

Echtzeit-Realisierung rekursiver Filterstrukturen auf einem digitalen Festkomma-Signalprozessor

Uwe Simmer

Institut für Hörtechnik und Audiologie, FH Oldenburg/Ostfriesland/Wilhelmshaven, D-26121 Oldenburg,
Email: uwe.simmer@fh-oldenburg.de

Einleitung

Dieser Beitrag beschreibt den Praktikumsversuch 'DSP-Programmierung' der zurzeit an der FHOOW Oldenburg im Bachelor-Studiengang Hörtechnik und Audiologie eingesetzt wird. Ziel des Praktikums ist es, die Wirkung verschiedener Filterstrukturen und numerischer Repräsentationen für die Studierenden akustisch erfahrbar zu machen. Es wurde ein Festkomma-Signal-Prozessor (Analog Devices Blackfin ADSP-BF533) gewählt, da diese Prozessorklasse wegen der (im Vergleich zu Fließkomma-Prozessoren) kleineren Chip-Fläche und des geringeren Stromverbrauchs für mobile, batteriebetriebene Geräten wie Hörhilfen sehr gut geeignet ist.

Im Praktikum werden den Studierenden zwei vorbereitete Softwarepakete zu Verfügung gestellt. Eine MATLAB-Software, welche die Filter in verschiedene Strukturen transformiert und eine Skalierung der Filterkoeffizienten auf den Zahlenbereich $[-1, 1)$ vornimmt und eine in C/C++ geschriebene DSP-Software, welche die verschiedenen Filterstrukturen in Echtzeit und parallel auf einem Festkomma-DSP realisiert. Durch Umschalten des Ausgangssignals können die Filterstrukturen akustisch verglichen werden. Die Entwicklungsumgebung ermöglicht zusätzlich die Analyse der Impulsantworten und Übertragungsfunktionen und die Messung der benötigten Taktzyklen auf der Zielhardware. Die Aufgabe der Studierenden in diesem Versuch ist der Entwurf eines Bandpass-Filters mit Telefonbandbreite, die Evaluierung der akustischen Eigenschaften der implementierten Filterstrukturen, ein Vergleich der Übertragungsfunktionen und die Bestimmung der benötigten Rechenleistung. Am Ende des Versuchs sollen die Studierenden in der Lage sein, eine geeignete Filterstruktur für das gegebene Problem auszuwählen.

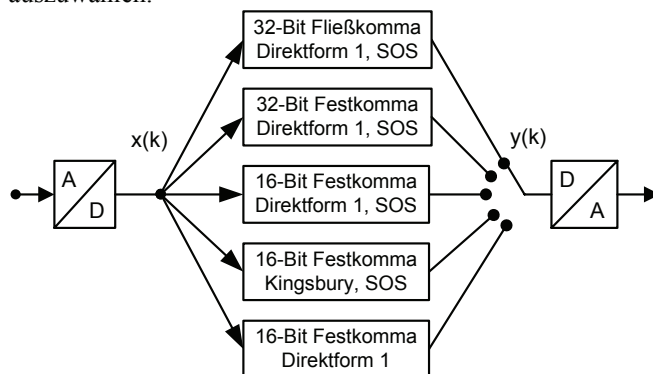


Abbildung 1: Blockschaltbild der DSP-Hard- und Software

Filterstrukturen

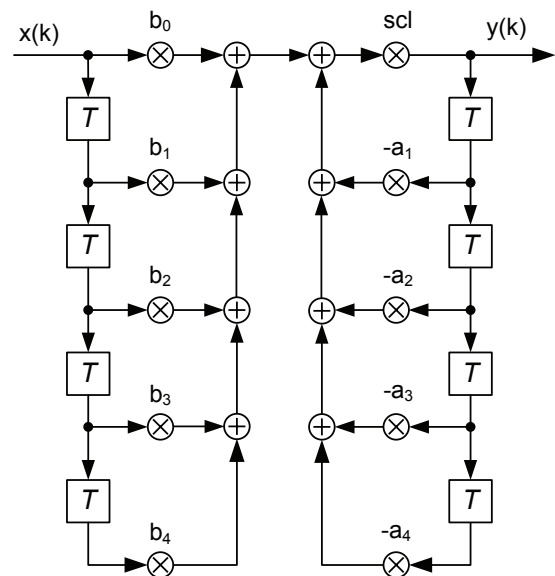


Abbildung 2: Filterstruktur Direktform 1 (DF1)

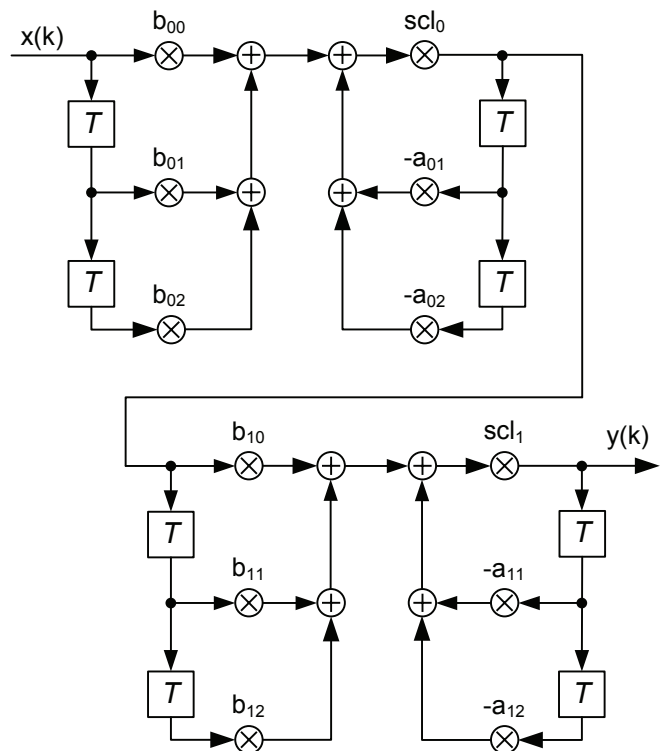


Abbildung 3: Serienschaltung von Abschnitten zweiter Ordnung (DF1-SOS)

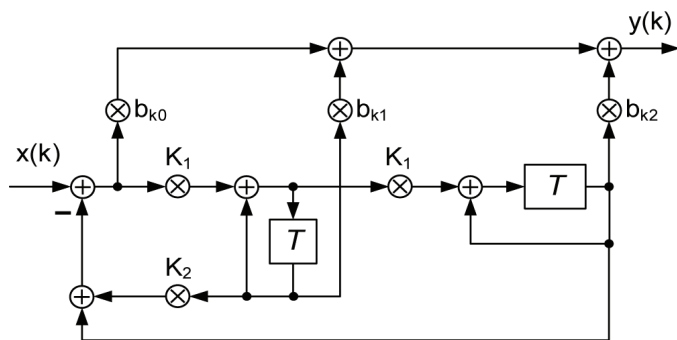


Abbildung 4: Kingsbury-Struktur zweiter Ordnung

Entwicklung des DSP-Praktikums

Damit die Durchführung des Praktikums in dem gegebenen Zeitrahmen von 3 Stunden möglich ist, können die Teilnehmerinnen und Teilnehmer teilweise auf vorbereitete Software-Module zurückgreifen. Diese wurden in der Entwicklungsphase des Versuchs innerhalb eines Semesters von Studierenden entwickelt [8]. Die Software besteht aus:

1. Einem MATLAB-Skript zur Konvertierung und Skalierung der Koeffizienten eines beliebigen IIR-Filters. Das MATLAB-Skript 'WriteCoeffs.m' erhält als Eingangsdaten die Koeffizienten-Vektoren des Filters, zerlegt diese in Abschnitte zweiter Ordnung, normalisiert die Verstärkung der einzelnen Abschnitte, skaliert die Koeffizienten und schreibt fünf verschiedene C/C++ Header-Dateien mit den Filter-Koeffizienten.

2. Einer DSP-Software (in C/C++), welche die zuvor erzeugten Header-Dateien mit den Koeffizienten einbindet, die fünf verschiedenen Versionen des gewünschten Filters parallel in Echtzeit auf einem Festkomma-DSP ausführt (Abbildung 1) und die Impulsantwort und die Übertragungsfunktion direkt auf dem DSP bestimmt. Die realisierten Varianten sind:

- a. 32-Bit Fließkomma Emulation, DF1-SOS Abbildung 3
- b. 32-Bit Festkomma, DF1- SOS Abbildung 3, [3]
- c. 16-Bit Festkomma, DF1- SOS, Abbildung 3, [2]
- d. 16-Bit Festkomma, Kingsbury, SOS Abbildung 4, [4]
- e. 16-Bit Festkomma, DF1 Abbildung 2, [2]

Die Praktikumssteilnehmer können nun ein Filter entwerfen

```
% Filtersynthese
[b, a] = cheby1(n, r, Wn);
WriteCoeffs(b, a, fs, 1, '...\IIR\');
```

und die verschiedenen Realisierungen ohne großen Aufwand auf einem 16-Bit DSP erproben.

Aufgabe der Studierenden

Im ersten Teil des Praktikums haben die Studierenden die Aufgabe, ein Bandpass-Filter vierter Ordnung mit Telefonbandbreite (300-3400 Hz bei 48 kHz Abtastfrequenz) und ein Hochpass-Filter vierter Ordnung mit einer Grenzfrequenz von 30 Hz zu entwerfen. Mit Hilfe des zur Verfügung gestellten MATLAB-Skripts 'WriteCoeffs.m' werden die Koeffizienten skaliert und als Header-Dateien in die DSP-Software eingebunden. Nach der Kompilierung der DSP-Software können Klangqualität, Übertragungsfunktion, Impulsantwort und die Rechenleistung der einzelnen Filter

direkt in der DSP-Entwicklungsumgebung evaluiert werden. Die verschiedenen Filterrealisierungen werden bei 48 kHz Abtastfrequenz parallel auf dem DSP ausgeführt und können durch Drücken eines Schalters auf der DSP-Platine umgeschaltet werden. Aufgabe der Studierenden ist die Evaluierung der Leistungsfähigkeit (Rauschen, Fehler in der Übertragungsfunktion) und Kosten (MIPS) der einzelnen Realisierungen sowie die Auswahl der am besten geeigneten Variante.

Im zweiten Teil des Praktikums entwickeln die Studierenden selbständig einen Delay-Effekt mit variabler Verzögerungszeit, (0,0625 0,125 5,0 50 und 125 ms) der in Echtzeit auf dem DSP lauffähig ist.

Ergebnisse

IIR-Filter hoher Ordnung können problemlos auf 16-Bit-Festkomma-Prozessoren realisiert werden, wenn eine geeignete Filterstruktur (DF1-SOS) und eine geeignete numerische Darstellung (32-Bit Festkomma) gewählt wird. Ungeeignete Strukturen können zu stark rauschenden Filtern, fehlerhaften Übertragungsfunktionen und zu instabilen Filtern führen.

Unsere Erfahrungen mit diesem Praktikum zeigen, dass diese Aufgabe mit Hilfe der zur Verfügung gestellten Softwaremodule von Studierenden des Bachelor-Studiengangs erfolgreich gemeistert werden kann.

Literatur

- [1] Analog Devices, ADSP-BF53x/BF56x Blackfin Processor Programming Reference, Rev. 1.0, Juni 2004.
- [2] Analog Devices, VisualDSP++ 4.0 C/C++ Compiler and Library Manual for Blackfin Processors, Rev. 3.0, Jan. 2005.
- [3] Analog Devices, Engineer to Engineer Note EE-186 Extended-Precision Fixed-Point Arithmetic on the Blackfin Processor Platform, Mai 2003.
- [4] N. G. Kingsbury: Second-order recursive digital filter element for poles near the unit circle and the real z-axis, Electronic Letters, pp. 155-156, März 1972.
- [5] U. Zölzer: Roundoff Error Analysis of Digital Filters, J. Audio Eng. Soc., Vol. 42 Nr. 4 S. 232-244, Apr. 1994.
- [6] R.J. Clark: Investigation into digital audio equaliser systems and the effects of arithmetic and transform errors on performance, Dissertation, University of Plymouth, Apr. 2001.
- [7] S. D. Stearns: Digital signal processing with examples in MATLAB, CRC Press, 2003.
- [8] P. Roeske, B. Ohl, E. Rasumow, J. Schaffmeister: Praktikumsdokumentation "DSP-Programmierung", Feb. 2006.
- [9] U. Simmer: A Hands-On Fixed-Point DSP Lab for Undergraduate Students", Proceedings of the 12th Digital Signal Processing Workshop and 4th Signal Processing Education Workshop, Wyoming, USA, Sept. 2006.