

U4 FIR Tiefpass Filter Koeffizientenberechnung

Ziel

Diese Übung hat zum Ziel ein FIR-Filter mit aus Grenzfrequenz und Länge als Funktion in Hochsprache direkt zu implementieren. Die Filterkoeffizienten werden mit der Window-Methode bestimmt.

Umfeld

FIR-Filterkoeffizienten können auf verschiedene Arten bestimmt werden. Die klassische Art läuft über die Auswertung der idealen Impulsantwort mit anschließender Koeffizientengewichtung durch ein Bewertungsfenster (Window).

Diese einfache Methode ist schlussendlich ungenau und erlaubt aber keine unabhängige Definition der Welligkeit im Sperr- und Durchlassbereich. Jedoch können die Berechnungen mit einem Taschenrechner durchgeführt werden.

Für die Praxis wird man aber im Regelfall auf ein Filtersyntheseprogramm zurückgreifen, das für den DSP und das geforderte Übertragungsverhalten optimierte Filterkoeffizienten liefert.

Berechnungsgrundlagen

Die zu implementierenden Filterkoeffizienten h_{2i} setzen sich aus der idealen Impulsantwort und den Gewichtskoeffizienten w_i des Fensters zusammen:

$$h_{2i} = h_{1i} \cdot w_i \quad i : 0 \dots N$$

Wobei die ideale Impulsantwort und die Fensterfunktion folgendermassen definiert sind:

$$\Omega_c = \frac{\omega_c}{f_s} = \quad N : \text{Anzahl Taps}$$

$$\text{Tiefpass : } h_1(i) = \begin{cases} \frac{\Omega_c}{\pi} & i = \frac{N-1}{2} \\ \frac{\sin\left(\left(i - \frac{N-1}{2}\right)\Omega_c\right)}{\left(i - \frac{N-1}{2}\right)\pi} & \text{sonst} \end{cases}$$

$$\text{Hochpass : } h_1(i) = \begin{cases} \frac{\pi - \Omega_c}{\pi} & i = \frac{N-1}{2} \\ \frac{\sin\left(\left(i - \frac{N-1}{2}\right)\pi\right) - \sin\left(\left(i - \frac{N-1}{2}\right)\Omega_c\right)}{\left(i - \frac{N-1}{2}\right)\pi} & \text{sonst} \end{cases}$$

Fenster	Sperrdämpfung [dB]	K	Fensterfunktion $w(i)$
Rechteck	21	2	$w(i) = 1$
Bartlett	25	4	$w(i) = 1 - \frac{2 \left i - \frac{N-1}{2} \right }{N}$
Hanning	44	4	$w(i) = 0.5 + 0.5 \cos\left(\frac{2\pi \cdot \left(i - \frac{N-1}{2}\right)}{N}\right)$
Hamming	53	4	$w(i) = 0.54 + 0.46 \cos\left(\frac{2\pi \cdot \left(i - \frac{N-1}{2}\right)}{N}\right)$

Blackman	74	6	$w(i) = 0.42 + 0.5 \cos\left(\frac{2\pi \cdot \left(i - \frac{N-1}{2}\right)}{N}\right) - 0.08 \cos\left(\frac{4\pi \cdot \left(i - \frac{N-1}{2}\right)}{N}\right)$
Kaiser ($\gamma=2.12$)	30	1.54	$w(i) = \frac{J_0\left(2\gamma \sqrt{i \cdot (N-i-1)}\right)}{J_0(\gamma)}$
Kaiser ($\gamma=4.54$)	50	2.93	dito
Kaiser ($\gamma=7.76$)	70	4.32	dito
Kaiser ($\gamma=8.96$)	90	5.71	dito

J_0 : Modifizierte Bessel-Funktion 0. Ordnung.

Beispiel

Vorgaben:

- $f_s := 16 \cdot 10^3$ [Hz] Sampling Frequenz
- $f_c := 2 \cdot 10^3$ [Hz] Grenzfrequenz Durchlass
- $N := 11$ Anzahl Taps

Berechnungen:

$\omega_c := 2\pi \cdot f_c$

$\Omega_c := \frac{\omega_c}{f_s} \quad \Omega_c = 0.785$ Normierte Grenzfrequenz des idealen Filters

$$h_{LP}(i, \Omega_c, N) := \begin{cases} \frac{\Omega_c}{\pi} & \text{if } i = \frac{N-1}{2} \\ \frac{\sin\left[\left(i - \frac{N-1}{2}\right)\Omega_c\right]}{\left(i - \frac{N-1}{2}\right)\pi} & \text{otherwise} \end{cases}$$
 Ideale Tiefpass FIR-Filterkoeffizienten

$i := 0..N-1$

$hammingWindow(i, N) := 0.54 + 0.46 \cdot \cos\left(\frac{2\pi \cdot \left(i - \frac{N-1}{2}\right)}{N}\right)$ $h_1 := h_{LP}(i, \Omega_c, N)$

$dB(x) := 20 \log(x)$

$w_i := hammingWindow(i, N)$

$h_2 := w_i \cdot h_1$

Amplitudengang mit Hamming-Fenster: (Sperrdämpfung > 53dB)

$f := 10, 14..10000$ [Hz] $G(z) := \sum_{k=0}^{N-1} h_{2,k} \cdot z^{-k}$ Frequenzdiskrete Übertragungsfunktion

ideale Filterkoeffizienten

$h_1 =$

0	0
0	-0.045
1	0
2	0.075
3	0.159
4	0.225
5	0.25
6	0.225
7	0.159
8	0.075
9	0
10	-0.045

Hamming-Fensterkoeffizienten

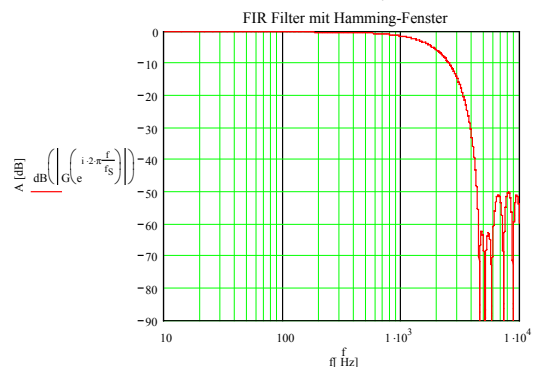
$w =$

0	0
0	0.099
1	0.239
2	0.475
3	0.731
4	0.927
5	1
6	0.927
7	0.731
8	0.475
9	0.239
10	0.099

Hamming-gewichtete Filterkoeffizienten

$h_2 =$

0	0
0	$4.44 \cdot 10^{-3}$
1	0
2	0.036
3	0.116
4	0.209
5	0.25
6	0.209
7	0.116
8	0.036
9	0
10	$4.44 \cdot 10^{-3}$



Aufgaben

1. Aufsetzen eines neuen, leeren C-Projektes (z.B. Uebung4_FIR_TP2kHz, keine Leerschläge erlaubt) auf der Grundlage des Prototypen-Files. Kommentieren und Anpassen des Prototypen. Vgl. hierzu auch Übung 1.

2. Implementieren der Übertragungsfunktion durch Definition der Filterfunktion, die die eigentliche Übertragungsfunktion aufnimmt:

```
void filterFIR(void)
{
    ....
    y= ....;   Ausgangssignal retournieren
}
```

Die Werte werden mittels Seiteneffekt (d.h. globale Variablen) übergeben.

Die Filterkoeffizienten sind als initialisiertes Array ausserhalb und vor der Filterfunktion zu definieren. Ebenso der History-Buffer zur Aufnahme der $x(n-k)$.

Hinweise:

Benutzen Sie zur Rechnung den Datentyp `_fract`. Er verkörpert den Festkommatentyp des DSP56002 Signalprozessors mit dem Wertebereich $[-1,1)$. Zu beachten ist, dass der Codec mit 16kHz Samplerate initialisiert wird. Dazu ist die symbolische Konstante `SAMPLE_RATE_16` aus `CodecConst.h` zu benutzen.

Die Filterkoeffizienten sind aus der Rechnung Beispiels zu übernehmen.

3. Austesten mit Musik, evtl. ausmessen des Amplitudenganges mit NF-Voltmeter.
4. Gleiche Aufgabe aber Berechnung und Implementierung für $N=41$.