

## Opdrachten Lab 1 – Frequentiedomein

### Leerdoelen

Aan het einde van dit lab ben je in staat om:

- Leren werken met GNU Radio Companion en de PLUTO SDR;
- Beperkingen van de PLUTO SDR te begrijpen;
- AM zenders kunnen demoduleren met een top- en productdetector.

### Vorbereiding op de les

- Installeer de laatste versie van GNU Radio en de drivers m.b.v. [deze handleiding](#).
- Lees de hoofdstukken van PySDR zoals vermeld in de planning in de cursushandleiding.

# 1 Opdrachten

**1.1** Bij de klassikale uitleg zijn een aantal figuren getoond om complexe e-machten uit te leggen. Deze opdracht is een herhaling hiervan, mocht alles duidelijk zijn, dan kun je meteen verder met [paragraaf 1.1](#). Je kunt altijd later op deze opdracht terug komen.

Deze opdracht maakt gebruik van Thonny, [download en installeer](#) dit, mocht je dit programma niet meer hebben. De scripts maken gebruik van drie python bibliotheken die je via Thonny kunt installeren. Open Thonny, ga naar **Tools** » **Manage Packages** en zoek numpy en installeer deze door op **Install** te klikken. Zoek nu op matplotlib en installeer deze ook. Installeer als laatste de bibliotheek scipy.

Download het bestandje [scripts.zip](#) en pak alles uit.

**A** Open het script `tel10_mixers.py` in Thonny. Het stelt je in staat simpel het tijd- en het frequentiedomein te bekijken van een signaal. Als voorbeeld zijn alvast een aantal sinusvormige en exponentiele signalen gedefinieerd met verschillende frequenties. Een mixer is een analoog component wat een vermenigvuldiging uitvoert. De functie van een mixer is om een signaal te verschuiven in het frequentiedomein door twee verschillende signalen te vermenigvuldigen. Het script doet de vermenigvuldiging in python maar het effect in het tijd en frequentiedomein is hetzelfde als bij een analoge mixer. Voer het script uit en beantwoord de volgende vragen:

- Gebruik de formule van Euler om de sinusvormige signalen uit het script om te schrijven naar complexe e-machten. Zie [vergelijking \(1\)](#).

$$\cos(\omega t) = \frac{e^{j\omega t} + e^{-j\omega t}}{2} \tag{1}$$

$$\sin(\omega t) = \frac{e^{j\omega t} - e^{-j\omega t}}{2j}$$

- Met de signalen omgerekend naar complexe e-machten, reken de vermenigvuldiging na die het script uitvoert. Kun je verklaren waar alle frequentiecomponenten te zien zijn in de figuren?
- Speel met de frequenties van de signalen en kijk wat er gebeurt in het frequentiedomein. Mocht je niet kunnen verklaren wat er gebeurt, vraag dan een docent om hulp.

**B** Open nu het script `tel10_complexe_exponent.py`. Dit script laat de complexe e-machten zien in het tijd en frequentiedomein, maar dan in 3D (tijd of frequentie op 1 as, imaginair en reeel op de andere twee assen). Een complexe e-macht is gewoon een vector welke ronddraait met een bepaalde snelheid. Als je deze draaiing over de tijd weergeeft, krijg je een spiraal te zien. Voer het script uit en beantwoord de volgende vragen:

- Hoe zie je aan het 3 dimensionale tijdsignaal of de frequentie positief of negatief is?
- Hoe ziet een negatieve amplitude er in het tijd domein uit?
- Druk op de  knop, dit laat alleen het reële deel van het signaal zien. Met alleen het reële deel, kun je dan nog onderscheid maken tussen positieve en negatieve frequenties en/of amplituden? Met de formules van euler, welke complexe e-machten/figuren worden dan opgeteld om een cosinus te maken? Herhaal dit voor het imaginaire deel voor een sinus. Wanneer je op het huisje drukt, onder aan het scherm, dan worden alle grafieken gereset naar de beginstand.
- SDR's werken intern volledig met complexe e-machten. Wat is het voordeel van het werken met complexe signalen in plaats van reële signalen?

**C** Hoe wordt een echt signaal dan complex gemaakt? En hoe wordt een complex signaal weer reëel gemaakt? Open het script `tel10_iqmixer.py` en voer het uit. De eerste twee rijen laten de analoge/reële situatie zien binnen een bepaalde bandbreedte. Het moduleren en demoduleren gebeurt met twee reële mixers/vermenigvuldigingen. Het verzonden signaal (figuur 1) wordt in de onderste rij ook gedemoduleerd door een IQ mixer. Let op dat binnen de SDR de bandbreedte kleiner is. Dit hangt af van de samplerate van de gebruikte ADC's. In het laatste figuur wordt een tweede IQ mixer toegepast om het complexe spectrum van binnen de SDR weer te plaatsen op de draaggolffrequentie. Beantwoord de volgende vragen:

- Wat is de signaalfrequentie? Zie je deze frequenties terug in de gedemoduleerde spectra? Pas de informatiefrequentie binnen het script aan en kijk wat er gebeurt.
- In de analoge demodulator zie je ook twee (ongewenste) frequenties rond de 100 kHz. Hoe zou je er analoog voor zorgen dat die frequenties weg worden gehaald?

- In de complexe IQ demodulator zie je twee frequenties. Hoe relateert dat aan het informatiesignaal?
- Welke samplerate zou de SDR hebben gebruikt in deze figuren? Zou de tweede IQ mixer signalen kunnen verzenden die buiten de samplerate van de SDR liggen? Probeer het uit door de verschuiving binnen de SDR groter te maken. Bijvoorbeeld 25 kHz in plaats van 10 kHz. Kun je verklaren wat je ziet?

## 1.1 GNU Radio Companion

We gaan op dit moment overstappen naar GNU Radio Companion. Net als Simulink maakt GNU Radio Companion gebruik van blokken om acties uit te voeren. Er zijn verder sinks en sources om respectievelijk signalen in te stoppen of uit te halen. We zullen in eerste instantie wat signalen gaan aanpassen en bekijken maar later zullen we dit ook koppelen aan een Software Defined Radio om de signalen ook echt te zenden en ontvangen. Zorg ervoor dat je de hoofdstukken uit de voorbereiding hebt gelezen (!).

In de volgende opdracht gebruiken we de volgende blokken:

- *Signal Source* om een signaal te genereren.
- *Noise Source* om (witte) ruis te genereren.
- *Add* om twee of meerdere signalen bij elkaar op te tellen.
- *Multiply* om twee of meerdere signalen met elkaar te vermenigvuldigen
- *Low Pass Filter* om een signaal te filteren.
- *QT GUI Frequency Sink* om het frequentiedomein van een signaal te bekijken.

**1.2** Het doel van deze opdracht is om beter begrip te krijgen van het frequentiedomein en basisoperaties. Dit ga je doen door het spectrum na te bouwen wat de docent op het bord heeft getekend. Gebruik hiervoor de eerder benoemde blokken. Je mag overleggen met elkaar. Mocht je vastlopen, lees dan de tips. <sup>1 2 3 4</sup>

---

<sup>1</sup> Vermenigvuldigen met een complexe e-macht levert een verschuiving in het frequentiedomein.

<sup>2</sup> Witte ruis is een soort DC in het frequentiedomein; een vlakke PSD.

<sup>3</sup> Gezien we complex werken is een filter met **reële** coëfficiënten altijd symmetrisch rond de 0 in het frequentiedomein. Ook het low pass filter!

<sup>4</sup> Je kunt het aantal ingangen instellen voor de *QT GUI Frequency Sink*. Je kunt dan meerder signalen bekijken voordat ze bij elkaar worden opgeteld.

**1.3** Laten we wat over gaan zenden! Zenden doen we met de PLUTO SDR en het *Soapy PLUTO Sink* blok, ontvangen met het *Soapy PLUTO Source* blok. Laat 1 computer/laptop een signaal zenden en een andere computer/laptop het signaal ontvangen. Voer de volgende acties uit:

- A** Geef de schakelingen een sample rate van 1 MHz door de *samp\_rate* variabele aan te passen.
- B** Kies een *Center Freq* van, dus stel de lokale oscillator van de SDRs in op, een frequentie tussen de 70 MHz en 1,7 GHz en geef de zender een maximale *RF Gain* onder de *RF Options* van het blok. De center frequency van de Soapy blokken gebruiken de variabele *freq*.
- C** Het is een goede gewoonte om bij de zender te kijken wat er verstuurd wordt en bij de ontvanger te kijken wat er ontvangen wordt, dus sluit ook een *QT GUI Frequency Sink* aan om dit te doen.
- D** Als laatste heeft de zender een signaal nodig om te zenden. Dit doen we met een *Signal Source*. Maak gebruik van een *QT GUI Range* blok om de frequentie van het signaal aanpasbaar te maken en koppel de signaalbron aan het Soapy blok.
- E** Run de zend- en ontvangschakelingen en bekijk of je het signaal kunt ontvangen. Vraag de docent om hulp als het niet lukt.

Gebruik deze schakeling als basis voor de volgende opdrachten. Je kunt schakelingen dupliceren met `CTRL` + `SHIFT` + `D` of via `File` > `Duplicate`.

**1.4** Een ander basisprincipe is dat het zendvermogen gelimiteerd is. Het theorema van Parseval bewijst dat je het vermogen van een signaal kunt berekenen door de PSD te integreren over het frequentiebereik of het tijdsignaal over een bepaalde tijd. Met de *RMS* en *QT GUI Number Sink* blokken kun je continu het signaalvermogen weergeven bij het zenden. We gaan bepalen bij hoeveel vermogen het signaal niet meer goed wordt verzonden.

- A** Dupliceer de vorige opdracht en pas de volgende dingen aan bij de zender.
  - Voeg een extra *Signal Source* toe van 50 kHz en tel deze op bij de eerste met een *Add* blok.

- Voeg aan de uitgang van het Add blok een *Fast Multiply Const* blok toe en maak deze aanpasbaar van 0-1 met een *QT GUI Range* blok. Hiermee kunnen we het signaal versterken.
  - Sluit een *QT GUI Time Sink* aan op de uitgang van de signaalversterker.
  - Maak de RF Gain van het *Soapy PLUTO Sink* aanpasbaar van 0-89 met een *QT GUI Range* blok.
  - Plaats een *RMS* aan de uitgang van het Fast Multiply Const. De uitgang van het RMS blok kan naar een *QT GUI Number Sink* blok.
- B** Zet de zender en ontvanger ongeveer een meter van elkaar af. Zorg ervoor dat de QT GUI Frequency Sink bij de ontvanger een FFT-grootte heeft van tenminste 8192. Zet de RF Gain nu maximaal en de versterking van het signaal op 0. Verhoog de signaalversterking nu stapsgewijs en kijk tot welk vermogen het signaal zonder vervorming wordt ontvangen. Schrijf dit vermogen op.
- C** Geef het Add blok een extra ingang en sluit hier een extra signal source op aan van 100 kHz. Stuur het opnieuw op en controleer het vermogen waarbij er vervorming plaatsvindt. Komt dit overeen met je eerdere meting?
- D** Wat gebeurt er als de maximale signaalspanning van de zender boven de 1 uitkomt?

Heel dit vak zul je rekening moeten houden met de maximale signaalspanning en vermogenslimieten van de zender.

## 1.5 De docent stuurt drie AM signalen uit: DSB, DSB-SC en SSB.

- Begin met het demoduleren van het AM signaal door het signaal apart te krijgen met filters, de amplitude te bepalen en het door te sturen naar een *Audio Sink* m.b.v. een *Fractional Resampler* zoals uitgelegd in de theorie. Als je het muziekje hoort kun je door naar DSB-SC.
- Demoduleer nu het DSB-SC signaal door het terug te schuiven naar 0 Hz, maak dit aanpasbaar met een Range blok in stappen van 10 Hz. Zoals je merkt komt het afstellen erg nauw. In de praktijk wordt daarom ook een draaggolf meegezonden.
- Herhaal dit voor het SSB signaal.