

DSP

Laboopdracht 5

IIR-filters & Windowed inc filters

Ing Patrick Van Houtven

s]

Labo 05 : IIR-filtering en windowed sinc filtering

1 Opgave 1

Ontwerp een IIR-filter die de 5^{de} harmonische uit een blokgolf filtert. De blokgolf heeft een amplitude van 4 V en de frequentie van de blokgolf is gelijk aan 100 Hz. Gebruik als samplefrequentie 2 kHz.

Tip:

Je kan vanuit een sinusgeneratie gemakkelijk een blokgolf maken in scilab via de functie "squarewave".

Schrijf hier je code voor de gevraagde filter:

```
clf
//sinus 4V amplitude
//frequentie 100Hz
//samplefrequentie 2kHz

t= 0:1/2000:(1/100)*16

sin_100Hz=4*(sin(2*%pi*100*t))

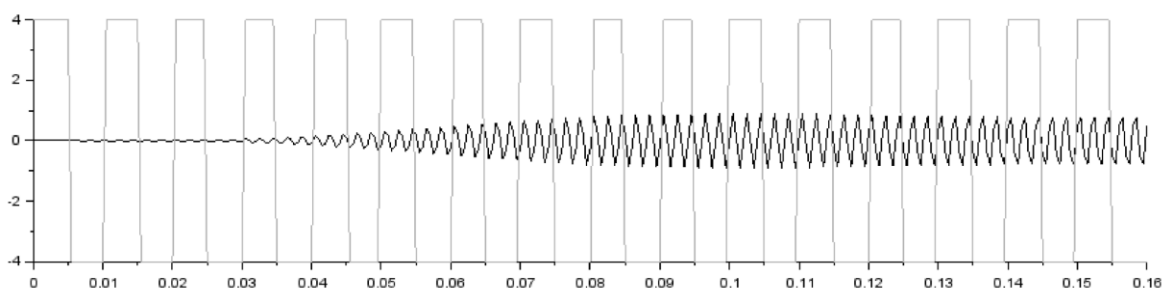
blok_100Hz=4* squarewave(0.5*sin_100Hz)
subplot(311)
plot2d(t,sin_100Hz)
plot2d(t,blok_100Hz)

BP_IIR = iir(5, 'bp', 'butt', [0.245 0.255], [0 0])

BP_resp = flts(blok_100Hz, BP_IIR)

subplot(312)
plot2d(t,BP_resp)
plot2d(t,blok_100Hz)
```

Toon aan met screenshots dat je code werkt



μ

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:

Ontwerp een windowed sinc filter volgens Blackman die de frequenties 120-150 Hz er uit filtert via het concept HD-filter. De samplefrequentie bedraagt 1 kHz en 70 Hz is de eerste frequentie die volledig onderdrukt moet worden.

Geef hier aan hoe je de afsnijfrequentie hebt bepaald:

Aangezien 70 Hz volledig onderdrukt moet worden, nemen we een frequentie die daar dicht bij ligt. 78 Hz is de keuze.

Geef hier aan hoe je het aantal punten van de filterkernel hebt bepaald:

$$78\text{Hz} - 70\text{Hz} = 8\text{Hz}$$

$$\frac{8\text{Hz}}{1\text{kHz}} = 8 * 10^{-3}$$

$$M = \frac{4}{BW} = \frac{4}{8 * 10^{-3}} = 500$$

$$\text{Dichtstbijzijnde macht van 2 is } 2^9 = 512$$

Geef hier in grote stappen weer hoe je de filterkernel hebt bepaald

1. Bepalen van afsnijfrequentie en aantal meetpunten.
 - a. $f_c = 78\text{ Hz}$
 - b. $M = 500$
2. Bepalen welke soort filter er gebruikt wordt en zijn formule toepassen.
 - a. Blackman
 - b. $h(i) = K \frac{\sin(2\pi f_c(i-M/2))}{i-M/2} \left[0.42 - 0.5 \cos\left(\frac{2\pi}{M}\right) + 0.08 \cos\left(\frac{4\pi i}{M}\right) \right]$
3. De filter kernel normaliseren.
4. De filter inverteren (omzetten van Laagdoorlaat naar Hoogdoorlaat).
5. De waarde van het nulpunt +1 doen.

Geef hier de code weer om de filterkernel op te bouwen

```
// Genormaliseerde afsnij frequentie 78Hz / 1kHz = 0.078
FC = 0.078
M = 500
HD = zeros(1 : M + 1)
for i = 1 : M + 1
    if ((i - M / 2) - 1) == 0 then
        HD(i) = 2 * %pi * FC;
    else
        HD(i) = sin(2 * %pi * FC * ((i - M / 2) - 1)) / ((i - M / 2) - 1);
    end
    HD(i) = HD(i) * (0.42 - (0.50 * cos(2 * %pi * i) / M) + (0.08 *
cos(4 * %pi * i) / M));
end
```

```
som = 0
for i = 1 : M + 1
    som = som + HD(i)
end
for i = 1 : M + 1
    HD(i) = HD(i) / som
end
for i = 1 : M + 1
    HD(i) = -HD(i)
end

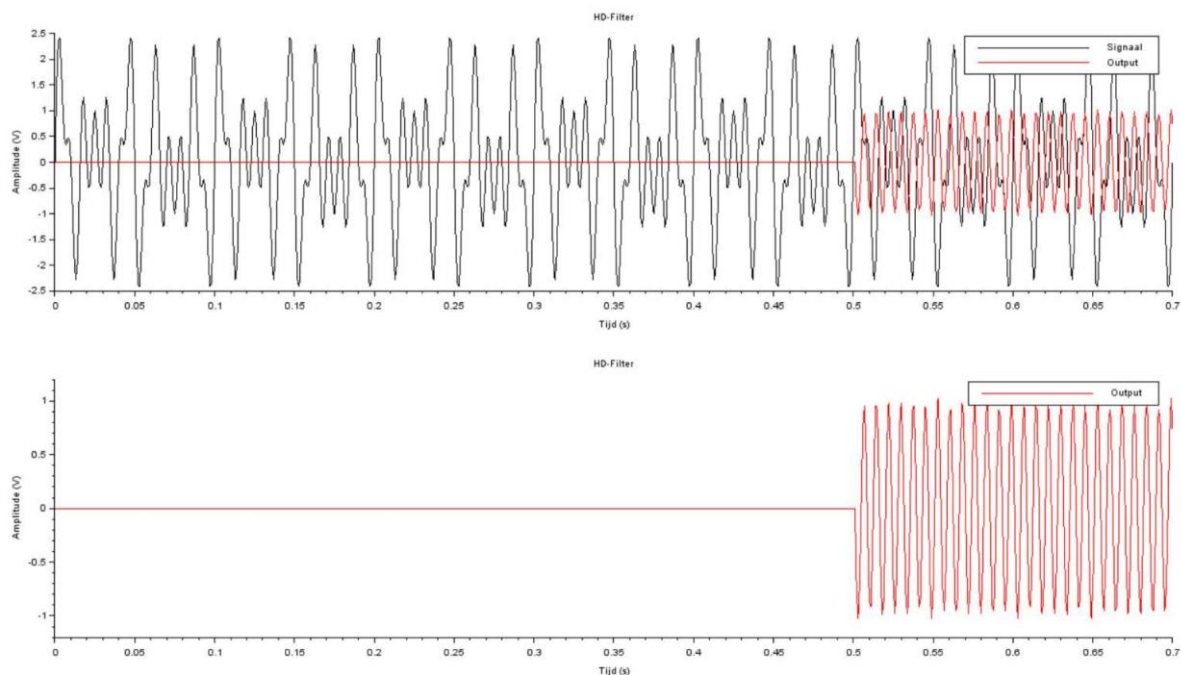
HD(M / 2 + 1) = HD(M / 2 +
1) + 1
```

Genereer een testsignaal met de frequenties 70 Hz, 50 Hz en 130 Hz

```
t = [0 : 1D-3 : 7D-1]
sin50hz = sin(2 * %pi * 50 * t);
sin70hz = sin(2 * %pi * 70 * t);
sin130hz = sin(2 * %pi * 130 * t);
signaal = sin50hz + sin70hz + sin130hz;

output = zeros(1 : length(t))
for i = M + 2 : length(t)
    for j = 1 : M + 1
        output(i) = output(i) + signaal(i - j) * HD(j)
    end
end
plot(t, output)
```

Toon hier de filterresponse in het tijdsdomein en toon aan dat je filter werkt



3 Opgave 3:

Dezelfde opgave als 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:

Maak een windowed -sinc filter die uit de 3 groepen van signalen enkel de groep B doorlaat.

Geef hier aan hoe je de afsnijfrequenties hebt bepaald:

Aangezien we 40 - 70 Hz volledig moeten doorlaten, kiezen we een frequenties die in de nabijheid liggen.

Voor 40 Hz nemen we 32 Hz. Hiermee kan je het hoogdoorlaatgedeelte testen en moet alles onder 40 Hz onderdrukt worden. Voor de 70 Hz-grens kiezen we 78 Hz. Hiermee kan het laagdoorlaatgedeelte getest worden vermits alle frequenties boven 70 Hz onderdrukt moeten worden.

Geef hier aan hoe je het aantal punten van de filterkernel hebt bepaald:

$$78\text{Hz} - 70\text{Hz} = 8\text{Hz} \text{ \& } 40\text{Hz} - 32\text{Hz} = 8\text{Hz}$$

$$\frac{8\text{Hz}}{1\text{kHz}} = 8 * 10^{-3}$$

$$M = \frac{4}{BW} = \frac{4}{8 * 10^{-3}} = 500$$

Dichtstbijzijnde macht van 2 is $2^9 = 512$

Geef hier in grote stappen weer hoe je de filterkernel hebt opgebouwd

1. Bepalen van afsnijfrequentie en aantal meetpunten.
 - a. $f_{\text{LP}} = 32 \text{ Hz}$
 - b. $f_{\text{HP}} = 78 \text{ Hz}$
 - c. $M = 500$
2. Bepalen welke soort filter er gebruikt wordt en zijn formule toepassen.
 - a. Blackman
 - b.
$$h(i) = K \frac{\sin(2\pi f_c(i-M/2))}{i-M/2} \left[0.42 - 0.5 \cos\left(\frac{2\pi i}{M}\right) + 0.08 \cos\left(\frac{4\pi i}{M}\right) \right]$$
3. Twee laagdoorlaatfilterkernels aanmaken.
4. Deze filterkernels normaliseren.
5. De 2de filter inverteren (omzetten van Laagdoorlaat naar Hoogdoorlaat).
6. De waarde van het nulpunt, van de 2de filter, +1 bijtellen.
7. De 2 filters bij elkaar optellen (verkrijgt bandsperfilter).
8. De bandsperfilter inverteren (verkrijgt Banddoorlaat).
9. De waarde van het nulpunt, van de nieuwe filter, +1 doen.

Geef hier de code weer om de filterkernel op te bouwen

```
M = 500
// Laagdoorlaat
FCL = 0.032
LD = zeros(1 : M + 1)
for i = 1 : M + 1
    if ((i - M / 2) - 1) == 0 then
        LD(i) = 2 * %pi * FCL;
    else
        LD(i) = sin(2 * %pi * FCL * ((i - M / 2) - 1)) / ((i - M / 2) - 1);
    end
    LD(i) = LD(i) * (0.42 - (0.50 * cos(2 * %pi * i) / M) + (0.08 * cos(4 * %pi * i) / M));
end
```

```

som = 0
for i = 1 : M + 1
    som = som + LD(i)
end
for i = 1 : M + 1
    LD(i) = LD(i) / som
end

// Hoogdoorlaat
FCH = 0.078
HD = zeros(1 : M + 1)
for i = 1 : M + 1
    if ((i - M / 2) - 1) == 0 then
        HD(i) = 2 * %pi * FCH;
    else
        HD(i) = sin(2 * %pi * FCH * ((i - M / 2) - 1)) / ((i - M / 2) - 1);
    end
    HD(i) = HD(i) * (0.42 - (0.50 * cos(2 * %pi * i) / M) + (0.08 * cos(4 * %pi * i) / M));
end

som = 0
for i = 1 : M + 1
    som = som + HD(i)
end
for i = 1 : M + 1
    HD(i) = HD(i) / som
end
for i = 1 : M + 1
    HD(i) = -HD(i)
end
HD(M / 2 + 1) = HD(M / 2 + 1) + 1

BD = HD + LD
for i = 1 : M + 1
    BD(i) = -BD(i)
end
BD(M / 2 + 1) = BD(M / 2 + 1) + 1

```

Genereer een testsignaal met de frequenties 25 Hz, 70 Hz, 50 Hz en 130 Hz

```

t = [0 : 1D-3 : 7D-1]
sin25hz = sin(2 * %pi * 25 * t);
sin50hz = sin(2 * %pi * 50 * t);
sin70hz = sin(2 * %pi * 70 * t);
sin130hz = sin(2 * %pi * 130 * t);
signaal = sin25hz + sin50hz + sin70hz + sin130hz;

output = zeros(1 : length(t))
for i = M + 2 : length(t)
    for j = 1 : M + 1
        output(i) = output(i) + signaal(i - j) * BD(j)
    end
end

```

Toon hier de filterresponse in het tijdsdomein en toon aan dat je filter werkt

