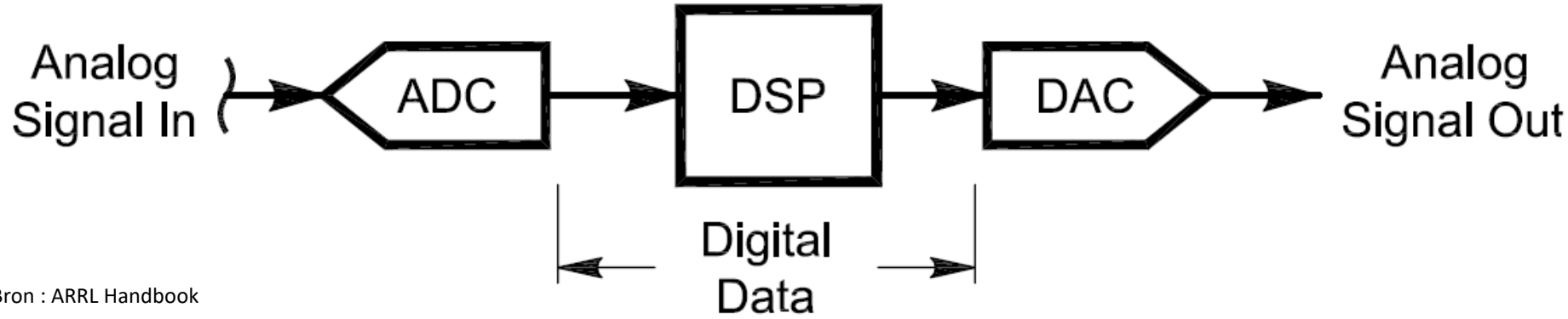
DSP en SDR voor de Ham, Arie PA3A

Versie okt 2026





Bron : ARRL Handbook

DSP en SDR voor de Ham

Belangrijk doel: Enig gevoel krijgen op de gebruikte DSP terminologie

Veel kretologie enigszins verklaard voor praktische begrip

Wat achtergronden van SDR ontwerpen

De wiskunde van DSP slaan we over.

Author:
Arie Kleingeld PA3A

Met wat (grote) stappen langs de DSP-velden…

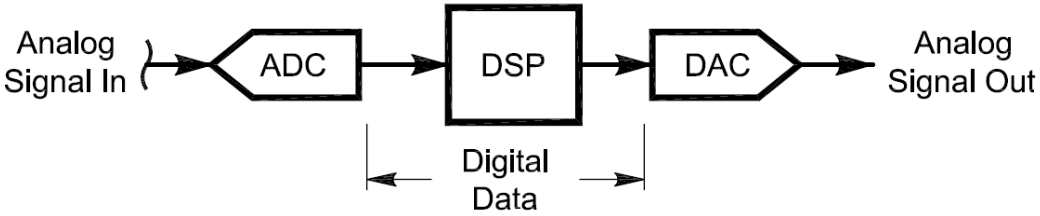
Als je niets van DSP weet, ga je nieuwe dingen leren
Dit kan soms pittig lijken…
maar hoe ingewikkeld kan het zijn?
Riemen vast en blijf erbij!

Inhoud



Met wat grote stappen door de **terminologie** van het digitale werk (de ‘basis’)

1. Van analoog naar Digitaal (ADC) en weer terug (DAC)
   - Nyquist theorema
   - ADC (uitgebreid)
   - Te kiezen sample-rates
   - Dithering
   - DAC



2. Waar het wonder gebeurt: DSP
   - FFT (Fast Fourier Transform) perikelen
   - Verschillende sample snelheden in de DSP (zgn. Multi-rate)
   - IQ en Positieve en Negatieve frequenties
   - Filters: FIR en Fast Convolution m.b.v. FFT, IIR (en evt.CIC)
3. Wat zijn de stappen van het signaal in een SDTRX
   - De digitale uitvoering in mode SSB vergeleken met analoge trx’s van het F-examen
   - Enkele populaire transceivers in blokken

Voorbeeld

- Een nieuwe SDR transceiver van Xiegu: de G7250

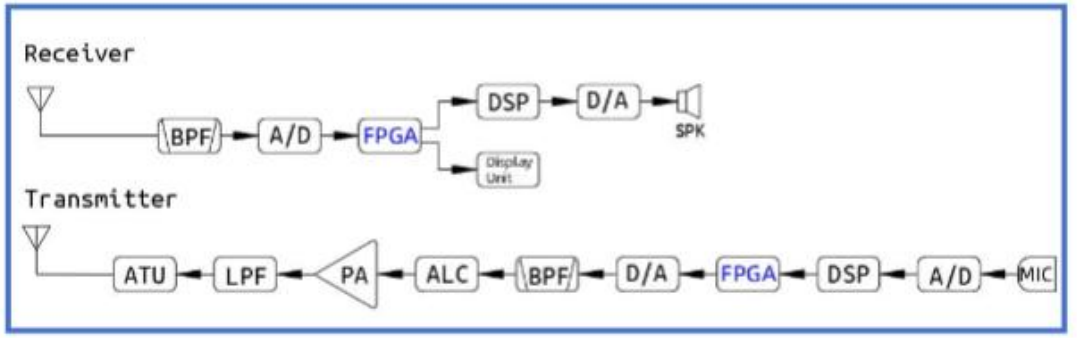
G7250

100W HF Amateur Radio Transceiver

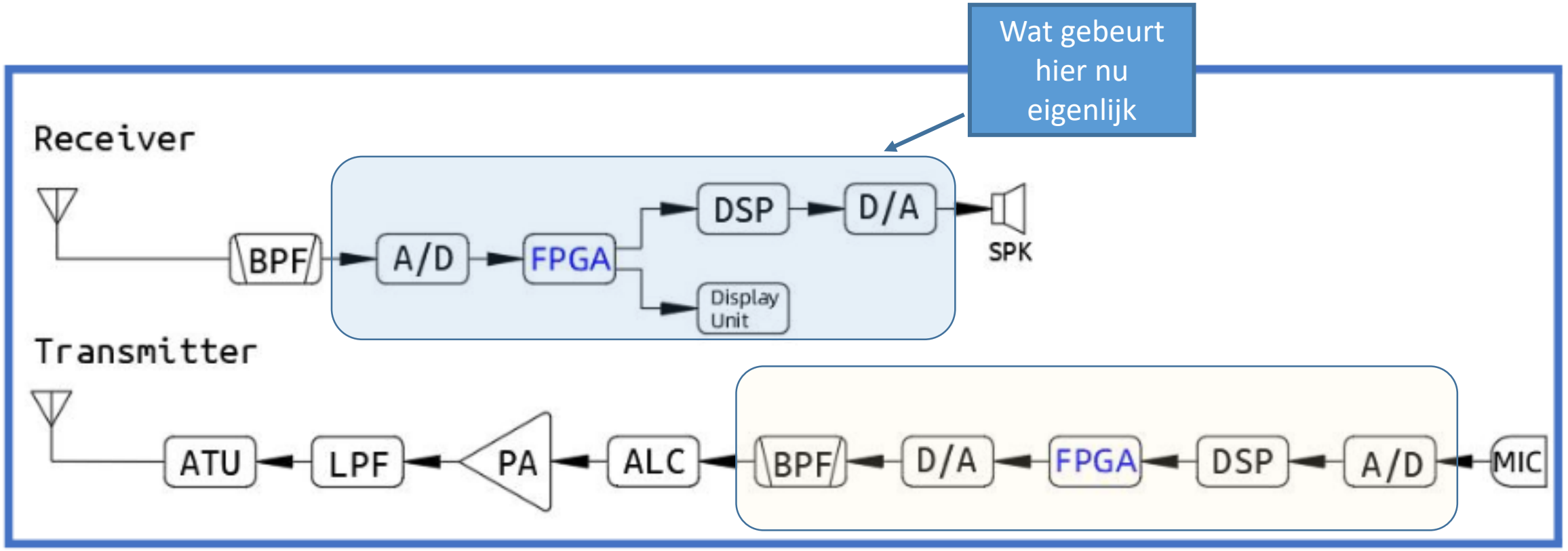


The G7250 features a top-tier Direct RF Sampling platform architecture, equipped with powerful baseband and RF units, giving amateur radio enthusiasts a smoother and more convenient operating experience. Its robust RF output power makes contacts feel effortless.

Thanks to its split design, the head unit (display and controller) and the main body can be deployed independently, making it a great fit for various setups and operating environments.

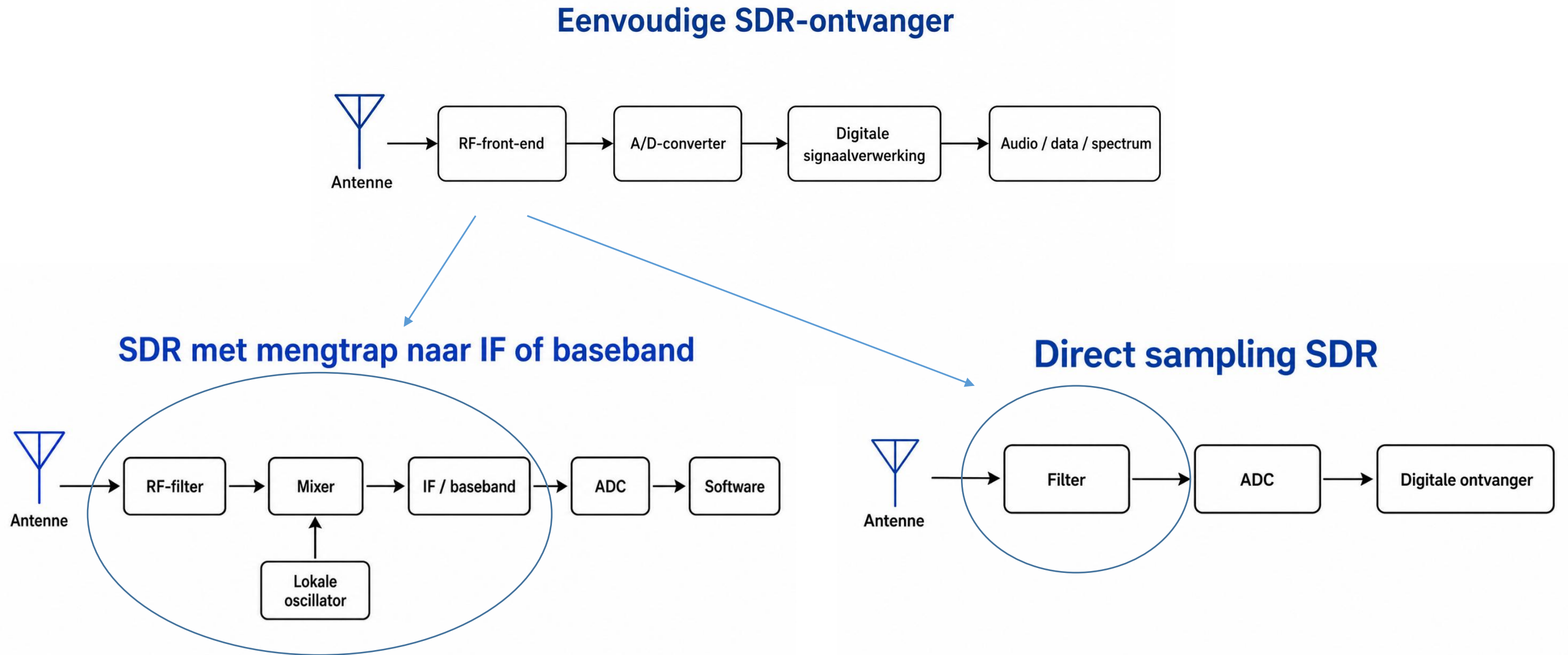


Blokschema van de Xiegu G7250:



- Logisch schema? Wat zegt het?

Simpele voorstellingen van een SDR RX



1. Van analoog naar digitaal (en weer terug)

- Stelling:
  Als we veel rekenkracht in een digital signal processor (DSP) hebben en prachtige software dan hebben we een goede transceiver.
  Ja, da's wel waar, maar… ***Er is meer!***

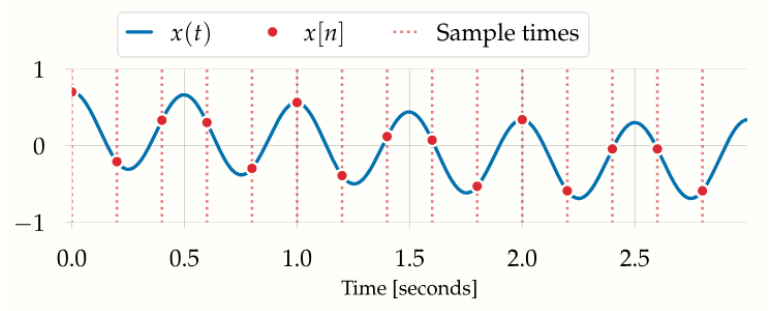
- De wereld is fysiek en analoog, dat moeten we eerst omzetten naar het digitale domein van DSP voordat we er dingen mee kunnen en later ook weer terug naar analoog als je wat wil horen.

- **Key**-elements:
  - Analoog Digitaal Converter (ADC)
  - Pre-Selector Filtering.
- Hier is veel over te vertellen! Er kan hier nl. veel fout gaan.
  De keten is zo sterk als de zwakste schakel.

DSP – Digital Signal Processing



- DSP is het bewerken (of analyseren) van informatie die is gemeten als een concrete serie gehele getallen (0, 1 , 2 tot ….)
- Signalen worden ***op gezette tijden gemeten en omgezet naar getallen***.
  Dit zijn de zgn. ***samples*** = **steekproeven**.
  Tussen die tijden zien we dus niks, met alle gevolgen van dien.



bijvoorbeeld

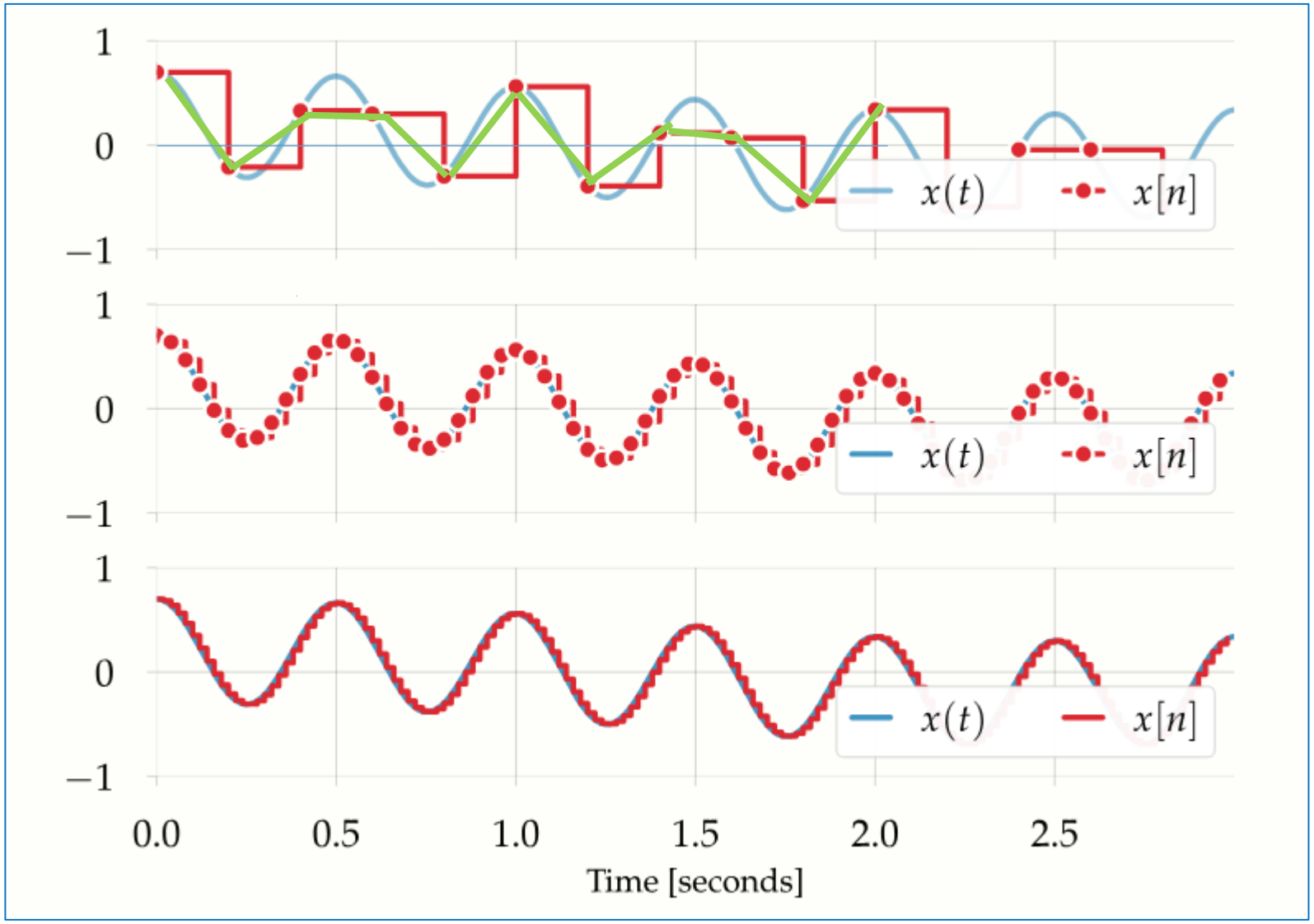
1002, 8, 24, 645, 2, 786, etc
De DSP ziet getallen en ‘weet’ niet wat het voorstelt

- Verwerkingstijd in onze ontvangers van antenne tot luidspreker…
  Graag een beetje snel alstublieft, b.v. liefst binnen 30 ms.
  Door dingen parallel te doen kan het tegenwoordig heel snel: het toverwoord is… **FPGA**
  (Field Programmable Gate Array = een set configureerbare poorten)

Signaal in samples (eigenlijk een foute voorstelling)

‘minder snel’

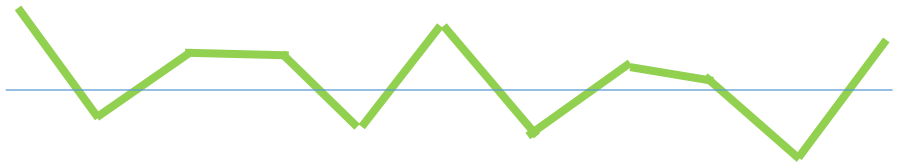
‘snel’ samplen



x(t) = signaal in de tijd

x[n] = signaal in samples

Signaal in samples

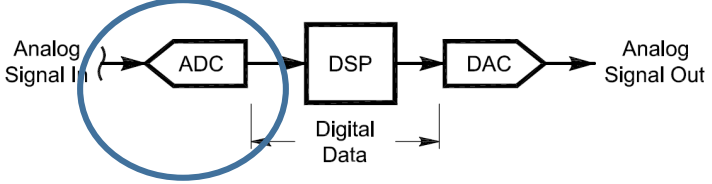


- Ziet er niet meer uit als die oorspronkelijke golfvorm…
- Je moet ze ook niet proberen te verbinden!
  Het zijn maar getallen, waarden op een bepaald moment.

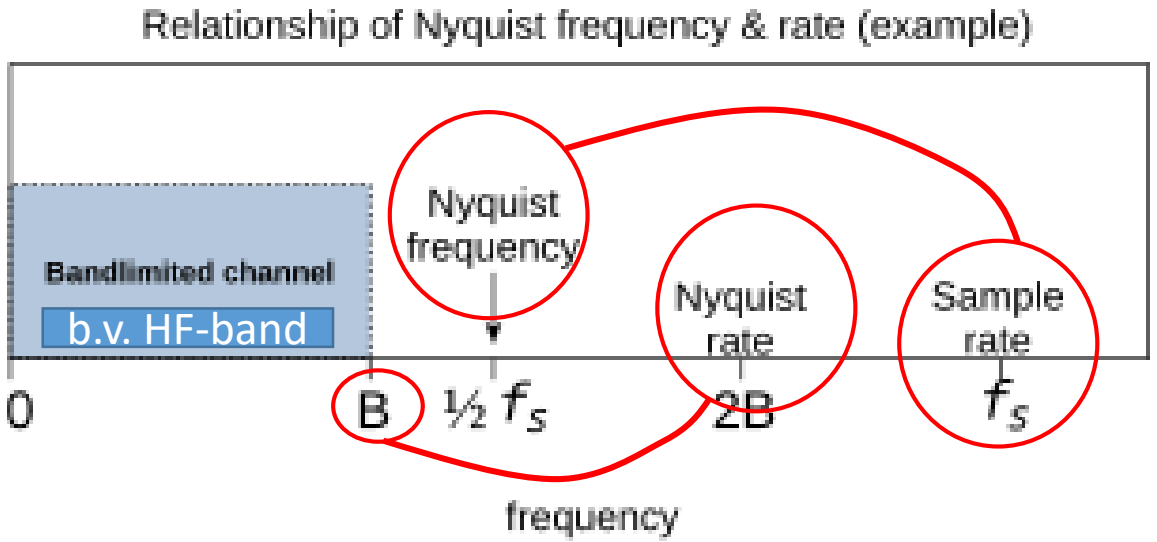
- Is die oorspronkelijke sinus uit deze samples nog terug te halen?

- Wat heb je nodig?

Sampling is niet zo maar sampling



- **Meten** van een (analoog) signaal **op vastgestelde tijdstippen** en het **digitaal vastleggen** van de gemeten waarde doen we met een **Analog – Digital Converter** - of kortweg **ADC**.

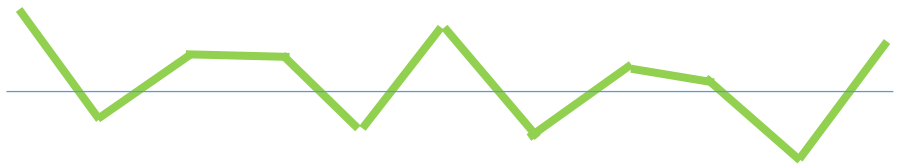


De kretologie

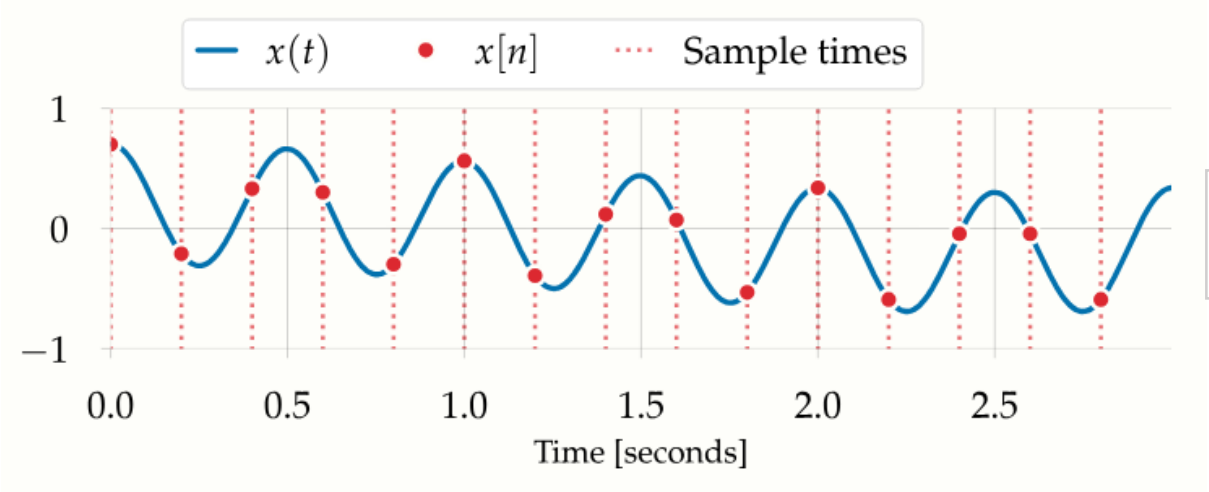
- **Claude Shannon**: er is geen informatie verlies van het signaal als de ***Sample rate*** groter is dan de ***Nyquist rate,*** dit komen we ook tegen als het ***Nyquist Sampling Criterium***.
  - De ***Nyquist rate*** is 2x bandbreedte van het gesampelde signaal (= minimaal benodigde Sample rate)
  - De ***Nyquist frequentie*** = ½ x ***Sample rate***

- ***Om de bandbreedte van het gesampelde signaal te beperken gebruiken we vooraf een analoog filter. We noemen dit een 'anti-alias filter'. Dit filter zit dus ook in een SDR ontvanger (preselectiefilter) en dat is nog steeds hartstikke analoog.***

Is die oorspronkelijke sinus uit deze samples nog terug te halen?



De samples x[n] als getallen, verbonden



Het was dit

- Laten we nog eens kijken

Sampling

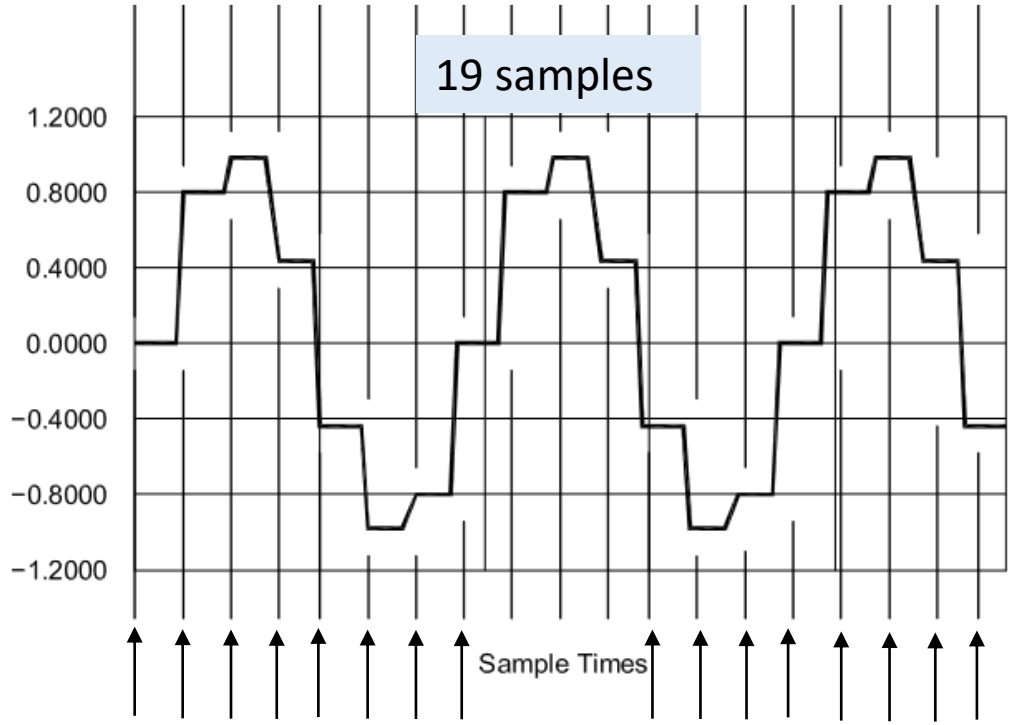
Hoge sampling frequentie voor grote bandbreedte is zalig makend?

- Ja
  - grote bandbreedte - vinden wij amateurs erg leuk
  - technisch positieve invloed op SNR (verklaring volgt later)
- Nee
  - erg veel data om te verstouwen - er zijn limieten…
  - stel dat je een frequentie van 1 Hz nog wil zien moet je 1 seconde lang samplen en die als geheel bekijken – bij elkaar heel veel samples om te analyseren

Bron : ARRL Handbook

Sampling foutjes

- Te traag samplen = te lage sample rate



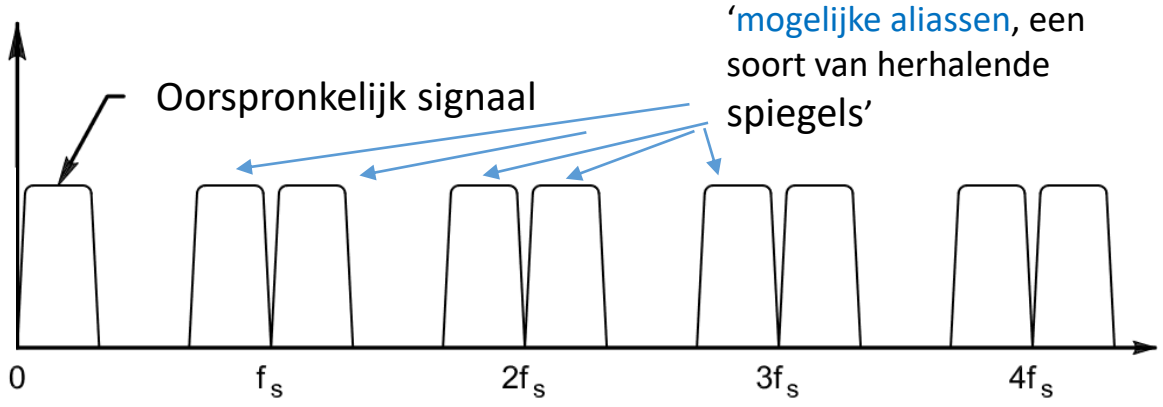
Ook weer doorverbonden,
Klopt dit wel?

Hoe zit het nou precies met die aliassen?

Een beetje achter de schermen: de samples

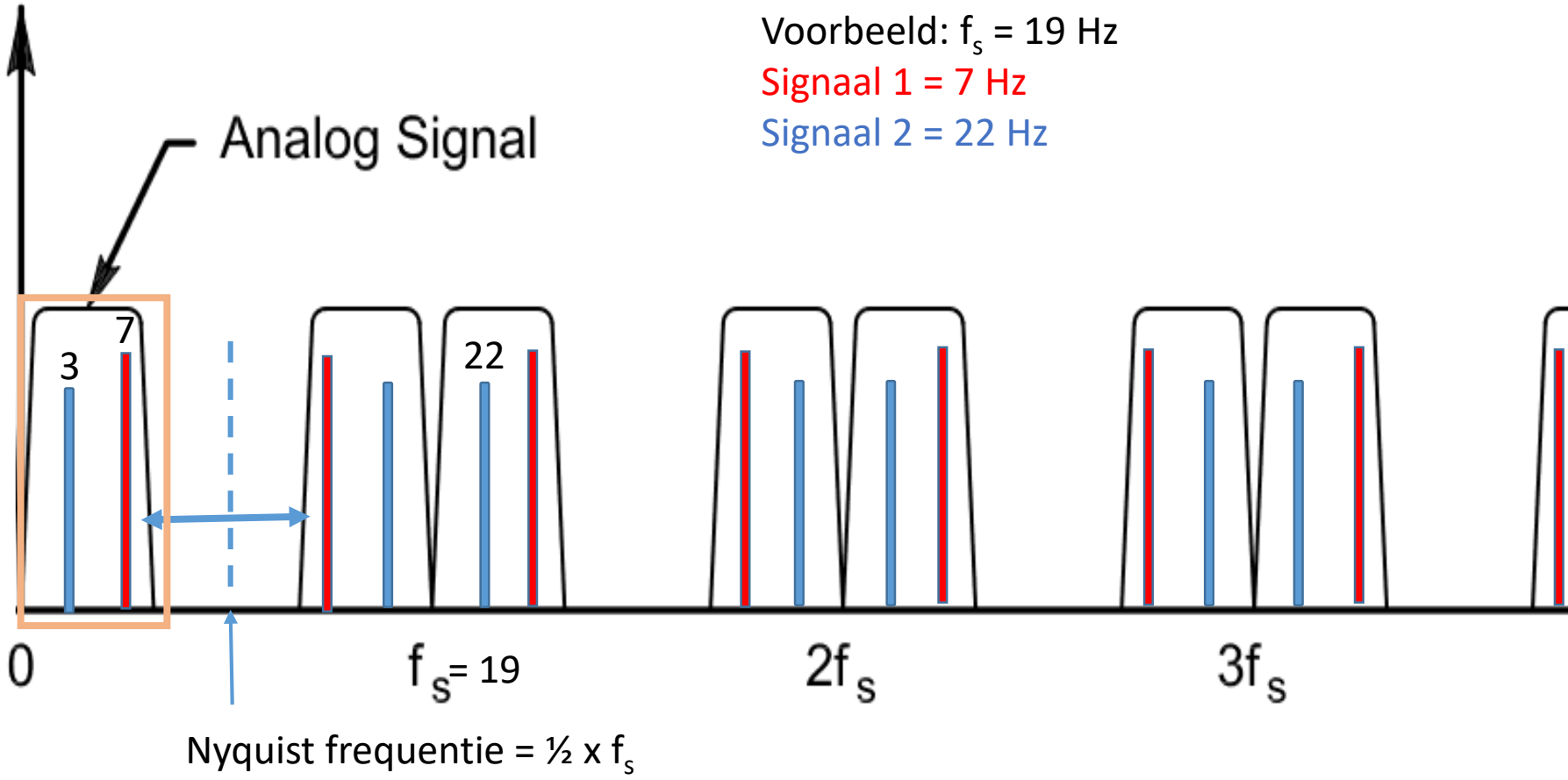
- Samples zijn **pulsen** met een discrete waarden, het zijn *dus geen blokjes*.
  Immers de waarden zijn bepaald **per *moment*, niet per tijdsperiode**
- We weten na sampling niet wat voor golfvorm het is. Misschien een alias?
- Dat heeft tot gevolg dat we rekening houden met:
  - 'Spiegels' rondom de n x de samplefrequentie dus f_s , $2f_s$, $3f_s$, enz
    (*dit is een gevolg van het rekenen met samples*)
  - Deze spiegels zijn ook de mogelijke aliassen van het oorspronkelijke signaal
  - (de wiskunde is geduldig… er is wiskundig ook exact hetzelfde gespiegelde negatieve spectrum)

Bron : figuur uit
ARRL Handbook



Alias

Het zgn. anti-alias filter zorgt voor het beperken van de bandbreedte zodat de spectra elkaar niet gaan overlappen bij de Nyquist frequentie
Een anti-alias filter is echter nooit ideaal)

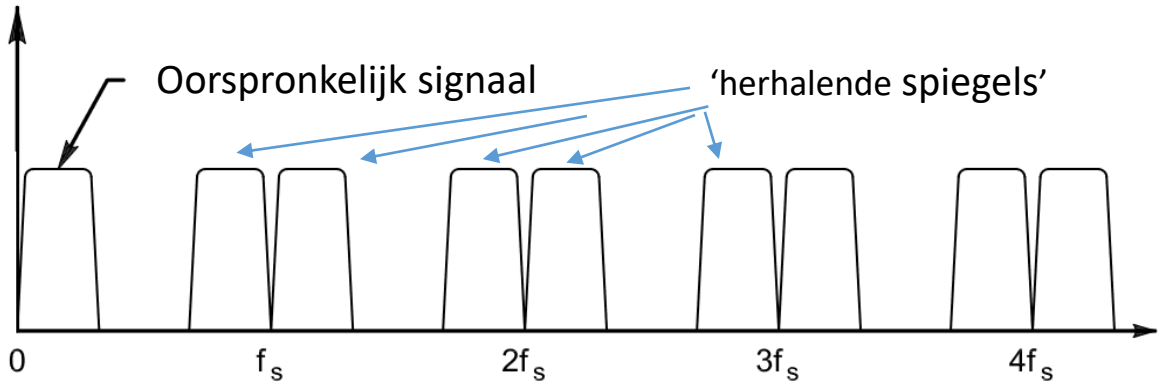


Voorbeeld: f_s = 19 Hz
Signaal 1 = 7 Hz
Signaal 2 = 22 Hz

Is het erg?

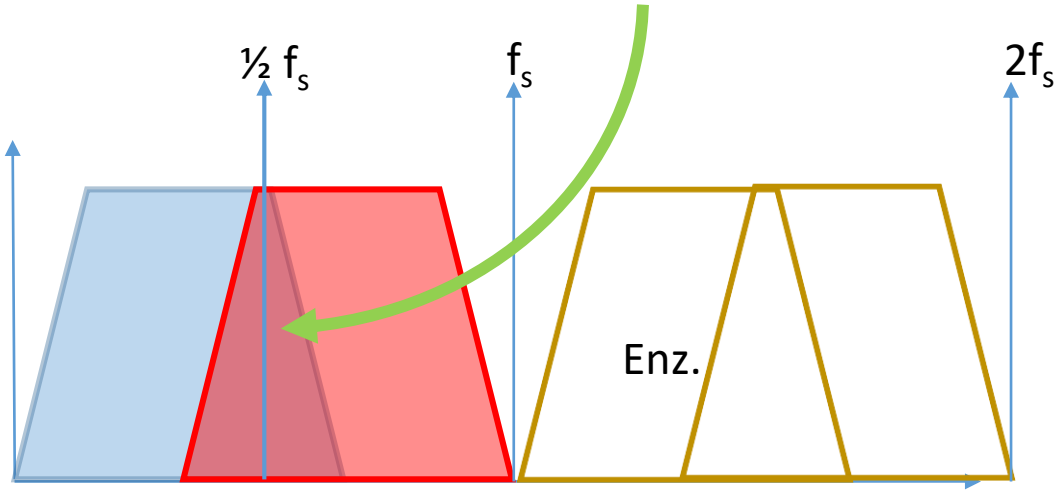
(die spiegels)

Bron : ARRL Handbook



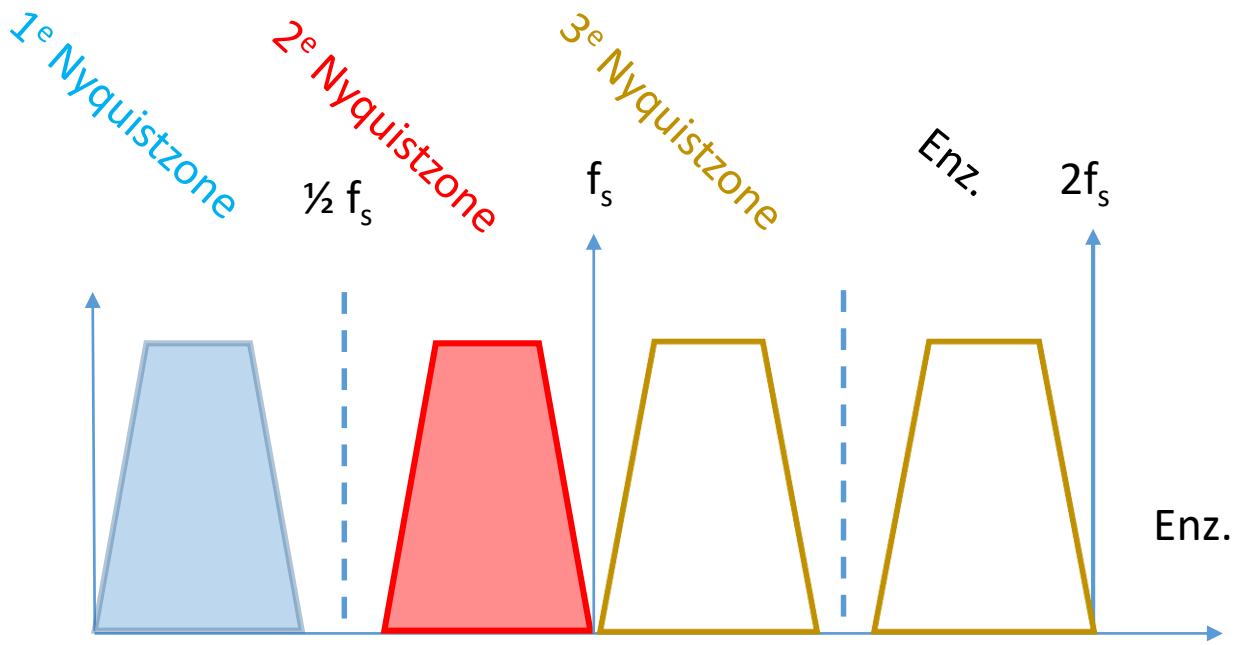
- Nee, zolang WIJ maar weten wat er werd bedoeld.
- En… zolang er maar geen overlapping van de spectra ('**aliasing**') plaatsvindt!
- Goede anti-alias filtering is super belangrijk → Bandbreedte B < ½ f_s

De hoge frequenties vouwen a.h.w. terug.



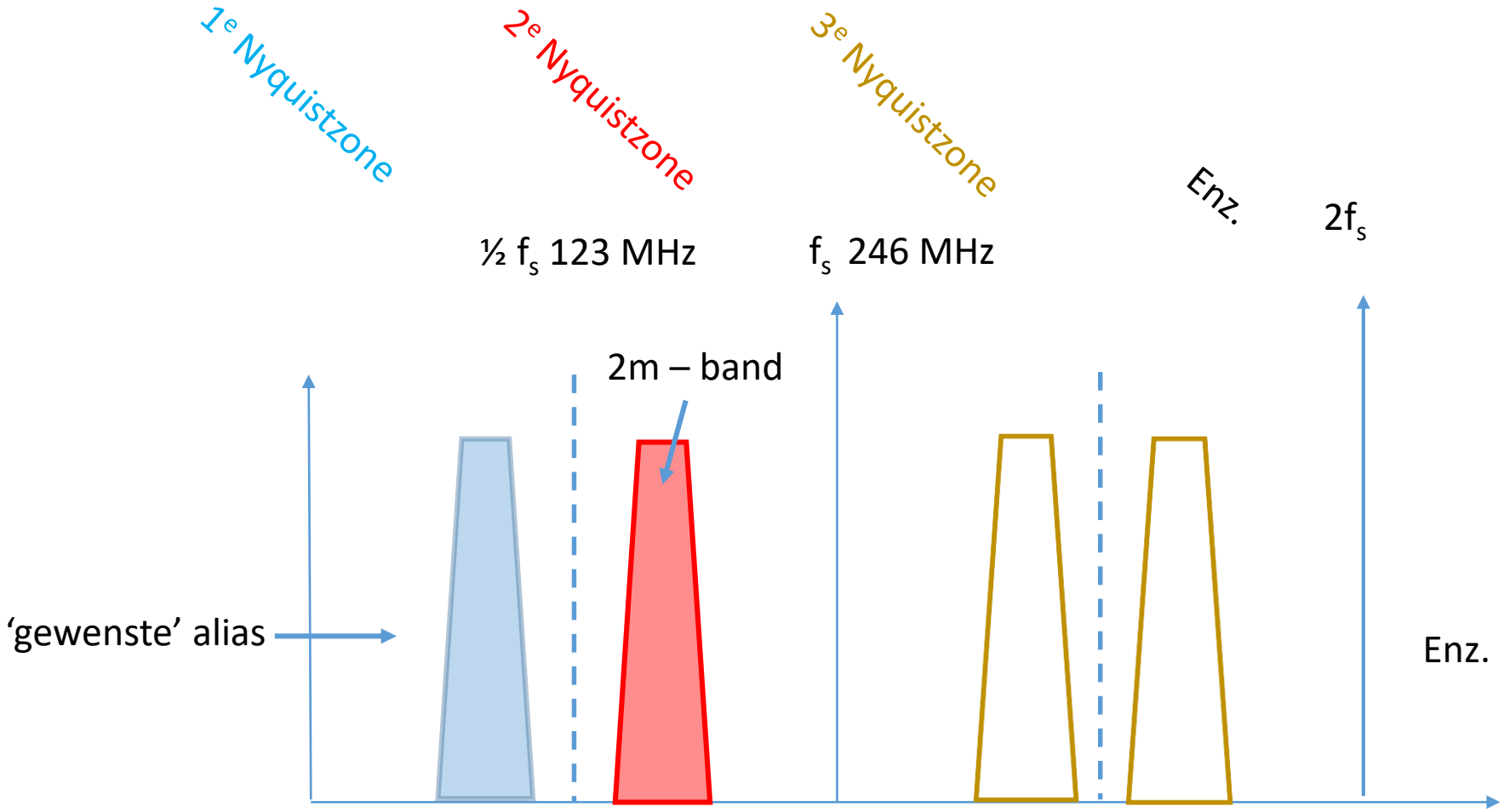
Zgn. Nyquistzones

- De plaatsen waar de spectra uit de vorige figuren staan noemen we ‘Nyquistzones’. Daar kunnen we nog iets mee.
- Dit is nog steeds die DSP voorstelling die in de sampleserie zit ingebakken.



FlexRadio 6700 op 2m



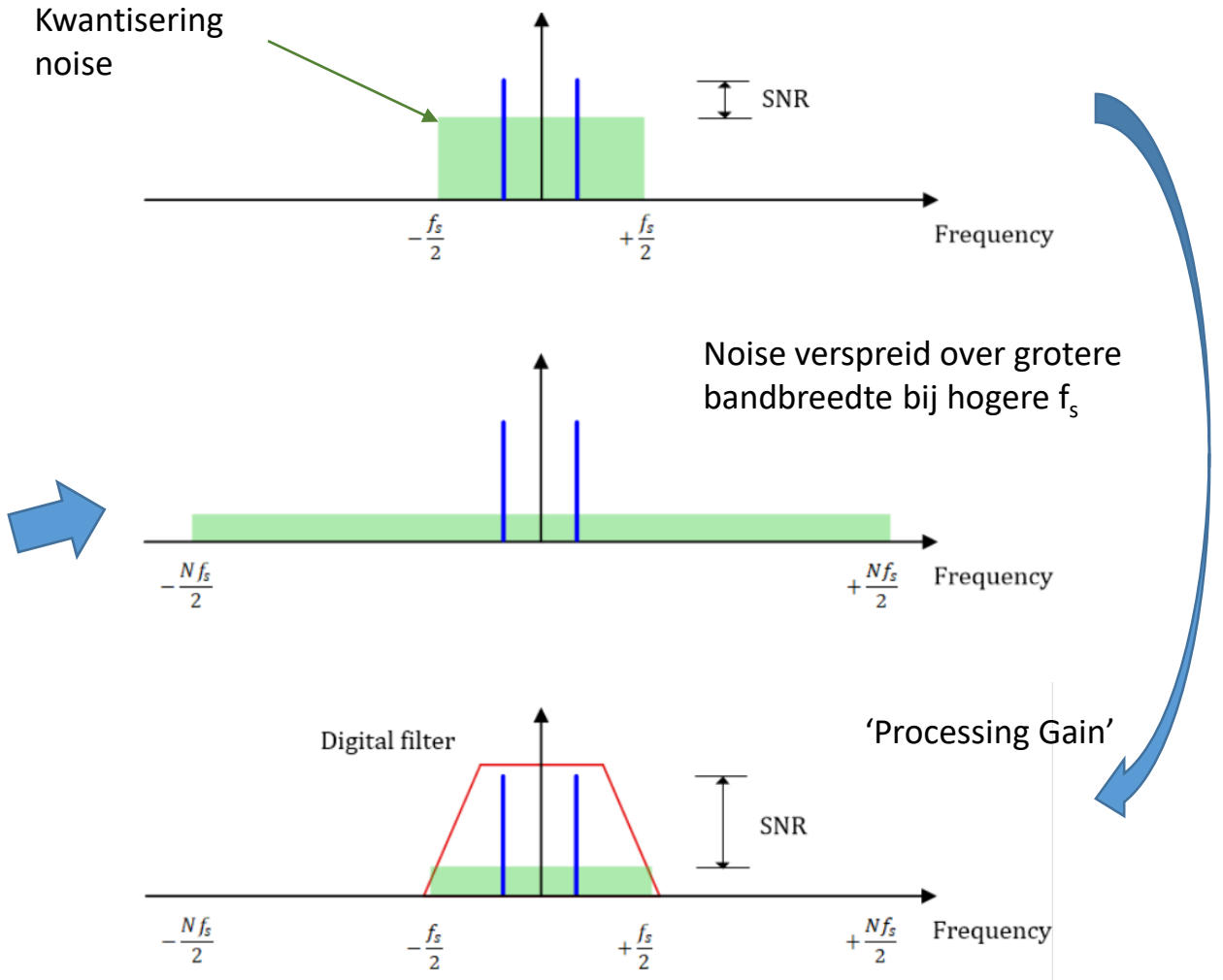


Er is een verschil tussen analoge werkelijke waarde en getalswaarde die de ADC rapporteert
b.v. *0,39875* (echte waarde) versus *614* (volgens de AD-converter)

Kwantiseringsruis

Signaal / Ruisverhouding in de ADC – aantal bits en sample rate

- **Max S/N ratio =**
  Full Scale signal/ kwantiseringsruis
  (zie sampling)
- Voor ***IDEALE ADC***: Max **S/N ratio =**
  6,02 x aantal bits + 1,76 dB
  16 bit → S/N = 98 dB
- Iedere bit extra levert 6dB extra S/N ratio op
- Iedere verviervoudiging van de samplerate Fs levert 6dB extra SNR op door noise verspreiding. Dit is a.h.w 1 bit extra!
- *Maar nogmaals, een ADC is nooit ideaal* ☺



En ook dat nog:
De stapgrootte van de ADC is niet constant

- D.w.z. als per 0,1V verschil de ADC met 1 LSB waarde moet veranderen, dan gebeurt dat niet. Het varieert een beetje.
  Afwijking t.o.v. ideale curve: DNL en INL

- ADC Differential Non Linearity (DNL) (Stap afwijking)

$$DNL(k) = \frac{W(k) - W_{ideal}}{W_{ideal}}$$

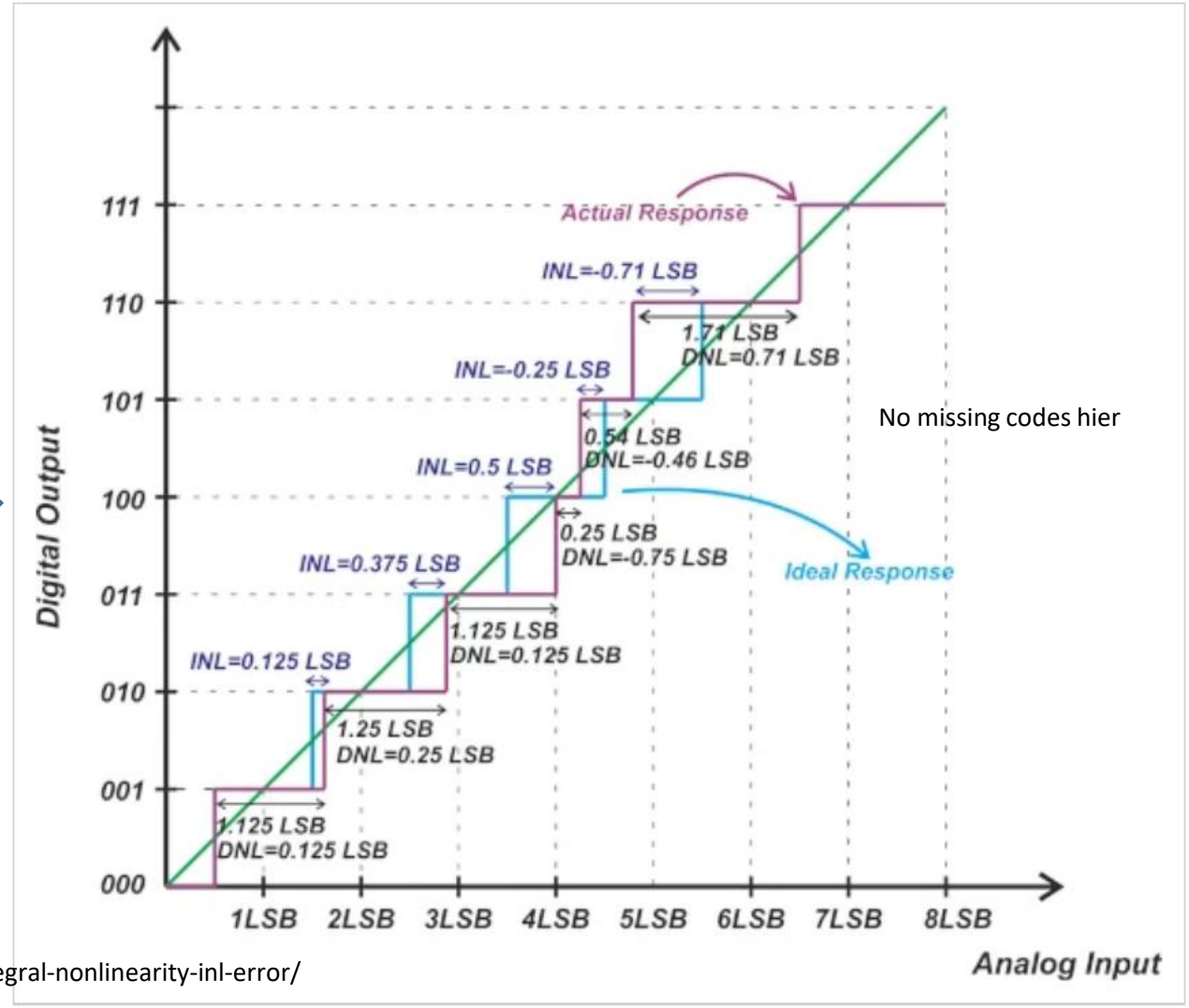
  Afwijking kan een max een hele bit bedragen

- ADC Integral Non Linearity (INL) (totale afwijking)

$$INL(k) = \frac{T_a(k) - T_{ideal}(k)}{Ideal\ Step\ Size}$$

  Totale afwijking (gemeten waarde t.o.v. echt) kan meerdere bits bedragen

- Al met al weer een bijdrage aan meetruis



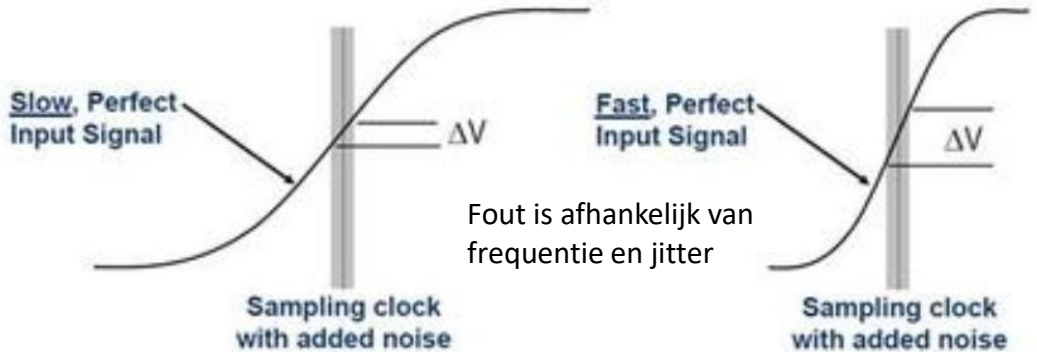
No missing codes hier

https://www.allaboutcircuits.com/technical-articles/understanding-analog-to-digital-converter-integral-nonlinearity-inl-error/

Klok jitter van de ADC

(de klok dicteert de sample momenten)

SNR in de ADC en klok-jitter



- Door de jitter op de ADC clock ontstaan fouten in de gemeten waarden en dit uit zich in ruis.
- Uit de figuur blijkt:
  - Hoe meer jitter, des te meer foutruis
  - Hoe hoger de signaalfrequentie, des te meer foutruis
- Volgens analog.com is de maximale SNR van de ADC wat betreft jitter:
  max SNR = -20 x log($2\pi f_{signaal}$ x t_{jitter})
  $f_{signaal}$ = **gesampelde** frequentie (niet de sample frequentie), t_{jitter} de hoeveelheid **jittertijd (RMS)**

$f_{signaal}$	t_{jitter}	SNR
15 MHz	80 fs	102 dB
50 MHz	100 fs	90 dB

Kortom:

- Het dynamisch bereik van de ADC wordt bepaald door de verhouding tussen:
  - Full Scale signal waarde (FS)

  en
  - De verschillende ruisfactoren bij elkaar opgeteld:
    - Temperatuurruis (hier ook aanwezig)
    - Kwantiseringsruis (afwijking op Least Significant Bit)
    - Jitter op het kloksignaal
    - Niet lineariteit van het kwantiseren (DNL en INL)

- Dit is in de datasheet uitgedrukt als: dBFS Noise Floor
  (Full scale waarde t.o.v. Ruisvloer)

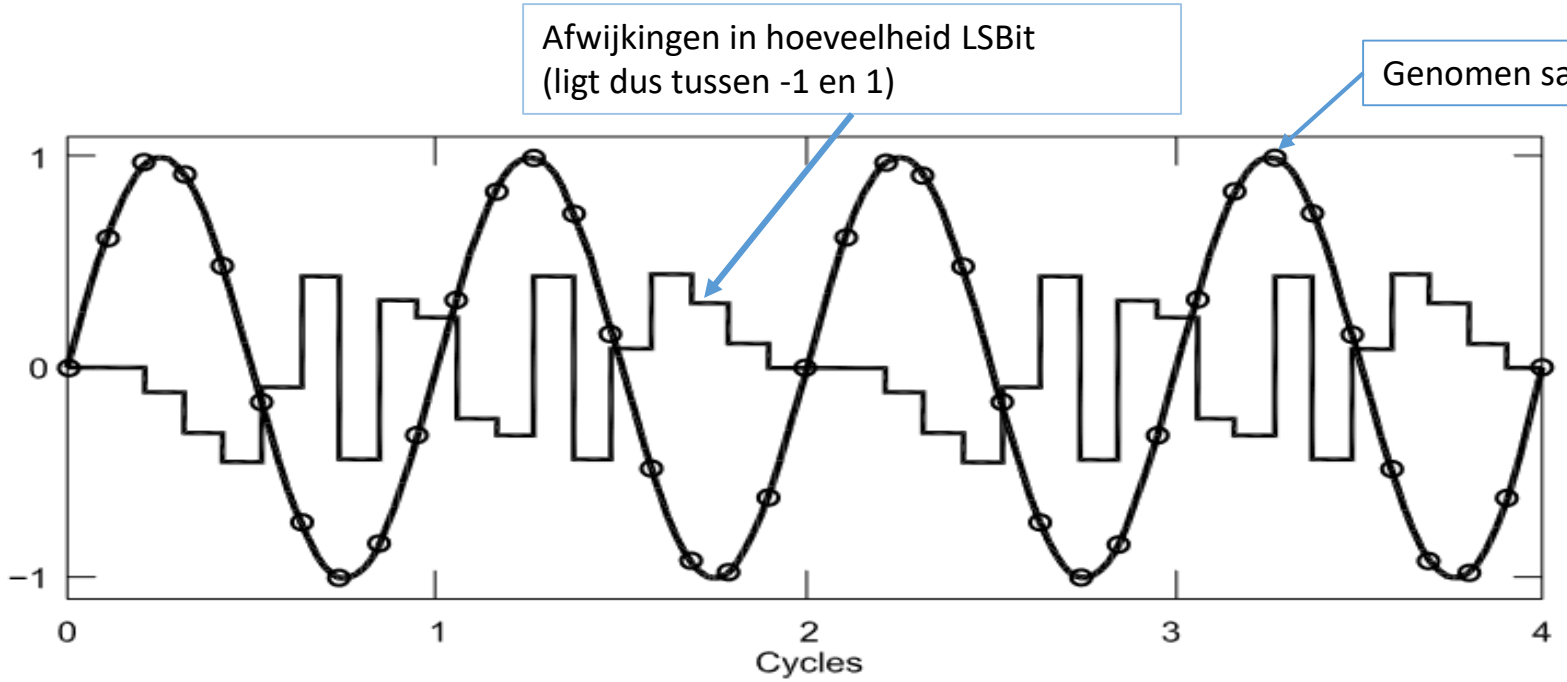
Ook dat nog… keiharde signalen

(Nog steeds kleiner dan de maximale spanning die de ADC kan hebben, geen overload)

Analoge sets vonden dat ook niet altijd leuk…

Harde signalen

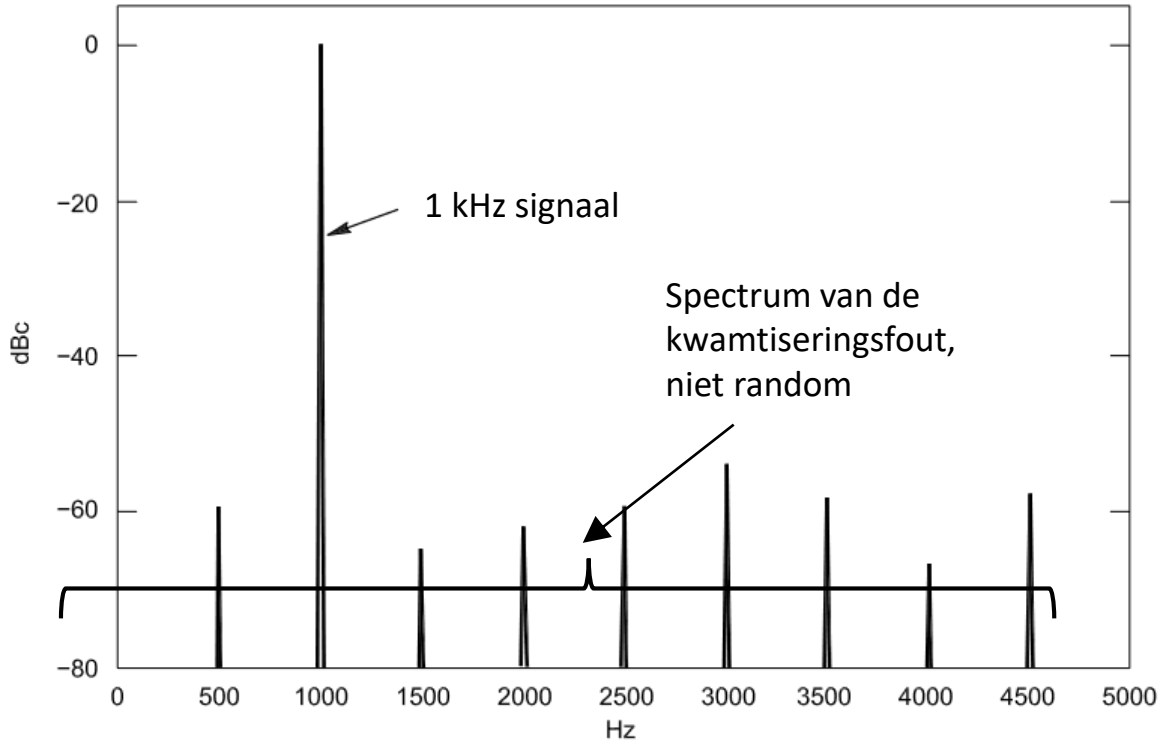
- Normaliter is kwantiseringsruis random en dus breed verdeeld over het spectrum, zie de eerdere opmerkingen over SNR
- Stel dat er een heel hard signaal of meerderde signalen de ADC domineren → de kwantiseringsfout wordt dan voornamelijk hierdoor bepaald en is ***niet meer random maar periodiek***!
- In de figuur:
  De sinus wordt gesampeld. Het bloksignaal geeft de kwantiseringsfout weer in aantal LSBit. Je ziet een zich herhalend foutsignaal (periodiek). Dit is géén random noise meer en het heeft dus een bepaald spectrum!



Signaal van 1 kHz
gesampeld met 9,5 kHz
ADC: 8 bits

Harde signalen gevolg

- De periodieke kwantiseringsruis heeft een eigen spectrum gekregen dat duidelijk zichtbaar is
- Dus: Spurious… **Schande!**
- Het niveau dat nog geen spurious boven de ruisvloer laat zien, heet Spurious Free Dynamic Range: '**SFDR**', de nu vaste kwaliteitsparameter voor SDR i.p.v. de analoge 'IP3'.



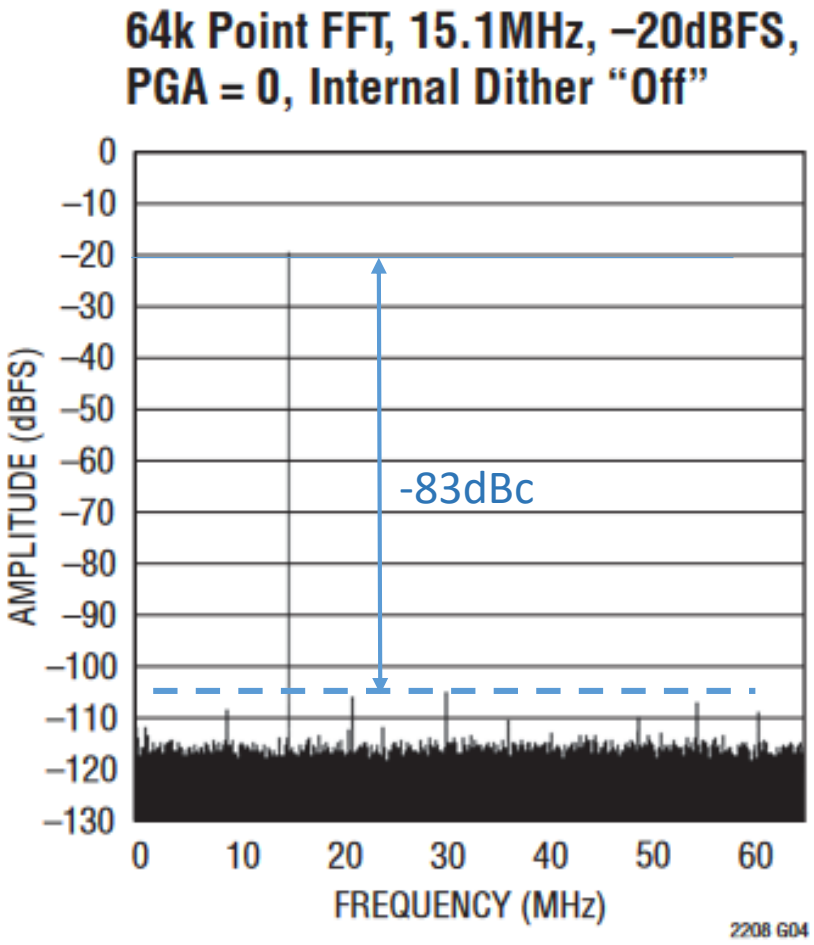
Te hulp: ***DITHERING*** ***(Icom-7300: IP+)*:**

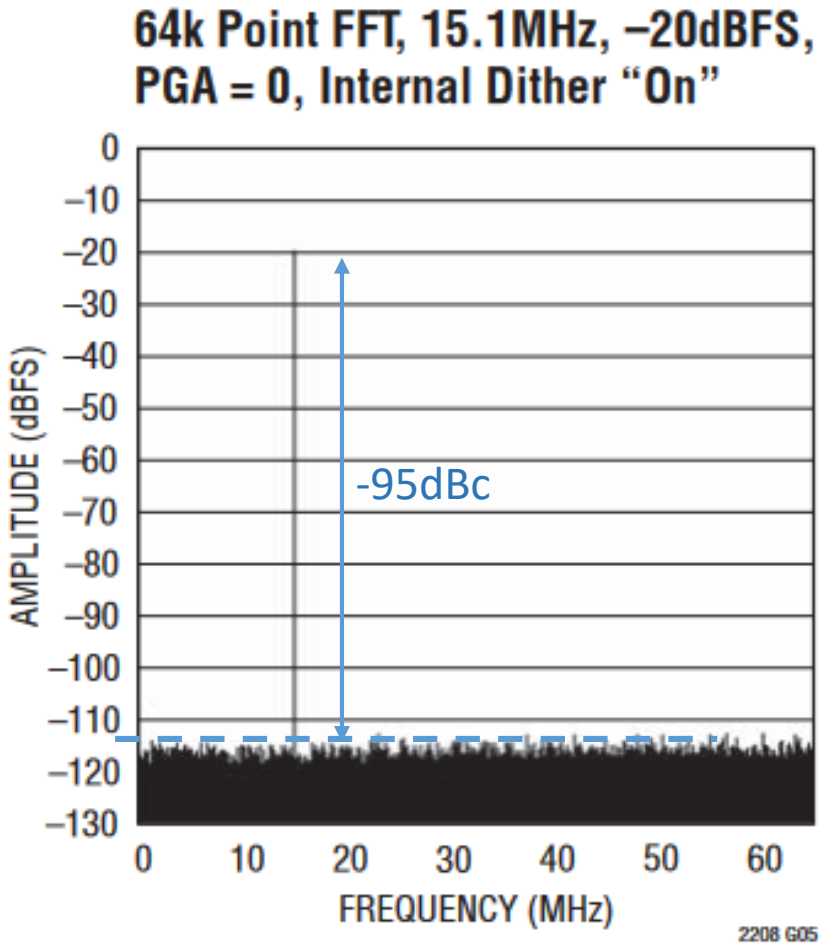
- Door extra randomruis toe te voegen aan de ADC, ordergrootte ca. 1á2 LSBits groot, wordt de kwantiseringsfout weer random gemaakt
- **Weg Spurious**…
  (… en weer een plekje omhoog in de Sherwood list) maar wel een hogere ruisvloer, b.v. bij een ic-7610 +3dB

ADC van de IC-7610

Dithering on/off vergelijking

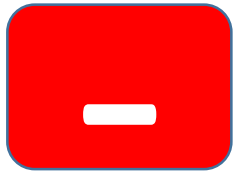
- PGA = programmable Gain amplifier
- dBFS = dB t.o.v. Full Scale
- dBc = dB t.o.v. de gestuurde carrier





De kreet: *ENOB*

- Praktijk meting voor S/N: **ENOB** – Effective Number Of Bits
- Definitie = (de ***echt gemeten*** SINAD (in dB) – 1,76dB) / 6,02
- Praktische waarde voor een 16-bit ADC (alle sh*t inclusief):
  - SNR van 72 dB → een 12,5 bit ADC.
  - Die 12,5 bits noemen we de ENOB





Maar…

- Stel ADC samplerate is 250 MS/s (b.v. van Flexradio)
- Later in de RX werken we met kleinere bandbreedtes en lagere samplerates, b.v. 24 kS/s
- De signaal sterkte blijft hetzelfde maar de ruis binnen de bandbreedte gaat factor 10.000 omlaag dus 40 dB ‘Processing Gain’
- SNR van het geheel van wordt dan 72 + 40 = 112 dB = 18-bit ENOB

Take away:



- De ADC meet niet 100% juist, zie o.a. sheetjes over sampling fouten, het jitter verhaal en de non-linearity, etc.
- Dithering helpt het Spurious Free Dynamisch bereik van de ADC

En:

- Extra ruis door de hardware om de ADC heen (filters, pre-amps, etc)
- De anti aliasfiltering (b.v. met bandfilters) is nooit ideaal

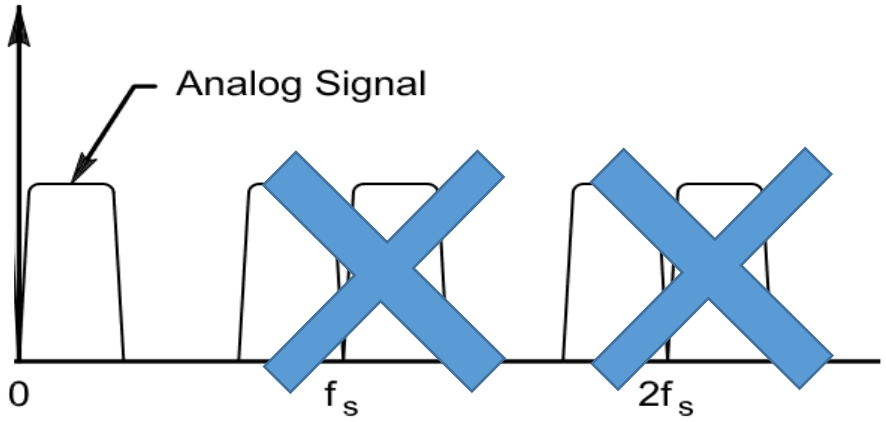
- **Ja, een ‘full SDR’ heeft genoeg analoog spul eromheen dat de zaak beïnvloedt**

Na de DSP weer terug naar analoog

D/A Converter

Na het DSP: terug van de samples naar analoge werkelijkheid

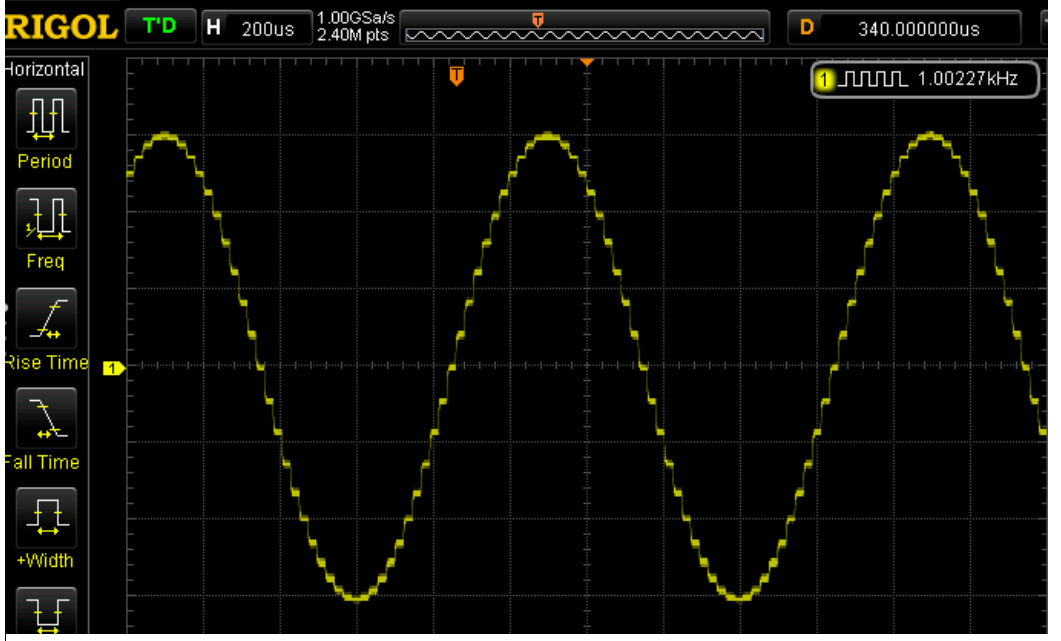
- D/A Converter: Digital to Analog Converter → van getal naar een spanning
- Zodra we verder naar de analoge output gaan (met een DAC), terug in de werkelijkheid, dan dienen alle hogere ‘spiegels’ te worden onderdrukt.
- Dit is de taak van het zgn. *reconstructie* filter. Dit is analoog.
  Vaak is dit een laag-doorlaat filter.



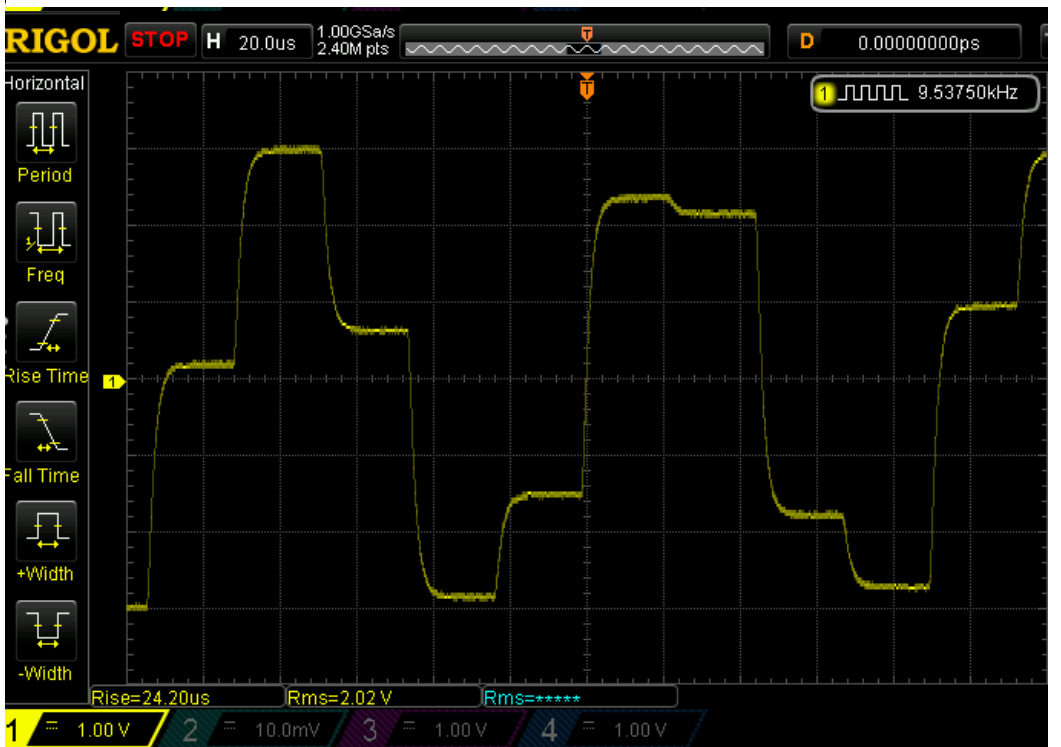
Intuïtieve benadering van de output van een DAC:

- Hoge sample rate bij de DAC ziet er minder slecht uit dan een signaal met lage samplerate.

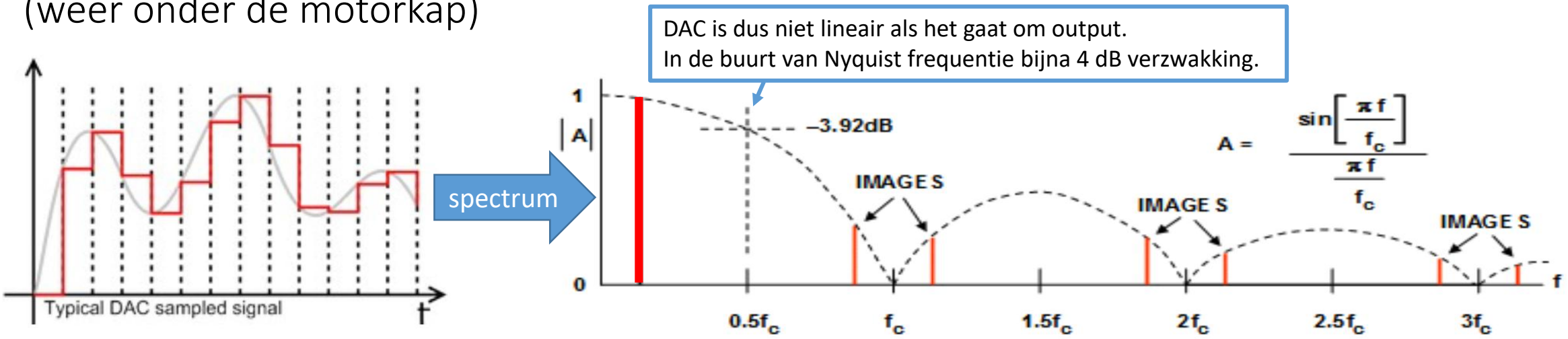
- Na deze blokkige output van de DAC volgt een analoog reconstructie filter. Dat is bij de onderste figuur een hele klus.



Bron: addictedtoaudio.com.au



DAC (‘zero order hold’) en het reconstructiefilter
(weer onder de motorkap)



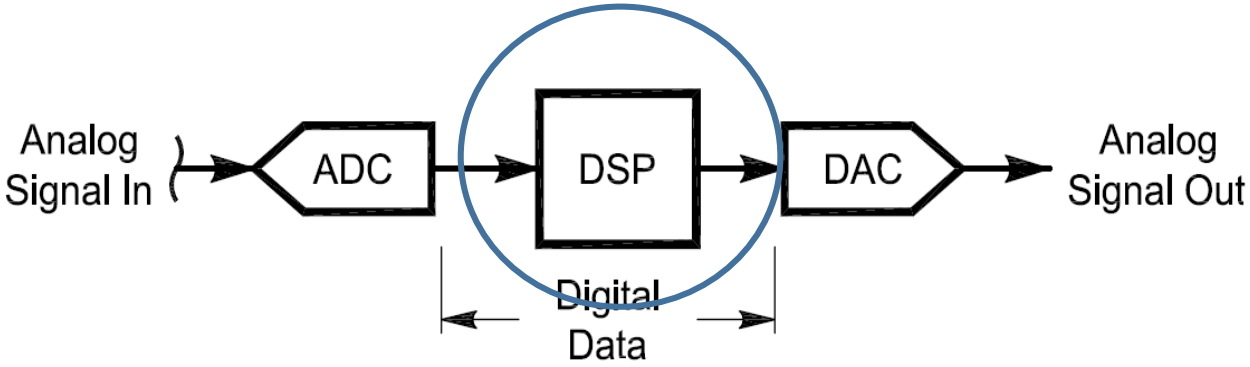
- DAC output en bijbehorend output-spectrum en heeft de vorm van de zgn. sinc-funtie.
- f_c is de sample frequentie.
- Zichtbaar is de verzwakking van de spiegels (images) rondom de harmonischen van de sample frequentie.

- **Hoe hoger de sample frequentie t.o.v. frequentie die gereconstrueerd moet worden, des te beter de onderdrukking.**
- **Maar hoe hoger de audio frequentie t.o.v. de sample frequentie, des te sterker de spiegels**
- **Het analoge ‘reconstructiefilter’ hoeft maar een deel te doen, als je het slim aanpakt.**

Anekdote: cd-spelers

2. SDR → de DSP

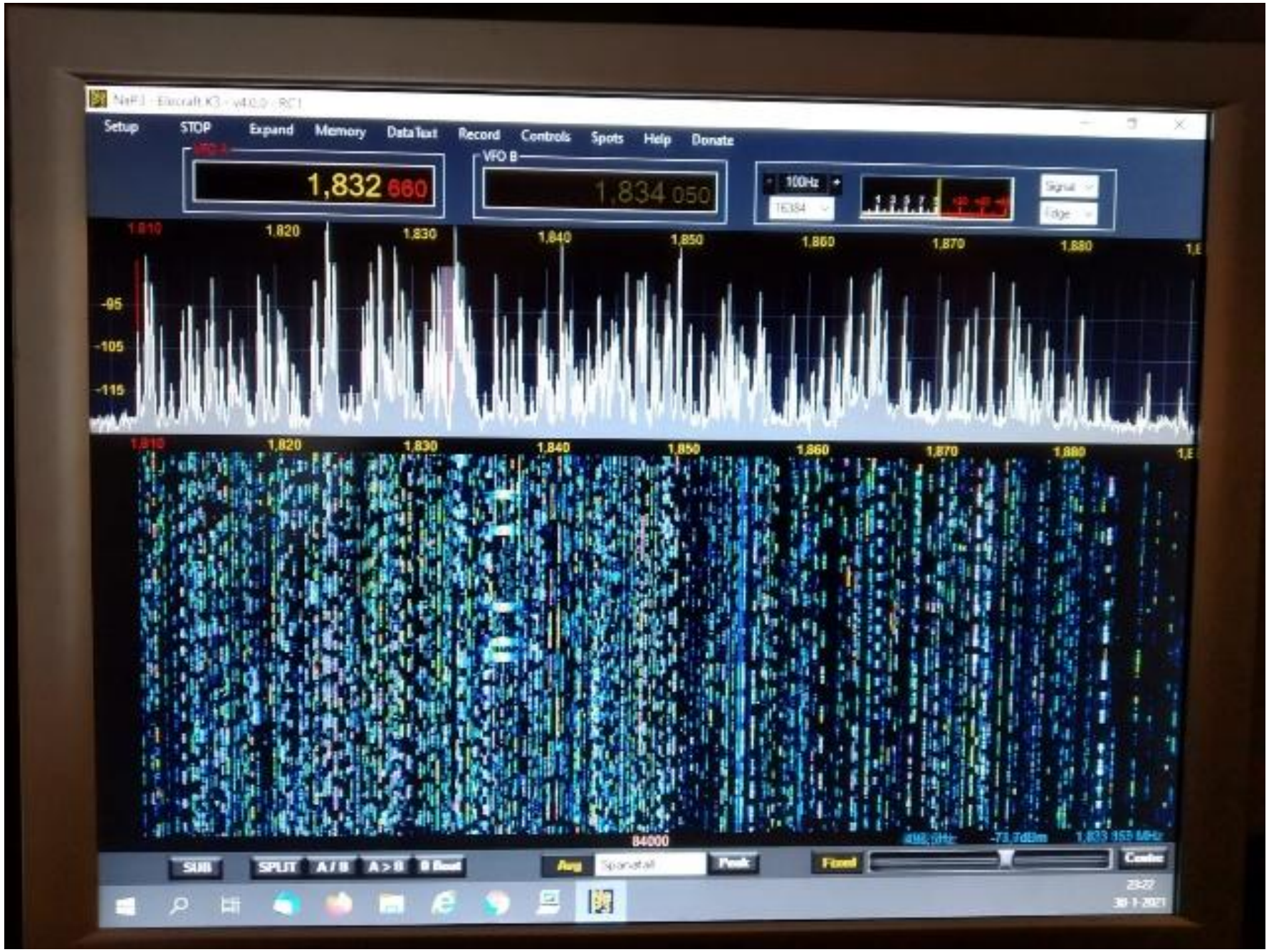
(d.w.z. de DSP berekening bepaalt ***voor een groot deel*** wat voor radio je hebt)



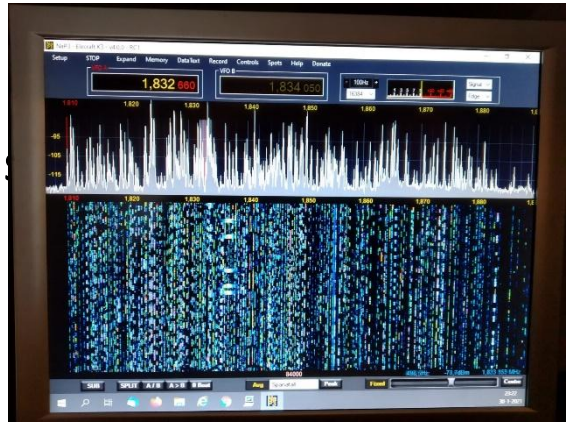
- De FFT – panadapter –
  (deze vinden we mooi)
- Verschillende sample rates in de DSP
- I-Q signaal, wat is dat
- Positieve en negatieve frequenties
- Verschillende filter typen

Een FFT scherm

(dit is een *benadering* van de werkelijkheid)



De Fast Fourier Transform - FFT - (wie kent hem niet)



Fourier Transform:
Vanuit een beschreven signaal in de tijd (tijdsdomein) naar een frequentie
Andersom heet het de Inverse Fourier Fransform

De twee *voor ons bekendste* Fourier transformaties:

1. **Fourier Series – voor periodieke signalen en uitrekenen van harmonischen**
   (befaamd voorbeeld: blokgolf met de 'oneven' harmonischen, analoge berekening)
2. **Discrete Fourier Transform (DFT) – *digitaal periodiek* (= onze interesse)**
   Een slimmere en snellere rekenmethode van de DFT is die FFT maar doet verder hetzelfde.

- Het principe van de **FFT** werd al gevonden door Gauss in 1805; in 1965 publiceerden Cooley & Tukey de methode

Wat doet de Discrete Fourier Transform (FFT) eigenlijk

Serie discrete serie getallen (de samples)



serie getallen die een frequentiespectrum voorstellen
(daar kunnen wij zendamateurs wat meer mee ☺)

Hier zijn een paar waarheden over de FFT…

Voorbeeld voor een FFT



- Gesampelde signaal: 3 Hz sinus

- Sample rate: 64 Hz
- Nyquist frequentie = 64 : 2 = 32 Hz (ruim voldoende)

- Hoe lang samplen voor de FFT : 8 s
- Totaal aantal samples: 8 x 64 = 512 samples

- In de FFT is de resolutie in dit geval: 64 : 512 = 0,125 Hz
  (hoe meer samples, des te hoger de nauwkeurigheid / resolutie)

Voorbeeld

- Signaal: 3 Hz
- 512 samples
- Begin sluit aan bij eind
  (past precies, ‘coherent’)

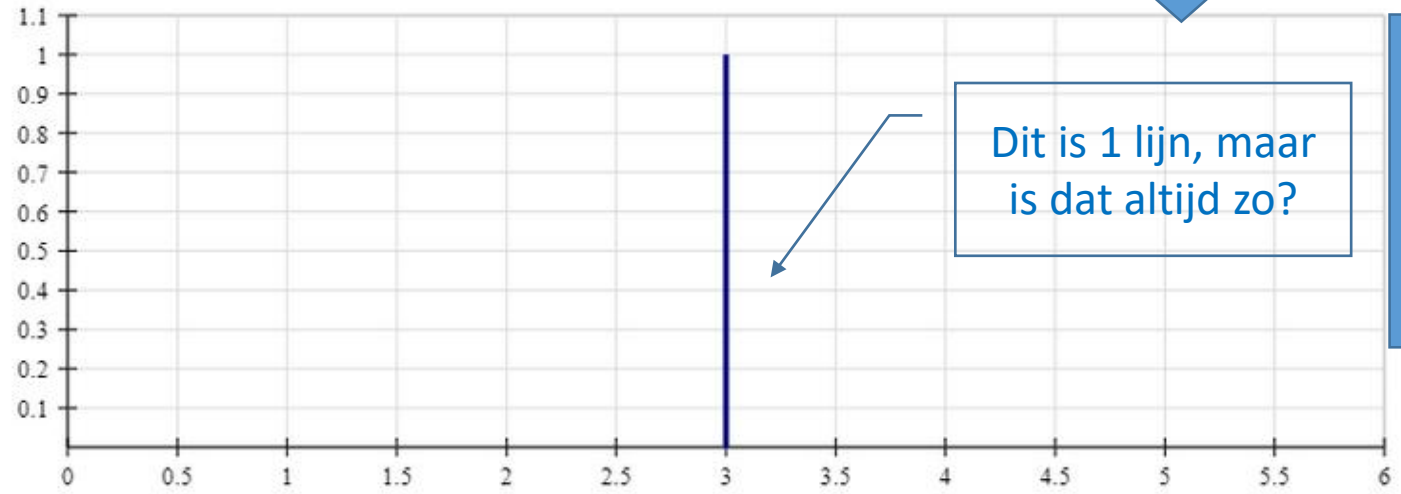
- Precieze uitkomst:
  - 256 ***bins*** gevuld
    (= ½ x aantal samples)
  - 1 lijn op 3 Hz.
    is **precies** de 24e bin,
    24 x 0,125

- Hoe meer samples, des te meer en des te smaller de bins (betere resolutie)
  (en des te dieper de ‘FFT scherm ruisvloer’ op je transceiver)



Un-windowed 3Hz Cosine Wave

FFT



Dit is 1 lijn, maar is dat altijd zo?

Als het wat breder is, betekent dat dan een slecht (breed) signaal?

Frequency Spectrum of the Un-windowed signal

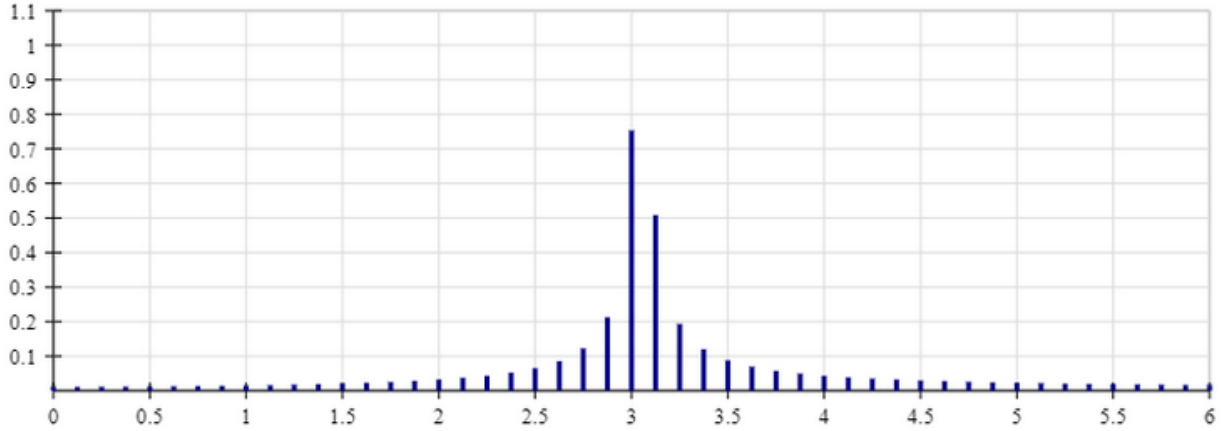
- Signaal: 3,05 Hz
- Ook 512 samples
- Begin sluit NIET aan bij eind
  Er past niet een heel aantal sinussen binnen de samples en day past dan niet precies in een bin



A 3.05Hz Cosine Wave in the Time Domain

- Precieze uitkomst:
  niet helemaal zichtbaar en erg breed. Er is geen bin voor 3,05 Hz dus het verspreidt zich

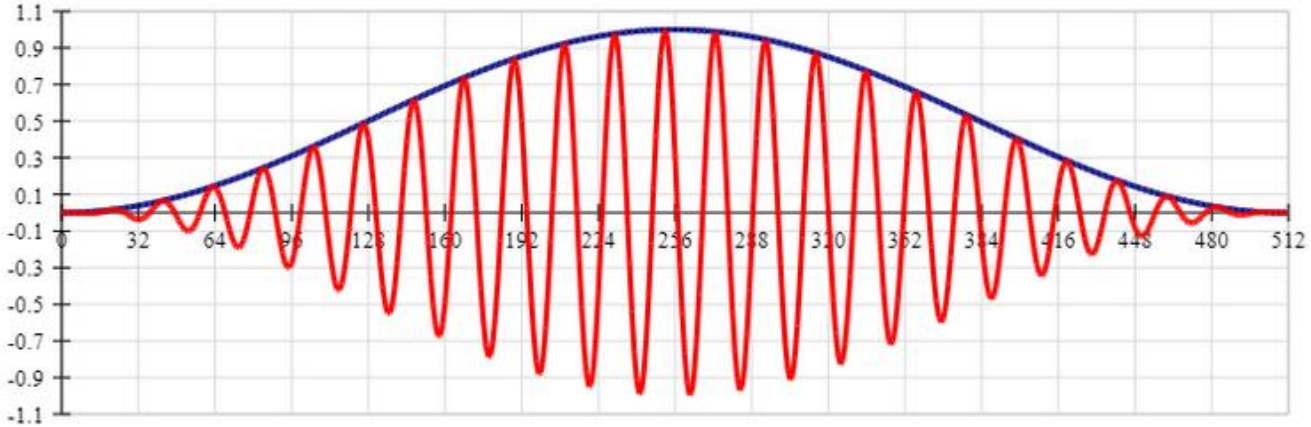
- Dit heet
  ‘Spectral Leaking’



Spectral Leakage in a 3.05Hz Cosine Wave in the Frequency Domain

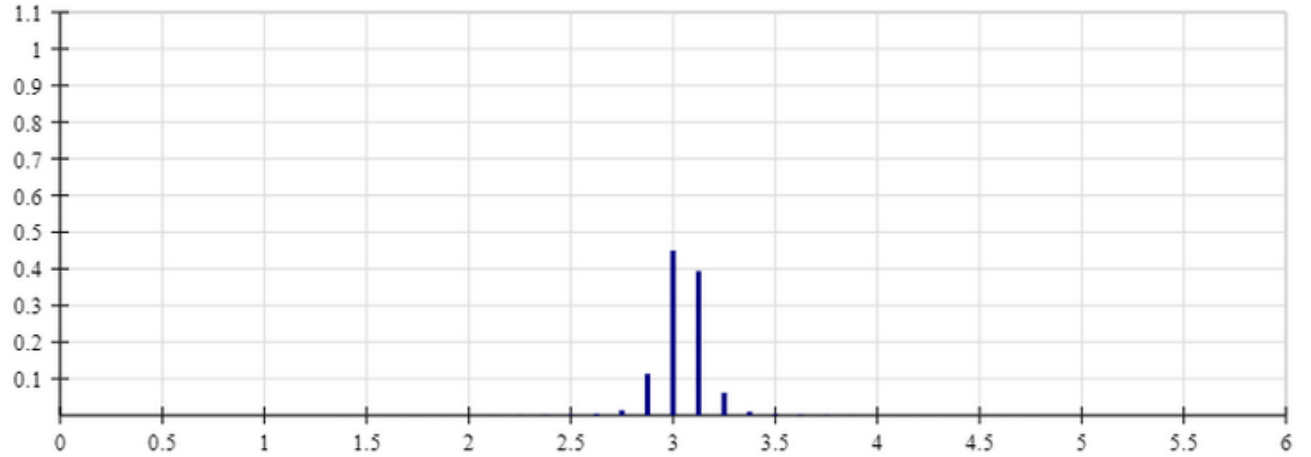
Toepassing van een ‘window’

- Signaal: 3,05 Hz
- Ook 512 samples
- Begin sloot NIET aan bij eind
- We passen een ‘Hanning Window’ toe



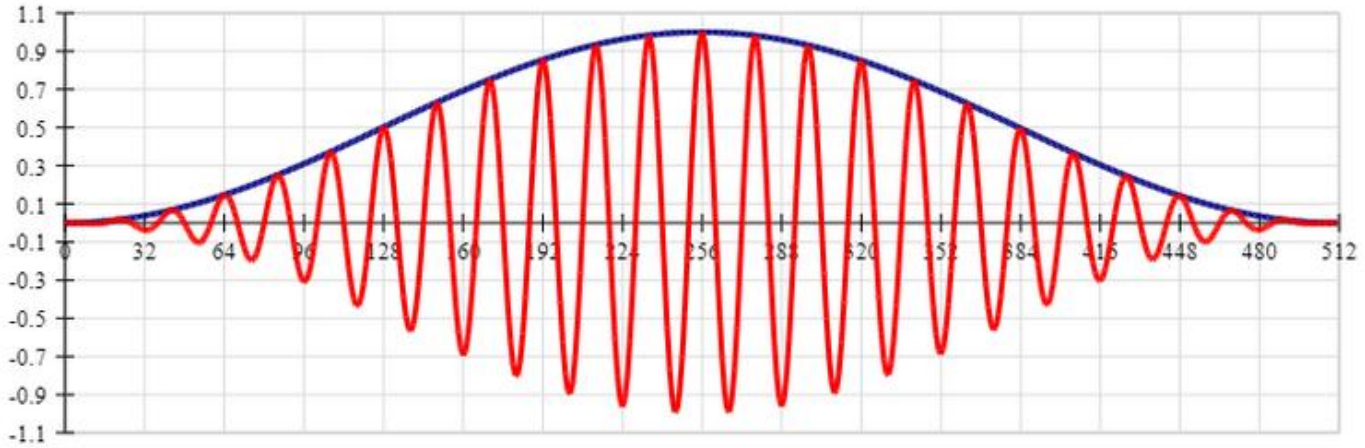
A 3.05Hz Cosine Wave with a Hanning Window applied

- Precieze uitkomst: niet helemaal zichtbaar, maar beter dan zonder window
- Spectral Leakage is opgelost
- De top is wel verbreed. Dit heet: ‘Smearing’



The frequency spectrum of a 3.05Hz Cosine Wave with a Hanning Window applied

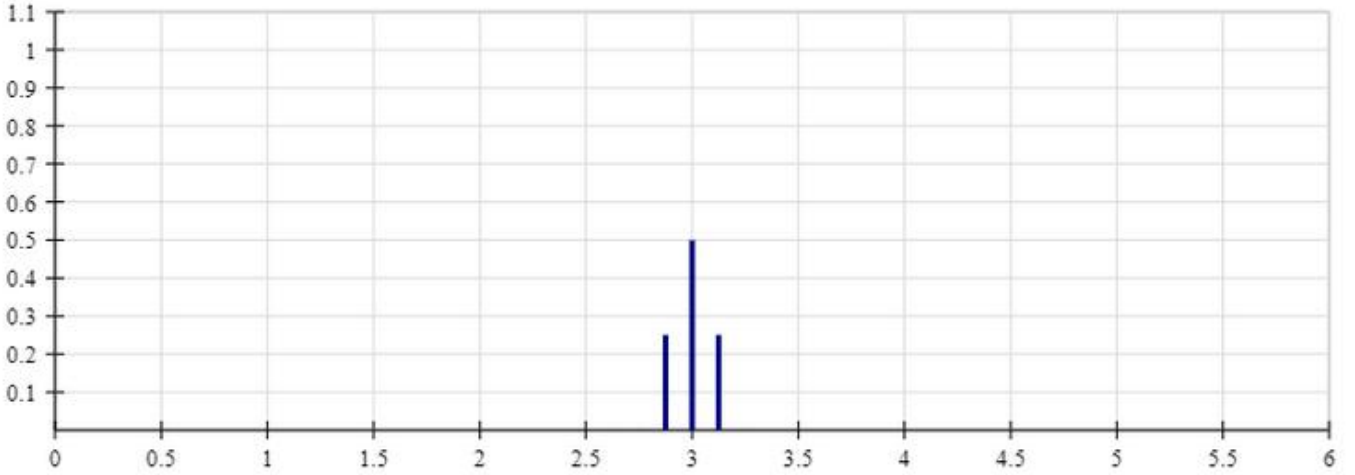
- Terug naar signaal: 3 Hz
- 512 samples
- Begin sluit aan bij eind dus goed voor FFT
- Maar… nog steeds met Hanning window



Windowed 3Hz Cosine Wave

- Redelijk precieze uitkomst: 1 lijn op 3 Hz, plus 2 kleine
- Het window zorgt onnodig voor Smearing in dit geval

"Ieder voordeel hep zun nadeel"



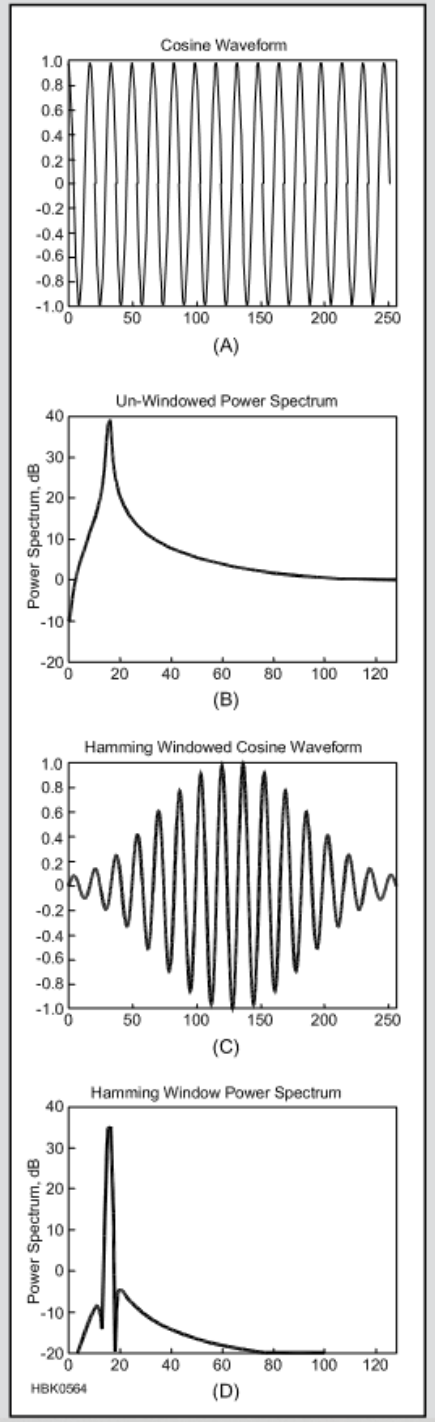
Frequency Spectrum of the Windowed signal

- **Een signaal in de waterval is dus nooit één streep door de manier van rekenen.**

Take away: FFT en foutcorrecties m.b.v. ‘*windows*’

- De FFT berekening heeft in de praktijk dus altijd last van ‘spectral leakage’
- Om dat te compenseren zijn er zgn. ‘windows’ die dit verminderen. Er zijn er verschillende windows met ieder eigen bij-effecten.
- Bekende windows zijn (vaak vernoemd naar de ontwerpers):
  - Blackman
  - Hamming
  - Hanning
  - Bartlett
  - Flattop (= geen correctie)

- Hoe meer samples de FFT gebruikt, des te nauwkeuriger is het frequentiespectrum dat eruit komt (er zijn immers meer bins).



Samplerate

Niet alles met dezelfde samplerate doen

Samplerate, hoe hoog?

- Hoger… → meer bandbreedte mogelijk b.v. compleet 1,8 MHz tot 54 MHz
  → meer data per seconde om te bewerken

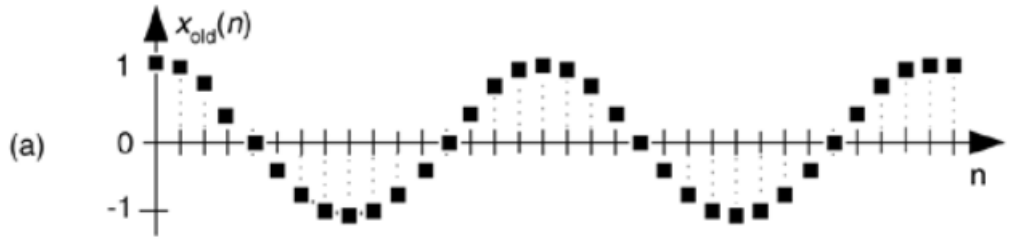
- Aan de kant van het RF (bij de antenne): hoge samplerate *nodig*
- Aan de kant van het AF (bij een luidspreker): lagere *voldoende* samplerate

- In de TRX kan de signaal verwerking van het gewenste signaal (inmiddels niet meer die grote bandbreedte) met een lagere samplerate worden verwerkt.

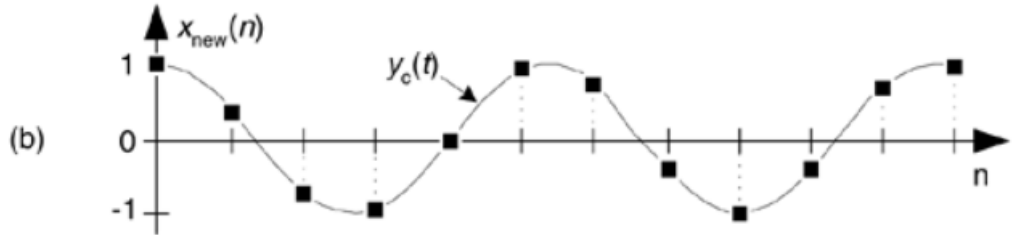
- KAN DAT ZOMAAR? JA (…maar wel onder bepaalde condities)
  - Verlagen van de samplerate in de DSP: **DECIMATE**
  - Verhogen van de samplerate in de DSP: **INTERPOLATE**

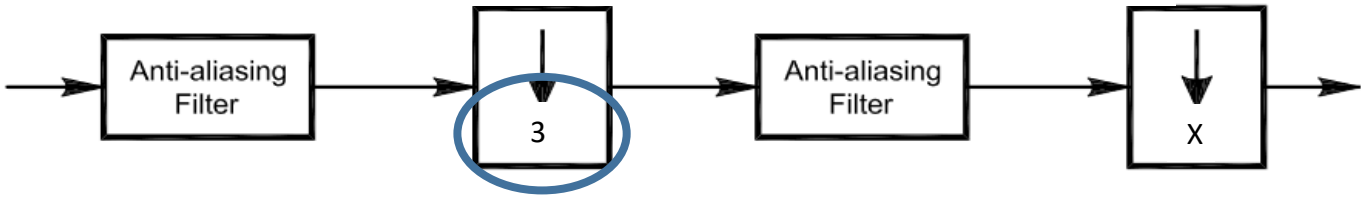
Sample rates: Decimate of decimeren

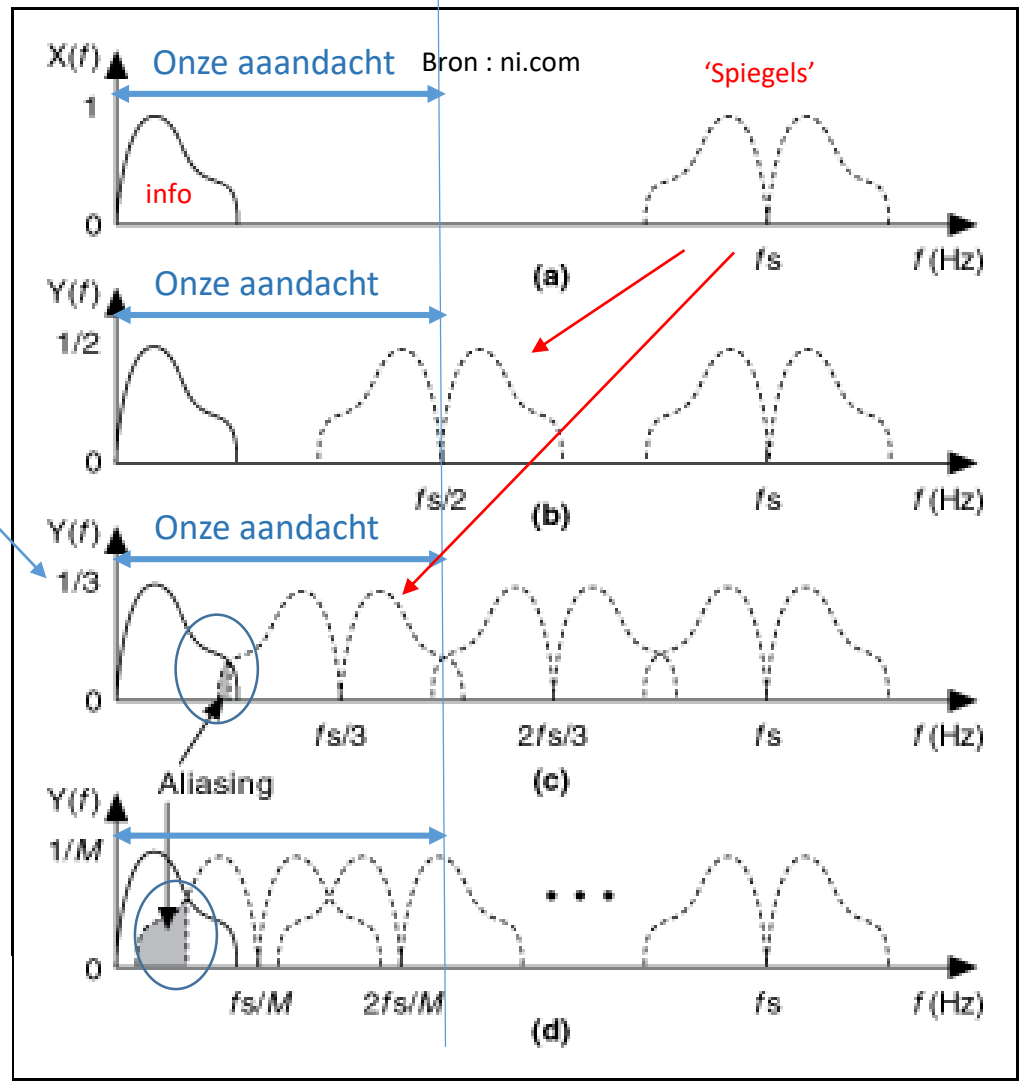
- ECHTER: Er kan aliasing ontstaan
  Dit kan ondervangen worden door een digitaal low pass filter (dus extra rekenwerk) **vooraf** aan de Decimate.



Delen door 3 → 2 op de 3 samples laten vallen

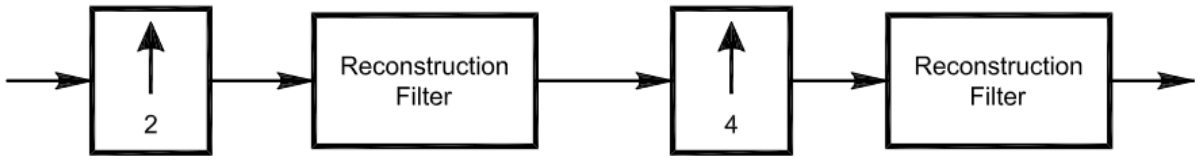


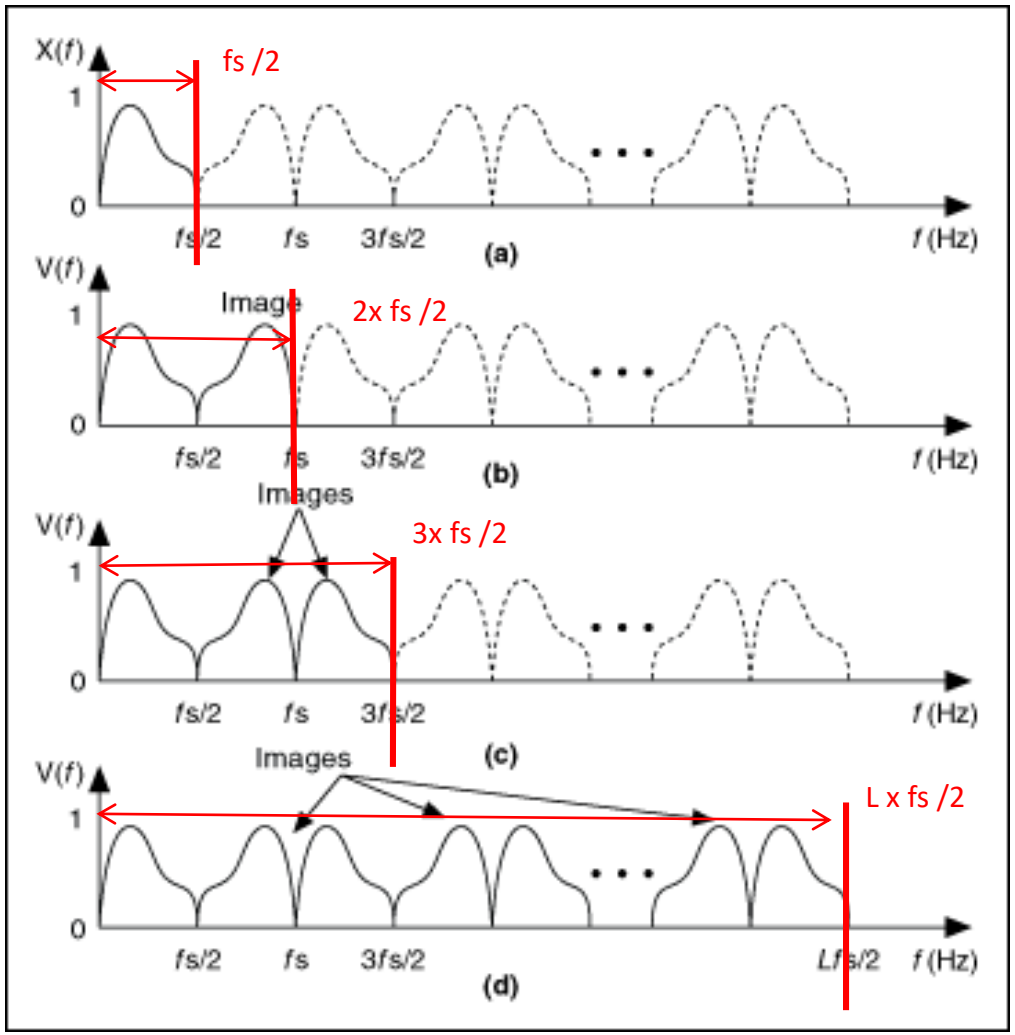




Samplerate omhoog: Interpoleren / **Interpolate**

- Je sampelt het vorige *digitale* spectrum nu sneller
- Effect op het digitale spectrum: Spiegels waar je eerst geen last van had komen zo vanzelf in het baseband spectrum (Nyquist zone 1)
- Geen aliasing, echter wel meerdere keren dezelfde informatie in de baseband.
- Filtering is daarmee sowieso nodig **na** de Interpolatie

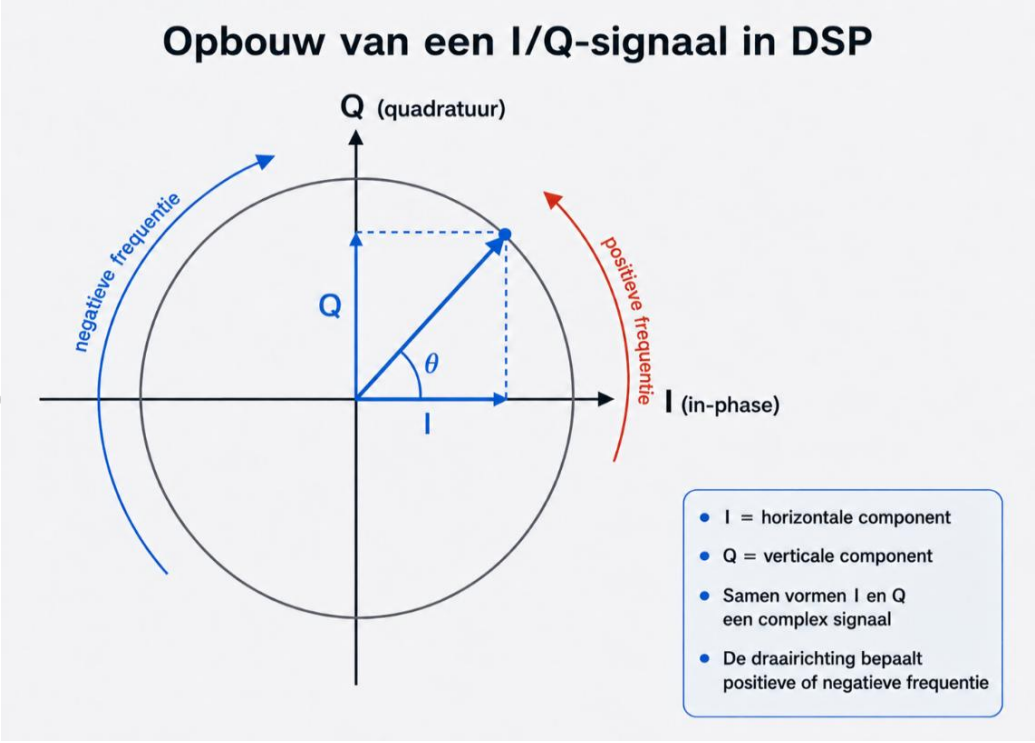




I / Q signalen

I en Q (= In fase en Quadratuur)

- Zit soms op achterkant van een trx
  - **I = In fase Q = Quadratuur fase** (90° verschoven)
  - **I** = COSINUS, **Q** = SINUS
  - Combinatie van de twee: I + jQ
    (j betekent: 90 graden verschoven – 'omhoog')
  - Het Q-signaal wordt met DSP wiskunde gemaakt

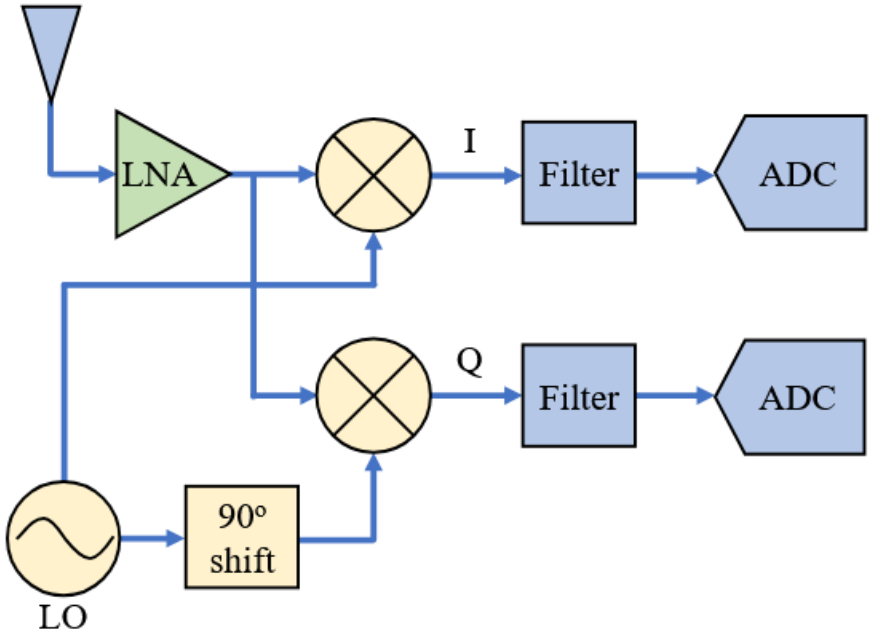


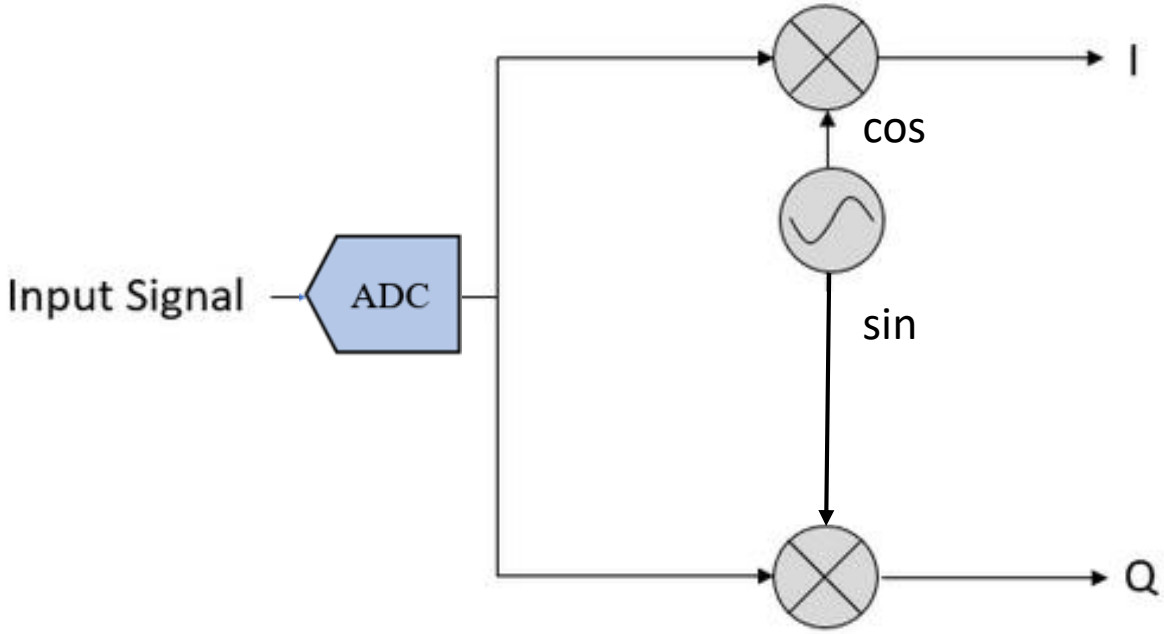
- $S(t) = I(t) + jQ(t) = I\cos(\omega t) + j\,Q\sin(\omega t) = A\,e^{j\omega t}$ (Euler)
  Moeilijk? Ja maar stelt niet mee voor dan draaiende vector met een bepaalde lengte die met tempo ωt rond draait.

- Wiskunde: I/Q maakt zo onderscheid tussen positieve en negatieve frequenties, dus linksom en rechtsom draaiende vector. Hierover later meer als we inzoomen op SDR.

Hoe wordt zo'n complex signaal gemaakt in een SDR

- Met andere woorden (met wat we al weten):
  Hoe kom ik aan dat 'dubbele signaal', onderling 90 graden verschoven



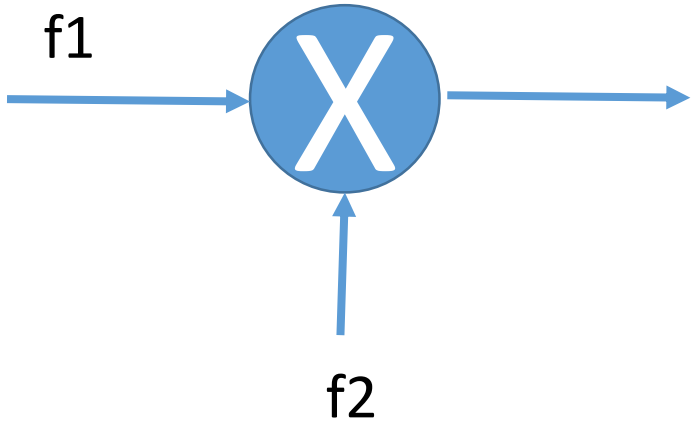


Zowel analoog als digitaal

Bron: pysdr.org

Wat zijn de gevolgen

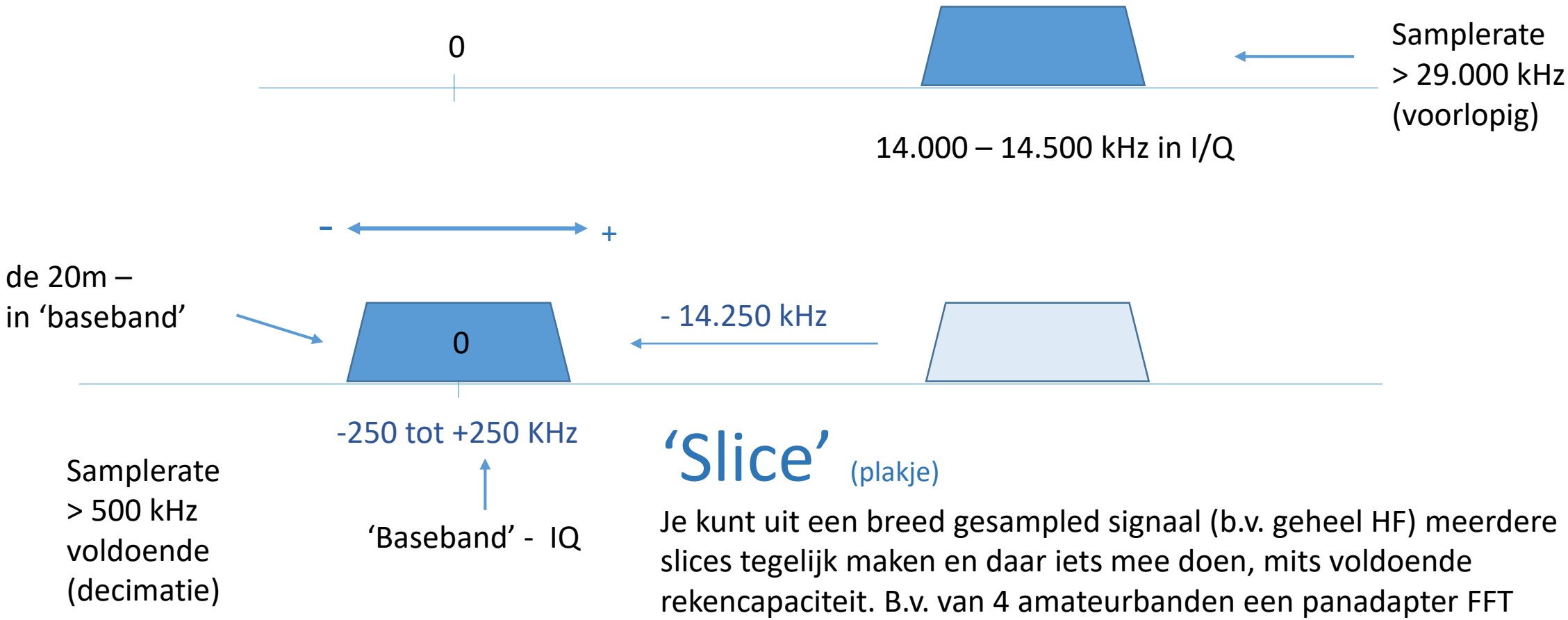
- Je kunt hiermee wiskundig positieve en negatieve frequenties onderscheiden.
- Als je twee positieve frequenties mixt, dan ontstaat de som van die twee
- Als er één negatief is wordt die frequentie afgetrokken van de andere.



f1	f2	output
+ 14250 kHz	- 14249 kHz	1 kHz
+ 1 kHz	+ 14249 kHz	14250 kHz
Audio 0 tot +3 kHz	+ 14250 kHz	14250 USB (14250 -14253)

Mixen is simpel geworden

Digitaal mengen naar baseband (complex signaal) (versimpeld voorbeeld)



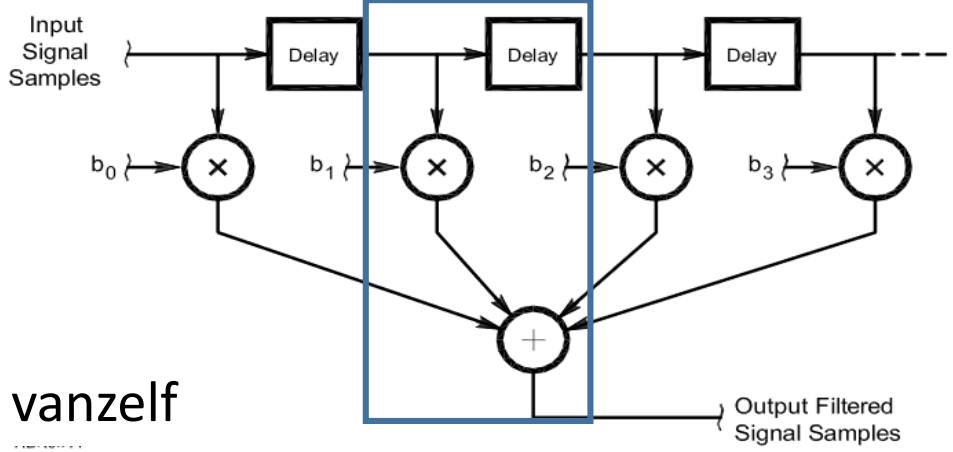
‘Slice’ (plakje)

Je kunt uit een breed gesampled signaal (b.v. geheel HF) meerdere slices tegelijk maken en daar iets mee doen, mits voldoende rekencapaciteit. B.v. van 4 amateurbanden een panadapter FFT scherm tegelijk.

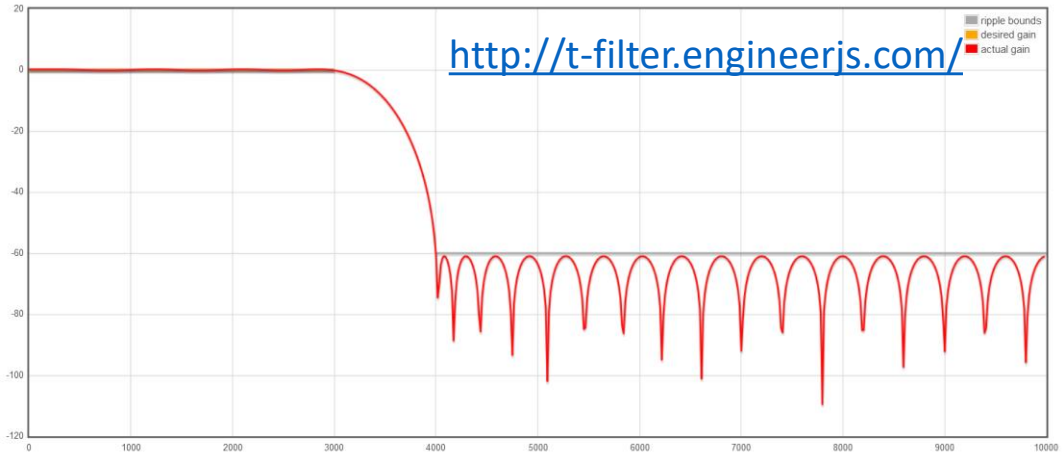
FILTERING

- FIR
- FFT (Fast convolution)
- IIR

FIR – filter (rekenwijze heet 'convolutie')

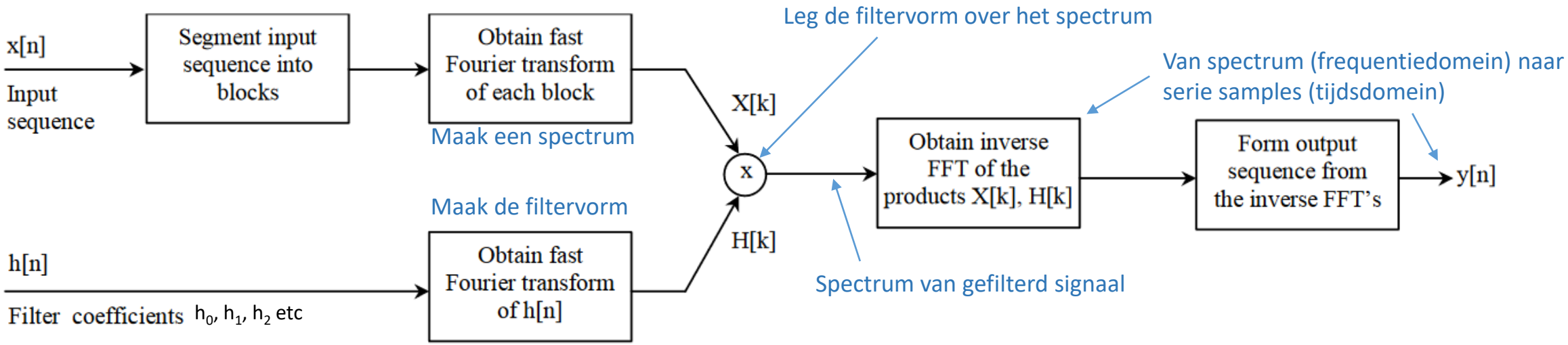


- **FIR**: **F**inite (=EINDIGE) **I**mpuls **R**esponse na nul-input stopt het vanzelf
- Eén delay + vermenigvuldiging + opteller (**MAC** – **M**ultiplicator/ **AC**cumulator) heet een ***'tap'***
- Pet tap een vermenigvuldig factor: b_0, b_1, b_2, …etc.
- Deze rekenwijze (schuif het signaal langs de verschillende MAC's en tel op) heet **'Convolutie'**
- **Praktisch: Hoe stijler het filter t.o.v. de samplerate, des te meer taps**
  De eigenlijke oneindige reeks wordt altijd ingekort ('truncated') tot 'eindige'. Hiervoor wordt weer een 'window' gebruikt, net als bij de FFT.
- Voorbeeld:
  F_{sample} = 20 kHz, LPF 0 - 3kHz, 0.5 dB rimpel,
  stopbandverzwakking = -60dB
  flank: 1000 Hz van 0 naar -60dB: # taps: 49
  flank 300 Hz van 0 naar -60dB: # taps = 157



Filtering m.b.v. FFT en IFFT (zgn. 'Fast Convolution')

- Als we het analoge signaal omzetten naar het frequentie domein (FFT), en vermenigvuldigen met de filterkarakteristiek van het filter ontstaat het gefilterde signaal
- Dat kan worden terug gezet naar het tijdsdomein (IFFT = inverse of omgekeerde FFT)
- Kun je zo een ideaal filter maken met rechte zijden? **Nee!**
  Er is nog steeds de beperking van de niet oneindige reeksen in de berekeningen.
- Het blijkt in de praktijk (bij veel taps) wel sneller dan de standaard FIR Convolutie berekening



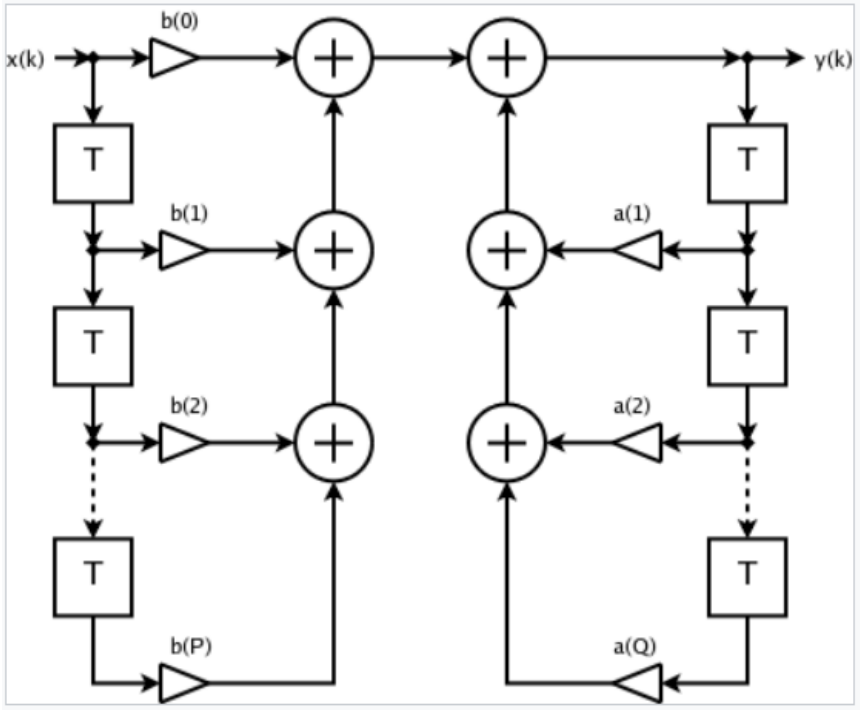
IIR – filter

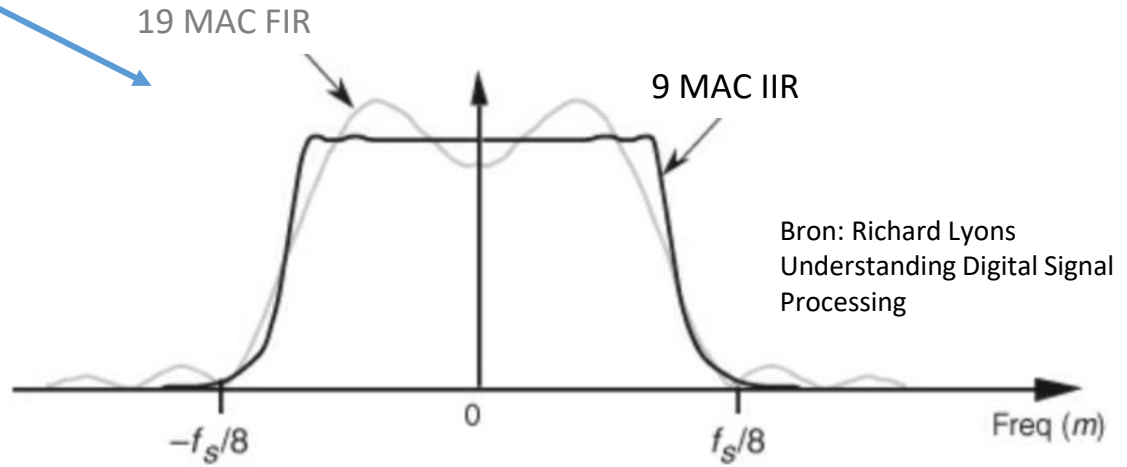
(Infinite (=ONEINDIG) Impulse Response)

- Ingewikkelder ontwerpproces dan FIR
- Uitgebreid rekenwerk gaat hieraan vooraf
  Dit ligt ver buiten het bestek van de lezing.

- Je kunt met weinig MAC's een scherp filter maken
- Nadeel: Kans op instabiliteit, immers de output wordt deels als input gebruikt en kan oscillatie ontstaan.

Bron: Wikipedia



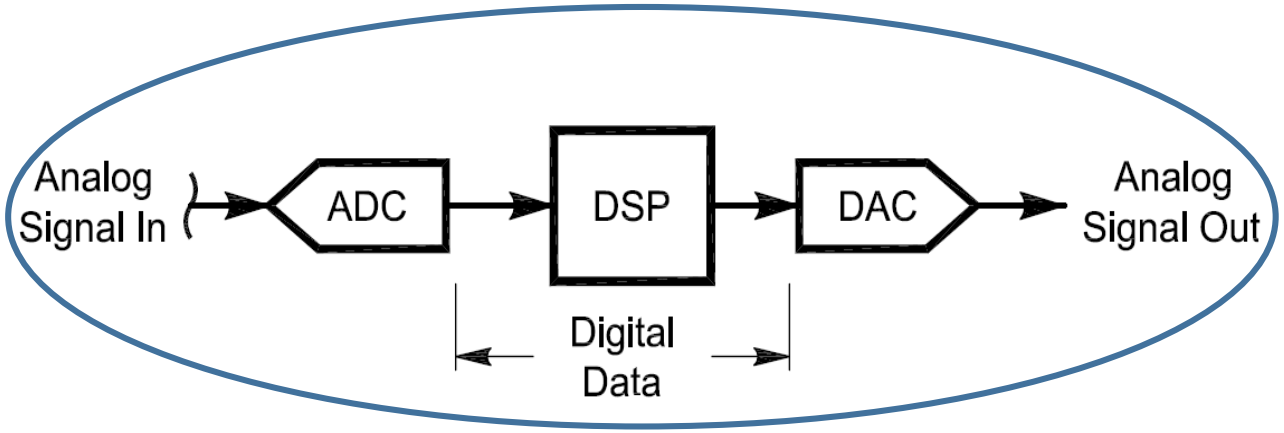


Bron: Richard Lyons
Understanding Digital Signal Processing

3. De TRX, wat er in een SD-RADIO gebeurt

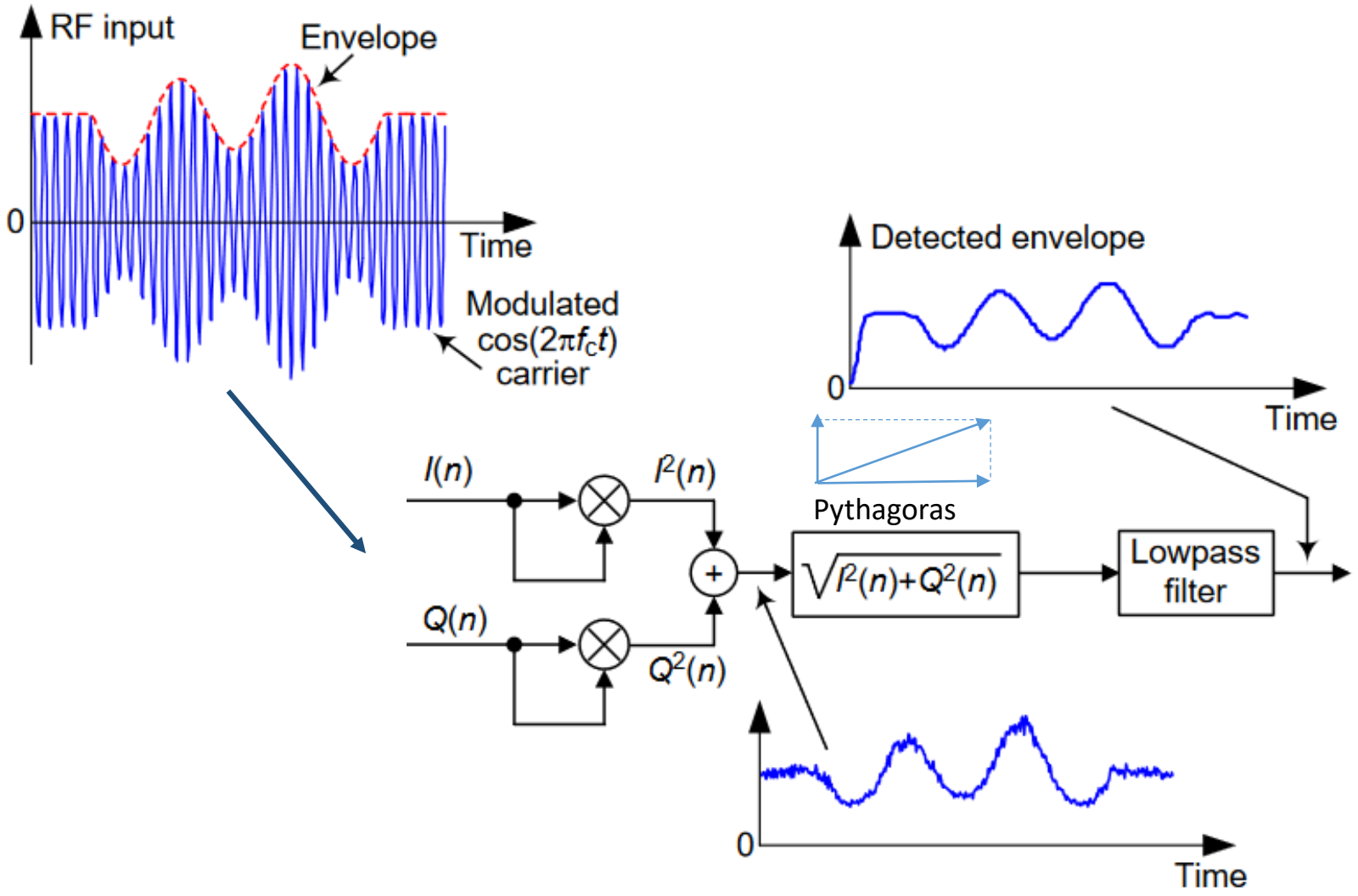
- B.v. detectie, voorbeeld AM

- Verschillende TRX:
  - RTL SDR stick
  - Yaesu FTDX-10
  - Yaesu FT-710
  - ICOM IC-7300
  - Flexradio Flex-6700 – 144 MHz



DSP AM detector

Pythagoras did it again! ☺



Onze praktijk, verschillende (T)RX’s

- RTL-SDR
- FTDX-10
- FT-710
- IC-7300
- Flex-6700 (2m)



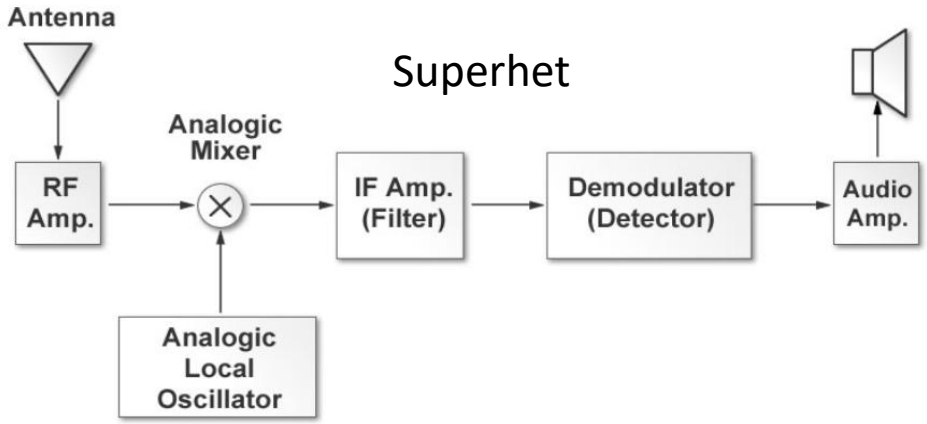








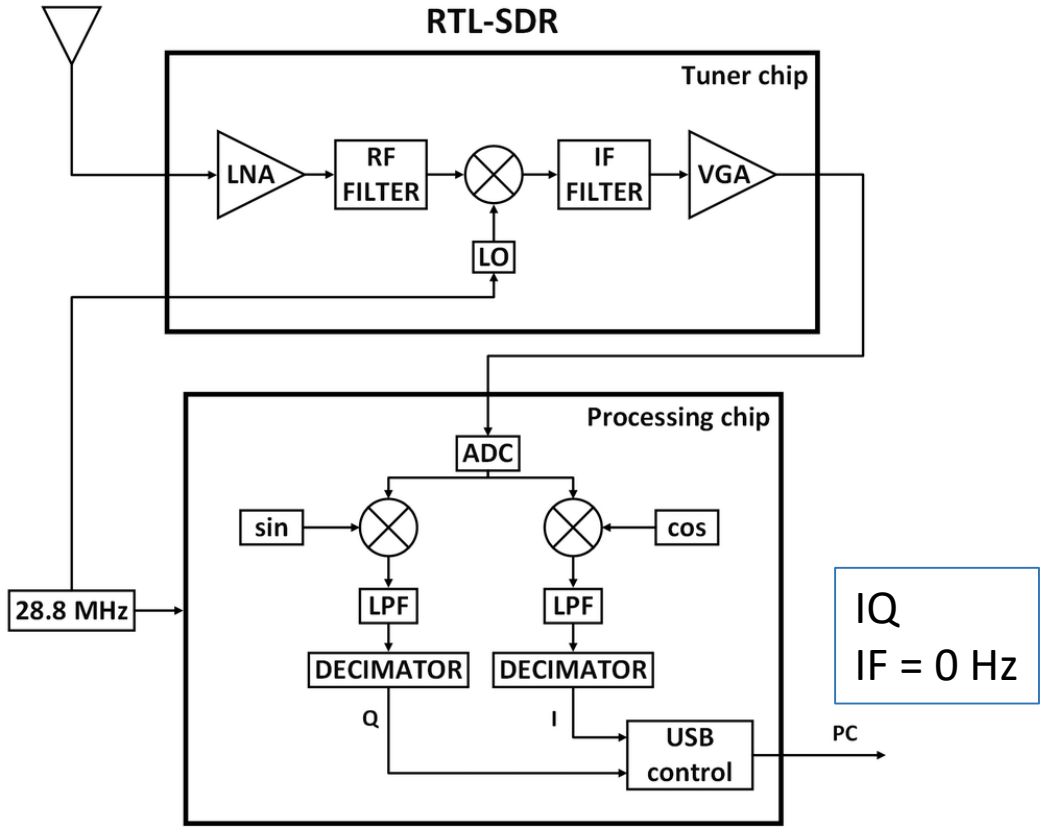
RTL-SDR en de Superhet



1e stap RTL-SDR is analoog á la een superhet.

LNA = Low Noise Amplifier
VGA = Variable Gain Amplifier

Wie doen dit b.v. ook?
Elecraft K3 (IF = 8.2MHz + 15 kHz)
Yaesu FTDX-101 (IF = 9 MHz)
Yaesu FTDX-10 (IF = 9 MHz)
Icom IC-7100 (IF = 124MHz + 455kHz + 36 kHz)
De ADC komt na de IF

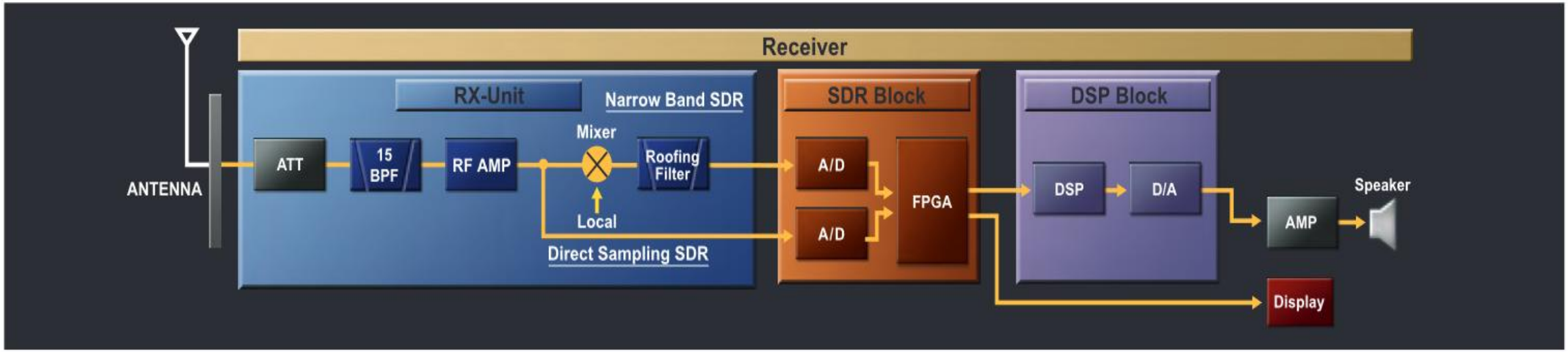


Bij de RTL-SDR wordt het digitale baseband signaal doorgezet in IQ vorm naar de pc.

Yaesu FTDX-10 (hybrid: heterodyne + SDR/DSP)

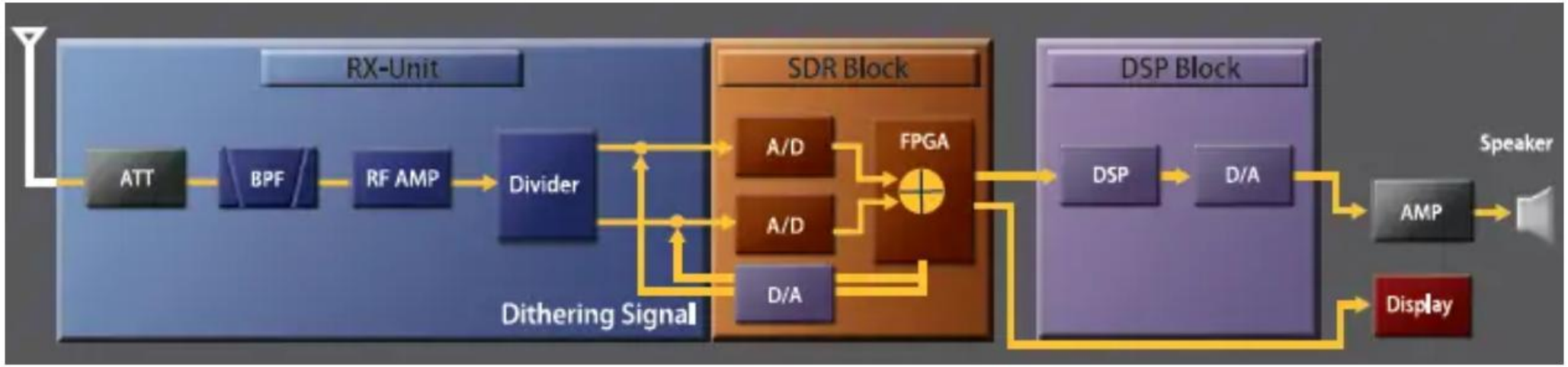
- Superhet met deel SDR, vanaf IF (= 9MHz) → dit wordt tegenwoordig **'IF Direct Sampling'** genoemd. IF filters: 3 kHz, 500 Hz, 300 Hz
- 2x 14-bit ADC
- Het tweede ADC-pad zorgt o.a. voor het spectrum display (FFT) en is breed
- De FPGA bevat het algoritme dat het te bewerken / te decoderen signaal naar baseband brengt in IQ-format met tegelijk een (veel) lagere samplerate

FPGA: Field Programmable Gate Array



Yaesu FT-710 ontvanger (‘FT-7300’)

- 2 x 14 bit ADC’s in de Yaesu’s (zowel FTDX-101 als FTDX-10 als FT-710 heeft deze)
- Wat gebeurt er:
  - De Divider (= signaal splitter) voedt een TI ADS62P45, dit is een dual-channel, 14-bit ADC
  - De 2 ADC’s samplen zelf in I/Q (FT-710 notes)
  - De FPGA levert signalen voor:
    - De DSP in I/Q format in baseband met lagere sample-rate (IF = 24 kHz)
    - FFT bandscope
    - Digital Dithersignaal voor de ADC’s staan altijd aan



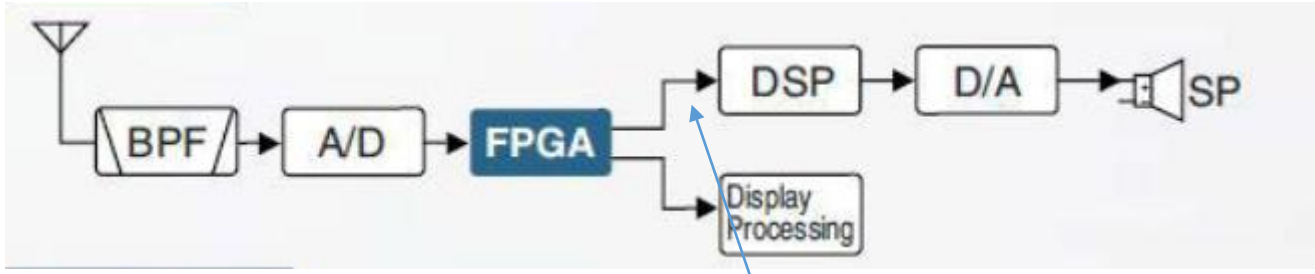
Yaesu FT-710 receiver block diagram

Zie: https://www.qsl.net/ab4oj/test/docs/ft710notes.pdf

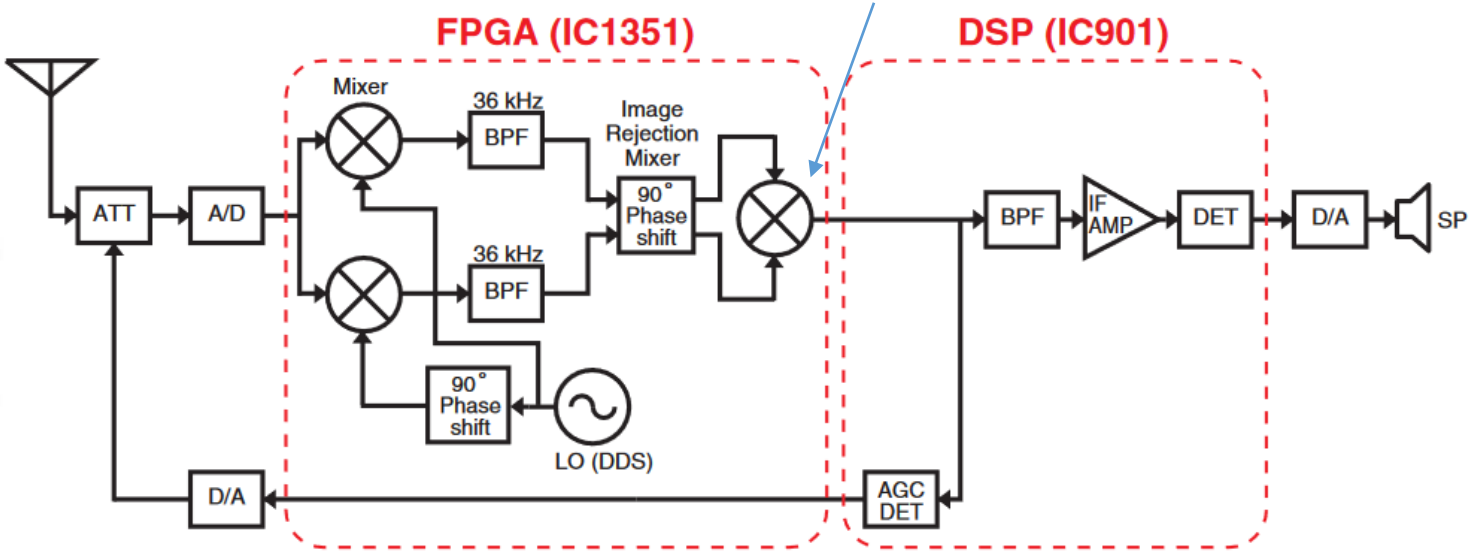
Icom IC-7300 (de game changer naar SDR)

- ADC: 14 bit,
  F_{sample}= 124 MS/s
- IP+ = Dithering aan

- FPGA → DSP:
  36 kHz 12kHz breed
  (zelfde als de ic-7100)



36 kHz – breedte 12 kHz
(ook IC-7100 !)





Take away

- Na de samping wordt:
  - Deel van het spectrum geselecteerd (“*Slice*”) en naar baseband gemengd
    Daarna gedecimeerd (lagere sample rate).
    Alles in het blokje FPGA zodat het verder verwerkt kan worden.
    Bij voldoende rekenkracht meerdere slices
- Signaal bewerkingen vinden plaats in IQ-vorm
  - Signaal via USB-poort naar computer SDR ook in IQ vorm
  - Met IQ kun je onderscheid maken tussen positieve en negatieve frequentie
- Filtering, detectie en b.v. NR gebeuren op lagere samplerates
- De panadapter berekening in de SDR volgt vaak via een aparte weg (hogere samplerate)
  - Grotere bandbreedte en simpele berekening m.b.v. FFT
- Alias is niet altijd verkeerd, als je de zgn. Nyquist zones gebruikt
  - De sample rate van de ADC bepaalt *niet de hoogste frequentie* die gesampled kan worden *maar de grootste bandbreedte*

PA3A set-up voor panadapter

IF output 8.2 MHZ



Softrock
IF Lite
f = 8.225 MHz
IQ mixer + LPF

Stereo (I en Q)
Audiokaart
192 kHz samplerate

NAP3
SDR

Vragen ?

Opmerkingen?



Bronnen:

- ARRL Handbook (2015)
- DSP Basics, Dr Chris Bore
- Sciencedirect.com
- Atx7006.com
- Wikipedia.org
- Ni.com
- Addictedtoaudio.com.au
- Binnenlandsbestuur.nl
- sharetechnote.com
- lntwww.de
- digitalsoundandmusic.com
- Windowing & the FFT - A Guide for the Perplexed, Mark Newman
- http://t-filter.engineerjs.com/
- https://pysdr.org
- https://brianmcfee.net/dstbook-site/content/ch02-sampling/Sampling.html
- Understanding the 'Phasing Method' of SingleSideband Demodulation, Richard Lyons
- Quadrature Signals: Complex, But Not Complicated, Richard Lyons
- Design Of Digital Down Converter And Signal Detection Techniques For Software Defined Radio, Venugopal and Nemo
- A Beginner's Guide To Cascaded Integrator-Comb (CIC) Filters, Richard Lyons
- Understanding Digital Signal Processing, Richard Lyons
- IC-7610 User Evaluation &Test Report, Adam Farson VA7OJ/AB4OJ
- Yaesu FT-710 User Evaluation & Test Report, Adam Farson VA7OJ/AB4OJ
- Researchgate.net
- Yaesu.com
- Allaboutcircuits.com
- UNIVERSITY OF NEWCASTLE UPON TYNE – DSP
- Service manual IC-7300
- https://www.dutchaudioclassics.nl/philips-oversampling-system-for-compact-disc-decoding/Analog.com
- Analog.com