

De Fourieranalyse (1)

Na diverse boegschoten in de vorige acht delen van deze reeks, komen we nu langzaam maar zeker toe aan het echte DSP-werk. In deze aflevering starten we een kleine serie binnen de serie. Die gaat over de hoeksteen van alle denkbare vormen van signaalverwerking: de Fourieranalyse.

Tot voor kort waren de filters die je in muziekapparatuur aantrof uitsluitend bedoeld om vertrouwde, akoestische filterwerkingen na te bootsen zoals echo's en galm in een effectprocessor, het zwiepende digi-toonbolfilter in een synthesizer, of de toonregeling in een versterker. Maar sinds de opmars van DSP begint daar verandering in te komen en krijgt een filter ook steeds vaker de functie van *analyzer* die niets anders doet dan informatie verstrekken over de spectrale eigenschappen van een geluid. Dat heeft grote gevolgen voor de klankopwekking, effecten en de komende generaties van digitale muziekinstrumenten en -apparatuur. Vroeger bediende je een synthesizer ofwel met de hand ofwel via Midi; je maakte met de hand de een of andere instelling en kreeg dan een bepaalde klank of klankontwikkeling. Daarnaast kon je gebruik maken van een sequence-programma, zoals Cubase, om via Midi ("System Exclusive") de parameters van een synthesizer of sampler te wijzigen. Die twee vormen van aansturing hebben met elkaar gemeen dat er geen enkele terugkoppeling bestaat met de resulterende klanken; ze zijn, met andere woorden, niet in staat om naar de geluiden die ze zelf veroorzaken te "luisteren", zoals een muzikant dat wèl doet.

Die situatie verandert drastisch wanneer je analyzers gaat inzetten. Die kunnen als "oren" fungeren en bepaalde kenmerken van een geluid naar voren halen waarmee je dan vervolgens een synthesizer, sampler of effectmachine kunt bijsturen of "bespelen". Daarnaast worden analyzers gebruikt om bijvoorbeeld een starre, mechanische synthesizertoon klankeigenschappen te geven van bijvoorbeeld een gesampelde viool, de branding of een grommend pekineesje ("cross-synthese"). In Sound Design Workstations en in allerlei Plug-Ins kom je die koppeling van een *analyzer* en een *synthesizer* steeds vaker tegen. Populaire toepassingen daarvan zijn vocoders, formantfilters (spraak en zang), pitch trackers (het "robotstemmetje" van Cher) en diverse *morphing*-software (zoals in de bekende *Smyrnof* reclames). Door zulke cyber-achtige mengsels van "echte" en "kunstmatige" klankeigenschappen vervaagt de grens tussen muziek en andere vormen van geluidskunst en geluidstechniek. Audio-DSP is duidelijk bezig muziekgeschiedenis te schrijven. Op vocoders, formantfilters en morphing komen we in latere afleveringen uitvoerig terug. Daarvoor is het nodig om eerst nog wat meer te vertellen over de basis daarvan, de filtertechniek die in analyzers wordt gebruikt.

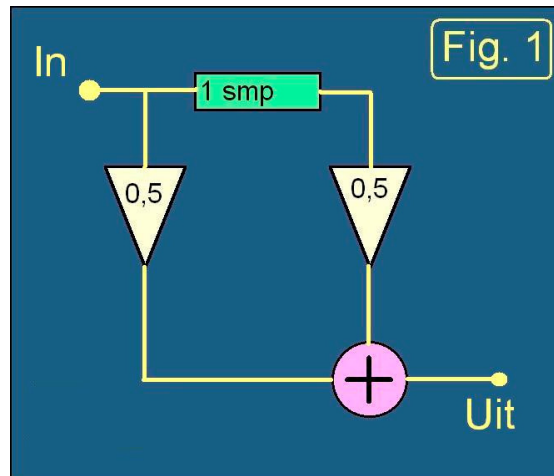
Zoals je inmiddels weet, is een filter een systeem dat sommige frequenties sterker doorlaat dan andere. Filters zijn dus vooral op te vatten als *spectrale* vervormers. In de *frequentiekaracteristiek* van een filter kun je die vervorming grafisch aflezen. Het idee achter een analyzer is nu om een filter zo te ontwerpen dat alleen een heel klein frequentiegebiedje (*frequentie-*

band) wordt doorgelaten en alle andere zo sterk mogelijk worden onderdrukt. Banddoorlaatfilters dus. Vervolgens kun je dan een *bank* aanleggen met een hele serie van zulke filters die elk een andere frequentieband doorlaten. We zagen dat Helmholtz over een bescheiden filterbankje beschikte in de vorm van een doos toonbollen waarmee hij het amplitudespectrum (= de sterkte van de boventonen) van een harmoniumtoon redelijk wist te schatten. Ook de display van een equalizer of een spectroscop wordt aangestuurd door een filterbank die is opgebouwd uit een (betrekkelijk klein) aantal bandfilters.

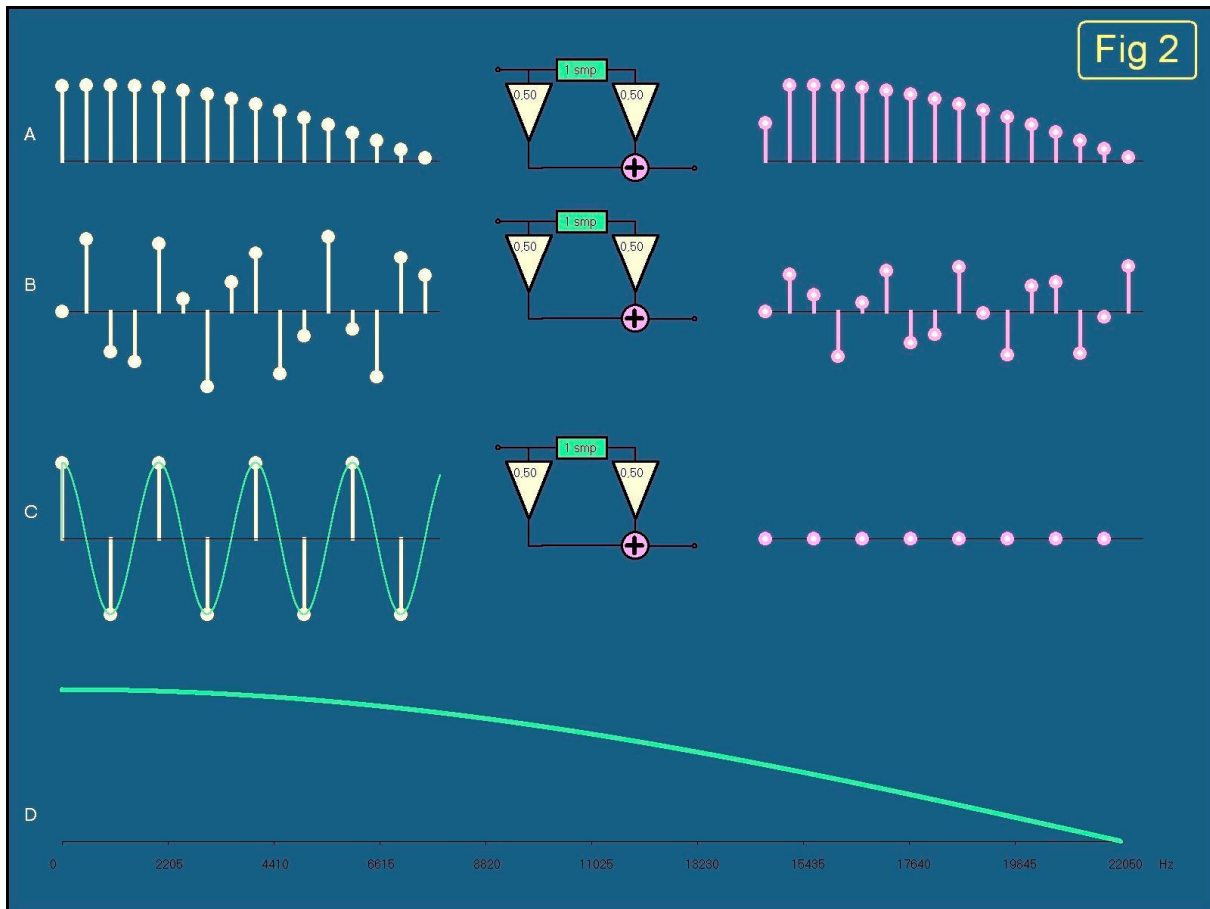
Het bekendste, belangrijkste en snelste digitale gereedschap voor spectraalanalyse is de FFT (Fast Fourier Transform). Ook de FFT is te beschouwen als een filterbank, zij het ook een heel speciale die bovendien niet alleen de sterkten, maar ook de fases van de frequentiecomponenten van een geluid boven tafel weet te krijgen (Afl. 7). Als je dat faseprobleem even buiten beschouwing laat, dan kun je zeggen dat het Fourier-filter is gebaseerd op twee vrij eenvoudige ideeën; samen leveren die een bandfilter op dat heel geschikt is om er een filterbank mee vol te stoppen. Het ene is Amplitude Modulatie en het andere is het principe van het *Lopend Gemiddelde*. Daarover gaat het in deze aflevering.

Lopend Gemiddelde

Filterwerking ontstaat zodra er een of meerdere reflecties in het spel zijn. In Afl. 5 werd al duidelijk met welk simpel rekenwerk zo'n reflectie te simuleren is, maar je kunt er ook ietsje anders tegenaan kijken waardoor het filtereffect beter zichtbaar wordt. In Fig. 1 is daarom nogmaals dat eenvoudige echocircuit te zien. Zowel de directe lijn van input naar output als de delaylijn hebben elk een eigen Gain, beide ingesteld op 0,5. De delay heeft de waarde van 1 sampletijd (Afl. 6). Dat laatste betekent dat in de delaylijn steeds "de vorige inputsample" zit. Zo krijg je een filter dat elke inputsample door twee deelt en daar de helft van de vorige inputsample bij optelt. Dat is een wat omslachtige manier om te zeggen dat elke outputsample het gemiddelde is van twee opeenvolgende inputsamples! Dus outputsample 2 is het gemiddelde van inputsamples 2 en 1; outputsample 3 is het gemiddelde van inputsamples 3 en 2 etc. Om die reden heet zo'n circuit een *Lopend Gemiddelde* filter (je komt in handleidingen vaak de engelse term *Moving Average* tegen).

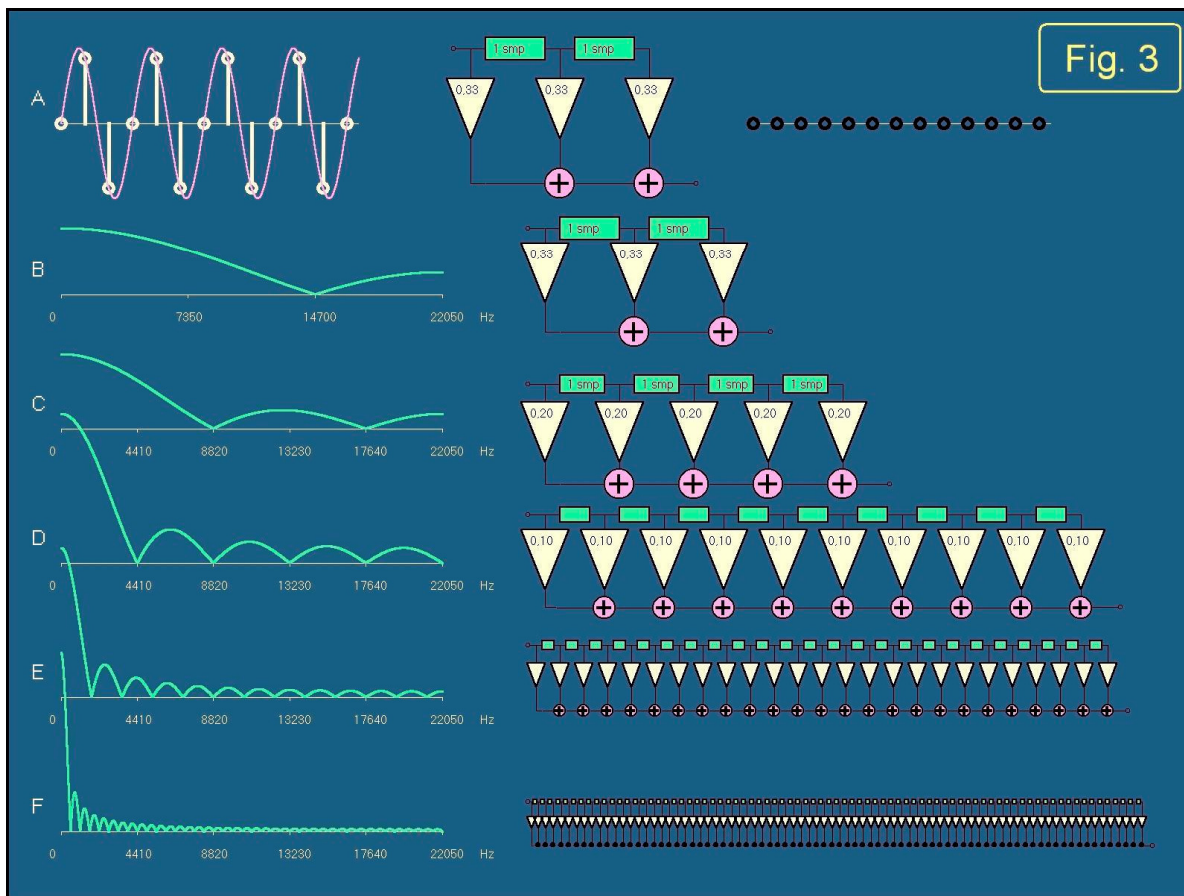


Waarom werkt zo'n Lopend Gemiddelde nu als een filter? Dat wordt snel duidelijk als je het filter eerst uitprobeert op een lage en dan op een hoge toon. In Fig. 2A is een lage toon te zien: de opeenvolgende samplewaarden veranderen maar heel geleidelijk en de verschillen ertussen zijn maar klein. Dus de gemiddelden tussen twee opeenvolgende samples liggen daar heel dicht bij in de buurt. Bij een hoge toon daarentegen, zoals in Fig. 2B, krijg je een samplepatroon dat veel heftiger op en neer springt; het verschil tussen opeenvolgende samplewaarden is veel groter. Twee getallen die veel van elkaar verschillen hebben een veel kleiner gemiddelde dan twee getallen die vlak bij elkaar liggen. Bij hoge tonen heeft het lopend gemiddelde dus ook veel kleinere waarden dan bij lage tonen. Daarom is de filteroutput van hoge tonen veel zachter dan die van lage. Het Lopend Gemiddelde filter is dus een laagdoorlaatfilter.



In Fig. 2D is de amplitudekarakteristiek ervan getekend. Je ziet de grafiek van links naar rechts geleidelijk dalen. Hoe hoger de toon, des te zwakker die wordt doorgelaten. Let wel, dit voorbeeld - evenals alle andere in deze serie - gaat ervan uit dat de samplefrequentie is ingesteld op 44100 Hz, zoals bij CD; het gedrag van alle signaalverwerking die je toepast, zoals filters, hangt daar natuurlijk van af. Helemaal rechts in de grafiek, precies bij de frequentie van 22050 Hz, gebeurt iets uitzonderlijks. Dat is de enige frequentie die volledig wordt tegengehouden! Die 22050 Hz is de halve samplefrequentie en daarmee de hoogste frequentie die je kunt weergeven (Afl. 2). In Fig. 2C is te zien dat een sinustoon van die frequentie door de AD-converter wordt gedigitaliseerd met slechts twee samples per periode die dezelfde waarde hebben, maar verschillen van teken. En het gemiddelde daarvan is natuurlijk nul. Elk geluid dat door dit simpele filtertje wordt gehaald zal een beetje dof klinken, alsof je verkouden bent. Als analyzer is het filter natuurlijk volkomen ongeschikt, maar toch heeft 't interessante toepassingen in klanksynthese.

We gaan nu aan dit filter een delaylijn toevoegen. De drie Gains zetten we alledrie op 0,33. Je kunt gemakkelijk nagaan dat je hiermee een Lopend Gemiddelde over drie samples krijgt: elke outputsample is niets anders dan het gemiddelde over de laatste drie inputsamples. Wat voor gevolgen heeft dat voor de amplitudekarakteristiek? In Fig. 3B zie je het merkwaardige resultaat. Ook hier heb je weer precies één frequentie die door het filter volledig wordt tegengehouden. Die frequentie blijkt 14700 Hz te zijn, dat is een derde deel van de samplefrequentie. In Fig. 3A is te zien dat één periode van een sinustoon van 14700 Hz door de DA-converter wordt gedigitaliseerd door precies 3 samples die bij een Lopend Gemiddelde over 3 samples natuurlijk weer nul opleveren. Door deze "nul-dip" bij 14700 Hz wordt de grafiek in twee segmenten verdeeld: het grote segment, links, wordt de *hoofdlob* genoemd en het kleine-



re, rechts, de *zijlob*. In de hoofdlob worden frequenties veel sterker doorgelaten dan in de zijlob.

Aan dit Lopend Gemiddelde filter kun je eindeloos delaylijnen en Gains blijven toevoegen. In Fig. 3C t/m 3F zijn links de frequentiekaracteristieken en rechts de circuits te zien van versies met 5, 10, 25 en 80 Gains. In het laatste geval is elke outputsample dus het gemiddelde van de laatste 80 inputsamples. Als je goed telt, zie je dat het aantal lobben steeds de helft van het aantal delaylijnen is. Ook valt op dat de "nul-dips" op gelijke afstanden van elkaar liggen. Iets soortgelijks hebben we op ook gezegd van harmonischen (Afl. 3). Vandaar dat dit filter uitermate geschikt is voor de analyse van harmonische tonen, zoals die van de meeste muziekinstrumenten. De Fourieranalyse is dan ook eigenlijk speciaal bedoeld voor het uitpluizen van harmonische spectra, ook al kun je er nog veel meer mee doen dan dat.

Amplitude Modulatie

Hoe meer delaylijnen je aan dit Lopend Gemiddelde filter toevoegt, des te meer lobben je krijgt en des te smaller ze worden. Dat laatste geldt ook voor de hoofdlob. Hoe smaller die is, des te kleiner het frequentiegebiedje dat wordt doorgelaten. Met andere woorden: hoe groter het aantal samples waarover je het gemiddelde neemt, des te duidelijker het laagdoorlaat-effect merkbaar wordt en des te groter het deel van het spectrum in de zijlobben terecht komt en dus onderdrukt wordt. Dat dit voor een analyzer een voorwaarde is, zal wel duidelijk zijn, maar dan blijft toch de vraag: wat heeft dit laagdoorlaatfilter te maken met de banddoorlaatfilters die we nodig hebben voor onze analyzer-filterbank? Het blijkt heel eenvoudig te zijn om het Lopend Gemiddelde filter te veranderen in een bandfilter. Daarvoor maken we gebruik van een bekend trucje.

In Afl. 8 bleek dat het oor een aantal merkwaardige afwijkingen heeft. Een daarvan is de vorming van de zogenaamde combinatietonen, waaronder de somtoon en de verschiltoon. Als je twee sinustonen hebt, een van 600 Hz en een van 400 Hz, dan is de somtoon een sinus van 1000 Hz en de verschiltoon een sinus van 200 Hz, zo simpel is dat. Niet alleen in het oor kunnen som- en verschiltonen optreden. Ook in allerlei muziekinstrumenten, in versterkers en andere audio-apparaten worden ze opgewekt. Het gaat dus om een veel voorkomend, natuurkundig verschijnsel. Misschien herinner je je van vroeger in de wiskundeles zo iets als de "som- en verschilformule". Die gaat over precies hetzelfde en vertelt bovendien dat de berekening die je moet uitvoeren om de som- en verschiltonen van twee sinussen op te wekken niets anders is dan een vermenigvuldiging. In de radiotechniek wordt dit vermenigvuldigen van signalen aangeduid als *Amplitude Modulatie (AM)* en zo is de term in muzikale klanksynthese terecht gekomen. Vroeger werd AM in analoge synthesizers voornamelijk gebruikt om gong- en bel-achtige klanken op te wekken.

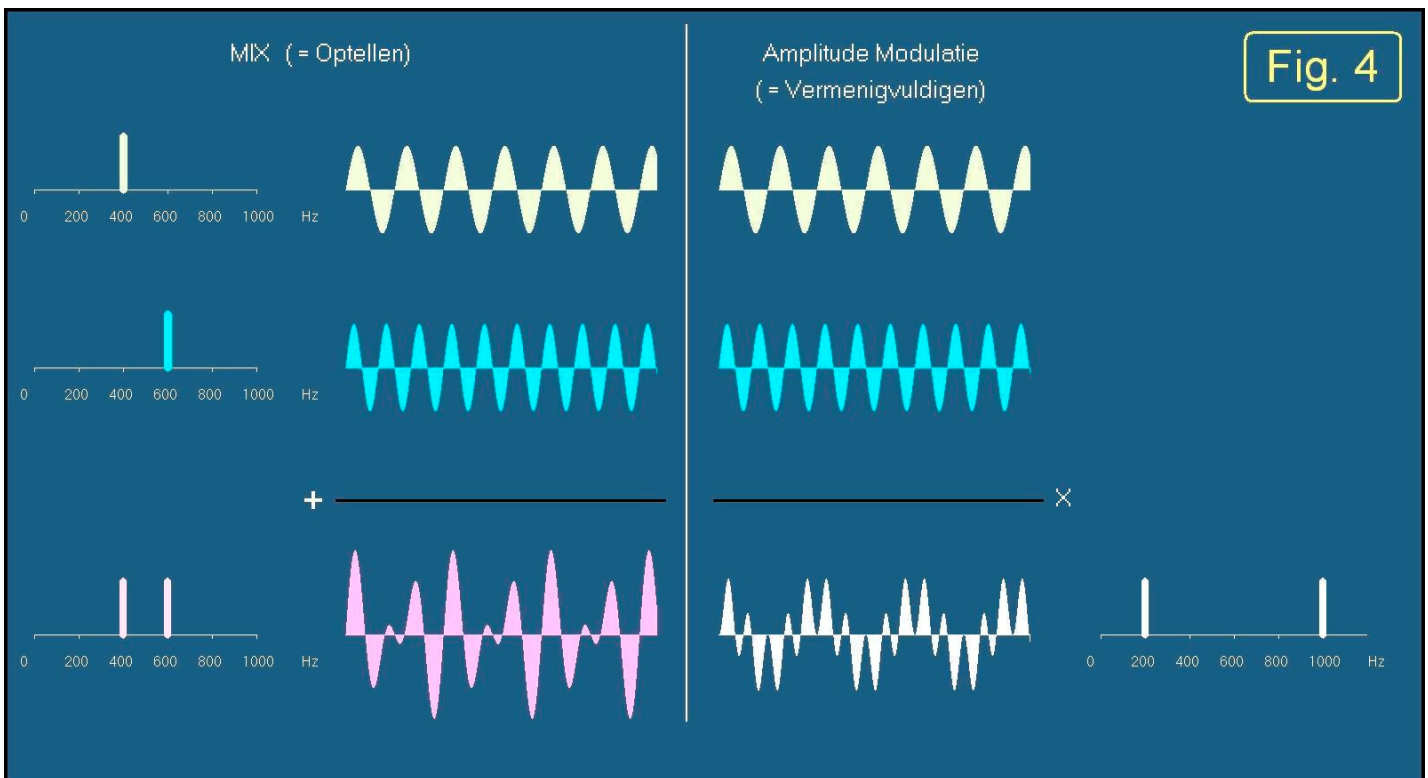
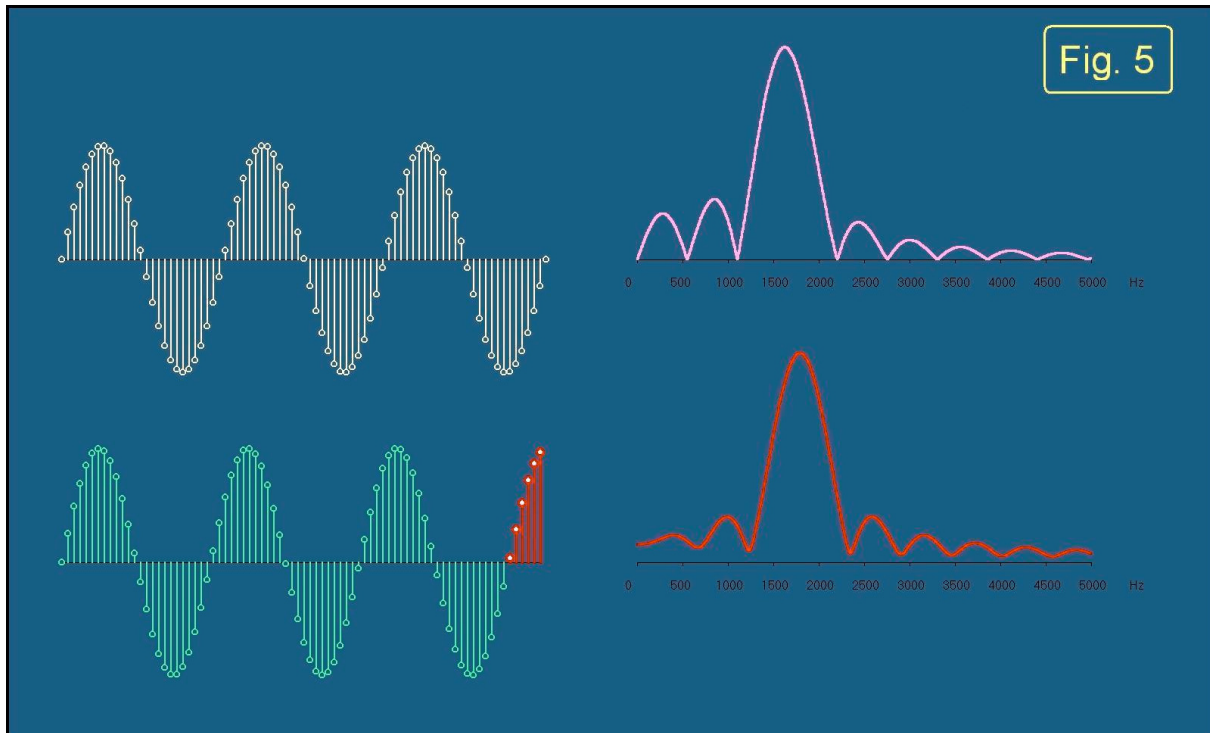


Fig. 4

In Figuur 4 is het onderscheid te zien tussen het optellen en het vermenigvuldigen van twee sinustonen. Wanneer je een sinustoon van 600 Hz samplegewijs optelt bij een van 400 Hz, dan klinkt er niets anders dan de mix van beiden: je hoort ze gewoon tegelijkertijd. Maar wanneer je ze samplegewijs vermenigvuldigt, dan krijg je een somtoon van 1000 Hz en een verschiltoon van 200 Hz te horen. En juist door het ontstaan van die verschiltoon bij vermenigvuldiging is AM zo interessant voor een combinatie met dat Lopend Gemiddelde filter. Want als je die sinustoon van 600 Hz vermenigvuldigt met een sinustoon van eveneens 600 Hz, dan krijg je een somtoon van 1200 Hz en een verschiltoon van 0 Hz. (Zo'n "0 Hz-toon" zagen we al eerder, in de Gobiwoestijn, in de vorm van de gemiddelde jaartemperatuur, zie Afl. 7, Figuur 1A) En de 0 Hz-component wordt in het Lopend Gemiddelde filter maximaal doorgelaten, zoals alle amplitudekarakteristieken van Figuur 3B t/m 3F laten zien. Wat je dus in feite doet is AM gebruiken om het gebiedje rond 600 Hz maximaal door te laten. Conclu-

sie: in feite is dus de combinatie van een Lopend Gemiddelde filter en AM equivalent met een banddoorlaatfilter!

Nu zetten we de laatste stap. Je hebt je misschien afgevraagd wat al die Gains in het Lopend Gemiddelde filter nu eigenlijk voor onnut werk staan te doen! Per slot van rekening hoef je niet eerst bijvoorbeeld honderd getallen waarover je het gemiddelde wilt berekenen *apart* met 0,01 te vermenigvuldigen en daarna op te tellen; je kunt ze simpelweg eerst bij elkaar optellen en dan de uitkomst delen door 100, dat scheelt honderd vermenigvuldigingen per outputsample! Toch hebben die Gains wel degelijk zin, die juist in de combinatie met AM duidelijk wordt. Als je zelf een beetje gaat puzzelen, zul je vaststellen dat het lood om oud ijzer is of je nu eerst een signaal moduleert en daarna het resultaat daarvan door het Lopend Gemiddelde filter haalt of dat je de Gain-registers vult met de samplewaarden van de moduleerende sinustoon en dan daarmee gaat filteren! AM kun je op die manier dus eenvoudig in het Lopend Gemiddelde filter inbouwen. Om dat te illustreren nemen we het filter uit Fig. 3F, met 80 Gains. Stel dat we de derde lob graag tot hoofdlob willen bombarderen. Dan vullen de Gains met de samples van een sinustoon waarvan de frequentie zo is gekozen dat er precies drie hele periodes binnen dat raam van 80 samples passen. Het resultaat is te zien in Fig. 5 boven. Links zijn de 80 "sinus-Gain"-waarden te zien en rechts de amplitudekarakteristiek van het bandfilter.



De AM-truc werkt alleen maar goed als het aantal sinusperiodes waarvan de samplewaarden aan de Gains worden toegekend precies een geheel getal is. Als je een sinus zo zou kiezen dat er niet precies een geheel aantal periodes binnen het raam van de 80 Gains valt, dan vervaagt de lobbenstructuur en gaan de "nul-dips" verloren, waardoor het filter juist z'n sterkste eigenschap kwijt raakt, zie Fig. 5 onder. Waarom dat zo is en hoe je door AM deze bandfilters tot een filterbank samenvoegt, zien we in volgende afleveringen aan de hand van praktische voorbeelden.