

DSP

Labo-opdracht 4

Frequentieselectieve filters

Ing Patrick Van Houtven

s]

Labo 06 : Frequentieselectieve filters

1 Opgave 1

Ontwerp een FIR-filter de 7^{de} harmonische uit een blok golf fitlert. De blok golf heeft een amplitude van 8 V en de frequentie van de blok golf is gelijk aan 1000 Hz. Gebruik als samplefrequentie 16 kHz.

Tip:

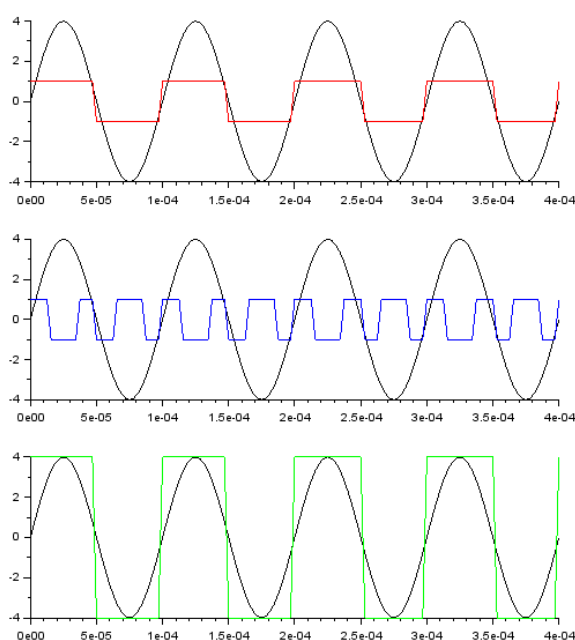
Je kan vanuit een sinusgeneratie gemakkelijk een blok golf maken in scilab via de functie "squarewave" en wel op volgende manier:

Voorbeeldcode maken van blok golf:

```
clc
clf
// maken van een sinus 4 perioden, 10 kHz, Vp = 4 V
// fs = 32 kHz
// t = 0: 0.0000003125 : 0.000025 ];
t = 0: 1/32000 : 4/10000;
sin_10kHz = 4 * sin(2 * %pi * 10000 * t);
// maken van blok golf 10 kHz, amplitude 4 V
blok_10kHz = squarewave(0.5 * sin_10kHz)
subplot(311)
plot2d(t, sin_10kHz,)
plot2d(t, blok_10kHz, style=[color("red")])

blok_10kHz = squarewave(sin_10kHz)
subplot(312)
plot2d(t, sin_10kHz,)
plot2d(t, blok_10kHz, style=[color("blue")])

blok_10kHz = 4 * squarewave(0.5 * sin_10kHz)
subplot(313)
plot2d(t, sin_10kHz,)
plot2d(t, blok_10kHz, style=[color("green")])
```



Schrijf hier je code voor de gevraagde filter:

16 kHz Samplefrequentie:

```
t = [0 : 6.25D-5 : 5D-3];
sin1kHz = 4 * sin(2 * %pi * 1000 * t)
stap = 4 * squarewave(0.5 * sin1kHz)
plot2d(t, sin1kHz)
plot2d(t, stap, style=[color("red")])

// 6.5kHz / 16 kHz = 0.40625
// 7kHz / 16 kHz = 0.4375
// 7.5kHz / 16 kHz = 0.46875
BD = wfir('bp', 50, [0.41 0.47], 'hm', [0 0]);
```

```
BD = poly(BD, 'z', 'coeff');
BD = horner(BD, (1/%z));
BD = syslin('d', BD);
```

```
BD = 4 * flts(stap, BD)
plot2d(t, BD, style=[color("blue")])
```

32 kHz Samplefrequentie:

```
t = [0 : 3.125D-5 : 5D-3];
sin1kHz = 4 * sin(2 * %pi * 1000 * t)
stap = 4 * squarewave(0.5 * sin1kHz)
plot2d(t, sin1kHz)
plot2d(t, stap, style=[color("red")])

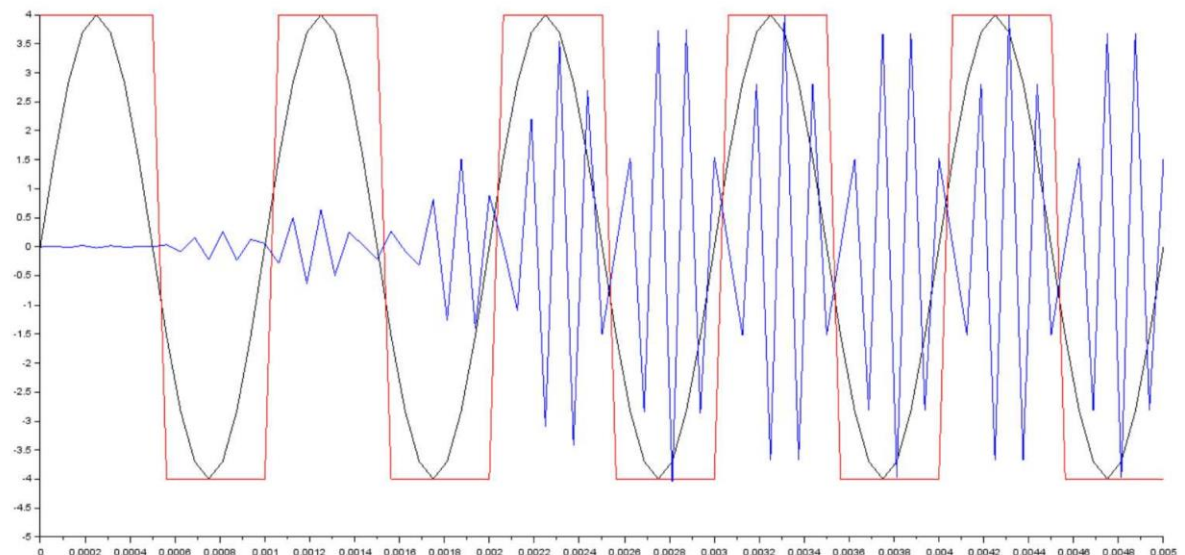
// 6.5kHz / 32 kHz = 0.203125
// 7kHz / 32 kHz = 0.21875
// 7.5kHz / 32 kHz = 0.234375
BD = wfir('bp', 50, [0.200 0.235], 'hm', [0 0]);
```

```
BD = poly(BD, 'z', 'coeff');
BD = horner(BD, (1/%z));
BD = syslin('d', BD);
```

```
BD = 4 * flts(stap, BD)
plot2d(t, BD, style=[color("blue")])
```

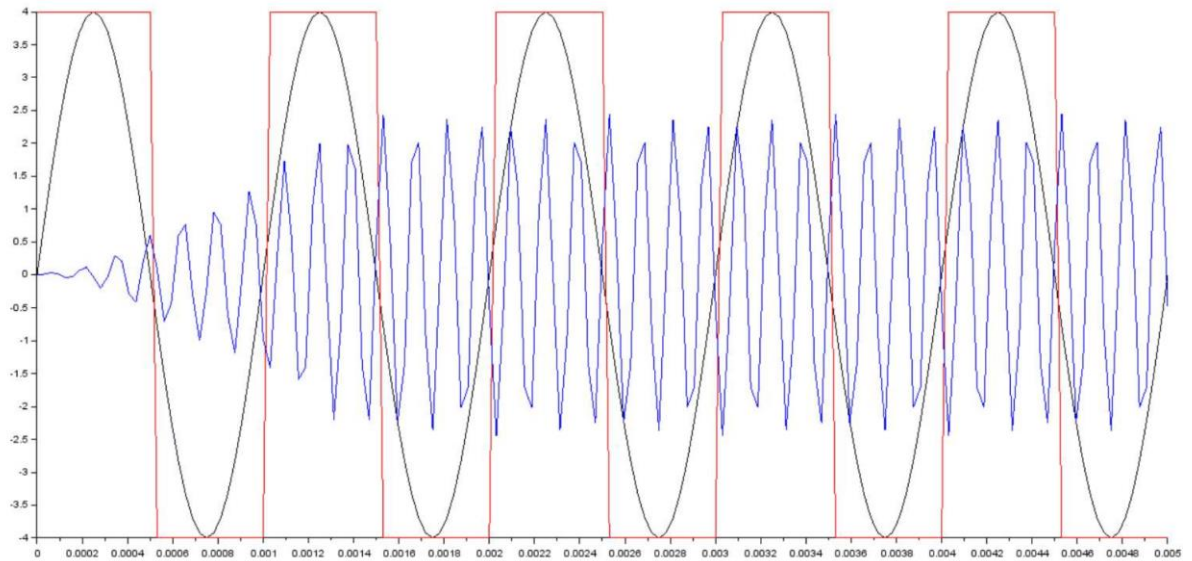
Toon aan met screenshots dat je code werkt

16 kHz samplefrequentie:



De vervorming komt doordat de Niquist frequentie 8 kHz is. Waardoor het 7 kHz signaal maar 1 à 2 meetpunten heeft per periode. Bij een samplefrequentie van 32 kHz (Niquist van 16 kHz) is het signaal duidelijker te zien (en is er minder vervorming).

32kHz samplefrequentie;



2 Opgave 2:

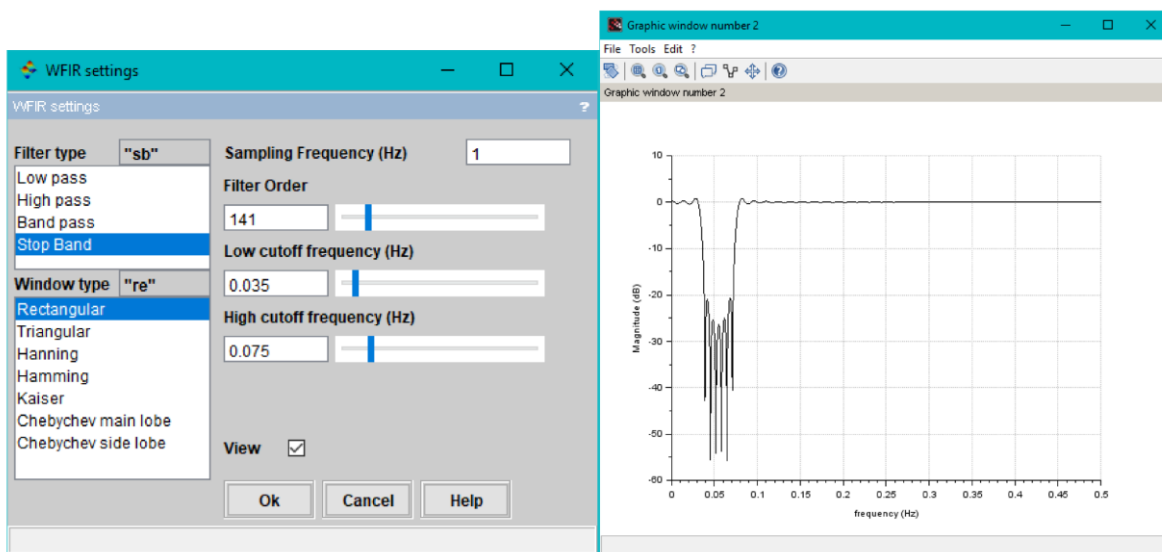
Een groep signalen bestaat uit volgende frequenties:

- Groep A : frequenties tussen 20 Hz en 30 Hz
- Groep B : frequenties tussen 40 Hz en 70 Hz
- Groep C : frequenties tussen 120 Hz en 150 Hz

Gevraagd:

- a) Ontwerp via `wfir()` een digitale frequentieselectieve filter die de signalen van groep B onderdrukt met minstens 20 dB ten opzichte van de andere signalen.
- b) Toon aan in het tijdsdomein dat deze signalen onderdrukt worden via volgende testsignalen:
 - Testsignaal 1 bestaat uit een signaal van 25 Hz (2 Vp) en een signaal van 42 Hz (4 Vp)
 - Testsignaal 2 bestaat uit een signaal van 42 Hz (4 Vp) en 130 Hz (1 Vp)
 - Testsignaal 3 bestaat uit een signaal van 25 Hz (2 Vp), 42 Hz, 65 Hz (4 Vp) en 125 Hz (2 Vp)
- c) Toon in het frequentiespectrum de frequentiekaractersitiek van je filter en gebruikt testsignaal 3 om aan te geven dat de frequenties van de signalen van groep B wel degelijk onderdrukt worden.

Geef hier via screenshots (instellingen en karakteristiek) hoe je de filter hebt ingesteld via het `wfir()`-interactievenster



Geef hier je code voor weergave van de filterresponse in het tijdsdomein (met testsignalen 1 en 2 en 3)

```
clf
```

```
t=0:1/1000:0.5
```

```
//signalen 25Hz (2V), 42Hz (4V)
```

```
testsig1=(2*sin(2*pi*25*t))+ (4*sin(2*pi*42*t))
```

```
//signalen 42Hz (4V), 130Hz (1V)
```

```
testsig2=(4*sin(2*pi*42*t))+ (1*sin(2*pi*130*t))
```

```
//signalen 25Hz (2V), 42Hz (4V), 65Hz (4V), 125Hz (2V)
```

```
testsig3=(2*sin(2*pi*25*t))+ (4*sin(2*pi*42*t))+ (4*sin(2*pi*65*t))+ (2*sin(2*pi*125*t))
```

```
[coeffi, amp,frequ]=wfir('sb',141,[0.035 0.075], 're', [0 0])
```

```
SB_poly = poly(coeffi, 'z', 'coeff');
```

```
SB_functie = horner(SB_poly, 1/%z);
```

```
SB_sys = syslin('d', SB_functie);
```

```
SB_out1 = flts(testsig1, SB_sys)
```

```
SB_out2 = flts(testsig2, SB_sys)
```

```
SB_out3 = flts(testsig3, SB_sys)
```

```
//zwart -> testsignaal
```

```
//rood -> filter signaal
```

```
//groen -> berekend signaal
```

```
subplot(311)
```

```
title("testsignaal 1 (25Hz, 42Hz)")
```

```
ylabel("spanning (V)")
```

```
xlabel("tijd (ms)")
```

```
plot2d(t*1000, testsig1)
```

```
plot2d(t*1000, SB_out1, style=[color("red")])
```

```
plot2d(t*1000, 2*sin(2*pi*25*t), style=[color("green")])
```

```
legend(["testsignaal 1", "filter signaal", "berekend signaal"])
```

```
subplot(312)
```

```
title("testsignaal 2 (42Hz, 130Hz)")
```

```
ylabel("spanning (V)")
```

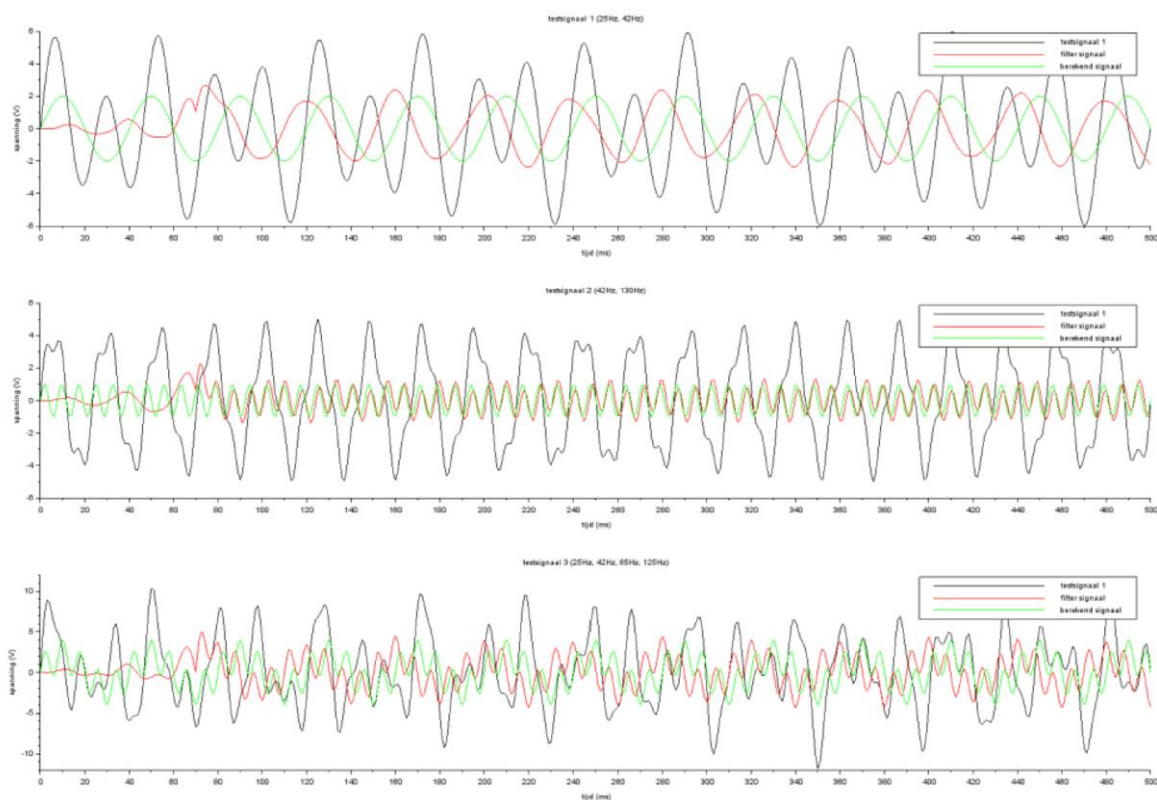
```
xlabel("tijd (ms)")
```

```

plot2d(t*1000, testsig2)
plot2d(t*1000, SB_out2, style=[color("red")])
plot2d(t*1000, 1*sin(2*%pi*130*t), style=[color("green")])
legend(["testsignaal 1", "filter signaal", "berekend signaal"])
subplot(313)
title("testsignaal 3 (25Hz, 42Hz, 65Hz, 125Hz)")
ylabel("spanning (V)")
xlabel("tijd (ms)")
plot2d(t*1000, testsig3)
plot2d(t*1000, SB_out3, style=[color("red")])
plot2d(t*1000, (2*sin(2*%pi*25*t))+(2*sin(2*%pi*125*t)), style=[color("green")])
legend(["testsignaal 1", "filter signaal", "berekend signaal"])

```

Geef hier de grafiek van de filterresponse in het tijdsdomein



Geef hier de code voor de filterresponse in het frequentiedomein (via testsignaal 3) te samen met de frequentiekarakteristiek van de filter.


```

//a)
scf(0)
clf

t=0:1/1000:0.5

//signalen 25Hz (2V), 42Hz (4V)
testsig1=(2*sin(2*pi*25*t))+(4*sin(2*pi*42*t))
//signalen 42Hz (4V), 130Hz (1V)
testsig2=(4*sin(2*pi*42*t))+(1*sin(2*pi*130*t))
//signalen 25Hz (2V), 42Hz (4V), 65Hz (4V), 125Hz (2V)
testsig3=(2*sin(2*pi*25*t))+(4*sin(2*pi*42*t))+(4*sin(2*pi*65*t))+(2*sin(2*pi*125*t))

[coeffi,amp,frequ]=wfir('sb',141,[0.035 0.075], 're', [0 0])
SB_poly = poly(coeffi, 'z', 'coeff');
SB_functie = horner(SB_poly, 1/%z);
SB_sys = syslin('d', SB_functie);
SB_out1 = flts(testsig1, SB_sys)
SB_out2 = flts(testsig2, SB_sys)
SB_out3 = flts(testsig3, SB_sys)

//zwart -> testsignaal
//rood -> filter signaal
//groen -> berekend signaal

subplot(311)
title("testsignaal 1 (25Hz, 42Hz)")
ylabel("spanning (V)")
xlabel("tijd (ms)")
plot2d(t*1000, testsig1)
plot2d(t*1000, SB_out1, style=[color("red")])
plot2d(t*1000, 2*sin(2*pi*25*t), style=[color("green")])
legend(["testsignaal 1", "filter signaal", "berekend signaal"])
subplot(312)
title("testsignaal 2 (42Hz, 130Hz)")
ylabel("spanning (V)")
xlabel("tijd (ms)")
plot2d(t*1000, testsig2)
plot2d(t*1000, SB_out2, style=[color("red")])
plot2d(t*1000, 1*sin(2*pi*130*t), style=[color("green")])
legend(["testsignaal 1", "filter signaal", "berekend signaal"])
subplot(313)
title("testsignaal 3 (25Hz, 42Hz, 65Hz, 125Hz)")
ylabel("spanning (V)")
xlabel("tijd (ms)")
plot2d(t*1000, testsig3)
plot2d(t*1000, SB_out3, style=[color("red")])
plot2d(t*1000, (2*sin(2*pi*25*t))+(2*sin(2*pi*125*t)), style=[color("green")])
legend(["testsignaal 1", "filter signaal", "berekend signaal"])

//c)
scf(1)
clf

N=size(t, "n")
freq_FFT=abs(fft(testsig3))
f=1000*(0:(N/2))/N
n=size(f, "n")
title("frequentie response + filter response")
xlabel("frequentie (Hz)")
ylabel("vermogen mV")
plot(f, freq_FFT(1:n))
plot(frequ*1000, amp*N, 'r')
legend(["testsignaal 3 frequentie response", "filter response"])

```

