

FIR filters

PAoWV

Inleiding

We hebben onze zendmactiging uitgereikt gekregen om te experimenteren en daarmee onszelf te ontwikkelen. Ik laat in dit artikel zien hoe je zelf digitale filters kunt bouwen, en ermee experimenteren.

Nyquist

Nyquist bekeek dat je een analoog signaal kunt bemonsteren, dat wil zeggen regelmatig kijken wat de signaalwaarde is. Dan weet je niks van wat er tussen twee monsters gebeurt, maar als de bandbreedte voldoende beperkt is weet je wel dat het signaal niet heel snel kan veranderen tussen twee monsters in.

Nyquist bedacht en bewees dat als je twee keer zoveel monsters neemt per seconde, als de maximum frequentie die in het signaal voor komt, dat je goed zit. Dat wil zeggen dat je geen informatie verliest en dat je het signaal onvervormd weer uit de monsters kunt halen.

Dat is ook wel makkelijk te verklaren. Stel namelijk dat je s monsters per seconde neemt van gelijkspanning, dan is dat een reeks naaldpulsen met frequentie s Hz en allemaal gelijke amplitude.

Die naalden hebben dus harmonischen op $2s$, $3s$ enzovoorts steeds hoger. Die harmonischen zijn allemaal even sterk als de naalden oneindig smal zijn; dat zijn ze doorgaans niet en dan neemt de amplitude van de harmonischen langzaam af naarmate ze hogere rangorde hebben. Denk maar aan QRN, bliksem is een smalle puls en die bedekt zowat de hele hf-band, dat kun je goed zien op de waterfall display van een websdr bijvoorbeeld. ["klik hier"](#)

Nu gaan we niet gelijkspanning bemonsteren maar een of ander signaal, muziek, spraak, noem maar op, met een bandbreedte van 0 tot B , bijvoorbeeld 300-2800 Hz.

Dat betekent dat de monsters ook in amplitude het verloop van dat signaal volgen. Amplitudemodulatie dus. Als je de amplitude van de naaldpuls halveert, halveren alle harmonischen ook in amplitude. Elke harmonische van de pulsen wordt op dezelfde wijze amplitude gemoduleerd en geeft dus AM zijbanden, tot B onder en B boven die harmonische ook trouwens van 0 tot B , dat als het ware de gelijkstroomcomponent van de in amplitude variërende monsters is.

Nu heb je net geen overlap van zijbanden bijvoorbeeld tussen de bovenzijband van $2s$ en de onderzijband van $3s$, als de harmonischen minimaal $2B$ uit elkaar liggen. De bemonsterfrequentie s moet dus minimaal $2B$ zijn. Je kunt het signaal terugwinnen door een van die AM gemoduleerde harmonischen uit te filteren met een $2B$ breed bandfilter met een diodedetector erachter, maar makkelijker is de band tussen 0 en B te pakken met een laagdoorlatend filter met B als grensfrequentie, waarbij B dus kleiner of theoretisch hoogstens gelijk is aan de halve bemonsterfrequentie f_s dus praktisch $B < f_s/2$.

WAV geluidsbestanden.

Elk monster kun je de grootte van digitaal als getal aangeven. Met 16 bits kun je dan van -2^{15} tot $+2^{15}-1$ dat is van

-32768 tot plus 32767 aangeven. Dat bemonsteren en de grootte van de monsters als een getal weergeven heet Pulscode-modulatie, oftewel PCM.

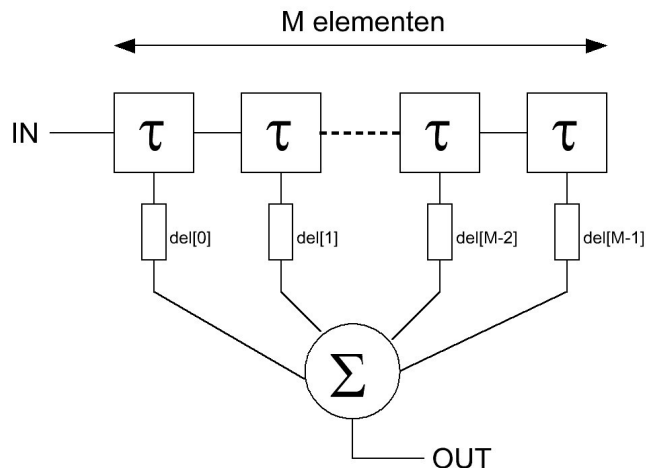
Je kunt dat natuurlijk ook stereo doen met 2 of meer kanalen, en meer of minder monsters per seconde nemen of een ander aantal bits per monster gebruiken. Daarom wordt dat in de eerste 44 bytes van een wav bestand aangegeven en dat wordt de header genoemd. Hoe de header van een wav bestand in elkaar zit kun je op Internet vinden, bijvoorbeeld op de volgende website: ["Klik hier"](#)

Wil je spraak bewerken dan zijn 8000 monsters per seconde elk in 16 bits getallen ruim voldoende. Dan kun je frequenties tot 4000 Hz weergeven. Telefoon-landlijn werkte met 8000 monsters/s elk monster slechts 8 bits, dus 64 kbits.s voor een spraakkanaal..

Digitale Filters

Stel je stopt een korte puls eenmalig in een digitaal filter, dan komen er doorgaans een aantal pulsen met de bemonsterfrequentie uit. Dat reeksje pulsen heet dan de impuls response, omdat een schop, een puls als input, een aantal pulsen als reactie geeft.

Weet je die impuls response die je wilt maken, dan is een schema dat dit doet snel verzonnen.



Namelijk een schuifregister dat de grootte van de pulsen als 16 bits getal per geheugenvakje kan bevatten en die inhoud van die vakjes wordt opgeteld met een weegfactor (versterkingsfactor) en de optelling is de output. De schuifklok is de bemonsterfrequentie

Dat lijkt hocus pocus maar dat is het niet; stop je er een puls in, dan geeft dat schuifregister maar een van 0 verschillend pulsje ter grootte van de weegfactor, dat is die weegweerstand naar de opteller. Bij elke klokpuls schuift dat inputpulsje een hokje verder en geeft dan op de output de grootte van de bij dat hokje horende weegfactor weer. De weegweerstand, hebben dan in volgorde de grootte van de gewenste output pulsreeks.

Een signaal is doorgaans opgebouwd uit een groot aantal monsters, elk monster geeft de impulsresponse die opgeteld wordt bij de impulsresponse van vorige monsters.

Dan krijg je dus de gefilterde output op het signaal dat op de input wordt aangeboden. Dat proces heet convolutie.

Nu kun je opmerken dat je niet een bepaalde impulsresponse wilt maar een bepaalde frequentie karakteristiek. Die hangen samen; de frequentie karakteristiek is de Fouriertransformatie van de impulsresponse en omgekeerd.

Experimenteren

Als je wat kunt programmeren met Basic peek & poke of een andere programmeertaal kun je met die header ervoor dus zelf afspeelbare wav files maken, en daar filters op loslaten. Trage computer geeft niks, want dan duurt het langer om een wav file te maken maar die kun je daarna gewoon afspelen om het resultaat te beluisteren.

Laagdoorlatende filters.

Laten we eens gek doen en eisen dat we tot een frequentie f_g doorlaten en daarboven helemaal niks. Zo'n filter wordt wel een brick wall filter genoemd, doorlaat demping 0 en alles voorbij de grensfrequentie f_g demping oneindig. We kunnen zoals gezien dat doen tot de halve bemonsterfrequentie, want hogere frequenties mogen niet voorkomen. Daarom wordt de frequentie karakteristiek aangegeven als de demping in dB als functie van de frequentie f die ligt tussen 0 en $f_s/2$, waarbij f_s dus de bemonsterfrequentie is.

Hoe ziet de impulsresponse van zo'n brickwall er uit? Fouriertransformatie toepassen op de brick wall frequentie karakteristiek levert voor de impulsresponse de formule

$$\sin(2\pi f_g t) / (2\pi f_g t)$$

Je weet hoe een sinus eruit ziet en hier zie je dat door die toenemende noemer bij toenemende t die steeds kleinere amplitude krijgt. Voor $t=0$ is de limietwaarde 1. Je moet namelijk niet 0 door 0 proberen te delen, wat dat is het moment dat het socialisme blijkt te falen en daar zitten die lieden niet op te wachten. Die besteden hun leven namelijk nuttig door het door anderen verdiende geld te besteden naar hun eigen communautaire goeddunken.

Die sinc functie $\sin(xt)/xt$, wordt bij toenemende t nooit 0, dus laten we eens grofstoffelijk doen en wel 129 schuifelementen nemen. 64 links van het midden voor $t=0$. en 64 rechts en het midden zelf maakt samen 129. Met negatieve tijden kunnen we niet uit de voeten dus dat compenseren we door $t=0$ 65 monstertijden naar rechts te schuiven.

Dat zijn wel erg veel elementen, die 129 stuks, als je het zou moeten solderen, maar besef wel dat je alles afkapt beneden pakweg 1/64 grootte dus eigenlijk kun je nog niet al te veel van die brick wall verwachten want 1/64 is toch minder dan 40 dB.

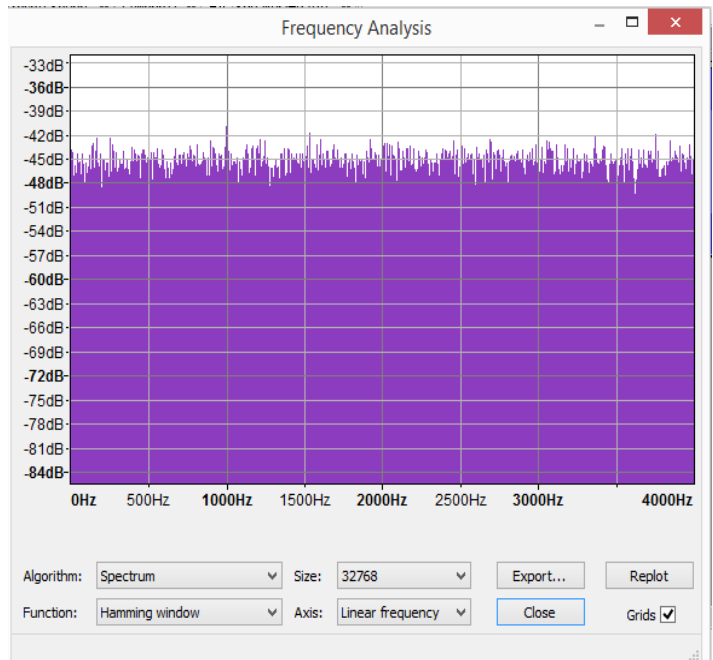
We gaan het proberen, want een aap die nooit op zijn bek valt heeft geen NATO-Z lippen. Het programma ziet er simpel uit, dat die 129 vakjes schuifregister en optellen programmeert, en dan willen we natuurlijk de frequentie karakteristiek weten. Kijken hoeveel die afwijkt van de brick wall. Voor de zekerheid: Met $M=129$ zijn de 129 weegfactoren

$$\sin(2\pi f_g \cdot (i - (M-1)/2)) / (i - (M-1)/2)$$

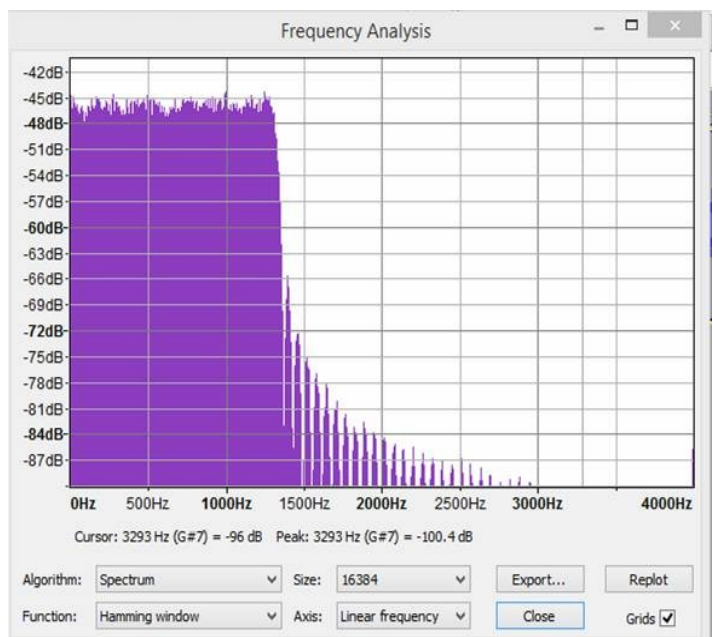
Met i van 0 tot en met 128

Frequentie karakteristiek

Als je een wav bestand maakt met random getallen als monsters, krijg je witte ruis met een spectrum dat vlak is tussen 0 en $f_s/2$. Kijk maar:

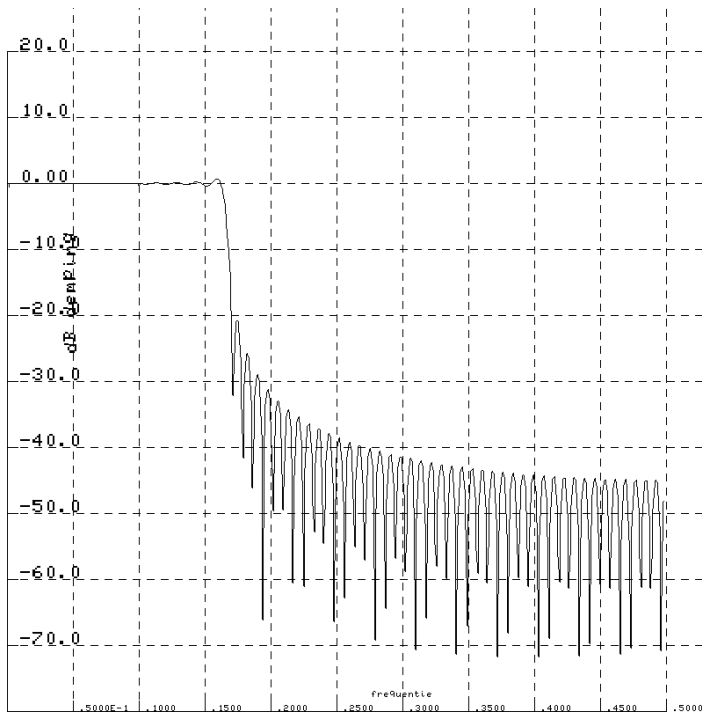


Gaan we die witte ruismonsters in het filter stoppen, dus als input voor het filter gebruiken, dan komt er ruis uit in de vorm van de frequentie karakteristiek. Dat is de makkelijkste methode, want die ruis kun je met Audacity, dat is een freeware programma dat je op Internet kunt vinden, het spectrum laten zien. Dat spectrum heeft dus de vorm van de filterdoorlaatband.



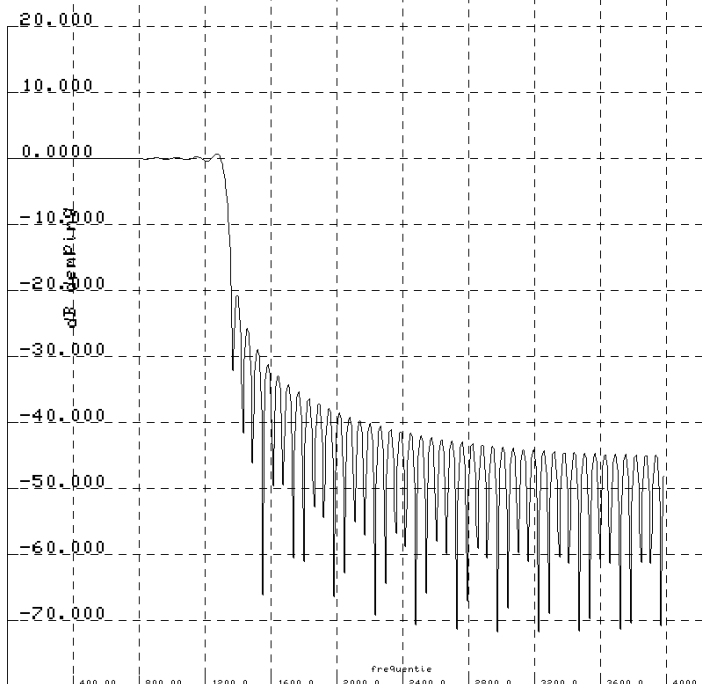
Een tweede methode is een sinusvormig signaal in het filter stoppen in de vorm van monsters van die sinus, en na inslingeren kijken naar de amplitude die er uit komt. Dat doe je dan voor pakweg 500 frequenties tussen 0 en $f_s/2$ en de output amplitude van die 500 punten plot je op een dB schaal.

Bij frequenties dicht bij $f_s/2$ heb je maar twee monsters per sinus dus kan het een tijd duren voor die op een maximum vallen, en vind je daar dus doorgaans wat meer demping dan er werkelijk is.



De derde methode zendt monsters van een sinus in het filter, ook weer voor 500 frequenties, en ook weer doen de eerste M stuks per testfrequentie niet mee om inschakelverschijnselen te vermijden. En dan bepaal je over in dit geval (arbitrair) 9 seconde hoeveel vermogen er in het filter gaat en hoeveel er uit komt, om de demping te bepalen. Ik heb die 9 aangehouden omdat ik dan met Audacity 8 seconde nodig heb om de FFT (Fouriertransformatie) te bepalen over 65536 monsters.

In onderstaande figuur zie je de derde methode toegepast op het hierboven beschreven laagdoorlatende filter met grensfrequentie $f_g = f_s/6$ dus hier $f_g = 1333$ Hz



Zoek de verschillen zou ik zeggen. In ieder geval is de tweede methode die het maximum zoekt na M pulsen input voorladen in het filter, natuurlijk de snelste. Je weet dan na 2M pulsen wat de demping is en bij de derde methode is dat pas na, in dit voorbeeld, 72000 pulsen.

In ieder geval kan duidelijk zijn dat ik over de witteruis spectrummethode met Audacity niet enthousiast ben.

Dit laagdoorlatende filter van 129 schuifhokjes valt een beetje tegen, voor we het gaan verbeteren is het zaak te weten hoe je nu een banddoorlaat, een bandsper en een hoogdoorlatend filter kunt maken.

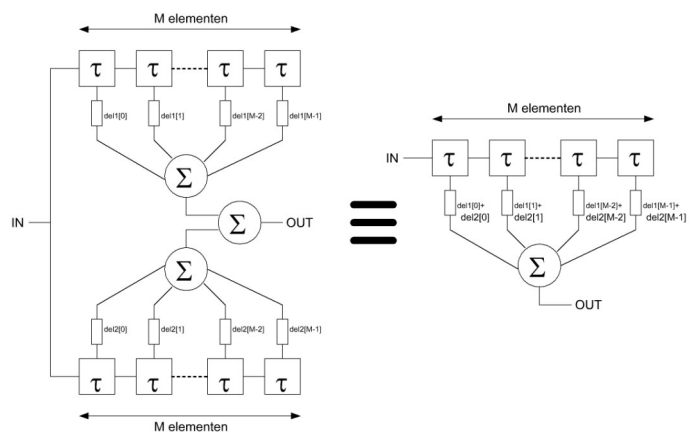
Hoogdoorlatend filter

Je kunt op twee manier een hoogdoorlaat maken.

1. Als je een all-pass maakt - dat is een filter dat alle frequenties ongewijzigd in amplitude doorlaat tussen 0 en $f_s/2$, maar dezelfde vertraging geeft (en fasekarakteristiek heeft) als een lowpass - en je zet die all pass parallel aan de lowpass, maar je trekt de outputs van elkaar af, dan laat de lowpass niks door boven zijn grensfrequentie en de all pass wel, daar zit dus een doorlaatgebied. Beneden de grensfrequentie laat de allpass evenveel door als de low pass, trek je de outputs van elkaar af, dan blijft er niets over.

Nu hoef je geen twee filters te maken want uit onderstaande tekening blijkt dat je door de associatieve wet van de optelling, dat betekent dat $(a+b+c)-(d+e+f) = (a-d)+(b-e)+(c-f)$ die twee samen kunt nemen tot een filter met de weegweerstand de verschillen van de weegweerstand van de afzonderlijke filters.

Een all pass heeft alle elementen 0 behalve het middelste.



(in feite volstaan voor een all pass hier dus 65 elementen). Om op 0 uit te komen bij aftrekken moet de som van de weegfactoren van de lowpass gelijk zijn aan die enige van 0 verschillende middelste van de allpass.

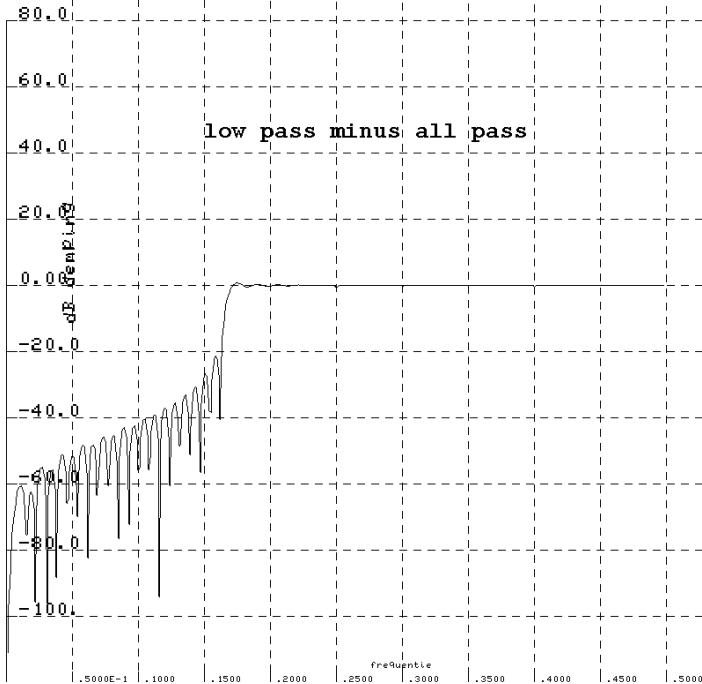
Je kunt de versterking van het filter op 1 brengen in de doorlaat als de som van alle weegweerstand 1 is. Elke berekende weegfactor delen door die som levert het gewenste resultaat. Dat is eenvoudig in te zien door een low pass met gelijkspanning (allemaal even grote monsters) aan te sturen.

2. Je kunt een andere methode toepassen namelijk de weegweerstand vermenigvuldigd met $\sin(f_s/2)$ Dat gebeurt als je elke volgende weegweerstand beurtelings met +1 en -1 vermenigvuldigt. Stop je dan DC pulsen in het filter dan komen er monsters die $f_s/2$ als output geven . Stop je $f_s/2$ in het filter dan komen er DC pulsen uit. Laagfrequent wordt dus hoog en hoog wordt laag. De grensfrequentie die f_g was wordt nu dus $f_s/2 - f_g$ Andere uitleg: je vermenigvuldigt het signaal dan met $\sin(f_s/2)$, dat geeft een boven en onderzijband; de onderzijband valt in de filterdoorlaat en die loopt van $f_s/2$ naar beneden als doorlaat tot de grensfrequentie.

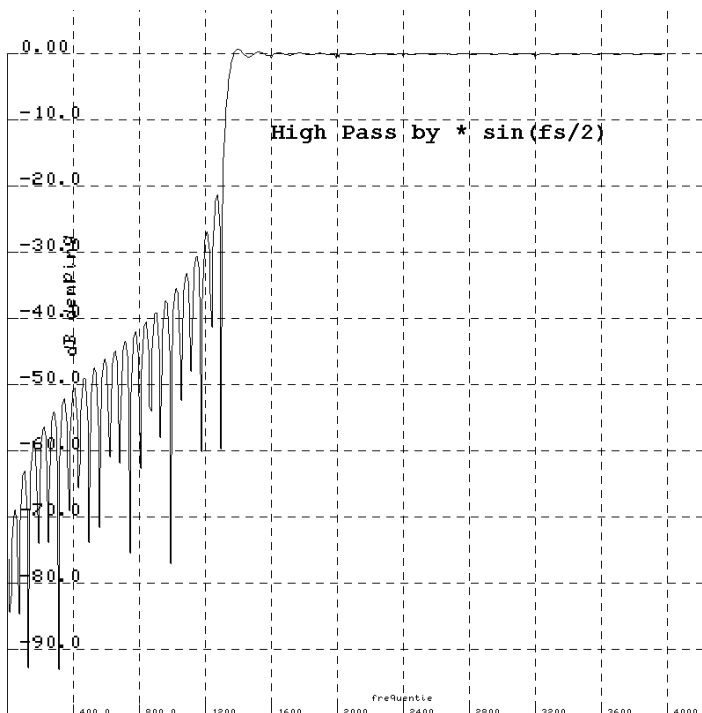
Platen hieronder geven de doorlaat aan voor een 129 lang

filter. Door om de andere weegweerstand van teken te wisselen krijg je dus een hoogdoorlatend filter.

Verskil met de eerste methode is dat als het low pass grensfrequentie f_g is, heeft de high pass by methode 1 ook f_g , maar



bij de tweede methode $f_s/2 - f_g$. Ten behoeve van de vergelijking heb ik ervoor gezorgd dat beide dezelfde grensfrequentie $f_s/6$ hebben.

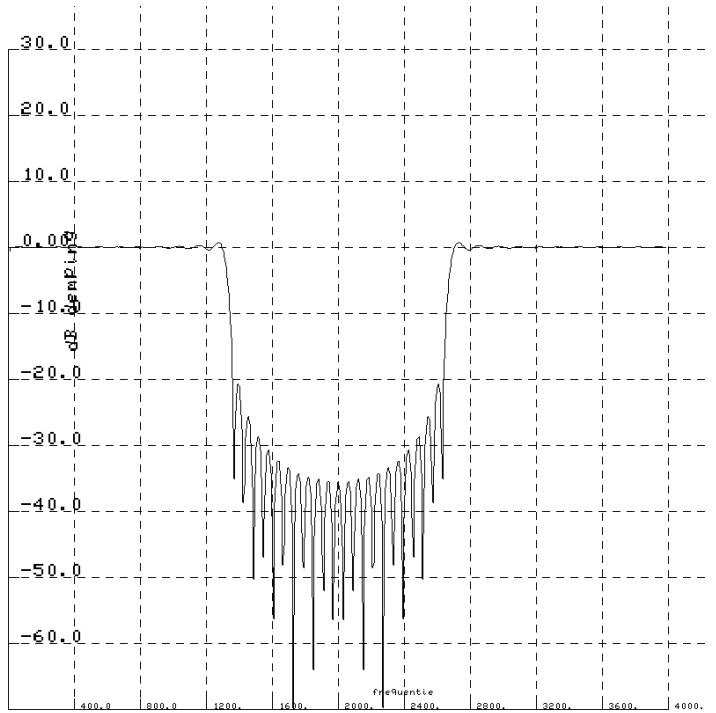


Bandsperfilter

Maak je een laagdoorlaat met grensfrequentie f_g en een hoogdoorlaat met grensfrequentie f_h en zet je die parallel, dan laat het ene filter door van 0 tot f_g en het andere van f_h tot $f_s/2$. Het stuk tussen f_g en f_h wordt niet doorgelaten als $f_h > f_g$. Dat kan weer gecombineerd tot een filter door de weegfactoren van de filters paarsgewijze op te tellen.

Plaatje hieronder van de frequentieresponse van een 129 lang

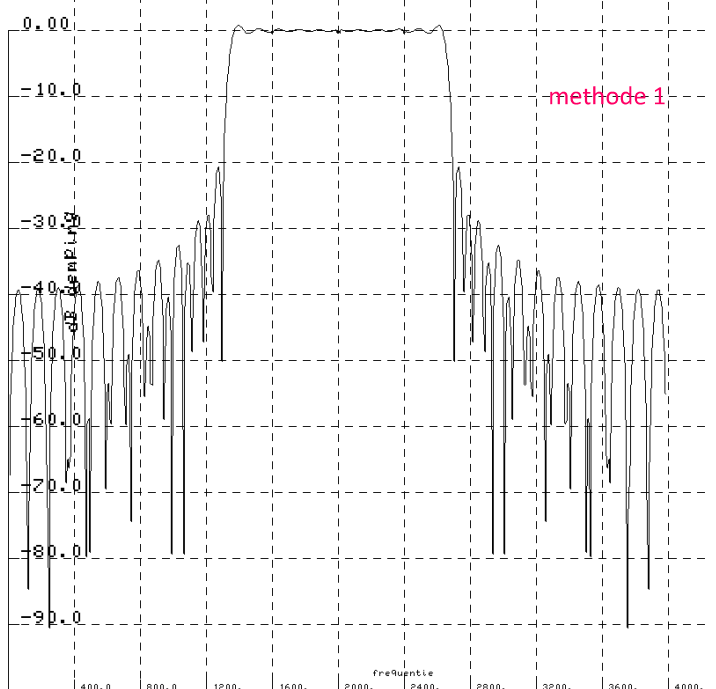
bandsperfilter van $f_s/6$ tot $f_s/3$. Dat plaatje wekt bij mij associaties op met een of andere haatbaard.



Banddoorlaat

Een banddoorlatend filter kun je op twee manieren maken

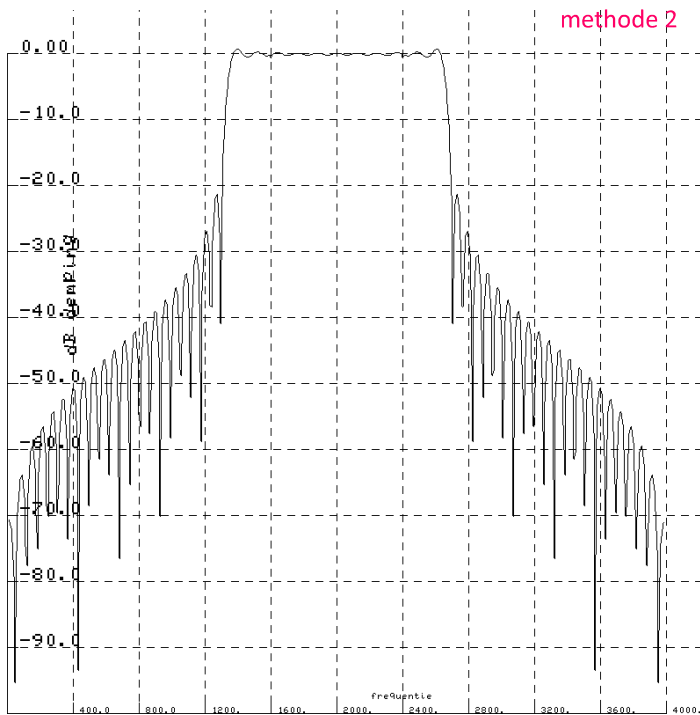
1. Door een bandsper parallel te zetten aan een all pass, en de outputs van elkaar af te trekken. Dat kan weer in een filter gecombineerd van 129 lang, je hoeft dus maar een weegfactor (de middelste) te wijzigen dus om een bandsper om te zetten in een banddoorlaat. Is dat geen uitdaging om een Selectoject te maken of zoiets?



2. Je kunt voor een banddoorlaat ook twee filters in serie zetten, een low pass met grensfrequentie f_g en een high pass met grensfrequentie f_h waarbij dan f_h kleiner moet zijn dan f_g , anders komt er niks door.

Tot nu toe konden we met parallel geschakelde filters die com-

bineren door weegweerstand paarsgewijze op te tellen of af te trekken. Nu echter staan de filters in serie en is de set weegweerstand dubbel zo groot omdat de set de convolutie is van de weegfactoren van beide filters.



Je kunt die bepalen door de waarde van de weegfactoren van het ene filter te sturen door het andere, en wat er dan uitkomt is een dubbel zo lange impulsrespons van de in serie geschakelde filters. Die impulsrespons zijn dan de weegweerstand als je er een filter van wilt maken. Stuur maar een enkele puls door de in serie geschakelde filters, dan geeft het eerste filter zijn impulsresponse af aan het tweede filter en dat dus op zijn beurt de impulsresponse van beide filters in serie oftewel het tweede filter gevoed met de impulsrespons van het eerste filter.

De twee frequentieresponses voor **methode 1 en 2** met dezelfde bandgrenzen zijn op de vorige bladzijde en hierboven getekend.

Blackman en Hamming window

De frequentieresponses zijn verre van brick wall en er is een truc om dat te verbeteren. De oorzaak is het afbreken van de $\sin(x)/x$ tot $x=64$ en $x=-64$ in dit voorbeeld. Brengen we nu een weging aan die de middenpoot hetzelfde laat maar naar buiten toe langzamerhand de zaak naar 0 brengt dan knapt dat een stuk op.

Blackman window heeft als weging:

$$0,42 - 0,5 * \cos \left(2 \pi \frac{i}{M-1} \right) + 0,08 * \sin \left(4 \pi \frac{i}{M-1} \right)$$

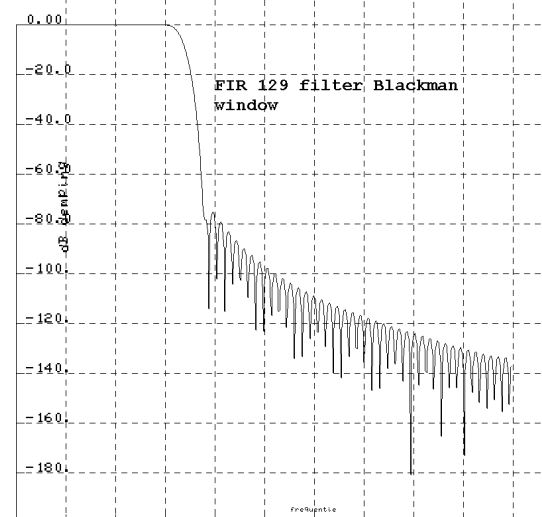
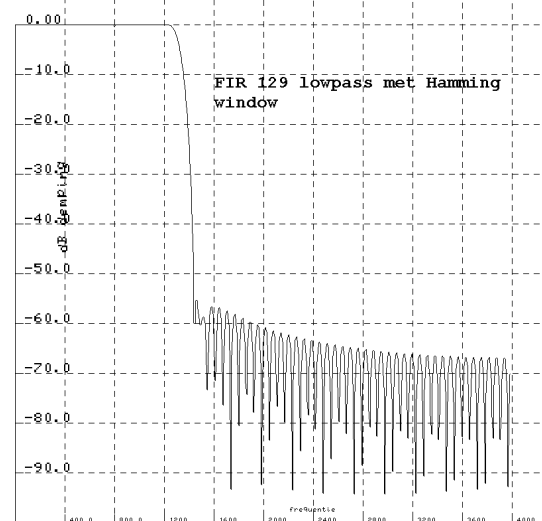
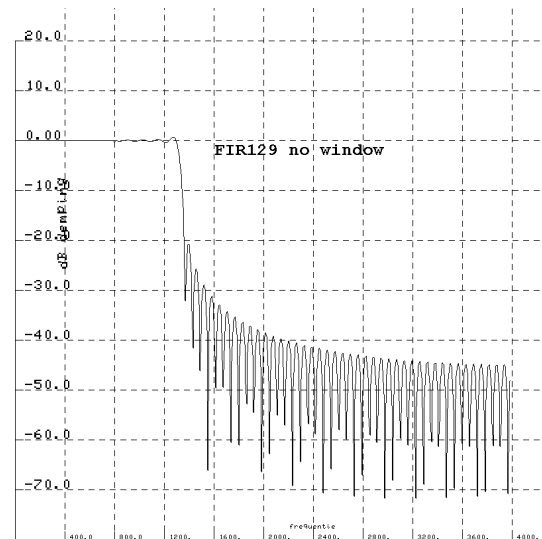
Hamming window heeft als weging:

$$0,54 - 0,46 * \cos \left(2 \pi \frac{i}{M-1} \right)$$

Hierbij moet M oneven zijn, hier is $M=129$. en i is het volgnummer van de correctiefactor van de weegweerstand en die loopt van 0 tot $M-1$.

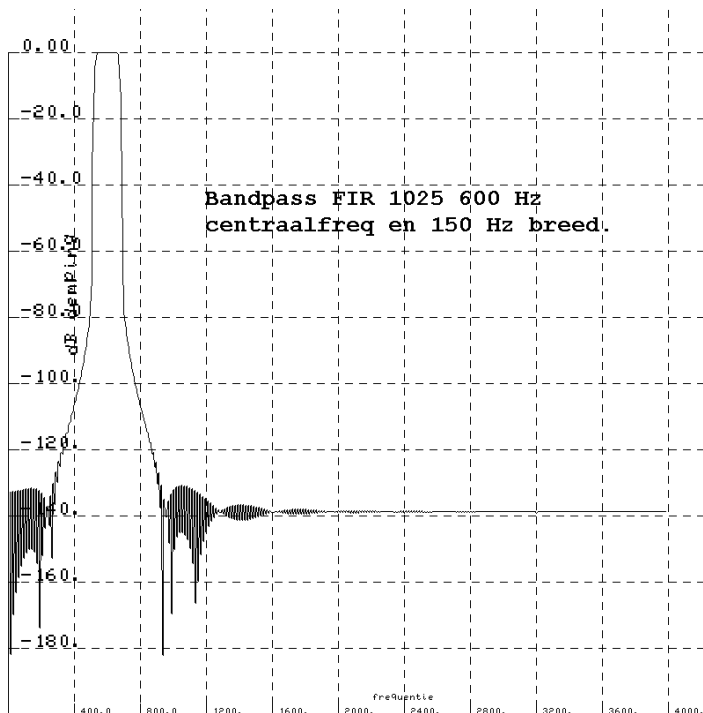
Het eerder genoemde low pass filter met $f_g=f_s/6=1333,3$ Hz, $M=129$ filterelementen zonder window en vervolgens met Hamming window en met Blackman window is ter vergelijking hieronder gezet.

Je ziet dat het ene filter iets steiler loopt maar hogere terugkomers heeft in het sfergebied. Door de hogere demping in het sfergebied is de dB schaal van het Hamming filter verdubbeld, daar dus op letten bij vergelijken.

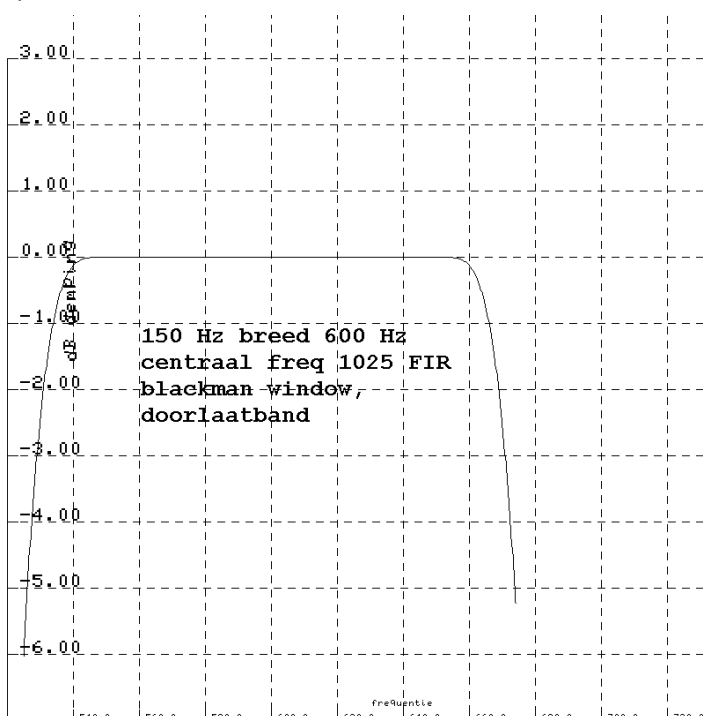


Lange filters

In een programma is het een koud kunstje het filter langer te maken. Ik kies vrij arbitrair 1025 elementen. De vertraging van het filter is dan wel 128 ms bij een samplefrequentie van 8000. Voorts zijn er per sample dan 8 keer zoveel vermenigvuldigingen en optellingen nodig. Als je het real time wilt realiseren. Neem ik 1025 elementen. met een Blackman window dan ziet een filter van 150 Hz breed op een centraalfrequentie van 600Hz er als volgt uit:



Je kunt alleen het 150 Hz brede doorlaatgebied plotten, dat ziet er als hieronder uit. De bandbegrenzing zijn kennelijk de 6 dB punten.



Tot slot enkele geluidsvoorbeelden

Telegrafie A1A, 12 wpm op 600 Hz in witte bandbepaalde ruis tussen 300 en 2800 Hz, met een S/N verhouding van -10dB en vervolgens hetzelfde signaal en ruis vermogen, maar nu door een filter van slechts 50 Hz breed op de 6 dB punten, 1025 lang, zonder demping op de 600 Hz centraalfrequentie, met een Blackman window.

Dat filter verzwakt het 12 wpm morsesignaal 0,09 dB en het ruisvermogen 17,68 dB, zodat de S/N verhouding 17,59 dB verbetert; hoe de neembaarheid wijzigt kunt u zelf beoordelen.

Er is ook nog een filter van 250 Hz breed aangelegd, een bandbreedte voor CW filters die commercieel verkrijgbaar is. De ruis krijgt dan dezelfde klankkleur als het signaal, wat de neembaarheid niet ten goede komt. Weer zelfde signaal en ruis als input gebruikt. Het resultaat is te horen op cw250Hz.wav.

Na het filter is de ruis 12,49 dB minder en het signaal 0,01 dB verzwakt, zodat de S/N 2,47 dB is.

Vergeet bij het beoordelen van de signalen niet het audiovolume te regelen, zodat u ook bemerkt hoe dat van invloed is op de neembaarheid door de fysiologische eigenschappen van het gehoor.

Die geluidsbestanden kunt u downloaden op website

["Klik hier"](#)

DXCC Most Wanted List top 25 per 28 august 2018

Rank	Prefix	Entity Name
1.	P5	DPRK (NORTH KOREA)
2.	3Y/B	BOUVET ISLAND
3.	FT5/W	CROZET ISLAND
4.	BS7H	SCARBOROUGH REEF
5.	CE0X	SAN FELIX ISLANDS
6.	BV9P	PRATAS ISLAND
7.	KH7K	KURE ISLAND
8.	KH3	JOHNSTON ISLAND
9.	VK0M	MACQUARIE ISLAND
10.	FT5/X	KERGUELEN ISLAND
11.	3Y/P	PETER 1 ISLAND
12.	FT/G	GLORIOSO ISLAND
13.	YV0	AVES ISLAND
14.	KH4	MIDWAY ISLAND
15.	ZS8	PRINCE EDWARD & MARION ISLANDS
16.	VP8O	SOUTH ORKNEY ISLANDS
17.	PY0T	TRINIDADE & MARTIM VAZ ISLANDS
18.	PY0S	SAINT PETER AND PAUL ROCKS
19.	VP6/D	DUCIE ISLAND
20.	SV/A	MOUNT ATHOS
21.	KH1	BAKER HOWLAND ISLANDS
22.	KP5	DESECHEO ISLAND
23.	VP8S	SOUTH SANDWICH ISLANDS
24.	EZ	TURKMENISTAN
25.	JD/M	MINAMI TORISHIMA

Meer te vinden op : <https://secure.clublog.org/mostwanted.php>