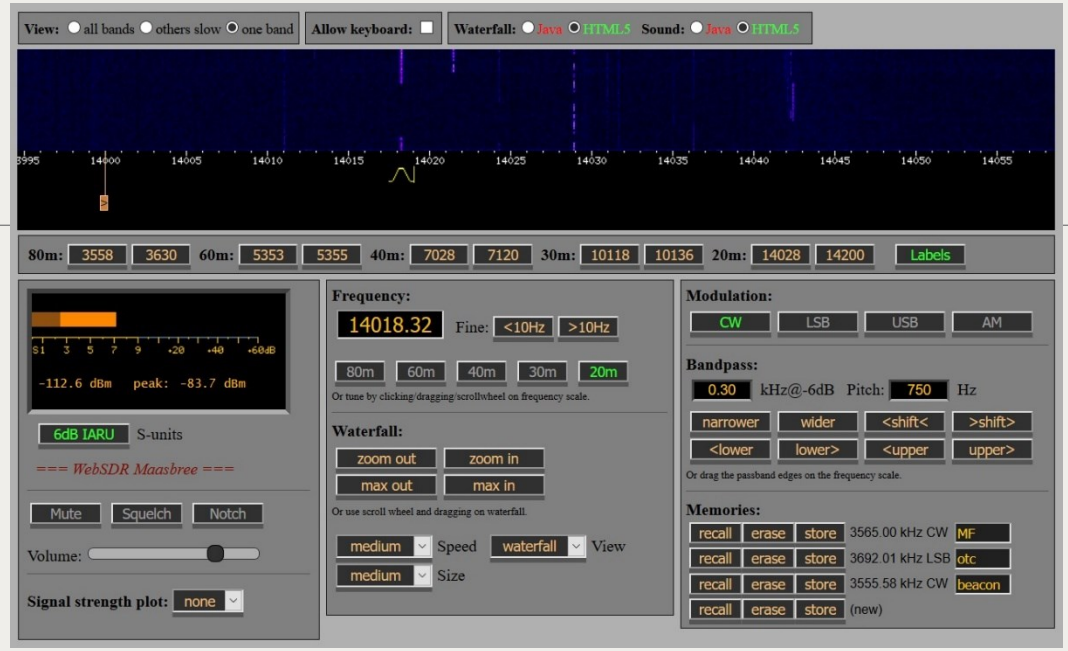
WebSDR Maasbree

REALISATIE VAN EEN WEBSDR IN DE REGIO VENLO.

LOEK PE0MJX / JAN PA0SIM



Doel

Radioamateurs **24/7** toegang te geven tot een **remote ontvanger** op een **rustige locatie** om hiermee **verbindingen** te kunnen maken.

=> zenden blijft vanaf de eigen locatie. Zenden wordt niet gehinderd door een hoog ruisniveau of door QRM van bv. zonnepanelen.

=> niveau moet in balans zijn met de zendmogelijkheden in een woonwijk.

Waarom een WebSDR

Twee doorslaggevende eigenschappen:

- latency <0.5sec = noodzakelijk voor het maken van QSO's
- voor een groot aantal gebruikers = optimaal gebruik van de locatie **

➔ Alleen de WebSDR van PA3FWM ondersteunt deze combinatie!

** Maasbree getest tot 200 gebruikers, maar kan gemakkelijk meer aan

Hoe begin je

Je vraagt de software aan PA3FWM met een uitleg over het doel.

De rest moet je zelf invullen zoals:

- locatie
- antenne en signaaldistributie
- keuze frequentiebanden
- Ontvangers (SDR, taal C, DSP)
- Computer (Linux)
- user interface aanpassingen (html, css, javascript)
- beveiliging: elektrisch en internet

Locatie keuze

Locatie is doorslaggevend:

- Ruisniveau's **relatief** gemeten op uiteenlopende locaties
- met RSP1a ontvanger, SDRuno* en kleine actieve loop antenne
- Identieke opstelling // op 2 locaties** waar PA0KDF gemeten heeft
- Meetresultaten van PA0KDF als referentie voor een absoluut niveau

* De combinatie met SDRuno heeft een gekalibreerde RMS S-meter

** Maasbree PH4RTM en Kronenberg PA3ARM

Antenne keuze en plaatsing

- Stralingsdiagram moet omnidirectioneel zijn en NVIS omvatten **
- Plaatsing antenne zover mogelijk van objecten en gebouwen. Niet alleen voor minder ruis. Objecten schermen af en kunnen het signaalniveau verlagen en op die manier de SNR

➔ Keuze voor een kleine actieve breedband loop met nulling mogelijkheid voor een evt. aanwezige stoorbron, plaatsing op 70m afstand van gebouwen

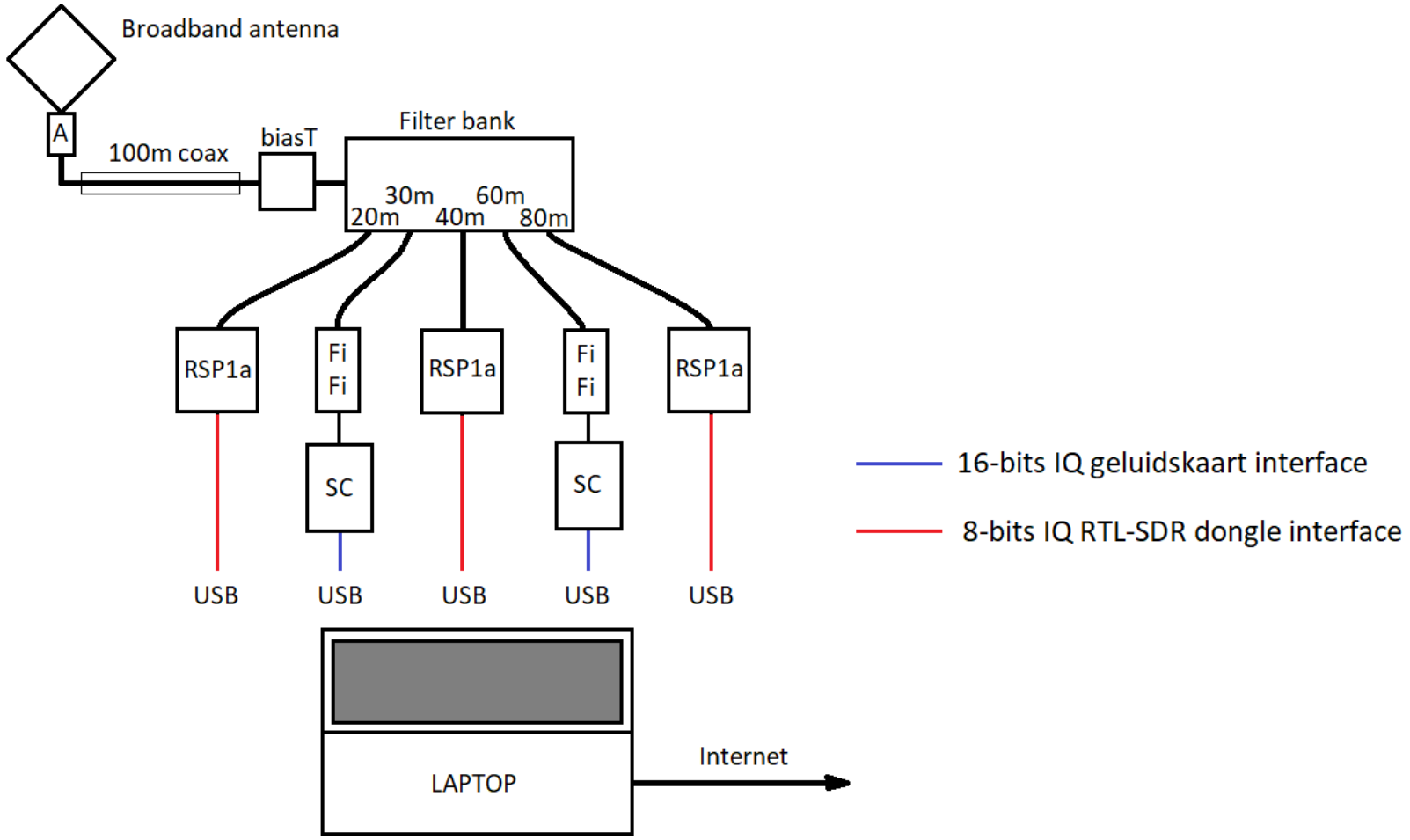
** Grondgolf propagatie (verticale polarisatie) speelt op die banden geen rol

Benodigde dynamisch bereik RX

Wordt bepaald door:

- Ruisniveau op de locatie (frequentieband afhankelijk)
- Omnidirectioneel stralingspatroon antenne (ontvangt alle ruis, directivity = hogere SNR)
- Verwachte signaalsterkte: filtering amateurbanden (géén zeer sterke omroepstations)
- Vrije opstelling van de antenne (sterkere signalen, de omgeving afschermt niet af)

Blokschema WebSDR



IQ data interfaces naar de WebSDR

- 16-bits geluidskaart interface: f_s = max. 192kHz**
- 8-bits RTL-SDR dongle interface: f_s = 256kHz, 512kHz, 768kHz, 1024kHz, 1536kHz, 2048kHz of 2880kHz

** IQ data: WebSDR bandbreedte is gelijk aan de sample frequentie f_s.

➔ 8-bits RTL-SDR dongle interface voor de bredere banden (80m, 40m en 20m)

Een goede uitleg over IQ data:

https://www.pe0sat.vgnet.nl/sdr/iq-data-explained/

FiFi SDR / geluidskaart

- Geluidskaarten vertonen ruistoename en versterkingsafname voor frequenties >65kHz
  - dynamisch bereik niet over hele 192kHz band te oogsten
  - effectieve bandbreedte ~130kHz: **1e beperking**
- Noodzaak voor IQ-balancing
  - WebSDR tool maakt >70dB haalbaar (=> geen echte beperking)

➔ Deze route is gekozen voor de smalle 60m en 30m band.

RSP1a SDRplay ontvanger (1)

Extra software voor de RSP1a ontvangers:

- SDRplay API
- F4FHH's versie van de RTL TCP server (rsp_tcp.c)**

<==> 14-bits API <==> rsp_tcp.c ==> 8-bits WebSDR

** aangepast door ON5HB en door ons verder gemodificeerd.

RSP1a SDRplay ontvanger (2)

Dynamisch bereik** bij N=8-bits, 512kHz sample frequentie en 2800Hz audio bandbreedte:

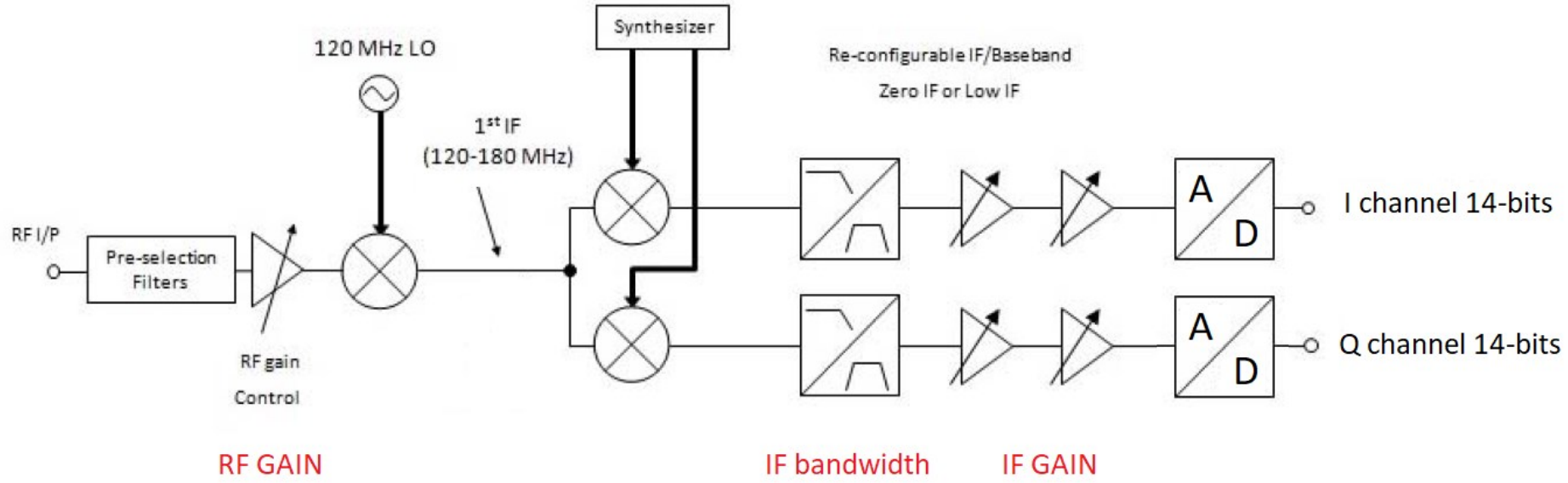
1. SQNR: 6.02N+1.7609 = 49.9dB
2. Processing gain: 10× log10(512kHz/2800Hz) = 22.6dB

Totaal: 72.5dB => **8-bits is 2^{e} beperking**

** 8-bits kwantisatieruis en ADC clipping niveau

(ENOB RSP1a = 10.5)

RSP1a blokschema



In te stellen: RF gain (LNA), IF bandbreedte en IF gain

Instelprocedure RSP1a

Strategie: vinden van de laagste gain die nodig is om net** de bandruis te kunnen ontvangen.

1. Selectie voor de 8 MSBits van de 14 bits
2. Instellen RF gain met een tijdelijk hoger dan nodige IF gain
3. Instellen IF gain met de gevonden RF gain

Overtollige gain gaat ten koste van groot signaal gedrag (ADC clipping)

** “net” is geen harde grens, werkt alleen bij gelijk ruisniveau over de band.

GNUradio voor meer bits (1)

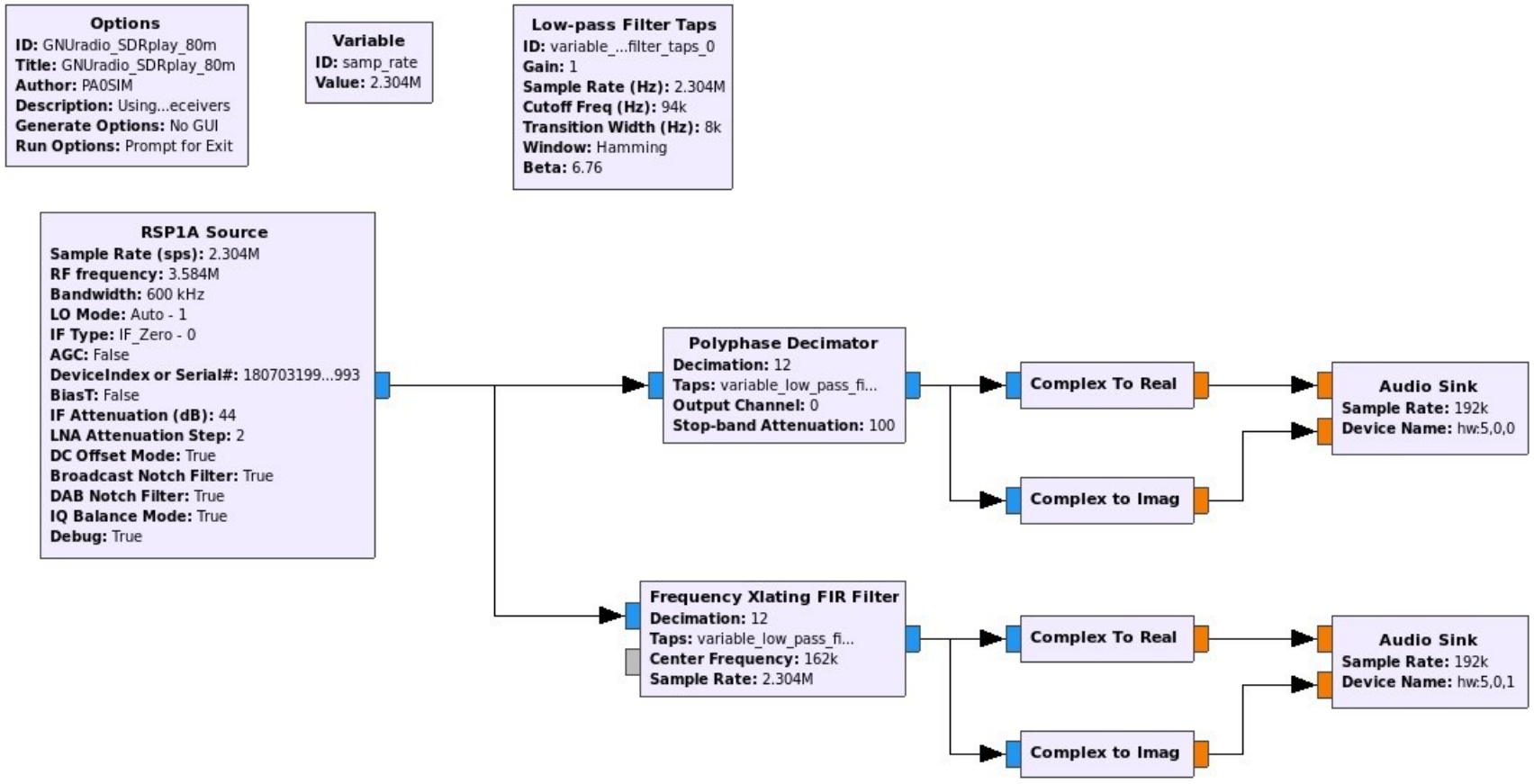
Maakt voor de RSP1a 16-bits mogelijk via geluidskaart interface.

Twee problemen:

1. we kregen het niet stabiel werkend
2. de geluidskaart interface buffering is niet beheersbaar waardoor de latency tot secondes kan oplopen ("2-clock-problem“)

Heeft veel potenties, bv. voor interfacing naar andere SDR ontvangers en voor implementatie van bv. een luxe noise blanker.

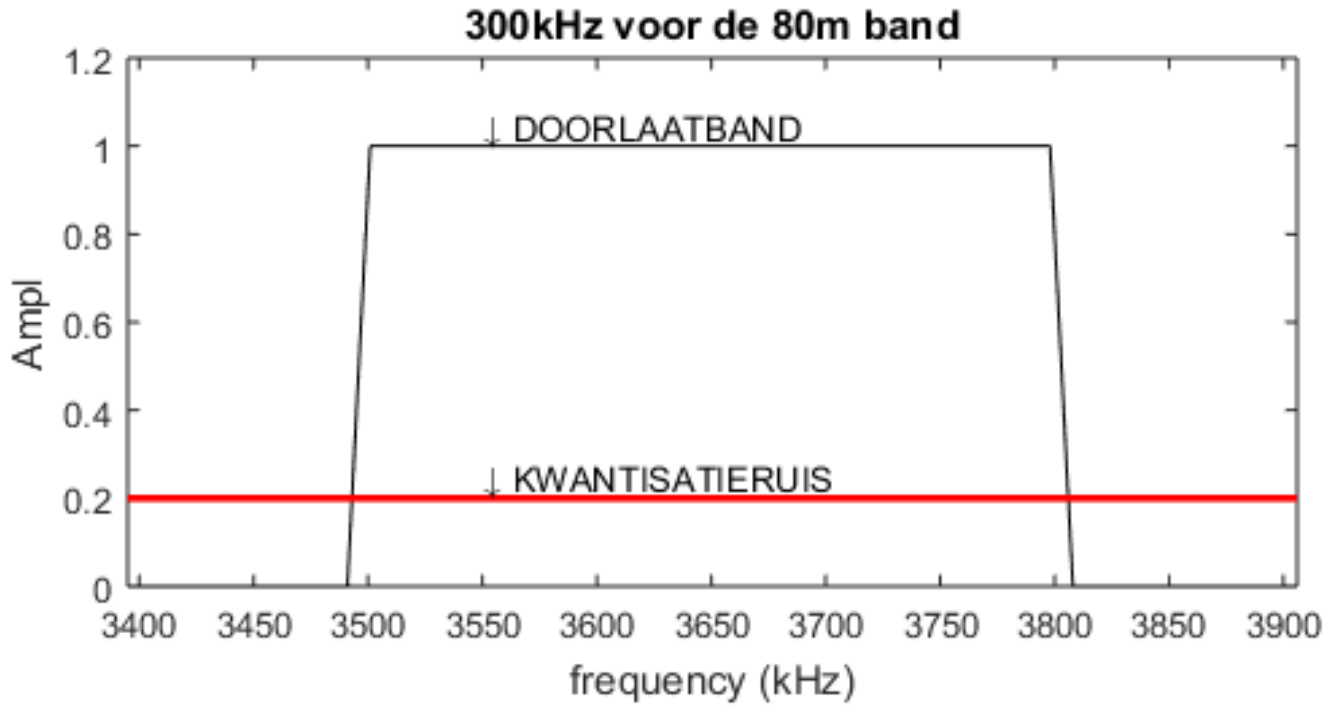
GNUradio (2) 80m blokschema



Noise shaping (1)

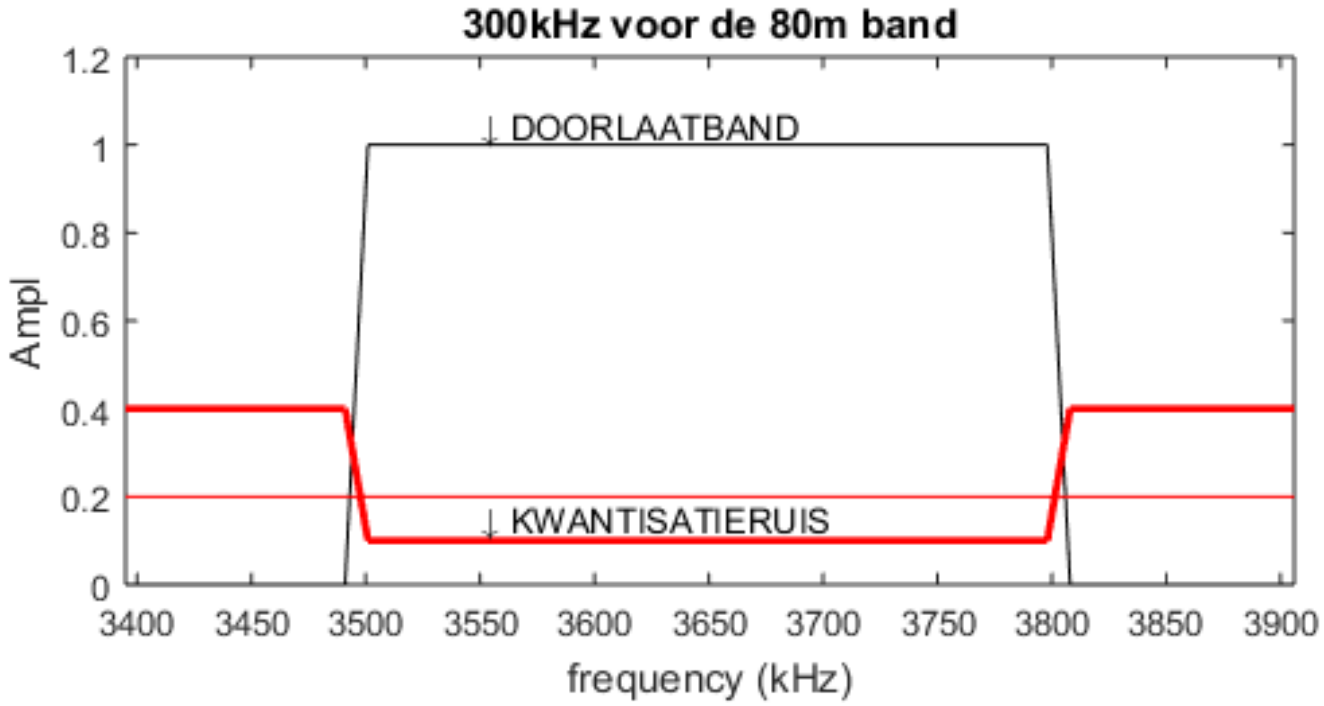
Kunnen we meer doen om het kwantisatieruis-niveau te verlagen?

<u>Gegeven:</u> we hebben op 80m maar 300kHz nodig van de 512kHz



Noise shaping (2)

Kwantisatieruis-niveau in de band kan met noise shaping verlaagd worden ten koste van het niveau aan de randen er buiten.



Noise shaping (3)

Met een eenvoudige noise shaping winnen we nu 4dB.

=> totaal dynamisch bereik** incl. noise shaping:

72.5dB + 4dB = 76.5dB

** 8-bits kwantisatieruis en ADC clipping niveau

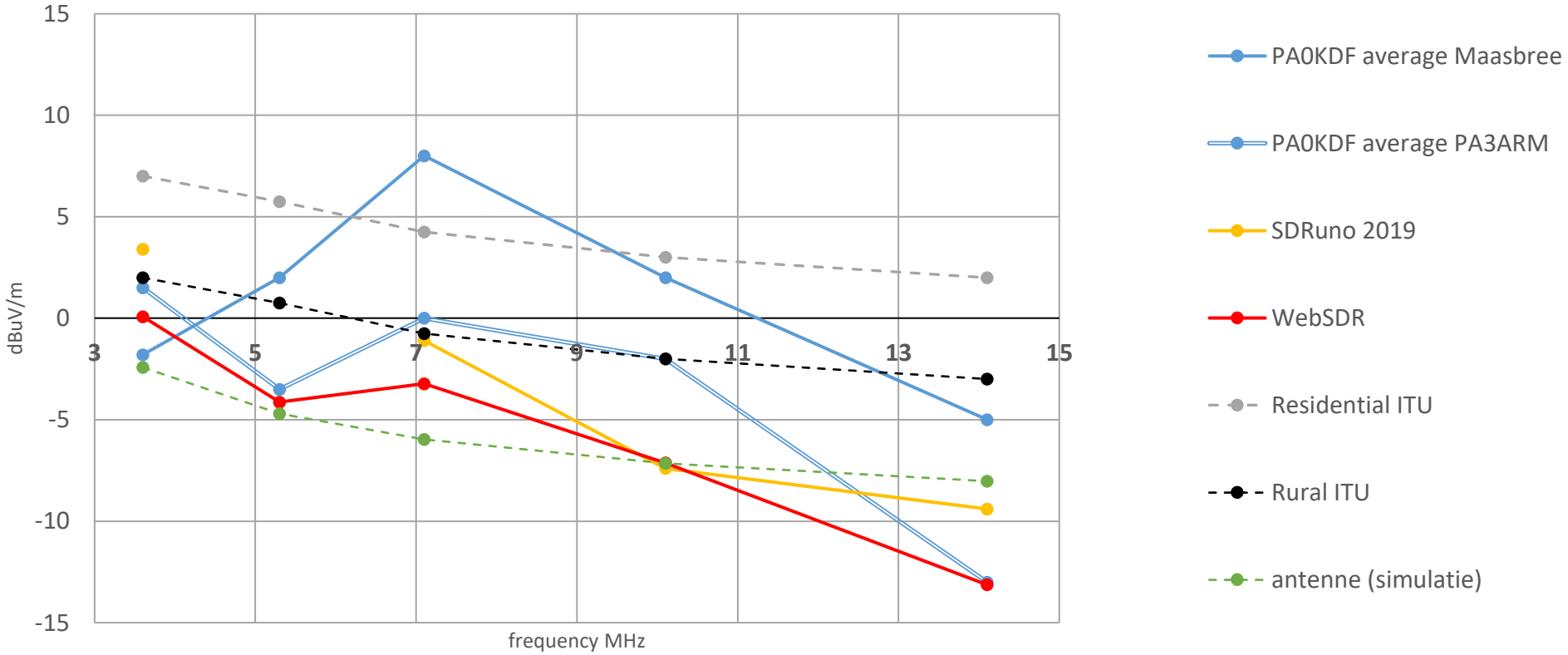
ADC clipping

De RSP1a meldt “overloading” via de API.

- Fallback WebSDR: na een melding wordt de gain tijdelijk met 6dB verlaagd tot signaalniveau’s weer onder een bepaald niveau zijn
- Meldingen en gain settings worden gelogd

Clipping oorzaken: sterke QRN, in-band OTHR, af en toe bij contesten

Ruisniveau's



NB: meetlokatie WebSDR verder van gebouwen dan PA0KDF

Kanttekeningen eigen metingen

- Antenne is niet gekalibreerd (via simulatie verkregen waarde wordt gebruikt voor de AF)
- Rustigste plek op de band op rustigste moment van de dag
- S-meter is afgelezen en is geen RMS (wel gecorrigeerd met -5dB)
- Geen average waarde (eerder minimum)
- Ruisniveau erg dicht bij ruisniveau ontvanger
- Ruisbandbreedte WebSDR filter niet bekend
- 20m: 2dB te lage gain correctie voor verzwakking in filter
- Coaxkabel verliezen zijn niet gemeten

Conclusie

De twee beperkingen resulteren in een compromis:

- geluidskaart interface alleen voor smalle banden bruikbaar
- 8-bits interface vraagt om een kritische instelling, de resulterende instelling is een compromis tussen gevoeligheid en ADC clipping.

Met deze beperkingen kunnen we ons doel waarmaken:

We horen in Maasbree beter dan we thuis kunnen zenden!

Pauze

Na de pauze tijd voor vragen.

Indien interesse enkele sheets over Noise Shaping principe.

Links NPR meting RSP2 ontvanger:

http://www.dc4ku.darc.de/SDRplay_RSP2.pdf

https://www.rfseminar.nl/cms/wp-content/uploads/2019/03/electron0617_226_231.pdf

Noise Shaping principe

Hoe kunnen we het spectrum van de kwantisatieruis veranderen?

Kwantisatieruis wordt ruis genoemd, maar:

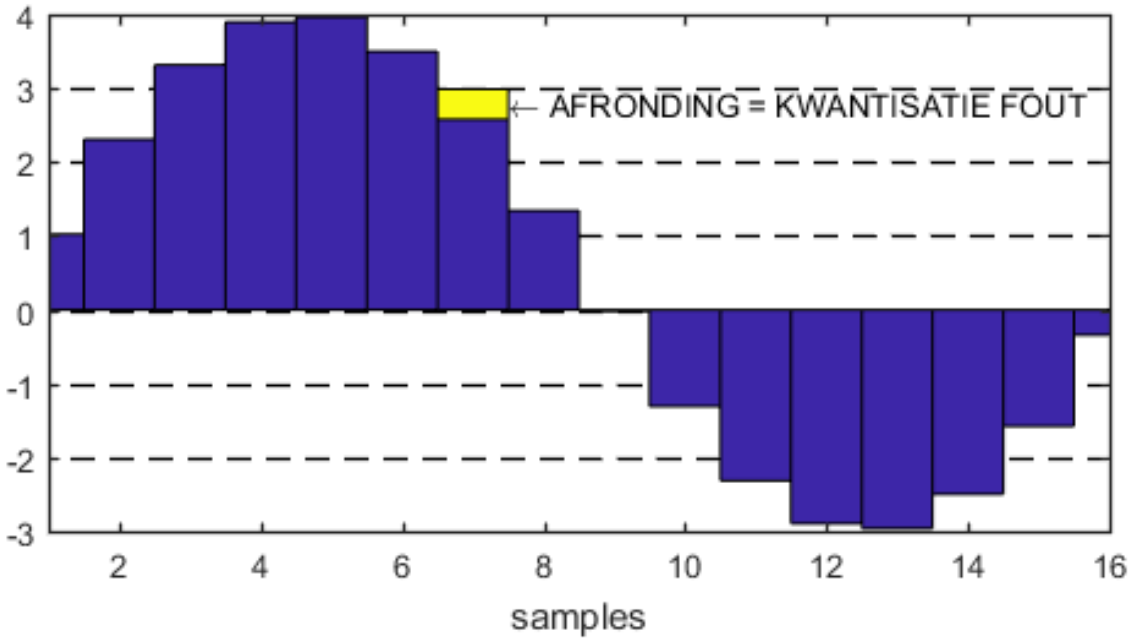
- is niet willekeurig, maar wordt bepaald door het signaal
- de afrondingsfouten zijn bekend

Kwantisatieruis

RSP1a levert 14-bits samples ➔ 8-bits samples WebSDR

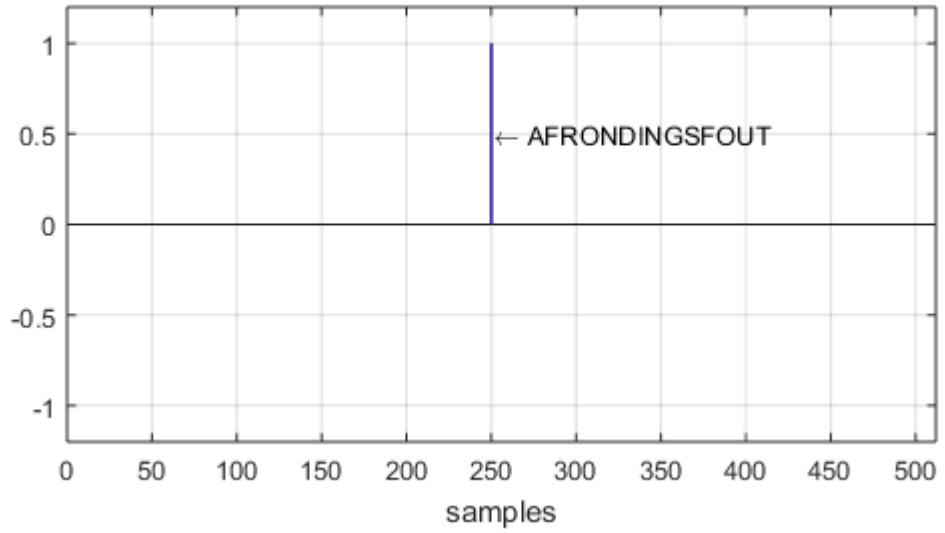
$S_{8bits} = S_{14bits} + fout$

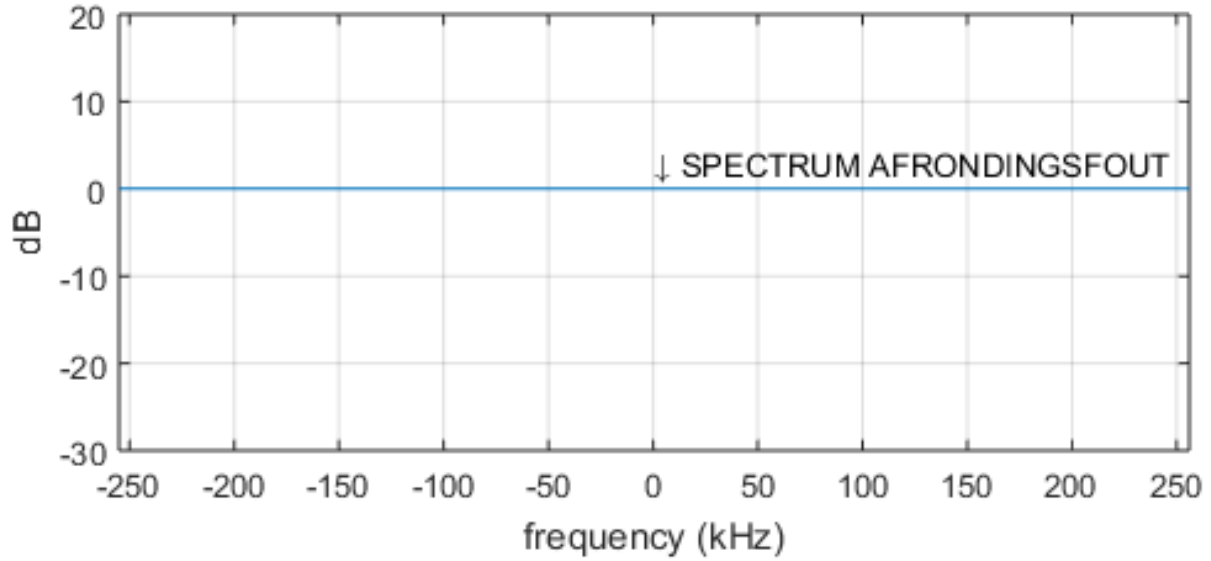
Som van alle fouten vormt de “opgetelde” kwantisatieruis.



Spectrum van één afrondingsfout

De fout is een gegeven => fout normeren naar 1

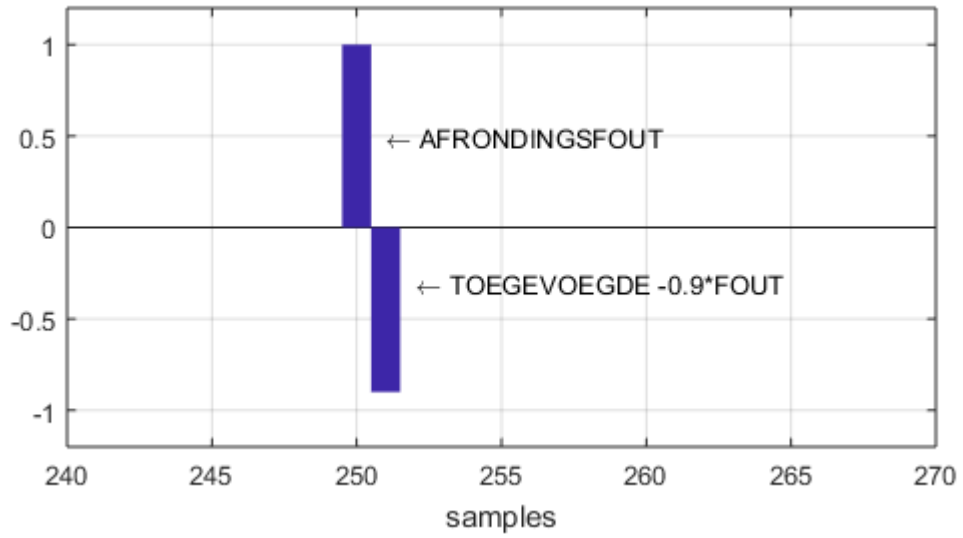


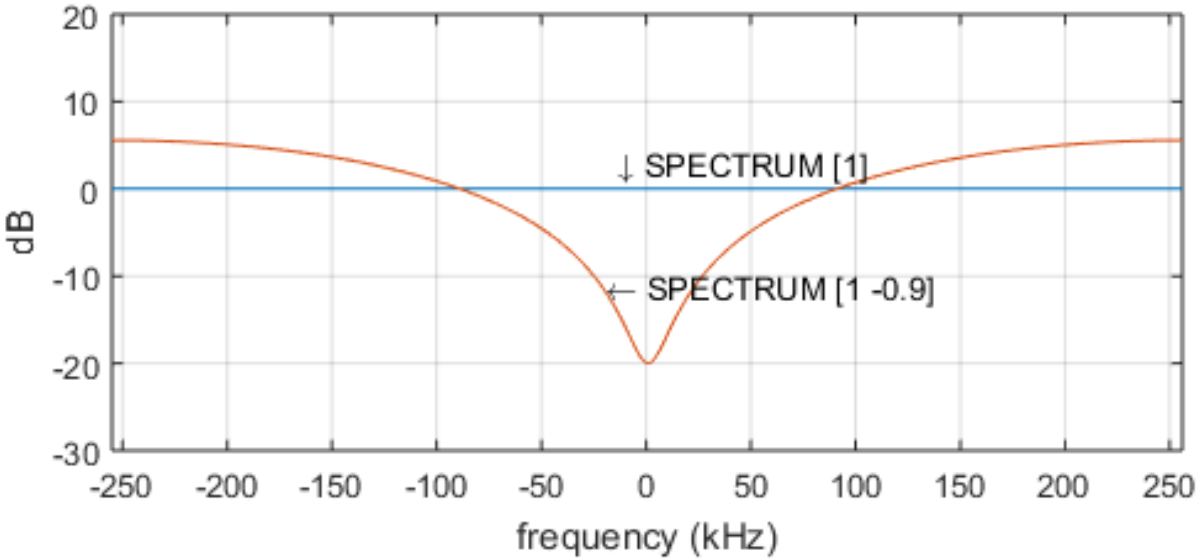


Spectrum van 512 samples met één afrondingsfout is recht.

Spectrum veranderen (shapen)

Bijvoorbeeld door **-0.9×fout** op te tellen bij het volgende 14-bits sample van het ingangssignaal:

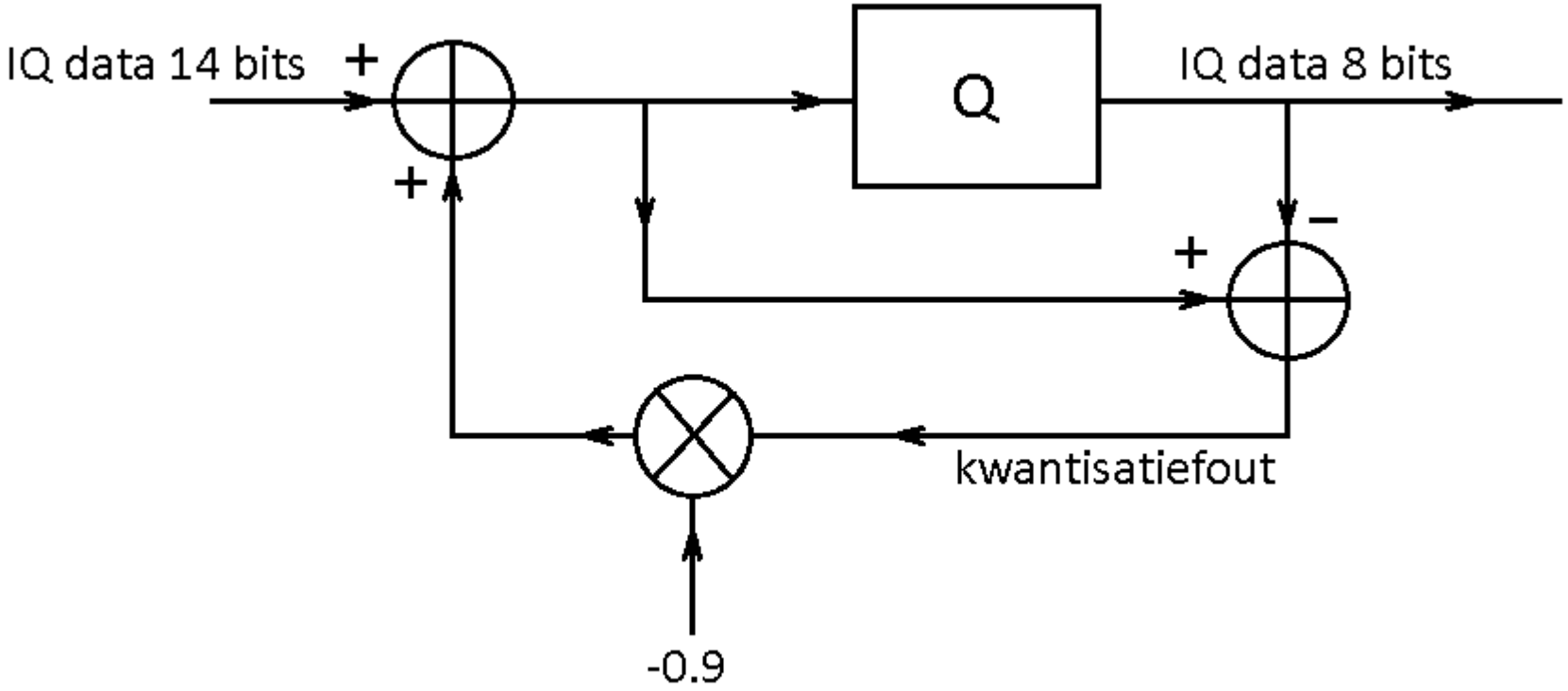




=> DC inhoud -20dB en hoogste frequenties +6dB

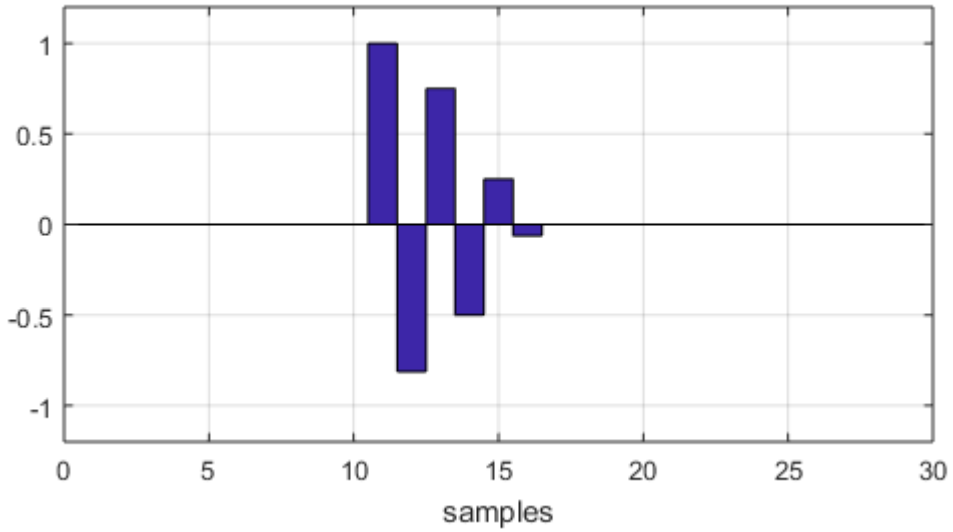
Implementatie

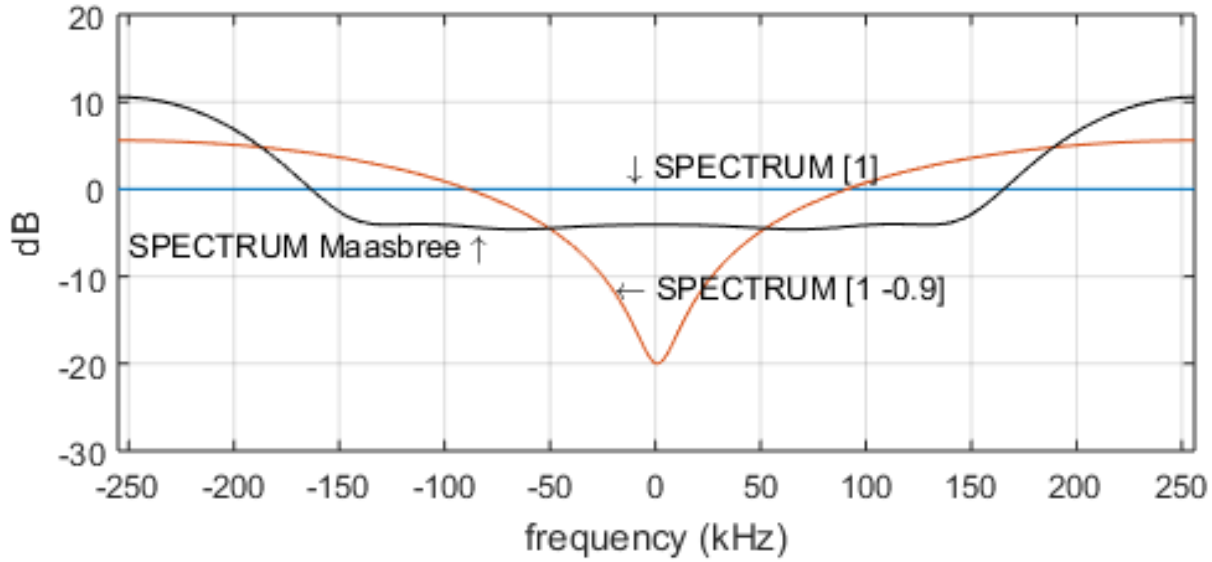
Noise shaping door -0.9×fout op te tellen:



Maasbree Noise Shaping => -4dB

Langere sequence van extra fouten:





NB.: de eerste in de sequence is altijd de gemaakte kwantisatiefout.
Die kun je niet veranderen!

Samengevat

- Door fouten evenredig met de amplitude van de afrondingsfout er later bij op te tellen verandert het kwantisatieruis spectrum
- Streven is in het gewenste gedeelte van het spectrum het niveau van de kwantisatieruis zo ver mogelijk te verlagen.
- Des te meer ruimte beschikbaar in het spectrum om ruis naar te shapen, des te meer valt er te winnen (bv. op 40m).

Goede uitleg is te vinden, inclusief hoe je dit uitvoert, op:

https://www.dsprelated.com/showarticle/184.php