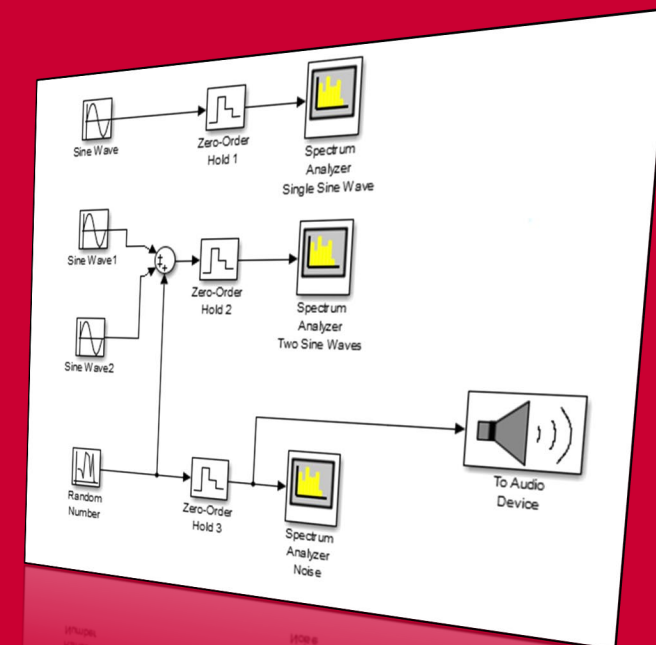
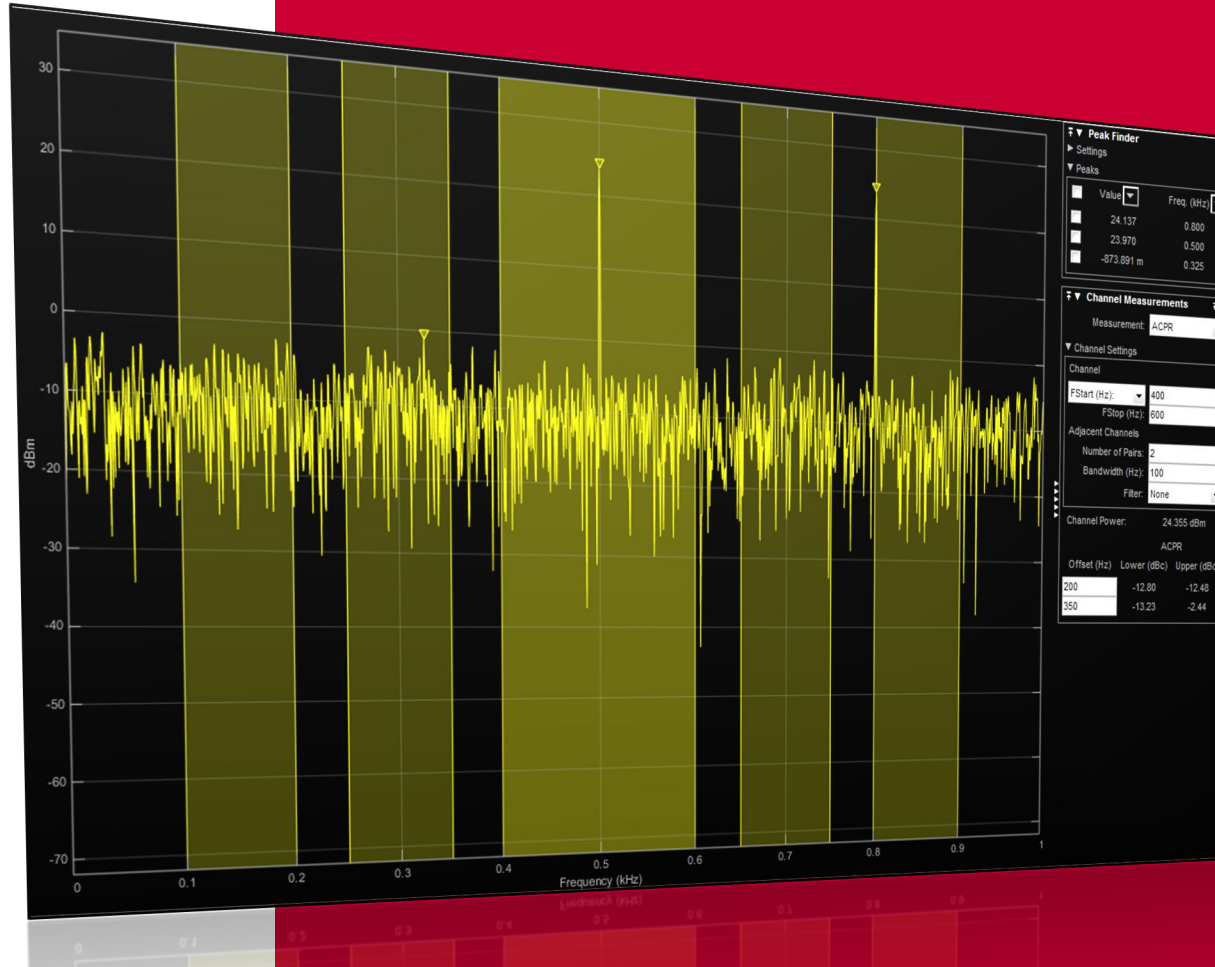




ELEDIS10

Digitale Systemen

Doel van DIS10

DIS10 gaat over **digitale signaalbewerking (DSP)**.

Waar denk je dan aan?

Doel van DIS10

Digitale signaalbewerking (DSP) houdt zich bezig met de **wiskundige bewerkingen** die we kunnen uitvoeren op (vaak) **gesamplede signalen**, zodat

- we er nuttige informatie uit kunnen **filteren**;
- of het signaal kunnen bewerken.

Doel van DIS10

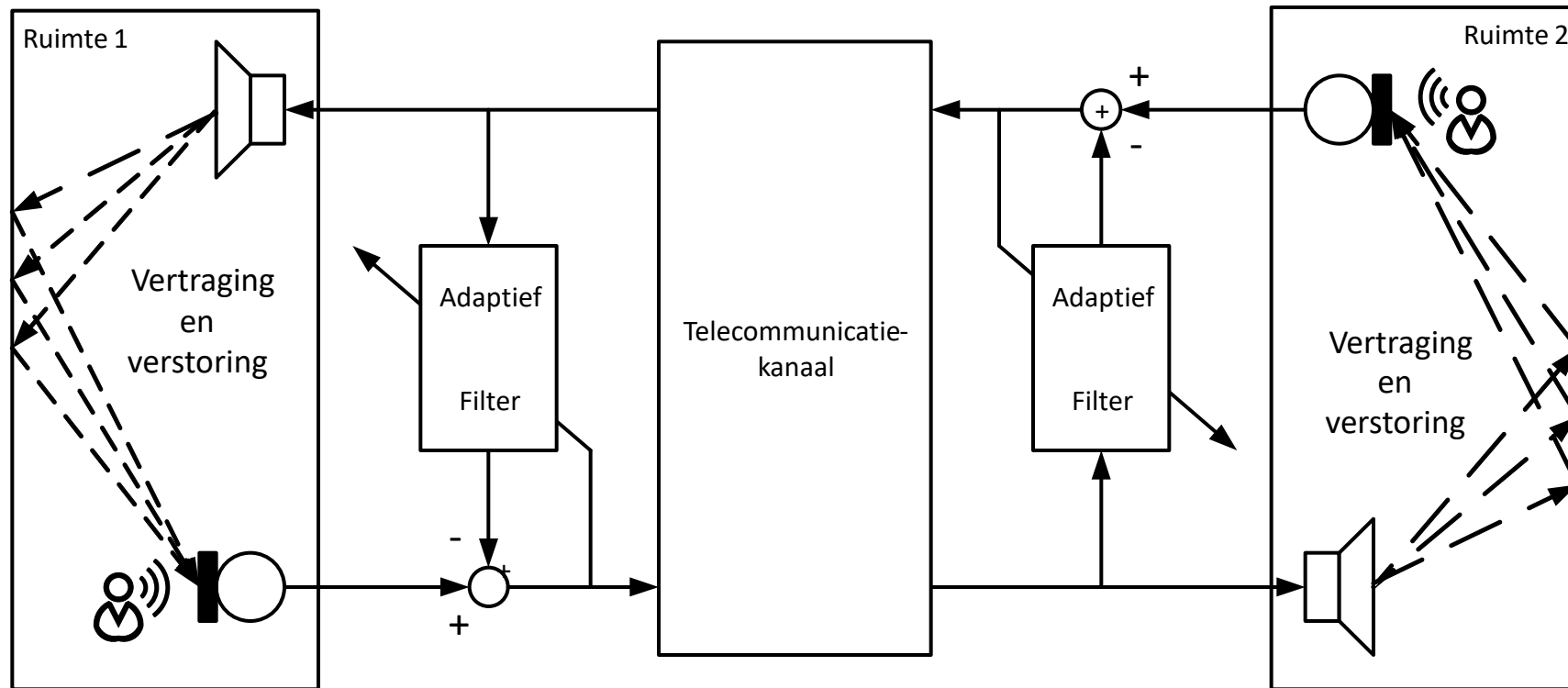
Digitale signalen van sensoren kunnen gebruikt worden om systemen te **regelen**, zodat

- eigenschappen van een regelaar dynamisch kunnen worden ingesteld;
- zichzelf instellende regelaars kunnen worden gemaakt.

Inhoud week 1

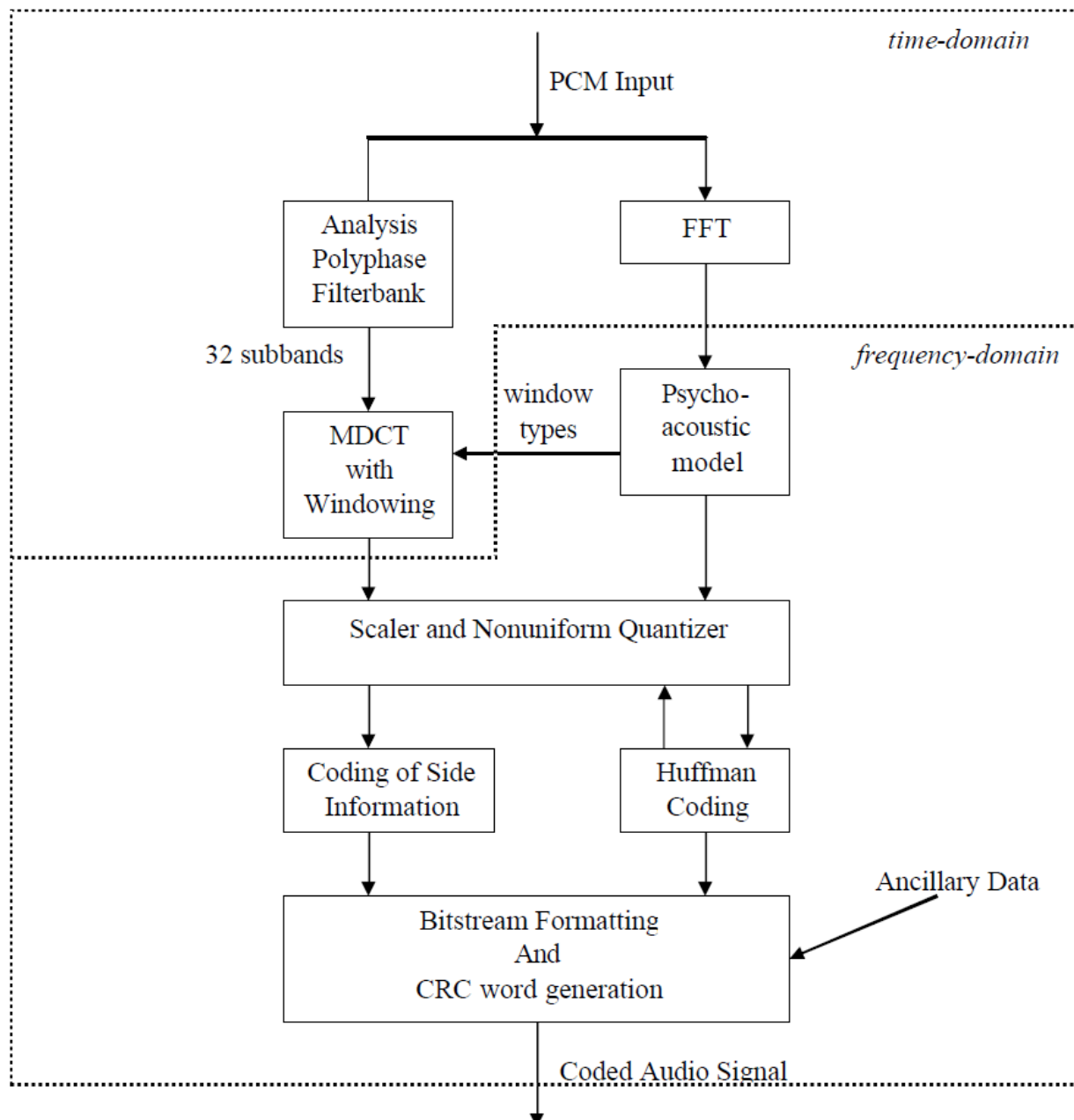
- Typische DSP applicaties
- Organisatie DIS10
- Sampling
- Basisvormen van tijd-discrete signalen
- Lineaire tijdsinvariante (LTI) systemen

Telecommunicatie toepassing: echo-onderdrukking



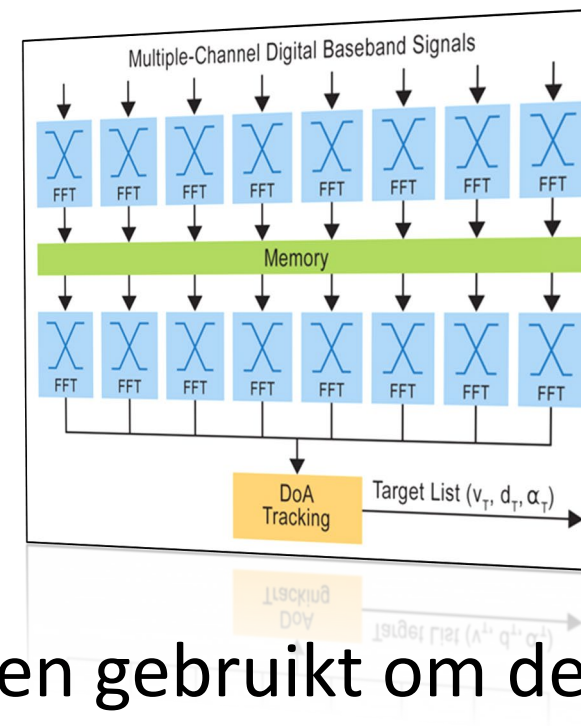
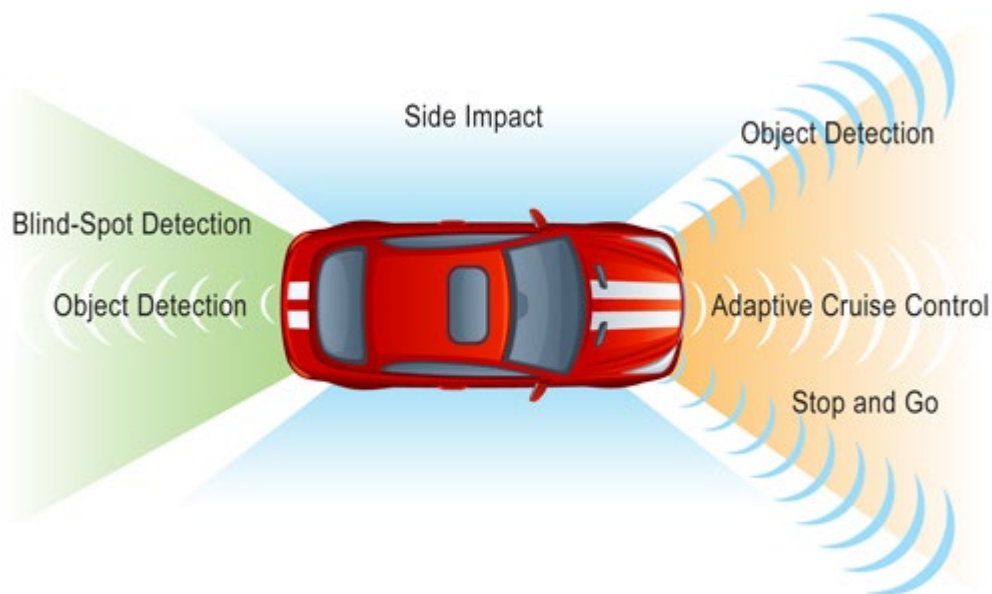
- Bijv. bij hands-free bellen; je microfoon pikt op wat de luidspreker uitstuurt.
- Dit kan d.m.v. DSP technieken worden onderdrukt voordat er weer verzending plaatsvindt.

Audio-toepassing: MP3 compressie



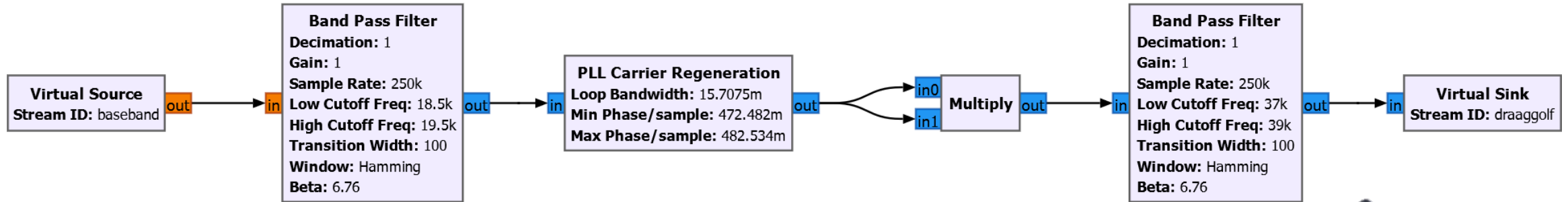
- Een belangrijk onderdeel van MP3 encoding is de Fast Fourier Transform (FFT) en de Modified Discrete Cosine Transform (MDCT).
- FFT is een “snel” Fourier transformatie DSP algoritme.
- Hiermee kun je van een weergave in de *tijd* naar een weergave in *frequenties* van een signaal.

Automotive toepassing: object detectie



- DSP algoritmes waaronder de FFT worden gebruikt om de doppler shift gemeten door radar te berekenen.
- Op deze manier wordt de snelheid waarmee objecten op je af komen bepaald.
- Ook militaire toepassingen gebruiken deze techniek.

Telecom toepassing: SDR



Bron: https://bytebucket.org/HR_ELEKTRO/tel10/wiki/practica/Lab_3.pdf

- SDR = Software Defined Radio
- Wordt uitgebreid behandeld in kwartaal 4 bij TEL10.



Bron: <https://pysdr.org/nl/content-nl/pluto.html>

Organisatie DIS10, zie [cursushandleiding](#)

Opzet:

6 ECTS = 168 uur = 1 dag/week gedurende twee semesters

Kwartaal 3

- 3 weken 2x theorieles (op school)
- 1 week practicum¹ (online)
- 2 weken 2x theorieles (op school)
- 1 week practicum¹ (online)
- 1 week 2x theorieles (op school)
- Tentamen van theorie kwartaal 3

¹ Tweetallen zelf te kiezen.

Kwartaal 4

- 3 weken practicum² (lab)
- 2 weken 2x theorieles (op school)
- 1 week practicum² (online)
- 2 weken practicum (lab)
- Aftekenen van practicum kw 4
- Verslag van practicum kwartaal 4

² Tweetallen worden ingedeeld aan de hand van de resultaten van het tentamen van kwartaal 3.

Organisatie DIS10, zie [cursushandleiding](#)

Toetsing:	Weging:	Min:	Deadline:
• Matlab Notebook 1	5 punten	-	Week 3.5
• Matlab Notebook 2	5 punten	-	Week 3.8
• Tentamen kw3:	40 punten	16	T3
• Aftekenen opdrachten:	10 punten	-	Week 4.8
• Verslag kw4:	40 punten	16	T4

Eindcijfer = Σ punten / 10 als minimaal aantal punten behaald is.

Boek (aanbevolen voor uitgebreide uitleg en verdieping):

“Inleiding Digitale Signaalbewerking met Maple en Matlab” – Lynn, P.A.; Fuerst, W.; Andriessen, J.W.M., ISBN 90-557-4448-4, tweede druk. Te bestellen bij [studiebijdehand](#).

Materiaal:

CC3220S-LAUNCHXL en pc

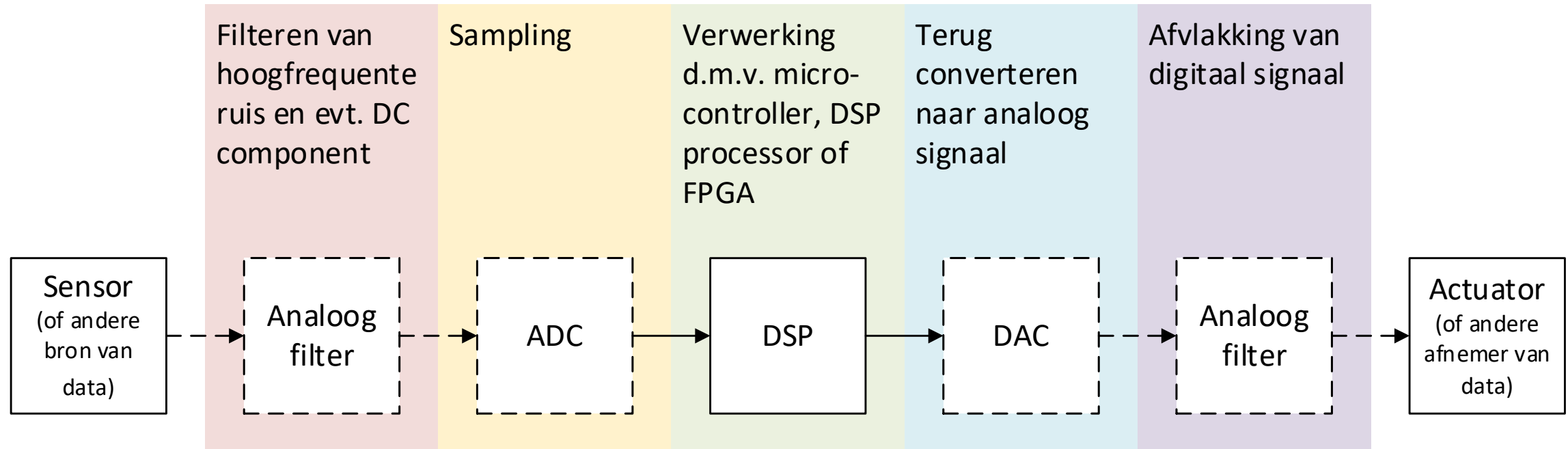
CC3200AUDBOOST boosterpack en HRDIS10 boosterpack

Leerdoelen, zie [cursushandleiding](#)

Aan het eind van de module ben je in staat om:

- ... een tijd-discreet signaal of LTI-systeem te **analyseren in het tijddomein**, zodanig dat bijvoorbeeld de impuls- en stapresponsie berekend kan worden.
- ... een tijd-discreet signaal of LTI-systeem gespecificeerd in het **tijddomein om te zetten naar** een representatie in een **frequentiedomein en vice versa**. (d.m.v. de DFT, DTFT, Z-transformatie) zodanig dat het signaal of systeem makkelijker te analyseren is.
- ... de **frequentieresponsie** van een systeem te **analyseren** aan de hand van de polen en nulpunten van dat systeem.
- ... een **digitaal filter** te ontwerpen, **realiseren** en testen met behulp van zowel software als hardware tools op een embedded platform.
- ... een discrete **PID-controller** voor een eenvoudig te regelen systeem te ontwerpen, **realiseren** en testen met behulp van zowel software als hardware tools op een embedded platform.

Typische werkwijze van een DSP-toepassing



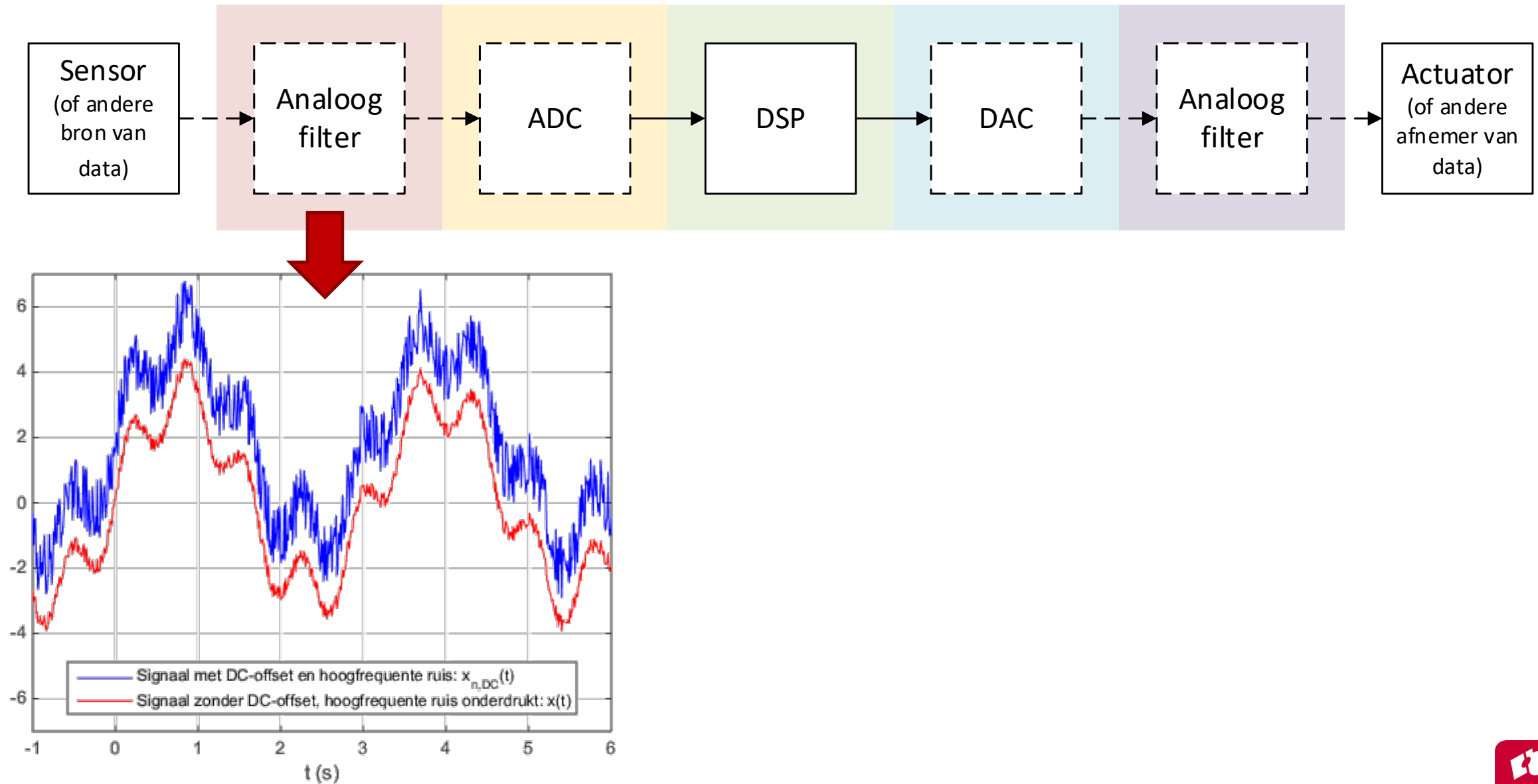
Waarom doen we zo moeilijk?
We kunnen signalen toch ook analoog bewerken.

Vershil Digital Signal Processing t.o.v. Analog SP

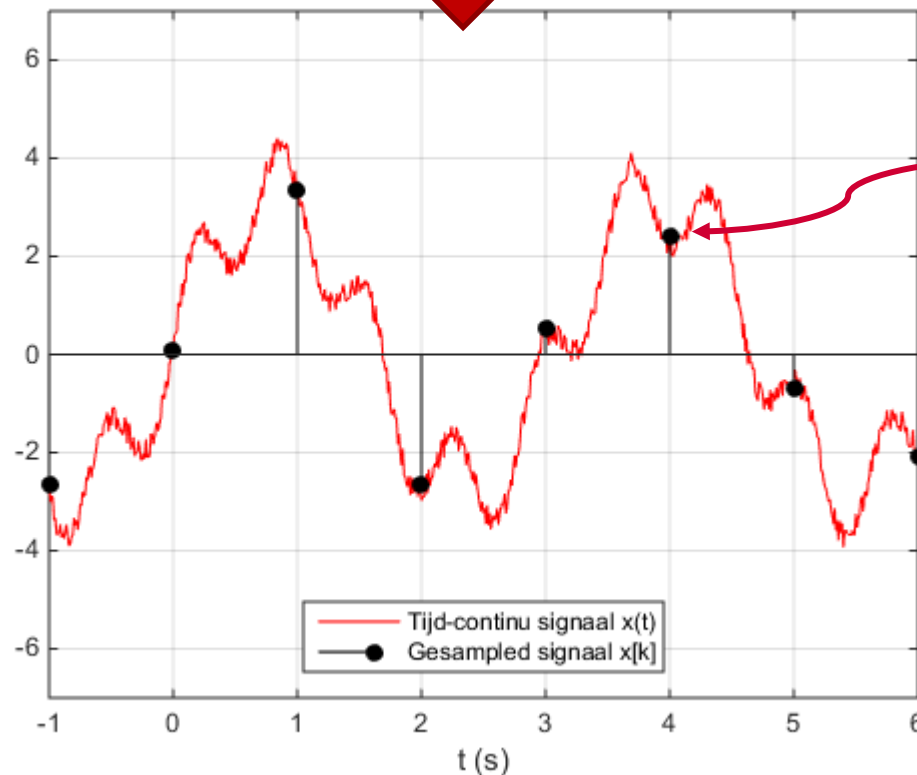
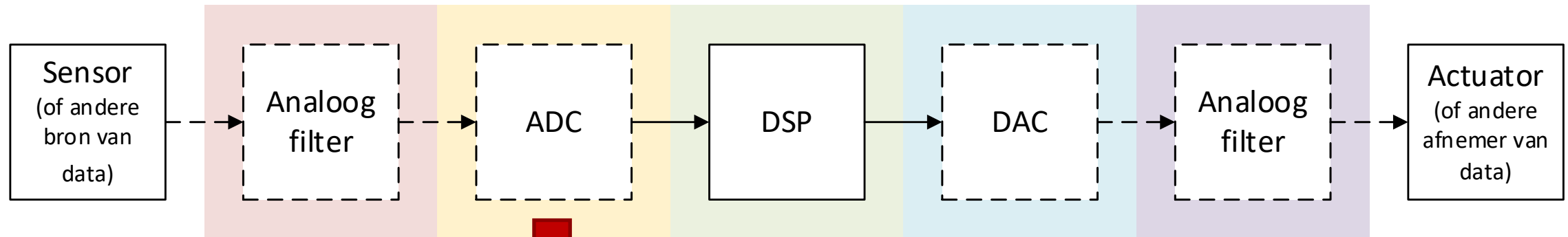
Verschillende aspecten om op te vergelijken:

- Flexibiliteit
- Betrouwbaarheid
- Reproduceerbaarheid
- Complexiteit
- Kosten

Ingang filteren

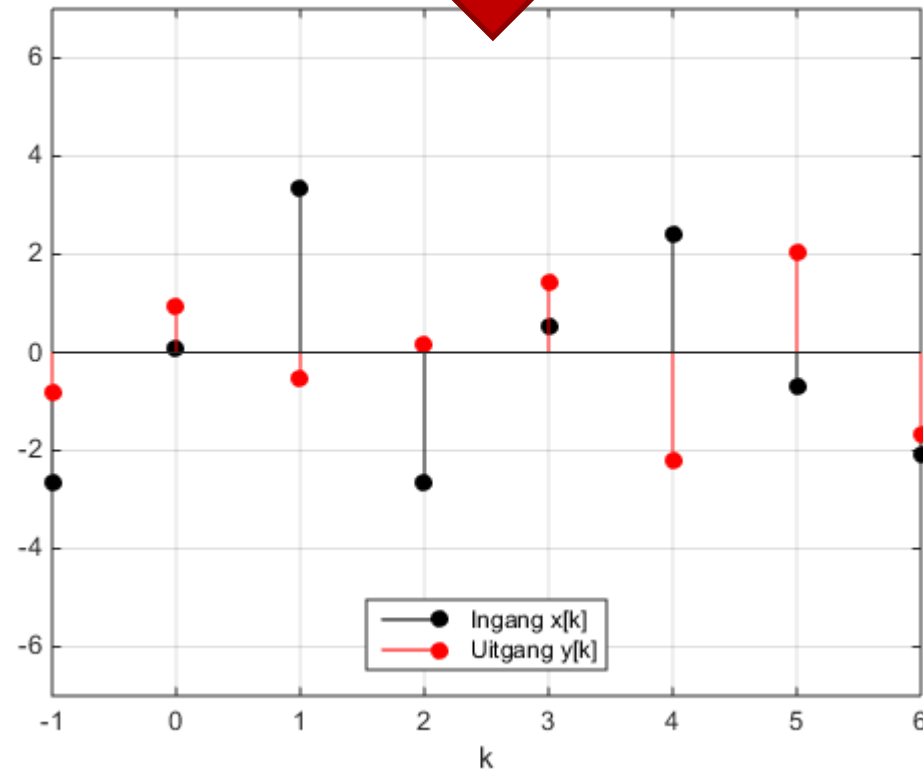
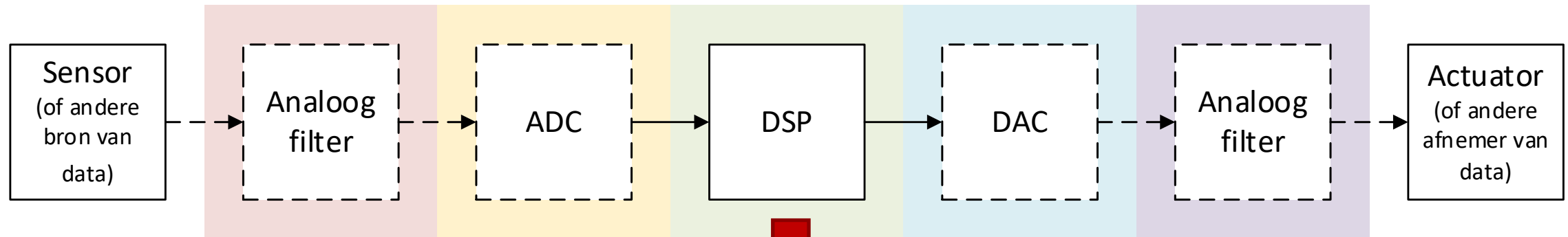


Analoog → Digitaal en Tijd-continu → Tijd-discreet

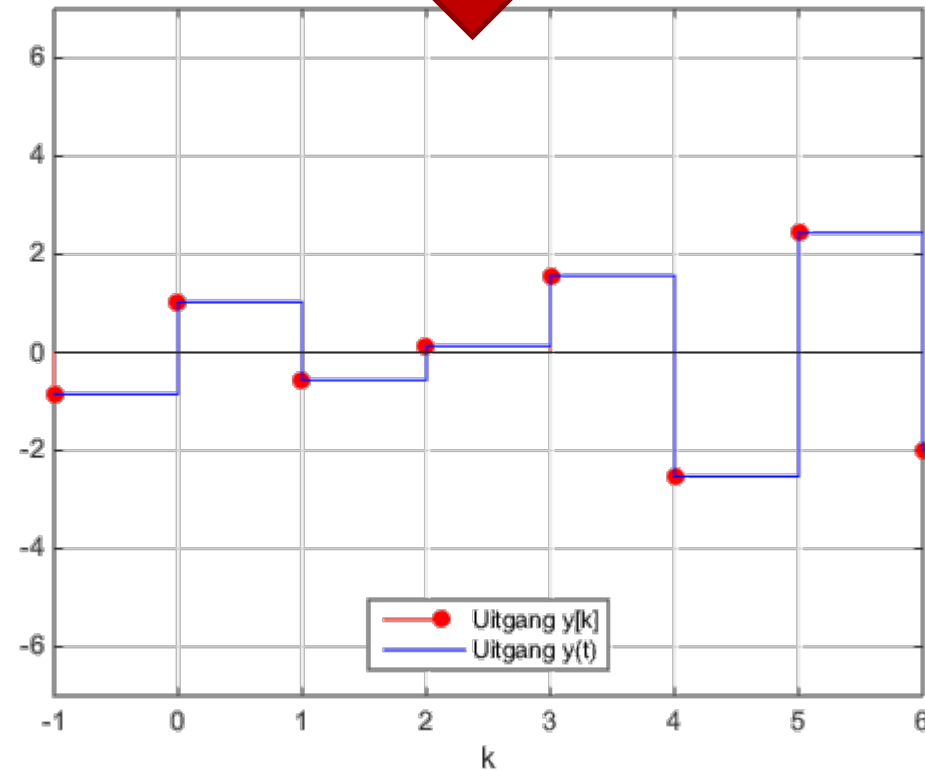
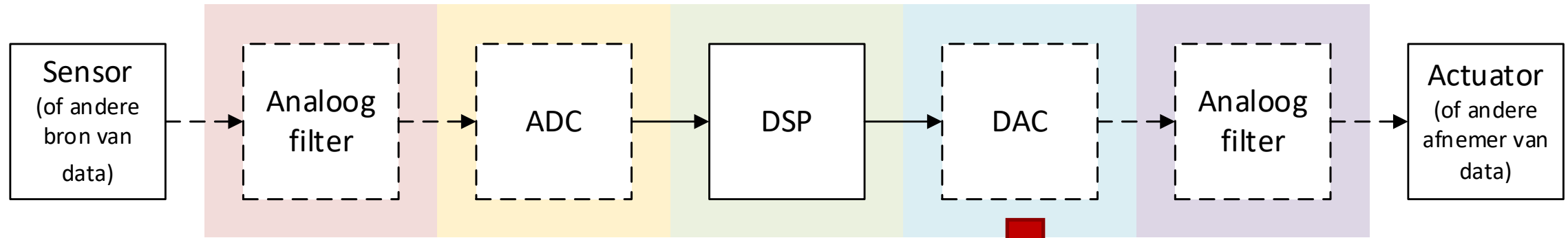


Door de ADC wordt de waarde van het sample zelf ook gekwantiseerd

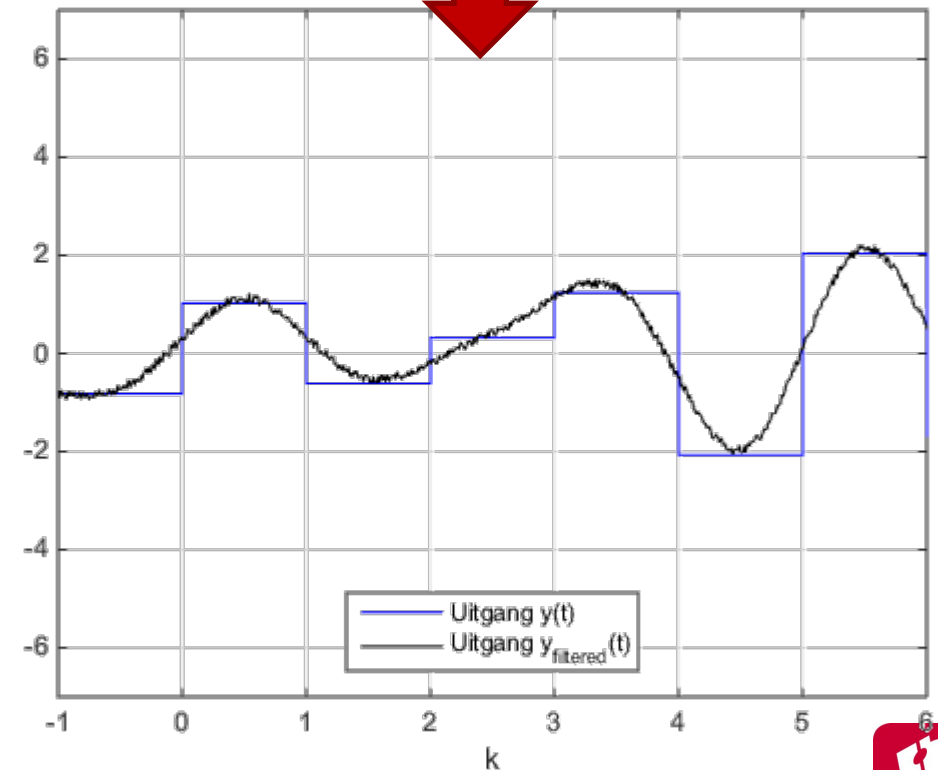
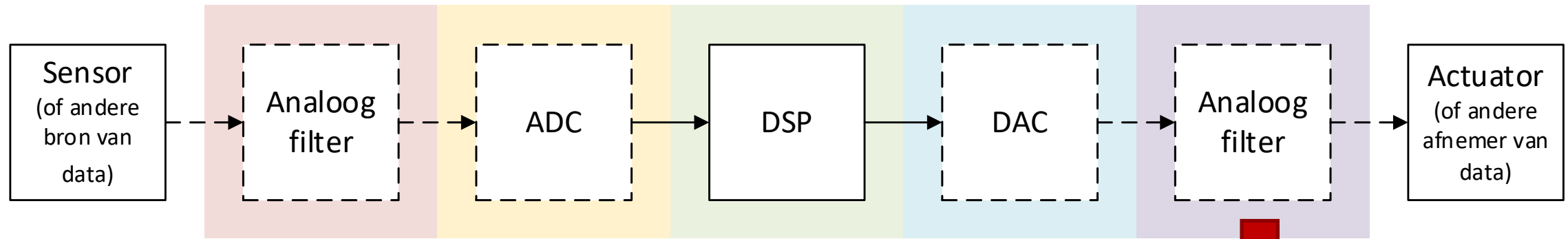
Verwerking in de DSP



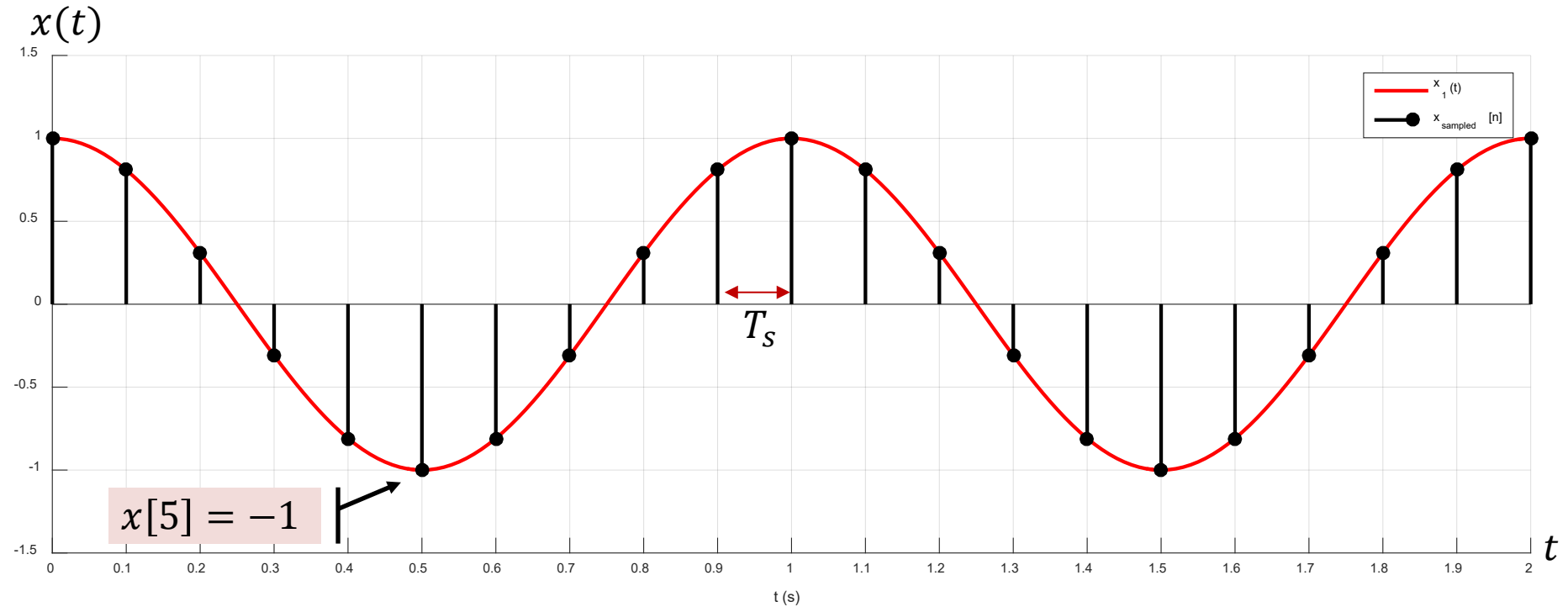
Digitaal → Analoo, Tijddiscreet → Tijdcontinu



Hogere harmonischen/frequenties wegfilteren



Sampling



Gesamplede signaal

$x[n]$

=

Oorspronkelijke signaal

$x(n \cdot T_s)$

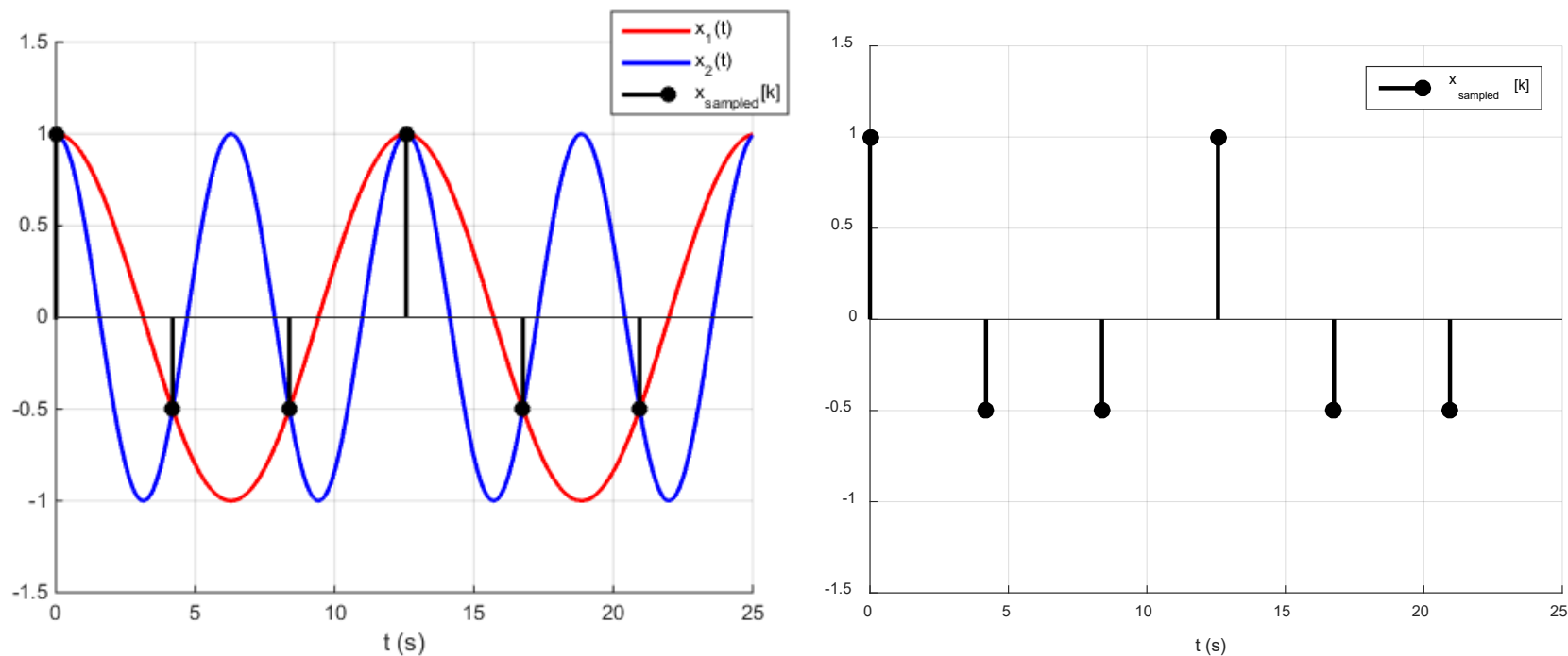
t : tijd

$n \in \mathbb{Z}$: sample nummer

T_s : sample interval / sample tijd

$f_s = \frac{1}{T_s}$: sample frequentie

Samplen is niet éénduidig



- Verschillende continue signalen leiden tot hetzelfde gesampled signaal
- Één gesampled signaal kan met vele verschillende continue signalen (replica's) geassocieerd worden

Spectrum

De grafiek, die ontstaat als we een signaal uitzetten tegen de frequenties (i.p.v. de tijd), heet het **spectrum** van het signaal.

Voorbeeld:

Beschouw een signaal met een bepaalde (hoek)frequentie ω_1 :

$$x(t) = A \cdot \cos(\omega_1 t)$$

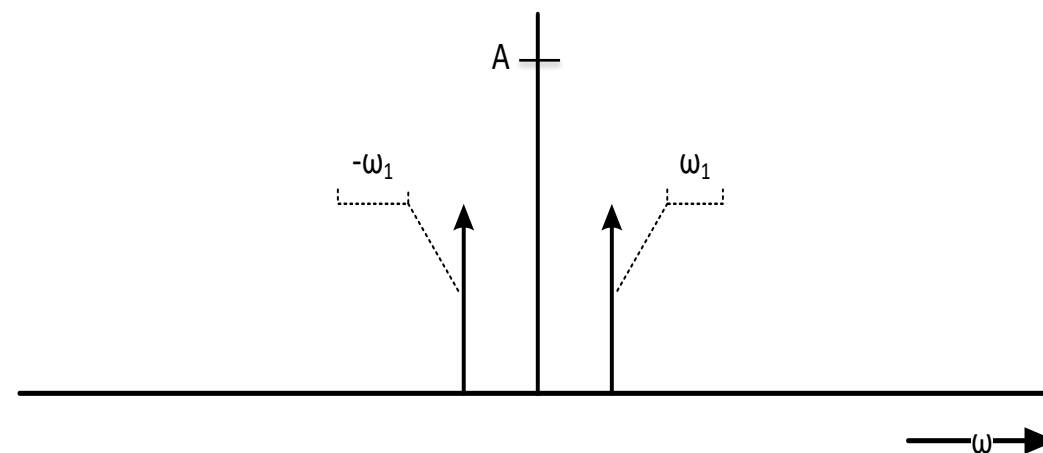
Met behulp van de formule van Euler:

$$e^{j\omega_1 t} = \cos(\omega_1 t) + j \sin(\omega_1 t),$$

kun je dit omschrijven als:

$$x(t) = \frac{A}{2} e^{j\omega_1 t} + \frac{A}{2} e^{j(-\omega_1)t}$$

Dus het **spectrum** van dit signaal ziet er als volgt uit:



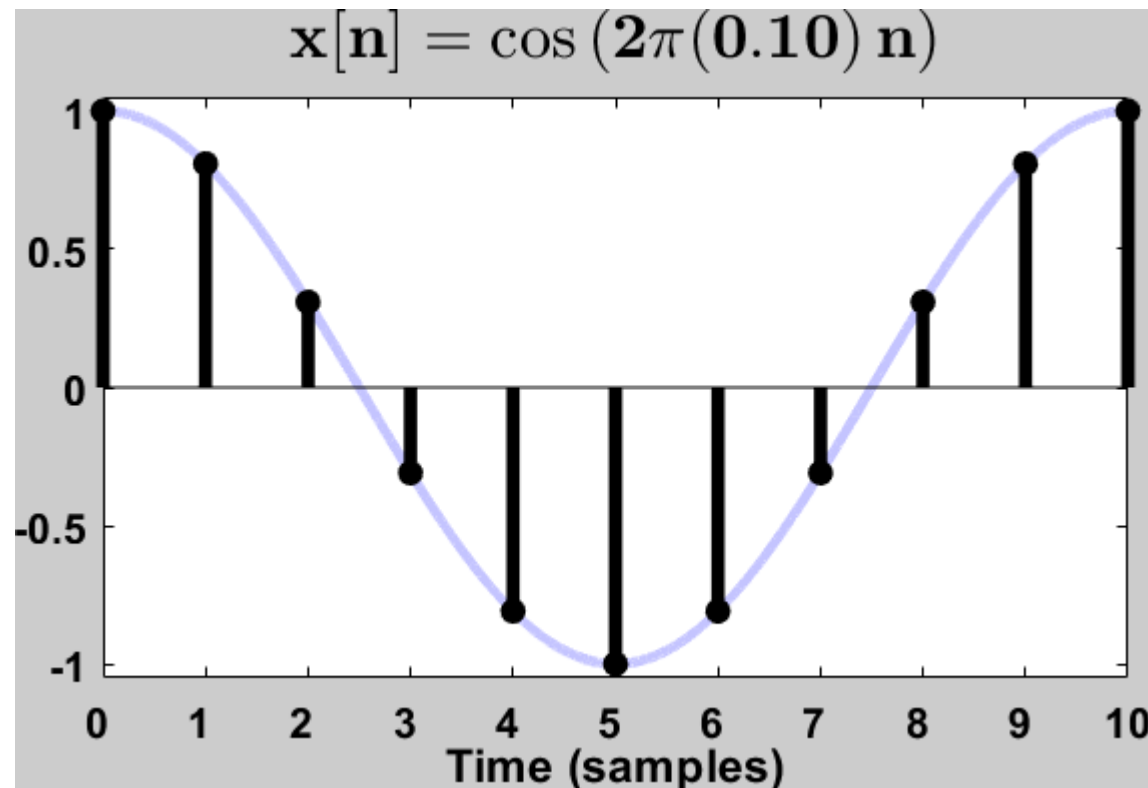
Zie [Rotating Phasors Demo](#)

Replica's

In het spectrum kan je, na sampling, geen onderscheid maken tussen de frequenties:

$$f_1, f_s - f_1, f_s + f_1, 2f_s - f_1, 2f_s + f_1, \dots$$

Deze gelijkwaardige frequenties worden **replica's** genoemd.



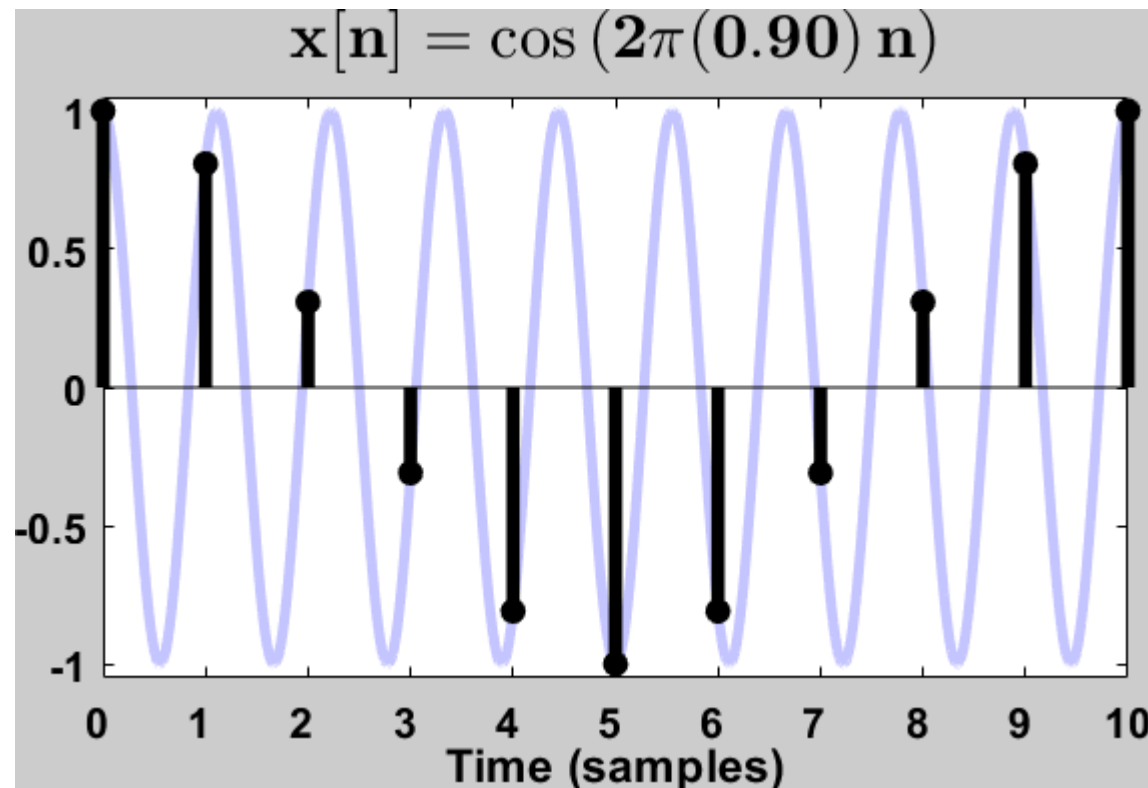
$$f_1 = 2\text{Hz}$$
$$f_s = 20\text{Hz}$$

Replica's

In het spectrum kan je, na sampling, geen onderscheid maken tussen de frequenties:

$$f_1, f_s - f_1, f_s + f_1, 2f_s - f_1, 2f_s + f_1, \dots$$

Deze gelijkwaardige frequenties worden **replica's** genoemd.



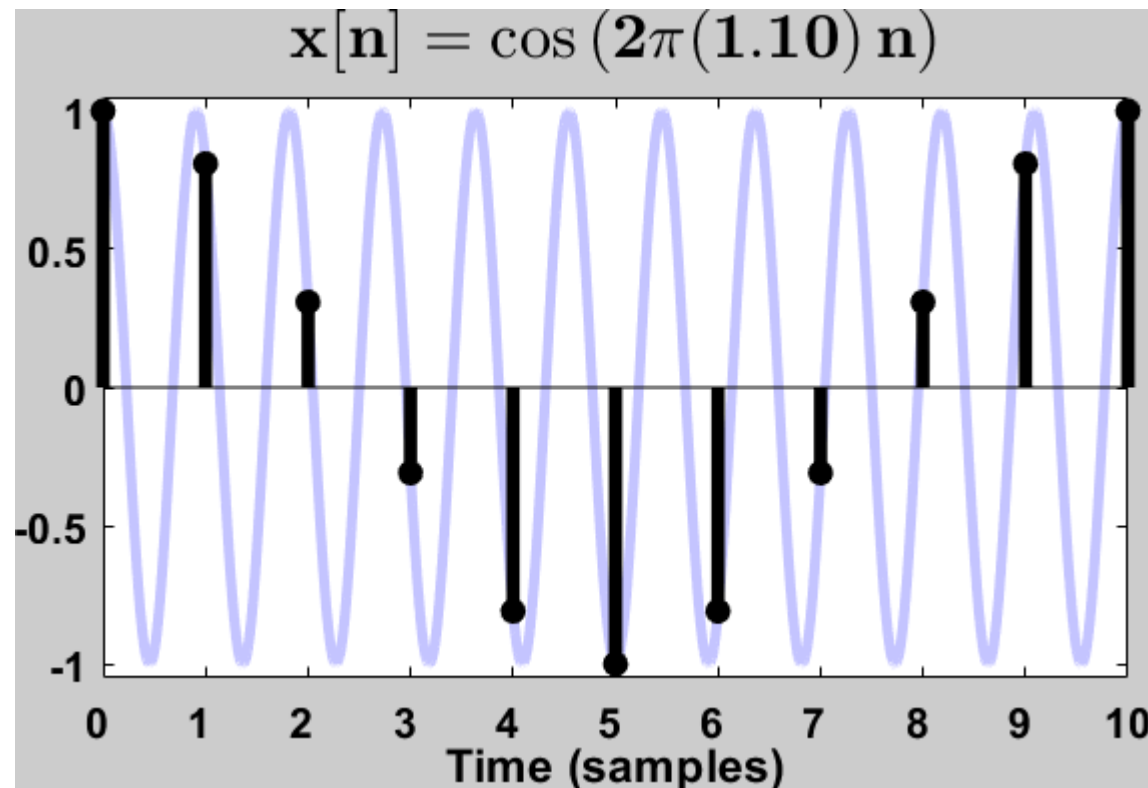
$$f_s - f_1 = 18\text{Hz}$$
$$f_s = 20\text{Hz}$$

Replica's

In het spectrum kan je, na sampling, geen onderscheid maken tussen de frequenties:

$$f_1, f_s - f_1, f_s + f_1, 2f_s - f_1, 2f_s + f_1, \dots$$

Deze gelijkwaardige frequenties worden **replica's** genoemd.



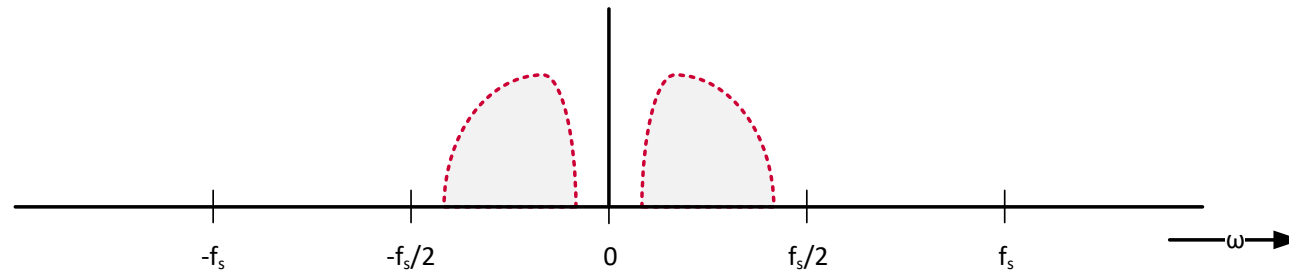
$$f_s + f_1 = 22\text{Hz}$$
$$f_s = 20\text{Hz}$$

Vouwvervorming (Aliasing)

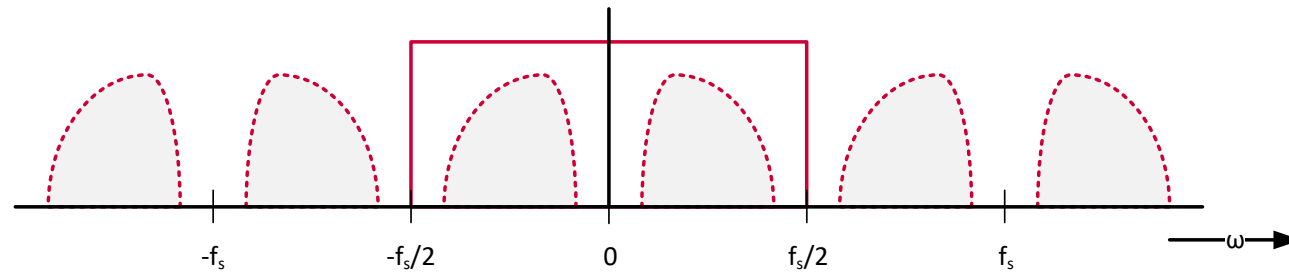
Als de sampling frequent f_s te laag gekozen is, ontstaat er **vouwvervorming** (aliasing):

- De replica's gaan elkaar overlappen.

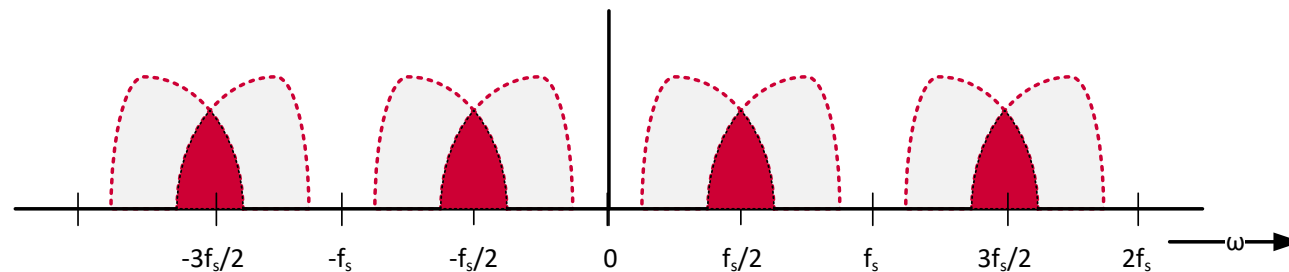
Originele spectrum:



Originele spectrum
met replica's:



Vouwvervorming
(aliasing):



Zie: Continuous-Discrete Sampling Demo, Bron: <http://dspfirst.gatech.edu/matlab/>

Nyquist–Shannon sampling theorema

Om een analoog signaal goed met digitaal signaal te kunnen representeren moet de sample rate voldoende hoog zijn.

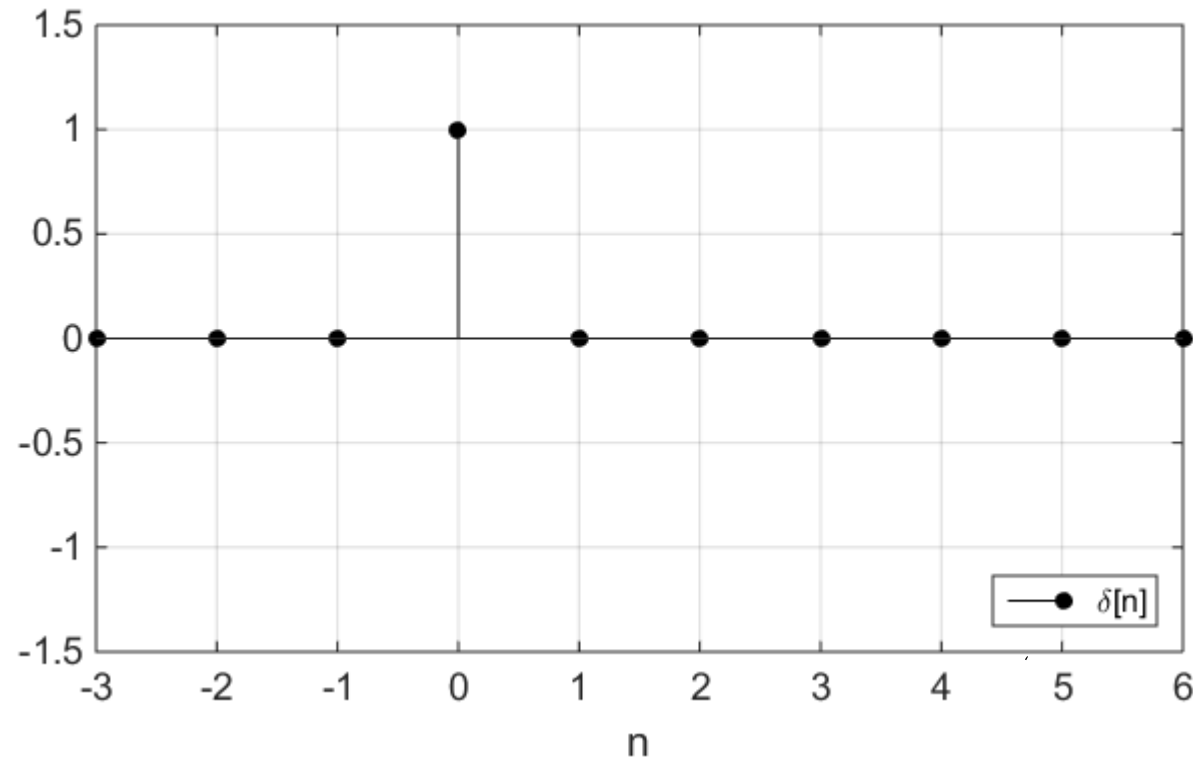
Het Nyquist-Shannon sampling theorema geeft aan wat voldoende hoog betekent:

Nyquist rate :

Hoogste frequentie in een signaal: f_{max}

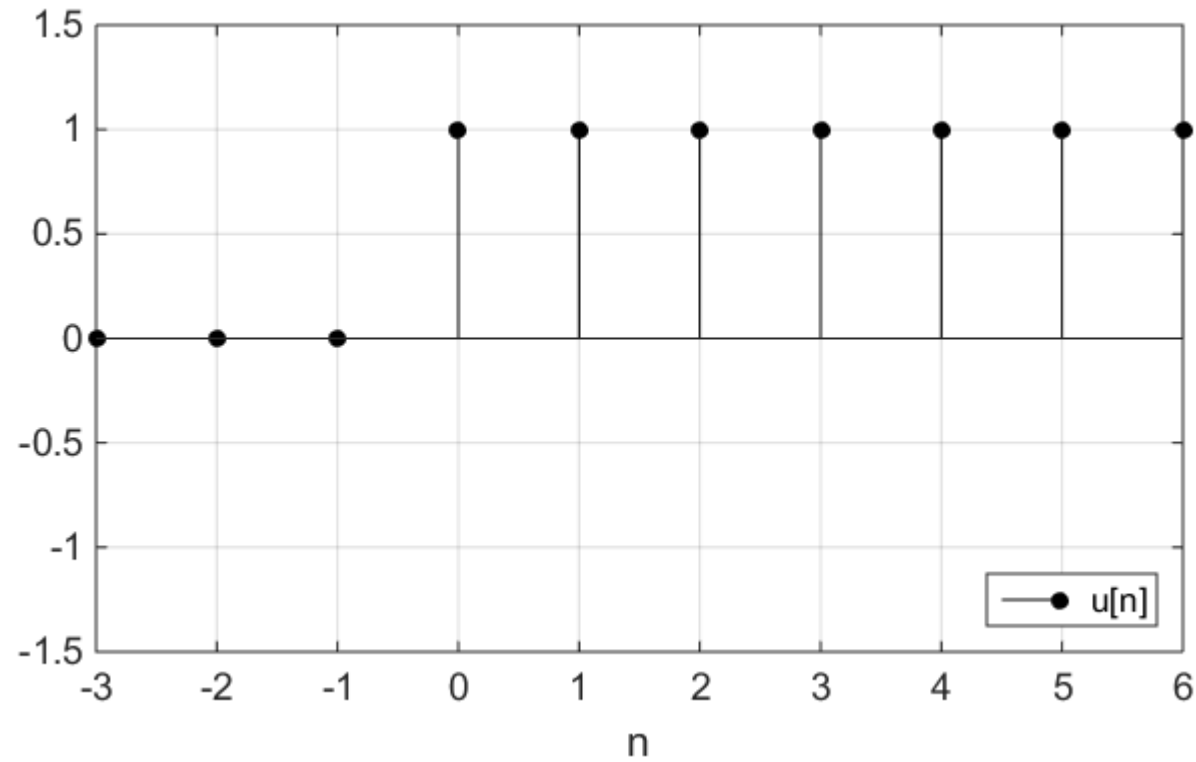
Nyquist rate (minimale sample rate): $f_s = 2 \cdot f_{max}$

Basissignalen: Impuls (impulse)



$$\delta[n] = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$$

Basissignalen: Stap (step)



$$u[n] = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$

Oefening: Teken de volgende signalen ...

A. $x[n] = \delta[n] - 2\delta[n - 1] + \delta[n - 2]$

$$\delta[n] = \begin{cases} 0, & n \neq 0 \\ 1, & n = 0 \end{cases}$$

B. $x[n] = u[n] - 2u[n - 1] + u[n - 2]$

$$u[n] = \begin{cases} 0, & n < 0 \\ 1, & n \geq 0 \end{cases}$$

C. $x[n] = \begin{cases} 0, & n < 0 \\ u[n] - x[n - 1], & n \geq 0 \end{cases}$

D. $x[n] = u[n] - 3\delta[n - 1], \quad -3 \leq n \leq 3$

Oefening: Teken de volgende signalen ...

Gegeven:

$$x[n] = u[n - 1] + \delta[n + 1]$$

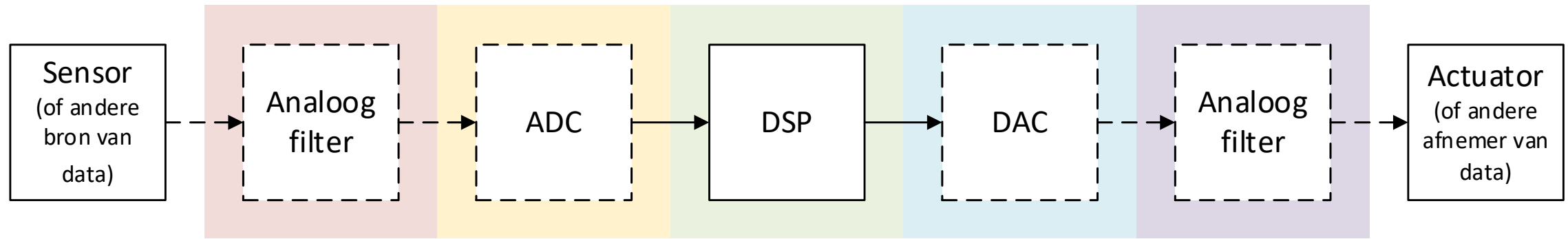
Teken $y[n]$ voor $-3 \leq n \leq 3$ voor:

- A. $y[n] = x[-n]$
- B. $y[n] = x[n + 1] \cdot u[n - 1]$
- C. $y[n] = x[-n] + 2x[n]$

Vorbereiding les 2

- [Huiswerk week 1 les 1](#)
- Opgaven: [1.4, 1.6a, 1.6b en 1.7a tot en met 1.7c](#)
- Lezen (verdieping):
 - §1.1 Het terrein van de digitale signaalbewerking
 - §1.2 Sampling en AD-conversie
 - §1.3 Basisvormen van digitale signalen**(23 pagina's)**

Lineaire tijdsinvariante (LTI) systemen



Bij dit vak beschouwen we alleen DSP-algoritmes die zich gedragen als **lineaire tijdsinvariante (linear time-invariant) systemen**:



- **Lineair:**

$$ax_1[n] \rightarrow ay_1[n]$$

$$x_1[n] + x_2[n] \rightarrow y_1[n] + y_2[n]$$

$$ax_1[n] + bx_2[n] \rightarrow ay_1[n] + by_2[n]$$

- **Tijdsinvariant:**

$$x_1[n - k] \rightarrow y_1[n - k]$$

Oefening: LTI-systemen?

Onderzoek of de volgende relaties LTI-systemen voorstellen:

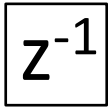
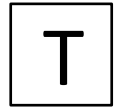
A. $y[n] = y[n - 1] + x[n] + x[n - 2] ,$

B. $y[n] = y[n - 1] + n x[n] ,$

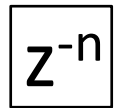
C. $y[n] = y^2[n - 1] + x[n] ,$

D. $y[n] = (x[n] + 1)(x[n - 1] - 1).$

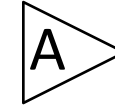
LTI-systeemschema's



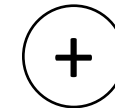
Vertraging van 1 sample



Vertraging van N samples

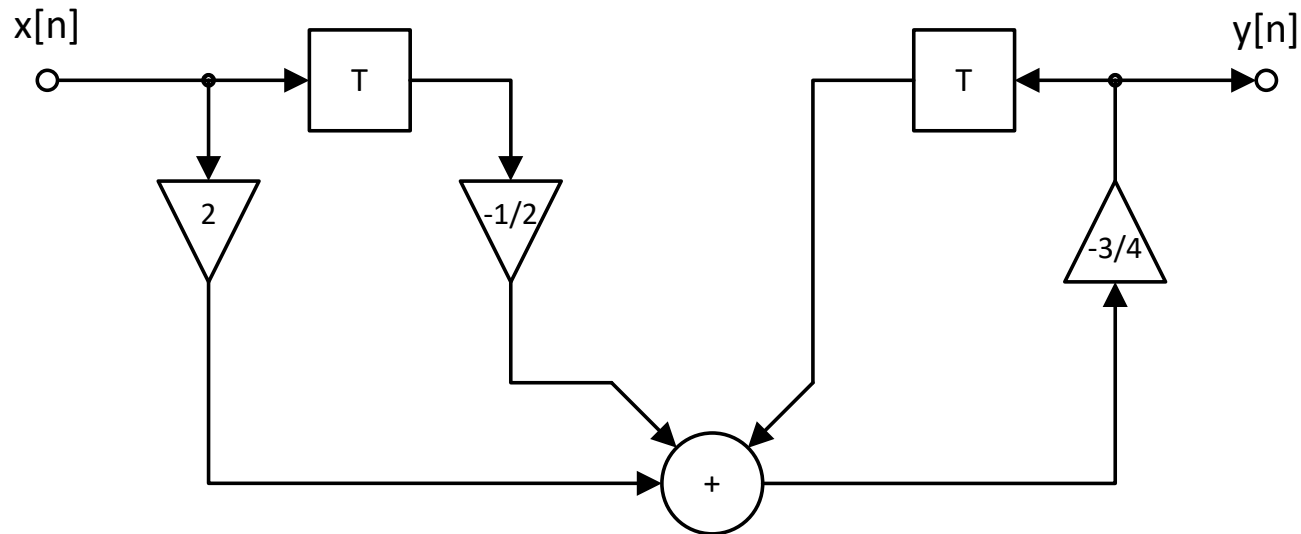


Vermenigvuldiging



Optelling

Voorbeeld:



Bepaal de formule voor $y[n]$?

Overige eigenschappen van LTI-systemen

In dit vak bestuderen we alleen LTI-systemen met de volgende eigenschappen:

Omkeerbaar:

- Uit het uitgangssignaal kan het ingangssignaal éénduidig bepaald worden.

Stabiel:

- Het uitgangssignaal groeit niet onbegrensd.

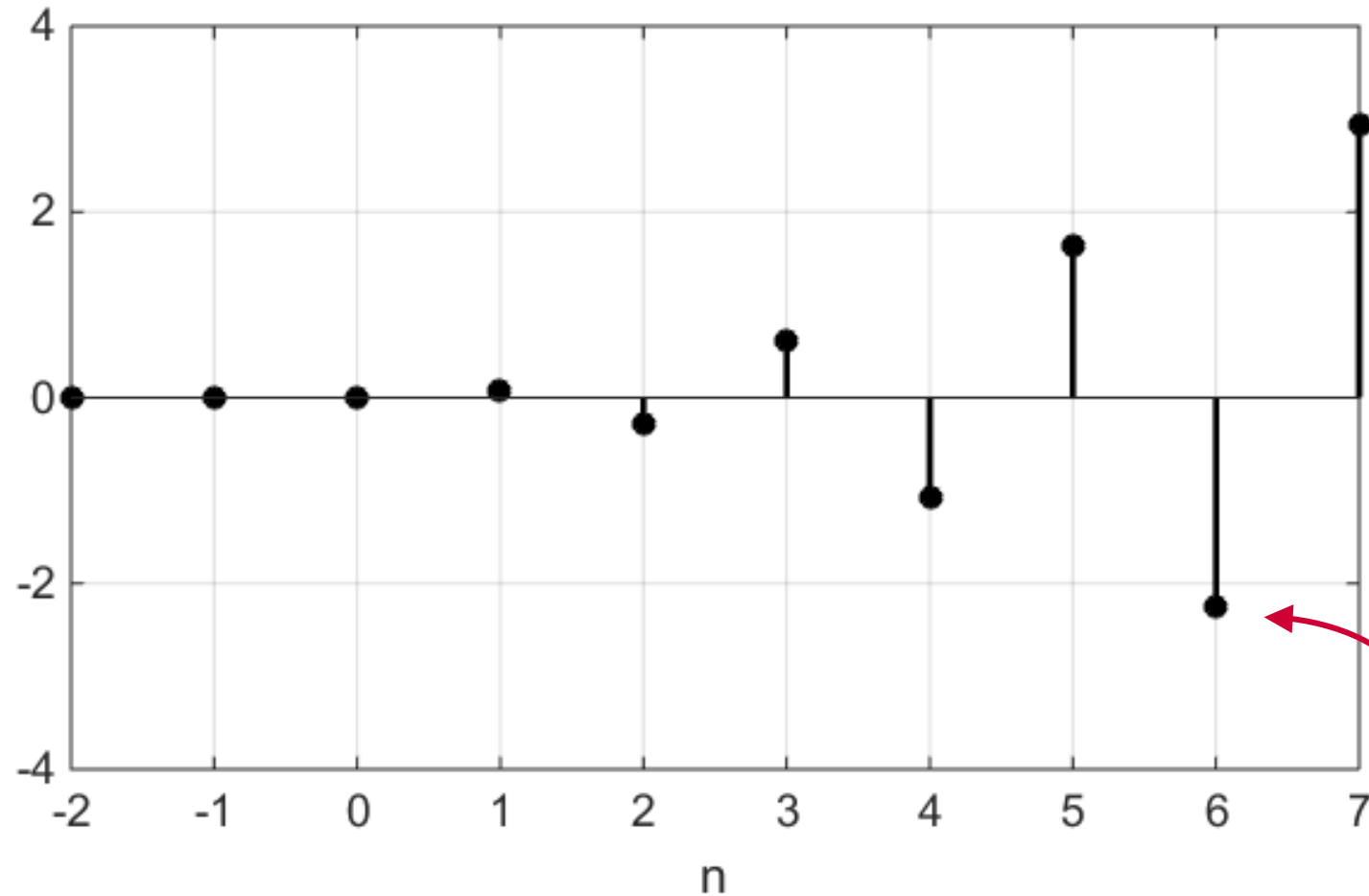
Causaal:

- Het huidige uitgangssignaal hangt alleen van huidige en voorafgaande ingangssignalen af en mogelijk van voorafgaande uitgangssignalen.

Geheugen:

- Het huidige uitgangssignaal hangt van voorafgaande in- en/of uitgangssignalen af.

Instabiliteit



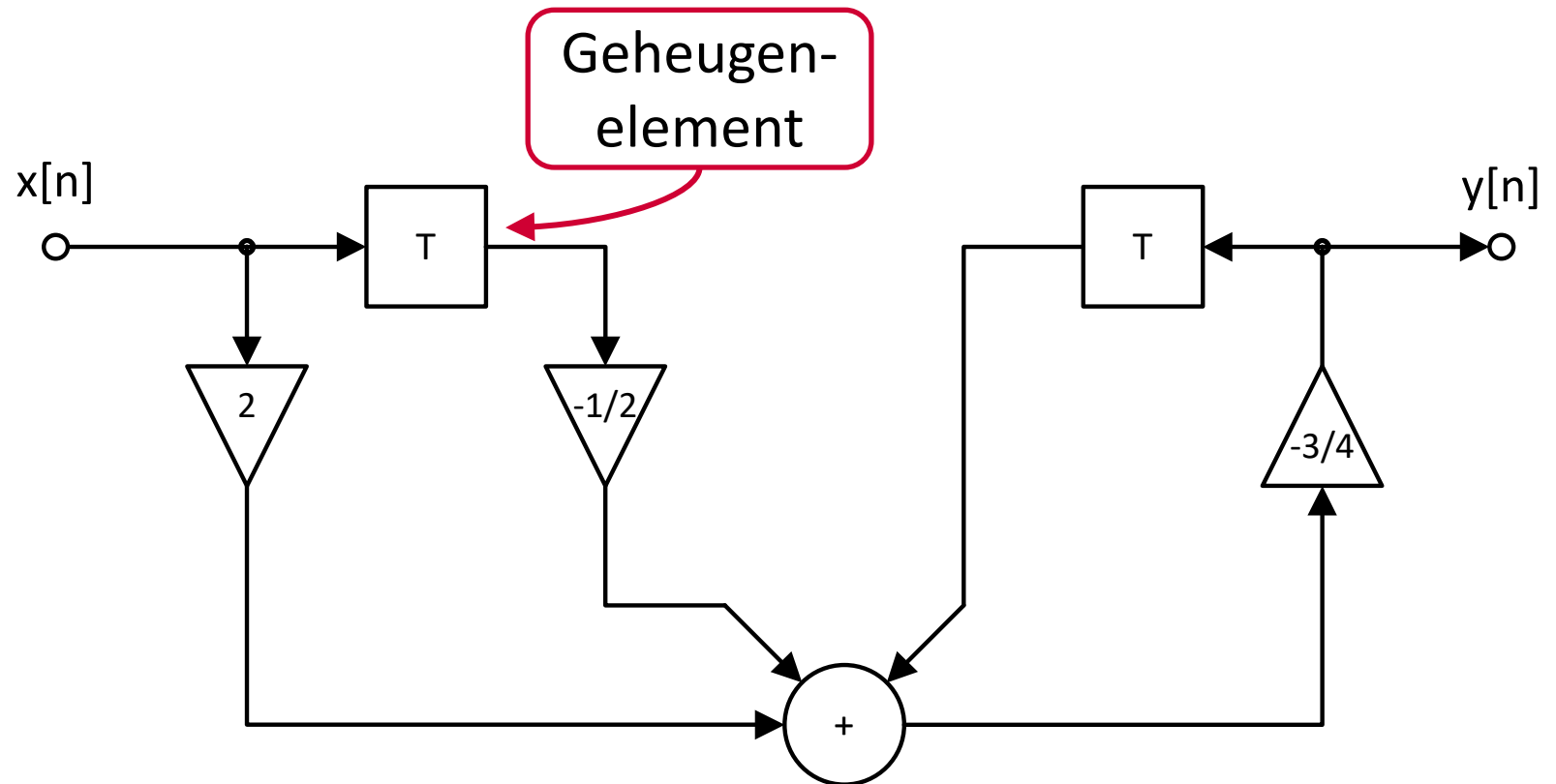
Als de output
maar blijft
groeien, dan is
het systeem niet
stabiel!

Causaliteit en geheugen

Niet causaal: $y[n] = y[n + 1] + x[n + 2]$

De huidige uitgang hangt van toekomstige ingangs- en/of uitgangssignalen af.

Geheugen:



Wat hebben we gezien deze week?

- Typisch DSP-systeem
- Sampling
 - Nyquist / Shannon
 - Aliasing
- Basissignalen
- Eigenschappen LTI-systemen

Vorbereiding week 2

- [Huiswerk week 1 les 2](#)
- Opgaven: [1.10, 1.13, 1.14](#)
- Doornemen presentatie week 1
- Lezen (verdieping):
 - §1.5 Digitale processoren
(6 pagina's)
- [Oefenopgaven WK1.pdf](#)
 - [Oefenopgaven WK1 uitwerkingen.pdf](#)