

4 Planungs- und Entwurfsphase für die digitalen IIR-Filter

Dieses Kapitel zeigt den Entwurf des digitalen IIR-Filters für das neu zu entwickelnde Kanalmodell. Das Verhalten des IIR-Filters soll dem des bestehenden Kanalmodells gleichen. Des Weiteren wird eine Simulation der unterschiedlichen Filter in Simulink vorgestellt. Nach der Simulation folgt ein Test zur Berechnungszeit eines realen IIR-Filters mit einem Mikrocontroller.

4.1 Umrechnung der analogen RC-Tiefpassfilter in digitale IIR-Filter

Es gibt mehrere Verfahren zur Umrechnung der analogen Filterkoeffizienten in digitale Koeffizienten. Zu ihnen gehören unter anderen die "bilineare Transformation", die eine Nachbildung des analogen Frequenzganges anstrebt, die "impulsinvariante Nachbildung", die das Verhalten im Zeitbereich approximiert, die "Fourierapproximation" usw.

In diesem Kapitel werden die Verfahren "bilineare Transformation" und "impulsinvariante Nachbildung" anhand des TP 1. Ordnung mit einem Eingangssignal von 64Kbit/s bzw. 32KHz-Rechteck und einem Oversamplingfaktor von 100 ($f_A = 6,4\text{MHz}$) näher beschrieben. Es folgt in Kap. 4.1.3 eine Umrechnung für höhere Ordnungszahlen mit Matlab.

4.1.1 Mathematische Bestimmung der digitalen Koeffizienten nach der "bilinearen Transformation"

Ziel dieser Methode ist, die Koeffizienten für ein digitales Filter zu bestimmen, welches den Frequenzgang eines analogen Filters möglichst exakt nachbildet (*nachzubildendes Frequenzverhalten*). Ausgangspunkt für die Transformation ist ein analoges Filter mit der gegebenen Übertragungsfunktion $G(s)$. Das Ergebnis ist dann eine Übertragungsfunktion $H(z)$ eines diskreten Filters mit dem Frequenzverhalten des analogen Filters.

Aus der Herleitung der z-Transformation (s.a. [4]) folgt:

Substitution mit e^{sT_A} stellt den Zusammenhang zwischen der analogen s-Ebene und der diskreten (*digitalen*) z-Ebene her.

Die komplexe Frequenz s der s-Ebene wird auf die Abtastfrequenz f_A bezogen:

à mit $s' = \frac{s}{f_A}$ folgt die Beziehung $e^{s'} = z$.

Bei der bilinearen Transformation wird die Übertragungsfunktion des analogen Referenzfilters auf f_A normiert. So ergibt sich die Übertragungsfunktion mit der Variablen z , indem die Substitution $s' = \ln(z)$ durchgeführt wird. Da dies nicht rational ist, wird die dazugehörige Reihenentwicklung herangezogen und nach dem 1. Glied abgebrochen:
 $s' = 2 \cdot \frac{z-1}{z+1}$. Mit Hilfe dieser Substitution entsteht dann aus der Übertragungsfunktion des analogen Referenzfilters eine realisierbare Übertragungsfunktion $H(z)$ eines diskreten Filters, welches das Frequenzverhalten approximiert. Für das nachzubildende Kanalmodell mit $T_A = 156,25ns$ und $T = 1,59155\mu s$ folgt für die Koeffizienten nach [4]:

$$A_0 = \frac{T_A/2T}{1 + T_A/2T} \quad B_1 = \frac{T_A/2T - 1}{T_A/2T + 1} = -0,90642$$

$$A_0 = A_1 = 0,0468$$

Daraus folgt für $H(z)$:

$$H(z) = \frac{0,0468 + 0,0468 \cdot z^{-1}}{1 - 0,90642 \cdot z^{-1}}$$

4.1.2 Berechnung nach der "impulsinvarianten Nachbildung"

Ziel dieser Methode ist die Bestimmung der Koeffizienten für ein digitales Filter mit der Übertragungsfunktion $H(z)$, welches das Zeitverhalten eines analogen Filters, mit der Übertragungsfunktion $G(s)$, möglichst exakt nachbildet (*nachzubildendes Zeitverhalten*). Ausgangspunkt für die Nachbildung ist wieder ein analoges Filter mit gegebener Übertragungsfunktion bzw. bekannter Impulsantwort $g(t)$.

Bsp.: RC-Tiefpass 1. Ordnung (Abb. 4.2)

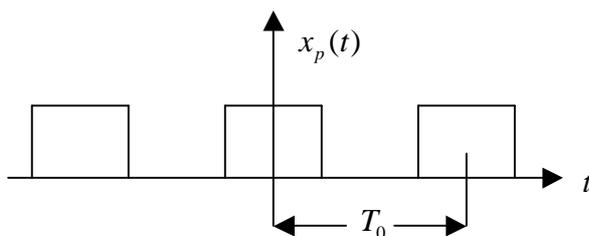


Abb. 4.1:
Eingangssignal z.B. mit $T_0 = 31,25\mu s$

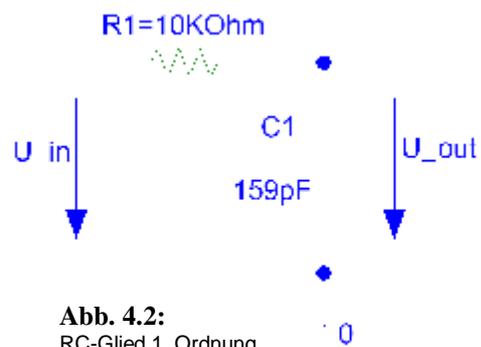


Abb. 4.2:
RC-Glied 1. Ordnung

Übertragungsfunktion

$$G(s) = \frac{1}{1+sT} \text{ mit } T = R \cdot C = 1,59155\mu\text{s}$$

Durch Koeffizientenvergleich von $G(s)$ mit $H(z)$ folgt mit einem Oversamplingfaktor von 100 nach [4]:

$$A_0 = b = \frac{T_A}{T} \text{ und } B_1 = -a = -e^{-\frac{T_A}{T}} \text{ mit}$$

$$T = 1,59155\mu\text{s} \text{ und } T_A = 156,25\text{ns} (6,4\text{MHz}).$$

à $A_0 = 0,09818$, $B_1 = -0,9065$ und es gilt für $H(z)$:

$$H(z) = \frac{0,09818}{1 - 0,9065 \cdot z^{-1}}$$

Eine Kontrolle der berechneten Übertragungsfunktion mit Matlab folgt in Kap. 4.1.3 und bestätigt (nach Erweiterung um einen bestimmten Faktor) die ermittelten Werte.

4.1.3 Bestimmung der digitalen Koeffizienten für die "Ti-TP Kette" mit Hilfe von Matlab

Da die Berechnung der Koeffizienten für höhere OZ sehr umfangreich wird (s.a. [7]), wurde zur Berechnung ab der 2. Ordnung das Programm Matlab zur Hilfe genommen. Die Befehle "bilinear(A,B,fa)" und "impinvar(A,B,fa)" führen die gewünschte Rechnung durch. Da jedoch der Befehl "bilinear" bei der Rechnung ab der 3. Ordnung eine Fehlermeldung verursacht, wurde der Befehl "c2dm(A,B,Ta,'tustin')" (continuous to discrete method) benutzt, der die gleiche "bilinare Transformation" durchführt. Ein Vergleich der Koeffizienten, durch die unterschiedlichen Befehle ermittelt, zeigte trotz der Fehlermeldung keine Abweichung bis einschließlich 8. Ordnung.

Gefordert ist ein Oversamplingfaktor von 100, d.h. $f_A = 64 \frac{\text{KBit}}{\text{s}} \cdot 100 = 6,4\text{MHz}$.

Berechnete analoge Ü-Funktion 2.Ordnung:

$$G(s) = 1/((sT)^2 + 3sT + 1) \text{ mit } T = 1,59155\mu\text{s}$$

Umrechnung nach der "bilinaren Transformation":

Ergebnis:

$$\begin{aligned} [Ab] &= [0.0021 \quad 0.0042 \quad 0.0021] \\ [Bb] &= [1 \quad -1.7354 \quad 0.7438] \end{aligned}$$

Impulsantwort

$$g(t) = \frac{1}{T} \cdot e^{-\frac{t}{T}} \cdot e(t) \text{ für } t \geq 0$$

$$g(n) = T_A \cdot g(n \cdot T_A) \text{ mit } T_A = 156,25\text{ns}$$

$$g(n) = \frac{T_A}{T} \cdot e^{-\frac{n \cdot T_A}{T}} \text{ für } n \geq 0$$

$$g(n) = b \cdot (e^{-k})^n = b \cdot a^n$$

$$H(z) = b \cdot \frac{1}{1 - a \cdot z^{-1}}$$

Umrechnung nach der "impulsinvarianten Methode":

Ergebnis:

$$\begin{aligned} [Ai] &= 1e-13 * [0 \quad 0.2111] \\ [Bi] &= 1e-11 * [0.2533 \quad -0.4399 \quad 0.1887] \end{aligned}$$

Hinsichtlich einer einfachen Programmiersvorschrift und einer möglichst schnellen Berechnung, sind so kleine Werte, wie sie nach der impulsinvarianten Methode bestimmt werden, zu vermeiden (*float schneller als double u.a.*). Daher sind nur die nach der "bilinearen Methode" ermittelten digitalen Übertragungsfunktionen dargestellt. Die ausführliche Ergebnistabelle, auch für die Berechnungsergebnisse nach der "impulsinvarianten Methode" ermittelt, ist im Anhang D.1 beigelegt.

1. Ordnung :

$$G(z) = \frac{A_0 + A_1 \cdot z^{-1}}{1 + B_1 \cdot z^{-1}} = \frac{0,0468 + 0,0468 \cdot z^{-1}}{1 - 0,9064 \cdot z^{-1}}$$

2. Ordnung :

$$G(z) = \frac{A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2}}{1 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2}} = \frac{0,0021 + 0,0042 \cdot z^{-1} + 0,0021 \cdot z^{-2}}{1 - 1,7354 \cdot z^{-1} + 0,7438 \cdot z^{-2}}$$

3. Ordnung :

$$G(z) = \frac{A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2} + A_3 \cdot z^{-3}}{1 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2} + B_3 \cdot z^{-3}} = \frac{1e^{-3} \cdot [0,0939 + 0,2816 \cdot z^{-1} + 0,2816 \cdot z^{-2} + 0,0939 \cdot z^{-3}]}{1 - 2,564 \cdot z^{-1} + 2,1749 \cdot z^{-2} - 0,6102 \cdot z^{-3}}$$

4. Ordnung :

$$G(z) = \frac{1e^{-4} \cdot [0,042 + 0,1682 \cdot z^{-1} + 0,2523 \cdot z^{-2} + 0,1682 \cdot z^{-3} + 0,042 \cdot z^{-4}]}{1 - 3,3925 \cdot z^{-1} + 4,2925 \cdot z^{-2} - 2,4006 \cdot z^{-3} + 0,5006 \cdot z^{-4}}$$

5. Ordnung :

$$G(z) = \frac{1e^{-5} \cdot [0,0188 + 0,0942 \cdot z^{-1} + 0,1883 \cdot z^{-2} + 0,1883 \cdot z^{-3} + 0,0942 \cdot z^{-4} + 0,0188 \cdot z^{-5}]}{1 - 4,221 \cdot z^{-1} + 7,0966 \cdot z^{-2} - 5,9399 \cdot z^{-3} + 2,4751 \cdot z^{-4} - 0,4107 \cdot z^{-5}}$$

6. Ordnung :

$$G(z) = \frac{1e^{-6} \cdot [0,0084 + 0,0506 \cdot z^{-1} + 0,1265 \cdot z^{-2} + 0,1687 \cdot z^{-3} + 0,1265 \cdot z^{-4} + 0,0506 \cdot z^{-5} + 0,0084 \cdot z^{-6}]}{1 - 5,0495 \cdot z^{-1} + 10,5871 \cdot z^{-2} - 11,7969 \cdot z^{-3} + 7,3677 \cdot z^{-4} - 2,4454 \cdot z^{-5} + 0,337 \cdot z^{-6}}$$

7. Ordnung :

$$G(z) = \frac{1e^{-7} \cdot [0,0038 + 0,0264 \cdot z^{-1} + 0,0793 \cdot z^{-2} + 0,1322 \cdot z^{-3} + 0,1322 \cdot z^{-4} + 0,0793 \cdot z^{-5} + 0,0264 \cdot z^{-6} + 0,0038 \cdot z^{-7}]}{1 - 5,8781 \cdot z^{-1} + 14,764 \cdot z^{-2} - 20,5402 \cdot z^{-3} + 17,0942 \cdot z^{-4} - 8,51 \cdot z^{-5} + 2,3465 \cdot z^{-6} - 0,2764 \cdot z^{-7}}$$

8. Ordnung :

$$G(z) = \frac{1e^{-8} \cdot [0,0017 + 0,0135 \cdot z^{-1} + 0,0474 \cdot z^{-2} + 0,0947 \cdot z^{-3} + 0,1184 \cdot z^{-4} + 0,0947 \cdot z^{-5} + 0,0474 \cdot z^{-6} + 0,0135 \cdot z^{-7} + 0,0017 \cdot z^{-8}]}{1 - 6,7066 \cdot z^{-1} + 19,6274 \cdot z^{-2} - 32,7387 \cdot z^{-3} + 34,0413 \cdot z^{-4} - 22,594 \cdot z^{-5} + 9,348 \cdot z^{-6} - 2,2043 \cdot z^{-7} + 0,2268 \cdot z^{-8}}$$

Dies sind die Übertragungskoeffizienten mit den entsprechenden Koeffizienten, die für ein digitales Modell zu programmieren sind, wenn ein Oversampling-Faktor von 100 für alle Ordnungszahlen erreicht wird.

4.2 Transformation der analogen, aktiven Hochpassfilter in digitale

IIR-Filter

Zunächst wurden Tests mit einem Oversampling-Faktor von 60 für OZ2 durchgeführt. Damit ergibt sich für die Abtastfrequenz am Filter $f_A = 64 \frac{\text{KBit}}{\text{s}} \cdot 60 = 6,4 \text{MHz}$. Diese Werte wurden in Matlab zur Berechnung der HP-Übertragungsfunktion nach der "bilinearen Transformation" angegeben.

Ergebnis:

$$\begin{aligned} [\text{Ab}] &= [0.9997 \quad -0.9997] \\ [\text{Bb}] &= [1 \quad -0.9993] \end{aligned}$$

Umrechnung nach der impulsinvarianten Methode:

Ergebnis:

$$\begin{aligned} [\text{Ai}] &= 1e-6 * [-0.2603 \quad -0.0001] \\ [\text{Bi}] &= 1e-3 * [0.398 \quad -0.03977] \end{aligned}$$

Berechnete Koeffizienten für die Nachbildung der HP-Einsteckkarte mit einer Abtast-rate von 3,84MHz am Filter:

Ordnungszahl	Ü-Funktion y/X	Bei Samplerate = 3,84MHz		
		Koeffizienten T=398µs		
		c2dm(A,B,ta,'tustin')	bilinear(A,B,fa)	impulsinvariant(A,B,fa)
1	fg = 400Hz T=398µs [1 0] oder evtl. doch [1] [1 1]	A = [0.9997 -0.9997] B = [1 -0.9993]	A = [0.9997 -0.9997] B = [1 -0.9993]	1e-6 *[-0.2603 -0.0001] 1e-3*[0.3980 -0.3977]
1	fg = 2KHz T=79,52µs [1 0] [1 1]	A = [0.9984 -0.9984] B = [1 -0.9967]	A = [0.9984 -0.9984] B = [1 -0.9967]	1e-6 *[-0.2604 -0.0] 1e-4*[0.7952 -0.7926]
1	fg = 3KHz T=53,03µs [1 0] [1 1]	A = [0.9976 -0.9976] B = [1 -0.9951]	A = [0.9976 -0.9976] B = [1 -0.9951]	1e-6 *[-0.2604 -0.0] 1e-4*[0.5303 -0.5277]
2	fg = 400Hz			
1. CR-Glied	T=701,8µs [1 0] [1 1]	A = [0.9998 -0.9998] B = [1 -0.9996]	A = [0.9998 -0.9998] B = [1 -0.9996]	1e-6 *[-0.2602 -0.0002] 1e-3*[0.7018 -0.7015]
2. CR-Glied	T=701,6µs [1 0] [1 1]	A = [0.9998 -0.9998] B = [1 -0.9996]	A = [0.9998 -0.9998] B = [1 -0.9996]	1e-6 *[-0.2602 -0.0002] 1e-3*[0.7016 -0.7013]
2	fg = 2KHz			
1. CR-Glied	T=123,67µs [1 0] [1 1]	A = [0.9989 -0.9989] B = [1 -0.9979]	A = [0.9989 -0.9989] B = [1 -0.9979]	1e-6 *[-0.2604 -0.0] 1e-3*[0.1237 -0.1234]
2. CR-Glied	T=123,78µs [1 0] [1 1]	A = [0.9989 -0.9989] B = [1 -0.9979]	A = [0.9989 -0.9989] B = [1 -0.9979]	1e-6 *[-0.2604 -0.0] 1e-3*[0.1238 -0.1235]
2	fg = 3KHz			
1. CR-Glied	T=82,48µs [1 0] [1 1]	A = [0.9984 -0.9984] B = [1 -0.9968]	A = [0.9984 -0.9984] B = [1 -0.9968]	1e-6 *[-0.2604 -0.0] 1e-4*[0.8248 -0.8222]
2. CR-Glied	T=82,46µs [1 0] [1 1]	A = [0.9984 -0.9984] B = [1 -0.9968]	A = [0.9984 -0.9984] B = [1 -0.9968]	1e-6 *[-0.2604 -0.0] 1e-4*[0.8246 -0.8220]

Tab. 4.1:
Ermittelte Übertragungsfunktionen für die digitalen HP-Einsteckkarte

4.3 Matlab/Simulink-Simulation

Die zur Simulation verwendeten Simulink-Blöcke für die digitalen Filter beinhaltet die in Kap. 4.5 vorgestellte Netzwerk für ein kanonisches Netzwerk. In diesem Kapitel wird das Zeitverhalten der analogen (*Transfer Fcn*) und der digitalen Filter verglichen.

Eine Simulation in Simulink bestätigt die identische Beeinflussung des Eingangssignals durch das Kanalmodell, unabhängig davon, ob analoger TP mit den berechneten Koeffizienten (*Kap. 3.3*) oder digitaler TP mit den transformierten Koeffizienten.

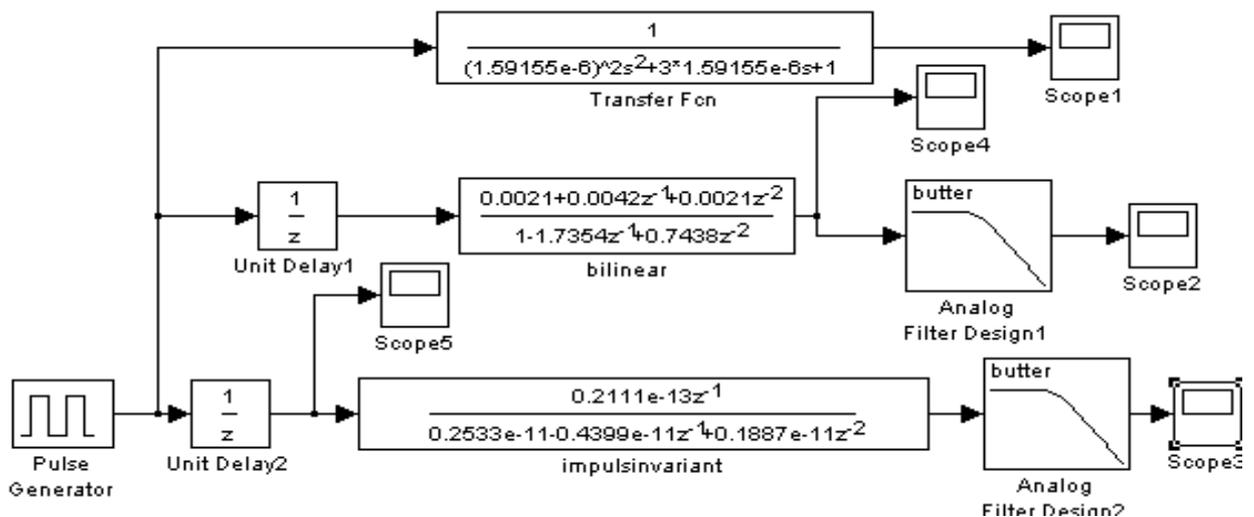


Abb. 4.3:

Simulink-Simulation des analogen TP und der ermittelten digitalen Tiefpässe für OZ2 mit $f_A = 6,4\text{MHz}$

Ergebnis der Simulation ist, dass ein Filter mit 100-fach Oversampling am Filter sehr gut approximiert wird. Gerade ausreichend zur Nachbildung ist mit der Simulation ein Faktor von 13-fach Oversampling (*s.a. Abb 4.4*).

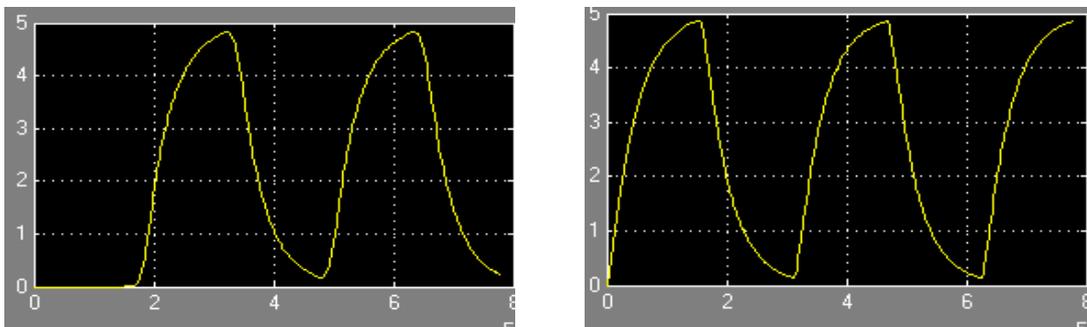


Abb. 4.4:

Rechteckantwort

4.4 Digitalfilter allgemein und entsprechende Theorie zur Realisierung

Dieses Kapitel soll einen Eindruck in die Funktionsweise von Digitalfiltern vermitteln. Bei der Betrachtung von Digitalfiltern ist immer auf die richtige Bewertung der Frequenzachse zu achten, dazu ein einführendes Beispiel: Wird z.B. ein Tiefpass mit der Übertragungsfunktion $Y_{[Z]} = 0,5 \cdot Z^{-2} \cdot X_{[Z]} + X_{[Z]} \cdot Z^{-1} + 0,5 \cdot X_{[Z]}$ betrachtet, wird der normierte Amplitudengang $G(j\Omega)$ durch die bezogene Frequenz F beschrieben.

$$F = \frac{f}{f_A} \text{ mit } f_A : \text{Abtastfrequenz, } f : \text{Signalfrequenz; beachte } \Omega = 2\pi F$$

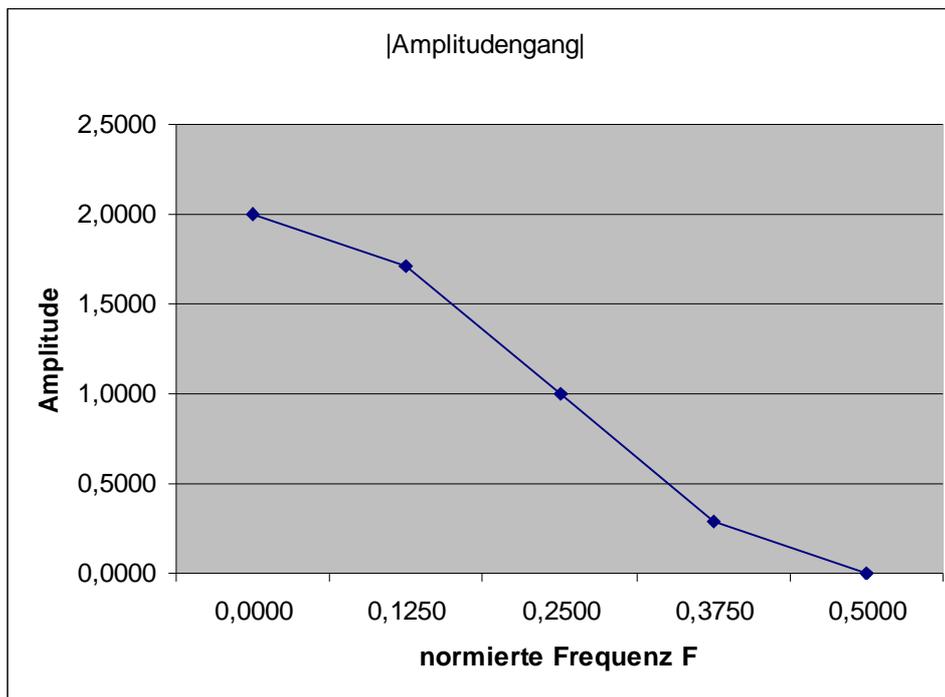


Abb. 4.5:
Amplitudengang
digitaler TP

4.4.1 Bestimmung des Amplitudenganges für die ermittelten digitalen TP-Filter

Zur Bestätigung der in Kap. 4.1 ermittelten Koeffizienten für das digitale Filter, zeigt dieser Teil die entsprechenden, mit Matlab ermittelten Amplitudengänge. Zusätzlich werden die ermittelten Verläufe mit denen des analogen Modells verglichen, um den approximierten Verlauf zu verdeutlichen. Die Angaben zur Grenzfrequenz in Abb. 4.6 und Abb. 4.7 sind auf Grund des sehr großen Darstellungsbereichs der Abzisse nur eine ungefähre Angabe. In Abb. 4.8 wird daher ein Ausschnitt der Abb. 4.7 von 10KHz bis 100KHz gezeigt, woraus die exakte Übereinstimmung der Grenzfrequenzen hervorgeht. Für die Nachbildung des Kanalmodells ist der Frequenzbereich unterhalb von 1MHz

entscheidend, da die Frequenzkomponenten oberhalb von 1MHz nicht signifikant zur Signalbildung beitragen.

1. Ordnung :

$$G(z) = \frac{A_0 + A_1 \cdot z^{-1}}{1 + B_1 \cdot z^{-1}} = \frac{0,0468 + 0,0468 \cdot z^{-1}}{1 - 0,9064 \cdot z^{-1}}$$

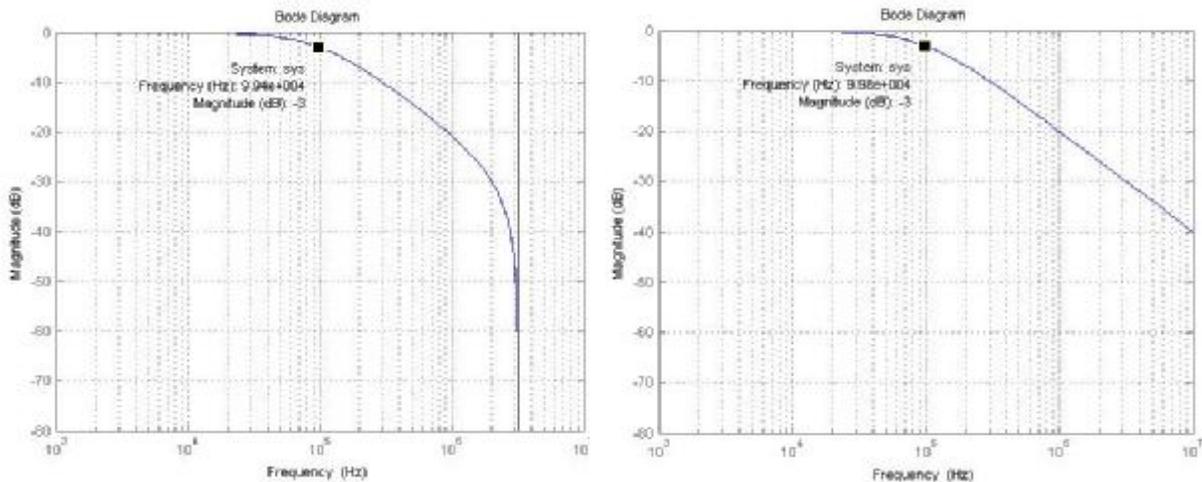


Abb. 4.6:
Amplitudengang digitaler TP OZ1 (links) und das analoge Filter OZ1 (rechts)

Anhand der Markierungen in Abb. 4.6 ist gut zu erkennen, dass die Grenzfrequenz beim digitalen Filter im linken Bild (99,4KHz) fast der des analogen (99,8KHz) auf der rechten gleicht. Beim digitalen Filter ist auch die, durch die Abtastung verursachte, Unterdrückung der Frequenzen um $f_A/2$ herum zu erkennen. Der Verlauf des analogen Filters wird durch die digitale Nachbildung bis zur Frequenz von ca. 1MHz sehr gut approximiert. Bis zu welcher Frequenz letztendlich eine exakte Nachbildung benötigt wird, ist stark vom Eingangssignal abhängig (s.a. Kap. 2.2) und am realen Modell zu untersuchen. Beim später zu realisierenden Modell ist zu beachten, dass die durch Abtastung verursachte, periodische Fortsetzung des Signalspektrums nicht aus diesen Abbildungen hervorgeht.

2. Ordnung :

$$G(z) = \frac{A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2}}{1 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2}} = \frac{0,0021 + 0,0042 \cdot z^{-1} + 0,0021 \cdot z^{-2}}{1 - 1,7354 \cdot z^{-1} + 0,7438 \cdot z^{-2}}$$

Um die Nachbildung des Amplitudenganges zu kontrollieren, wurde wieder der Amplitudengang der digitalen Übertragungsfunktion mit dem der analogen Übertragungsfunktion in Matlab verglichen.

Abb. 4.7 auf der nächsten Seite zeigt im linken Bild den Amplitudengang für das digitale Filter und im rechten Bild den nachzubildenden Verlauf des analogen Filters.

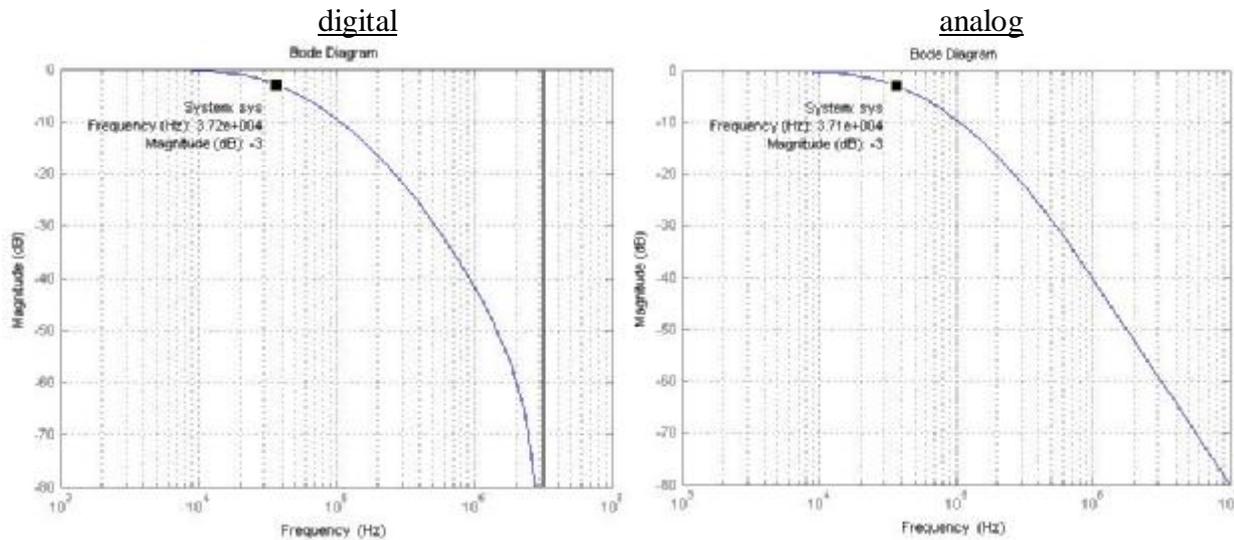


Abb. 4.7:
Amplitudengang digitaler TP OZ2 (*links*) und analoger TP zum Vergleich (*rechts*)

Deutlich ist in Abb. 4.7 links wieder die, durch die Abtastung entstandene, Unterdrückung des Signalspektrums bei $f_A/2$ zu erkennen. Bei einem Vergleich der beiden Bilder lässt sich das ähnliche Verhalten bereits erkennen. Da die Nachbildung des Amplitudenganges im unteren Frequenzbereich entscheidend ist, wurde in Abb. 4.8 der Bereich bis 100KHz betrachtet und die Approximationsgüte eingeschätzt.

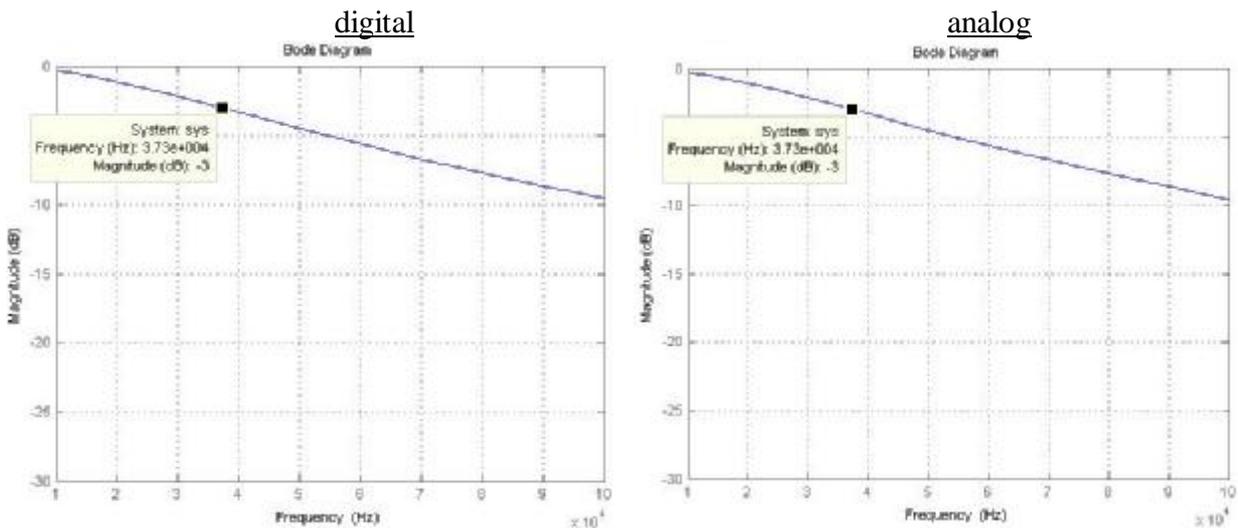


Abb. 4.8:
Amplitudengang digitaler TP OZ2 links und analoger TP rechts (*Ausschnitt*)

In Abb. 4.8 links ist die digitale Nachbildung der analogen Übertragungsfunktion aus Abb. 4.7 rechts im Bode-Diagramm dargestellt. Klar ist die exakte Approximation bis 100KHz

zu erkennen. Bis 1MHz wird das analoge Filter OZ2 immer noch sehr gut approximiert. Da dieser Sachverhalt ebenso für die höheren Ordnungszahlen gilt, wird die Betrachtung für höhere Ordnungszahlen nicht mehr mit dargestellt.

4.4.2 Bestimmung der Grenzfrequenzen für die digitalen TP-Filter

Die Bestimmung der 3dB-Grenzfrequenzen für die ermittelten digitalen Filter wurde in Matlab anhand des Amplitudenganges durchgeführt. Dazu wurde der gewünschte Frequenzbereich gezoomt. Aufgrund des beschränkten Anzeigeformats wurde mit einer dreistelligen Anzeige des Signalpegels gearbeitet. Dies ist absolut ausreichend, da solche geringe Abweichungen beim realen Modell nicht von Bedeutung sind.

Im Matlab Bode-Diagramm ermittelte 3dB Grenzfrequenzen (in KHz)		
"TI-TP Kette Ordnungszahl"	MatLab 7.0 Bodediagramm	
	digitale TP-Filter	analoge TP-Filter
1	99,4	99,8
2	37,3	37,3
3	19,4	19,4
4	11,8	11,8
5	7,95	7,95
6	5,7	5,71
7	4,29	4,3
8	3,34	3,34

Tab. 4.2:
Ermittelte 3dB Grenzfrequenzen für die analogen und digitalen Tiefpassfilter

Da die in der Tabelle 4.2 gezeigten Grenzfrequenzen für analoge und digitale Tiefpassfilter fast identisch sind, bestätigt dies noch einmal die gewünschte Nachbildung des Frequenzverhaltens. Da sich die Abtastfrequenz von 6,4MHz auf den Verlauf auswirkt.

4.4.3 Polstellenbestimmung für die digitalen TP-Filter

Ein kausales, diskretes LTI-System ist stabil, wenn sämtliche Pole der Übertragungsfunktion im Inneren des Einheitskreises liegen [3]. Wie bei den analogen Polen (Kap. 3.5.1) wird der Nenner der Übertragungsfunktion auf 0 gesetzt und nach z aufgelöst.

1. Ordnung :

$$G(z) = \frac{A_0 + A_1 \cdot z^{-1}}{1 + B_1 \cdot z^{-1}} = \frac{0,0468 + 0,0468 \cdot z^{-1}}{1 - 0,9064 \cdot z^{-1}}$$

$$0 = 1 - 0,9064 \cdot z^{-1}$$

$$z = 0,9064$$

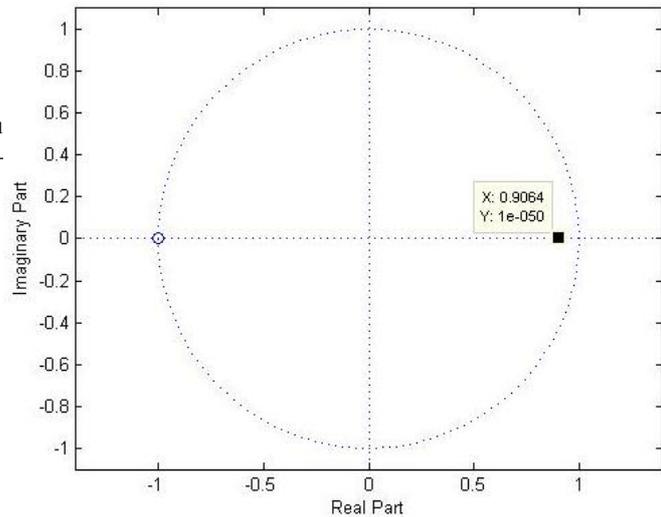


Abb. 4.9:
Pol-/Nullstellenplan für den digitalen TP OZ1

2. Ordnung :

$$G(z) = \frac{A_0 + A_1 \cdot z^{-1} + A_2 \cdot z^{-2}}{1 + B_1 \cdot z^{-1} + B_2 \cdot z^{-2}} = \frac{0,0021 + 0,0042 \cdot z^{-1} + 0,0021 \cdot z^{-2}}{1 - 1,7354 \cdot z^{-1} + 0,7438 \cdot z^{-2}} \rightarrow z_1 = 0,9632 \text{ und } z_2 = 0,7722$$

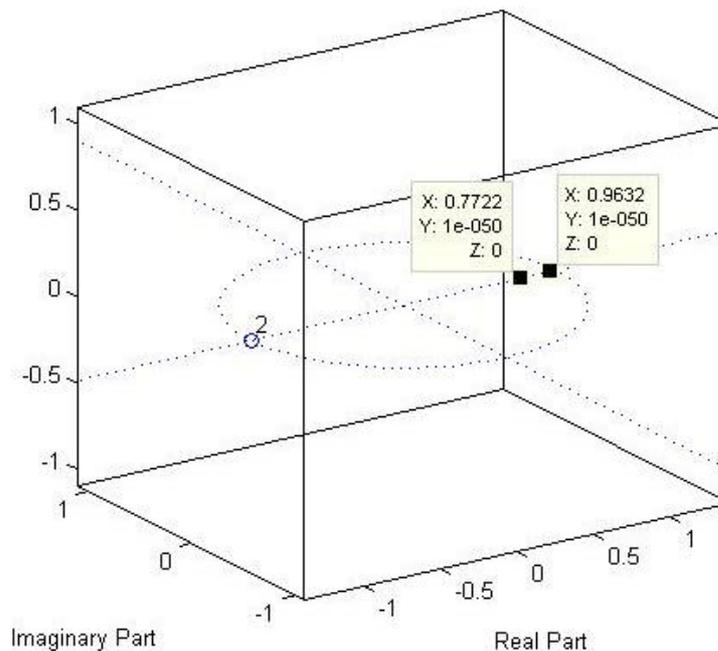


Abb. 4.10:
Pol-/Nullstellenplan für OZ2 als 3-dimensionale Grafik mit Einheitskreis

Es folgen die Polstellen für die digitalen TP-Filter bis zur 6. Ordnung:

Ordnungszahl	Ü-Funktion 1/X		Polstellen -X/T mit T=10µs für die direkte Umrechnung
1	A = [0.0468 0.0468] B = [1 -0.9064]	1.Pol	[0.9064]
2	A = [0.0021 0.0042 0.0021] B = [1 -1.7354 0.7438]	1.Pol 2.Pol	[0.9631 0.7723]
3	1e-3*[0.0939 0.2816 0.2816 0.0939] B = [1 -2.5640 2.1749 -0.6102]	1.Pol 2.Pol 3.Pol	[0.9823 0.8561 0.7256]
4	A = 1e-4*[0.0420 0.1682 0.2523 0.1682 0.0420] B = [1 -3.3925 4.2925 -2.4006 0.5006]	1.Pol 2.Pol 3.Pol 4.Pol	[1 0.8672 0.8287 0.6965] <i>Der 1. Pol liegt bereits auf dem Einheitskreis</i>
5	A = 1e-5*[0.0188 0.0942 0.1883 0.1883 0.0942 0.0188] B = [1 -4.2210 7.0966 -5.9399 2.4751 -0.4107]	1.Pol 2.Pol 3.Pol 4.Pol 5.Pol	[(0.9989 + 0.0717i) (0.9989 - 0.0717i) (0.7815 + 0.0976i) (0.7815 - 0.0976i) 0.6602] <i>ab hier werden die Pole komplex</i> <i>Dies ist der einzige reelle Pol</i>
6	A = 1e-6*[0.0084 0.0506 0.1265 0.1687 0.1265 0.0506 0.0084] B = [1 -5.0495 10.5871 -11.7969 7.3677 -2.4454 0.3370]	1.Pol 2.Pol 3.Pol 4.Pol 5.Pol 6.Pol	[1 (0.9451 + 0.0754i) (0.9451 + 0.0754i) (0.7367 + 0.0621i) (0.7367 - 0.0621i) 0.6858] Der Pol-/Nullstellenplan für diese Ordnung ist in Abb. 4.11 dargestellt

Tab. 4.3:
Polstellenplan für die ermittelten digitalen Filter

Ab der 5. Ordnung werden die Pole mit der Matlab-Berechnung komplex. Zu erkennen ist in Abbildung 4.12 die Abweichungen der Pole (*als Kreuz gekennzeichnet*) von der reellen Achse bei der 6. Ordnung.

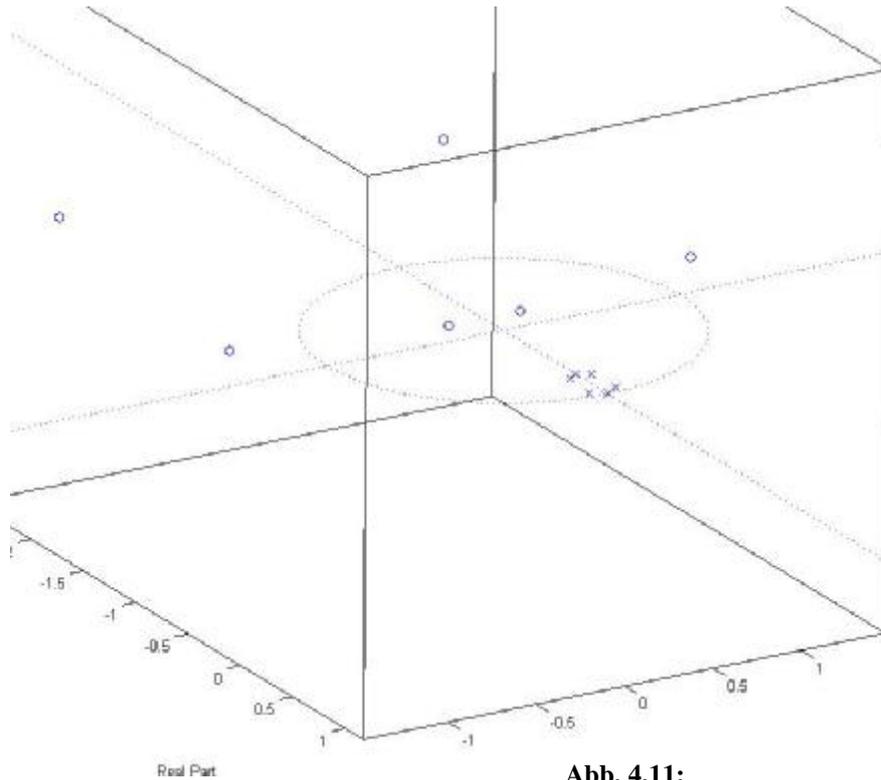


Abb. 4.11:
Polstellenplan für den ermittelten digitalen Filter 6. Ordnung

Es handelt sich bei den komplexen Polen um Koeffizientenpaare, die innerhalb der Dreiecksumrandung (siehe Abb 4.12) liegen. Diese führen auf stabile kausale Systeme 2. Ordnung [9]. In der Abbildung sind die Gebiete eingetragen, in denen sich einerseits konjugiert komplexe und andererseits reelle Pole ergeben.

Die Abweichung der Pole von der reellen Achse bei den höheren Ordnungszahlen ist also nicht auf Ungenauigkeiten der Matlab Berechnung zurückzuführen.

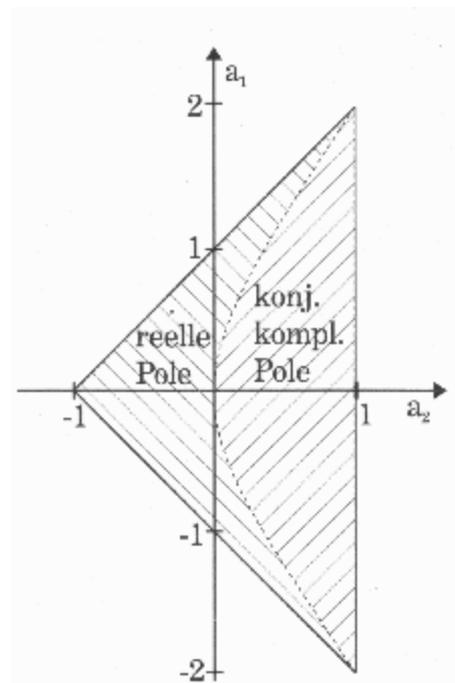


Abb. 4.12:
Koeffizientenkonstellation für stabile, kausale, reellwertige Systeme 2. Ordnung [9]

4.5 Theorie zur Filterrealisierung

Als Grundlage für eine algorithmische Beschreibung bzw. Realisierung eines digitalen Filters wird dessen Übertragungsfunktion herangezogen [4],

z.B. für 2. Ordnung:
$$H(z) = \frac{A_0 + A_1 z^{-1} + A_2 z^{-2}}{1 + B_1 z^{-1} + B_2 z^{-2}}$$

Betrachtet wird das dazugehörige Netzwerk mit einem Minimum an Verzögerungsgliedern (*kanonische Form: N Verzögerungsglieder bei Nennerpolynom von Grade N*). Zum besseren Verständnis wurde die Übertragungsfunktion $H(z)$ in eine transversale Übertragungsfunktion

$$H_1(z) = A_0 + A_1 z^{-1} + A_2 z^{-2} = \frac{Y_1(z)}{X_1(z)}$$

und eine rein rekursive

$$H_2(z) = \frac{1}{1 + B_1 z^{-1} + B_2 z^{-2}} = \frac{Y_2(z)}{X_2(z)}$$

zerlegt. Zur Darstellung von $H_1(z)$ durch ein kanonisches Netzwerk wird geschrieben:

$$Y_1(z) = A_0 z X_1(z) + A_1 z X_1(z) z^{-1} + A_2 z X_1(z) z^{-2}.$$

Diese Formel lässt sich auch in der dem Horner Schema zugrunde liegenden Form darstellen. Durch Ausmultiplizieren ergibt sich:

$$Y_1(z) = [A_2 z X_1(z) z^{-1} + A_1 z X_1(z)] z^{-1} + A_0 z X_1(z).$$

Damit lässt sich das dazugehörige Netzwerk erstellen (*Abb. 4.13*):

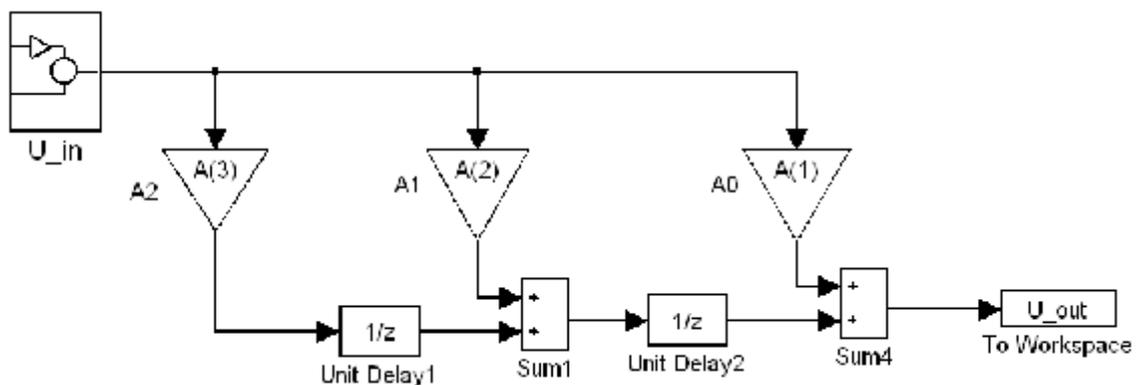


Abb. 4.13:
 Filternetzwerk für ein transversales Filter 2. Ordnung auf der Grundlage der Polynomschreibweise nach Horner

Für das rein rekursive Netzwerk ergibt sich folgender Zusammenhang:

$$Y_2(z) = X_2(z) - B_1 g Y_2(z) z^{-1} - B_2 g Y_2(z) z^{-2}$$

Oder durch Darstellung des Nennerpolynoms in der Horner'schen Form:

$$Y_2(z) = X_2(z) - \left\{ \left[B_2 g Y_2(z) z^{-1} + B_1 g Y_2(z) \right] z^{-1} \right\}.$$

Hieraus wird das in Abb. 4.14 dargestellte Netzwerk für $H_2(z)$ entwickelt.

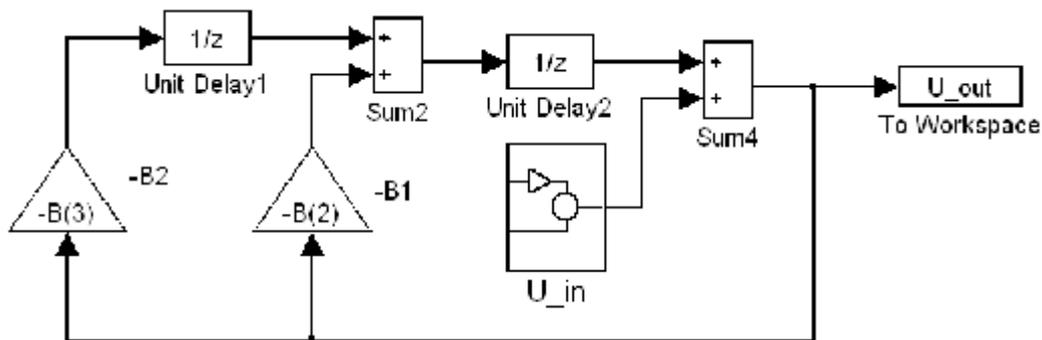


Abb. 4.14:
Rein rekursives Filternetzwerk

Werden die beiden vorgestellten Filternetzwerke für $H_1(z)$ und $H_2(z)$ zusammengefasst, entsteht ein kanonisches Netzwerk, welches die Übertragungsfunktion $H(z)$ repräsentiert (Abb. 4.15).

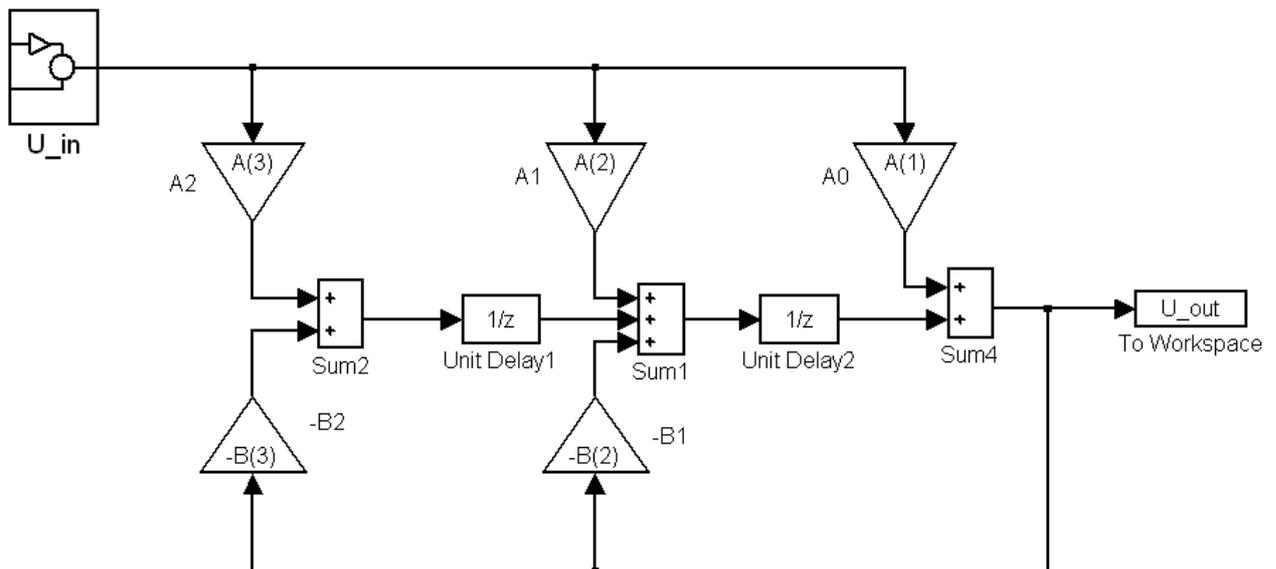


Abb. 4.15:
Allgemeine Übertragungsfunktion 2. Ordnung, dargestellt durch ein kanonisches Netzwerk

Um ein zeitraubendes Umspeichern (*s.a. Anhang C.1*) zu vermeiden, werden Zwischenvariablen eingeführt. Für die Zwischenvariablen an den Additionsstellen vom Ausgang zum Eingang hin betrachtet ergibt sich:

$$Y(z) = A_0 \mathbf{g}X(z) + V(z) \mathbf{g}^{-1}$$

$$V(z) = A_1 \mathbf{g}X(z) + U(z) \mathbf{g}^{-1} - B_1 \mathbf{g}Y(z)$$

$$U(z) = A_2 \mathbf{g}X(z) - B_2 \mathbf{g}Y(z)$$

Dieses Gleichungssystem wird mit Hilfe der inversen z-Transformation in den Folgenbereich transformiert, wodurch die entsprechenden Differenzgleichungen (DGL) entstehen. Wird nun derselbe Speicherplatz für $u(n)$ und alle vergangenen $u(n)$ -Werte (*z.B. $u(n-1)$, $u(n-2)$ etc.*) sowie für $v(n)$ und dessen vergangenen Werten vorgesehen, wird durch die Bildung der drei Teilgleichungen aus der Gesamt-DGL ein für den Controller zeitraubendes Umspeichern vermieden. Die Umsetzung der drei Teilgleichungen in einem Programm ist in Abb. 4.16 dargestellt.

```

while(1)          //Endlosschleife
{
    u0 = read_adc; //Eingangswert vom AD

    u3 = a0*u0+u2;
    u2 = a1*u0-b1*u3+u1;
    u1 = a2*u0-b2*u3;

    send_dac = u3; //Ausgabewert zum AD
}

```

Abb. 4.16:
Umsetzung der drei Teilgleichungen

In dem Programmausschnitt ist zu erkennen, dass der Umspeichervorgang entfällt. Ob eine in [4] beschriebene "Modulo-Adressierung" mit Verwendung von Arrays, bei der das Umspeichern der Registerinhalte durch Verändern der entsprechenden Adressen erreicht wird, einen Zeitgewinn bringt, ist an der letztendlich verwendeten Hardware zu untersuchen.

4.6 Berechnungszeit für ein IIR-Filter nach Kap. 4.5 mit einem Mikrocontroller

Da es gilt, einen möglichst preisgünstigen Nachbau des bestehenden analogen Kanalmodells zu erstellen, wurde zunächst die Zeit ermittelt, die ein gängiger Mikrocontroller (μC) benötigt, um ein IIR-Filter 2. Ordnung zu berechnen. Daher wurde für einen ersten Realisierungstest der betriebsbereite ATmega32 der Firma Atmel verwendet.

Der Analog-Digital-Wandler (ADC) und Digital-Analog-Wandler (DAC) sowie weitere Hardware sind bereits auf dem Bord vorhanden, allerdings keine Hardwaremultiplizierer. Da die Multiplikationen somit auf Softwareebene ausgeführt werden, war zu erwarten, dass bei der Lösung einer solchen Aufgabe mit einem μC nicht die geforderte Berechnungsgeschwindigkeit erreicht wird. Dafür entstand ein Überblick über die Anforderungen an die benötigte Hardware.

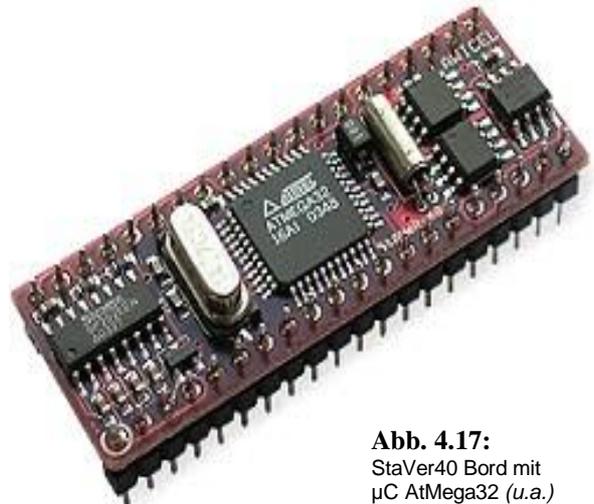


Abb. 4.17:
StaVer40 Bord mit μC AtMega32 (u.a.)

Mit Blick auf den sehr kurzen und einfachen (mit Hilfe der Programmiersprache "C" erstellt) Code für ein IIR-Filter 2. Ordnung wird verständlich, warum die benötigte Rechenzeit einmal kurz ermittelt wurde. Zudem konnten so bereits ein paar Versuche mit herunterfaktorierten Geschwindigkeiten und Grenzfrequenzen durchgeführt werden, bis ein geeignetes Gerät gefunden war. Das dazugehörige Testprogramm befindet sich im Anhang D.3. Es folgt in Abbildung 4.18 das entsprechende Struktogramm.

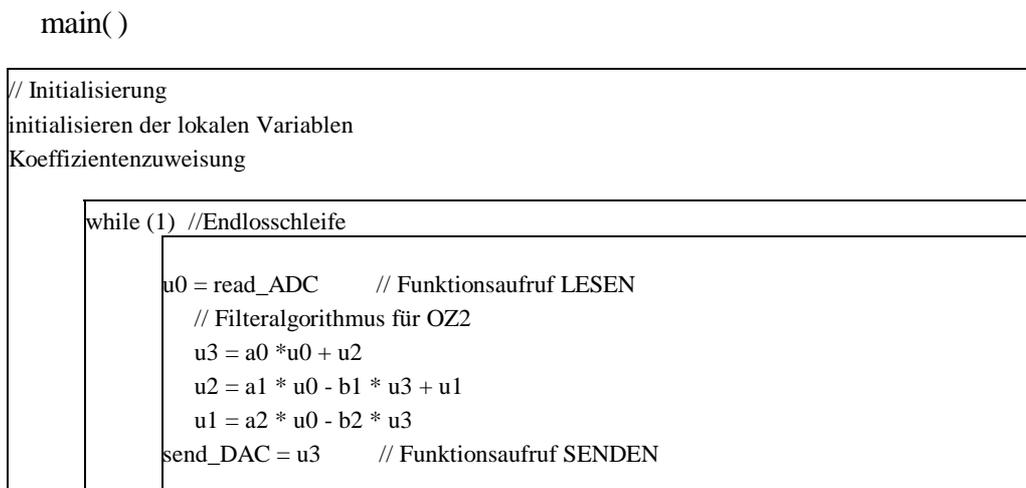


Abb. 4.18:
Struktogramm für die Ermittlung der Berechnungszeit mit dem StaVer40 Bord mit μC AtMega32 (u.a.)

Als Ergebnis wurde ermittelt, dass der ATmega32 mit 14,745 MHz Quarztakt für einen Berechnungsdurchlauf der "While(1)-Endlosschleife", bei einem TP 2. Ordnung (9 Multiplikationen/Additionen oder auch "MAC" multiply/accumulate) 160µs (6,25KHz) benötigt. Da dies weit von einer akzeptablen Berechnungszeit (ca. 500ns bei 30-fach Oversampling) entfernt ist, wurde mit einem realistischen "Short-Filter" (ganzzahlige Koeffizienten) eine Durchlaufzeit von 19,5µs (51KHz) ermittelt.

Das Resultat dieser Tests ist, dass ein µC bei der Multiplikation auf Softwareebene viel zu langsam rechnet und daher die Verwendung von Hardwaremultiplizierern erforderlich ist. Später wird ab Kap. 5.2 eine Unterabtastung (oder für den speziellen Fall eines digitalen Eingangssignals.: Oversampling) des Eingangssignals genutzt, um ein zeitraubendes Einlesen des aktuellen, digitalen Signalpegels nach jedem Berechnungsdurchlauf zu vermeiden. Dies ist möglich, da eine Änderung des digitalen Eingangssignals für mehrere Filterberechnungsdurchläufe nicht zu erwarten ist.

Laut Matlab-Simulation (Kap. 4.2) genügt eine Durchlaufzeit von ca. 770ns (mit gutem ITP) bis 156,25ns für ein Filter 10. Ordnung. Ein wesentlich schnellerer "Acorn Risc Machine" (ARM)-Controller schafft dies auch nicht. Daher wurde ein "mittelschwerer" bis "schwerer" Digitaler Signalprozessor (DSP) oder Field Programmable Gate Array (FPGA) gesucht. Für die Suche nach einem passenden Controller mit Hardwaremultiplizierern wurde als Leistungsangabe in erster Linie auf die Berechnungszeit für eine Multiplikation bzw. Addition (MAC) geachtet. Dabei gilt für das zu erstellende Kanalmodell:

Bei einem Eingangssignal von 64Kbit/s und einem Oversampling um den Faktor 100 wird eine Abtastfrequenz von 6,4MHz am Filter benötigt. Ein Filter 10. Ordnung (41 MAC) benötigt zur Berechnung also $41MAC \cdot 64 KBit/s = 262,4 MegaMAC/s$ (MMACs). Bei beispielsweise 8 Hardwaremultiplizierern, die z.B. jeweils 10 CPU Taktschritte für ein MAC benötigen, muss der Prozessor mit

$$\frac{262,4MMACs \cdot 10CPU\text{Taktschritte}}{8\text{Multiplizierer}} = 328MHz$$

laufen.

Benötigt wird also ein Controller, der 262,4MMACs schafft (diese Angabe findet sich bei fast allen Controllern mit Hardwaremultiplizierern im Datenblatt).