

ELT-ESE-3 DSBL



Digitale Signaalbewerking practicum

HAN Elektrotechniek/Embedded Systems

ir drs E.J Boks

Opdracht 4: Desktop implementatie van een configureerbaar fixed-point FIR-type banddoorlaatfilter



Doel:

Een applicatie te schrijven die het mogelijk maakt om een digitaal filter te ontwerpen via de grafische interface en het filter te testen met een in de applicatie aanwezige signaalgenerator.

Tijd:

6-8 weken.

Benodigde zaken:

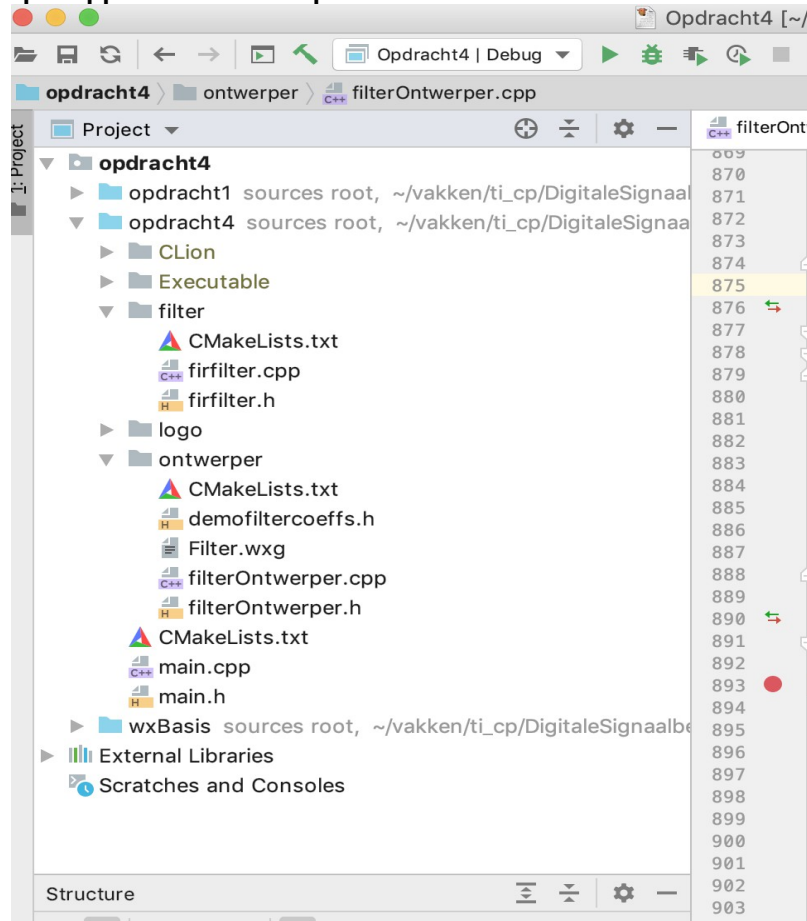
- Werkstation in B1.29/B1.33.
- Theorieboek .
- WxWidgets online handleiding op <http://www.wxwidgets.org>
- Geschreven en geteste klassen uit Opdracht 1.

Beschrijving:

Voltooi een applicatie die het voor de gebruiker eenvoudig maakt om een FIR filter zodanig in te stellen dat een bepaalde signaalband wordt gestopt/doorgelaten. De applicatie moet daarom aan de volgende eisen voldoen:

- Het programma moet worden geschreven in de taal C++.
- Voor de grafische interface wordt de wxWidgets toolkit gebruikt. De voorbeeldcode die als start wordt aangereikt zorgt voor een werkende maar lege applicatie..
- De gebruiker geeft in de GUI de volgende parameters aan:
 - de bemonsteringfrequentie.
 - De start en stopfrequenties van de doorlaatband.
 - De orde van het filter.
 - Een versterkingsfaktor.
 - Het toegepaste venster.
 - Het gebruikte aantal bits voor de Fixed-point codering.
- De applicatie moet FIR coëfficiënten kunnen genereren, welke het door de gebruiker opgegeven frequentiegedrag zullen afdwingen (filteren) in een FIR filter.
 - De FIR coëfficiënten moeten allereerst in floating point formaat worden berekend.
 - De uiteindelijke filter FIR coëfficiënten van het moeten in fixed point (Qx formaat) zijn. De omzetting gebeurt aan de hand van de GUI instellingen voor het aantal te gebruiken bits ([wxSpinCtrl](#) element).
 - Voor afkapping van de coëfficiënten moet het Rechthoek, het Driehoek of het Hamming, venster worden gebruikt. De vensterselectie volgt uit de GUI ([wxChoice](#) element). Het geselecteerde venster neemt de filterorde voor het aantal coëfficiënten.
 - De aangegeven versterkingsfaktor moet worden ingebracht bij de coëfficiënten.
 - De filtercoëfficiënten moeten worden opgeslagen in de [wxArrayShort](#) *filterCoeffs*, aanwezig in de klasse *FilterVenster*.
 - De *FilterFirInt16* klasse moet nog worden uitgebouwd om als FIR filter te functioneren. De klasse constructor verwacht het aantal taps dat het filter heeft, een pointer naar de array waar de coëffs staan en een schaalFaktor waarmee het MAC resultaat wordt teruggebracht naar Int16 formaat.
- De gebruiker moet het gefabriceerde filter kunnen analyseren. Gebruik hiervoor de bovenste twee vensters en teken:
 - In het linkerbovenvenster een plot van de impulsresponsie van het ontworpen venster in het tijddomein.

- In het rechterbovenvenster een plot van het ontworpen venster in het frequentiedomein.
- De gebruiker moet het gefabriceerde filter kunnen testen:
 - De applicatie heeft een cosinus signaalgenerator in zich, welke een testsignaal kan produceren met een frequentie tussen 0 en de 4 maal de bemonsteringsfrequentie. Daarnaast kan de amplitude over het volle 16 bit bereik worden ingesteld.
 - De gekozen testfrequentie wordt als inputsignaal aan het filter worden aangeboden. Het filter moet kunnen filteren met de door de applicatie gegenereerde coëfficiënten.
 - Naast het cosinussignaal kunnen ook worden gegenereerd:
 - een blokgolf met de zelfde amplitude en frequentieeigenschappen als de cosinus.
 - een impuls
 - een stap
- In de aangeleverde code zijn vier bestanden aanwezig : **firfilter.cpp** en **firfilter.h**, alsmede **filterOntwerper.cpp** en **filterOntwerper.h** :



firfilter.cpp moet worden ingevuld met de implementatie van het FIR filter. Zorg er voor dat het filter in fixed point werkt en over loop unrolling (faktor 4) beschikt.
filterOntwerper.cpp moet worden voltooid om de coëfficiënten generatie mogelijk te maken.

- De gebruiker moet de gegenereerde FIR coëfficiënten ook kunnen opslaan in een C Header bestand. Dit is al in de voorbeeldcode voorbereid en hoeft niet door de student te worden uitgewerkt.

Opdracht:

Jullie moeten de reeds aangeleverde voorbeeldcode uitbreiden tot een geheel dat werkt volgens de hierboven gegeven beschrijving.

De onderdelen die jullie moeten schrijven zijn:

- 1 Een invulling van de *FilterFirInt16* klasse. Het filter is gebaseerd op de *RingBuffer* klasse uit opdracht 1. Twee ringbuffers worden gebruikt, een voor de coëfficiënten en een voor de data.
- 2 De *driehoek(const Int32)*, *hamming(const Int32)* en *sinc(const double)* functies in *filterOntwerper.cpp*
- 3 De functies *berekenFixedPoint()* en *berekenFloatingPoint()* welke de conversie tussen floating point $h[n]$ en fixed-point uitvoeren. Deze functies hangen af van de ingestelde hoeveelheid coderingsbits. Let op dat van het aantal opgegeven bits in de GUI er altijd een wordt gebruikt voor het teken.
- 4 De functie *berekenFilterHandler(wxCommandEvent &)* die de filter coëfficiënten berekent en tekent, daarbij rekening houdend met het gekozen venster (driehoek of Hamming).
- 5 De functie *berekenFreqResponsie()* berekent het frequentiebeeld en bepaalt minima en maxima (in dB) van de responsie over 0 tot en met π .
- 6 De functie *tekenFreqSpectrum()* die het frequentiebeeld van de impulsresponsie tekent.
- 7 [optioneel] in `void FilterVenster::tijdViewMuisBewegingHandler(wxMouseEvent &event)` en `void FilterVenster::freqViewMuisBewegingHandler(wxMouseEvent &event)` kan informatie over de grafische coördinaat worden berekend, die iets zegt over $h[n]$ of $H(\Omega)$, bijvoorbeeld de filtererking op een bepaalde frequentie of de waarde van een impulsresponsiesample. Bekijk de demo versie van de opdracht of vraag de docent over meer informatie. Uitwerking van deze optie levert een bonuspunt voor deze opdracht op!

Bij een goed werkende applicatie wordt een beeld verkregen dat lijkt op het onderstaande beeld :



Oplevering:

Toon de werkende applicatie aan de docent.

Schrijf een klein verslag, met daarin de volgende zaken:

- broncode applicatie
- screendump van de werkende applicatie

Oplevering geschiedt tezamen met de andere opdrachten. Ga echter niet verder zonder mondelinge goedkeuring van het werk bij de docent.

Tips

Bekijk eerst **firfilter.h** en **filterOntwerper.h** . Hier staan alle variabelen die jullie moeten gaan gebruiken opgesteld. Bekijk ook waar de wxWidgets event handlers staan en hoe zij de door de gebruiker aangeboden informatie opslaan.

Het **filterOntwerper.cpp** bestand bevat het GUI gedeelte voor een groot gedeelte al voorbereid. Jullie moeten alleen de signaalbewerking delen invullen. Laat het wxGlade commentaar met rust (niet uitvegen). Dit heeft het grafische tool wxGlade nodig, mocht je de GUI lay-out willen bewerken.