

Reader Meten Regelen en Besturen

TCTI-V2MRB-14

Inhoud

Introductie.....	3
Sensoren.....	5
Sensoren – voorbeeld databladen.....	11
Bemonsteren (sampling) & Aliasing	12
Tijd / Frequentie domein	14
Digitaal filteren	18
PID control.....	20
Motoren	24
OpAmp	25
ADC	28
Hardware matig filteren	29
Soorten verwerkingseenheid	33

Revisie historie

1.0	startdocument
2.0	Diverse aanpassingen

Introductie

Deze reader voor TCTI-V2MRB-14 is het basisdocument voor het vak. De stof in dit document en in de sheets van het college moet je kennen voor het voor het tentamen. De sheets dekken een deel de inhoud van deze reader af en geven extra informatie. Deze toegevoegde informatie moet je ook kennen. De tentamenstof voor dit vak bestaat dus uit de inhoud van deze reader en de sheets uit het college.

Dit document is blijvend in ontwikkeling. Controleer op Sharepoint of je de laatste versie hebt.

Basis

(bron: wikipedia)

Er wordt verondersteld dat de student kennis heeft van (en kan toepassen):

- De wet van Ohm

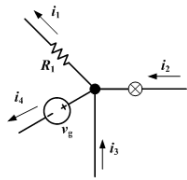
$$U = I \cdot R$$

- Stroom wet van Kirchhoff:

In elk knooppunt is de som van de stromen (waarbij ingaande stromen positief en uitgaande negatief worden genomen) gelijk aan nul:

$$\sum i_i = 0$$

Het knooppunt kan stroom opslaan noch afgeven

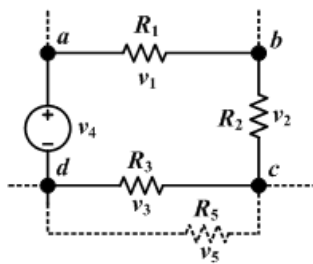


$$-i_1 + i_2 + i_3 - i_4 = 0$$

- Spanningswet van Kirchhoff:

De som van de elektrische potentiaalverschillen (rekening houdend met de richting) in elke gesloten lus in een kring is gelijk aan nul.

$$\sum u_i = 0$$



$$u_1 + u_2 + u_3 + u_4 = 0$$

(let op! De richting van de spanning consequent meenemen!)

- Omzetten decimale getallen naar binair en vica versa:

Om een binair getal te vertalen naar een decimaal getal, hoeft men slechts te kijken naar de posities waar een 1 staat. Voor ieder binair cijfer 1 berekent men de door de positie van dit cijfer aangegeven macht van twee, en wel: $2^{\text{positie} - 1}$. De som van de op deze wijze berekende reeks decimale getallen geeft de waarde van het binaire getal decimaal weer. De eerste positie is de meest rechtse en komt overeen met het getal 1. De tweede positie, de tweede van rechts, komt overeen met het getal 2, de derde van rechts met 4, enz.

Binair	$2^{(\text{positie van de 1}) - 1}$	Decimaal	Binair	$2^{(\text{positie van de 1}) - 1}$	Decimaal
100000	2^5	32			
010000	2^4	16	010000	2^4	16
001000	2^3	8			
000100	2^2	4	000100	2^2	4
000010	2^1	2			
000001	2^0	1	000001	2^0	1
111111	$2^5 + 2^4 + 2^3 + 2^2 + 2^1 + 2^0$	63	010101	$2^4 + 2^2 + 2^0$	21

Simpel gezegd: Bereken voor elk cijfer 1 in het binaire getal, de overeenkomstige macht van 2. Een binair getal van 6 cijfers, bijvoorbeeld 111111, wordt vertaald in (van links naar rechts) 32, 16, 8, 4, 2 en 1. De som $32 + 16 + 8 + 4 + 2 + 1 = 63$ is de decimale waarde van dit binaire getal. Zo wordt 010101 $16 + 4 + 1 = 21$ in decimale waarden.

Een besturingssysteem

Een besturingssysteem (een systeem waarmee je iets kan besturen of regelen) bestaat in de meeste gevallen uit 5 delen.

Sensor

Interface^I

Verwerking

Interface^O

Actuator

Zoals je in bovenstaand figuur kunt zien bestaat het uit 1 of meerdere sensoren, een interface, een verwerkingseenheid, weer een interface en 1 of meerdere actuatoren.

Sensoren

- Een sensor zet een fysische grootte (bv. druk, temperatuur, debiet) om naar een elektrisch signaal. Een sensor wordt gekarakteriseerd door een aantal eigenschappen (bv. nauwkeurigheid, uitgangsweerstand, hysteresis). Zie ook het hoofdstuk sensoren.

Interface^I

- Het signaal wat van de sensor afkomstig is zal vaak eerst versterkt moeten worden (als bijvoorbeeld de spanning te laag is). Dit kan met een OpAmp (operational amplifier) gebeuren (zie hoofdstuk OpAmp).
- Tevens zal het signaal vaak eerst gefilterd moeten worden (zie hoofdstuk Filteren) om bv. aliasing te voorkomen (zie hoofdstuk Bemonsteren & Aliasing).
- Het signaal van de sensor is een analoge waarde (sensoren kunnen ook uitgerust worden met een digitale interface, maar die sensoren worden binnen deze reader buiten beschouwing gelaten). Om de verwerkingseenheid met het signaal te kunnen laten rekenen zal het signaal eerst omgezet moeten worden naar het digitale domein. Hiervoor dient een ADC (analoog naar digitaal converter). (zie hoofdstuk ADC).

Verwerking

- De verwerkingseenheid gebruikt de meetgegevens van de sensoren om de juiste waarden naar de actuator(en) te sturen. De verwerkingseenheid kan uit verschillende soorten "computers" bestaan (zie hoofdstuk Soorten verwerkingseenheid).
- De verwerkingseenheid kan onder andere gebruikt worden om een systeem te regelen (zie hoofdstuk PID control).
- Tevens kan de verwerkingseenheid gebruikt worden om naar het signaal in het frequentie domein te kijken (zie hoofdstuk Tijd – Frequentie domein).

Interface^O

- Om de actuator aan te kunnen sturen zal vaak het signaal eerst moeten worden omgezet van digitaal naar analoog (zie hoofdstuk DAC) en vervolgens nog gefilterd worden (zie hoofdstuk Filteren).
- In het geval van een motor aansturing wordt vaak PWM (pulse width modulation) toegepast (zie hoofdstuk PWM & H-brug).

Actuatoren

- In deze reader wordt alleen de motor behandeld. Zie hoofdstuk: Motoren.

Sensoren

Een sensor zet een fysische grootheid om naar een elektrisch signaal. Sensoren kunnen een digitale of een analoge uitgang hebben. Bij beide typen is het sensing element (het element wat de meting echt doet) vaak nog gewoon analoog. Bij een digitale sensor echter zit er op de sensor een microcontroller die het signaal vertaalt naar een digitaal signaal.

Eerst zullen de voor en nadelen van digitale sensoren worden behandeld. In de daaropvolgende paragrafen worden de kenmerken van een sensor behandeld.

Analoge sensoren

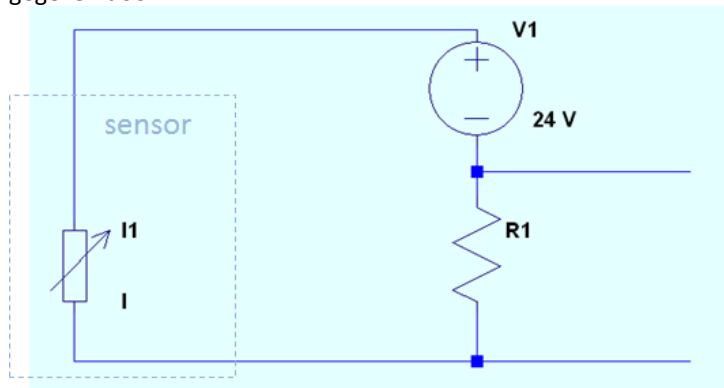
De uitgang van een analoge sensor is te verkrijgen in 3 smaken:

- 1) puur de fysieke meetwaarde. Dus bijvoorbeeld een weerstand.
- 2) een 0-5 volt signaal (of 0-10 volt).
- 3) een 4-20mA signaal.

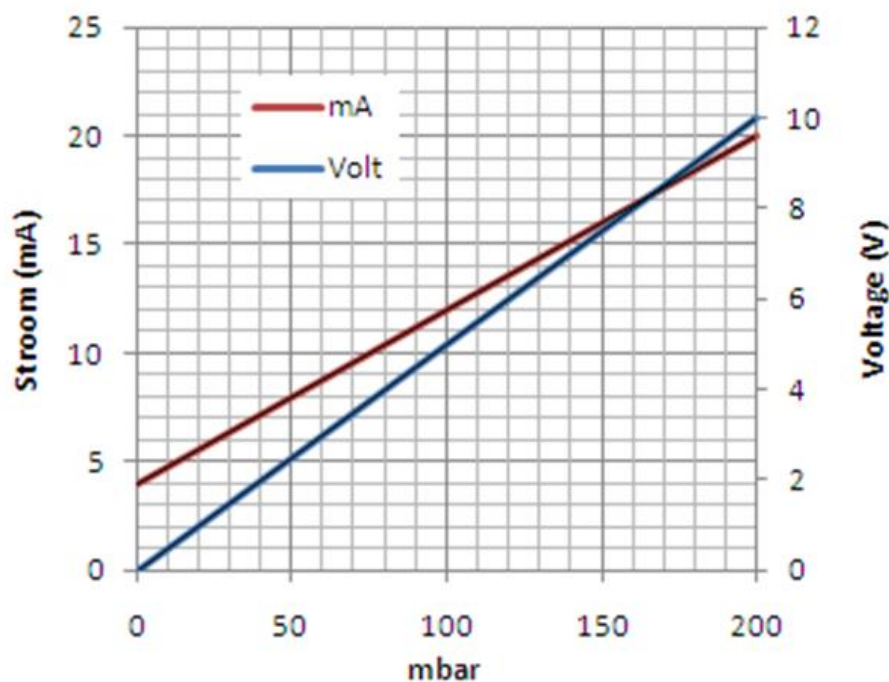
Voor het eerste type zijn niet of nauwelijks standaarden beschikbaar. Er zal dus voor elk soort een specifieke meetversterker moeten worden gebouwd.

Voor het tweede type zijn standaard units verkrijgbaar (bv een ADC met een 0-5 volt ingang).

Ook voor het derde type zijn standaard units verkrijgbaar. Het schema voor het uitlezen van de sensor wordt gegeven door:



Zoals te zien is regelt de sensor zijn stroom die hij afneemt. De hoeveelheid stroom die hij afneemt geeft aan wat zijn meetwaarde is. 4 mA is daarbij de minimale waarde en 20mA de maximale.



Een 4-20mA uitgang heeft een aantal bijzondere eigenschappen:

- de sensor kan eventueel met de meetstroom gevoed worden zodat geen aparte voeding meer nodig is.
- er kan makkelijk gedetecteerd worden of de sensor nog aangesloten is (aangezien de stroom naar 0 zal gaan).
- ook een kortgesloten sensor zal snel te detecteren zijn.
- de aansluitdraden kunnen een stuk langer zijn aangezien een eventuele spanningsval over de draden de meting niet zal beïnvloeden.
- de meting is relatief ongevoelig voor een storing op de aansluitdraden (veel ongevoeliger dan bij een 0-10 V systeem).

Nadelen zijn:

- 4-20 mA is niet echt low power. Tegenwoordig zijn er sensoren die veel minder stroom nodig hebben om te werken.
- de ontvanger moet nauwkeurig stroom kunnen meten, wat een stuk moeilijker is dan spanning meten. In de praktijk wordt de stroom via een weerstand omgezet naar een spanning. Eventuele variaties van de weerstand beïnvloeden de uitkomst van de meting.

Als laatste type is er nog de Hart interface. Dit is een mengvorm tussen een analoge en digitale interface. Het digitale signaal wordt gesuperponeerd op het analoge signaal met een 1200 en 2200 Hz signaal.

Door het plaatsen van een laag doorlaat filter kan een Hart sensor zonder verdere aanpassingen in een ouderwetse (legacy) fabriek gebruikt worden. In de digitale modus kunnen Hart sensoren doorgelust worden waarbij elke sensor vast 4 mA verbruikt en de meetwaarden enkel digitaal worden verzonden.

Digitale of “smart” sensoren

Digitale sensoren hebben een digitale interface. De omzetting van de grootte naar de digitale waarde gebeurt in de sensor. Hij heeft hiervoor een micro controller aan boord

Het nadeel van de digitale sensor is vaak een hogere aanschafsprijs.

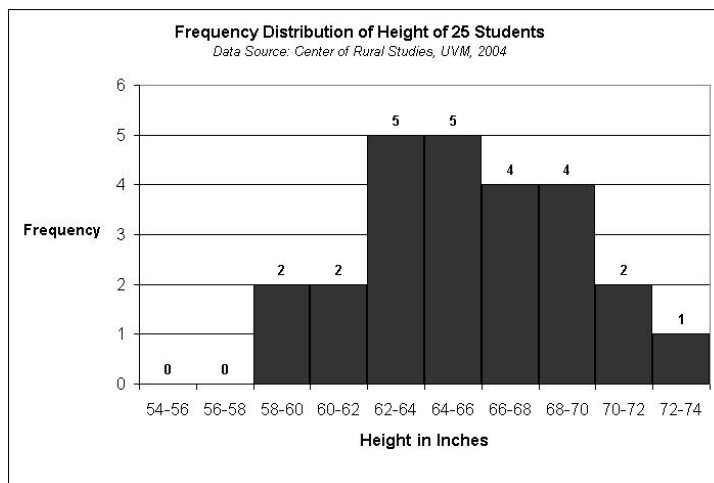
De voordelen zijn:

- Interoperability
 - uitwisselen van sensoren van verschillende fabrikanten
- robuustheid
 - ruis ongevoelig, betrouwbaar
- kabel efficiëntie/reductie
 - bus systemen
- remote configureerbaar
 - instellen van range van sensoren, via centraal systeem
- maintainability
 - makelijker onderhoud
- sensor lifetime prediction
 - sensor meld actief dat hij bijna stuk is, of bijna leeg is.
- asset management
 - Welke sensor zit waar? Is hij al gecalibreerd?

Histogram

Om de meetwaarden van een sensor beter te kunnen begrijpen moeten we eerst het begrip histogram behandelen.

Bij een histogram wordt op de x-as de gemeten waarden weergegeven. Op de y-as wordt aangegeven hoe vaak die waarde voorkomt. Dit kan in absolute aantallen, maar ook als percentage of fractie worden aangegeven.



Voorbeeld histogram

In bovenstaand voorbeeld zijn er dus 5 studenten met de lengte tussen 64 en 66 inches.

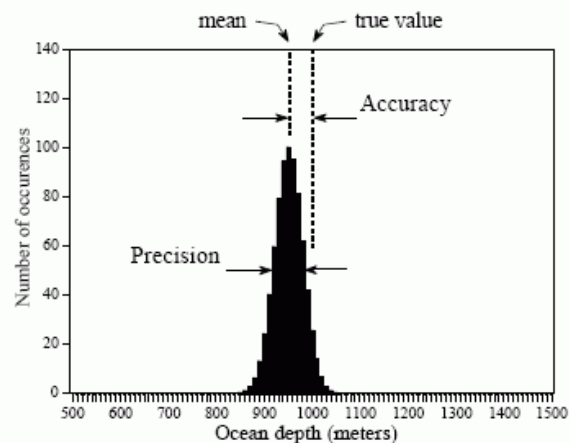
Type fouten

Een sensor heeft altijd een fout in zijn meting.

Hieronder zie je een meting van een sensor die op dezelfde plek de diepte van de zee heeft proberen te meten. Ideaal gezien zou daar natuurlijk 1 getal uit moeten komen.

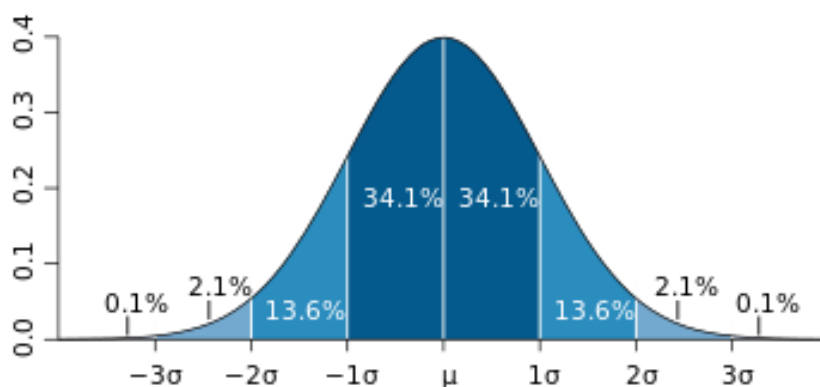
FIGURE 2-11

Definitions of accuracy and precision. Accuracy is the difference between the true value and the mean of the under-lying process that generates the data. Precision is the spread of the values, specified by the standard deviation, the signal-to-noise ratio, or the CV.



Zoals je kunt zien zit er een spreiding in de meet resultaten. Aangezien op dezelfde plek gemeten is, moet de fout wel in de meting zitten!

Indien nu met een sensor meerdere malen hetzelfde ding wordt gemeten zal er bijna altijd een normaal verdeling uit volgen (zoals ook te zien was op de vorige pagina).



De normaal verdeling wordt gekenmerkt door 2 kengetallen: μ (het gemiddelde) en σ (de standaard deviatie). Een ideale sensor zal een standaarddeviatie van nul hebben en zal dus steeds hetzelfde uitkomst (μ) geven als je de meting herhaalt.

In de praktijk echter zal de meetwaarde schommelen rond de μ . Soms zal hij er boven zitten, soms er onder.

Hoe gebruik je nu de normaal verdeling grafiek?

Zoals je in de grafiek kunt lezen zal een sensor met een gemiddelde van 10 en een standaard deviatie van 1 in 34,1% van de metingen een uitkomst tussen de 9 en de 10 hebben.

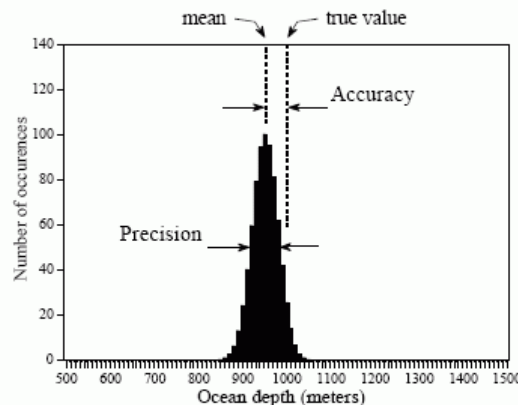
Wil je bij een sensor met een standaard deviatie van 2, voor 95.4% ($34,1 \times 2 + 13,6 \times 2$) zeker zijn dat je meting klopt zul je dus een onnauwkeurigheid van +4 en -4 moeten accepteren.

Voor de wiskundigen onder ons: de normaal verdeling is het gevolg van een optelling van onderling ongekoppelde fouten. Voor het merendeel van de sensoren geldt dat er onderling ongekoppelde fouten in de meting zitten. Vandaar dat er bij de meeste sensoren een normaal verdeling uit komt.

Precisie (precision /repeatability)

Wanneer we weer terug gaan naar de zee diepte meting zien we dat precisie (de spreiding in de meting) bepaalt wordt door de willekeurige (of toevallige) fouten in de meting.

FIGURE 2-11
Definitions of accuracy and precision. Accuracy is the difference between the true value and the mean of the under-lying process that generates the data. Precision is the spread of the values, specified by the standard deviation, the signal-to-noise ratio, or the CV.



Precisie wordt normaal uitgedrukt in standaarddeviatie (σ). Men geeft precisie ook vaak aan als “repeatability”.

Accuraatheid (accuracy)

Zelfs als de meting elke keer precies dezelfde waarde geeft (als de st.dev. σ dus nul is) kan er nog steeds een fout in de meting zitten. In de zee grafiek zie je dat de echte diepte van de zee (true value) verschilt van het gemeten gemiddelde. De sensor heeft dus een *systematische* afwijking (de sensor heeft een afwijking, maar die is wel elke keer hetzelfde).

Samenvattend is de accuraatheid van de meting gekoppeld aan systematische fouten terwijl precisie (of nauwkeurigheid) gekoppeld is aan toevallige fouten.

Drift

Een sensor kan ook afwijking vertonen als gevolg van drift. Er zijn twee grote oorzaken van drift:

- 1) verloop van de sensor als gevolg van veroudering.
- 2) verloop van de sensor als gevolg van temperatuur invloed.

Precisie & Accuraatheid & Drift

Bovenstaande fouten (precisie, accuraatheid, drift) kunnen op verschillende manieren worden aangegeven. Ze kunnen als absolute waarden worden aangegeven (bv accuraatheid van de sensor is $\pm 3^\circ\text{C}$), ze kunnen als percentage van de fullScale reading (het bereik van de sensor) worden aangegeven of ze kunnen als percentage van de meting worden aangegeven.

Stel een sensor heeft een bereik van 150°C .

	Fout aangegeven als:
--	----------------------

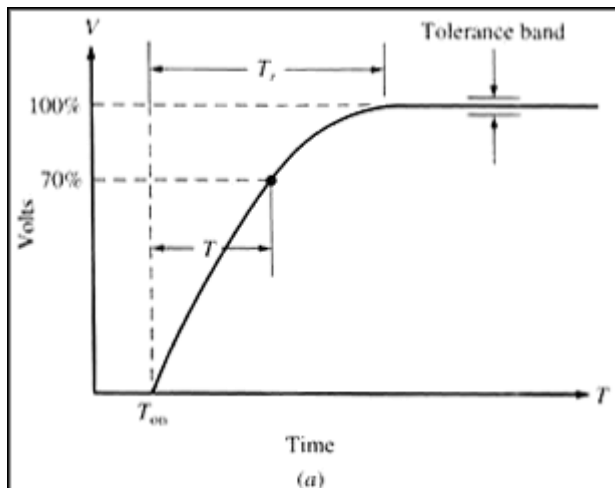
Precisie	$\pm 3^\circ\text{C}$
Accuraatheid	$\pm 1\% \text{ FS}$
Drift	$\pm 2\% \text{ of reading / year}$

Als er een temperatuur van 25 graden gemeten wordt dan zijn de verschillende fouten voor die sensor gegeven door:

Precisie	$\pm 3^\circ\text{C}$
Accuraatheid	$\pm 1.5^\circ\text{C}$
Drift	$\pm 0.5^\circ\text{C / jaar}$

Responsie tijd

Indien het te meten object plotseling van waarde veranderd (stel de temperatuur wordt 10 graden hoger) dan heeft een sensor vaak even tijd nodig voordat hij de goede waarde aangeeft.

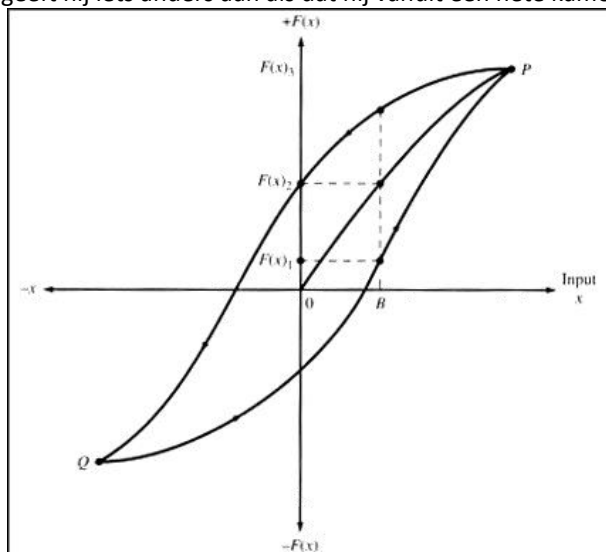


Dit geeft men aan met de responsietijd van een sensor. De responsie tijd wordt meestal gegeven als de $T_{90\%}$ tijd. Dus dat is de tijd die de sensor nodig heeft voordat hij op 90% van de echte waarde zit. $T_{70\%}$ of $T_{95\%}$ (of andere percentages) komt men ook tegen.

Hysteresis

Een sensor kan ook last hebben van hysteresis. Bij hysteresis geeft een sensor een andere waarde afhankelijk van wat zijn vorige meetwaarde was.

Stel dat een temperatuur sensor hysteresis heeft dan houdt dat in dat de temperatuur die hij aangeeft afhangt van uit welke kamer hij gebracht was. Stel dat hij uit een koude kamer gebracht wordt naar de meetkamer dan geeft hij iets anders aan als dat hij vanuit een hete kamer gebracht zou zijn.

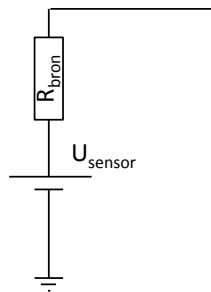


Een voorbeeld van hysteresis.

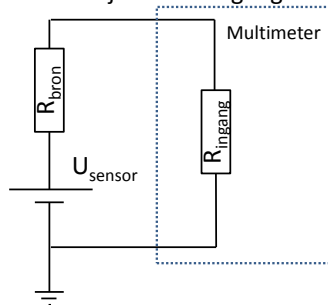
Hysteresis ziet men vaak bij mechanische sensoren waarbij er speling in de sensor zit. Afhankelijk van welke richting de sensor wordt bewogen druk je dan de speling de ene of de andere kant op.

Bronweerstand

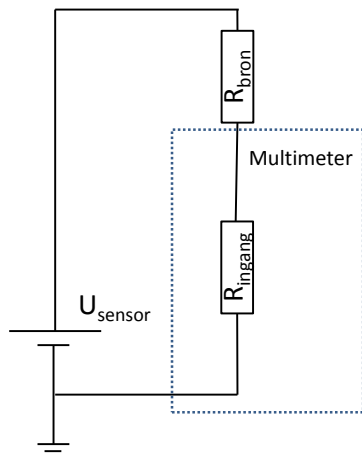
De analoge sensoren kunnen in de meerderheid van de gevallen gekarakteriseerd worden als een spanningsbron met een bronweerstand:



Indien er nu een multimeter (meetapparaat) op wordt aangesloten is de spanning die gemeten wordt afhankelijk van de ingangswaarde van de multimeter.



In feite wordt er een spanningsdeler gecreëerd:



waarin de gemeten spanning gegeven wordt door:

$$U_{\text{gemeten}} = \frac{R_{\text{ingang}}}{R_{\text{bron}} + R_{\text{ingang}}} \cdot U_{\text{sensor}}$$

Zolang R_{ingang} veel groter ($>1000\times$) is dan R_{bron} kan de fout als gevolg van R_{bron} verwaarloosd worden. Indien R_{ingang} echter niet veel groter is dan R_{bron} is de fout niet te verwaarlozen.

Een voorbeeld: stel dat R_{ingang} 10 x zo groot is als R_{bron} . Dan bedraagt de meting 10/11 van het echte signaal. Dit is een fout van 9%! Een remedie hiertegen is het signaal eerst te versterken met een opamp (zie hoofdstuk opamp).

Sensoren – voorbeeld databladen

Zuurstof meter:

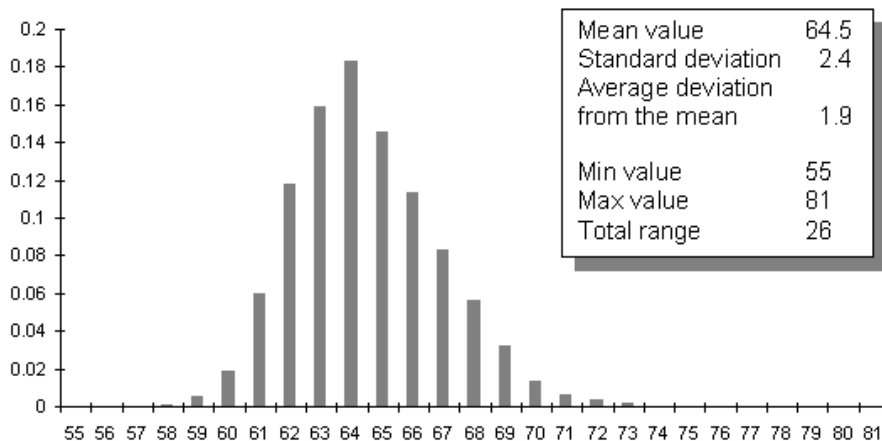
Range:	0 to 100% Oxygen
Display Accuracy:	0.1%
Electrical Connector:	Battery Operated
Operating Temp:	0 to 50° C
Output Signal:	Oxygen in %
Sensor Type:	Galvanic (Electro Chemical)
Response Time:	<5 Sec. to T90
Accuracy:	±1% at full scale
Repeatability:	± 1% of signal
Temperature Compensation:	Compensated
Maximum Storage Temp.	-20 to 60°C
Warranty:	12 Months
Expected life in room air:	18 months

Zuurstofmeter:

Specifications for Low to Ultra-Low Values

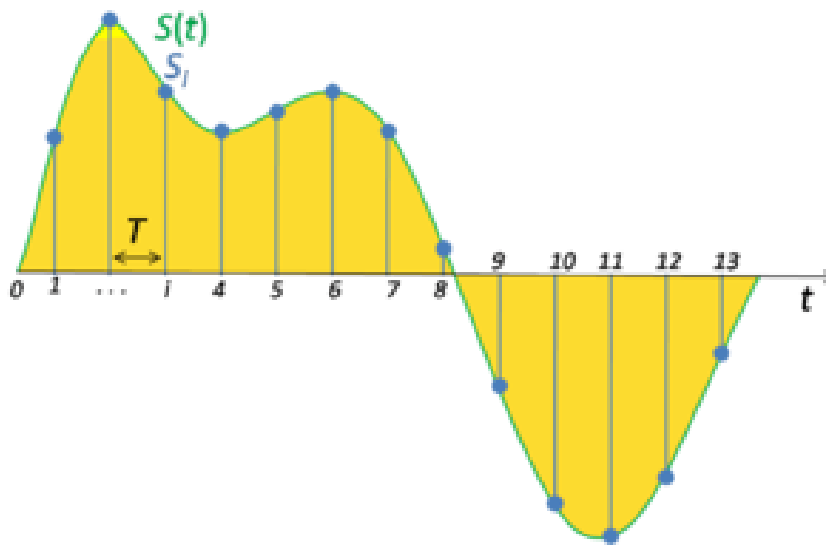
Model (Nominal Ω)	Initial ¹ Tolerance \pm ppm	Stability (\pm ppm) ²		Maximum Limits		Temperature Coefficient \pm ppm/°C	Voltage ⁴ Coefficient \pm ppm/V _{dc}
		1 st Year ³	2 nd Year or 12 Months	mAmps	Voltage		
9334A-1 μ	500	250	50	100A	0.0001	50	-
9334A-10 μ	200	100	25	50A	0.0005	25	-
9334A-100 μ	50	25	15	20A	0.002	8	-
9334A-0.001	20	15	10	6A	0.006	1.5	-
9334A-0.01	10	10	5	3A	0.03	0.5	-

Afstandsmeter:



Bemonsteren (sampling) & Aliasing

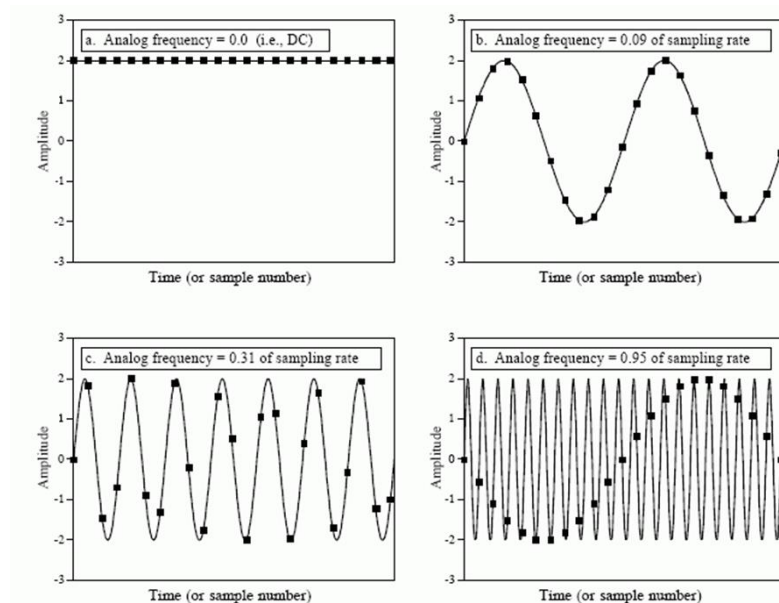
Met een ADC kan een analogoog signaal in een digitaal signaal worden omgezet.



Door met een vaste tussentijd steeds een sample (monster) van een signaal te nemen kan het analoge signaal omgezet worden in een rijtje getallen. De tijd tussen 2 samples (in de grafiek aangegeven met T) wordt de sample tijd genoemd. De sample frequentie (hoe vaak er gesampled wordt per seconde) wordt gegeven door:

$$f_{\text{sample}} = \frac{1}{t_{\text{sample}}}$$

De sample frequentie moet hoog genoeg zijn om alle veranderingen van het signaal te kunnen volgen.



Zoals hierboven te zien is, is de sample frequentie in grafiek a en b hoog genoeg. In grafiek c is het kantje boord. In grafiek d is hij duidelijk te traag. Zoals te zien is in grafiek d lijkt het net of er een “spook” sinus gemeten wordt. Dit effect noemt men aliasing.



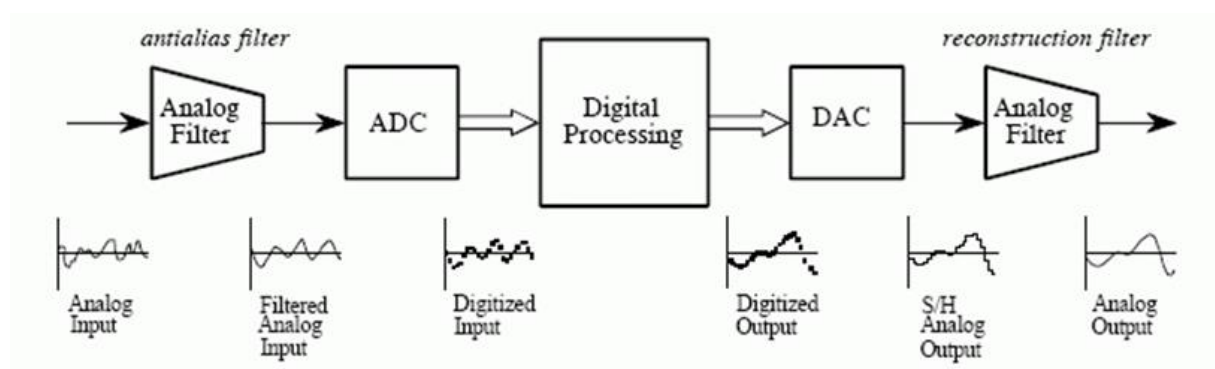
Ook in bovenstaande foto treedt er aliasing op. De grote rimpels in het flatgebouw zijn niet echt maar het gevolg van een te lage samplefrequentie.

Harry Nyquist heeft in 1924 een theoretische grens bepaald waarboven de sample frequentie moet zitten:

$$f_{\text{sample}} > 2 \cdot f_{\text{signaal}}$$

Dus de sample frequentie moet minimaal 2 x zo hoog zijn als de hoogste frequentie die in het signaal zit (dus er moeten minimaal meer dan 2 samples per sinus genomen worden).

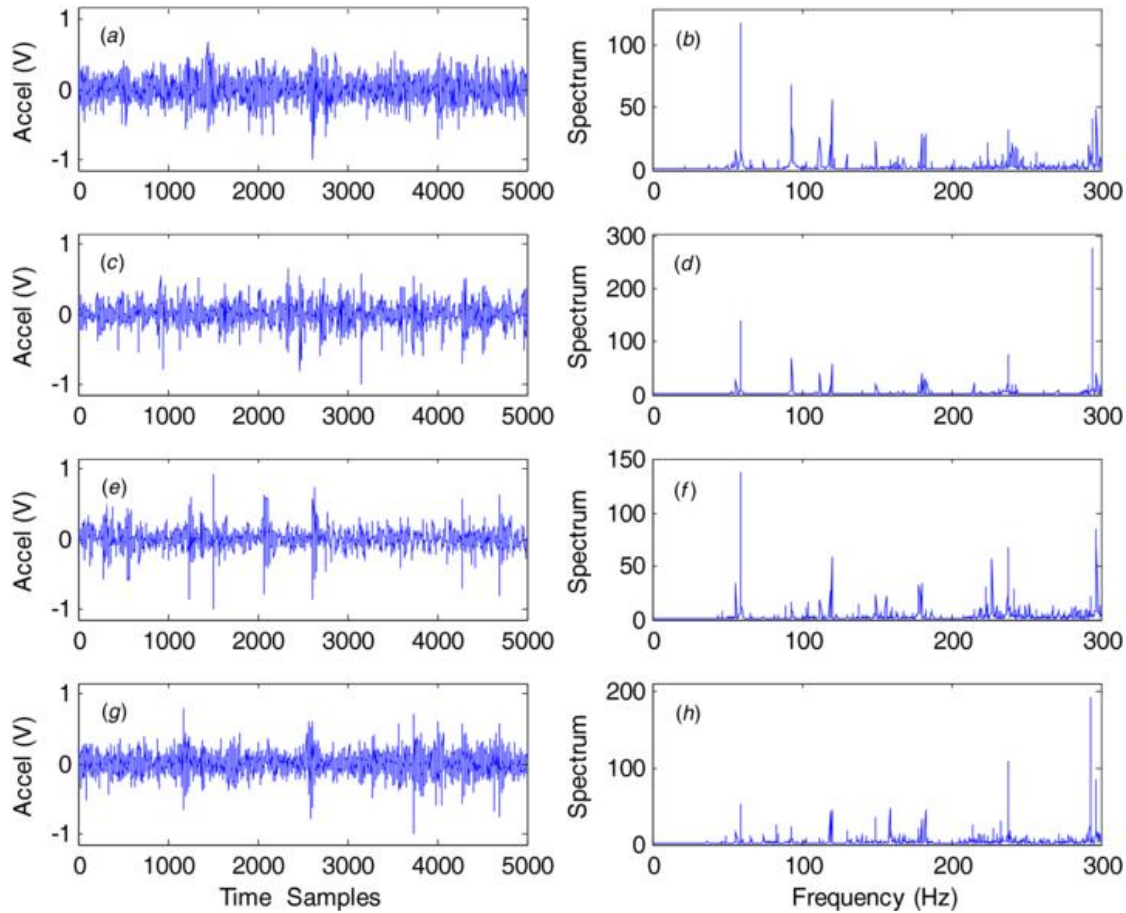
In de praktijk plaatst men een filter (het anti-alias filter) voor de ADC om frequenties die hoger zijn dan de helft van de sample frequentie uit het signaal te filteren. In het hoofdstuk filteren wordt uitgelegd hoe een filter gebouwd kan worden.



Het antialias filter geplaatst voor de ADC.

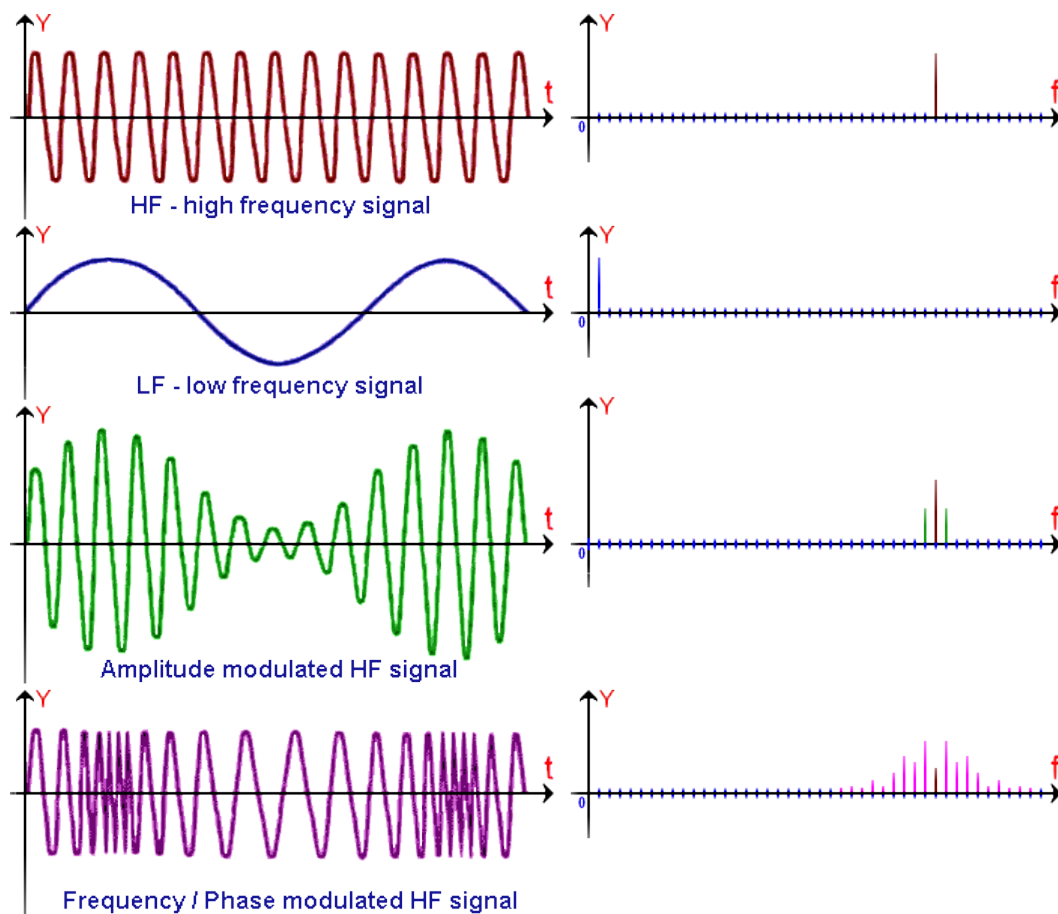
Tijd / Frequentie domein

We zijn gewend om signalen in het tijd domein te bekijken. Dat betekent dat als we het signaal in een grafiek zetten dat op de y-as de waarde van het signaal staat en op de x-as de tijd. Op de grafiek hieronder staan links het signaal van verschillende motoren uitgezet tegen de tijd.



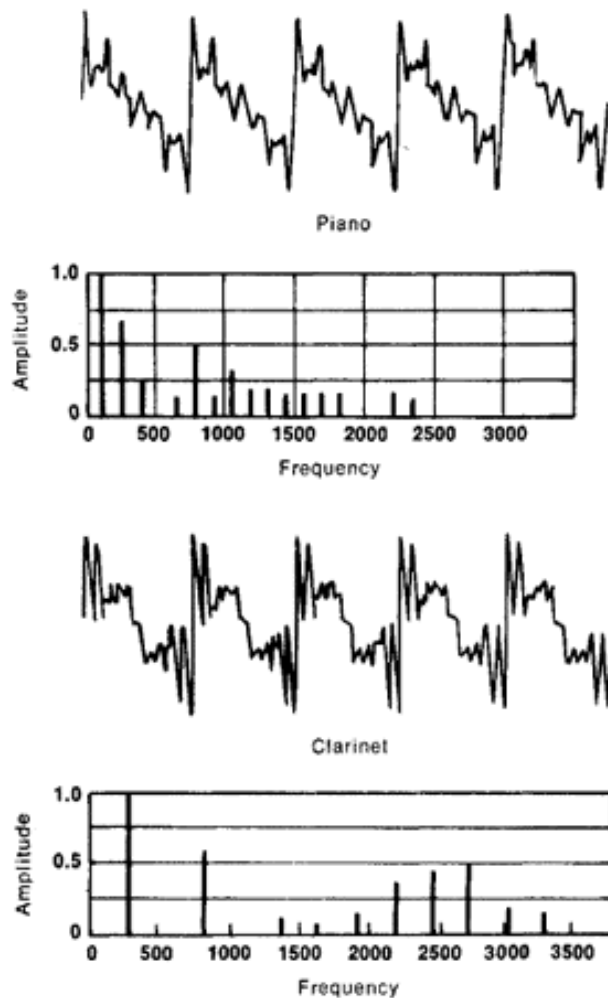
Het is echter ook mogelijk om te kijken welke frequenties allemaal in een signaal zitten. In de rechter grafiek zien we de verschillende motoren uitgezet in het frequentie domein. Indien we weten dat een bepaalde trilling (frequentie) overeenkomt met een falend kogellager is in de rechter grafiek veel sneller te zien of de kogellager nog correct is.

In de volgende grafiek zijn nog een aantal voorbeelden te zien van hoe een signaal er uit ziet in het tijd en het frequentie domein.



Zoals te zien is wordt een perfect sinus weergegeven als 1 piekje in het frequentie domein. Een vervormde sinus bestaat uit meer piekjes in het frequentie domein.

Muziek tonen bestaan ook uit vervormde sinussen. In de grafiek op de volgende bladzijde zijn de signalen van een piano en klarinet uitgezet in het tijd en het frequency domein. Het verschil tussen de twee instrumenten bestaat uit de "vorm" van de sinus.



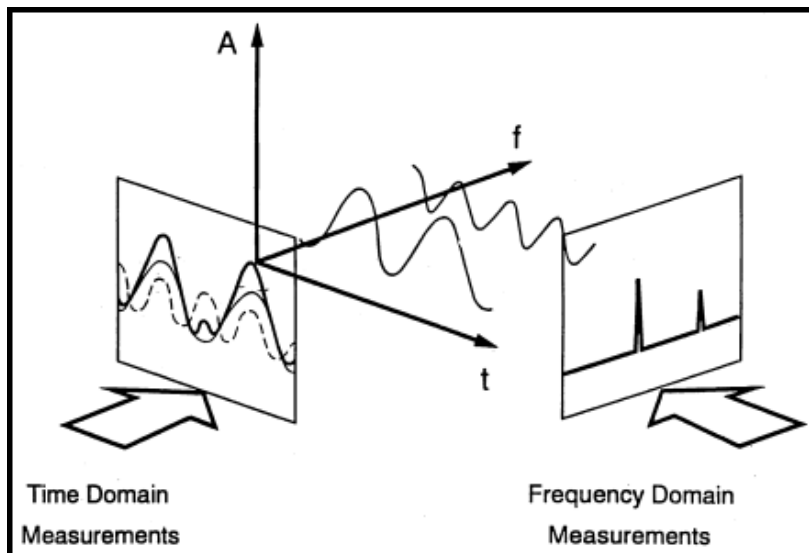
Als we naar de signalen in het frequentie domein kijken zien we dat er 1 grote piek in linker kant van de grafiek staat. Dit is de *grondtoon*. De pieken aan de rechterkant zijn de *boventonen*. De boven tonen worden gegeven door:

$$f_{boventoon} = n \cdot f_{grondtoon}$$

Waarbij n een geheel getal is (groter dan 0). De boventoon frequentie bestaat dus uit een veelvoud van de grondtoon frequentie (voor de klarinet lijkt dit niet geheel op te gaan, dit zou inhouden dat de toon over de tijd veranderd, hij “zweeft”).

De kracht van het frequentie domein is dat analyses vaak veel makkelijker worden.

De muziek herkenningdienst “shazam” bijvoorbeeld transformeert het signaal eerst naar het frequentie domein. Vervolgens onthoudt hij enkel de frequenties die het vaakst in het nummer voorkomen, en maakt daar vervolgens een vingerafdruk van (die hij vergelijkt met zijn database om het juiste nummer op te zoeken).

**Fourier transformatie**

Het transformeren van het signaal van het tijdsdomein naar het frequentiedomein noemt men "Fourier" transformatie.

Het transformeren van het signaal van het frequentiedomein naar het tijdsdomein noemt men "inverse Fourier" transformatie.

Een fourier transformatie gebeurt via een wiskundig algoritme. Een veel gebruikte is de FFT (fast fourier transform). Het algoritme is aardig ingewikkeld en valt buiten de scope van dit vak.

Digitaal filteren

De verwerkingseenheid kan gebruikt worden om het signaal digitaal te filteren. Hierbij wordt gebruik gemaakt van het verschil in eigenschappen tussen het signaal en eventuele stoorsignalen.

Moving average (voortschrijdend gemiddelde)

Het simpelste filter. Bij dit filter wordt het gemiddelde genomen van de laatste “n” metingen. Een filter wat het gemiddelde neemt van bv de laatste 3 metingen wordt een 3-tabs moving average filter genoemd.

Signaal:	5	6	7	8	9	3	4	5	3
3 tabs moving average filter:			6	7	8	6,6	5,3	4	4

Mediaan filter

Bij een mediaan filter wordt de middelste waarde genomen. Een 3-tabs mediaan filter pakt de laatste 3 meetwaarden en bepaalt de middelste waarde.

Signaal:	5	6	7	8	9	3	4	5	3
3 tabs mediaan filter:			6	7	8	8	4	4	4

FIR filter

Een FIR filter lijkt op een moving average filter. Waarbij bij een 3-tabs moving average in feite 3 keer vermenigvuldigd wordt met 1/3 zijn de coëfficiënten bij een FIR filter niet constant. De coëfficiënten van een FIR filter wordt de “kernel” genoemd.

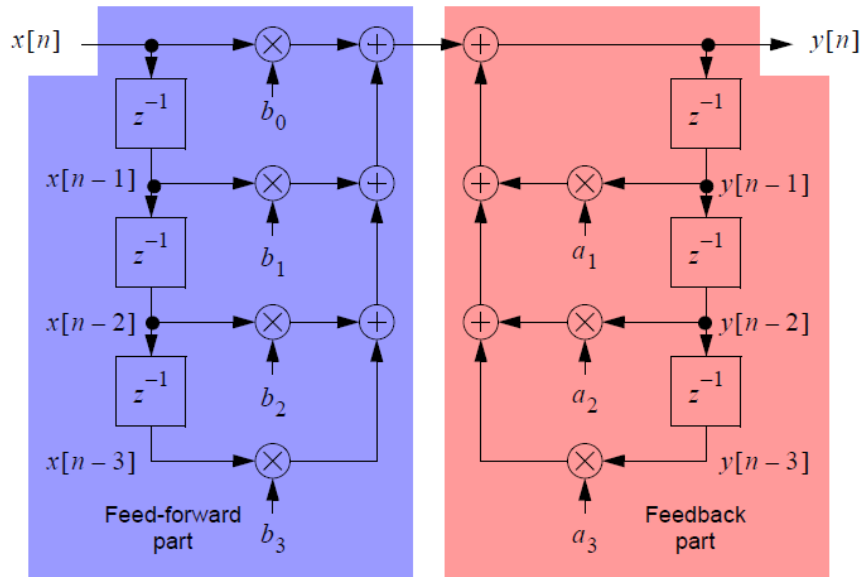
Signaal:	5	6	7	8	9	3	4	5	3
3 tabs FIR filter met kernel -1,2,-1:			0	0	0	7	-7	0	3

Zoals hierboven te zien kan met een FIR filter ook een hoog doorlaat filter gemaakt worden. De keuze van de coëfficiënten van de kernel bepaald welke frequenties het filter doorlaat.

IIR filter

Een FIR filter lijkt op een FIR filter. Bij een IIR filter wordt echter ook de uitgang meegenomen in de berekening:

$$y[n] = \sum_{l=1}^3 a_l y[n-l] + \sum_{k=0}^3 b_k x[n-k]$$



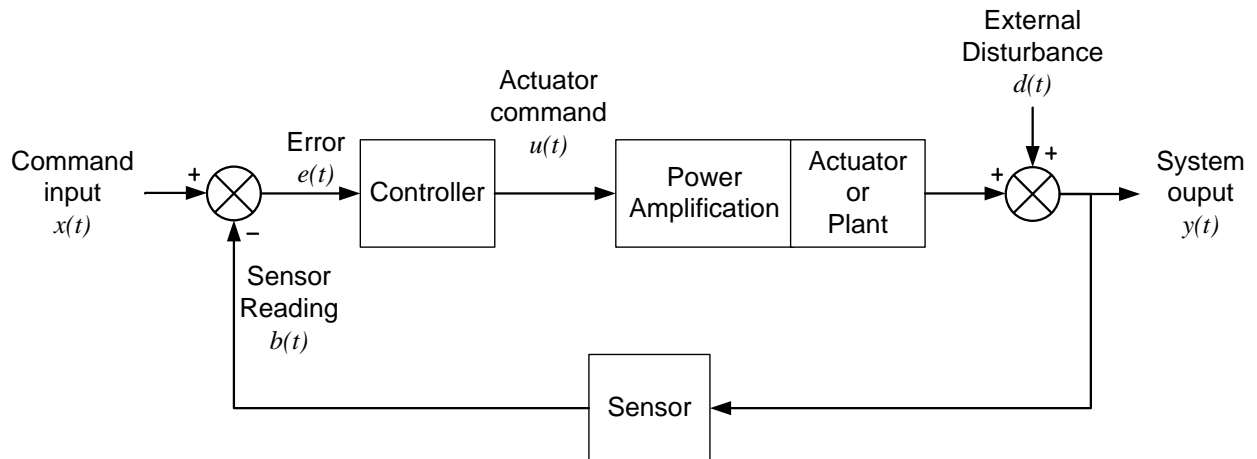
Direct Form I Structure

In feite is het een recursieve berekening. Een voordeel is dat het een stuk efficiënter werkt dan een FIR filter (minder berekeningen nodig).

PID control

Laten we het voorbeeld van een schip nemen. We willen een besturingssysteem bouwen dat een schip automatisch op koers houdt. Dan hebben we een sensor nodig die de positie van het schip meet. We moeten ook de koers weten welke het schip moet varen (het setpoint). Tevens moeten we een manier hebben om het roer te kunnen bewegen (de actuator).

Schematisch kunnen we het als volgt weergeven:



Waarbij $x(t)$ de gewenste koers van het schip is. De koers die het schip echt vaart wordt gegeven door $y(t)$. Het verschil tussen wat de sensor meet ($b(t)$) en de gewenste koers is $e(t)$ (de fout).

De uitdaging is nu om een geschikt controller algoritme te vinden wat met de fout $e(t)$ de correcte commando's naar het roer geeft ($u(t)$).

fout $e(t)$



Indien we een schip zoals hier links zien dan wordt de gewenste koers gegeven door de lijn recht omhoog. Zoals te zien is vaart het schip niet scherp aan de koers maar heeft een fout $e(t)$.

Hoe zou men het roer nu moeten bewegen om weer op koers te brengen? Een eerste intuïtieve manier is om de uitslag van het roer te schalen met de fout. Dus als het schip heel erg fout lijkt te gaan geef je het een grote uitslag. Als het schip een kleine afwijking heeft geef je het roer een kleine uitslag.

Deze manier van regelen noemt men "proportioneel" regelen. De regelwet hiervoor wordt gegeven door:

$$u(t) = K_p \cdot e(t)$$

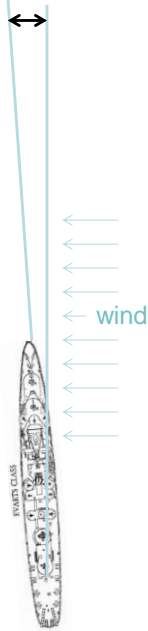
Zoals je in bovenstaande plaatje ziet wordt de fout, $e(t)$ gedefinieerd als de gewenste koers-actuele koers. Dus als hij naar links afwijkt wordt het roer naar rechts bewogen.

De factor K_p is de proportionele constante of factor. Hoe hoger de K_p , hoe sterker (agressiever) de stuur actie.



Wat gebeurt er nu als er een wind opsteekt? De wind zal een structurele fout veroorzaken die je niet weg geregeld krijgt met enkel een proportionele actie. In feite zit je met je regelactie steeds een beetje fout.

fout $e(t)$



Als je nu de fouten uit het verleden optelt kun je hiervoor compenseren. Hiermee kom je op de volgende regelwet uit:

$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) \cdot d\tau$$

Zoals je ziet tel je de fouten uit het verleden bij elkaar op. Deze component noemt men de I (integration) actie.

Hoe kun je de regeling nog slimmer maken?

Een slimme schipper zal kijken hoe snel het schip reageert op een stuuractie. Dus hoe snel verandert de fout over de tijd.

Hierdoor wordt de regelwet uitgebreid met:

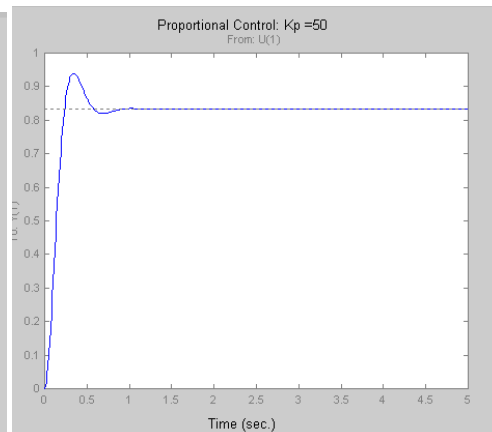
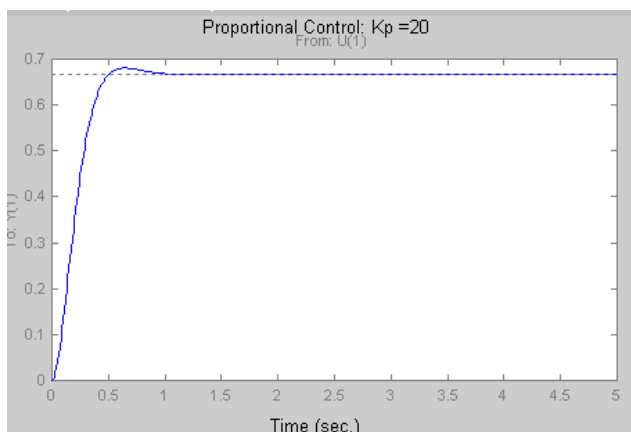
$$u(t) = K_p \cdot e(t) + K_i \cdot \int_0^t e(\tau) \cdot d\tau + K_d \cdot \frac{de(t)}{dt}$$

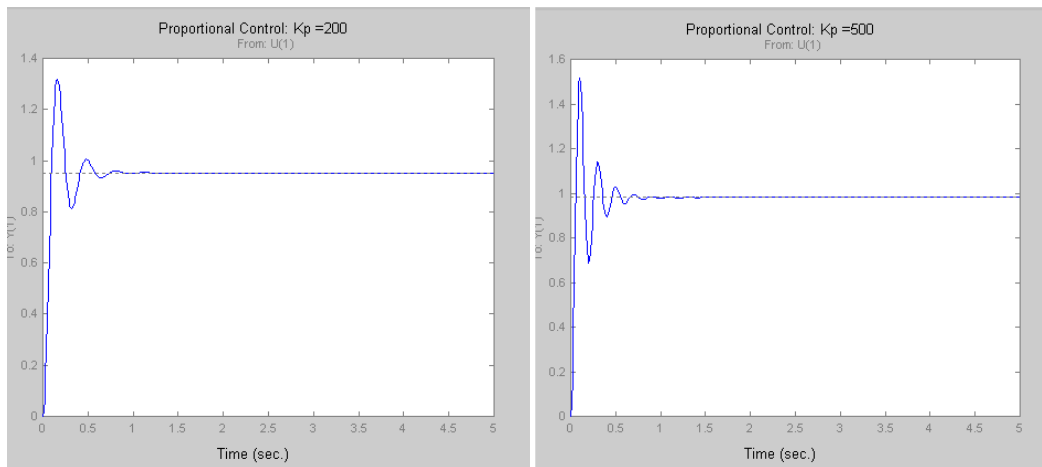
De d-actie noemt men de derivatieve actie.

Het geheel van de 3 regelacties geeft een PID controller. Je kan ook een PI controller, of een PD controller hebben (enz.).

Het inregelen van de K factoren kan op verschillende manieren. Een bekende manier is de *Ziegler-Nichols* methode. Zie het internet voor meer informatie hierover.

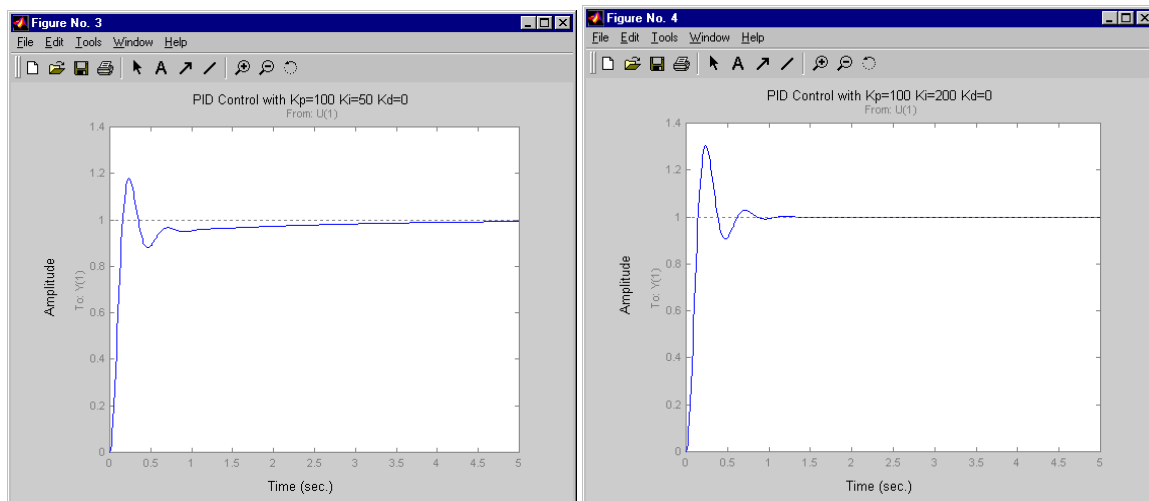
Wat is nu de invloed van de K_p constante. Hieronder wat voorbeelden van een regelsysteem met verschillende K_p waarden:





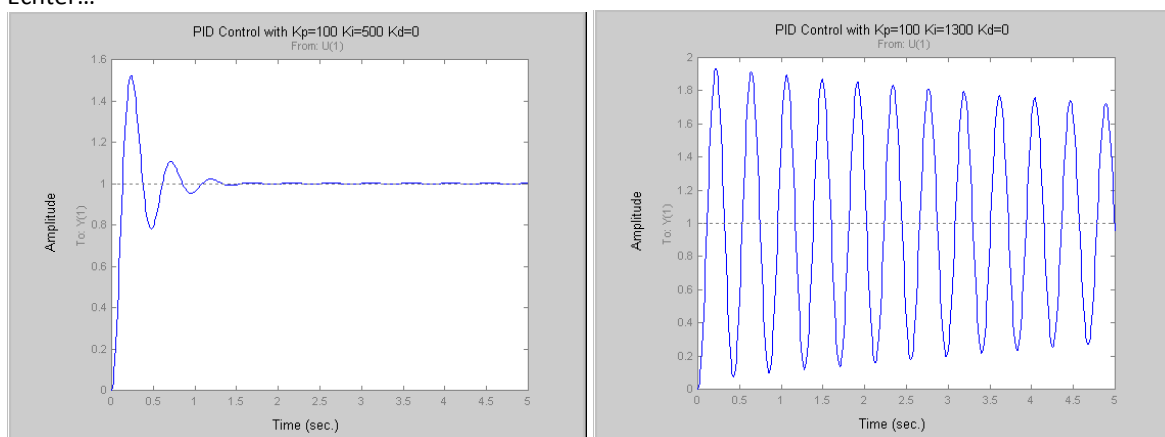
Zoals je ziet zorgt een hogere K_p waarde voor een snellere regeling, maar ook voor meer overshoot en meer oscillatie.

Wat is de invloed van de K_i waarde?



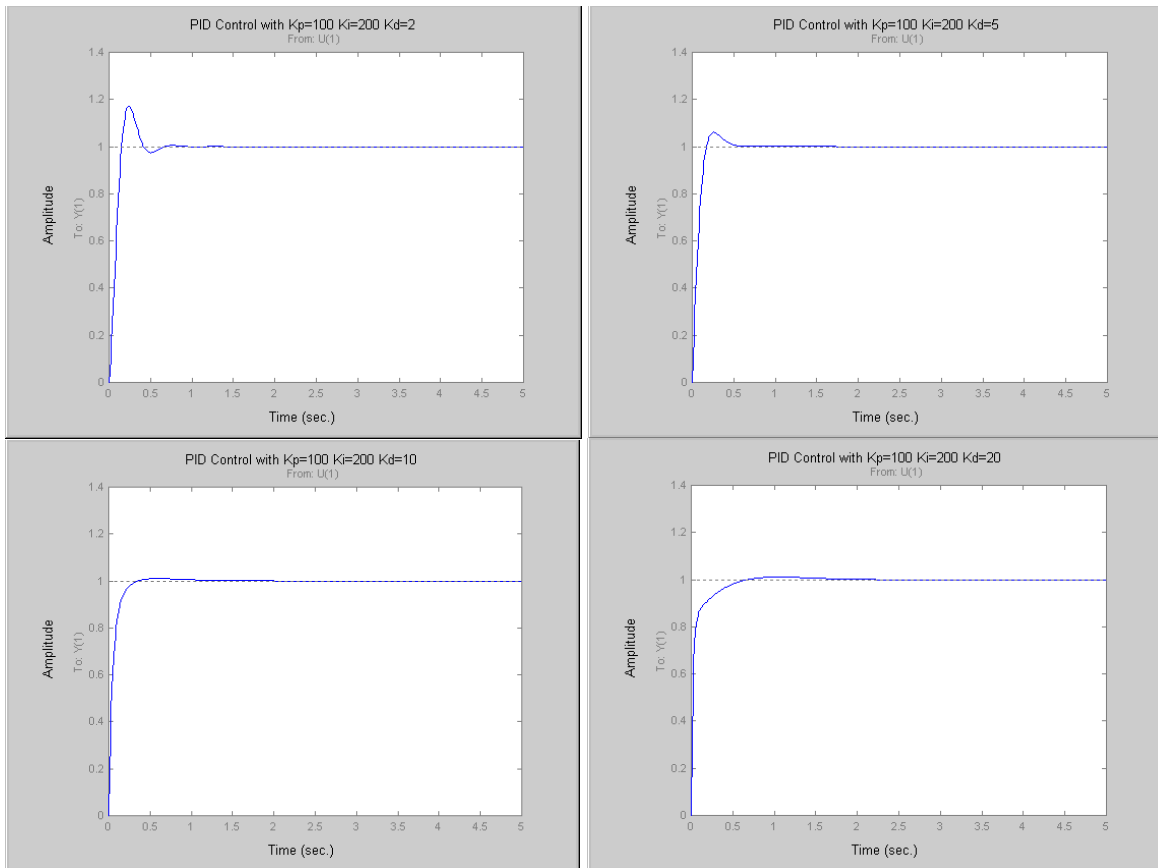
Een hogere K_i waarde lijkt te zorgen voor een snellere/nauwkeurigere regeling.

Echter...



Een te hoge waarde geeft een instabiele regeling!

Als laatste de d-actie:



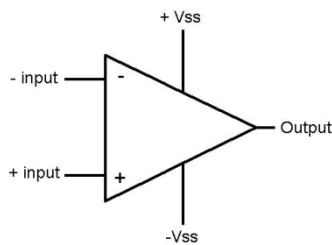
Ook hier weer: een te hoge d-waarde geeft oscillaties!

Motoren

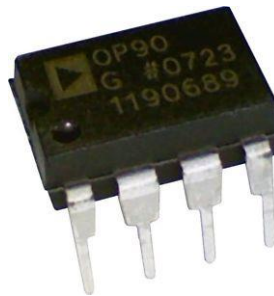
Zie reader / slides over motoren.

OpAmp

Een OpAmp is een geïntegreerd circuit (IC of chip) bestaand uit een groot aantal transistoren. Het symbool van een OpAmp wordt gegeven door een driehoek met 2 ingangen en 1 uitgang.



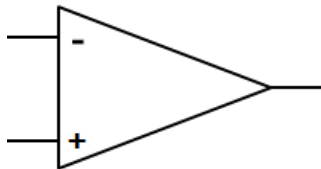
Symbool van een Opamp



De OP90, een voorbeeld van een OpAmp.

De positieve voedingsspanning wordt aangesloten op +Vss. De negatieve voedingsspanning (of in het geval van een single-supply opamp, de aarde) wordt aangesloten op -Vss.

Vaak worden in schema's de voedingslijnen weggelaten (maar dan zijn ze nog steeds wel aangesloten, anders werkt hij niet):



De OpAmp werkt als volgt:

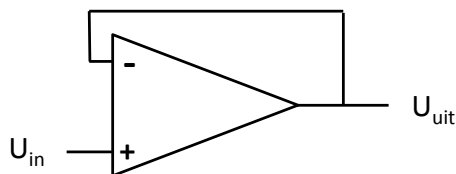
- 1) De OpAmp kijkt naar het verschil tussen de + en de - ingang.
- 2) Dat verschil versterkt hij met een factor van 1 miljoen of meer (afhankelijk van de OpAmp) en zet dat signaal op de uitgang.
- 3) De ingangsweerstand van de OpAmp is zo hoog dat er normaal gesproken geen stroom in de + en - ingang loopt.

Door de OpAmp op verschillende manieren aan te sluiten kunnen verschillende functionaliteiten gemaakt worden (bv een versterker, comparator).

Een aantal verschillende configuraties zullen nu behandeld worden:

Buffer (spanningsvolger)

Indien de uitgang van de OpAmp aangesloten wordt op de - ingang van de OpAmp en het ingangssignaal (U_{in}) op de + ingang is er sprake van een Buffer of spanningsvolger (follower).



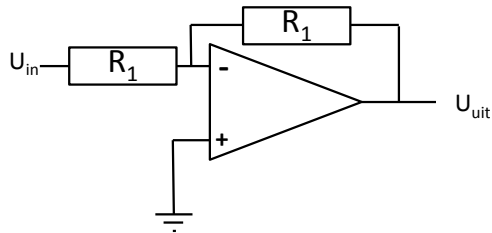
Voor deze schakeling geldt dat:

$$U_{uit} = U_{in}$$

Wat is het nut van deze schakeling? Stel dat er een sensor is met een hoge bronweerstand. Aangezien de ingangsweerstand van de OpAmp zeer hoog is kan hij zonder problemen op de sensor aangesloten worden. De uitgang van de OpAmp heeft een lage bronweerstand, dus daar kunnen zonder problemen ADC's of multimeters op aangesloten worden (zie ook het stuk over bronweerstand in het hoofdstuk Sensoren).

Geïnverteerde Buffer (Geïnverteerde spanningsvolger)

Indien de OpAmp via onderstaand schema wordt aangesloten is er sprake van een geïnverteerde buffer of spanningsvolger (inverted follower).



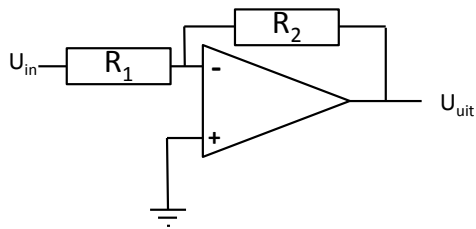
Voor deze schakeling geldt dat:

$$U_{uit} = -U_{in}$$

Wat is het nut van deze schakeling? Ook deze volgt de ingangsspanning, echter dan wel geïnverteerd (min wordt plus, en plus wordt min).

Geïnverteerde versterker

Indien de rechter weerstand uit voorgaand schema vervangen wordt door een andere weerstand is het mogelijk een geïnverteerde versterker (inverted amplifier) te bouwen.



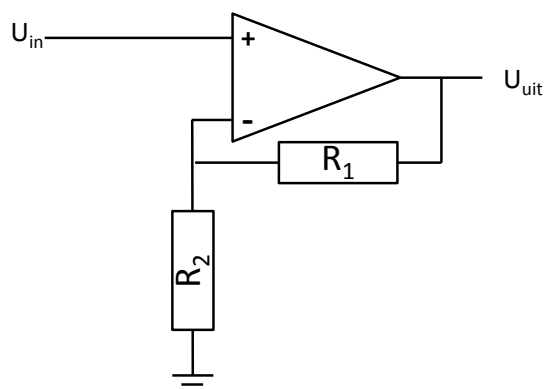
Voor deze schakeling geldt dat:

$$U_{uit} = -U_{in} \cdot \frac{R_2}{R_1}$$

Wat is het nut van deze schakeling? Hij versterkt (weliswaar geïnverteerd) de spanning. De versterkingsfactor wordt gegeven door $-R_2/R_1$.

Versterker

Indien de OpAmp via onderstaand schema wordt aangesloten is er sprake van een versterker (amplifier).



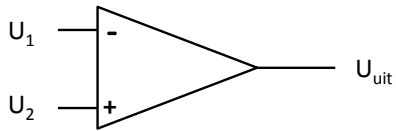
Voor deze schakeling geldt dat:

$$U_{uit} = U_{in} \cdot \left(1 + \frac{R_1}{R_2}\right)$$

Wat is het nut van deze schakeling? Hij versterkt de spanning. De versterkingsfactor wordt gegeven door $(1+R_1/R_2)$.

Comparator

Indien de OpAmp via onderstaand schema wordt aangesloten is er sprake van een comparator (vergelijker)



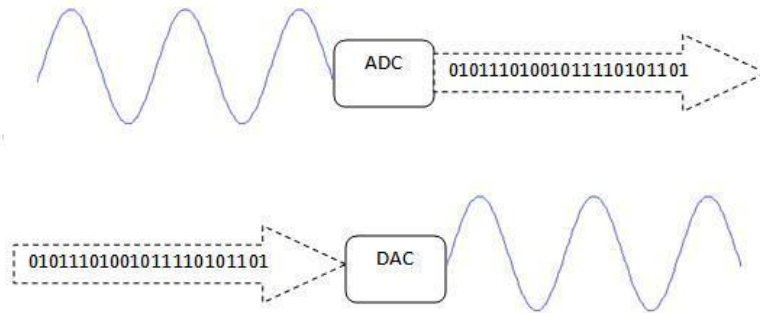
Indien U_2 groter is dan U_1 zal U_{uit} naar de positieve voedingsspanning gaan. Indien U_1 groter is zal U_{uit} naar de negatieve voedingsspanning gaan.

Wat is het nut van deze schakeling? Twee spanningen kunnen ermee vergeleken worden. De uitgang van de OpAmp zal hoog of laag worden, afhankelijk van welke van de 2 spanningen groter is.

ADC

Een ADC (Analoog Digitaal Converter) is een geïntegreerd circuit (IC of chip) dat een analoog signaal om kan zetten naar een digitaal signaal.

Een ADC heeft een aantal karakteristieken die hieronder behandeld worden.



Resolutie (bit diepte)

Zoals in bovenstaande afbeelding te zien is wordt het analoge signaal omgezet naar een digitaal signaal.

Een digitaal signaal wordt opgeslagen als een reeks getallen. Elk getal wordt intern opgeslagen als een binair getal.

Binaire getallen bestaan uit een aantal bits. Een byte (hap) bestaat bijvoorbeeld uit 8 bits, een nibble (hapje) uit 4 bits. Met een byte kun je dus 256 (van 0-255) verschillende waarden hebben, met een nibble 16 (0-15).

De resolutie van een ADC wordt bepaald door hoeveel bits de ADC is.

Een 8-bits ADC kan dus de meetwaarde in 256 stapjes opslaan. Een 16-bits ADC kan de meetwaarde in 65536 stapjes opslaan.

ADC bit diepte	Aantal stapjes
8	256
10	1024
12	4096
16	65536
20	1048576

Indien een sensor een output heeft van 5 Volt, en de sensor wordt uitgelezen met een 5V, 8 bits ADC dan is de kleinste spannings stappen die je kunt meten gelijk aan $5/(2^8)=0.02$ Volt.

De nauwkeurigheid van een (goede!) ADC wordt bepaald door de kleinste stapjes die hij kan maken. Dit noemt men de *quantisatie* fout. Slechte ADC's hebben andere afwijkingen waardoor de fout groter kan zijn dan die ene bit (waarbij die bit dus eigenlijk nutteloze informatie is).

Range (bereik)

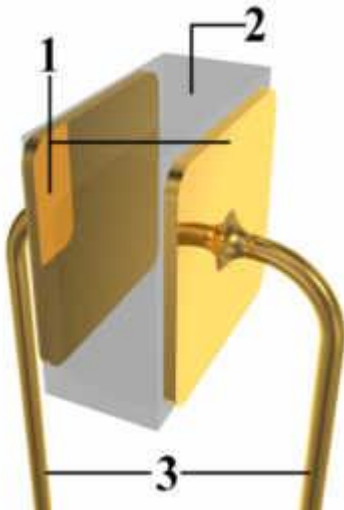
Zoals je in voorgaande paragraaf hebt kunnen zien wordt een ADC niet alleen bepaald door zijn bit diepte maar ook door zijn input range. Je hebt ADC's die van 0 tot 5 volt kunnen meten, maar ook ADC's met hele andere bereiken.

Maximale snelheid (Acquisition time)

De laatste belangrijke parameter van een ADC is zijn snelheid. Een ADC heeft even tijd nodig om het analoge signaal om te zetten naar het digitale signaal. Het grote verschil in types ADC (bv succesive approximation, Delta sigma, SAR, etc) is met name in de maximale snelheid van de ADC. Dit gaat echter ook in de prijs zitten!

Hardware matig filteren

Voor het filteren van een signaal moeten we eerst een condensator (Engels: capacitor) behandelen. Een condensator is een elektrische component die elektrische lading en elektrische energie opslaat. Hij is opgebouwd uit twee geleiders met een relatief grote oppervlakte, die zich dicht bij elkaar bevinden en gescheiden zijn door een niet-geleidend materiaal of vacuüm (bron wiki).



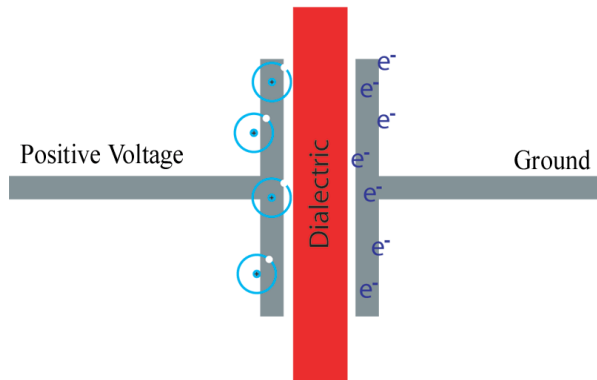
Condensator, 1=parallele platen, 2= niet geleidend materiaal (diëlektricum), 3= aansluitdraden

Een condensator kan er zeer verschillend uitzien:



Een condensator kun je beschouwen als een oplaadbare batterij. Zodra je er een spanning op aansluit begint hij lading op te slaan.

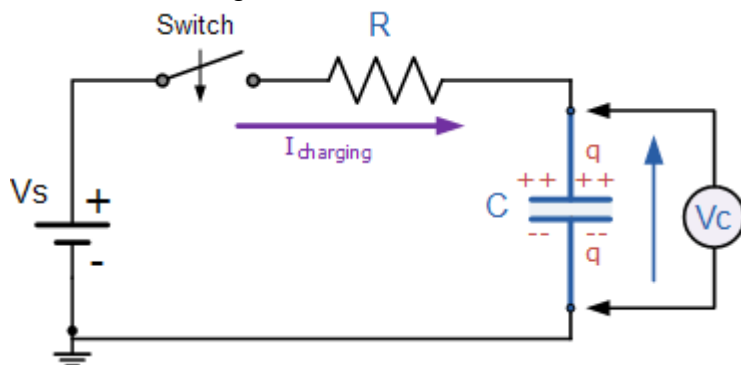
Dit komt omdat de ene plaat gevuld wordt met elektronen en de andere plaat met gaten (het gebrek aan elektronen). De gaten en de elektronen trekken elkaar aan (positief en negatief trekken elkaar aan).



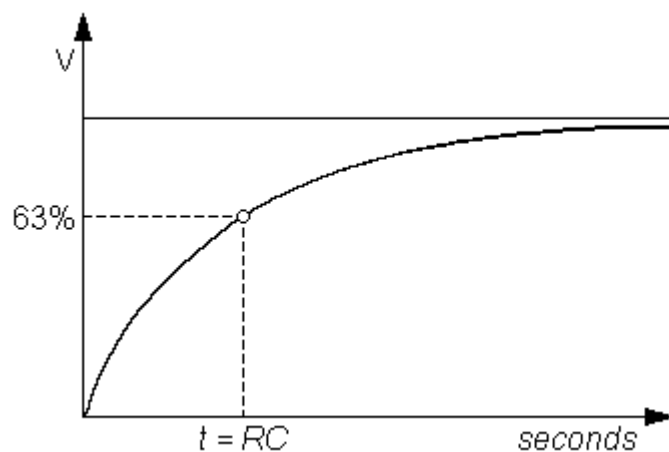
Hierdoor blijft de lading, ook als de spannings bron is weggehaald, op de condensator staan.

Voor gelijkspanning zal de condensator zich gedragen als een batterij met een interne weerstand van 0 ohm (als hij ontladen is). Zodra hij opgeladen is zal de weerstand zeer groot bedragen (voor gelijkstroom).

De condensator kan gebruikt worden in een RC filter:



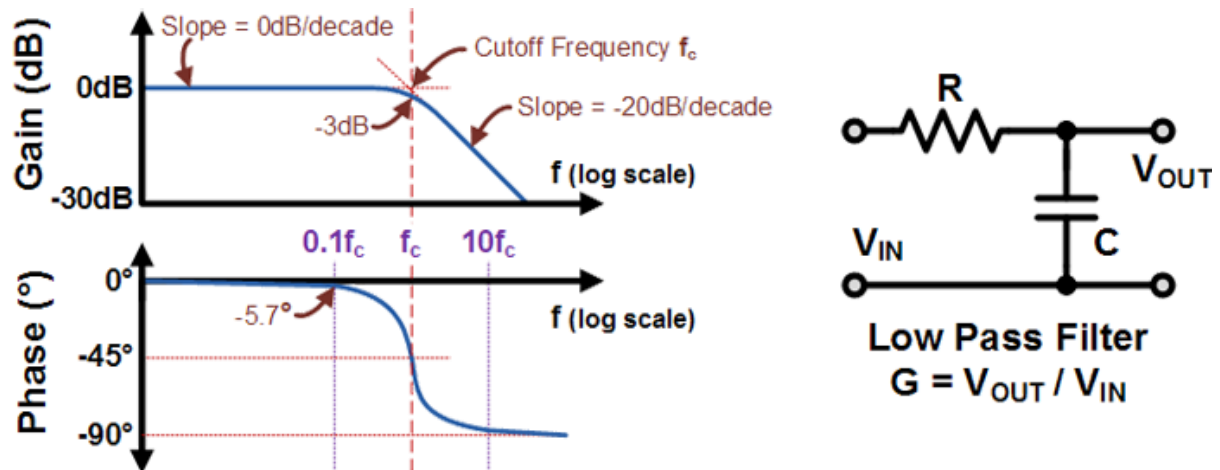
Indien de schakelaar gesloten wordt zal de condensator opgeladen worden:



De *tau*-tijd wordt gegeven door $t = R \cdot C$. Dit is de tijd die de condensator nodig heeft totdat hij 63% opgeladen is. De tau tijd wordt puur bepaald door de waarde van de weerstand (R) en de capaciteit van de condensator (C).

Een RC filter gedraagt zich als een *low pass filter* (laag doorlaat filter). Dat houdt in dat lage frequenties ongehinderd door gelaten worden, terwijl hoge frequenties uitgefilterd worden.

De overdrachtkarakteristiek van een RC filter wordt gegeven door:



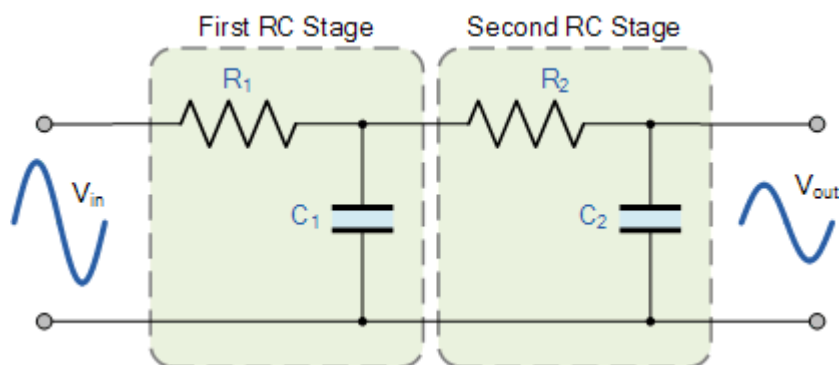
Waarbij de grafiek linksboven bekend staat als de Bode grafiek. Op de x-as staat de frequentie aangegeven. Op de Y-as de versterking (of verzwakking). Zoals te zien is laat hij lage frequenties ongehinderd door (0 dB = geen verzwakking). Frequenties boven de *cutoff frequency* worden verzwakt.

De cutoff frequency wordt gegeven door:

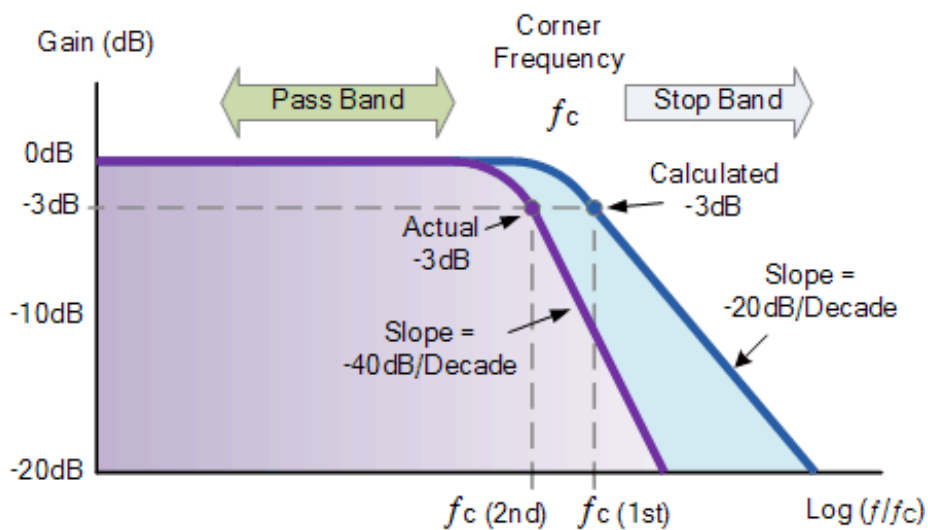
$$f_{cutoff} = \frac{1}{2 \cdot \pi \cdot R \cdot C}$$

Waarbij R weer de weerstand waarde is en C de capaciteit van de condensator.

Het is ook mogelijk om hogere orde RC filters te maken. Hierbij worden meerdere RC filters achter elkaar geplaatst.

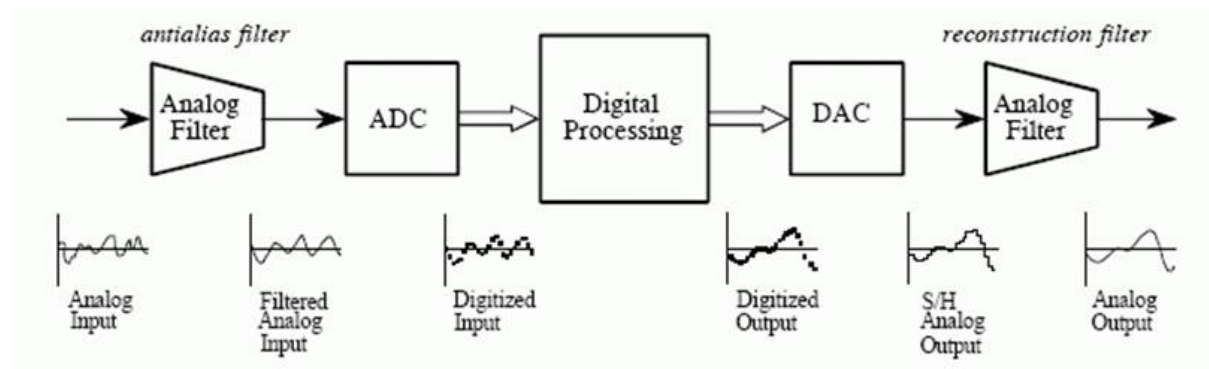


De overdracht (bode grafiek) van een 2^{de} orde filter is steiler:



De paarse grafiek is de overdracht van het 2^{de} orde filter.

Zoals ook in het hoofdstuk bemonsteren & Aliasing al beschreven is kan in een besturingssysteem meerdere filters voorkomen.



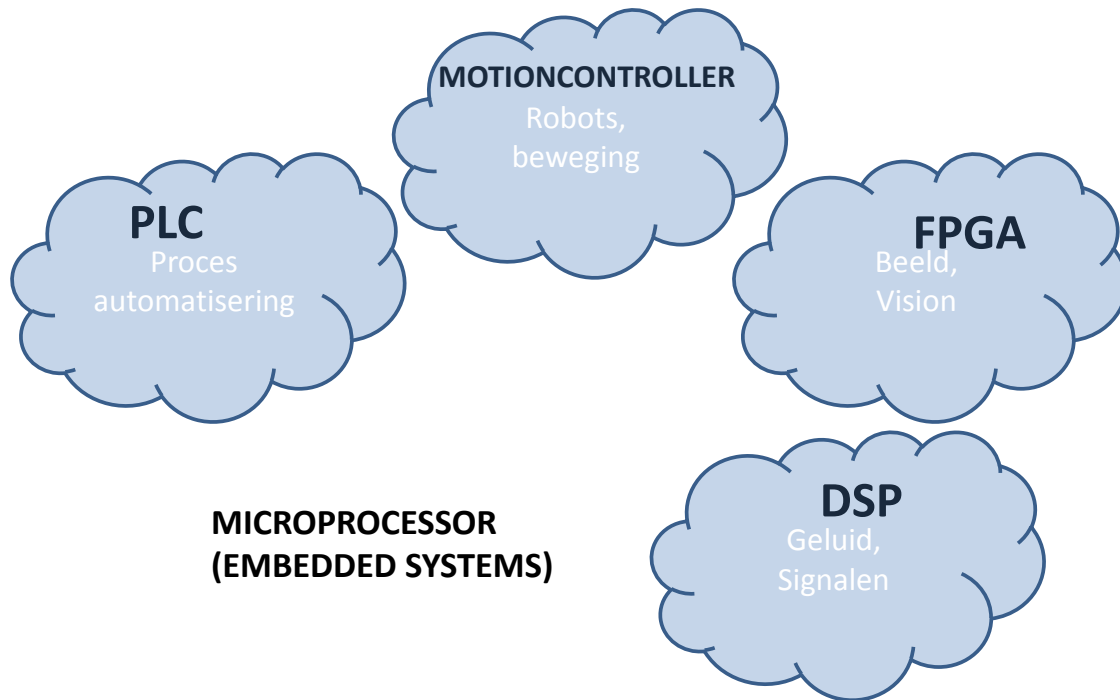
Het eerste filter in bovenstaand schema is de anti-alias filter (zie bemonsteren & aliasing voor de functie). Het tweede filter in bovenstaand schema is het reconstructie filter. Dit filter vlak de stapjes van de DAC af zodat het uitgangssignaal een vloeiend signaal wordt.

Soorten verwerkingseenheid

Voor de verwerkingseenheid zijn verschillende mogelijkheden. De verschillende mogelijkheden zijn in te delen als:

- PLC's
- Motion controllers
- FPGA's
- DSP's.
- Microprocessor (embedded systemen)

Deze verschillende verwerkingseenheden hebben allemaal hun sterkten & toepassingsgebied.



PLC's

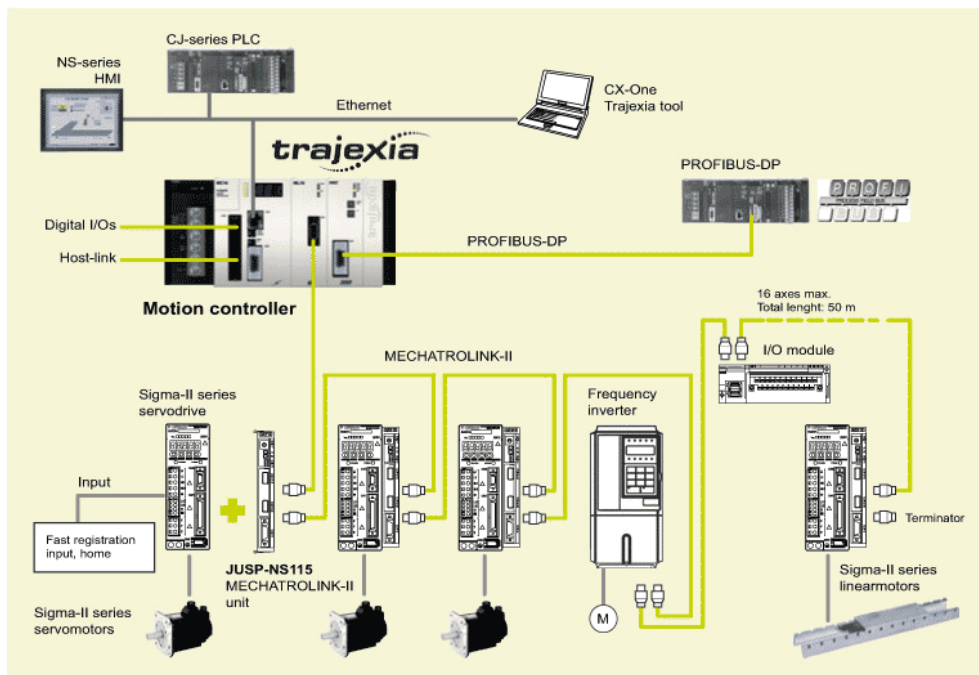
- Sterk gestandariseerd
- Modulair
- Voor processen (beetje ook motion)
- bv aansturen van kleppen etc.
- Industrieel
- Gebruiksvriendelijk



Motioncontrollers

- Sterk gestandariseerd

- Modulair
- Voor motoren/ etc.
- Industrieel
- Redelijk kant en klaar/gebruiksvriendelijk

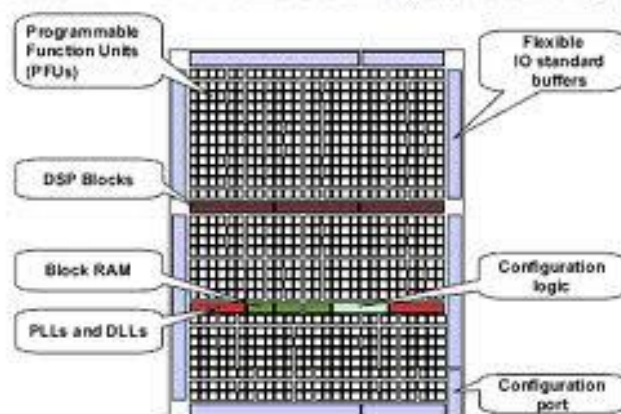


FPGA's

- Zeer grote parallele verwerkingscapaciteit!!!
- In feite programmeerbare hardware. Dus je kan het hardware design flexibel aanpassen!!
- Heel veel gebruikt voor video/beeld/vision toepassingen.
- Wil je veel data parallel bewerken? FPGA is king
- Niet heel eenvoudig om te programmeren

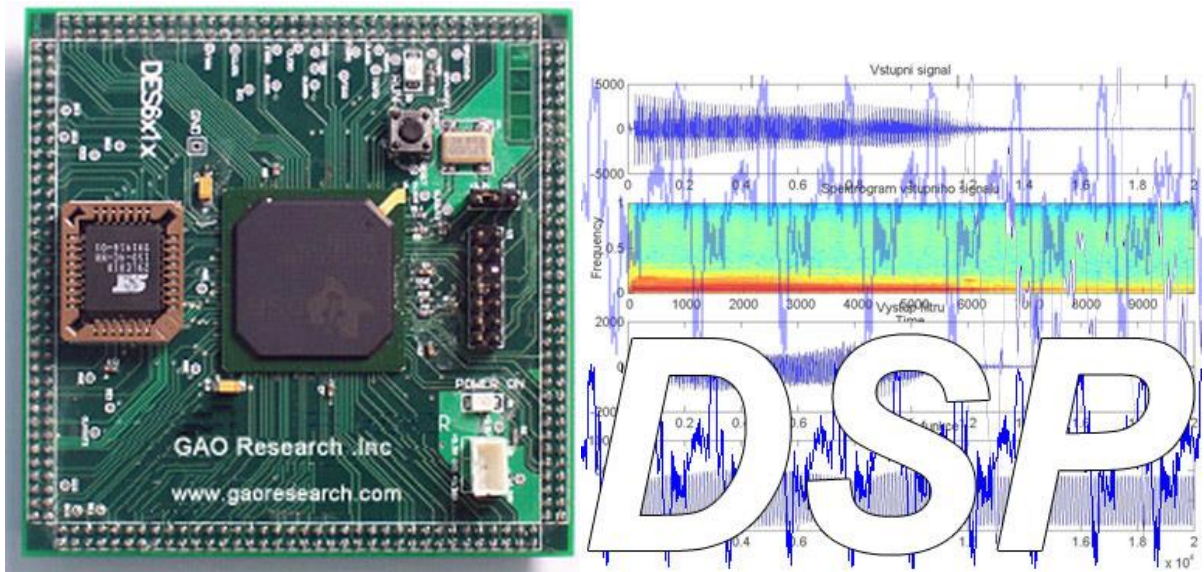


Modern FPGA Architecture (LatticeECP2)



DSP

- Heel veel gebruikt voor audio/signaal bewerkingen.
- Geoptimaliseerd voor signaal bewerkingen (bv convolutie sommen, FIR filters, etc)
- Niet heel eenvoudig om te programmeren



Microprocessor (embedded systemen)

Eigenschappen:

- Zeer flexibel, niet gespits op 1 toepassing. Kan alles eigenlijk wel
- Veel in embedded toepassingen. Bv telefoon, was machine, etc...
- Niet heel eenvoudig om te programmeren

