

Aufgabe

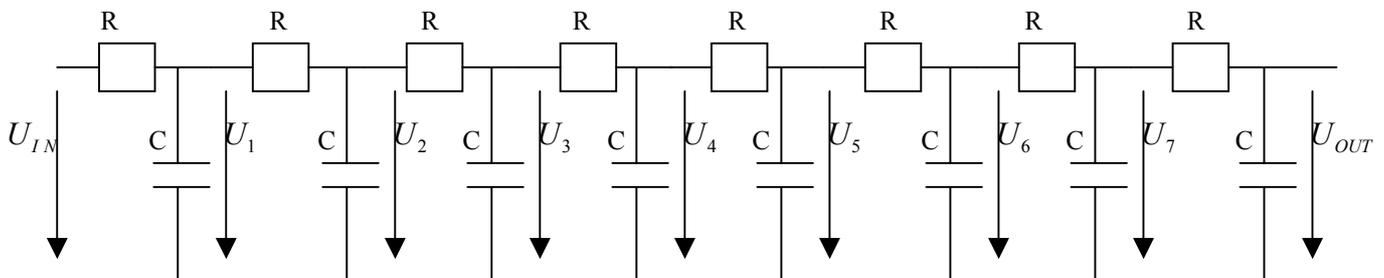
Aufgabe ist die digitale Nachbildung einer Einrichtung zur Simulation von Übertragungskanälen (*Kanalmodell*). Ein analoges Kanalmodell wird bereits für den Praktikumsversuch „Entzerrung von Digitalsignalen“ im Labor der Übertragungstechnik an der HAW-Hamburg eingesetzt. Dabei werden Rechteckimpulse über RC-Kettenglieder¹, die für den Praktikumsversuch eine Leitung mit Tiefpasscharakteristik nachbilden, übertragen. Die Pulse werden dabei derart verzerrt, dass Nachbarzeichenbeeinflussung (*ISI*) auftritt. Das Signal muss dann im Empfänger vor der Abtastung so entzerrt werden, dass die ISI zu den Abtastzeitpunkten beseitigt wird.

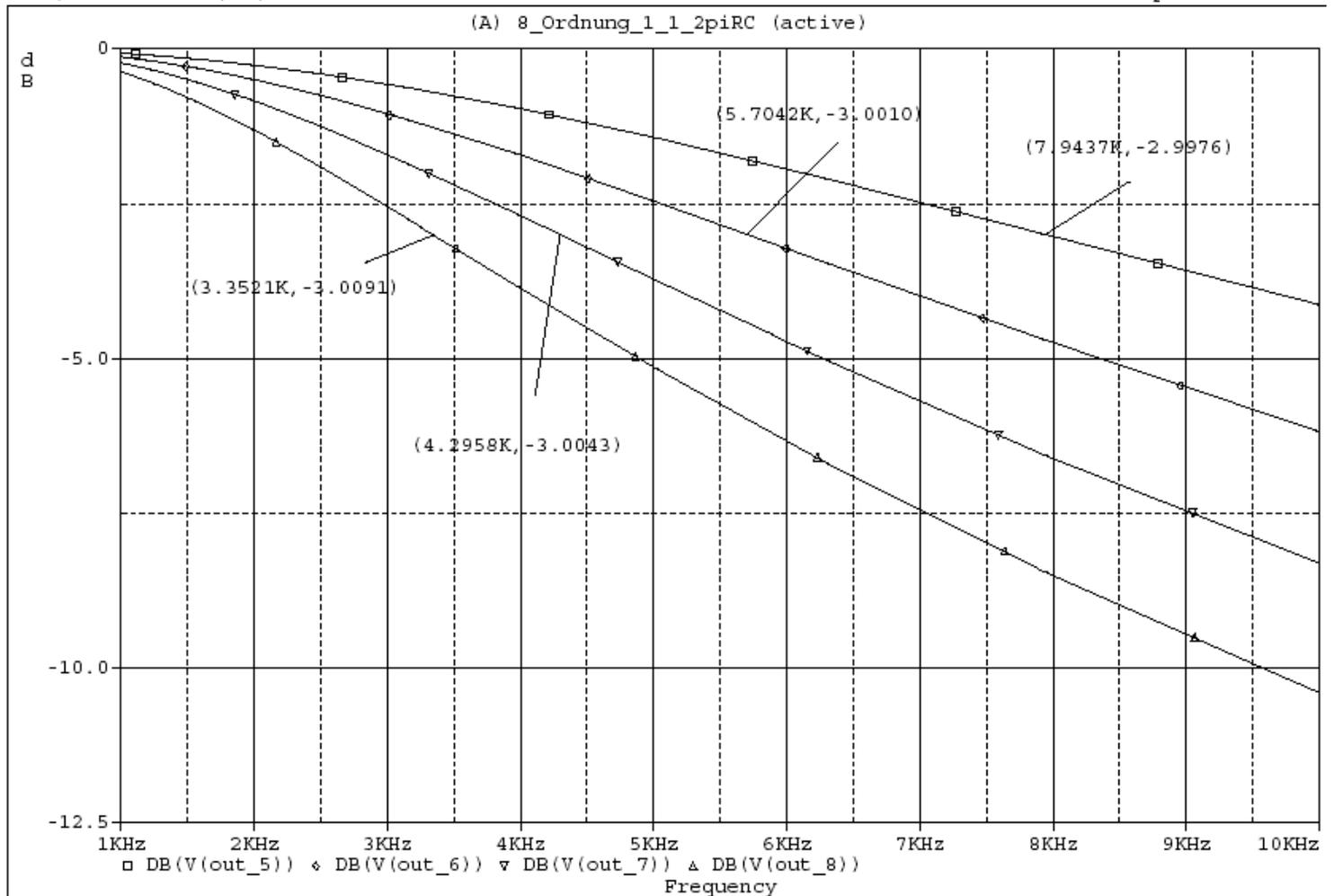
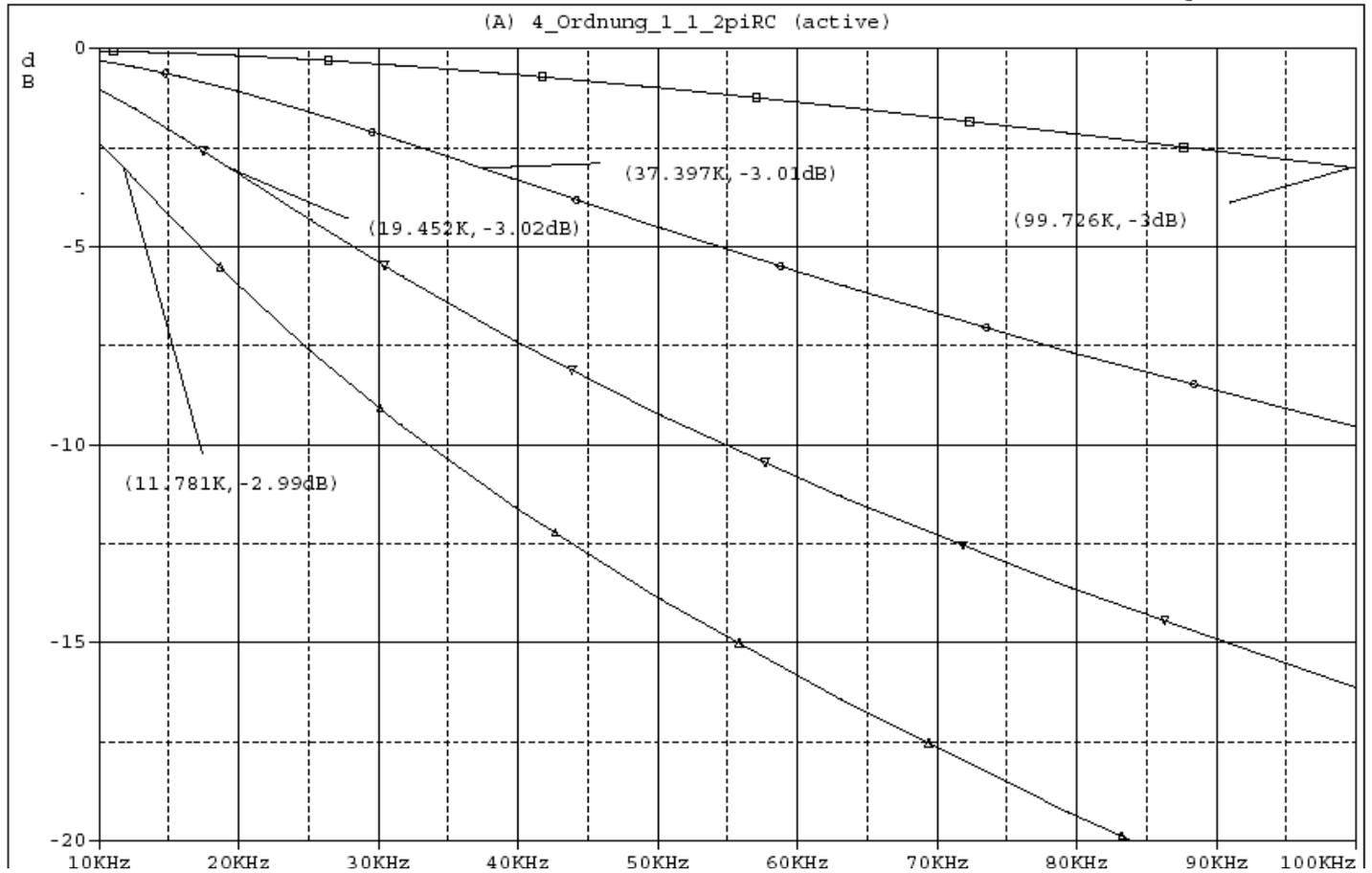
Den Praktikumsversuch „Entzerrung von Digitalsignalen“ habe ich zur Sicherheit einmal mitgeschickt (Dateien: Praktikum_Entzerrung.pdf und Entzerrung_korrigiert.pdf)

¹ Gemäß der Leitungstheorie lässt sich für ein infinitesimal kurzes Leitungsstück eine Ersatzschaltung angeben, die aus der Reihenschaltung eines Widerstands-, eines Induktivitätselementes und der Parallelschaltung eines Leitwert- und Kapazitätselements besteht. Für niedrige Frequenzen lässt sich die Längsinduktivität und der Querleitwert vernachlässigen. Des Weiteren wird das infinitesimal kleine Leitungsstück durch eine endliche Länge ersetzt. So ist das Kanalmodell aufgebaut (BILD). Es besteht aus einer Reihenschaltung einfacher RC-Glieder, deren Anzahl zwischen 1 und 8 einstellbar ist. Mit zunehmender Anzahl der Glieder wird damit eine zunehmende Leitungslänge nachgebildet. Dieses einfache Modell ist zwar keine gute Approximation einer realen Leitung, erzeugt aber sehr gut die für den Praktikumsversuch benötigten Verzerrungseffekte. [1]

Analoges RC-Kanalmodell

Die Ordnungszahl des Kanalmodells lässt sich mit Kippschaltern einstellen, die die gewünschte Anzahl von RC-Kettengliedern hinzuschalten bzw. wegschalten.





Eine Ermittlung der Grenzfrequenzen am Kanalmodell mit einem selektiven Voltmeter bestätigten die in PSpice 9.2 simulierten Werte beinahe exakt und auch eine Messung mit einem Spektrumanalysator zeigt exakt den simulierten Verlauf der Amplitudengänge.

Nachdem ich jetzt das analoge Kanalmodell berechnet, simuliert und nachgemessen habe, begann ich mit der Umrechnung der analogen Filterkoeffizienten in Digitale nach der "Bilinearen Transformation" und "Impulsinvarianten Nachbildung" mit Hilfe von MatLab7.0

Bsp.: Umrechnung der analogen Koeffizienten in Digitale

berechnete analoge Ü-Funktion 2.Ordnung:

$$G(s) = 1/(1+3sT+(sT)^2) \quad \text{mit z.B. } T=250\mu\text{s}$$

Eingabe In Matlab:

```
[A] = 1
[B] = [(250e-6)^2 3*250e-6 1] //Koeffizienten 1 3 1
[Ab,Bb] = bilinear(A,B,100e3) //Filtersamplefrequenz 100KHz gewählt, da der AIC23
//nur bis max.96KHz arbeiten kann.→ später genau!!!
```

Ergebnis:

```
[Ab] = 1e-3* [0.3772 0.7544 0.3772]
[Bb] = [1 -1,8853 0.8868]
```

Nach der impulsinvarianten Methode:

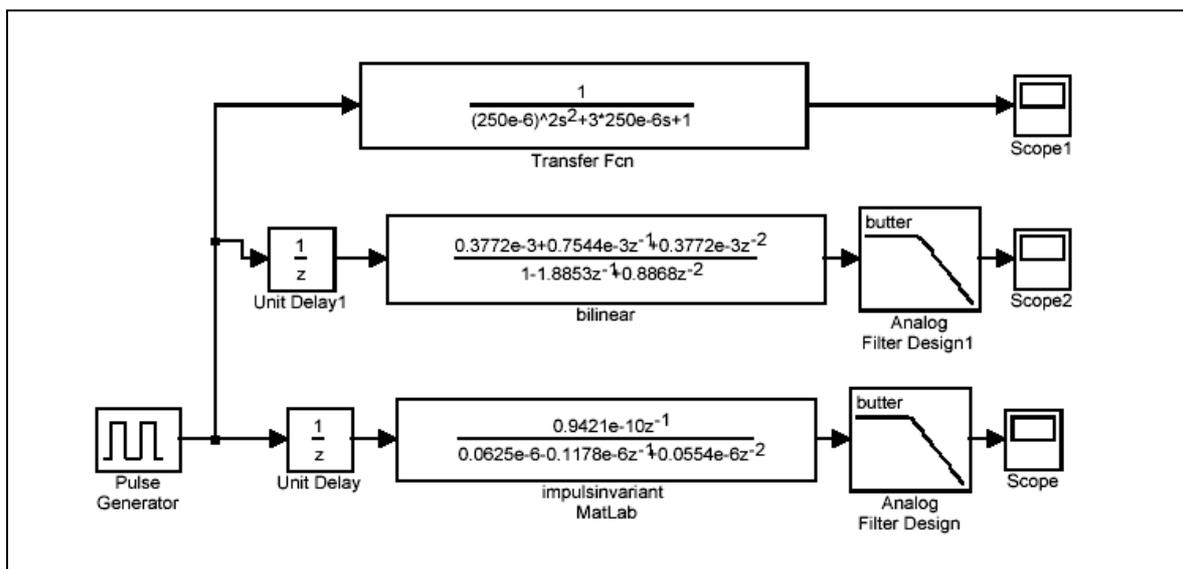
```
[Ai,Bi] = impinvar(A,B,100e3)
```

Ergebnis:

```
[Ai] = 1e-10* [0 0.9421]
[Bi] = 1e-6* [0.0625 -0.1178 0.0554]
```

Eine Simulation in MatLab Simulink bestätigt die identische Beeinflussung des Eingangssignal für die hier gezeigte 2.Ordnung, egal ob analoger TP (*Transfer Fcn*) oder Digitaler (*bilinear, impulsinvariant*).

MatLab Simulation



Realisierung des IIR Filters (ohne Buffer)

Mein Digitaler Filter soll also folgendes machen:

Das 64Kbit/s Eingangssignal muss synchron mit 64KHz am AD-Wandler abgetastet werden. Mit diesem Abtastwert werden dann hundert Ausgabewerte berechnet und mit 6,4MHz ($100 \cdot 64\text{KHz}$) am DA-wandler ausgegeben.

Das dazugehörige Programm soll dann folgendermaßen aussehen:

```
#include "DSK6713_AIC23.h"           //codec-DSK file support
Uint32 fs=DSK6713_AIC23_FREQ_6.4MHZ;-) //Da sich AD und DA-Wandler meist nicht ge-
                                        trennt initalisieren lassen, wähle ich die
                                        am DA-Wandler geforderte Frequenz und neh-
                                        me beim AD-Wandler bei 100-fach Over-
                                        sampling nur jeden 100.Wert auf.

void main()
{
short sample_data;

double a0,a1,a2,b1,b2,u0=0,u1=0,u2=0,u3=0; //Declare local variables
short u=0;    int j=100,k=0;

a0 = 0.0957;    //Zählerkoeffizienten
a1 = 0.1915;
a2 = 0.0957;
b1 = -0.5143;   //Nennerkoeffizienten
b2 = -0.1028;

comm_poll();           //init DSK, codec, McBSP
DSK6713_LED_init();

while(1)               //infinite loop
{
    if (j==100)
    {
        sample_data = input_sample(); //input sample mit 64KHz
        u0 = (double)sample_data; //short/integer to double conversion
        j=0;
        k=0;
    };

//IIR 2.Ordnung mit 100fach Oversampling
    if (k<100)
    {
        u3 = a0*u0+u2;
        u2 = a1*u0-b1*u3+u1;
        u1 = a2*u0-b2*u3;
        u = (short)u3; //double to short/integer conversion
        output_sample(u); //Ausgabe am DA-Wandler mit 6,4MHz
        k=k+1;
    };
    j=j+1;
}
}
```

Diese Programm ist allerdings noch nach dem Polling-Verfahren aufgebaut, wodurch der Filter nicht mit dem Bitgenerator synchronisiert ist und immer mal wieder ein Bit bei der Abtastung am AD-Wandler verloren geht.

Um dieses Problem zu vermeiden bin ich gerade dabei, ein Interrupt gesteuertes Programm zu entwerfen.

Nun zu meinen Problemen bei der Realisierung dieses Projektes:

Für erste Zeitmessungen habe ich zunächst ein einfaches IIR-Filter auf dem TMS320C6713 DSK Bord der Firma Texas Instruments programmiert und eine Berechnungszeit von $0,6\mu\text{s}$ ($1,67\text{MHz}$) für ein Filter 2. Ordnung ermittelt. Da ein Oversampling von 100 ziemlich großzügig dimensioniert ist wird hier ein Oversampling von $25 \cdot 64\text{kHz} = 1,6\text{MHz}$ vermutlich ausreichen.

Allerdings ist die Filterberechnungszeit hier schon fast zu langsam und wird bei 8. Ordnung ziemlich sicher nicht ausreichen. Deshalb wollte ich einmal fragen, ob Sie eine Lösung kennen, mit der ich noch schneller multiplizieren kann?

Ich vermute das Virtex2 Bord wird das schneller können?

Zudem wird beim DSK-Bord das Ausgangssignal vom AIC23-Codec nach der Ausgabe über einen Kondensator DC-frei gemacht, was ich für meine Anwendung nicht gebrauchen kann. Man kann das Signal zwar vorher abgreifen, ist aber ziemlich eng auf dem Bord.

Zudem kommt, das der Antialias-TP bei der AD-Wandlung mit dem AIC23 automatisch auf die halbe Abtastfrequenz eingestellt wird. Da sich AD und DA-Wandler allerdings nicht getrennt initialisieren lassen, wird der Antialias-TP auf die vom DA-Wandler geforderten $6,4\text{MHz}/2$ eingestellt, also $3,2\text{MHz}$ statt wie bei der AD-Wandlung gefordert auf 32kHz .

Da der AIC23 allerdings nur mit minimal 8kHz beim AD-Wandler und beim DA-Wandler mit maximal 96kHz arbeiten kann, ist mit dem AIC23 Codec maximal ein 12fach oversampling ($96/8=12$) möglich. So habe ich das Bord als Filternachbildung mit einem 8Kbit/s Eingangssignal und 12fach Oversampling betrieben, wobei ich dann auch bemerkte, dass der AntialiasTP bei 48kHz statt 4kHz arbeitet und das noch eine Synchronisierung zwischen Bitgenerator und AD-Wandler erfolgen muss.

Das nächste Problem ist die schnelle DA-Wandlung.

Ich habe jetzt schon einige Lösungen mit dem C6713 gefunden, z.B:

http://dsp.gbm.de/produkt_detail.php?agrp=912&anr=80107-1

mit folgendem Aufsteckmodul:

http://dsp.gbm.de/produkt_detail.php?agrp=930&anr=80020-24

allerdings liegen die Preise hierfür zusammen bei ca. 2500€ und das ist eindeutig zu viel.

Auch die PCI-Steckkarten sind nicht günstiger. Ein bezahlbarer Preis wäre ein Betrag zwischen 500€ und 1000€ .

Deshalb wollte ich Sie fragen ob Sie evtl. eine günstigere Realisierung hierfür kennen, oder gibt es noch schnellere DA-Wandler Module (*als 500kHz*) für das Virtex2 Bord? Ist so etwas vielleicht sogar schon im Informatiklabor verwendet worden?

Ich würde dies auch gerne noch einmal mit Ihnen persönlich besprechen.

Als Anhang noch die Originalnachricht mit den Anforderungen, die ich mit Herrn Kröger abgestimmt habe.

Mit freundlichen Grüßen

Christoph Lang

----- Original-Nachricht -----

Datum: Thu, 12 Jul 2007 08:30:59 +0200

Von: "Prof. Dr.-Ing. P. Kröger" <Kroeger@etech.haw-hamburg.de>

An: christoph.lang@gmx.de

CC: missun@etech.haw-hamburg.de, neugebauer@etech.haw-hamburg.de

Betreff: DSP-Board

> Sehr geehrter Herr Lang,

>

> ich habe das Thema DSP-Board gestern noch mal mit dem Laborleiter

> Herrn Missun besprochen. Sie können ein Board aussuchen, das Ihren

> Vorstellungen entspricht und es dann über Herrn Neugebauer

> bestellen lassen. Wir wollen dann später dieses Board u.a. für die

> Basisband-Kanalmodelle im Laborpraktikum einsetzen.

>

> Das Board mit ADC/DAC soll aus unserer Sicht darüber hinaus

> folgende Randbedingungen erfüllen:

>

> 1) Gleicher Festkomma-Prozessor wie der, der im DSP-Labor

> eingesetzt wird (TMS..)

>

> 2) Hinreichend schnelle Abtastung möglich (bei einem Kanal 10 MHz)

> mit freizügiger Einstellung der Abtastraten.

>

> 3) Mehrere analoge Multiplex-Eingänge/Ausgänge (Kanäle)

>

> 4) Mindestens 12 bit ADC/DAC

>

> 5) PC-externes Board über USB/LAN an den PC anschliessbar (entgegen

> meiner letzten Mail).

>

> Mit freundlichen Grüßen

>

> Prof. Dr.-Ing. P. Kröger

>

> HAW Hamburg, Department Informations- u. Elektrotechnik

> Berliner Tor 7, 20099 Hamburg

>

> Email: office@profkroeger.de