

Het Tijdvenster

De term lijkt afkomstig uit een exotisch avontuur van Heer Bommel, maar is te vinden in elke FFT-implementatie. Je 'kijkt' altijd naar een geluid via een Tijdvenster (Time Window). De breedte en vorm daarvan bepalen de kwaliteit van het spectrum dat je voor de synthese van nieuwe klanken gebruikt.

Nog maar nauwelijks bekomen van het Bingo-gevoel dat ons bekreep na het succesvol oppompen van voorbijvliegende sinussen, of we worden direct al weer geconfronteerd met nieuwe problemen. In de vorige drie afleveringen viel opvallend vaak het woord 'perfect'. De resultaten van de spectrumanalyse door middel van de Fourierfilterbank - die we verder zullen aanduiden met de meer gangbare term DFT (Discrete Fourier Transform) - bleken 'perfect' te zijn, dat wil zeggen dat ze voor de volle 100% voldeden aan de verwachtingen. Dat kwam omdat we het aantal delays van de DFT perfect konden afstemmen op de periodetijd van de grondtoon van het proefsignaal waarvan we het spectrum wilden weten. Zoiets is alleen maar mogelijk met een 'perfect', statisch, puur harmonisch signaal. Nu neemt elke muzikant en elke studietechnicus het woord 'perfect' wel eens in de mond, maar doorgaans wordt er dan toch iets heel anders mee bedoeld; de 'perfecte' harmonische toon is namelijk oer- en oersaai en allesbehalve muzikaal interessant. Muziek komt pas tot leven door geluiden die juist *niet* perfect en *niet* statisch zijn. Als je zulke niet-perfecte klanken gaat analyseren, zullen ook de resultaten van de DFT en FFT niet 'perfect' zijn.

Muziekinstrumenten produceren tonen waarvan de boventoonstructuur slechts bij benadering harmonisch is. Dat harmonisch-zijn is dan wel een voorwaarde voor het goed samenklanken van meerdere tonen in akkoorden, maar het is niet de reden waarom muziekinstrumenten zo levendig klinken. Levendigheid is het woord dat je gebruikt om allerlei moeilijk voorspelbare, kleine variaties in de toon samen te vatten: variaties in toonhoogte, in volume en in het amplitude- en fasespectrum. Die kleine variaties tref je in elke toon van elk akoestisch muziekinstrument aan. Strijkers en blazers produceren bovendien ook een klein beetje ruis, die allesbehalve harmonisch is, maar wel heel karakteristiek kan zijn voor de klank. Veel percussieve instrumenten zoals klokken, bellen, gongen, marimba's, xylofoons en pauken hebben wel een 'toonachtige' klank, maar produceren boventonen die kris-kras over de frequentie-as zijn uitgestrooid en die dus bepaald niet harmonisch zijn.

De meeste audiosignalen zijn afkomstig van niet-instrumentale klanken, zoals spraak, straatgeluiden, een knappend hardvuur, voetstappen op een grindpad. Dergelijke signalen variëren continu, zijn vaak verre van harmonisch en hebben geen duidelijke "grondtoon". Daarmee ontbreken nu precies de eigenschappen die je voor perfecte resultaten met de DFT nodig hebt. Als je het spectrum van dergelijke klanken kunt berekenen, dan beschik je over de basis voor allerlei mogelijke geluidseffecten, zoals morphing en vocoding, die in media als

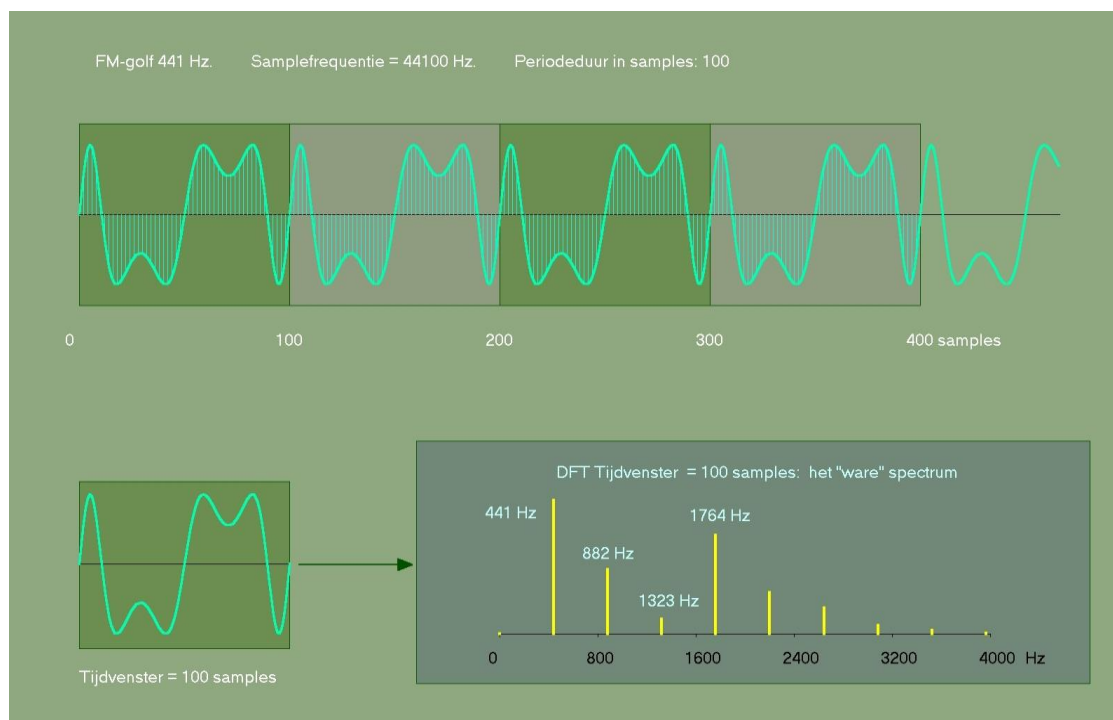
film, video, in computerspellen en websites een speciale functie of klankbetekenis kunnen hebben.

De vraag is nu of je de DFT, en de vliegensvlugge variant daarvan, de FFT, eigenlijk wel kunt gebruiken voor elk willekeurig audiosignaal, perfect of niet. Dat het antwoord wel zo ongeveer 'ja, mits...' zal luiden, kun je min of meer afleiden uit het feit dat de FFT, ondanks de zeer beperkte keuze wat betreft het aantal delays, toch zeer populair is. Dat aantal delays moet altijd een macht van 2 zijn, dus 2, 4, 8, 16, 32 etc. In deze aflevering zal blijken dat de FFT desondanks wel degelijk als een universeel gereedschap voor de berekening van spectra te gebruiken is, mits je een paar verschijnselen en begrippen goed in de peiling hebt. Het belangrijkste daarvan is zonder meer de notie van het *tijdvenster*, in audiosoftware vaak aangeduid met *time window*, of simpelweg window (wat uiteraard helemaal niets te maken heeft met 'Windows 95/98', het operating system van de PC, of met de schermvensters of 'windows' die je in elk computerprogramma tegenkomt).

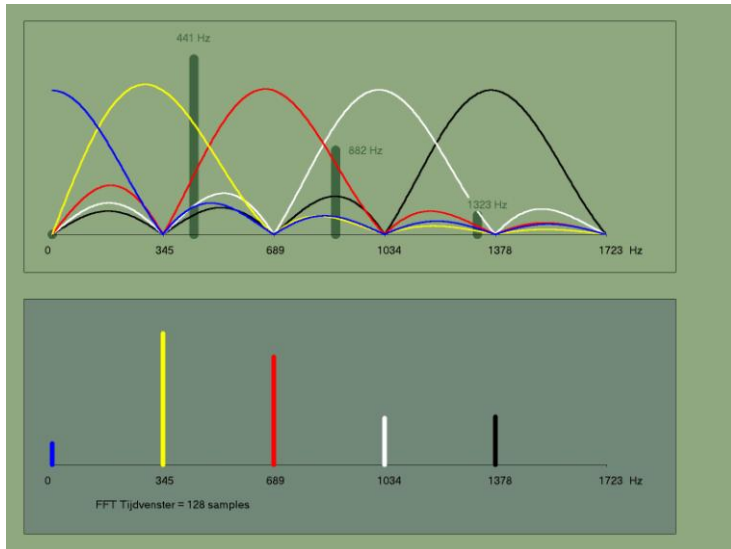
De term tijdvenster slaat precies op dat aantal delays dat je in de DFT of FFT gebruikt. Het vormt een sleutelbegrip in de berekening van het spectrum. Een spectrum heeft namelijk altijd betrekking op een 'stukje' geluid, dus een geluidsfragment van een bepaalde tijdsduur. Die tijdsduur komt precies overeen met wat we tot nu toe de transiënt van de bandfilters in de DFT noemden. (De transiënt is de tijd die verstrijkt voordat alle delays van het filter gevuld zijn met inputsamples en is gelijk aan het aantal delays gedeeld door de samplefrequentie.) De tijdsduur van het geluidsfragment en de transiënttijd zijn dus altijd even lang en worden ook wel aangeduid met de *breedte* van het tijdvenster. Hoe breder het tijdvenster des te langer het geluidsfragment dat de DFT of FFT 'ziet' en in de spectrumberekening betreft. Welke grote invloed dat heeft op de resultaten maken we duidelijk aan de hand van een voorbeeld.

Lekkende Filterbank

Net als voorheen kiezen we weer een statische, puur-harmonische proeftoon. Ditmaal een



klankje dat je met een FM-synth of geluidskaat gemakkelijk kunt opwekken. Zie Figuur 1 (boven). We stellen de grondtoon op 441 Hz. Als je daarbij van een samplefrequentie van 44100 Hz uitgaat, dan is de periodetijd precies gelijk aan 100 samples. Kies je voor de DFT een tijdvenster met een breedte van eveneens 100 samples, dan krijg je weer een perfect amplitude- en fasespectrum, waarvan in Figuur 1 (onder) alleen het amplitudespectrum is getekend. Geheel volgens verwachting vinden we boventonen bij veelvouden van de



grondtoonfrequentie, dus bij 882 Hz (octaaf), bij 1323 Hz (octaaf + kwint) etcetera. Dit spectrum zou je het 'ware' spectrum kunnen noemen: het geeft precies aan hoe sterk de boventonen van deze FM-klank zijn.

Nu gaan we kijken wat er gebeurt wanneer je deze FM-toon gaat analyseren met de FFT. Die heeft, zoals gezegd, een zeer beperkt repertoire aan vensterbreedten en daarom zijn we genoodzaakt een breedte te kiezen die zo dicht mogelijk in de buurt ligt van de breedte 100, waarmee we zojuist het 'ware' spectrum vonden. Die dichtstbijzijnde FFT-

vensterbreedte is 128, een macht van 2. Wat we met de FFT in feite doen is een Fourierfilterbank aanmaken met bandfilters die elk 128 delays hebben. Daardoor ontstaan in de frequentiekaracteristiek van elk bandfilter op 128 plaatsen de bekende nul dips (Afl. 9) die, net als bij harmonischen, op gelijke afstanden van elkaar liggen, in dit geval om de 345 Hz (dat is de samplefrequentie gedeeld door het aantal delays, dus $44100 / 128 = 345$).

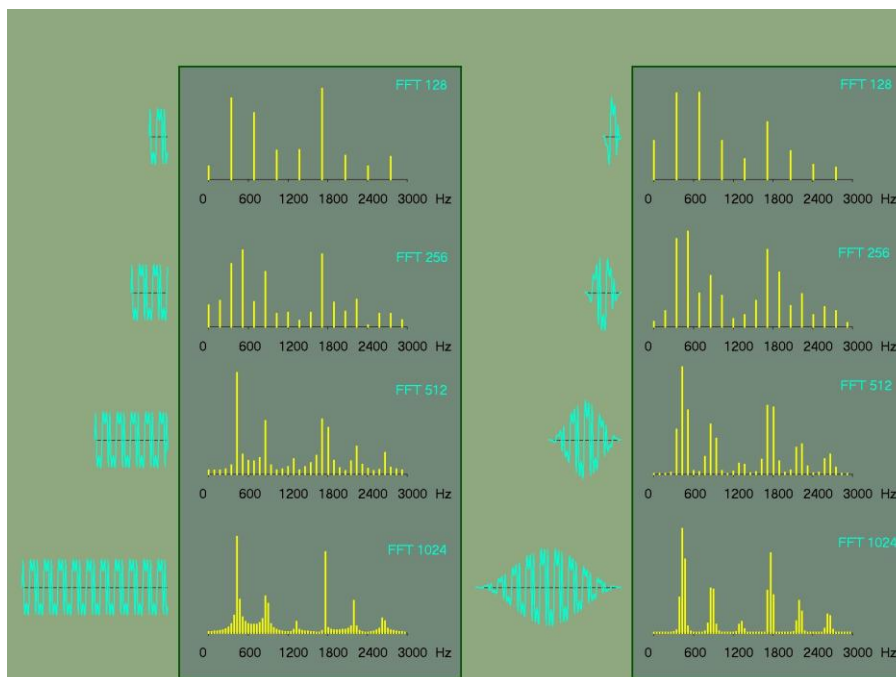
In Figuur 2 (boven) zijn de frequentiekaracteristieken van eerste vijf bandfilters in verschillende kleuren weergegeven. Daarachter zijn, in grijsgroen, enkele spectraallijnen te zien van het 'ware' spectrum van de FM-proeftoon. Het is in één oogopslag duidelijk: de perfectie is hier ver te zoeken. Wat is er precies aan de hand? Alle bandfilters hebben om de 345 Hz een nul dip, dus als de grondtoon van het FM-proefsignaal eveneens 345 Hz zou zijn geweest dan zou het gele bandfilter alle frequenties behalve de grondtoon hebben tegengehouden; het rode bandfilter zou alle frequenties behalve de tweede harmonische van 689 Hz hebben tegengehouden en de andere filters zouden eveneens elk maar één harmonische hebben doorgelaten en alle andere volledig hebben tegengehouden. Maar omdat in dit geval de proeftoon om elke 441 Hz een harmonische heeft, valt er geen enkele samen met een nul dip en zal *elk* bandfilter *elke* harmonische van de proeftoon doorlaten!

Neem bijvoorbeeld het blauwe bandfilter; de harmonische van 441 Hz valt in de eerste zijlob (Afl. 9) en zal dus verzwakt worden doorgelaten (zie Afl. 10, kader "*Filteren is Vermenigvuldigen*"). De harmonische van 882 Hz valt in de tweede zijlob en zal eveneens worden doorgelaten, zij het ook wat zachter. In Figuur 2 (onder) is het spectrum te zien zoals het door de FFT werd berekend. De blauwe spectraallijn suggereert dat ons FM-proefsignaal een 0 Hz component van een bepaalde sterkte heeft, terwijl het 'ware' DFT-spectrum van Figuur 1 laat zien dat de proeftoon helemaal geen 0 Hz component heeft! Ook de andere spectraallijnen van Figuur 2, die in kleur corresponderen met de bandfilters erboven, 'melden' de aanwezigheid van harmonischen die er in werkelijkheid niet zijn. Je kunt het ook anders

formuleren: wat elk FFT-bandfilter 'hoort' en doorlaat is een mengsel van alle harmonischen en dat mengsel klinkt in elk bandfilter weer anders: een maat daarvoor geven de lengten van de verschillend gekleurde spectraallijnen. De harmonischen van de proeftoon hebben zich dus over alle bandfilters verspreid, ze zijn "weggelekt", zoals dat in DSP-jargon heet (*leakage*). Dus wanneer de Fourierfilterbank en het proefsignaal niet perfect op elkaar zijn afgestemd, representeren de bandfilters geen zuivere sinussen maar toonmengsels en daarom kun je de spectrumresultaten niet meer interpreteren als de boventonen van de proeftoon!

Gitaarbrul

Voor al deze narigheid zijn twee oorzaken aan te wijzen. De eerste heeft te maken met die vervelende zijlobben. Daar hadden we het nog niet eerder mee aan de stok, omdat het tijdvenster en de periodetijd van de grondtoon van het signaal perfect op elkaar waren afgestemd; daardoor vielen de harmonischen mooi samen met de nul dips tussen de zijlobben.



Door die zijlobben zijn de bandfilters verre van ideaal en het is duidelijk dat de lekkage van de harmonischen van de proeftoon een heel eind kan worden tegengegaan als we die zijlobben op een of andere manier kunnen wegschroeven of afgrazen. Nu bestaat er gelukkig een heel simpele methode waarmee de zijlobben inderdaad behoorlijk worden platgewalst. Zie Kader "Het Bolle Tijdvenster".

De tweede oorzaak zit 'm simpelweg in het aantal filters in de filterbank. In het voorbeeld hierboven waren dat er 128. Je kunt het effect daarvan min of meer vergelijken met resonanties die optreden in een akoestische gitaar of een piano. Stel, je geeft een stevige brul vlakbij de klankkast van een gitaar. Je hoort een zacht galmeffect, afkomstig van de zacht meetrillende gitaarsnaren. Welke snaren meetrillen en hoe hard is afhankelijk van de toonhoogte waarop je brult. Nu doe je hetzelfde bij de piano, terwijl je het sustainpedaal ingetrapt houdt. Ook nu hoor je een galmeffect, maar het is veel harder en helderder. Ook blijkt de toonhoogte van de brul niet zoveel uit te maken. Dat komt omdat de piano simpelweg meer snaren heeft waarvan de frequenties veel dichter opeengepakt liggen. Daardoor is de kans dat een of meerdere boventonen van je stem pianosnaren in trilling brengen veel groter dan bij de 6-snarige gitaar. In DSP-taal: de *frequentieresolutie* van de piano is hoger dan die van de gitaar. Daarbij moet je wel één ding goed in de gaten houden. Het galmeffect dat je bij de piano hoort mag dan zelfs zo helder zijn dat je je eigen stem

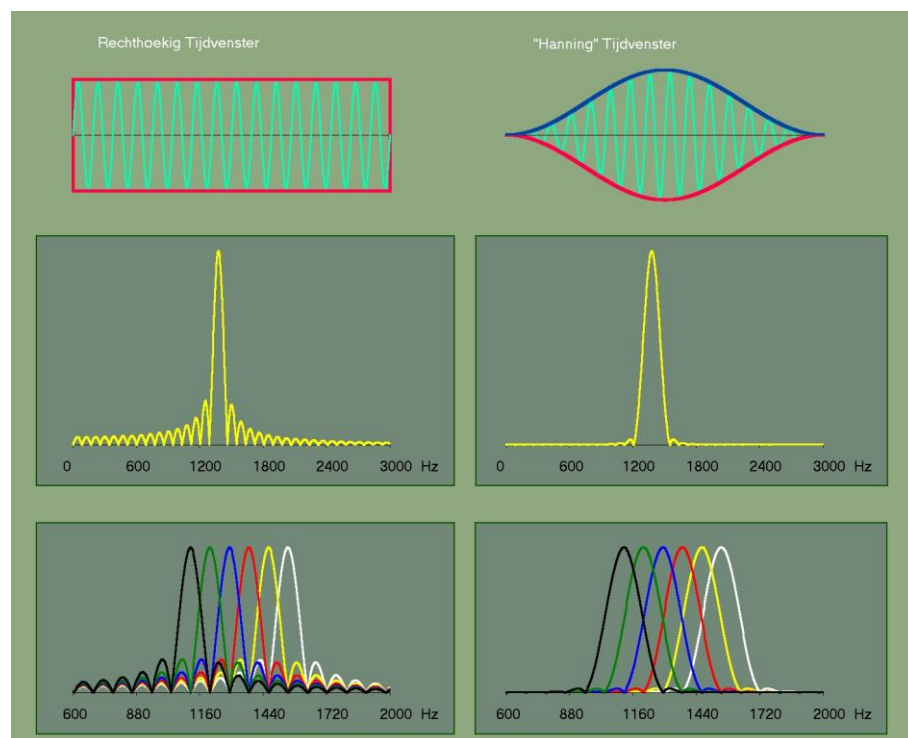
herkenbaar terug hoort in de ronkende snaren, toch is daarmee niet gezegd dat de boventonen die je stem tijdens de brul genereerden ook *perfect* overeenstemmen met de frequenties van de snaren; ze liggen er wel heel dicht bij in de buurt.

De FFT kun je vergelijken met een piano waarvan je zelf het aantal snaren/toetsen kunt instellen. Hoe breder het tijdvenster dat je kiest, des te meer bandfilters, des te meer 'snaren' de FFT krijgt, dus des te beter de frequentieresolutie wordt. Een FFT met een tijdvenster van 1024 samples heeft een 8 keer zo hoge frequentieresolutie als een FFT met een tijdvenster van 128 samples. In Figuur 4 zie je maar liefst 8 maal de FM-klank geanalyseerd door middel van de FFT, links 4 maal met een rechthoekig tijdvenster, rechts 4 maal met een Hanning-tijdvenster, van boven naar beneden met vensterbreedten van resp. 128, 256, 512 en 1024 samples. Hoe groter de vensterbreedte des te meer het FFT-spectrum op het 'ware' spectrum gaat lijken. Met name bij de FFT over 1024 samples is goed te zien dat het Hanning venster veel minder 'lekt'.

Kader: Het Bolle Tijdvenster

De DFT en FFT tref je in audio-applicaties meestal aan met twee 'instelknopjes'. Die hebben alletwee betrekking op het tijdvenster. De ene instelling specificceert de breedte van het tijdvenster; de effecten daarvan worden elders in dit artikel besproken. De andere instelling heeft betrekking op de *vorm* van het venster. Die venstervorm heeft grote invloed op de hoogte van de zijlobben van de bandfilters. In Figuur 3 (boven) zijn de twee verschillende venstervormen afgebeeld; de linker heet om begrijpelijke redenen *rechthoekig* en de rechter, met de korrel- of bolvorm, heet een *Hanning-tijdvenster*. Dit laatste venster kwamen we al eerder tegen in Aflevering 4, over granulaire synthese. Daar werd het gebruikt om geluidsgains uiterst zacht te laten beginnen en eindigen, zodat er geen storende, abrupte schakelklicks

tussen de grains hoorbaar is. Nog weer eerder, in Aflevering 2, zagen we hoe je een geluid harder of zachter kunt maken door geluidssamples te vermenigvuldigen met de samples van een "volumecontour". Rechtsboven in Figuur 3 is in blauw de volumecontour weergegeven die zowel bij granulaire synthese als bij het



bolle Hanning-tijdvenster wordt gebruikt. Tevens is te zien hoe daardoor een stukje sinustoon zodanig wordt "ingepakt" dat een soort sinuskorrel ontstaat.

Zo'n korrel bestaat natuurlijk gewoon uit samples, getallen. En die getallen kun je onder vele andere gebruiken om de Gainregisters van een Lopend Gemiddelde-filter mee te vullen, zoals in Afl. 9 gebeurde. Daar werd duidelijk dat je een Lopend Gemiddelde-filter, dat een laagdoorlatend karakter heeft, kunt ombouwen tot een bandfilter, door de Gains ervan te vullen met een of meerdere hele periodes van een sinustoon (Amplitude Modulatie). Dat is nog eens geïllustreerd in de gele frequentiekaracteristiek van Figuur 3 links: het bandfilter heeft in de buurt van 1400 Hz z'n hoofdlob, waar het filter maximaal doorlaat.

Duidelijk zijn de zijlobben te zien die daarbij ontstaan en die we eigenlijk liever kwijt dan rijk zijn. Rechts in Figuur 3 is nog een andere gele frequentiekaracteristiek te zien. Die hoort bij een Lopend Gemiddelde-filter, waarvan de Gains gevuld zijn met de bolle sinuskorrel van rechtsboven. De zijlobben blijken sterk te zijn onderdrukt. De prijs die je daarvoor moet betalen is dat de hoofdlob wel een stuk breder is geworden. Onder in de figuur is nog eens het verschil te zien tussen een filterbank met "rechthoekige" sinussen en een met "korrels sinussen". Die laatste overlappen elkaar weliswaar meer, maar buiten de hoofdlobben scheiden ze de frequenties van het inputsignaal veel beter. In een latere aflevering zullen we zien waarom hetzelfde effect van zijlob-onderdrukking ook optreedt wanneer je het signaalfragment "verpakt" in zo'n Hanning-tijdvenster alvorens het door een Fourierfilterbank te sturen waarvan de Gains met gewone "rechthoekige" sinusgolven zijn gevuld. In Figuur 4 rechts zie je voorbeelden van zulke ingepakte, bolle inputsignalen.