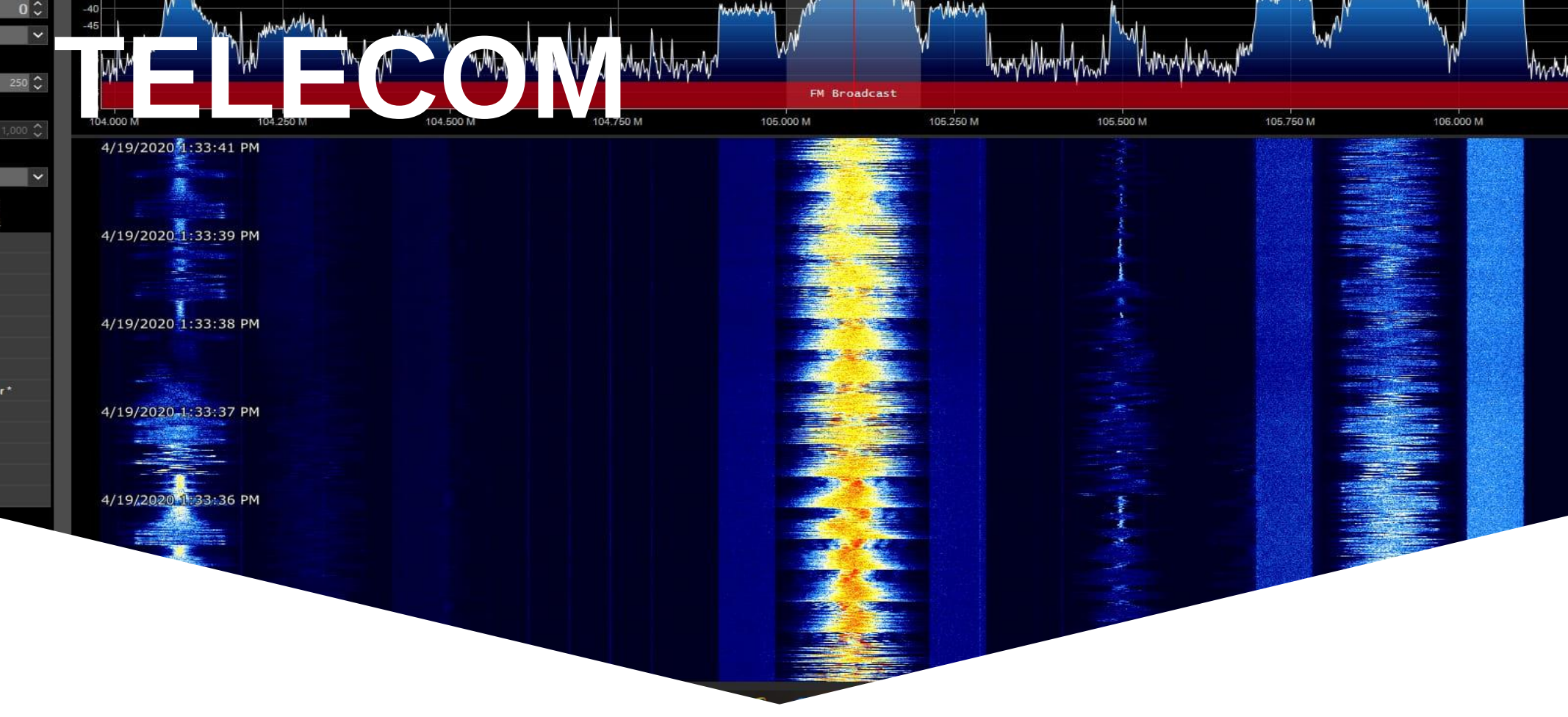

TELECOM
-40
-45
FM Broadcast
104.000 M
104.250 M
104.500 M
104.750 M
105.000 M
105.250 M
105.500 M
105.750 M
106.000 M
4/19/2020 1:33:41 PM
4/19/2020 1:33:39 PM
4/19/2020 1:33:38 PM
4/19/2020 1:33:37 PM
4/19/2020 1:33:36 PM

TEL10 – Lab 3.2

Overzicht

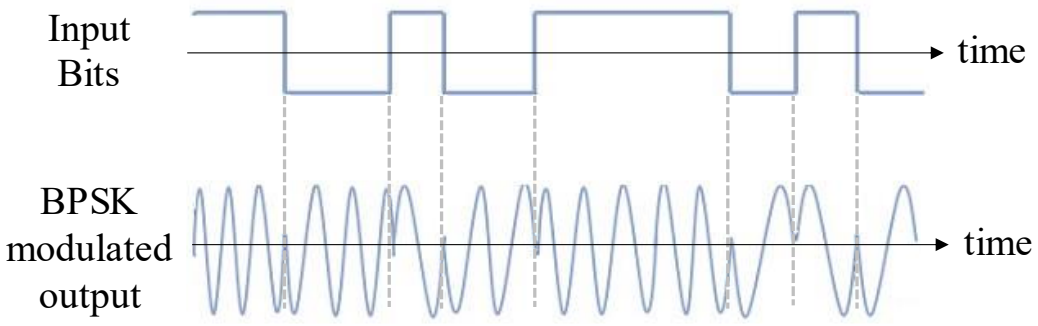
Planning

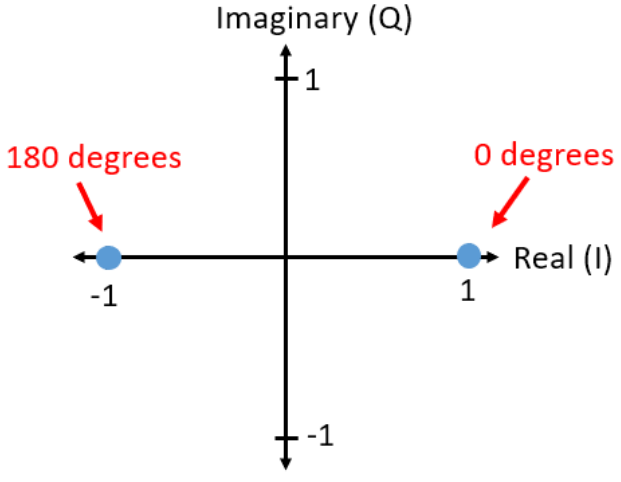
- PSK
- Inter symboolinterferentie
- Frequentiedetectie
  - Grove
  - Fijne
- Blokken

PSK

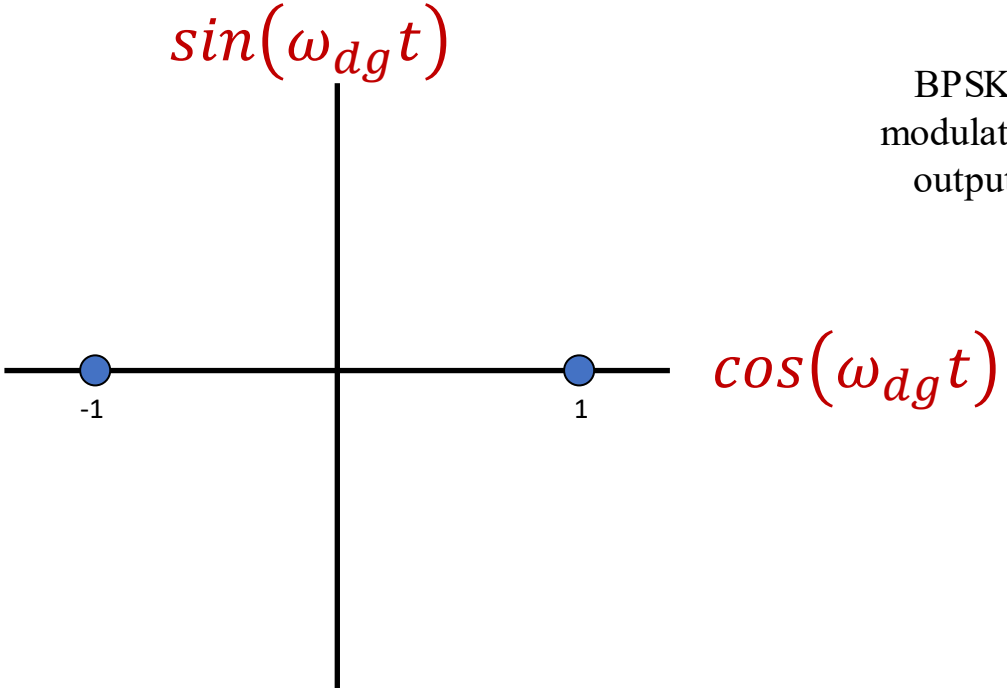
Phase Shift Keying

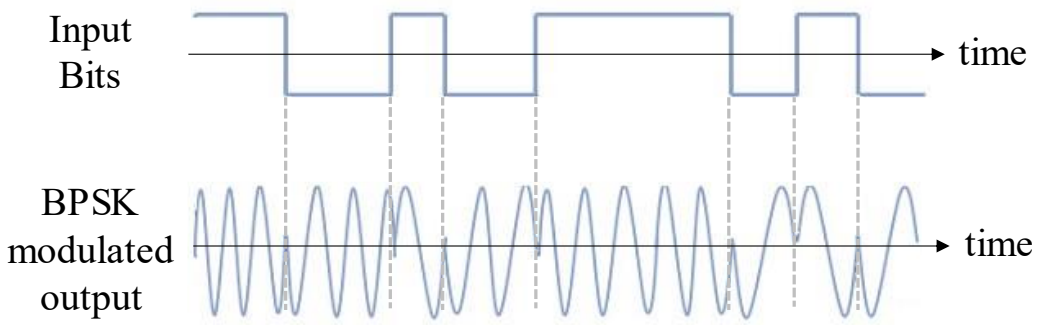
- ASK zonder offset/draaggolf
- Ontvangen op dezelfde manier als DSB-SC (productdetector).
- Efficiënter dan ASK
- Handiger te bekijken als mogelijke waarden in het complexe vlak.



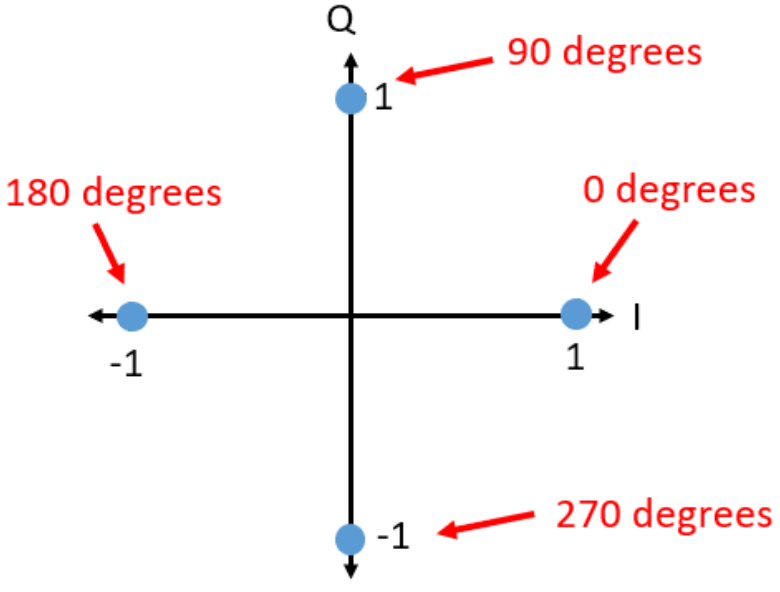


IQ diagram

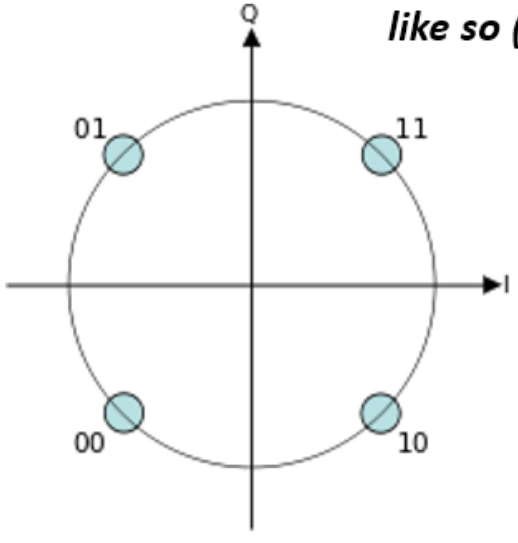




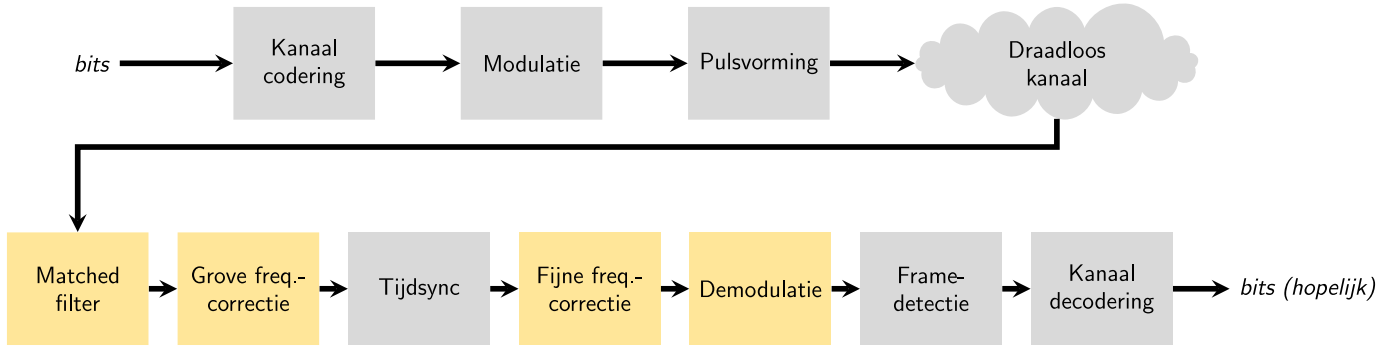
QPSK



You can also represent it like so (it is equivalent)



Overzicht



Dit lab

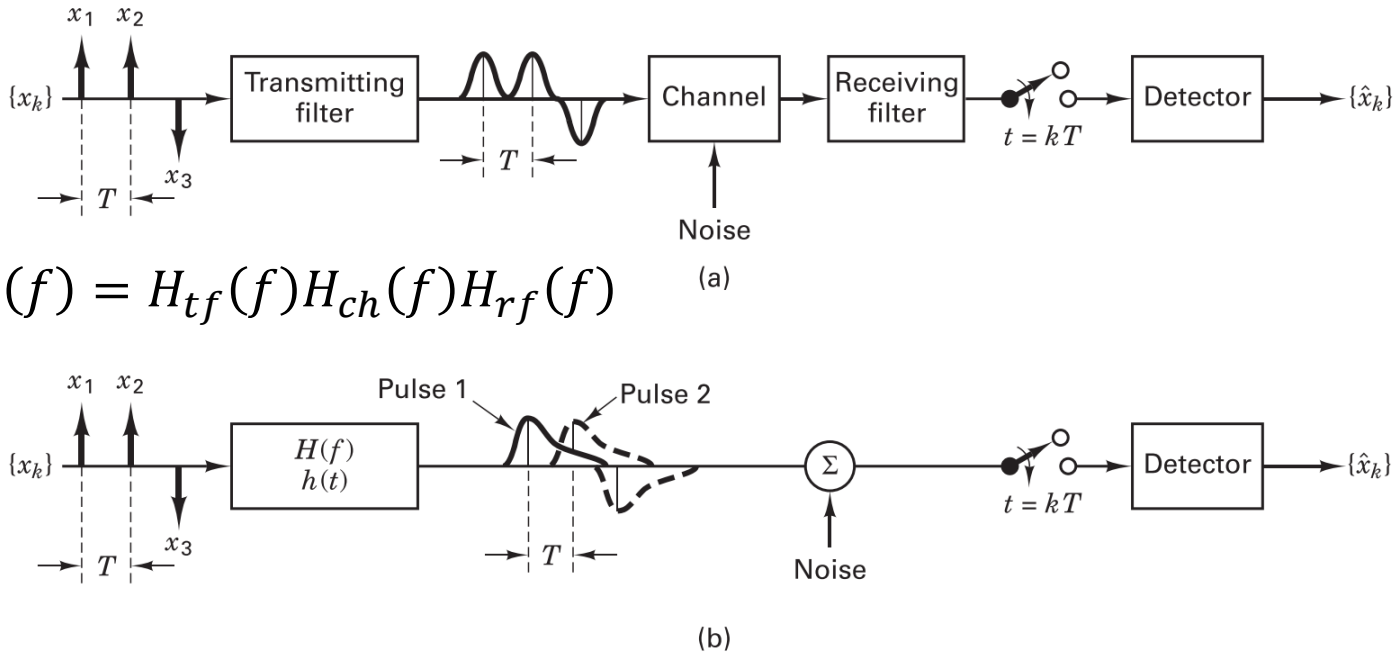
1. Matched filter
2. Grove frequentiecorrectie
3. Fijne frequentiecorrectie

ISI

Probleem

Bandbreedteminimalisatie en kanaaleffecten leiden tot uitsmeren symbolen in de tijd bij de ontvanger.

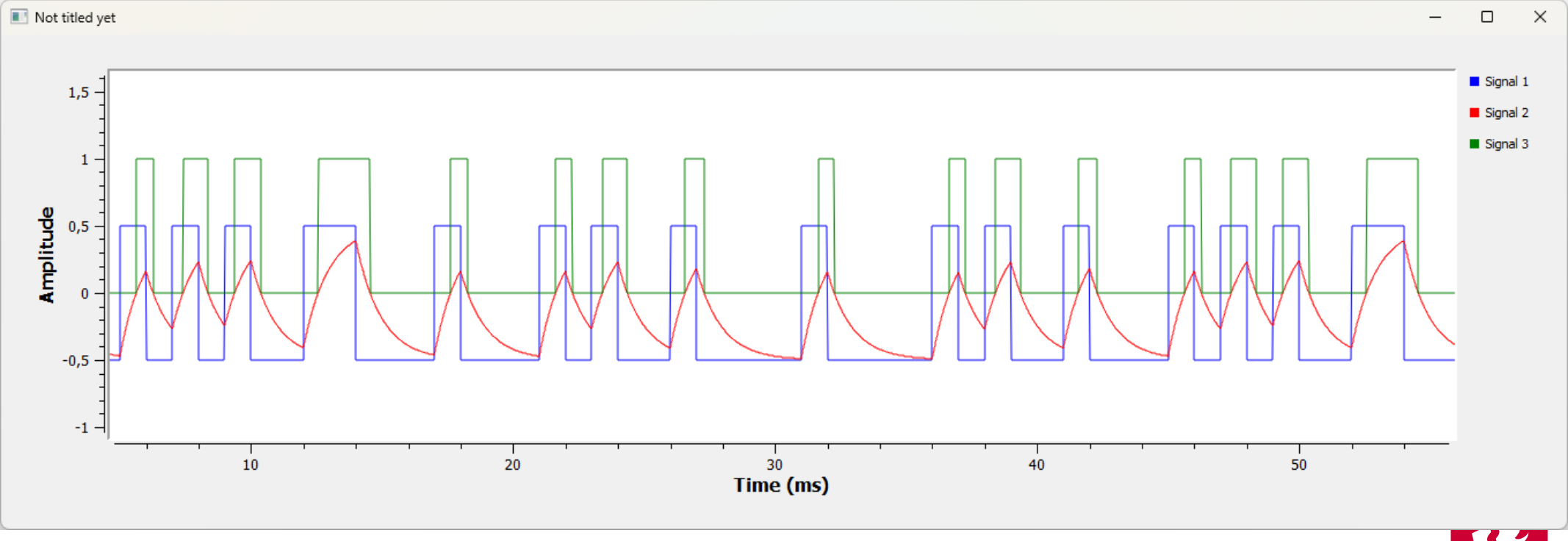
Dit maakt detectie moeilijker (meestal met threshold)



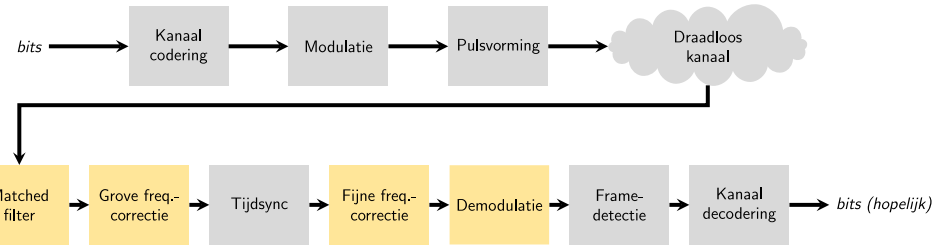
$$H(f) = H_{tf}(f)H_{ch}(f)H_{rf}(f)$$

Figure 3.15 Intersymbol interference in the detection process. (a) Typical baseband digital system. (b) Equivalent model.

Simulatie ISI met simpel RC filter



ISI



Oplossing

Speciale filters gebruiken.

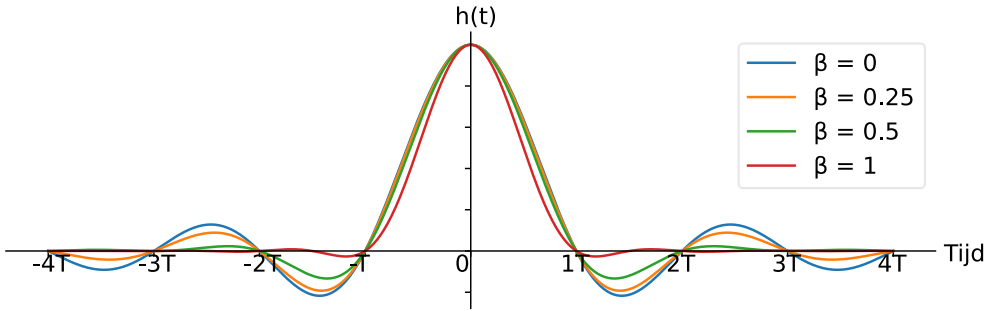
Impulsresponsie waarbij elke periodetijd de waarde 0 is.

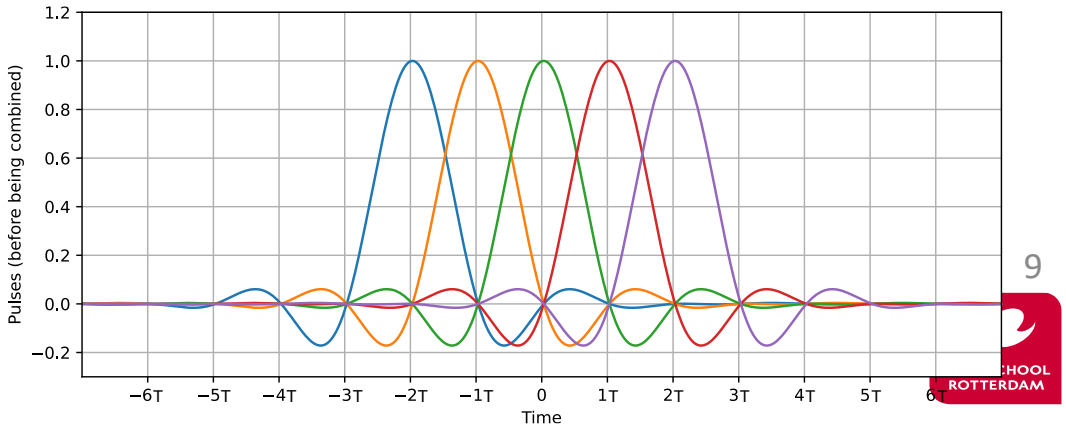
$$h(nT) = 0, n \neq 0$$

Geen invloed op volgend symbool!

Vaak een **Raised Cosine Filter**

$$h(t) = \frac{1}{T} sinc\left(\frac{t}{T}\right) \frac{\cos\left(\frac{\pi\beta t}{T}\right)}{1 - \left(\frac{2\beta t}{T}\right)^2}$$





ISI

Vaak een **Raised Cosine Filter**

Probleem

Beide zender en ontvanger willen filteren (**waarom**?)

Oplossing

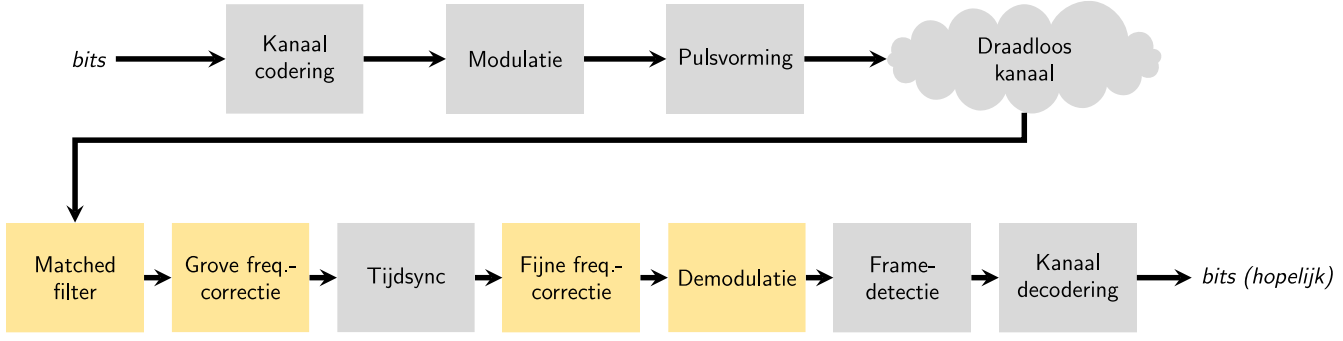
Filter splitsen in twee stukken

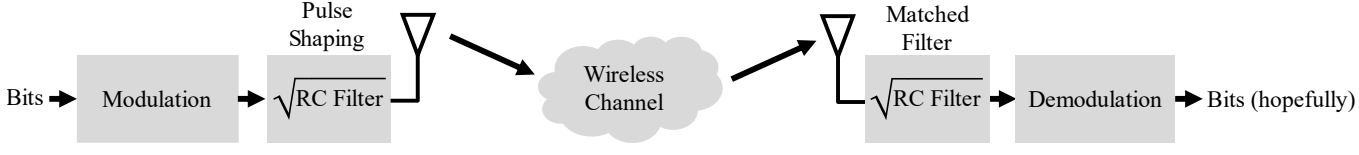
$$X_{rc}(f) = X_{rrc}(f)X_{rrc}(f)$$

Met

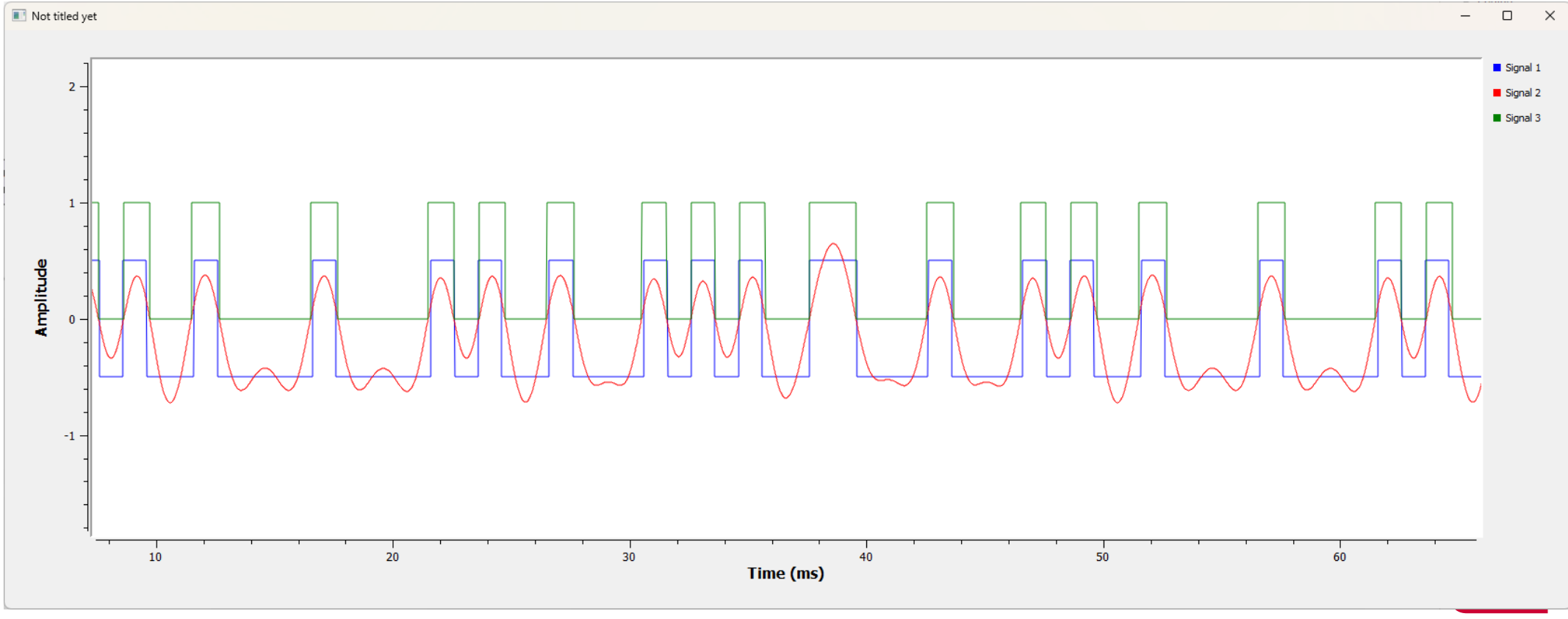
$$X_{rrc}(f) = \sqrt{X_{rc}(f)}$$

Matched filters!





Simulatie met Raised Cosine filter $\beta = 0.35$

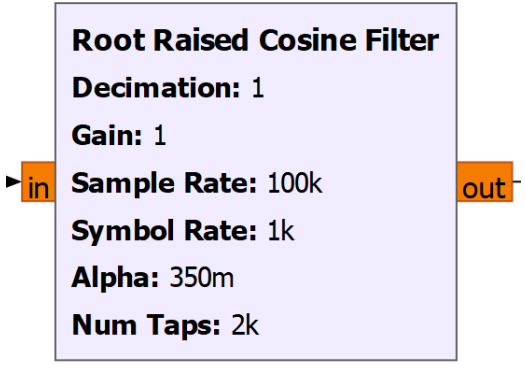


Root Raised Cosine Filter

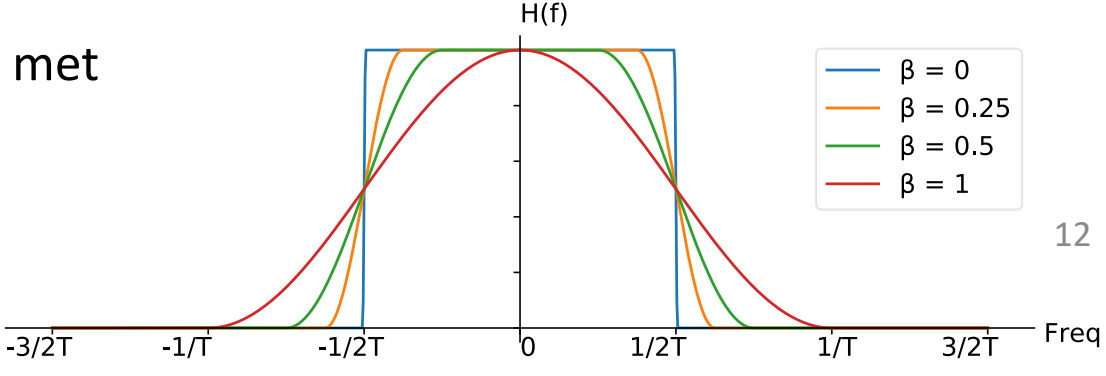
Apart blok voor RRC filter $= \sqrt{X_{rc}(f)}$

- **Symbol Rate:** symbolen per seconde
- **Alpha** (β in voorgaande slides): tussen 0 en 1

- **Num Taps:** Aantal coëfficiënten
  => meer coëfficiënten betere benadering



Dus 1 bij de zender en 1 bij de ontvanger met dezelfde instellingen (matched!!)



Frequentiedetectie

Probleem

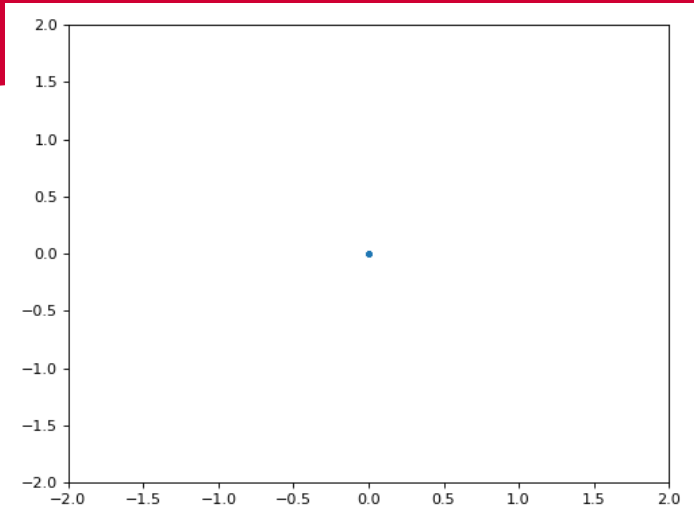
Frequentieafwijking

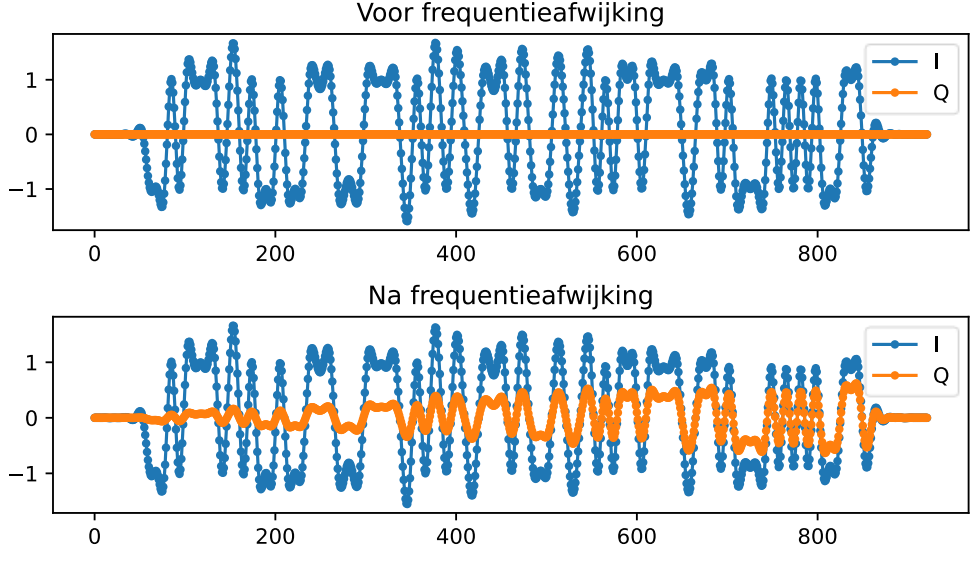
Met de hand erg lastig om precies te corrigeren

Gevolg

Draaiing in I/Q – diagram. Waarom?

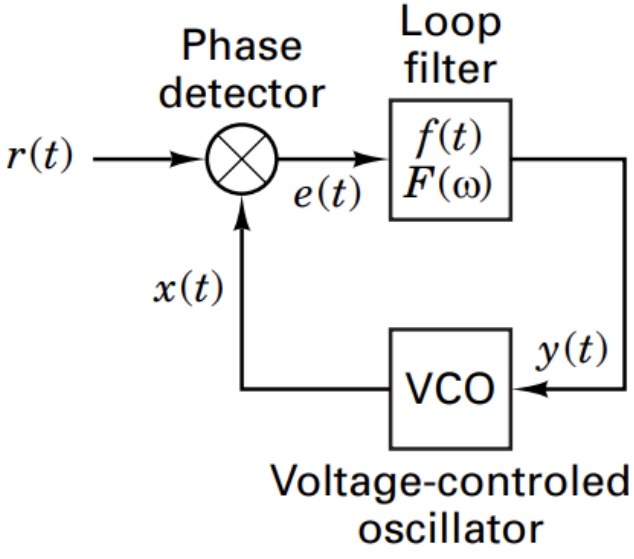
Fase niet meer duidelijk





Synchronisatie

1. **Frequentie/Fase**
2. Symbool
3. Frame



Bernard Sklar, Digital communications

Phase Locked Loop (PLL)

$$\mathrm{r}(t) = \cos\left(\omega_o t + \phi(t)\right)$$

$$x(t) = -2\sin\left(\omega_0 t + \hat{\phi}(t)\right)$$

$$\begin{aligned} e(t) &= x(t)r(t) \\ &= -2\sin\left(\omega_0 t + \hat{\phi}(t)\right)\cos\left(\omega_o t + \phi(t)\right) \\ &= \sin\left(\phi(t) - \hat{\phi}(t)\right) + \sin\left(2\omega_0 t + \hat{\phi}(t) + \phi(t)\right) \\ &\approx \phi(t) - \hat{\phi}(t) \end{aligned}$$

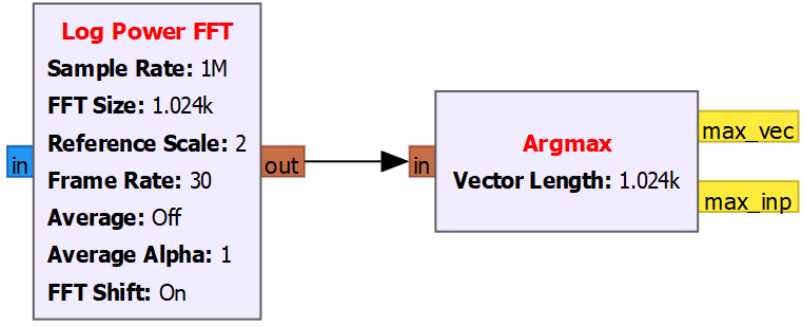
Grove detectie

In boek: neem macht en bepaal max frequentie

GNURadio kan met vectoren werken

Vector is een combinatie van bijv. 1024 samples

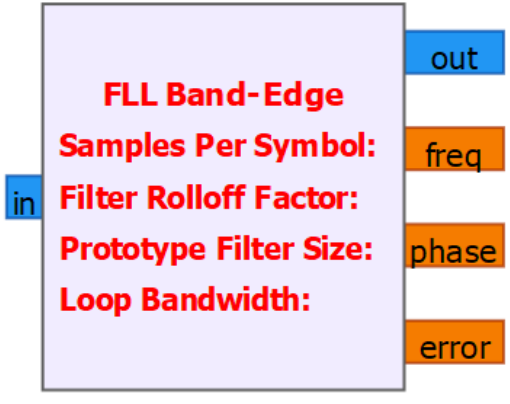
Handig voor een “breedbandige” aanpak



smalbandige oplossing (gebaseerd op PLL):

- Filter Rolloff Factor, gebruikte alpha/beta RRC
- Prototype Filter Size: hoeveel coëfficiënten?
- Loop bandwidth: Bandbreedte regelaar, tussen $\pm\pi$
  Prima waarde is $\frac{2\pi}{100} \approx 62.8m$

Uitgang is verschoven versie signaal



Band-Edge

Afgeleide van het RRC filter.

Zolang het signaal nog binnen 1 van de bandpass filters zichtbaar is, kan het corrigeren.

Dus voor een datarate van $1kbps$ en filterbreedte van $2kHz$, dan mag de “draaggolf” van het ontvangen signaal uiterlijk $\pm 2kHz$ afwijken

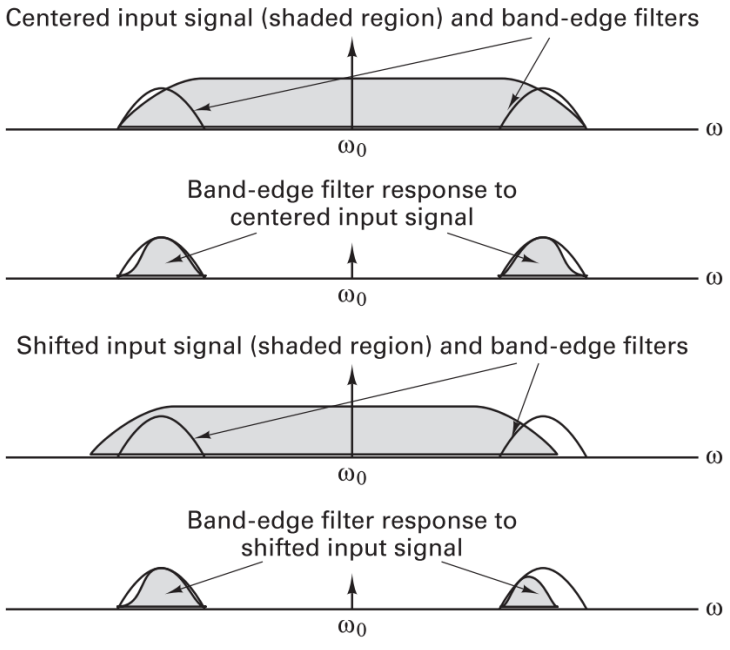
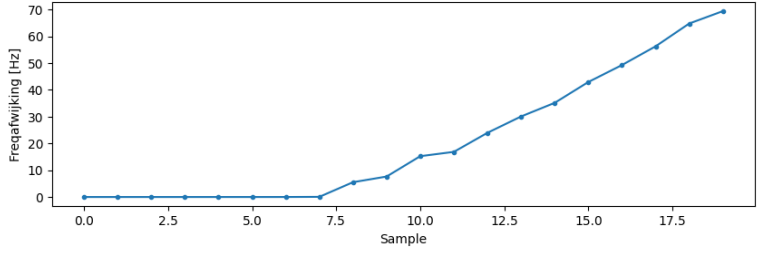


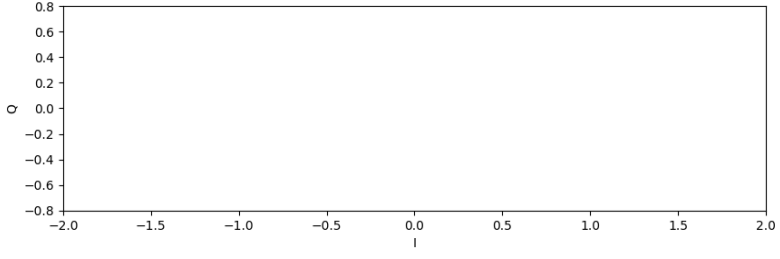
Figure 10.10 Band edge filter example.
Bernard Sklar, Digital communications

Fijne Detectie

Costas-Loop

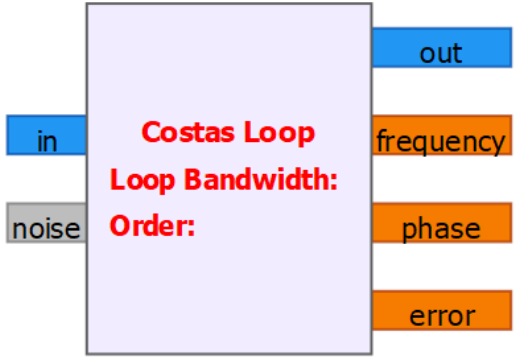
```
fase = 0
# Deze volgende twee parameters bepalen of de feedback loop sneller of langzamer reageert (wat de stabiliteit beinvloed
alpha = 0.132
beta = 0.00932
uit = np.zeros(N, dtype=complex)
for i in range(N):
    uit[i] = samples[i] * np.exp(-1j*fase) # pas de ingang aan met de inverse van de geschatte faseafwijking
    fout = np.real(uit[i]) * np.imag(uit[i]) # De is de foutvergelijking voor de 2e orde Costas-loop (dus voor BPSK)

    # Update de fase en frequentie
    freq += (beta * fout)
    fase += freq + (alpha * fout)
```





Costas Loop

Alleen geschikt voor 2, 4 of 8-PSK



Costas loop

Loop Bandwidth: Bandbreedte regelaar, goede waarde $\frac{2\pi}{100}$

Order: 2 voor BPSK, 4 voor QPSK of 8 voor 8-PSK

Uitgang is ingang zonder frequentie/fase verschuiving.