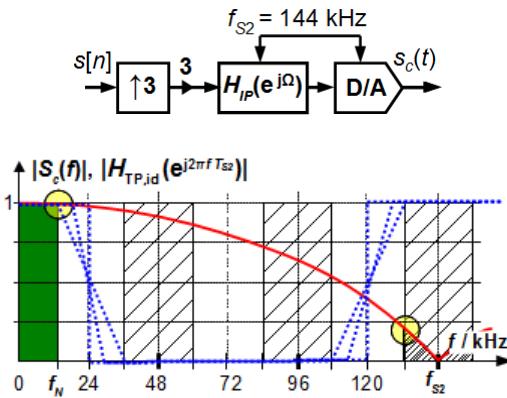
$$f_{im} \quad f_{S1} - f_N = 36 \text{ kHz}$$

$$a_{im} \quad \text{si}(\pi f_{im}/f_{S2}) = \text{si}(\pi/4) = 0,90 \\ \equiv -0,91 \text{ dB}$$

Die Erhöhung der Abtastrate ohne entsprechende digitale Filterung vor der D/A Wandlung „schont“ zwar das Nutzband, aber die Images sind deutlich stärker als in Unterpunkt a).

### c) Überabtastung um den Faktor $I = 3$ mit idealer digitaler Interpolation

Ideale Interpolation heißt, dass im neuen Basisband  $0 \dots f_{S2}/2 = If_{S1}/2$  alle Images des alten Basisbands  $0 \dots f_{S1}/2$  unterdrückt werden. Die maximale Eckfrequenz des Sperrbands ist daher die niedrigste Imagefrequenz,  $f_{SB} = f_{im,1} = f_{S1} - f_N = 36 \text{ kHz}$ . Das Nutzband  $0 \dots f_N < f_{S1}/2$  muss das Interpolationsfilter ohne Dämpfung passieren können, die minimale Eckfrequenz des Durchlassbands beträgt daher  $f_{DB} = f_N = 12 \text{ kHz}$ . Die gestrichelten Linien im Bild unten erfüllen alle die Bedingung für ein ideales Interpolationsfilter.



$$a_N \quad \text{si}(\pi f_N / f_{S2}) = \text{si}(\pi/12) = 0,99 \\ \equiv -0,10 \text{ dB}$$

$$f_{im} \quad f_{S2} - f_N = 132 \text{ kHz}$$

$$a_{im} \quad \text{si}(\pi f_{im} / f_{S2}) = \text{si}(11\pi/12) \\ \equiv -20,9 \text{ dB}$$

Da der DAC hier wie in Unterpunkt b) mit  $f = f_{S2}$  arbeitet, hat sein Betragsgang auch hier die erste Nullstelle bei  $f_{S2}$ . Da aber durch digitale Tiefpassfilterung alle Images des alten Basisbands bereits entfernt wurden, tritt das erste Image nach der D/A Wandlung erst bei  $f_{im} = f_{S2} - f_N$  auf.

#### d) Überabtastung um den Faktor $I = 3$ mit digitalem Zero-Order Hold

Die Wiederholung des letzten Abtastwerts (digitales Zero-Order Hold, ZOH) beim Upsampling um  $I = 3$  anstelle von Nullenstopfen lässt sich signaltheoretisch beschreiben durch Nullenstopfen, gefolgt von einem Moving-Average (MA) Filter der Länge  $L = I = 3$ ,<sup>1</sup> das mit  $f_{S2} = 3f_{S1}$  getaktet wird (Abb. M9.2).

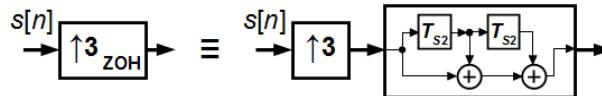
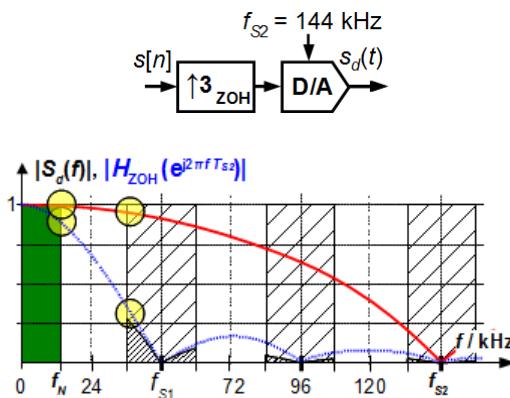


Abb. M9.2: Ersatzschaltbild für Interpolation mit ZOH zu Aufgabe 9.2 d)

Das MA-Filter hat eine Verstärkung von 3 bei DC und übernimmt damit gleichzeitig die Amplitudenkorrektur nach dem Nullenstopfen. Die Filtercharakteristik eines MA-Filters der Länge  $N = 3$  ist alles andere als ideal (vgl. Aufgabe 2.7), sie wird beschrieben durch die Dirichletfunktion  $\text{di}_3(\Omega)$  („periodische“ oder „aliased si-Funktion“):

$$|H_{MA,3}(e^{j2\pi f T_{S2}})| = 3 \text{ di}_3(2\pi f T_{S2}) = \frac{\sin 3\pi f T_{S2}}{\sin \pi f T_{S2}} = \frac{\sin \pi f T_{S1}}{\sin \frac{\pi f T_{S1}}{3}} \approx \frac{\sin \pi f T_{S1}}{\pi f T_{S1}/3} = 3 \text{ si} \pi f T_{S1}$$

Im neuen Basisband  $0 \dots f_{S2}/2 = 3f_{S1}/2$  werden jetzt nicht mehr alle Images des alten Basisbands  $0 \dots f_{S1}/2$  unterdrückt, das Nutzband  $0 \dots f_N < f_{S1}/2$  wird dagegen bei der Eckfrequenz  $f_{DB} = f_N = 12 \text{ kHz}$  durch das „Interpolationsfilter“ schon beträchtlich um  $a_{N,I} = 0,9 \text{ dB}$  gedämpft.



$$a_{N,MA} \quad \text{di}_3(2\pi f_N / f_{S2}) \approx \text{si}(\pi f_N / f_{S1}) \\ = \text{si}(\pi/4) \equiv -0,91 \text{ dB}$$

$$a_{N,DAC} \quad \text{si}(\pi f_N / f_{S2}) = \text{si}(\pi/12) \equiv -0,10 \text{ dB}$$

$$a_N \quad a_{N,MA} a_{N,DAC} = 0,89 \equiv -1,01 \text{ dB}$$

$$f_{im} \quad f_{S1} - f_N = 36 \text{ kHz}$$

$$a_{im,MA} \quad \text{di}_3(2\pi f_{im} / f_{S2}) \approx \text{si}(\pi f_{im} / f_{S1}) \\ = \text{si}(3\pi/4) \equiv -10,5 \text{ dB}$$

$$a_{im,DAC} \quad \text{si}(\pi f_{im} / f_{S2}) = \text{si}(\pi/4) \equiv -0,91 \text{ dB}$$

$$a_{im} \quad a_{im,MA} a_{im,DAC} = 0,27 \equiv -11,4 \text{ dB}$$

<sup>1</sup>entsprechend einer Ordnung von  $N = 2$

Der Betragsgang der gesamten Struktur und damit die Dämpfung im Nutzband  $a_N$  bzw. bei der niedrigsten Imagefrequenz  $a_{im}$  ist das Produkt aus

- dem Betragsgang  $H_{MA,3}(f)$  der Abtastratenerhöhung mit Wiederholung des letzten Werts (digitales Zero-Order-Hold), einer  $\text{di}_3(f/f_{S1})$ -Funktion (blau gestrichelte Linie)
- und dem Betragsgang  $H_{DAC}(f)$  des DACs mit analogem Zero-Order-Hold, einer  $\text{si}(f/f_{S2})$ -Funktion (rote durchgezogene Linie).

e) **Vergleicht man die Spektren** aus a) - d) fallen die folgenden Unterschiede und Gemeinsamkeiten auf:

- Bei der direkten D/A-Wandlung (a) und beim Oversampling mit digitalem ZOH (d) wird das Durchlassband am stärksten gedämpft.
- Bei der D/A-Wandlung mit Oversampling ohne Interpolationsfilter (b) werden die Images am schlechtesten unterdrückt.
- Bei der D/A-Wandlung mit Oversampling und perfekter Interpolation (c) werden die Images am besten unterdrückt, nur hier tauchen die ersten Images erst bei  $f_{S2} - f_N$  auf. Die Dimensionierung des analogen Rekonstruktionsfilters ist bei (c) daher besonders einfach.
- Die D/A-Wandlung mit Überabtastung und Wiederholung des letzten Wertes (digitales ZOH) (d) lässt sich vorteilhaft einsetzen, wenn das Signal  $s[n]$  selbst schon oversampled ist ( $f_N \ll f_{S1}$ ): Man benötigt kein (bzw. ein schwächeres) digitales Interpolationsfilter wie bei (c), keine Amplitudenkorrektur und hat eine bessere Image-Unterdrückung als bei (a) und (b). Die Dämpfung des Nutzbandes fällt aufgrund des schmalen Nutzbands nicht so sehr ins Gewicht, hier ist  $\text{si}\pi f_N T_{S1} \approx 1$ .

### M9.3 Ideales Interpolationsfilter → A9.3

- Falsch:** Es filtert *nicht* die Wiederholspektren aus dem Basisband - im Basisband gibt es definitivonsgemäß keine Wiederholspektren. Da sich die Spektren zeitdiskreter Signale mit der Abtastfrequenz wiederholen (-> Wiederholspektren), ist es unmöglich Wiederholspektren zu beeinflussen (ohne gleichzeitig das Basisband zu ändern)! Nur Images, die bei einer vorangegangenen Erhöhung der Abtastfrequenz innerhalb des neuen Basisbands entstanden sind, werden von einem *iI* aus dem Basisband entfernt.
- Falsch:** Es filtert *nicht* die Wiederholspektren aus dem Nutzband, da das Nutzband ein Teil des Basisbands ist und ... (s.o.). Im Nutzband können aus diesem Grund auch keine Images auftreten, die ja Wiederholungen des alten Basisbands sind.
- Falsch:** Es hat *nicht zwangsläufig* möglichst weit auseinanderliegende Eckfrequenzen von Sperr- und Durchlassband - das sollte für einen entspannten Filterentwurf zwar so sein, ist aber keine Voraussetzung.
- Richtig:** Es entfernt alle Images - das ist die wichtigste Eigenschaft eines *iI*!
- Richtig:** Es lässt das Nutzband unverändert.
- Falsch:** Es *hat* Einfluss auf Wiederholspektren - es unterdrückt Images und damit auch deren Wiederholspektren!
- Falsch:** Es lässt Images *nicht* unverändert - im Gegenteil, Images sollen vollständig unterdrückt werden.

- h) **Falsch:** Es verhindert *nicht* Rückfaltungen ins Basisband; Rückfaltungen (Aliasing) treten nur beim Downsampling auf und müssen durch ein Anti-Aliasingfilter *vor* dem Downsampling verhindert werden.
- i) **Falsch:** Es verhindert *nicht* Rückfaltungen ins Nutzband (s.o.).
- j) **Richtig:** Oft wird im Interpolationsfilter gleichzeitig die Signalamplitude korrigiert, die durch Nullenstopfen beim Upsampling reduziert wurde.

## M9.4 Oversampling DAC → A9.4

- a) ZOH-Signal und Amplitudenspektrum ohne Oversampling (Abb. M9.3)

Das periodische Amplitudenspektrum  $|X_1(e^{j\omega T_{S1}})|$  des zeitdiskreten Signals hat Linien mit  $A = 2,5 \text{ V}$  bei  $f = n f_{S1} \pm f_{sig}$ . Die kritischen Frequenzen bei der Rekonstruktion des analogen Signals sind die höchste Frequenz des Nutzbandes (hier:  $f_{sig} = 3,3 \text{ kHz}$ ), deren Dämpfung durch den si-Frequenzgang der Zero-Order-Hold (ZOH)-Stufe an anderer Stelle ausgeglichen werden muss und die niedrigste Imagefrequenz (hier:  $f_{S1} - f_{sig} = 4,7 \text{ kHz}$ ), die unterdrückt werden muss (Tab. M9.1).

Allgemein ist der Frequenzgang eines ZOH-Gliedes gegeben durch

$$\frac{H_{ZOH}(f)}{H_{ZOH}(0)} = \text{si}(\pi f / f_S) = \frac{\sin \pi f / f_S}{\pi f / f_S},$$

die erste Nullstelle tritt also bei  $f = f_S$  auf.

Frequenz in kHz	3,3	4,7	11,3	12,7
Dämpfung	0,743	0,521	0,217	0,193
Dämpfung in dB	2,58	5,66	13,28	14,29

Tab. M9.1: si-Dämpfungen für DAC ohne Oversampling (Abb. 9.3a)

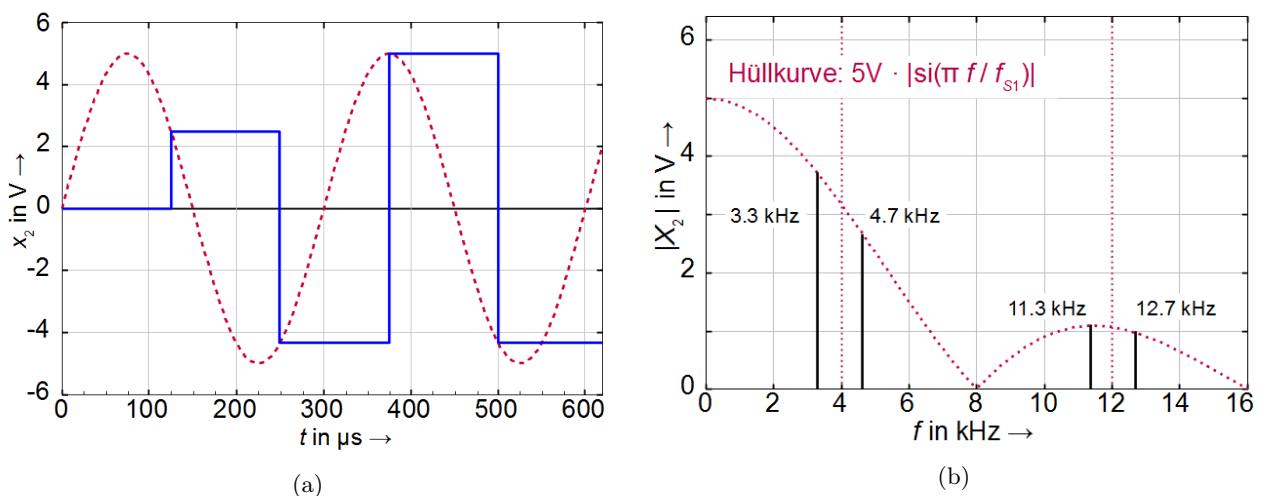


Abb. M9.3: ML: ZOH-Signal  $x_2(t)$  und Amplitudenspektrum  $|X_2(f)|$  zu Aufgabe 9.4a

b) Spezifikationen für Interpolations-Tiefpass aus Abb. 9.3b)

Der digitale Interpolations-Tiefpass soll Images bei 4,7 kHz bzw. 11,3 kHz unterdrücken, damit das analoge Rekonstruktionsfilter diese Aufgabe nicht übernehmen muss. Der digitale Interpolations-Tiefpass muss eine Sperrdämpfung haben von

$$H_{int}(e^{j\omega T_{S2}}) = 20 \log_{10} \frac{5 \text{ V}}{50 \text{ mV}} = 40 \text{ dB} \text{ bei } f_{S1} - f_{sig} = 4,7 \text{ kHz.}$$

Halbband-Filter haben symmetrische Koeffizienten und sind daher linearphasig. Da außerdem jeder zweite Koeffizient bis auf den mittleren Null ist, können sie mit vergleichsweise geringem Hardware-Aufwand realisiert werden.

Dafür haben Halbband-Filter enge Vorgaben für den Frequenzgang: Bei  $|H(f = f_S/4)| = 1/2$  und  $|H(f)| = 1 - |H(f_S/2 - f)|$  (jeweils normiert auf  $|H(f = 0)| = 1$ ). In diesem Fall folgt damit z.B.

$$|H(f = f_{SB})| = -40 \text{ dB} \hat{=} 0,01 \Rightarrow |H(f = f_{DB})| = 1 - |H(f = f_{SB})| = 0,99 \hat{=} 0,087 \text{ dB.}$$

Aus der geforderten Sperrdämpfung von 40 dB folgt damit ein Ripple von 1% oder 0,087 dB im Durchlassband.

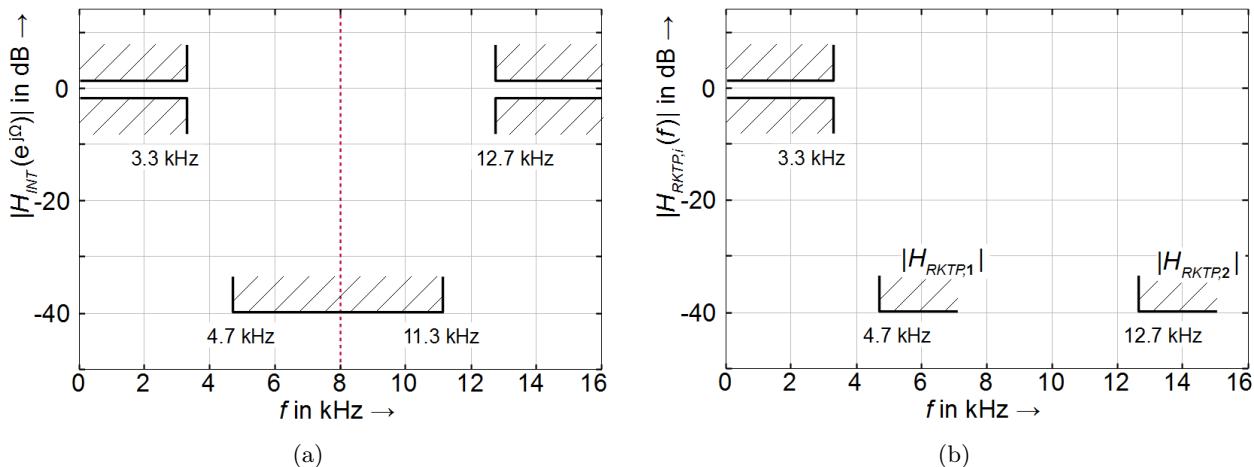


Abb. M9.4: ML: Spezifikationen für Interpolationsfilter  $H_{int}$  (a) und Rekonstruktionstiefpässe  $H_{RKTP,i}$  (b) zu Aufgabe 9.4b und d)

c) ZOH-DAC mit Oversampling: (Abb. M9.5)

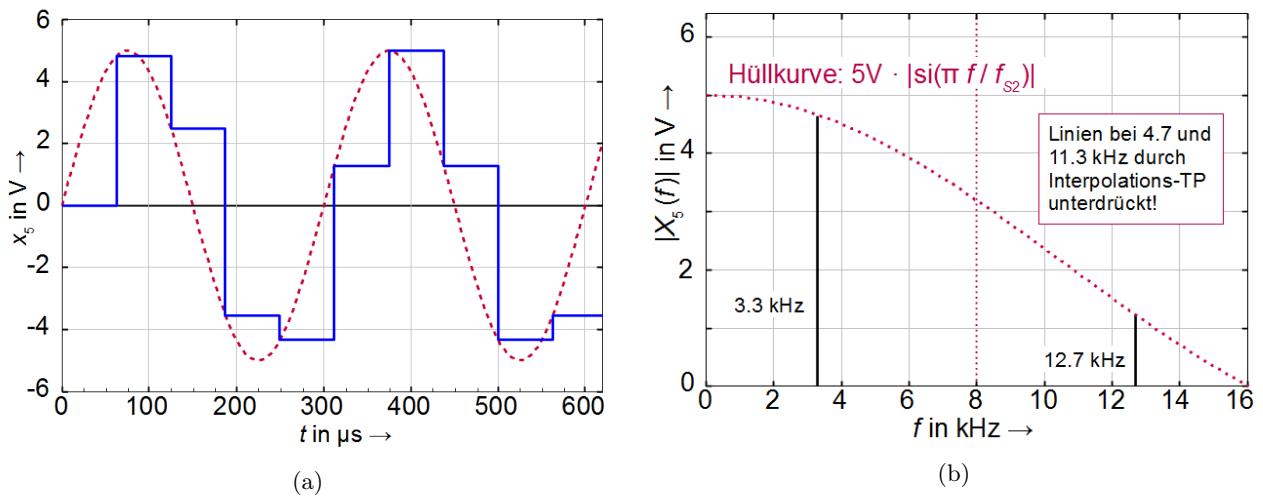
Durch die Erhöhung der Abtastfrequenz tritt das erste Image erst bei  $f_{S2} - f_{sig} = 12,7 \text{ kHz}$  auf, auch die Dämpfung der Signalfrequenz fällt geringer aus (Tab. M9.2).

Frequenz in kHz	3,3	12,7
Dämpfung	0,931	0,242
Dämpfung in dB	0,62	12,32

Tab. M9.2: si-Dämpfungen für DAC mit Oversampling (Abb. 9.3b)

d) Spezifikationen für analoge Rekonstruktions-Tiefpässe aus Abb. 9.3a) und b):

Wie man in Abb. M9.4b) sehen kann, ist für den DAC mit Oversampling ( $H_{RKTP,2}$ ) der Übergangsbereich zwischen Durchlass- und Sperrband breiter. Man kann also ein analoges Rekonstruktionsfilter mit geringerer Ordnung verwenden, um die geforderte Image-Unterdrückung zu erreichen.

Abb. M9.5: ML: ZOH-Signal  $x_5(t)$  und Amplitudenspektrum  $X_5(f)$  zu Aufgabe 9.4c

Die Images werden durch die ZOH-Funktion zusätzlich gedämpft (5,7 dB bzw. 12,3 dB), man kommt daher mit einer entsprechend geringeren Sperrdämpfung aus.

## M9.5 + Zweistufige Interpolation → A9.5

- Der „Amplitudengang“ der Kaskade entspricht genau dem der zweistufigen Dezimation aus Aufgabe 8.6 bis auf die hier erforderliche Amplitudenkorrektur um den Faktor 2 bzw. 3.
- Das **Spektrum des Ausgangssignals** besteht aus Wiederholungen des Eingangssignals, das im Basisband nur eine Komponente mit der Frequenz  $f_{sig,BB} = f_{S3}/4$  enthält. Diese Komponente ist im Ausgangsspektrum periodisch mit  $f_{S3}$  wiederholt vorhanden bei  $f_{sig,wied} = \pm f_{sig,BB} \pm k f_{S3}$  mit  $k = 0, 1, 2, \dots$

Bei Erhöhung der Abtastfrequenz auf zunächst  $f_{S,2} = 2f_{S,3}$  und dann auf  $f_{S,1} = 6f_{S,3}$  werden Wiederholspektren im Frequenzbereich  $\pm f_S/2$  „real“, d.h. zu Images, die von den Filtern gedämpft werden.

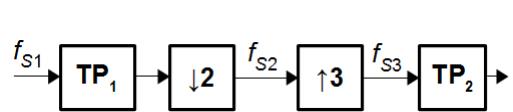
$f/f_{S3}$	$f/f_{S1}$	Dämpfung nach TP2 in dB	Gesamtdämpfung am Ausgang in dB
1/4	1/24	0	0
1/2	1/12	40	$40 + 40/3 = 53,3$
3/4	3/24	40	$40 + 40 \cdot 2/3 = 66,7$
5/4	5/24	40	80
7/4	7/24	0	40
9/4	9/24	0	40
11/4	11/24	40	80

Tab. M9.3: Dämpfung der Images im zweistufigen Interpolator (Aufgabe 9.5)



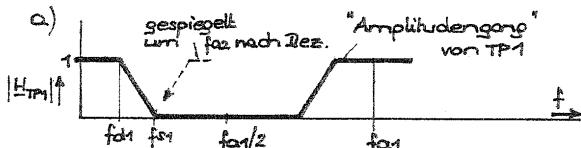
## M10 SRC: Abtastratenwandlung

### M10.1 Einfache Abtastratenwandlung → A10.1



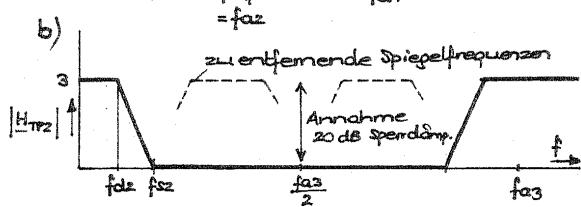
$F_{DB,1}$	0,15	$F_{DB,2}$	0,1
$F_{SB,1}$	0,25	$F_{SB,2}$	0,167

Abb. M10.1: Synchrone Abtastratenwandlung zu Aufgabe 10.1



damit kein Aliasing nach Dezimierung auf  $f_{az}$  auftritt, muß

$$f_{s1} = \frac{f_{az}}{2} = \frac{f_{az}}{4} \text{ sein } \Rightarrow \frac{f_{s1}}{f_{az}} = \frac{1}{4}$$

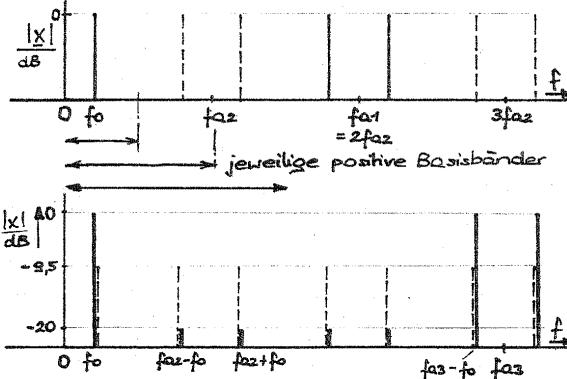


$f_{d1} = f_{d2}; f_{s2} = f_{s1}$   
für Toleranzschema beziehen auf  $f_{az}=3$ :

$$\frac{f_{d2}}{f_{s3}} = \frac{f_{d1}}{f_{s1}} \cdot \frac{f_{s1}}{f_{az}} \cdot \frac{f_{az}}{f_{s3}} = 0,15 \cdot 2 \cdot \frac{1}{3} = 0,1$$

$$\frac{f_{s2}}{f_{s3}} = \frac{f_{s1}}{f_{s1}} \cdot \frac{f_{s1}}{f_{az}} \cdot \frac{f_{az}}{f_{s3}} = 0,25 \cdot 2 \cdot \frac{1}{3} = 0,167$$

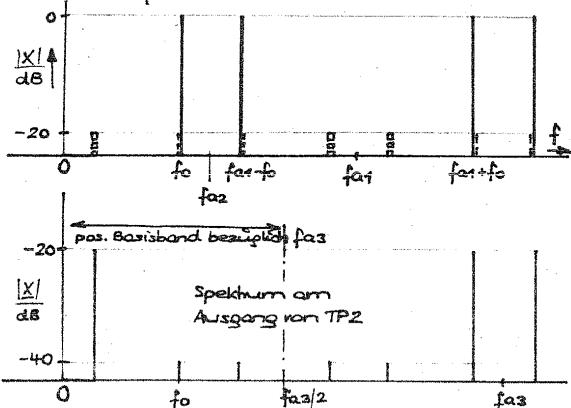
c) α) Anregung mit Sinus  $f_0 = f_{az}/10$  im Durchlaßbereich von TP 1



| Linien nach TP1 bei  $f_0 = 0,1 \cdot f_{az}$

| zusätzliche Linien nach Dezimierung mit  $R=2$  bei  $f_0 = 0,2 f_{az}$

β) Anregung mit Sinus  $f_0 = 0,4 f_{az}$  im Sperrbereich von TP1



| Linien nach  
Erhöhung der Abtastrate um  $I=3$   
durch Einfügen von Nullen; Amplituden  
sinken um Faktor 3  $\approx -9,5$  dB  
bei  $f_0, f_{az} \pm fo$

| Linien nach TP2 unter Annahme  
von 20 dB Sperrdämpfung und  
Verstärkungsfaktor 3 insgesamt  
 $\rightarrow$  unbedämpft bei  $f_0 = 0,1 \cdot \frac{2}{3} f_{az}$   
um 20 dB bedämpft bei  $f_0/faz = 0,266$  und  $0,1$

| Linien am Eingang (bei  $f_0$  im pos. Basisb.)

| Linien nach TP1 um 20 dB gedämpft

| zusätzliche Linien nach Dezimierung  
Alias bei  $f = f_{az} - fo = 0,2 f_{az}$ !

somit ist Spektrum vor  
Abtastraten erhöhung identisch  
wie unter α), jedoch um 20 dB bedämpft

es ergeben sich gleiche Spektren  
vor und nach TP2 wie unter α),  
jedoch um zusätzlich 20 dB bedämpft

## M10.2 Ideale Abtastratenwandlung → A10.2

### a) Verhältnis der Abtastraten

Bei Multiratensystemen ist die Abtastfrequenz der verschiedenen Teilsysteme eine wichtige Entwurfsgröße. Hier gilt:

$$f_{S1} = 100 \text{ kHz} \Rightarrow f_{S2} = \frac{f_{S1}}{5} = 20 \text{ kHz} \Rightarrow \frac{f_{S2}}{f_{S1}} = \frac{1}{5}$$

$$f_{S3} = 6f_{S2} = \frac{6f_{S1}}{5} = 120 \text{ kHz} \Rightarrow \frac{f_{S3}}{f_{S1}} = \frac{6}{5}$$

Damit ist die Abtastrate des Ausgangssignals um den Faktor 6/5 höher als die des Eingangssignals.

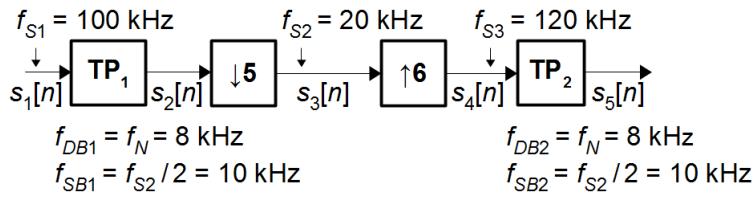


Abb. M10.2: Abtastratenwandelung zu Aufgabe 10.2 mit Abtast- und Eckfrequenzen

### b) Eckfrequenzen von $TP_1$

Die Dezimation entspricht einer erneuten Abtastung (Resampling) mit geringerer Abtastrate. Vor der Dezimation muss  $TP_1$  alle Frequenzkomponenten außerhalb des *neuen* Basisbands  $-f_{S2}/2 \dots + f_{S2}/2$  unterdrücken, da ansonsten *Aliasing* auftritt.  $TP_1$  ist also ein Anti-Aliasingfilter.

$$f_{SB,1} \leq \frac{f_{S2}}{2} = \frac{f_{S1}}{10} = 10 \text{ kHz} \Rightarrow F_{SB,1,max} = \frac{f_{SB,1}}{f_{S1}} = 0,1$$

Die Eckfrequenz des Durchlassbandes wird so gewählt, dass das Nutzsignal gerade noch durchgelassen wird. Höhere Eckfrequenzen für das DB würden zwar auch funktionieren, aber der Übergangsbereich zwischen DB  $f_{DB,1}$  und Sperrband  $f_{SB,1}$  würde schmäler, damit wäre ein Filter höherer Ordnung (= höherer Hardwareaufwand) nötig.

$$f_{DB,1} \geq f_N = 8 \text{ kHz} \Rightarrow F_{DB,1,min} = \frac{f_N}{f_{S1}} = 0,08$$

Abb. M10.3 zeigt den Frequenzgang  $|H_{TP1}(e^{j2\pi F})|$  von  $TP_1$  und das resultierende Spektrum  $|S_2(e^{j2\pi F})|$ .

### c) Eckfrequenzen von $TP_2$

Die Erhöhung der Abtastrate (Resampling) um den Faktor  $I$  wird erzielt durch Einfügen von  $I-1$  Nullen zwischen den Abtastwerten (*zero stuffing*). Dadurch entstehen  $I-1$  Kopien (*Images*) des alten Basisbandes  $-f_{S2}/2 \dots + f_{S2}/2$ . Nach der Erhöhung der Abtastrate von  $f_{S2}$  auf  $I f_{S2} = f_{S3}$  muss  $TP_2$  alle Frequenzkomponenten außerhalb des *alten* Basisbands  $-f_{S2}/2 \dots + f_{S2}/2$  unterdrücken, da ansonsten die *Images* das Signal verfälschen.  $TP_2$  wird daher Anti-Image-, Interpolations- oder Rekonstruktionsfilter genannt.

$$f_{SB,2} \leq \frac{f_{S2}}{2} = f_{SB,1} = 10 \text{ kHz} \Rightarrow F_{SB,2,max} = \frac{f_{SB,1}}{f_{S1}} = \frac{10 \text{ kHz}}{100 \text{ kHz}} = 0,1$$

$$F_{SB,2,max} = \frac{f_{SB,2}}{f_{S3}} = \frac{f_{SB,1}}{\frac{6}{5} f_{S1}} = \frac{10 \text{ kHz}}{120 \text{ kHz}} = 0,083$$

Normierte Frequenzen müssen immer auf die zugehörige Abtastfrequenz ( $f_{S1}$  bzw.  $f_{S3}$ ) bezogen werden!

Die Eckfrequenz des Durchlassbandes (DB)  $f_{DB,2}$  von  $TP_2$  wird so groß gewählt, dass gerade noch alle Frequenzkomponenten des Nutzbandes durchgelassen werden, sie hat daher den gleichen Wert wie  $f_{DB,1}$ :

$$f_{DB,2} = f_{DB,1} = 8 \text{ kHz} \Rightarrow F_{DB,1} = \frac{f_{DB,1}}{f_{S1}} = \frac{8 \text{ kHz}}{100 \text{ kHz}} = 0,08$$

$$F_{DB,2} = \frac{f_{DB,2}}{f_{S3}} = \frac{f_{DB,1}}{\frac{6}{5} f_{S1}} = \frac{8 \text{ kHz}}{120 \text{ kHz}} = 0,067$$

Die absoluten Eckfrequenzen von  $TP_1$  und  $TP_2$  sind identisch; aufgrund der unterschiedlichen Abtastfrequenzen ergeben sich aber anderen *relative* Grenzfrequenzen - und damit auch unterschiedliche Filterentwürfe für beide Filter!

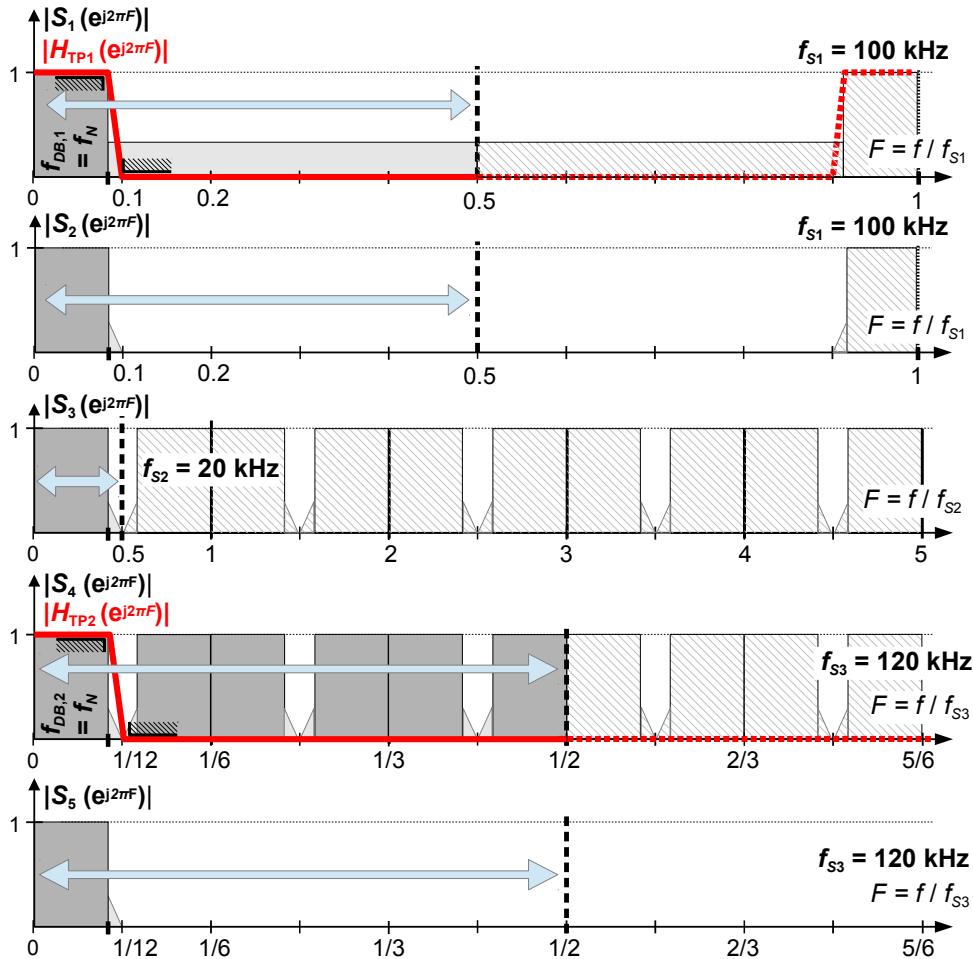


Abb. M10.3: Spektren und Frequenzgänge (Pfeile kennzeichnen das pos. Basisband) zu Aufgabe 10.2

#### d) Blockschaltbild von „vertauschter“ Kaskade

Bei Vertauschung von Dezimator und Interpolator müssen auch Dezimations- und Interpolationsfilter ihren Platz tauschen (Abb. M10.4)! Da hier zunächst die Abtastrate erhöht wird, wird

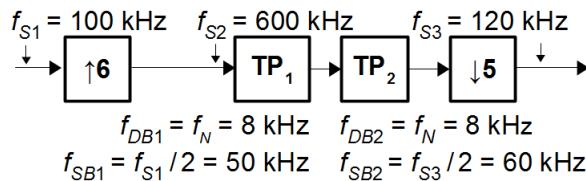


Abb. M10.4: Alternative Abtastratenwandlung zu Aufgabe 10.2 mit vertauschter Dezimation und Interpolation

TP<sub>1</sub> jetzt benötigt um die Images oberhalb  $f_{S1}/2$  zu entfernen (Interpolations- oder Rekonstruktionsfilter). Vor der Dezimation wird ein weiteres Filter benötigt, um Frequenzanteile oberhalb  $f_{S3}/2$  zu entfernen (Anti-Aliasingfilter). Mit der Vertauschung von Dezimation und Interpolation ändern sich auch die Spezifikationen für die Eckfrequenzen des Sperrbereichs (Abb. M10.4). Die absoluten Eckfrequenzen für die Durchlassbereiche bleiben gleich; die höheren Abtastfrequenzen

von  $\text{TP}_1$  und  $\text{TP}_2$  führen aber im Vergleich zu Abb. M10.2 geringeren *relativen Grenzfrequenzen*.

$$\begin{aligned} f_{DB,1} = f_{DB,2} = f_N &= 8 \text{ kHz} \Rightarrow F_{DB,1} = F_{DB,2} = f_N/f_{S2} = 8/600 = 1,33 \cdot 10^{-2} \\ f_{SB,1} = f_{S1}/2 &= 50 \text{ kHz} \Rightarrow F_{SB,1} = f_{SB,1}/f_{S2} &= 8,33 \cdot 10^{-2} \\ f_{SB,2} = f_{S3}/2 &= 60 \text{ kHz} \Rightarrow F_{SB,2} = f_{SB,2}/f_{S2} &= 0,1 \end{aligned}$$

Beide Tiefpassfilter werden normalerweise in einem Filter zusammengefasst (Abb. M10.5) um Hardware einzusparen. Dieses Filter muss die jeweils schärfere Anforderung für Durchlass- und Sperrbereich erfüllen.

$$\begin{aligned} f_{DB} &= \max(f_{DB,1}, f_{DB,2}) = 8 \text{ kHz} \Rightarrow F_{DB} = f_{DB}/f_{S2} = 1,33 \cdot 10^{-2} \\ f_{SB} &= \min(f_{SB,1}, f_{SB,2}) = 50 \text{ kHz} \Rightarrow F_{SB} = f_{SB}/f_{S2} = 8,33 \cdot 10^{-2} \end{aligned}$$

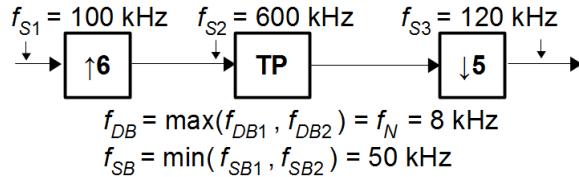


Abb. M10.5: Alternative Abtastratenwandelung zu Aufgabe 10.2 mit zusammengefasstem TP-Filter

### e) *SQNR* und *ENOB*

Die gesamte *Rauschleistung*  $N_Q$ , die bei einer (Re-)Quantisierung entsteht, hängt nur von der Größe des LSBs ab, nicht von der Abtastrate  $f_S$ , genauso wenig wie die Leistung eines Sinustons von dessen Frequenz abhängt. Es ist daher  $N_Q = q^2/12$  mit  $q = FSR/2^W$  nur vom Aussteuerbereich (Full Scale Range, *FSR*) und von der Anzahl der Quantisierungsstufen  $2^W$  abhängig. Das maximale *SQNR* (= bei maximaler Aussteuerung) ist unabhängig vom *FSR*, da hier die maximale Signalleistung und damit ebenfalls der Aussteuerbereich eingeht (siehe auch Aufgabe 7.1).

Die *Rauschleistungsdichte*  $N'_Q = N_Q/f_S$ , die bei der (Re-)Quantisierung entsteht, hängt von der Bandbreite des Basisbands  $f_S$  ab: Nach dem Parcevalschen Theorem sind die Signalleistung im Zeit- und Frequenzbereich identisch, tastet man das gleiche Signal mit höherer Abtastrate  $f_{S2}$  ab, verteilt sich daher die Rauschleistung  $N_Q$  über die größere Rauschbandbreite  $B_{N2} = f_{S2}$ , es gilt  $N_Q = N'_{Q1}f_{S1} = N'_{Q2}f_{S2}$ . Da es sich um abgetastete (zeitdiskrete!) Signale mit periodischen Spektren handelt, ist  $B_N$  natürlich höchstens gleich  $f_S$ .

Bei *Reduktion der Abtastrate* um den Faktor  $R$  ändern sich die Spektralamplituden nicht, allerdings werden Komponenten außerhalb des *neuen* Basisbands  $-f_S/2R \dots + f_S/2R$  in dieses zurückgefaltet und verursachen so Verzerrungen und eine Verschlechterung des *SQNR* (Aliasing), die nicht wieder korrigiert werden können! Hier ist also eine Begrenzung der Bandbreite auf  $B = f_S/2R$  vor der Reduktion der Abtastrate erforderlich (Anti-Aliasfilter).

Bei *Erhöhung der Abtastrate* um den Faktor  $I$  auf  $If_S$  durch Nullenstopfen werden die Wiederholspektren des Nutzsignals und die des Quantisierungsrauschen real bis  $If_S/2$ . Die Amplituden beider Spektralanteile werden um  $1/I$  reduziert<sup>1</sup>, das *SQNR* im alten Basisband  $-f_S/2 \dots + f_S/2$  ändert sich daher nicht. Nach der Erhöhung der Abtastrate muss die Bandbreite des Signals daher auf das alte Basisband begrenzt werden, da die Spektralkomponenten zwischen  $f_S/2 \dots If_S/2$  Verzerrungen verursachen und das *SQNR* verschlechtern.

<sup>1</sup>Diese Amplitudenreduktion wird üblicherweise im nachfolgenden Interpolationsfilter ausgeglichen und daher oft gar nicht erwähnt.

Eine *nachträgliche Erhöhung der Abtastrate* verbessert also nicht das *SQNR*, bei ungenügender Filterung kann es sogar deutlich schlechter werden! Das funktioniert nur bei gleichzeitigem (Re-)Sampling und (Re-)Quantisierung, z.B. bei der Abtastung analoger Signale, da hier das neu entstehende Quantisierungsrauschen umso „dünner“ verteilt wird, je größer die neue Abtastrate ist.

Reduziert man aber die äquivalente Rauschbandbreite  $B_N$  durch Tiefpassfilterung ohne dabei das Nutzsignal zu beeinträchtigen (das geht nur bei Überabtastung, also wenn für die Bandbreite des Signals gilt  $B < f_S/2!$ ), reduziert man auch die gesamte Rauschleistung, also  $N_2 = N_1 B_{N,2}/B_{N,1}$ . Das *SQNR* verbessert sich im gleichen Verhältnis, da  $S$  unverändert ist.

Zur Berechnung des *SQNR* an verschiedenen Stellen der Signalverarbeitungskette muss man also das *SQNR* des Wandlers am Eingang  $s_1[n]$  des Multiratensystems bestimmen und es dann mit dem Verhältnis der Rauschbandbreiten  $B_{N,i}$  an verschiedenen Stellen des Systems skalieren<sup>2</sup>:

$$SQNR_1 = (6,02W + 1,76) \text{ dB} = 49,92 \text{ dB} \text{ für } W = 8.$$

Die Rauschbandbreite von  $s_1$  beträgt  $2B_{N1} = f_{S1} = 100 \text{ kHz}$ . Daraus ergibt sich z.B.

$$SQNR_2 = SQNR_1 + 10 \log \frac{100 \text{ kHz}}{20 \text{ kHz}} = (49,92 + 6,99) \text{ dB} = 56,91 \text{ dB}.$$

Signal	$f_S$	$2B_N$	$SQNR / \text{dB}$	$ENOB$
$s_1[n]$	100 kHz	100 kHz	49,92	8,0
$s_2[n]$	100 kHz	20 kHz	56,91	9,2
$s_3[n]$	20 kHz	20 kHz	56,91	9,2
$s_4[n]$	120 kHz	—	— <sup>3</sup>	—
$s_5[n]$	120 kHz	20 kHz	56,91	9,2

Tab. M10.1: Werte zu Abb. [M10.2](#)

Signal	$f_S$	$2B_N$	$SQNR / \text{dB}$	$ENOB$
$s_1[n]$	100 kHz	100 kHz	49,92	8,0
$s_2[n]$	600 kHz	—	(s.o.)	—
$s_3[n]$	600 kHz	100 kHz	49,92	8,0
$s_4[n]$	600 kHz	100 kHz	49,92	8,0
$s_5[n]$	120 kHz	100 kHz	49,92	8,0

Tab. M10.2: Werte zu Abb. [M10.4](#) bzw. [M10.5](#)

<sup>2</sup>Da nur die *Verhältnisse* der Rauschbandbreiten betrachtet werden, ist es gleich ob man die Rauschleistungsdichte über  $-f_S/2 \dots f_S/2$  oder über  $0 \dots f_S/2$  bestimmt.

<sup>3</sup>Vor einer *SQNR*-Betrachtung müssen zunächst die Images entfernt werden.

### f) Vergleich der Systeme

Der größte Nachteil von **System Abb. M10.2** liegt darin, dass die Signalbandbreite  $B$  des Gesamtsystems beschränkt wird durch die niedrige Nyquistfrequenz  $f_{S2}/2 = f_{S1}/2R$  nach der Dezimation. Damit ist diese Form der Abtastratenwandelung im Allgemeinen nur für Oversampling-Systeme geeignet, bei denen die Nutzbandbreite viel geringer als die Abtastrate  $f_{S1}$  ist. Durch die Verringerung der Nutzbandbreite verbessert sich als positiver Nebeneffekt das  $SQNR$ .

Die Abschätzung der Filterordnung liefert folgende Ergebnisse:

$$N_{FIR,1} \approx \left\lceil \frac{-20 \log_{10} \sqrt{\delta_{SB} \delta_{DB}} - 13}{14,6(F_{SB,1} - F_{DB,1})} \right\rceil = \left\lceil \frac{50 - 13}{14,6 \cdot 0,1 - 0,08} \right\rceil = 127 \quad (\text{Matlab: } 127)$$

$$N_{FIR,2} \approx \left\lceil \frac{-20 \log_{10} \sqrt{\delta_{SB} \delta_{DB}} - 13}{14,6(F_{SB,2} - F_{DB,2})} \right\rceil = \left\lceil \frac{50 - 13}{14,6 \cdot (0,083 - 0,067)} \right\rceil = 153 \quad (\text{Matlab: } 153)$$

Der Vergleich mit der einfachen Abschätzung (5.4) zeigt deren größeren Fehler:

$$N_{FIR,1} \approx \left\lceil \frac{3f_{S1}}{f_{SB,1} - f_{DB,1}} \right\rceil = \left\lceil \frac{3}{F_{SB,1} - F_{DB,1}} \right\rceil = \left\lceil \frac{300 \text{ kHz}}{(10 - 8) \text{ kHz}} \right\rceil = 150$$

$$N_{FIR,2} \approx \left\lceil \frac{3f_{S3}}{f_{SB,2} - f_{DB,2}} \right\rceil = \left\lceil \frac{3}{F_{SB,1} - F_{DB,1}} \right\rceil = \left\lceil \frac{360 \text{ kHz}}{(10 - 8) \text{ kHz}} \right\rceil = 180$$

Der größte Nachteil von System **Abb. M10.4** bzw. **Abb. M10.5** liegt in der hohen „Zwischenfrequenz“  $f_{S2} = If_{S1}$ . Da die Nutzbandbreite des Signals im ganzen System gleich bleibt, bleibt auch der  $SQNR$  unverändert. Es wird nur ein Tiefpassfilter mit vergleichsweise geringen Anforderungen benötigt, das gleichzeitig als Dezimations- und Interpolationsfilter dient:

$$N_{FIR} = N_{FIR,1} \approx \left\lceil \frac{50 - 13}{14,6 \cdot (8,33 \cdot 10^{-2} - 1,33 \cdot 10^{-2})} \right\rceil = 37 \quad (\text{Matlab: } 36)$$

$$N_{FIR,2} \approx \left\lceil \frac{50 - 13}{14,6 \cdot (0,1 - 1,33 \cdot 10^{-2})} \right\rceil = 30 \quad (\text{Matlab: } 29)$$

System	$B_{max}$	$f_{S,max}$	$N_{FIR,1}$		$N_{FIR,2}$	
			≈	sim	≈	sim
a)	$f_{S1}/2R = 10 \text{ kHz}$	120 kHz	127	127	153	153
d), 2 TP	$\min(f_{S1}/2, f_{S3}/2) = 50 \text{ kHz}$	$If_{S1} = 600 \text{ kHz}$	37	36	30	29
d), 1 TP	$\min(f_{S1}/2, f_{S3}/2) = 50 \text{ kHz}$	$If_{S1} = 600 \text{ kHz}$	37	36	—	—

Tab. M10.3: Ergebnisse zu Aufgabe 10.2

- g) Die **Spezifikationen der analogen Filter** sind für beide Systeme gleich, da jeweils am Eingang ein ADC mit  $f_{S1} = 100 \text{ kHz}$  und am Ausgang ein DAC mit  $f_{S3} = 120 \text{ kHz}$  sitzt.

Die Eckfrequenz des analogen AA-Filters ist aufgrund des Oversamplings unkritisch, es genügt  $f_{SB,AA} = f_{S1} - B = 92 \text{ kHz}$ , da das abgetastete Signal bereits in der folgenden Stufe mit einem digitale Tiefpass nachbearbeitet wird. Es muss lediglich verhindern, dass sich hochfrequente Störungen ins Basisband zurückfalten.

Für das analoge RK-Filter gilt entsprechend  $f_{SB,IP} = f_{S3} - B = 112 \text{ kHz}$ .

### M10.3 Abtastratenwandlung mit Aliasing im Übergangsbereich → A10.3

#### a) Eckfrequenzen $TP_1$

Die Eckfrequenzen des Durchlassbereichs für  $TP_1$  und  $TP_2$  bleiben unverändert. Die Sperrfrequenz von  $TP_1$  darf jetzt so gewählt werden, dass bei Abtastung mit  $f_{S2}$  Alias-Komponenten mit der niedrigsten Frequenz bei  $f = f_N = f_{DB,1}$  entstehen:

$$f_{DB,1} = f_N = 8 \text{ kHz} \quad \Rightarrow \quad F_{DB,1} = \frac{f_N}{f_{S1}} = \frac{8 \text{ kHz}}{100 \text{ kHz}} = 0,08$$

$$f_{SB,1} = f_{S2} - f_N = 20 \text{ kHz} - 8 \text{ kHz} = 12 \text{ kHz} \quad \Rightarrow \quad F_{SB,1} = \frac{f_{SB,1}}{f_{S1}} = \frac{12 \text{ kHz}}{100 \text{ kHz}} = 0,12$$

#### b) Eckfrequenzen $TP_2$

Die Spezifikationen für  $TP_2$  ändern sich hier nicht,  $TP_2$  soll ja nur die Images außerhalb des Basisbands  $\pm f_{S2}/2$  unterdrücken.

#### c) Filterordnung

$$N_{FIR,1} \approx \left\lceil \frac{50 - 13}{14,6 \cdot (0,12 - 0,08)} \right\rceil = 64 \quad (\text{Matlab: } 64)$$

Durch das Zulassen von Aliasing ins Übergangsband kann hier die Ordnung von  $TP_1$  halbiert werden!

### M10.4 Abtastratenwandlung HiRes-Audio → A10.4

- a) Für die **Reduktion der Abtastrate** von  $f_{S1} = 192 \text{ kHz}$  auf  $f_{S2} = 48 \text{ kHz}$  ist keine Filterung erforderlich, da der Audiostream bereits auf 20 kHz bandbreitenbegrenzt ist (Abb. M10.6a). Requantisieren von 24 auf 16 bit kann man wahlweise durch Wertschneiden oder Runden (falls der resultierende minimale systematische DC-Fehler von  $q/2$  stört.)
- b) Nach der **Erhöhung der Abtastrate** von  $f_{S2} = 48 \text{ kHz}$  auf  $f_{S1} = 192 \text{ kHz}$  müssen Images entfernt werden, es ist also ein Filter mit den Eckfrequenzen  $f_{DB} = 20 \text{ kHz}$  und  $f_{SB} = f_{S2} - B = 28 \text{ kHz}$  erforderlich (Abb. M10.6b). Die Filterung erhöht die Wortlänge (Multiplikation und Addition), anschließend requantisiert man auf die gewünschte Wortlänge von 24 bit.

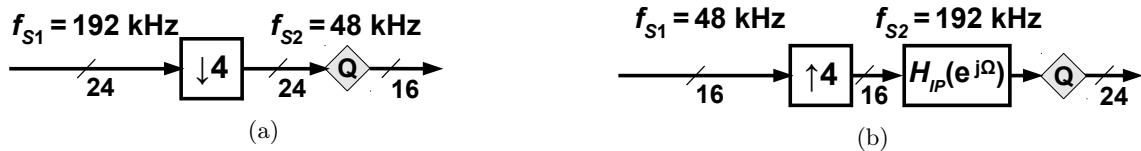


Abb. M10.6: Systeme zu Aufgabe 10.4a) und b)

### M10.5 Abtastratenwandlung CD Audio → A10.5

Wandlung eines digitalen Audiostreams vom CD-Audioformat (16 Bit Auflösung, Abtastrate  $f_{S1} = 44,1 \text{ kHz}$ ) auf  $f_{S3} = 48 \text{ kHz}$ :

- a) Die **kleinsten Up- bzw. Downsamplingfaktoren** für eine synchrone Abtastratenwandlung erhält man aus der Zwischenfrequenz  $f_{S2}$ , die ein gemeinsames Vielfaches von  $f_{S1}$  und  $f_{S3}$  sein muss;  $f_{S2} = I f_{S1} = R f_{S3}$ , also

$$f_{S2} = f_{S1} f_{S3} / \text{ggT}(f_{S1}, f_{S3}) = 44100 \cdot 48000 / (3 \cdot 100) = 2,352 \text{ MHz}$$

$$\Rightarrow I = \frac{f_{S2}}{f_{S1}} = \frac{f_{S3}}{\text{ggT}(f_{S1}, f_{S3})} = 160 \quad \text{und} \quad R = \frac{f_{S2}}{f_{S3}} = \frac{f_{S1}}{\text{ggT}(f_{S1}, f_{S3})} = 147$$

- b) **Filterspezifikationen:** Der digitale Interpolationsfilter muss Frequenzkomponenten oberhalb der alten Nyquistfrequenz  $f_{S1}/2 = 22,05 \text{ kHz}$  ausfiltern, der Anti-Aliasfilter Komponenten oberhalb der neuen Nyquistfrequenz,  $f_{S3}/2 = 24 \text{ kHz}$ . Die Interpolation stellt hier also die härteren Anforderungen. Die Eckfrequenz des Sperrbands muss die niedrigste Imagefrequenz unterdrücken und liegt bei  $f_{SB} = f_{S1} - B = 24,1 \text{ kHz}$ .

Das Durchlassband muss wie immer breit genug für das Nutzsignal sein,  $f_{DB} = B = 20 \text{ kHz}$ .

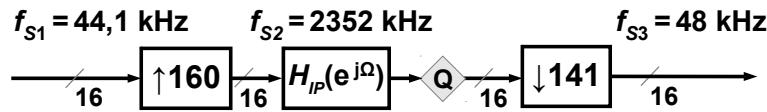


Abb. M10.7: System zu Aufgabe 10.5a) und b)



## **Teil III**

### **Anhang**



## A Signalflussgraphen (SFG)

Die Analyse komplizierter *linearer*<sup>1</sup> Systeme wird durch *Signalflussgraphen* (SFG) vereinfacht, die die Topologie des unabhängigen linearen Gleichungssystems visualisieren. Die Variablen des Gleichungssystems werden durch die Knoten des Graphen repräsentiert, die verbindenden Zweige (auch Kanten genannt) definieren deren Zusammenhang (Abb. A.1). Die Visualisierung erleichtert die Manipulation und Auflösung des Gleichungssystems. SFGs wurden in den 1950ern zur Analyse linearer Netzwerke eingeführt. Seitdem werden SFGs zur Lösung der unterschiedlichsten ingenieurwissenschaftlichen Probleme erfolgreich eingesetzt, wie z.B. zur Errechnung der Ausfallwahrscheinlichkeiten im NASA Apollo Projekt oder für die symbolische Analyse von Analogschaltungen. In diesem Kurs werden SFGs für die Analyse von zeitdiskreten Systemen in der  $z$ -Ebene verwendet. Weitere Unterlagen zu SFGs finden Sie unter [https://en.wikipedia.org/wiki/Signal-flow\\_graph](https://en.wikipedia.org/wiki/Signal-flow_graph) und <https://de.wikipedia.org/wiki/Signalflussplan>.

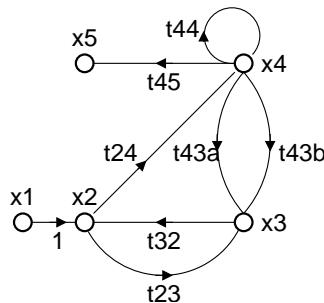


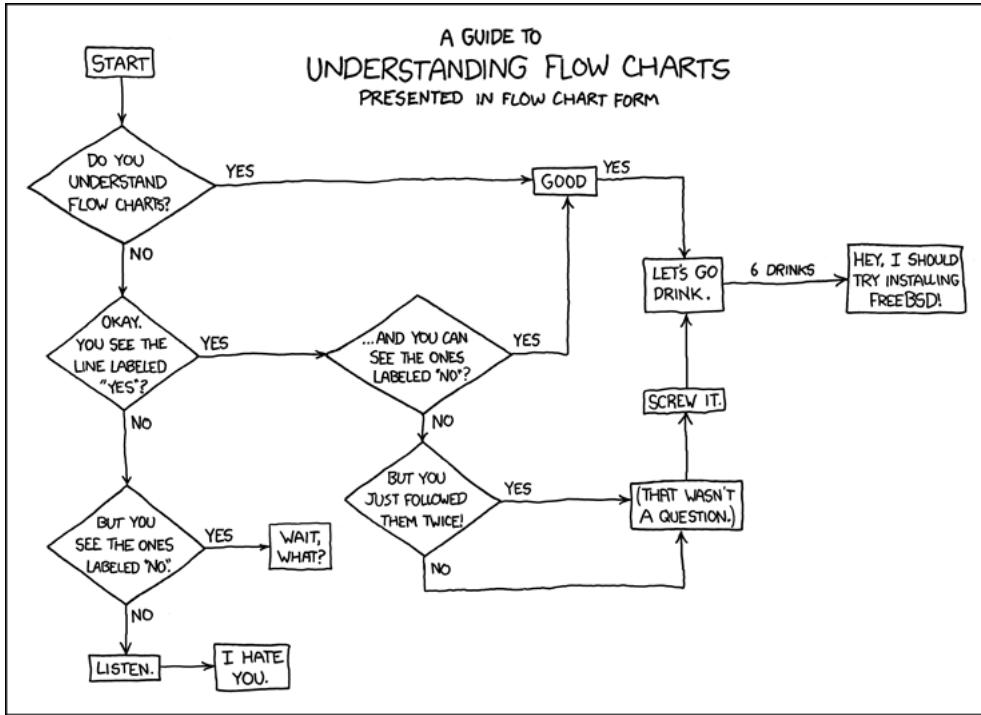
Abb. A.1: Beispiel für einen Signalflussgraphen

Ein SFG ist ein **gerichteter Graph**, d.h. eine Menge von **Knoten** und **gerichteten Zweigen**. Jeder Zweig beginnt und endet auf einem Knoten  $x_i$  (das kann auch der gleiche sein) und hat einen Verstärkungsfaktor (Gewicht), die Zweigverstärkung oder **Zweig-Transmittanz**  $t_{ij}$ . Eine **Schleife** ist eine Folge von Zweigen bei der jeder Knoten zu genau zwei Zweigen der Schleife gehört, von denen einer auf dem Knoten beginnt und einer endet. In einer Schleife erster Ordnung kann jeder Knoten von jedem anderen Knoten erreicht werden. Ähnlich einer Schleife erster Ordnung ist ein **Vorwärtspfad** eine Abfolge von Zweigen zwischen einem Start- und einem Endknoten. Dabei ist jeder Knoten mit genau zwei Zweigen des Pfads verbunden, außer dem Startknoten, der keinen Eingangszweig hat und dem Endknoten, der keinen Ausgangszweig hat. Eine Schleife der Ordnung  $m$  ist eine Menge von  $m$  Schleifen erster Ordnung, die sich nicht berühren, d.h. Schleifen ohne gemeinsame Knoten. Die **Schleifenverstärkung**  $L$  einer Schleife erster Ordnung bzw. die **Pfadverstärkung**  $P$  eines Pfades ist definiert als das Produkt aller Zweigverstärkungen entlang der Schleife bzw. des Pfads. Die Schleifenverstärkung einer Schleife höherer Ordnung ist das Produkt der Schleifenverstärkungen aller Schleifen erster Ordnung. Drei einfache Regeln definieren das Verhalten von SFGs:

**Richtung:** Ein Signal fließt entlang eines Zweigs in die Richtung, die durch den Pfeil angezeigt wird und wird mit der Transmittanz des Zweigs multipliziert, z.B.  $x_5 = t_{45}x_4$  in Abb. A.1.

**Senke:** Ein Knotensignal ist die Summe aller am Knoten ankommenden Signale, z.B.  $x_3 = t_{23}x_2 + t_{43a}x_4 + t_{43b}x_4$  in Abb. A.1.

<sup>1</sup>Ohne diese wichtige Voraussetzung sind die im Folgenden geschilderten Methoden zur Umformung von Netzwerken nicht gültig!

Abb. A.2: Flowchart als Flowchart [<http://xkcd.com/518/>]

**Quelle:** Das Knotensignal wird an jeden abgehenden Zweig angelegt.

**Anmerkung:** Verwechseln Sie nicht ein SFG mit einem *Zustandsdiagramm* oder *Automatengraphen* - auch wenn die Form und z.T. die Bezeichnungen die gleichen sind: Bei letzteren entsprechen die Knoten Zuständen und die Zweige Booleschen Bedingungen für den Übergang von einem Zustand in einen anderen!

## A.1 Transformationen von SFGs

Von diesen drei Regeln können einige nützliche Regeln zur Umformung von SFGs abgeleitet werden (Abb. A.3a - A.5b). Mit deren Hilfe können Zweige eines SFGs zusammengefasst und Knoten eines SFGs der Reihe nach eliminiert werden, um zum Schluss die Übertragungsfunktion zwischen zwei Knoten zu erhalten.

### A.1.1 Zusammenfassen von seriellen bzw. parallelen Zweigen

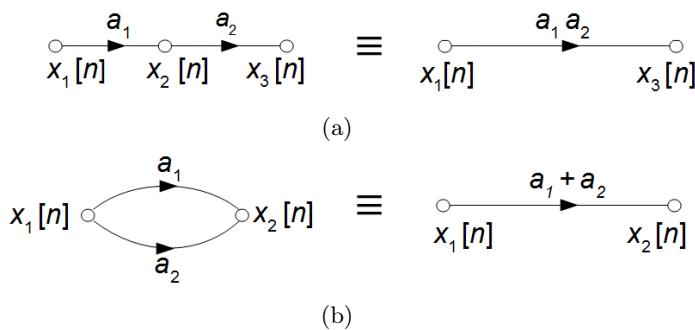


Abb. A.3: Zusammenfassen von seriellen (a) und parallelen (b) Zweigen

Die Regel in Abb. A.3a erhält man über  $x_3 = a_2 x_2 = a_2(a_1 x_1) = a_1 a_2 x_1$ .

In einem Knoten werden alle einlaufenden Zweige aufsummiert, damit ergibt sich Abb. A.3b,  $x_2 = a_1x_1 + a_2x_1 = (a_1 + a_2)x_1$  ganz automatisch.

### A.1.2 Distributivität („Schieben über Knoten“)

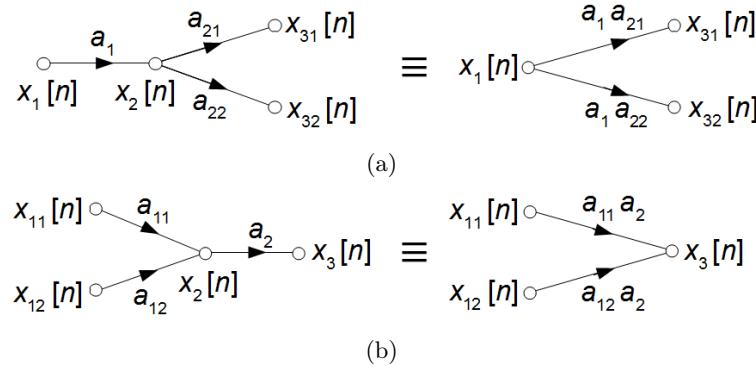


Abb. A.4: Distributivität

Die Regeln zur Distributivität (= Reihenfolge von Multiplikation und Addition darf vertauscht werden) in Abb. A.4 ergeben sich aus der Linearität des Systems (die ja vorausgesetzt worden war). Diese Regeln sind sehr praktisch, um einen Verstärkungsfaktor über einen Knoten zu „verschieben“. Dabei darf ein Register als Multiplikation mit  $z^{-1}$  betrachtet werden!

### A.1.3 Transponieren

Man *transponiert* einen Graph, indem man nacheinander die folgenden Operation durchführt:

- Umdrehen sämtlicher Signalrichtungen.
- Ersetzen von Verzweigungen durch Addierer.
- Ersetzen von Addierern durch Verzweigungen. Bei Subtrahierern muss das negative Vorzeichen ersetzt werden, entweder durch Umkehr des Vorzeichens eines Koeffizientenmultipliziers im Pfad oder durch Umwandeln eines Addierers am Ende des Pfads in einen Subtrahierer (jeweils falls vorhanden).
- Vertauschen von Eingangs- und Ausgangsknoten.

### Schleifen

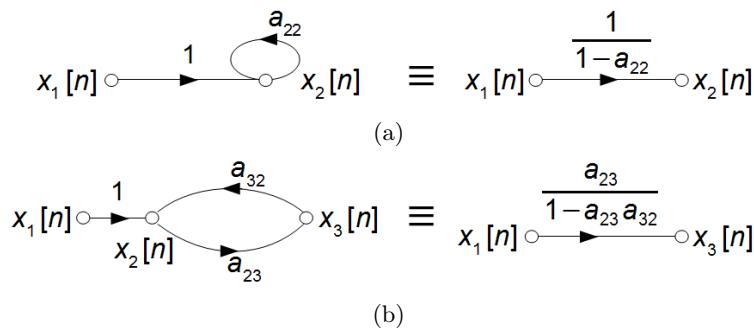


Abb. A.5: Eliminieren von Schleifen, (a) Eigenschleifen und (b) allgemeine Schleifen

Die Regeln in Abb. A.5 zum Eliminieren von Schleifen ergeben sich durch Aufstellen und Auflösen der Knotengleichungen (bekannt aus der Regelungstechnik).



## B Wichtige Formeln und Zusammenhänge

### B.1 Komplexe Ebene

$$r = |z| = \sqrt{x^2 + y^2}; \quad x = \Re\{z\} \text{ und } y = \Im\{z\}$$

$$z = x + jy = re^{j\phi} \text{ mit } \phi = \text{atan2}(z) = \begin{cases} \arctan \frac{y}{x} & \text{für } x > 0 \quad \bullet \circ \\ \arctan \frac{y}{x} + \pi & \text{für } x < 0, y \geq 0 \quad \blacksquare \\ \arctan \frac{y}{x} - \pi & \text{für } x < 0, y < 0 \quad \blacksquare \\ \frac{\pi}{2} & \text{für } x = 0, y > 0 \quad \blacktriangle \\ -\frac{\pi}{2} & \text{für } x = 0, y < 0 \quad \blacktriangledown \end{cases} \quad (\text{B.1})$$

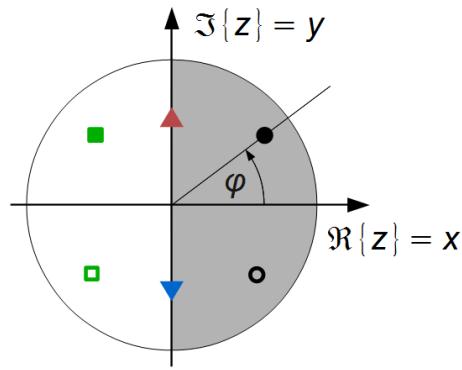


Abb. B.1: Komplexe  $z$ -Ebene

### B.2 Trigonometrische und Eulersche Identitäten

$$e^z = \cos z + j \sin z \quad (\text{B.2})$$

$$\sin(z) = \frac{e^{jz} - e^{-jz}}{2j} \quad (\text{B.3})$$

$$\cos(z) = \frac{e^{jz} + e^{-jz}}{2} \quad (\text{B.4})$$

$$\tan(z) = \frac{\sin(z)}{\cos(z)} = -j \frac{e^{jz} - e^{-jz}}{e^{jz} + e^{-jz}} = -j \frac{1 - e^{-2jz}}{1 + e^{-2jz}} \quad (\text{B.5})$$

### B.2.1 Trigonometrische Identitäten

#### Sinus

$$\sin \alpha = \cos(\alpha - \frac{\pi}{2}) \quad (\text{B.6})$$

$$\sin \alpha \cdot \sin \beta = \frac{1}{2} [\cos(\alpha - \beta) - \cos(\alpha + \beta)] \quad (\text{B.7})$$

$$\sin^2 \alpha = \frac{1}{2} (1 - \cos 2\alpha) \quad (\text{B.8})$$

$$\sin^3 \alpha = \frac{1}{4} (3 \sin \alpha - \sin 3\alpha) \quad (\text{B.9})$$

$$\sin^n \alpha = \quad (\text{B.10})$$

$$\sin(\alpha \pm \beta) = \sin \alpha \cos \beta \mp \cos \alpha \sin \beta \quad (\text{B.11})$$

$$\sin \alpha + \sin \beta = 2 \sin \frac{\alpha + \beta}{2} \cdot \cos \frac{\alpha - \beta}{2} \quad (\text{B.12})$$

$$\sin \alpha - \sin \beta = 2 \cos \frac{\alpha + \beta}{2} \cdot \sin \frac{\alpha - \beta}{2} \quad (\text{B.13})$$

#### Cosinus

$$\cos(\alpha) = \sin(\alpha + \frac{\pi}{2}) \quad (\text{B.14})$$

$$\cos \alpha \cdot \cos \beta = \frac{1}{2} [\cos(\alpha - \beta) + \cos(\alpha + \beta)] \quad (\text{B.15})$$

$$\cos^2 \alpha = \frac{1}{2} (1 + \cos 2\alpha) \quad (\text{B.16})$$

$$\cos^3 \alpha = \frac{1}{4} (3 \cos \alpha \cos 3\alpha) \quad (\text{B.17})$$

$$\cos^n \alpha = \quad (\text{B.18})$$

$$\cos(\alpha \pm \beta) = \cos \alpha \cos \beta \mp \sin \alpha \sin \beta \quad (\text{B.19})$$

$$\cos \alpha + \cos \beta = 2 \cos \frac{\alpha + \beta}{2} \cdot \cos \frac{\alpha - \beta}{2} \quad (\text{B.20})$$

$$\cos \alpha - \cos \beta = -2 \sin \frac{\alpha + \beta}{2} \cdot \sin \frac{\alpha - \beta}{2} \quad (\text{B.21})$$

$$1 + 2 \sum_{n=1}^{\infty} \cos(2\pi n t) = \sum_{n=-\infty}^{\infty} \delta(t - n) \quad (\text{B.22})$$

$$\arccos(\alpha) = \arctan \frac{\sqrt{1 - \alpha^2}}{\alpha} \quad (\text{B.23})$$

#### Sinus und Cosinus gemischt

$$\sin \alpha \cdot \cos \beta = \frac{1}{2} [\sin(\alpha - \beta) + \sin(\alpha + \beta)] \quad (\text{B.24})$$

$$\sin \alpha \cdot \cos \alpha = \frac{1}{2} \sin 2\alpha \quad (\text{B.25})$$

## B.2.2 Taylorreihenentwicklung für Sinus und Cosinus Funktionen

$$\sin(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n+1}}{(2n+1)!} = x - \frac{x^3}{6} + \frac{x^5}{120} - \dots \approx x \text{ for } |x| \ll 1 \quad (\text{B.26})$$

$$\cos(x) = \sum_{n=0}^{\infty} (-1)^n \frac{x^{2n}}{(2n)!} = 1 - \frac{x^2}{2} + \frac{x^4}{24} - \dots \approx 1 \text{ for } |x| \ll 1; \quad (\text{B.27})$$

$$1 - \cos(x) = \frac{x^2}{2} - \frac{x^4}{24} + \dots \approx \frac{x^2}{2} \text{ for } |x| \ll 1 \quad (\text{B.28})$$

## B.3 Exponentialfunktion und Logarithmus

### B.3.1 Allgemeines

$$a^x = e^{x \cdot \ln a} = 10^{x \cdot \lg a} \quad (\text{B.29})$$

$$a^{(x+y)} = a^x \cdot a^y \quad (\text{B.30})$$

$$a^{(x-y)} = a^x / a^y \quad (\text{B.31})$$

$$a^{(x \cdot y)} = (a^x)^y \quad (\text{B.32})$$

$$(a \cdot b)^x = a^x \cdot b^x \quad (\text{B.33})$$

$$\log_a x = \lg x / \lg a = \ln x / \ln a \quad (\text{B.34})$$

$$\log(a \cdot b) = \log a + \log b \quad (\text{B.35})$$

### B.3.2 Reihenentwicklung

$$e^z = \sum_{n=0}^{\infty} \frac{z^n}{n!} \approx 1 + z \text{ für } |z| \ll 1 \quad (\text{B.36})$$

$$\ln(z) = \ln|z| + j(\arg(z) + 2k\pi); k \in \mathbb{Z} \text{ wegen Periodizität der Exponentialfunktion} \quad (\text{B.37})$$

$$= \sum_{n=0}^{\infty} \frac{2}{2n+1} \left( \frac{z-1}{z+1} \right)^{2n+1} \approx 2 \frac{z-1}{z+1} \text{ für } z \approx 1 \text{ im Hauptzweig } (k=0) \quad (\text{B.38})$$

## B.4 Quadratische Gleichung

Die quadratische Gleichung  $x^2 + px + q = 0$  hat die Wurzel(n)

$$x_{1,2} = \frac{-p \pm \sqrt{p^2 - 4q}}{2} \quad (\text{B.39})$$

in den folgenden Bereichen

i)	$p^2 < 4q$	$\Rightarrow$	zwei komplexe Lösungen
ii)	$p^2 = 4q$	$\Rightarrow$	eine reelle Lösung
iii)	$p^2 > 4q$	$\Rightarrow$	zwei reelle Lösungen
iv)	$p > 0, p^2 \neq 4q$	$\Rightarrow$	zwei Lösungen in der linken Halbebene
v)	$p < 0, p^2 \neq 4q$	$\Rightarrow$	zwei Lösungen in der rechten Halbebene

Die Gleichung  $x^N \pm a = 0$  lässt sich auflösen mit Hilfe der Identität  $1 \equiv e^{j2\pi k}$  für alle  $k \in \mathbb{Z}$ :

$$x^N - a = 0 \Leftrightarrow x_{0,k} = \begin{cases} \sqrt[N]{a e^{j2\pi k}} = \sqrt[N]{a} e^{j2\pi k/N} & \text{für } a \in \mathbb{R} > 0 \\ \sqrt[N]{|a|} e^{j(2\pi k+1)/N} & \text{für } a \in \mathbb{R} < 0 \end{cases}; \quad k = 0 \dots N-1 \quad (\text{B.40})$$

## B.5 Summenformeln

Bei zeitdiskreten LTI-Systemen durchläuft das Signal eine endliche (transversale Systeme) bzw. unendliche (rekursive Systeme) Anzahl von Verzögerungen, die in der  $z$ -Ebene mit  $z^{-k}$  beschrieben werden. Das Ausgangssignal wird aus der Kombination (= Summe) von verzögerten Kopien des Signals bestimmt und lässt sich in einfachen Fällen mit Hilfe der Summenformeln für die geometrische Reihe bequem bestimmen:

### B.5.1 Unendliche geometrische Reihe

$$\sum_{k=0}^{\infty} z^k = \frac{1}{1-z} \quad \text{für } |z| < 1 \quad (\text{B.41})$$

$$\Rightarrow \sum_{k=0}^{\infty} z^{-k} = \frac{1}{1-z^{-1}} \quad \text{für } |z| > 1$$

Ersetzt man  $z$  durch  $z^{-1}$ , erhält man die Systemfunktion eines Integrators.

### B.5.2 Endliche geometrische Reihe

$$\sum_{k=0}^N z^k = \frac{1-z^{N+1}}{1-z} \quad \text{für } z \neq 1 \quad (\text{B.42})$$

$$\Rightarrow \sum_{k=0}^N z^{-k} = \frac{1-z^{-(N+1)}}{1-z^{-1}} \quad \text{für } z \neq 1$$

Letztere Formel kann man leicht interpretieren als Systemfunktion eines MA-Filters der Ordnung  $N = L - 1$ . Man kann erkennen, dass man den gleichen Frequenzgang erhielt, wenn man ein Kammfilter der Ordnung  $N$  mit  $H_1(z) = 1 - z^{-(N+1)}$  und einen Integrator mit  $H_2(z) = 1/(1 - z^{-1})$  kaskadiert. Praktisch ist das allerdings nur mit Einschränkungen möglich, z.B. bei CIC-Filtern in Zweierkomplementarithmetik.

### B.5.3 Dirichlet-Kern

Um den Frequenzgang eines MA-Filters der Länge  $L$  zu berechnen, ersetzt man  $z = e^{j\Omega}$  und erhält:

$$\begin{aligned} \text{Dirichlet: } \quad \text{di}_L(2\Omega) &= \frac{1}{L} \sum_{k=-(L-1)}^{L-1} e^{jk\Omega} = \frac{1}{L} \frac{e^{jL\Omega} - e^{-jL\Omega}}{e^{j\Omega} - e^{-j\Omega}} \\ &= \begin{cases} \frac{\sin L\Omega}{L \sin \Omega} & \text{für } \Omega \neq k\pi \\ (-1)^{k(L-1)} & \text{sonst} \end{cases} \end{aligned} \quad (\text{B.43})$$

Der Betrag des Dirichlet-Kerns (die Dirichlet-Funktion ist etwas völlig anderes!) entspricht dem Betragsspektrum eines  $N$ -Punkte Rechteckfensters bzw. eines Moving Average Filters der Länge  $L$  (Ordnung  $N = L - 1$ ), skaliert mit  $1/L$ :

$$\begin{aligned} h[n] &= \frac{1}{L} \sum_{k=0}^{L-1} \delta[n - k] \\ \circ \bullet H(e^{j\Omega}) &= \frac{1}{L} \sum_{k=0}^{L-1} e^{jk\Omega} = \frac{e^{jL\Omega} - 1}{e^{j\Omega} - 1} = \frac{e^{-j(L-1)\Omega/2}}{L} \frac{e^{jL\Omega/2} - e^{-jL\Omega/2}}{e^{j\Omega/2} - e^{-j\Omega/2}} \\ \Rightarrow |H(e^{j\Omega})| &= |\text{di}_L(\Omega)| = \begin{cases} \frac{\sin L\Omega/2}{L \sin \Omega/2} & \text{für } \Omega \neq 2k\pi \\ (-1)^{k(L-1)} & \text{sonst} \end{cases} \end{aligned} \quad (\text{B.44})$$

Es gilt ferner:

$$\text{di}_L(\Omega) = \frac{\sin L\Omega/2}{L \sin \Omega/2} \approx \frac{\sin L\Omega/2}{L\Omega/2} = \text{si}(L\Omega/2) \quad \text{für } \Omega \ll 1 \quad (\text{B.45})$$

$$\text{di}_{L \rightarrow \infty}(\Omega) = \sum_{k=-\infty}^{\infty} \delta(\Omega - 2k\pi) \quad (\text{Dirac-Schar, ohne Beweis}) \quad (\text{B.46})$$

### B.5.4 Verwandte Reihen

$$\text{Unendlich: } \sum_{k=0}^{\infty} ka^k = \frac{a}{(a-1)^2} \quad \text{für } |a| < 1 \quad (\text{B.47})$$

$$\text{Endlich: } \sum_{k=0}^{N-1} ka^k = \frac{(N-1)a^{N+1} - Na^N + a}{(a-1)^2} \quad \text{für } a \neq 1 \quad (\text{B.48})$$

$$\text{Unendlich: } \sum_{k=0}^{\infty} k^2 a^k = \frac{a(1+a)}{(a-1)^3} \quad \text{für } |a| < 1 \quad (\text{B.49})$$

$$\text{Endlich: } \sum_{k=0}^{N-1} k^2 a^k = \frac{(N-1)^2 a^{N+2} - (2N^2 - 2N - 1)a^{N+1} + N^2 a^N - a^2 - a}{(a-1)^3} \quad \text{für } a \neq 1 \quad (\text{B.50})$$

### B.5.5 Einfache Reihen

Mit Hilfe der folgenden einfachen Reihen kann man z.B. Rechnungen zu Fensterfunktionen für die Fouriertransformation einfacher durchführen.

$$\text{Lineare Reihe} \quad \sum_{k=1}^n k = \frac{n(n+1)}{2} \quad (\text{B.51})$$

$$\text{Quadratische Reihe} \quad \sum_{k=1}^n k^2 = \frac{n(n+1)(2n+1)}{6} \quad (\text{B.52})$$

$$\text{Kubische Reihe} \quad \sum_{k=1}^n k^3 = \left( \frac{n(n+1)}{2} \right)^2 \quad (\text{B.53})$$

Lösungen für beliebige ganzzahlige Exponenten gibt die sog. „Faulhabersche Formel“ (-> Bronstein, Wikipedia, ...).



Abb. B.2: Gefährliche Fußnoten [<http://xkcd.com/1184/>]

### B.6 Reihenentwicklung

Zur Annäherung der trigonometrischen Funktionen werden die Taylorreihen um den Punkt 0 herum berechnet (MacLaurinsche Reihe),  $x$  muss dabei in Bogenmaß (rad) angegeben werden:

$$\sin(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{(2k+1)!} = x - \frac{x^3}{6} + \frac{x^5}{120} \mp \dots \quad (\text{B.54})$$

$$\cos(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k}}{(2k)!} = 1 - \frac{x^2}{2} + \frac{x^4}{24} \mp \dots \quad (\text{B.55})$$

$$\tan(x) = x + \frac{x^3}{3} + \frac{2x^5}{15} + \dots \quad \text{für } |x| < \frac{\pi}{2} \quad (\text{B.56})$$

$$\arctan(x) = \sum_{k=0}^{\infty} (-1)^k \frac{x^{2k+1}}{2k+1} = x - \frac{x^3}{3} + \frac{x^5}{5} \mp \dots \quad \text{für } |x| \leq 1 \text{ und } x \neq \pm j \quad (\text{B.57})$$

### B.7 Ein paar Integrale

$$\int_{t=0}^{T_1} \cos^2(2\pi k f_1 \tau) d\tau = \frac{\tau}{2} + \frac{\sin(2\pi k f_1 \tau) \cos(2\pi k f_1 \tau)}{2 \cdot 2\pi k f_1} \Big|_{\tau=0}^{T_1} = T_1/2 \quad (\text{B.58})$$

$$\int_{t=0}^{T_1} \sin^2(2\pi k f_1 \tau) d\tau = \frac{\tau}{2} - \frac{\sin(2\pi k f_1 \tau) \cos(2\pi k f_1 \tau)}{2 \cdot 2\pi k f_1} \Big|_{\tau=0}^{T_1} = T_1/2 \quad (\text{B.59})$$

## C Fourier-Transformation

### C.1 Fourier-Transformation kontinuierlicher Signale

$$\text{Fourierintegral: } s(t) = \int_{-\infty}^{\infty} S(f) e^{j2\pi f t} df \quad \circ \bullet \quad S(f) = \int_{-\infty}^{\infty} s(t) e^{-j2\pi f t} dt \quad (\text{C.1})$$

$$= \frac{1}{2\pi} \int_{-\infty}^{\infty} S(\omega) e^{j\omega t} d\omega \quad \circ \bullet \quad S(\omega) = \int_{-\infty}^{\infty} s(t) e^{-j\omega t} dt \quad (\text{C.2})$$

**Anmerkung:** Aufgrund der Symmetrie und da die Integration jeweils über  $-\infty$  bis  $\infty$  läuft, ist die Wahl der Vorzeichen in der Hin- und Rücktransformation im Prinzip willkürlich. Auch der Vorfaktor kann beliebig auf Hin- und Rückrichtung aufgeteilt werden. In diesem Skript wird der Faktor immer so gewählt, dass  $S(0)$  dem DC-Wert entspricht, in der Literatur (und in Software-Bibliotheken!) gibt es aber auch andere Definitionen.

Die **Fourierreihe** erhält man für Signale, die mit  $T_1 = 1/f_1$  periodisch sind. Hier ergeben sich diskrete Linienspektren mit komplexen Koeffizienten  $c_k \equiv S[kf_1]$ :

$$\text{Fourierreihe: } s(t) = \sum_{k=-\infty}^{\infty} c_k e^{j2\pi k f_1 t} \quad \circ \bullet \quad c_k \equiv S[kf_1] = \frac{1}{T_1} \int_{-T_1/2}^{T_1/2} s(t) e^{-j2\pi k f_1 t} dt; \quad k \in \mathbb{Z} \quad (\text{C.3})$$

$$= \frac{a_0}{2} + \sum_{k=1}^{\infty} a_k \cos 2\pi k f_1 t + b_k \sin 2\pi k f_1 t \quad \circ \bullet \quad a_k \equiv \Re \{S[kf_1]\} = \frac{1}{T_1} \int_{-T_1/2}^{T_1/2} s(t) \cos(2\pi k f_1 t) dt; \quad k \in \mathbb{N}_0 \quad (\text{C.4})$$

$$b_k \equiv \Im \{S[kf_1]\} = \frac{1}{T_1} \int_{-T_1/2}^{T_1/2} s(t) \sin(2\pi k f_1 t) dt; \quad k \in \mathbb{N} \quad (\text{C.5})$$

$$\text{Ähnlichkeit: } a \cdot s(bt) \quad \circ \bullet \quad \frac{a}{|b|} S\left(\frac{f}{b}\right) \quad (\text{C.6})$$

<b>Diracstoß:</b>	$\delta(t)$	○•	1	(C.7)
<b>Rect-Puls:</b>	$\text{rect}\left(\frac{t}{T_w}\right)$	○•	$T_w \frac{\sin \pi T_w f}{\pi T_w f} = T_w \text{si}(\pi T_w f)$	(C.8)
<b>si-Puls:</b>	$\text{si}(f_0 t)$	○•	$\frac{1}{f_0} \text{rect} \frac{f}{f_0}$	(C.9)
<b>Period. <math>\delta</math> – Funktion:</b>	$\sum_{n=-\infty}^{\infty} \delta(t - nT)$	○•	$\frac{1}{T} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right)$	(C.10)
<b>Ideale Abtastung:</b>	$s(t) \sum_{n=-\infty}^{\infty} \delta(t - nT)$	○•	$\frac{1}{T} \sum_{n=-\infty}^{\infty} S\left(f - \frac{n}{T}\right)$	(C.11)
<b>Cosinus-Fkt.:</b>	$\cos 2\pi f_0 t$	○•	$\frac{1}{2} [\delta(f - f_0) + \delta(f + f_0)]$	(C.12)
<b>Sinus-Fkt.:</b>	$\sin 2\pi f_0 t$	○•	$\frac{j}{2} [\delta(f - f_0) - \delta(f + f_0)]$	(C.13)
<b>BP-Transformation:</b>	$s(t) \cdot \cos 2\pi f_0 t$	○•	$\frac{1}{2} [S(f - f_0) + S(f + f_0)]$	(C.14)
<b>Frequenzverschiebung:</b>	$s(t) \cdot e^{j2\pi f_0 t}$	○•	$S(f - f_0)$	(C.15)
<b>Verzögerung:</b>	$s(t - \Delta T)$	○•	$S(f) e^{-j2\pi f \Delta T} = S(f) e^{-jf \Delta \phi T_0}$ mit $\Delta \phi \doteq \frac{2\pi \Delta T}{T_0}$	(C.16)
<b>Integrator:</b>	$\int_{-\infty}^t s(\tau) d\tau$	○•	$\frac{S(f)}{j2\pi f} + \frac{1}{2} S(0) \delta(f)$	(C.17)
<b>Einheitssprung:</b>	$u(t)$	○•	$\frac{\delta(f)}{2} + \frac{1}{j2\pi f}$	(C.18)

$$\text{ZOH: } \text{rect}\left(\frac{t}{\tau}\right) * \left[ s(t) \sum_{n=-\infty}^{\infty} \delta(t - nT) \right] \xrightarrow{\text{○•}} \frac{\tau}{T} \text{si}(\pi \tau f) \sum_{n=-\infty}^{\infty} S\left(f - \frac{n}{T}\right) \quad (\text{C.19})$$

$$\text{Period. Signal: } \sum_{n=-\infty}^{\infty} s(t - nT) = s(t) * \sum_{n=-\infty}^{\infty} \delta(t - nT) \xrightarrow{\text{○•}} \frac{S(f)}{T} \sum_{n=-\infty}^{\infty} \delta\left(f - \frac{n}{T}\right) \quad (\text{C.20})$$

$$\text{Period. rect-Signal: } \text{rect}\left(\frac{t}{T_w}\right) * \sum_{n=-\infty}^{\infty} \delta(t - nT) \xrightarrow{\text{○•}} \sum_{n=-\infty}^{\infty} \frac{T_w}{T} \text{si}\left(\frac{\pi n T_w}{T}\right) \delta\left(f - \frac{n}{T}\right) \quad (\text{C.21})$$

## C.2 Fourier-Transformation diskreter Signale mit unendlicher Länge (DTFT)

### C.2.1 Definition

Die Fouriertransformation einer *unendlichen* mit  $T_S$  abgetasteten äquidistanten Folge von Abtastwerten  $s[n] \equiv s[nT_S]$  ist die **Discrete Time Fourier Transform** oder **DTFT** mit den normalisierten Frequenzen  $F = f/f_S$  und  $\Omega = \omega/\omega_S$  und  $\omega = 2\pi f$ . Die DTFT ist periodisch mit  $f_S$  (bzw.  $\omega_S$ ,  $\Omega_S$  etc.).

Abhängig von der Frequenzeinheit ergeben sich unterschiedliche Skalierungsfaktoren und Integrationsgrenzen. Die Skalierungsfaktoren wurden so gewählt, dass sich physikalisch sinnvolle Größen im Zeit- und Frequenzbereich ergeben und Parsevals Gesetz ohne weitere Skalierungsfaktoren gilt. Im Folgenden sind (sofern nicht anders definiert)  $k, m, n \in \mathbb{Z}$ ,  $M, N \in \mathbb{N}$ ,  $c, F_1, W \in \mathbb{R}$  mit  $|F_1| < 0,5$  und  $0 < W < 1$ .

$$\text{DTFT: } s[n] \equiv s[nT_S] = T_S \int_{-f_S/2}^{f_S/2} S(e^{j2\pi f T_S}) e^{j2\pi f T_S n} df \quad \circ \bullet \quad S(e^{j2\pi f T_S}) = \sum_{n=-\infty}^{\infty} s[n] e^{-j2\pi f T_S n} \quad (\text{C.22})$$

$$= \int_{-1/2}^{1/2} S(e^{j2\pi F}) e^{j2\pi F n} dF \quad \circ \bullet \quad S(F) = \sum_{n=-\infty}^{\infty} s[n] e^{-j2\pi F n} \quad (\text{C.23})$$

$$= \frac{T_S}{2\pi} \int_{-\pi f_S}^{\pi f_S} S(e^{j\omega T_S}) e^{j\omega T_S n} d\omega \quad \circ \bullet \quad S(e^{j\omega T_S}) = \sum_{n=-\infty}^{\infty} s[n] e^{-j\omega T_S n} \quad (\text{C.24})$$

$$= \frac{1}{2\pi} \int_{-\pi}^{\pi} S(e^{j\Omega}) e^{j\Omega n} d\Omega \quad \circ \bullet \quad S(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} s[n] e^{-j\Omega n} \quad (\text{C.25})$$

### C.2.2 Eigenschaften

$$\text{Linearität: } as[n] \quad \circ \bullet \quad aS(e^{j2\pi F}) \quad (\text{C.26})$$

$$\text{Zeitverschiebung: } s[n - m] \quad \circ \bullet \quad S(e^{j2\pi F}) e^{-j2\pi F m} \quad (\text{C.27})$$

$$\text{Modulation: } s[n] e^{j2\pi F_1 n} \quad \circ \bullet \quad S(e^{j(2\pi F - F_1)}) \quad (\text{C.28})$$

$$\text{Diracstoß: } c\delta[n] \quad \circ \bullet \quad c \quad (\text{C.29})$$

$$\text{Konstante: } c = c \text{III}[k] \quad \circ \bullet \quad c\delta(e^{j2\pi F}) \quad (\text{C.30})$$

$$\text{Rect-Puls (Breite } N\text{): } \text{rect}\left[\frac{n-m}{N}\right] = \begin{cases} 1 & \text{für } |n-m| \leq N/2 \\ 0 & \text{sonst} \end{cases} \quad \circ \bullet \quad \frac{\sin(NF)}{\sin(F/2)} e^{-j2\pi F m} \quad (\text{C.31})$$

$$\text{si-Puls: } \text{si}(Wn) = \frac{\sin(\pi Wn)}{\pi Wn} \quad \circ \bullet \quad \frac{1}{W} \text{rect}(WF) = \begin{cases} 1/W & \text{für } |F| \leq 1/2W \\ 0 & \text{sonst} \end{cases} \quad (\text{C.32})$$

$$\text{si}^2\text{-Puls: } \text{si}^2(Wn) = \left( \frac{\sin(\pi Wn)}{\pi Wn} \right)^2 \quad \circ \bullet \quad \frac{1}{W} \text{tri}(WF) = \begin{cases} \left| \frac{F}{W} \right| & \text{für } |F| \leq \frac{1}{W}; 0 < W < \frac{1}{2} \\ 0 & \text{sonst} \end{cases} \quad (\text{C.33})$$

$$\text{Cosinus-Fkt.: } \cos[2\pi F_1 n] \quad \circ \bullet \quad \frac{1}{2} (\delta(F - F_1) + \delta(F + F_1)) \quad (\text{C.34})$$

$$\text{Sinus-Fkt.: } \sin[2\pi F_1 n] \quad \circ \bullet \quad \frac{j}{2} (\delta(F - F_1) - \delta(F + F_1)) \quad (\text{C.35})$$

$$\text{Faltung im Zeitbereich: } x[n] * y[n] \quad \circ \bullet \quad X(e^{j2\pi F}) \cdot Y(e^{j2\pi F}) \quad (\text{C.36})$$

$$\text{Mult. im Zeitbereich: } x[n] \cdot y[n] \quad \circ \bullet \quad \int_{-1/2}^{1/2} X(e^{j2\pi\nu}) \cdot Y(e^{j2\pi(F-\nu)}) d\nu \quad (\text{C.37})$$

$$\text{Parcevals Theorem: } E_{xx} = \sum_{n=-\infty}^{\infty} x[n] \cdot x[n]^* = \sum_{n=-\infty}^{\infty} |x[n]|^2 = \int_{-1/2}^{1/2} |X(e^{j2\pi F})|^2 dF \quad (\text{C.38})$$

$$E_{xy} = \sum_{n=-\infty}^{\infty} x[n] \cdot y[n]^* = \int_{-1/2}^{1/2} X(e^{j2\pi F}) \cdot Y(e^{j2\pi F})^* dF \quad (\text{C.39})$$

$$P_{xx} = \frac{1}{N} \sum_{n=-N/2}^{N/2} |x[n]|^2 = \frac{1}{N} \int_{-1/2}^{1/2} |X(e^{j2\pi F})|^2 dF \quad (\text{C.40})$$

## Anmerkungen

**si-Puls:**  $W$  ist die Breite des „Hauptzipfels“ der si-Funktion

**Multiplikation im Zeitbereich:** Aufgrund der periodischen Spektren muss im Frequenzbereich eine zyklische oder periodische Faltung angewendet werden.

**Parcevals Theorem** gilt nur für zeitlich begrenzte Signale (Energiesignale); periodische Signale haben unendliche Energie. Hier kann man im Zeitbereich über die Periodendauer  $T_1$  aufsummieren und die Gleichung auf beiden Seiten mit  $1/T_1$  skalieren - so setzt man die Leistung anstatt der Energien (genau genommen Satz von Plancherel).

## C.3 Fourier-Transformation diskreter Signale der Länge N (DFT)

### C.3.1 Definition

Für eine periodische Funktion mit  $N$  Samples pro Periode  $T_1$  erhält man die **Diskrete Fourier Transformation** oder **DFT**. Mit  $T_1 = NT_S$  bzw.  $f_1 = f_S/N$  kann die DFT aus der Fourierreihe oder aus der DTFT abgeleitet werden. Die Skalierung mit  $1/N$  (vergleichbar mit der Fourierreihe und  $1/T_1$ ) wird in der Frequenzdomäne vorgenommen:

$$\begin{aligned} \text{DFT: } s[n] &\equiv s[nT_S] = \sum_{k=0}^{N-1} S[kf_S/N] e^{j2\pi n T_S k f_S / N} & \circ \bullet & S[k] \equiv S[kf_1] = \frac{1}{N} \sum_{n=0}^{N-1} s[nT_S] e^{-j2\pi n T_S k f_S / N} \\ &= \sum_{k=0}^{N-1} S[k] e^{j2\pi kn / N} = \sum_{k=0}^{N-1} S[k] W_N^{kn} & &= \frac{1}{N} \sum_{n=0}^{N-1} s[n] e^{-j2\pi kn / N} = \frac{1}{N} \sum_{n=0}^{N-1} s[n] W_N^{-kn} \quad (\text{C.41}) \\ & \text{mit } W_N^k \equiv e^{j2\pi k / N} \end{aligned}$$

### C.3.2 Eigenschaften

Die folgenden Eigenschaften gelten für eine DFT über  $N$  Samples. Wie üblich, kann man die Eigenschaften auch als Eigenschaften eines periodischen Signals mit Periode  $N$  ansehen.

$$\text{Linearität: } as[n] \circ \bullet aX[k] \quad (\text{C.42})$$

$$\text{Periodizität: } s[n] = x[n - N] \circ \bullet X[k] = X[k - N] \quad (\text{C.43})$$

$$\text{Zeitverschiebung: } s[n - m], m \in \mathbb{Z} \circ \bullet X[k] e^{-j2\pi km / N} \quad (\text{C.44})$$

$$\text{Modulation: } s[n] e^{j2\pi nm / N} \circ \bullet X[k - m], m \in \mathbb{Z} \quad (\text{C.45})$$

$$\text{Diracstoß: } \delta[n] \circ \bullet \frac{1}{N} \sum_{i=0}^{N-1} \delta[k - i] = \frac{1}{N} \text{III}_N[k] = \frac{1}{N} \quad (\text{C.46})$$

$$\text{Konstante: } c = c \text{III}_N[k] \circ \bullet \frac{c}{N} \delta[k] \quad (\text{C.47})$$

$$\text{Rect-Puls (Breite } 2N_1\text{): } \text{rect}_N[n/N_1] = \begin{cases} 1 & \text{für } |n| \geq N_1 > N/2 \\ 0 & \text{sonst} \end{cases} \circ \bullet \frac{\sin(\pi k(2N_1 + 1)/N)}{N \sin(\pi k/N)} \quad (\text{C.48})$$

$$\text{si-Puls: } \frac{\sin(\pi n(2N_1 + 1)/N)}{\sin(\pi n/N)} \circ \bullet \text{rect}_N[k/N_1] = \begin{cases} 1 & \text{für } |k| \geq N_1 > N/2 \\ 0 & \text{sonst} \end{cases} \quad (\text{C.49})$$

Cosinus-Fkt.:	$\cos[2\pi mn/N]$	$\circ \bullet$	$\frac{1}{2} (\delta[k - m] + \delta[k + m - N])$	(C.50)
Sinus-Fkt.:	$\sin[2\pi mn/N]$	$\circ \bullet$	$\frac{j}{2} (\delta[k - m] - \delta[k + m - N])$	(C.51)
Zyklische Faltung:				(C.52)
Multiplikation im Zeitbereich:				(C.53)

## D $z$ -Transformation

<b>Definition:</b>	$x[n]$	$\circ \bullet$	$\mathcal{L}\{x[n]\} = X(z) = \sum_{n=-\infty}^{\infty} z^{-n} x[n]$	(D.1)
<b>Linearität:</b>	$a_1 x_1[n] + a_2 x_2[n]$	$\circ \bullet$	$a_1 X_1(z) + a_2 X_2(z)$	(D.2)
<b>Diracstoß:</b>	$\delta[n]$	$\circ \bullet$	1	(D.3)
<b>Faltung:</b>	$x_1[k] * x_2[k]$	$\circ \bullet$	$X_1(z)X_2(z)$	(D.4)
<b>Verzögerung um <math>k</math>:</b>	$x[n - k]$	$\circ \bullet$	$X(z)z^{-k}$	(D.5)
<b>Spiegelung:</b>	$x[-n]$	$\circ \bullet$	$X(-z)$	(D.6)
<b>Einheitssprung:</b>	$u[n]$	$\circ \bullet$	$\frac{z}{z - 1} = \frac{1}{1 - z^{-1}}$	(D.7)
<b>Rampe:</b>	$nu[n]$	$\circ \bullet$	$\frac{z}{(z - 1)^2} = \frac{z^{-1}}{(1 - z^{-1})^2}$	(D.8)
<b>Exponentielle Folge:</b>	$a^n u[n]$	$\circ \bullet$	$\frac{z}{z - a} = \frac{1}{1 - az^{-1}}$	(D.9)
<b>Exponentielle Folge:</b>	$a^{n-1} u[n]$	$\circ \bullet$	$\frac{1}{z - a} = \frac{z^{-1}}{1 - az^{-1}}$	(D.10)
<b>Upsampling um <math>L</math>:</b>	$y[n] = \begin{cases} x[n/L] & \text{für } n = 0, \pm L, \pm 2L, \dots \\ 0 & \text{sonst (Nullenstopfen)} \end{cases}$	$\circ \bullet$	$Y(z) = X(z^L)$	(D.11)
<b>Downsampling um <math>R</math>:</b>	$y[n] = x[Rn]$	$\circ \bullet$	$Y(z) = \frac{1}{R} \sum_{k=0}^{R-1} X(z^{1/R} e^{j2\pi k/R})$	(D.12)



# Abbildungsverzeichnis

1	XKCD: Latex rules . . . . .	iii
1	XKCD: Was Sie in diesen Unterlagen finden . . . . .	xix
2	XKCD: Fehler finden und ganz für sich behalten ist voll egoistisch! . . . . .	xxi
3	Screenshot eines Jupyter Notebooks . . . . .	xxii
4	PYNQ Board (c) Digilent . . . . .	xxiii
5	pyFDA Screenshot . . . . .	xxiii
6	XKCD: Darf's etwas mehr sein? . . . . .	xxiv
1.1	XKCD: Digital Data . . . . .	3
1.2	IIR-Filter zweiter Ordnung in Direktform mit eingezeichnetem kritischen Pfad . . . . .	4
1.3	Moving Average (MA) (a), kaskadiertes MA - Filter (b) und verlustbehafteter (gedämpfter) Integrator (c) . . . . .	7
1.4	FIR-Filter mit Eingangssignal . . . . .	8
1.5	IIR-Systeme zweiter Ordnung mit identischer Systemfunktion in Direktform 1 und Direktform 2 . . . . .	9
1.6	Filterstrukturen zu Aufgabe 1.6, 1.7 und 2.6 . . . . .	10
2.1	XKCD: Imaginäre Erklärungen . . . . .	13
2.2	Pol-/Nullstellenpläne zu Aufgabe 2.5 . . . . .	18
2.3	FIR-Filterstruktur und Amplitudengang . . . . .	19
2.4	FIR-Filter zu Aufgabe 2.8 . . . . .	20
2.5	IIR-Filter (a) mit Eingangssignal (b) zu Aufgabe 2.10 . . . . .	21
3.1	XKCD: That cat has some serious periodic components . . . . .	23
3.2	Signalflussgraph einer DFT mit 8 Punkten . . . . .	25
3.3	Signalflussgraph einer FFT mit 8 Punkten . . . . .	25
3.4	Python Plot zu Listing 3.1 . . . . .	30
3.5	Seismogramm eines Erdbebens am 7.11.2012 in Guatemala. Mit freundlicher Genehmigung der Bundesanstalt für Geowissenschaften und Rohstoffe [ <a href="http://www.seismologie.bgr.de/sdac/erdbeben/big_quakes/guatemala_121107_deu.html">http://www.seismologie.bgr.de/sdac/erdbeben/big_quakes/guatemala_121107_deu.html</a> ] . . . . .	31
4.1	XKCD: Warum gibt es so viele verschiedene Fenster? . . . . .	33
4.2	Leckeffekt bei Rechteck- (a) und Hann-Fensterung (b) . . . . .	35
4.3	Screenshot des pyFDA Windowviewers . . . . .	39
5.1	XKCD: Irgendwo da draußen ist das perfekte Filter . . . . .	43
5.2	Filterspezifikationen für FIR und IIR-Filter . . . . .	44
5.3	Amplitudengang des Filters $F$ im (a) linearen und (b) logarithmischen Maßstab . . . . .	51
5.4	XKCD: Konjugiert <i>und</i> komplex . . . . .	51
5.5	Betragsfrequenzgang zu Aufgabe 5.6 . . . . .	53
6.1	XKCD: Schlaflose Programmierer . . . . .	57

6.2	Verteilungsdichtefunktionen des Quantisierungsfehlers $\epsilon_n$ für verschiedene Requantisierungen . . . . .	59
6.3	Fixpoint Moving Average Filter . . . . .	61
6.4	FIR-Filter mit Quantisierung zu Aufgabe 6.4 . . . . .	62
6.5	Betragsfrequenzgang zu Aufgabe 6.5 . . . . .	63
6.6	Integratoren mit endlicher Wortbreite . . . . .	64
7.1	XKCD: And if you labeled your axes ... . . . . .	67
7.2	XKCD: Korrelation . . . . .	68
7.3	Auswirkung der Signalquantisierung im Zeit- und Frequenzbereich . . . . .	73
7.4	Spektrale Leistungsdichte am Ausgang eines ADCs (Aufgabe 7.1d) . . . . .	75
8.1	XKCD: Abtasten ist gefährlich . . . . .	77
8.2	Amplituden- und Phasenspektren von zeitkontinuierlichen und abgetasteten Signalen . . . . .	80
8.3	Zweistufige Dezimation zu Aufgabe 8.6 . . . . .	81
8.4	Multiraten-Tiefpassfilter und Betragsgang der Teilternehmen zu Aufgabe 8.7 . . . . .	82
9.1	$I = 4$ : Basisband und Images . . . . .	85
9.2	Spektrum von $s[n]$ , (a) vor und (b) nach Abtastraten erhöhung um $I = 3$ . . . . .	88
9.3	DAC ohne (a) und mit (b) Oversampling zu Aufgabe 9.4 . . . . .	92
9.4	ZOH-Signal $x_2(t)$ und Amplitudenspektrum $ X_2(f) $ zu Aufgabe 9.4a . . . . .	92
9.5	Spezifikationen für Interpolationsfilter $H_{int}$ (a) und Rekonstruktionstiefpässe $H_{RKTP,i}$ (b) zu Aufgabe 9.4b und d) . . . . .	93
9.6	ZOH-Signal $x_5(t)$ und Amplitudenspektrum $X_5(f)$ zu Aufgabe 9.4c . . . . .	93
9.7	Zweistufige Interpolation zu Aufgabe 9.5 . . . . .	93
10.1	XKCD: Wenn Ingenieure blinken . . . . .	95
10.2	Synchrone Abtastratenwandlung zu Aufgabe 10.1 . . . . .	98
10.3	Abtastratenwandlung zu Aufgabe 10.2 und 10.3 . . . . .	99
10.4	Spektren und Frequenzgänge zu Aufgabe 10.2 . . . . .	99
M1.1	XKCD: Flowchart . . . . .	105
M1.2	Eingangssignal $x(t) = 1,5 \text{ V} + 0,5 \text{ V} \cos(2\pi \cdot 50 \text{ Hz} \cdot t)$ , abgetastet mit verschiedenen Samplingraten $f_s$ . . . . .	106
M1.3	MA-Filter der Ordnung $M - 1$ und effiziente Implementierung mit kritischen Pfaden . . . . .	109
M1.4	FIR-Filter mit Eingangssignal . . . . .	110
M1.5	FIR Filter in Direktform und in „gefalteter Form“ . . . . .	113
M1.6	Graphische Umwandlung eines Filters von DF1- zu DF2-Form . . . . .	113
M1.7	Systeme zweiter Ordnung in Direktform 1 und 2 mit kritischen Pfaden . . . . .	115
M1.8	Addierer mit drei Eingängen . . . . .	116
M1.9	Kritische Pfade zu Aufgabe M1.6 . . . . .	117
M1.10	Transponierte Filterstrukturen zu Abb. 1.6 mit kritischen Pfaden . . . . .	118
M2.1	PN-Diagramme zu Aufgabe 2.1 . . . . .	125
M2.2	Betrags- und Phasengang sowie Gruppenlaufzeit zu Aufgabe 2.1 . . . . .	128
M2.3	Pol/Nullstellenplan zu Aufgabe 2.2 . . . . .	129
M2.4	Betrags- und Phasengang zu Aufgabe 2.2 . . . . .	130
M2.6	MA-Filter mit $N = 32$ Taps, (a) Amplitudengang und (b) P/N-Diagramm . . . . .	138
M2.7	. . . . .	139
M2.8	Betrags- und Phasengang . . . . .	140
M2.9	System a) . . . . .	140
M2.10	Pol-Nullstellen-Plan für $\alpha = 0,5$ (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9a	141

M2.11	Impulsantwort für $\alpha = 0,5$ (a) und $\alpha = 0,9$ (b) zu Aufgabe 2.9a . . . . .	141
M2.12	System b) . . . . .	141
M2.13	Pol-Nullstellen-Plan (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9b . . . . .	142
M2.14	Impulsantwort für $\alpha = 0,5$ (a) und $\alpha = 0,9$ (b) zu Aufgabe 2.9b . . . . .	142
M2.15	System c) . . . . .	143
M2.16	Pol-Nullstellen-Plan für $\alpha = 0,5$ (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9c . . . . .	143
M2.17	Impulsantwort für $\alpha = 0,5$ (a) und $\alpha = 0,9$ (b) zu Aufgabe 2.9c . . . . .	144
M2.18	System d) . . . . .	144
M2.19	Pol-Nullstellen-Plan für $\alpha = 0,5$ (a) und Amplitudenfrequenzgang (b) zu Aufgabe 2.9d . . . . .	145
M2.20	Impulsantwort für $\alpha = 0,5$ (a) und $\alpha = 0,9$ (b) zu Aufgabe 2.9d . . . . .	145
M2.21	Pol-Nullstellen-Plan (a) und Amplitudenfrequenzgang (b) zu Abb. 2.5a . . . . .	148
M3.1	Eingangssignal $s(t)$ und abgetastetes Signal $s[n]$ (zwei Perioden) zu Aufgabe 3.2 . . . . .	155
M3.2	Betrag und Phase der DFT $S[k]$ für $f_S = 3000$ Hz und $N = 3$ zu Aufgabe 3.2c . . . . .	156
M3.3	Betrag und Phase der DFT $S[k]$ für $N = 6$ mit $f_S = 3000$ Hz zu Aufgabe 3.2e (a) und mit $f_S = 6000$ Hz zu Aufgabe 3.2f (b) . . . . .	157
M3.4	Frequenzgang mit Zeropadding . . . . .	158
M3.5	Erwartetes ein- und zweiseitiges Spektrum zu Aufgabe 3.6 . . . . .	162
M3.6	Betrag und Phase der DFT $S[k]$ für $f_S = 4000$ Hz und $N = 4$ (a) bzw. $f_S = 100$ kHz und $N = 100$ (b) zu A3.7b . . . . .	166
M5.1	Frequenzgänge und Spezifikationen im linearen und log. Maßstab sowie P/N-Diagramme für IIR- und FIR-Filter zu Aufgabe 5.1 . . . . .	172
M5.2	Amplitudengang mit Nullstellen . . . . .	175
M5.3	P/N-Diagramm . . . . .	176
M5.4	Plots zu maximalphasigem Filter $F_1$ (Aufgabe 5.3b) . . . . .	179
M5.5	Plots zu minimalphasigem Filter $F_2$ (Aufgabe 5.3d) . . . . .	181
M5.6	Plots zu linearphasigem Filter $F$ (Aufgabe 5.3e) . . . . .	182
M5.7	Blockschaltbild (a) und Implementierung (b) für Filter $F$ (Aufgabe 5.3e) . . . . .	183
M5.8	Equiripple (a) und Kaiser Window (b) Hochpassentwurf mit pyFDA zu Aufgabe 5.5	187
M5.9	Betragsfrequenzgänge zu Aufgabe 5.6 . . . . .	188
M6.1	Fixpoint Moving Average Filter mit Wortbreiten . . . . .	195
M6.2	FIR-Filter mit optimierter Struktur und Quantisierung zu M6.4 . . . . .	196
M6.3	SFG in Direktform zu Aufgabe 6.5 . . . . .	198
M7.1	Einseitiges Spektrum von Schmal- und Breitbandsignalen . . . . .	208
M8.1	Amplituden- und Phasenspektren von zeitkontinuierlichen und abgetasteten Signalen	209
M8.2	Zeitkontinuierliche Signale mit identischen Abtastwerten . . . . .	210
M8.3	Effiziente Implementierung eines MA-Filters der Ordnung $N$ . . . . .	212
M8.4	Zweistufige Dezimation, mit (a) und ohne (b) Aliasing ins Basisband am Ausgang zu Aufgabe 8.6a) . . . . .	213
M8.5	Frequenzgänge der Multiraten-Filterkaskade . . . . .	217
M9.1	ML: Spektrum von $s[n]$ , (a) vor und (b) nach Abtastratenerhöhung um $I = 3$ . . . . .	219
M9.2	Ersatzschaltbild für Interpolation mit ZOH zu Aufgabe 9.2 d) . . . . .	221
M9.3	ML: ZOH-Signal $x_2(t)$ und Amplitudenspektrum $ X_2(f) $ zu Aufgabe 9.4a . . . . .	223
M9.4	ML: Spezifikationen für Interpolationsfilter $H_{int}$ (a) und Rekonstruktionstiefpässe $H_{RKTP,i}$ (b) zu Aufgabe 9.4b und d) . . . . .	224
M9.5	ML: ZOH-Signal $x_5(t)$ und Amplitudenspektrum $X_5(f)$ zu Aufgabe 9.4c . . . . .	225

---

M10.1 Synchrone Abtastratenwandlung zu Aufgabe 10.1 . . . . .	227
M10.2 Abtastratenwandlung zu Aufgabe 10.2 . . . . .	229
M10.3 Spektren und Frequenzgänge zu Aufgabe 10.2 . . . . .	230
M10.4 Alternative Abtastratenwandlung zu Aufgabe 10.2 (vertauschte Dezimation und Interpolation) . . . . .	230
M10.5 Alternative Abtastratenwandlung zu Aufgabe 10.2 (zusammengefasstes TP-Filter) . . . . .	231
M10.6 Systeme zu Aufgabe 10.4 . . . . .	234
M10.7 System zu Aufgabe 10.5 a und b) . . . . .	235
A.1 Beispiel für einen Signallflussgraphen . . . . .	239
A.2 XKCD: Flowchart als Flowchart . . . . .	240
A.3 Zusammenfassen von seriellen (a) und parallelen (b) Zweigen . . . . .	240
A.4 Distributivität . . . . .	241
A.5 Eliminieren von Schleifen, (a) Eigenschleifen und (b) allgemeine Schleifen . . . . .	241
B.1 Komplexe $z$ -Ebene . . . . .	243
B.2 XKCD: Gefährliche Fußnoten . . . . .	248
D.1 Richtig zitieren . . . . .	264

# Tabellenverzeichnis

1.1	Spezielle Befehle zu Kap. 1 . . . . .	5
1.2	Jupyter Notebooks und Labs zu Kap. 1 . . . . .	6
1.3	Ergebnisse zu Aufgabe 1.1 . . . . .	7
2.1	Spezielle Befehle zu Kap. 2 . . . . .	15
2.2	Jupyter Notebooks zu Kap. 2 . . . . .	16
3.1	Spezielle Befehle zu Kap. 3 . . . . .	26
3.2	Jupyter Notebooks zu Kap. 3 . . . . .	27
4.1	Eigenschaften verschiedener Fensterfunktionen . . . . .	37
4.2	Spezielle Befehle zu Kap. 4 . . . . .	39
4.3	Jupyter Notebooks zu Kap. 4 . . . . .	40
5.1	Spezielle Befehle zu Kap. 5 . . . . .	48
5.2	Jupyter Notebooks zu Kap. 5 . . . . .	49
5.3	Verständnisfragen zu Filtern (Aufgabe 5.8) . . . . .	55
6.1	Codeschnipsel und Jupyter Notebooks zu Kap. 6 . . . . .	60
6.2	Verständnisfragen zu Filtern mit Quantisierung (Aufgabe 6.7) . . . . .	65
7.1	Jupyter Notebooks zu Kap. 7 . . . . .	72
8.1	Spezielle Befehle zu Kap. 8 . . . . .	78
8.2	Jupyter Notebooks zu Kap. 8 . . . . .	79
8.3	Eigenschaften der Filterkaskade aus Aufgabe 8.7 . . . . .	83
9.1	Spezielle Befehle zu Kap. 9 . . . . .	86
9.2	Jupyter Notebooks zu Kap. 9 . . . . .	87
10.1	Spezielle Befehle zu Kap. 10 . . . . .	96
10.2	Jupyter Notebooks und Labs zu Kap. 10 . . . . .	98
10.3	Lösungen zu Aufgabe 10.2 . . . . .	100
M1.1	Ergebnisse zu Aufgabe 1.1 . . . . .	107
M1.2	Zeitdiskrete Faltung $y[n] = h[n] * x[n]$ zu Aufgabe 1.2 . . . . .	112
M1.3	Zeitdiskrete Faltung $y[n] = x[n] * h[n]$ zu Aufgabe 1.2 . . . . .	112
M1.4	Maximale Taktfrequenz für verschiedene Filterimplementierungen . . . . .	116
M2.1	Dämpfungen der verschiedenen Filter aus Aufgabe M2.1 bei $f_1 = 50$ Hz . . . . .	124
M2.2	Daten zu Aufgabe 2.11 . . . . .	150
M2.3	Ergebnisse zu Aufgabe M2.12 . . . . .	152
M3.1	Anzahl der Rechenoperationen für eine FFT in Abhängig von der Anzahl der Datenpunkte ( $f_S = 44,1$ kHz, Aufgabe 3.5) . . . . .	161

M5.1	Eckfrequenzen für den Filterentwurf zu Aufgabe 5.1d . . . . .	173
M5.2	Nullstellen des Zählerpolynoms . . . . .	176
M5.3	Hilfswerte für Filter F <sub>1</sub> (Aufgabe 5.3b) . . . . .	178
M5.4	Hilfswerte für Filter F <sub>2</sub> (Aufgabe 5.3c) . . . . .	180
M5.5	Musterlösung zu Aufgabe 5.8 . . . . .	190
M6.1	<b>10101</b> in verschiedenen Zahlenformaten (Aufgabe 6.1a) . . . . .	191
M6.2	Verschiedene Zahlen im Q2.6 Format (Aufgabe 6.1b) . . . . .	191
M6.3	Zweierkomplement und Requantisierung (Aufgabe 6.1c) . . . . .	192
M6.4	Wortlängenwachstum zu Aufgabe M6.3 . . . . .	195
M6.5	Filterkoeffizienten in verschiedenen Zahlenformaten zu M6.4 . . . . .	197
M6.6	Musterlösung zu Aufgabe 6.7 . . . . .	200
M7.1	Zusammenfassung der Ergebnisse aus Aufgabe M7.3 . . . . .	206
M8.1	Eigenschaften der Filterkaskade aus Aufgabe 8.7 . . . . .	215
M9.1	si-Dämpfungen für DAC ohne Oversampling (Abb. 9.3a) . . . . .	223
M9.2	si-Dämpfungen für DAC mit Oversampling (Abb. 9.3b) . . . . .	224
M9.3	Dämpfung der Images im zweistufigen Interpolator (Aufgabe 9.5) . . . . .	225
M10.1	Werte zu Abb. M10.2 . . . . .	232
M10.2	Werte zu Abb. M10.4 bzw. M10.5 . . . . .	232
M10.3	Ergebnisse zu Aufgabe 10.2 . . . . .	233

## Listings

3.1	Python Skript zu Aufgabe 3.3 (erster Ansatz, <code>DFT_MA_Filt.py</code> ) . . . . .	30
3.2	DFT mit Python ( <code>DFT_period_py.py</code> ) . . . . .	31
9.1	Upsampling mit Python . . . . .	87
9.2	Upsampling mit Matlab . . . . .	87
10.1	Interpolation / Resampling mit Python . . . . .	97
M1.1	Python-Implementierung der gcf - Funktion zu Aufgabe M1.1 . . . . .	110
M2.1	Python Listing zu M2.7f . . . . .	138
M2.2	Matlab Listing zu M2.7f . . . . .	138
M3.1	Korrigierter Code aus Notebook <code>DFT-MA_Filt_ML</code> zu Aufgabe 3.3 . . . . .	158
M3.2	Symbolische Integration mit SymPy . . . . .	164
M4.1	Matlab-Code zu Aufgabe 4.2 . . . . .	169

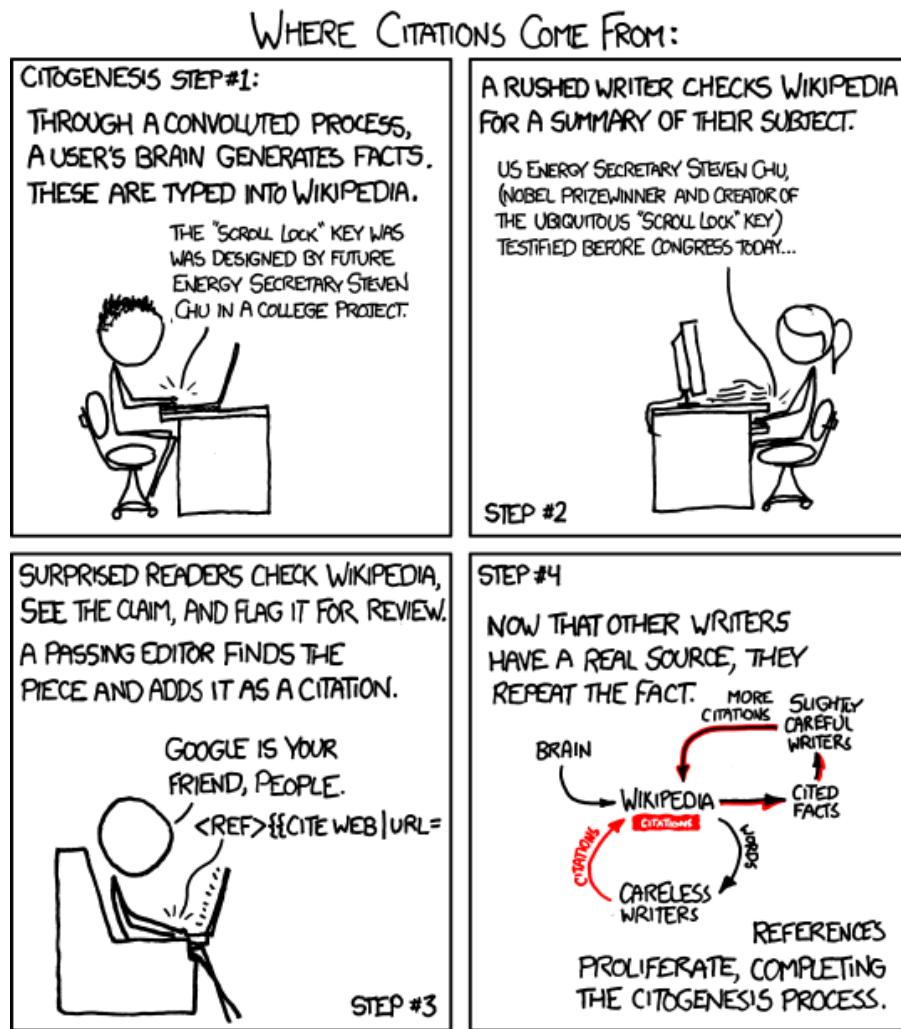


Abb. D.1: Richtig zitieren [<https://xkcd.com/978/>]

## Literatur

- [Gen11a] Norbert Geng. *Aufgabensammlung zu Signale und Systeme Teil 2: Zeitdiskrete Signale und Systeme*. 128 S. Hochschule München. Feb. 2011.
- [Gen11b] Norbert Geng. *Skriptum zu Signale und Systeme Teil 2: Zeitdiskrete Signale und Systeme*. 128 S. Hochschule München. Feb. 2011.
- [Ger07] Gerhard Doblinger. *Zeitdiskrete Signale und Systeme - Eine Einführung in die grundlegenden Methoden der digitalen Signalverarbeitung*. 1. Aufl. 230 S., 24,90 €. Wilburgstetten: J. Schlembach Fachverlag, 2007, S. 230. URL: <https://www.nt.tuwien.ac.at/about-us/staff/gerhard-doblinger/>.
- [Har08] Fred Harris. *WINDOWS: Finite Aperture Effects and Applications in Signal Processing*. 2008.
- [HRS02] G. Heinzel, A. Rüdiger und R. Schilling. *Spectrum and spectral density estimation by the Discrete Fourier transform (DFT), including a comprehensive list of window functions and some new flat-top windows*. Forschungsber. Max-Planck-Institut für Gravitationsphysik (Albert-Einstein-Institut), 15. Feb. 2002. URL: <http://edoc.mpg.de/395068>.
- [KK18] Karl-Dirk Kammeyer und Kristian Kroschel. *Digitale Signalverarbeitung: Filterung und Spektralanalyse mit MATLAB-Übungen*. 9. Aufl. 628 S., 45 €. Wiesbaden: Springer-Vieweg, 2018.
- [Leh] Lehrstuhl für Nachrichtentechnik an der TU München. *Ein Lerntutorial für Nachrichtentechnik im World Wide Web*. URL: <http://www.lntwww.de/>.
- [Mer10] Alfred Mertins. *Signaltheorie: Grundlagen der Signalbeschreibung, Filterbänke, Wavelets, Zeit-Frequenz-Analyse, Parameter- und Signalschätzung*. 2. Aufl. 393 S., 34,95 €. Vieweg+Teubner Verlag, 2010. ISBN: 9783834807373.
- [Mey07] Uwe Meyer-Baese. *Digital Signal Processing with Field Programmable Gate Arrays*. 3. Aufl. 774 S., ca. 90 €, DAS Standardwerk für schnelle digitale Signalverarbeitung auf FPGAs. Berlin: Springer, 2007.
- [Orf10] Sophocles J. Orfanidis. *Introduction to Signal Processing*. 2. Aufl. 783 S., Prentice Hall, Inc., 2010. URL: <http://www.ece.rutgers.edu/~orfanidi/intro2sp>.
- [OS10] Alan V. Oppenheim und Ronald W. Schafer. *Discrete-Time Signal Processing*. 3. Aufl. 1120 S., ab 50 €. Pearson, 2010.
- [Smi11] Julius O. Smith. *Spectral Audio Signal Processing*. W3K Publishing, März 2011. URL: <https://www.dsprelated.com/freebooks/sasp/>.
- [Smi99] Steven W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. 2. Aufl. <http://www.DSPguide.com>. San Diego, USA: California Technical Publishing, 1999.
- [Wer08] Martin Werner. *Signale und Systeme*. 3. Aufl. 404 S., 29,95 €. Vieweg + Teubner, 2008.
- [Wer11] Martin Werner. *Digitale Signalverarbeitung mit Matlab: Grundkurs*. 5. Aufl. 332 S., 39,90 €. Wiesbaden: Vieweg + Teubner, 2011.
- [ZA11] U. Zölzer und X. Amatriain. *DAFX: digital audio effects*. 2. Aufl. 624 S., 85 €. Wiley, 2011. ISBN: 9780471490784.

- [Zöl05] Udo Zölzer. *Digitale Audiosignalverarbeitung*. 3. Aufl. 328 S., 44,90 €. Wiesbaden, Germany: Teubner, 2005.