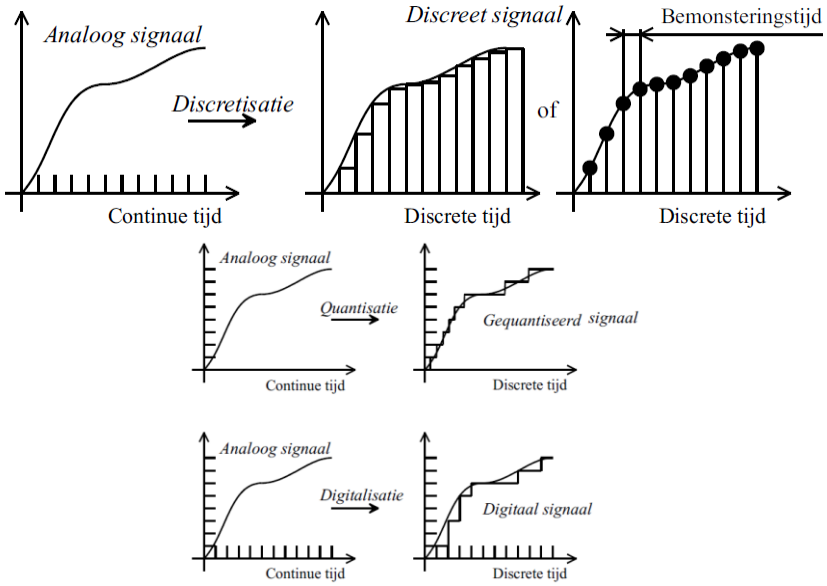
Automatisering

Differentie vergelijkingen
Z transformatie

Inleiding

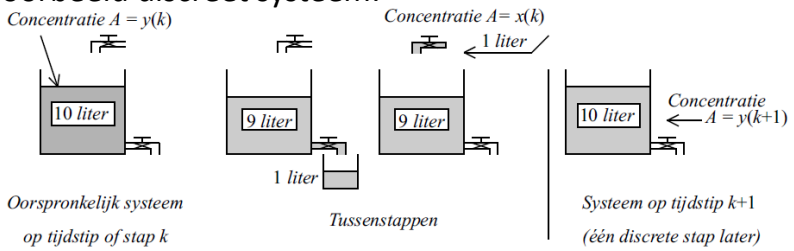
- Analoog naar digitaal
- Differentievergelijkingen
- Z-transformatie

ADC



Differentievergelijkingen

Voorbeeld discreet systeem:



Differentievergelijking: $y(k+1) = 0.9y(k) + 0.1x(k)$

Indien $y(0) = 0$ en $x(k) = 0.5$

$$y(1) = 0.9y(0) + 0.1x(0) = 0.05$$
$$y(2) = 0.9y(1) + 0.1x(1) = 0.095$$
$$y(3) = 0.9y(2) + 0.1x(2) = 0.1355$$
$$\dots$$

Differentievergelijkingen

- **Doel**: zoek gesloten vorm formule. Hierin vullen we k in om de waarde van de uitgang te kennen. Geen iteratieve berekening meer nodig.
- Bv. oplossing voor vorige vb. $y(k) = 0.5\left(1 - 0.9^k\right)$

  $k = 3 \rightarrow y(3) = 0.1355$
- Oplossing kan rechtstreeks gevonden worden in het tijdsdomein.
- Oplossing is makkelijker te berekenen in het frequentiedomein → Z-transformatie

Z-transformatie

- Z-transformatie op discrete systemen is equivalent met Laplace transformatie op continue systemen
- Sample continue functie

$$f(k) = f(t)\Big|_{t=kT} = f(kT) = f(k)$$

Vertrek van Laplace transformatie

$$F(p) = \int_0^\infty f(t)e^{-pt}dt$$

en vervang $t = kT$ we krijgen dan

$$F(p) = \sum_{k=0}^{\infty} f(kT)e^{-pkT}$$

Z-transformatie

- Kies nu een nieuwe veranderlijke

$$z = e^{pT}$$

- En vind

$$Z\{f(kT)\} = F(z) = \sum_{k=0}^{\infty} f(kT)z^{-k}$$

Bv. $\delta(k) = \{1,0,0,0, \dots\}$ geeft

$$F(z) = 1z^0 + 0z^{-1} + \cdots = 1 \rightarrow Z\{\delta(k)\} = 1$$

- Z-transformatieparen af lezen in tabel

$$f(k) \leftrightarrow F(z)$$

Met $Z\{f(k)\} = F(z)$ en $Z^{-1}\{F(z)\} = f(k)$

Z-transformatie paren

	$f(k), k \geq 0$	F(z)
1	$\delta(k)$	1
2	$u(k)$	$\frac{z}{z-1}$
3	$ku(k)$	$\frac{z}{(z-1)^2}$
4	$k^2u(k)$	$\frac{z(z+1)}{(z-1)^3}$
5	$k^3u(k)$	$\frac{z(z^2+4z+1}{(z-1)^4}$
6	$a^ku(k)$	$\frac{z}{z-a}$
7	$ka^ku(k)$	$\frac{za}{(z-a)^2}$
8	$k^2a^ku(k)$	$\frac{za(z+a)}{(z-a)^3}$

Eigenschappen van de Z-transformatie

- *Lineariteit*: $Z\{af(k) + bg(k)\} = aZ\{f(k)\} + bZ\{g(k)\}$
  $= aF(z) + bG(z)$
- *Integratie:* $\sum_{k=0}^{n} f(n) \leftrightarrow F(z)\frac{z}{z-1}$
- *Verschuivingstheorema's*:
  $f(k+1) \leftrightarrow z[F(z) - f(0)]; f(k-1) \leftrightarrow z^{-1}F(z)$
- *Beginwaarde theorema*: $\lim_{z\to+\infty} F(z) = f(0)$
- *Eindwaarde theorema*: $\lim_{k\to+\infty} f(k) = \lim_{z\to 1} F(z)\frac{z-1}{z}$
- *Vermenigvuldiging met k*: $kf(k) \leftrightarrow -z\frac{d}{dz}(F(z))$
- *Schaalfactor theorema*: $a^k f(k) \leftrightarrow F\left(\frac{z}{a}\right)$
- *Convolutie*: $f(k) * g(k) \leftrightarrow F(z)G(z)$

Automatisering

Discreet systeemgedrag

Inleiding

- Systeemgedrag nagaan in **discreet** frequentie en tijdsdomein
- Verband polen en tijdsgedrag
- Verband continue systemen (met bijhorende polen) met hun discrete tegenhangers
  - Belangrijk bij ontwerp van discrete regelaars
- Verbanden en verschillen bij harmonische analyse van continue en discrete systemen
- Beperkingen inschatten van discrete regelaar (tov continue regelaar)

Verband tussen Z-, tijd- en P-domein

- Vertrek van volgend continu signaal als impulsrespons

$$e(t) = e^{at}\cos(bt)\,1(t),$$

met $1(t)$ de continue eenheidsstapfunctie (1 voor positieve tijden anders 0 → causaal).

- Laplace getransformeerde is

$$E(p) = \frac{p-a}{(p-a)^2 + b^2}$$

- De (continue) polen van deze TF $p_{1,2} = a + jb$. Het nulpunt is $p = a$.

Verband tussen Z-, tijd- en P-domein

- Gediscritizeerd met sampleperiode T krijgen we

$$e(k) = e^{akT}\cos(bkT)\,1(k),$$

met $1(k)$ de discrete eenheidsstapfunctie.

- Anders geschreven krijgen we

$$e(k) = r^k\cos(\theta k)\,1(k),$$

waarbij $\begin{cases} r = e^{aT} \\ \theta = bT \end{cases}$.

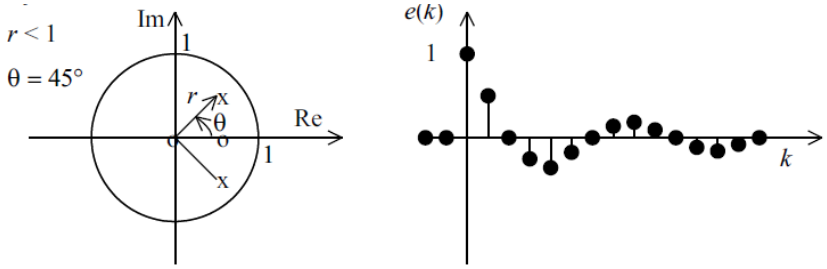
- Via transformatietabel vinden we

$$E(z) = \frac{z(z - rcos\theta)}{z^2 - 2rcos\theta z + r^2},$$

met nulpunten $0, rcos\theta$ en polen $z_{1,2} = rcos\theta \pm jrsin\theta = r\,e^{\pm j\theta}$

Verband tussen Z-, tijd- en P-domein

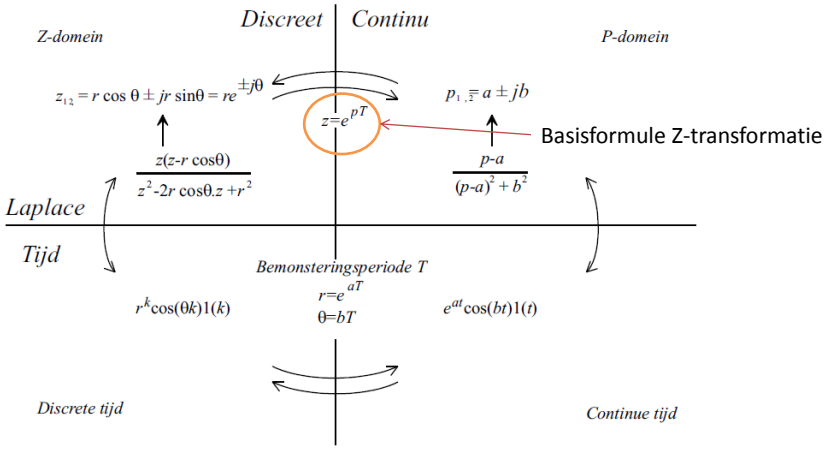
- Ligging van polen (x) en nulpunten (o) met bijhorende impulsrespons



$$z_{1,2} = e^{p_{1,2}T} = e^{(a\pm jb)T} = e^{aT}.e^{\pm jbT} = e^{aT}\left(\cos bT \pm j\sin bT\right) = r\cos\theta \pm jr\sin\theta$$

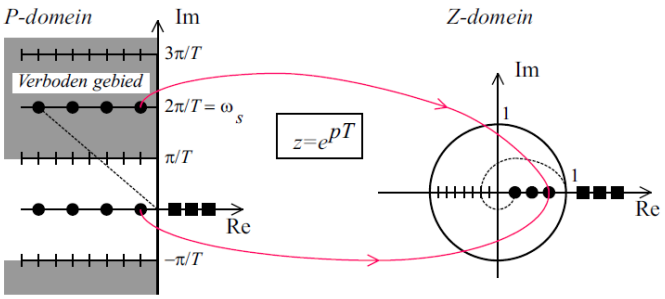
Verband tussen Z-, tijd- en P-domein

$$z_{1,2} = e^{p_{1,2}T} = e^{(a\pm jb)T} = e^{aT}.e^{\pm jbT} = e^{aT}\left(\cos bT \pm j\sin bT\right) = r\cos\theta \pm jr\sin\theta$$

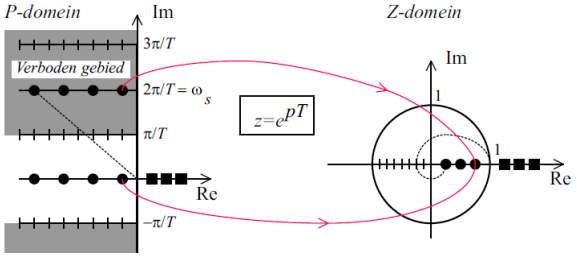


Verband tussen Z-, tijd- en P-domein

- Aliasing: geen eenduidig verband tussen discrete tijd en continue tijd. Uit de discrete tijdfunctie kunnen we niet altijd de continue functie reconstrueren.
- $p_2 = p_1 + \frac{j2\pi n}{T}$ en neem $z_1 \leftrightarrow p_1 \rightarrow z_2 = z_1$

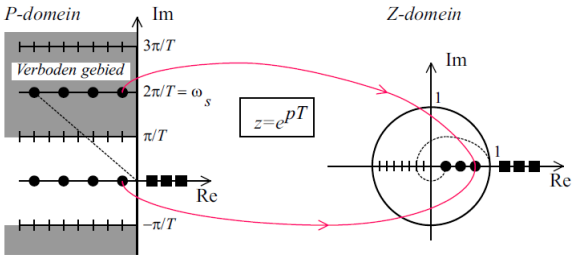


Verband tussen Z-, tijd- en P-domein



- $r > 1$ instabiel systeem ($a > 0$)
- $r = 1$ marginaal stabiel ($a = 0$)
- $r < 1$ stabiel (a<0)
- $\theta = 0$ geen oscillaties
- r bepaalt de transient gedrag ($r \approx 0 \rightarrow a \ll 0; r \approx 1 \rightarrow a \approx 0 \,\&\, a \in \Re^-$)

Verband tussen Z-, tijd- en P-domein



- θ → pulsatie overgangsverschijnsel

$$\omega = b = \frac{\theta}{T}$$

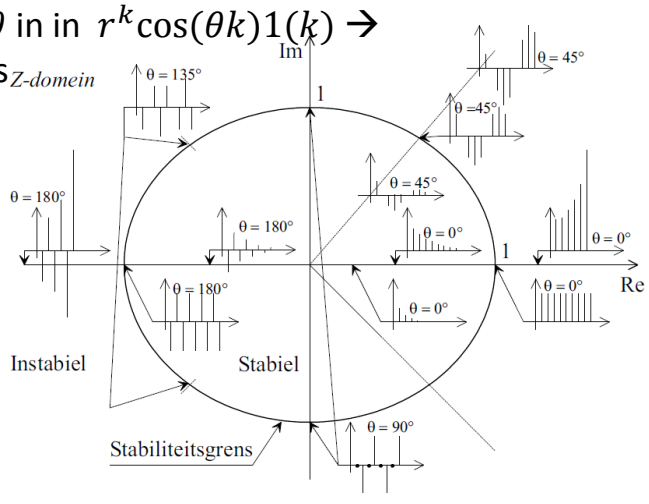
- θ bepaalt het aantal bemonsteringen N per periode van de oscillatie

$$N = \frac{2\pi}{\omega T} = \frac{2\pi}{\theta}$$

- Algemeen geldend verband: $z_{1,2} = e^{p_{1,2}T}$

Verband tussen de ligging van de polen en de impulsrespons

- Indien $z = re^{j\theta}$ gegeven is
- Vul dan r en θ in in $r^k \cos(\theta k) 1(k)$ → impulsrespons



Harmonische analyse bij discrete systemen

- Regimerespons bij discreet, LTI systeem met TF H op harmonische inputsequentie $x(k) = Asin(\theta k)$ is eveneens sinusoïdaal
  $y_r(k) = A\left|H(e^{j\theta})\right| \sin\left(k\theta + \angle H(e^{j\theta})\right)$,
- Is dus analoog aan continue geval
- Om discrete bode plot te tekenen, gebruik
  $\begin{cases} M = \left|H(e^{j\theta})\right| \\ \gamma = \angle H(e^{j\theta}) \end{cases}$.
- Frequentierespons discreet systeem is periodisch

$$e^{j(\theta+2\pi)} = e^{j\theta} \rightarrow H\left(e^{j(\theta+2\pi)}\right) = H\left(e^{j\theta}\right).$$

Voorbeeld

Neem de volgende TF:

$$H(z) = \frac{0,5}{z - 0,5}$$

De frequentierespons volgt uit

$$H(e^{j\theta}) = \frac{0,5}{e^{j\theta} - 0,5} = \frac{0,5}{\cos\theta + j\sin\theta - 0,5}$$

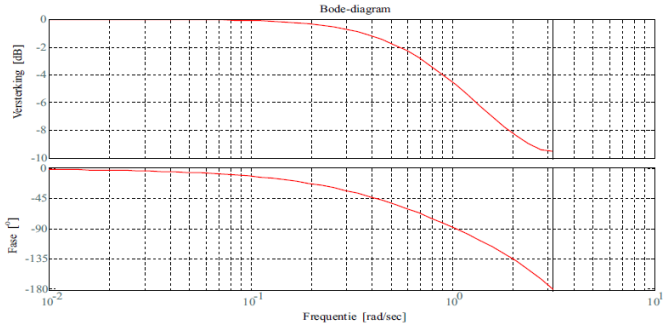
waaruit

$$\left|H(e^{j\theta})\right| = \frac{0,5}{\sqrt{1,25 - \cos\theta}}$$

en

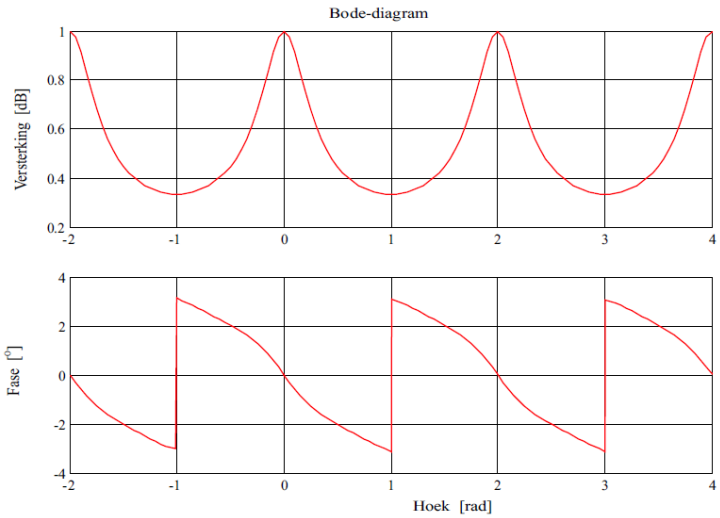
$$\gamma = -\text{bgtg}\frac{\sin\theta}{\cos\theta - 0,5}$$

Voorbeeld



- verband tussen de continue en discrete tijdrespons is enkel eenduidig bepaald voor hoeken tussen 0 en π (in feite tussen $-\pi$ *en* π*)*
- Met als gevolg ook een periodische frequentierespons

Voorbeeld



Automatisering

Discreet equivalent

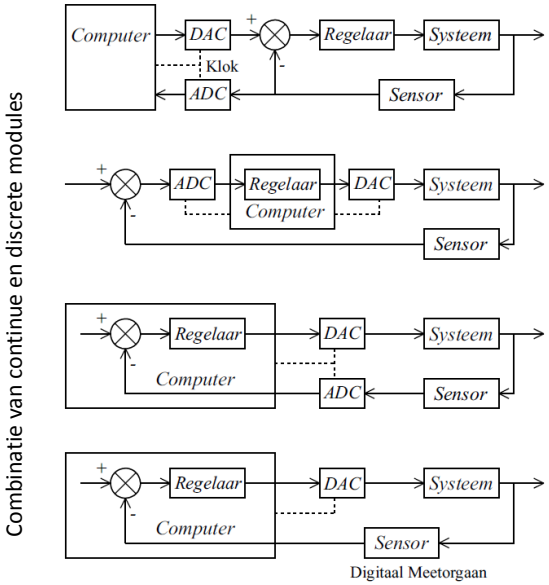
Inleiding

- Steeds meer digitale regelaars (bv. computer)
- Voordeel van computer is dat naast de controlefunctie er ook beslissingslogica kan zijn
- Effect van de bemonsteringsperiode *T* en van de quantisatie-eenheid *q* is belangrijk
- Indien T en q zeer klein → technieken voor continue systemen gebruiken
  - Economisch niet interessant
- Hier enkel effect van samplefrequentie f_s nagaan → *discrete regelaar*
- Hier houden we geen rekening met q → quantisatie is in de limiet aan-uit regelaar

Inleiding

- Ontwerp digitale regelaar:
  - Discreet equivalent van klassieke continue regelaar
  - Rechtstreeks in het z-domein met gelijkaardige technieken zoals wortellijnendiagram , frequentiedomein ontwerp zoals bij continue regelaars
  - Digitale toestandsregelaar
- Laatste 2 methodes eisen dat alle elementen van de regelkring discreet zijn of discreet worden voorgesteld.

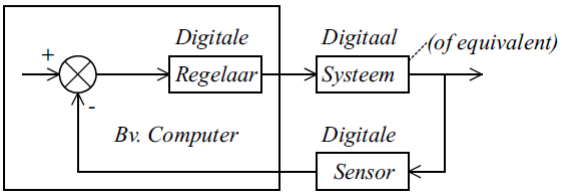
Digitale regelschema's



Combinatie van continue en discrete modules

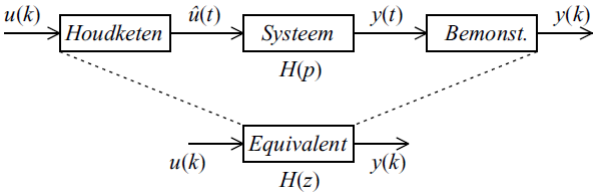
Digitale regelschema's

- Indien alle modules digitaal → kunnen een aantal klassieke instelmethoden gebruikt worden
- In vergelijking met alternatieven is deze methode het eenvoudigste
- In praktijk is het te regelen systeem dikwijls continu
  - Beschrijf als digitaal systeem



Digitale regelschema's

- Beschrijving als digitaal systeem



- Verschil discreet equivalent van het proces en discreet equivalent regelaar
  - Bij regelaar: benadering van continue regelaar (volgende slides)
  - Bij proces: geen benadering (discrete regelkring)

Digitale regelschema's

- Discreet equivalent:
  1. Ontwerp van een discreet equivalent door numerieke integratie;
  2. Ontwerp van een discreet equivalent door transformatie van polen en nulpunten;
  3. Ontwerp van een discreet equivalent m.b.v. een houdschakeling.

Methode 1: numerieke integratie

Principieel bevat de methode volgende stappen:

I) Stel de te benaderen TF $H(p)$ voor door een differentiaalvergelijking

II) Voer een numerieke integratie uit op deze differentiaalvergelijking. Dit levert een differentievergelijking die een voorstelling is van de discrete regelaar.

III) Zet de differentievergelijking om naar een discrete TF

Methode 1: voorbeeld

Neem als voorbeeld de integrator $H(p) = 1/p$. Dan is het stuursignaal $U(p)$

$$U(p) = H(p)E(p) = \frac{1}{p}E(p),$$

met $E(p)$ het foutsignaal. De overeenstemmende differentiaalvergelijking is

$$u(t) = u(0) + \int_0^t e(\iota)d\iota$$

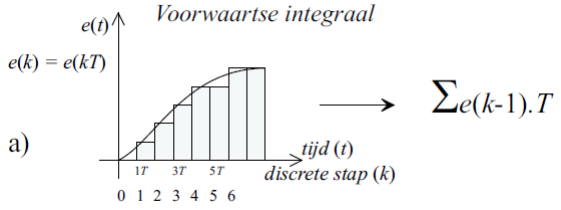
Numerieke integratie toegepast op deze vergelijking levert

$$u(kT) = u(0) + \int_0^{kT} e(\iota)d\iota = u(0) + \int_0^{kT-T} e(\iota)d\iota + \int_{kT-T}^{kT} e(\iota)d\iota$$

$$u(kT) = u\left[(k-1)\,T\right] + \int_{kT-T}^{kT} e(\iota)d\iota$$

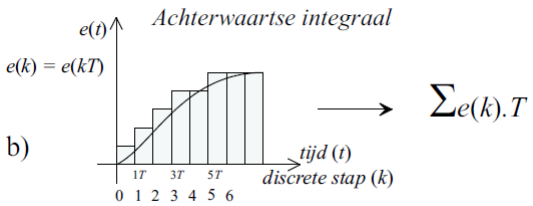
Methode 1: voorbeeld

- Evaluatie integraal rechterlid
  - Voorwaartse rechthoekregel
  - Achterwaartse rechthoekregel
  - Trapeziumregel (bilineaire transformatie)



Methode 1: voorbeeld

- Evaluatie integraal rechterlid
  - Voorwaartse rechthoekregel
  - Achterwaartse rechthoekregel
  - Trapeziumregel (bilineaire transformatie)



Methode 1: voorbeeld

- Invullen van de benadering voor de integraal geeft elk:

$$\text{a)}\ \ u(kT) = u\left[(k-1)T\right] + e\left[(k-1)T\right] \cdot T,$$

$$\text{b)}\ \ u(kT) = u\left[(k-1)T\right] + e(kT) \cdot T,$$

$$\text{c)}\ \ u(kT) = u\left[(k-1)T\right] + \frac{e((k-1)T) + e(kT)}{2} T.$$

- Z-transformatie

$$\text{a)}\ \ \left(1 - z^{-1}\right) U(z) = T z^{-1} E(z) \quad \rightarrow \quad \frac{U(z)}{E(z)} = \frac{T}{z-1},$$

$$\text{b)}\ \ \left(1 - z^{-1}\right) U(z) = T \cdot E(z) \quad \rightarrow \quad \frac{U(z)}{E(z)} = \frac{Tz}{z-1},$$

$$\text{c)}\ \ \left(1 - z^{-1}\right) U(z) = \frac{1}{2} T(z^{-1} + 1) E(z) \quad \rightarrow \frac{U(z)}{E(z)} = \frac{T\,(z+1)}{2\,(z-1)}.$$

Methode 1: voorbeeld

- Vergelijking met $U(p) = \frac{1}{p}E(p)$ leert dat p kan vervangen worden door:

a) voorwaartse rechthoekregel: $p = \dfrac{z-1}{T}$,

b) achterwaartse rechthoekregel: $p = \dfrac{z-1}{Tz}$

c) trapeziumregel: $p = \dfrac{2(z-1)}{T(z+1)}$.

Methode 1: voorbeeld 2

$$U(p) = Kr\left(1 + \frac{1}{\tau_i p}\right)E(p)$$

Het discreet equivalent van een PI-regelaar volgens numerieke integratie met trapezium regel

zodat

$$U(z) = Kr\left(1 + \frac{T(z+1)}{2\tau_i(z-1)}\right)E(z)$$

of

$$(z-1)U(z) = Kr\left((z-1) + \frac{T}{2\tau_i}(z+1)\right)E(z) = Kr\left(\left(1 + \frac{T}{2\tau_i}\right)z + \frac{T}{2\tau_i} - 1\right)E(z).$$

Herschrijf dit als:

$$(z-1)U(z) = (a.z + b)E(z)$$

of

$$(1 - z^{-1})U(z) = (a + b.z^{-1})E(z)$$

waaruit

$$u(k) = u(k-1) + a.e(k) + b.e(k-1)$$

Makkelijk te programmeren

Methode 2: transformatie van nulpunten en polen

- Verband polen in Z en P domein → $z = e^{pT}$
- Extrapoleer dit verband naar de nulpunten

Methode 2: transformatie van nulpunten en polen

- Procedure:

1. Transformeer alle polen van $H(p)$ naar het Z-domein volgens vergelijking $z = e^{pT}$
2. Transformeer alle eindige nulpunten van $H(p)$ op dezelfde wijze naar nulpunten van $H(z)$;
3. Laat met elk nulpunt op oneindig van $H(p)$, d.i. bijvoorbeeld $p = \infty$, een nulpunt $z = -1$ overeenstemmen;
4. Pas de versterkingsfactor van de digitale regelaar aan zodat hij voor een bepaalde frequentie overeenstemt met die van de continue regelaar. Meestal wordt een gelijke statische versterkingsfactor gekozen of

$$H(p)|_{p=0} = H(z)|_{z=1}\,. \tag{4.7}$$

Methode 2: voorbeeld 1

$$H(p) = \frac{a}{p+a} \quad \text{levert} \quad H(z) = \frac{(z+1)\left(1-e^{-aT}\right)}{2(z-e^{-aT})}$$

Methode 2: voorbeeld 2

Bepaal de discrete PI-regelaar door transformatie van polen en nulpunten. De TF

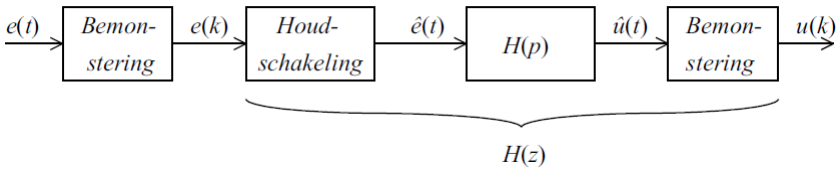
$$H_{PI}(p) = Kr\left(1+\frac{1}{\tau_i p}\right) = Kr\left(\frac{\tau_i p+1}{\tau_i p}\right) = Kr\left(\frac{p+1/\tau_i}{p}\right)$$

levert

$$H_{PI}(z) = C \cdot Kr\left(\frac{z-e^{-T/\tau_i}}{z-1}\right)$$

Methode 3: 'houd'-equivalent

- Beschouw benaderende discrete signalen $e(t)$ en $u(t)$



$$= \hat{e}(t) = e(k) \quad \text{voor} \quad kT \leq t < (k+1)\,T.$$

- Bereken overeenstemmende $H(z)$
  - Beschouw eenheidsimpuls aan de ingang

$$e(k) \begin{cases} = 1 & \text{als} \quad k = 0; \\ = 0 & \text{als} \quad k \neq 0. \end{cases} \quad \leftrightarrow \quad E(z) = 1.$$

Methode 3: 'houd'-equivalent

- $\hat{e}(t)$ bestaat dan uit een eenheidsstap op $t = 0$, en een negatieve eenheidsstap op $t = T$.

$$\hat{U}(p) = \frac{1}{p}H(p) - \frac{e^{-pT}}{p}H(p)$$

- Inverse Laplace

$$\hat{u}(t) = L^{-1}\left\{\frac{H(p)}{p}\right\} - L^{-1}\left\{\frac{H(p)}{p}e^{-pT}\right\}$$

- Na Z-transformatie

$$U(z) = H(z).E(z) = Z\left\{L^{-1}\left\{\frac{H(p)}{p}\right\}\right\} - z^{-1}.Z\left\{L^{-1}\left\{\frac{H(p)}{p}\right\}\right\}$$
$$= \left(1 - z^{-1}\right).Z\left\{L^{-1}\left\{\frac{H(p)}{p}\right\}\right\}$$

- *U(z)=H(z)E(z)=H(z)*

Methode 3: voorbeeld

$$H(p) = \frac{a}{p+a}$$

Dan is

$$\frac{H(p)}{p} = \frac{1}{p} - \frac{1}{p+a}$$

De invers Laplace-getransformeerde hiervan, met toepassing van de bemonstering, is

$$L^{-1}\left\{\frac{H(p)}{p}\right\} = u(t)\left(1 - e^{-at}\right) \overset{t=kT}{\rightarrow} = u(k)(1 - e^{-akT})$$

met $u(k)$ en $u(t)$ de éénheidstapfuncties. Toepassing van de Z-transformatie en vermenigvuldiging met $(1 - z^{-1})$ levert uiteindelijk

$$H(z) = (1 - z^{-1})\left(\frac{z}{z-1} - \frac{z}{z - e^{-aT}}\right) = \frac{1 - e^{-aT}}{z - e^{-aT}}$$

Methode 3: voorbeeld

- Kan ook door toepassing van transformatietabel
- Vergeet niet te delen door *p*
- En te vermenigvuldigen met $(1 - z^{-1})$

	$f(t), f(kT)$	$F(p)$	$F(z)$
1	$\delta(t)$	1	1
2	$\delta(t - nT)$	e^{-nTp}	z^{-n}
3	$u(t)$	$\frac{1}{p}$	$\frac{z}{(z-1)}$
4	t	$\frac{1}{p^2}$	$\frac{Tz}{(z-1)^2}$
5	$\frac{1}{2}t^2$	$\frac{1}{p^3}$	$\frac{Tz(z+1)}{2(z-1)^3}$
6	e^{-at}	$\frac{1}{p+a}$	$\frac{z}{(z-e^{-aT})}$
7	te^{-at}	$\frac{1}{(p+a)^2}$	$\frac{Tze^{-aT}}{(z-e^{-aT})^2}$
8	$\frac{1}{2}t^2e^{-at}$	$\frac{1}{(p+a)^3}$	$\frac{T^2ze^{-aT}(z+e^{-aT})}{2(z-e^{-aT})^3}$
9	$1 - e^{-at}$	$\frac{a}{p(p+a)}$	$\frac{(1-e^{-aT})z}{(z-1)(z-e^{-aT})}$

Stabiliteit en nauwkeurigheid

- Blijft de stabiliteit van discreet equivalent behouden?

Stabiliteit en nauwkeurigheid

- Numerieke integratie: transformatie polen van P naar Z domein

a) voorwaartse rechthoekregel : $z = 1 + Tp$;

b) achterwaartse rechthoekregel : $z = \dfrac{1}{1 - Tp}$;

c) trapeziumregel : $z = \dfrac{2 + Tp}{2 - Tp}$.

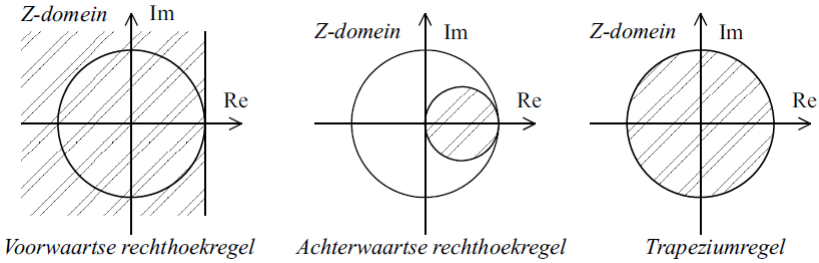
- Gelijkstelling p $= j\omega$ geef aan hoe stabiliteitsgrens getransformeerd wordt

a) $z = 1 + j\omega T$, d.w.z. de verticale as door 1;

b) $z = \dfrac{1}{1 - jT\omega} = \dfrac{1}{2} + \left(\dfrac{1}{1 - jT\omega} - \dfrac{1}{2}\right) = \dfrac{1}{2} + \dfrac{1}{2}\left(\dfrac{1 + jT\omega}{1 - jT\omega}\right)$, d.w.z. een cirkel met middelpunt $1/2$ en straal $1/2$;

c) $z = \dfrac{1 + jT\omega/2}{1 - jT\omega/2}$, d.w.z. een cirkel met middelpunt 0 en straal 1.

Stabiliteit en nauwkeurigheid



Stabiliteit en nauwkeurigheid

- Polen nulpunten transformatie
  - Behoud van stabiliteit is per definitie gegarandeerd $z = e^{pT}$.
- Houd-equivalent
  - Indien $H(p)$ stabiel dan $H(z)$ ook stabiel. Enkel input signaal wordt aangepast $\hat{e}(t)$ ipv $e(t)$.

Nauwkeurigheid

- Hoe goed benadert de discrete TF de continue TF
- Het discreet equivalent is slechts geldig in het frequentiegebied (0, $\frac{f_s}{2}$).

	Numerieke integratie regels			Polen-nulpunten-transform.	Houd-equivalent
	Voorwaartse rechthoek	Achterwaartse rechthoek	Trapezium		
Behoud Stab.	Neen	ja	Ja	Ja	Ja
Nauwkeurigheid	slecht	slecht	matig	Goed	Goed

Vergelijkende oefening

- Bepaal het discreet equivalent van een eerste orde systeem met τ = 1s. Vergelijk de TF's, de stapresponsies en de frequentieresponsies.

Oplossing: Neem de bemonsteringsperiode $T = 0,1$ s.

1. Numerieke integratie, trapeziumregel:

$$p = \frac{2\,(z-1)}{T\,(z+1)} \text{zodat} H(p) = \frac{1}{1+p} \quad \rightarrow \quad H(z) = \frac{z+1}{21z-19}$$

2. Nulpunten polen transformatie:

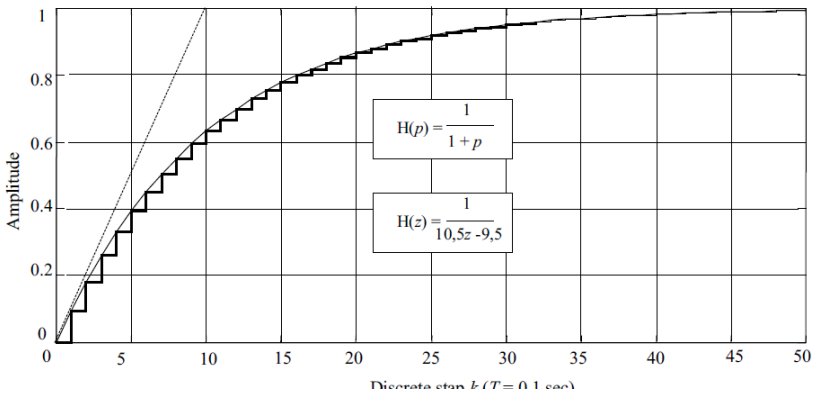
$$H(z) = \frac{(1-e^{-0,1})\,(z+1)}{2\,(z-e^{-0,1})} = \frac{z+1}{21,017z-19,017}$$

3. Houdequivalent:

$$H(z) = \frac{1-e^{-0,1}}{z-e^{-0,1}} = \frac{1}{10,508z-9,507}$$

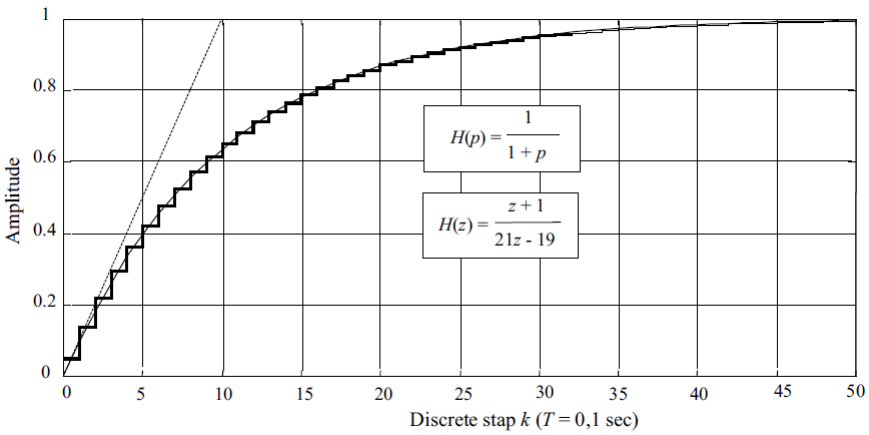
Vergelijkende oefening

- Vergelijking met houdequivalent



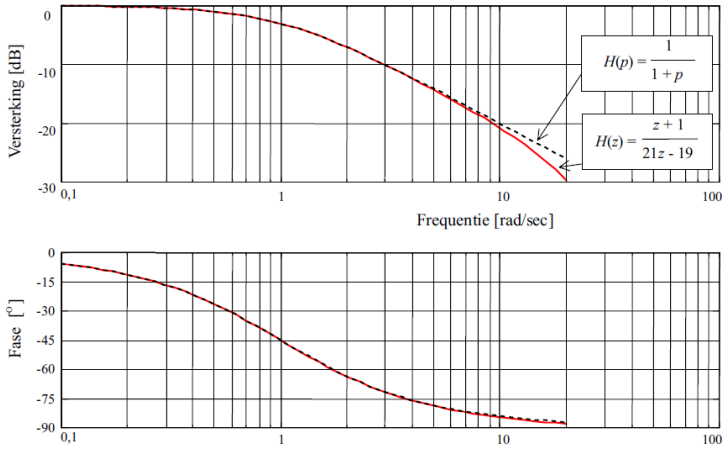
Vergelijkende oefening

- Vergelijking met nulpunten polen transformatie en numerieke integratie



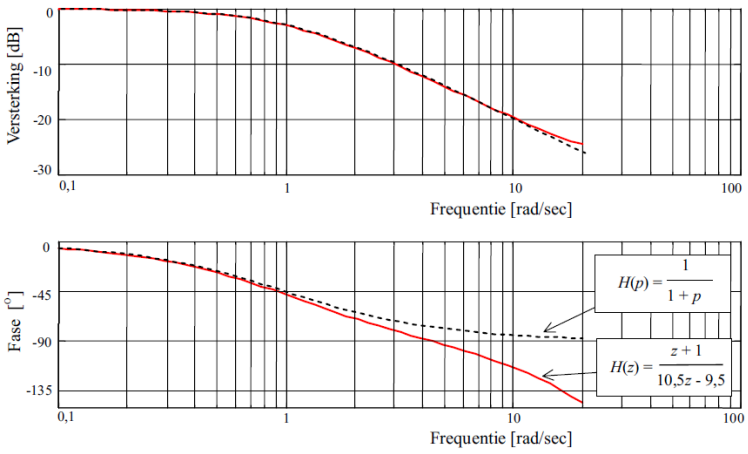
Vergelijkende oefening

- Vergelijking met numerieke integratie



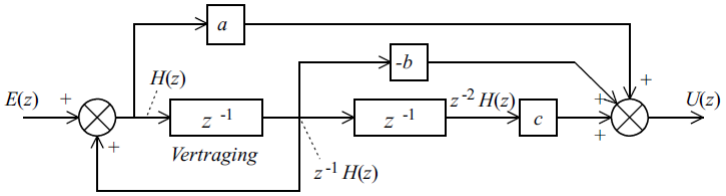
Vergelijkende oefening

- Vergelijking met houdequivalent



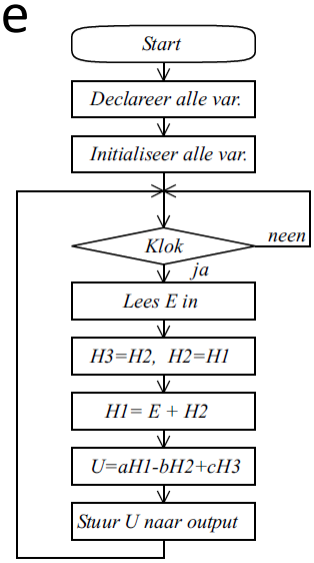
Implementatie

- Simulatie schema beschrijft de werking van regelaar



Implementatie

- Omzetten naar flowchart

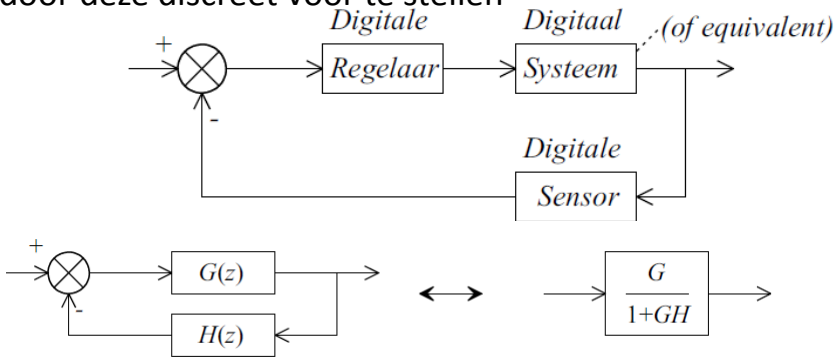


Automatisering

Discrete regelkring

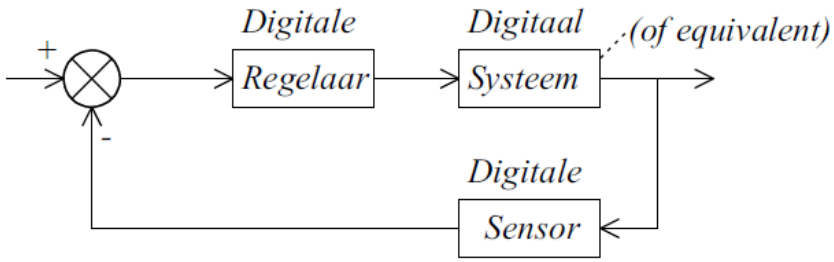
Inleiding

- Indien regelkring volledig discreet → ontwerp regelaar rechtstreeks in z-domein
- Kan ook toegepast worden op continue processen door deze discreet voor te stellen



Inleiding

- Volledig equivalent met continue tegenhanger. Daarom hanteren we hier dezelfde formules mbt open en gesloten TF's



Stabiliteit

- Systeem is stabiel wanneer de polen binnen de eenheidscirkel vallen.
- Instelling van de regelaar bepaalt de ligging van de polen van het gesloten systeem.
- Versterkingsfactor K_r van de regelaar speelt belangrijke rol. → onderzoek invloed dmv wortellijnendiagram
- Wortellijnendiagram wordt op dezelfde wijze geconstrueerd als bij continue systemen.

$$1 + KG(z)H(z) = 0 \text{ voor } K = 0, \dots, \infty$$

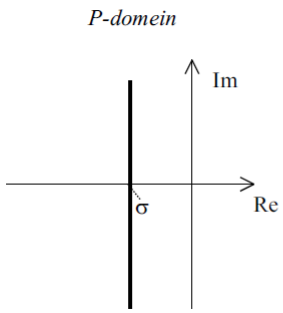
Systeemkarakteristieken

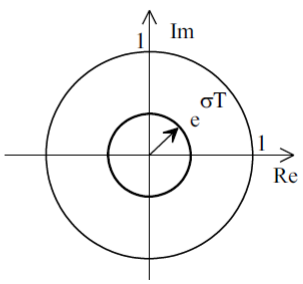
- Systeemkarakteristieken aflezen op complexe cirkel door:
  - lijnen van constante settling time (σ)
  - lijnen van constante dempingsfactor (ζ)
  - lijnen van constante gedempte eigenpulsatie (ω_p)
- We kennen deze in het P-domein
- Zet nu om naar Z-domein met volgende vergelijkingen:

  $z = e^{pT}$ met $p = \sigma + j\omega_p$

Lijnen van constante σ

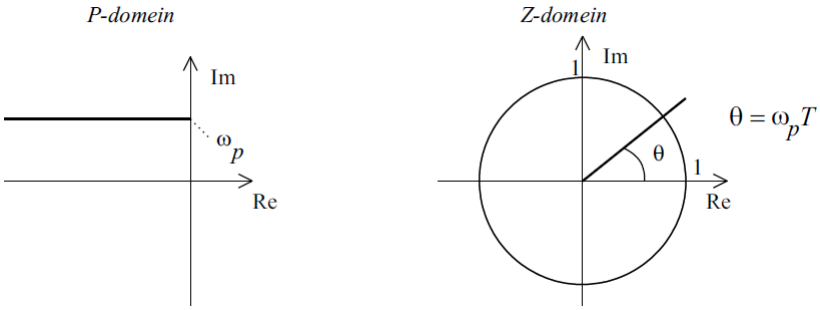
- $L^{-1}\left\{\frac{1}{p+\sigma}\right\} = e^{\sigma t}$



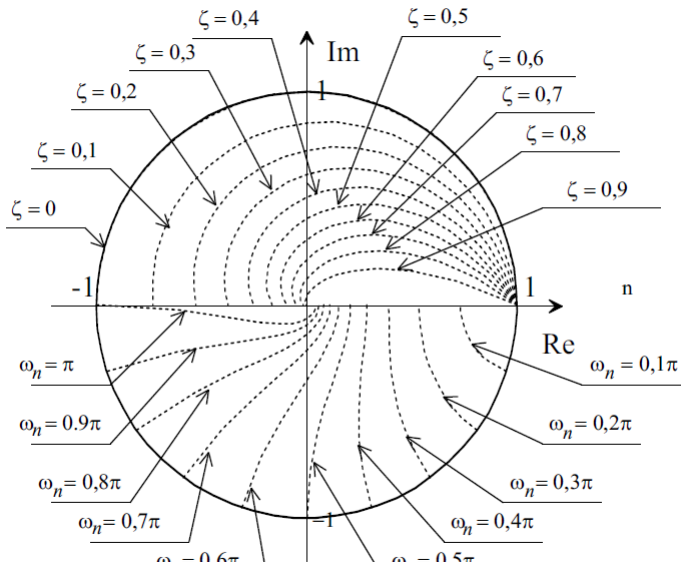


Lijnen van constante ω_p

- Grotere ω_p komt overeen met (bij vaste F_s) met grotere θ
- $\omega_p = \omega_n\sqrt{1-\zeta^2}$

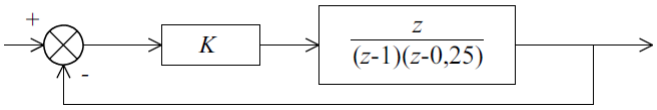


Lijnen van constante ζ en ω_n

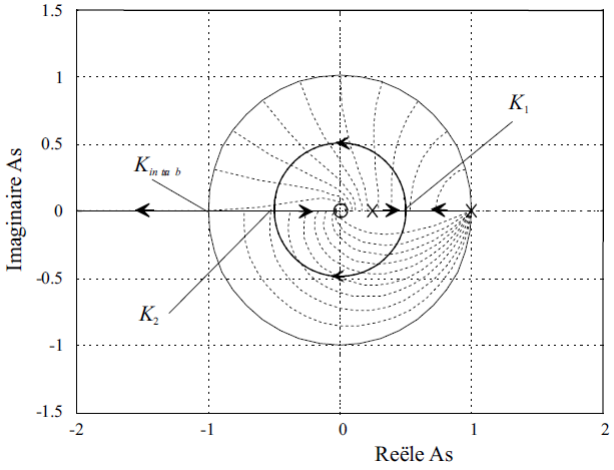


Voorbeeld 1

1. Bepaal K_m om onderstaande regelkring marginaal stabiel te maken.
2. Bepaal K_{ω_p} waarbij $\omega_p = 1r/s$ bij $F_s = 1Hz$.
3. Maak gebruik van het wortellijnendiagramma



Voorbeeld 1

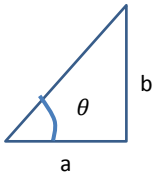


$$K_m = -\frac{1}{G(z)H(z)}\bigg|_{z=-1} = 2{,}5$$

Voorbeeld 1

- $K_{\omega_p} = ?$

$$\theta = \omega_p T = 1\text{r} = \frac{360}{2\pi} = 57{,}3°$$



- Merk op:
  - $cos\theta = \frac{a}{\sqrt{a^2+b^2}}$ (**)
  - Karakteristieke vgl: $z^2 + (K - 1{,}25)z + 0{,}25 = 0$
- Zoek K_{ω_p} zodat karakteristieke vgl. bepaald wordt door 2 complex toegevoegde polen waarbij $\theta = 57{,}3°$

$$\big(z - (a - jb)\big)\big(z - (a + jb)\big) = z^2 - 2az + a^2 + b^2 = 0$$

Voorbeeld 1

$$z^2 - 2az + a^2 + b^2 = z^2 + \left(K_{\omega_p} - 1{,}25\right)z + 0{,}25$$

$$\begin{cases} \left(K_{\omega_p} - 1{,}25\right) = -2a \\ a^2 + b^2 = 0{,}25 \end{cases}$$

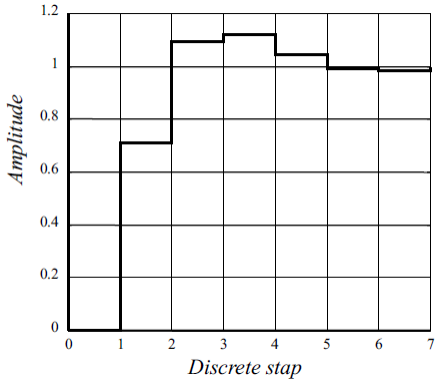
samen met (**) geeft dit

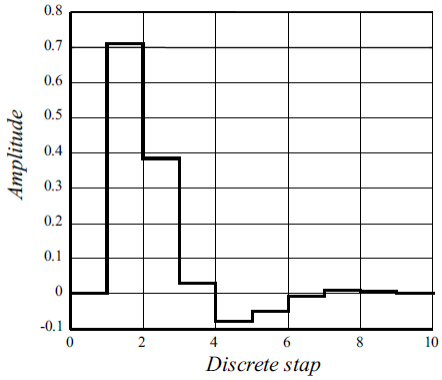
$$cos\theta = -\frac{K_{\omega_p} + 1{,}25}{2\sqrt{0{,}25}} \rightarrow K_{\omega_p} = -2\sqrt{0{,}25}\cos\theta + 1{,}25$$

$$\rightarrow K_{\omega_p} = 0{,}71$$

Voorbeeld 1

Bij K_{ω_p} krijg je volgende stap- en impulsrespons





Voorbeeld 2

Bepaal K_m voor marginale stabiliteit bij regeling van systeem $\frac{K}{(z-0{,}1)(z-0{,}7)}$.

$$z^2 - 2az + a^2 + b^2 = 0$$
$$z^2 - 0{,}8z + 0{,}07 + K_m = 0$$

$$\begin{cases} (0{,}07 + K_m) = a^2 + b^2 \\ \sqrt{a^2 + b^2} = 1 \ (polen\ op\ eenheidscirkel) \end{cases}$$
$$K_m = 0{,}93$$

State-space based controller

P. Karsmakers

KATHOLIEKE UNIVERSITEIT LEUVEN

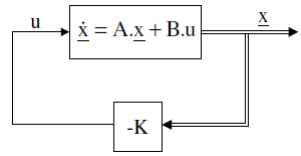
KHK
KATHOLIEKE HOGESCHOOL KEMPEN

Outline

- State-space models and controllers
- Example: Inverted Pendulum

1. STATE-SPACE MODELS AND CONTROLLERS

1. Introduction



- Outputs do not necessarily describe a system fully
- It is the current states of the system that fully describe it which, together with the inputs, determine its future condition.
- Using the states for feedback therefore offers the potential for improved control

General state-space model of a linear system

$$\begin{cases} \dot{\underline{x}}(t) = A(t)\underline{x}(t) + B(t)\underline{u}(\mathrm{t}) \\ \underline{y}(t) = C(t)\underline{x}(t) + D(t)\underline{u}(\mathrm{t}) \end{cases}$$

- $\underline{x}(.) \in \mathbb{R}^n$ (state vector)
- $\underline{y}(.) \in \mathbb{R}^q$ (output vector)
- $\underline{u}(.) \in \mathbb{R}^p$ (input or control vector)
- $A(.) \in \mathbb{R}^{n\times \mathrm{n}}$ (system matrix)
- $B(.) \in \mathbb{R}^{n\times p}$ (input matrix)
- $C(.) \in \mathbb{R}^{q\times \mathrm{n}}$ (output matrix)
- $D(.) \in \mathbb{R}^{q\times p}$ (feedforward matrix)
- $\dot{\underline{x}}(t) \coloneqq \frac{dx(t)}{dt}$

From state-space to transfer function

$$\begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(\mathrm{t}) \\ \underline{y}(t) = C\underline{x}(t) + D\underline{u}(\mathrm{t}) \end{cases}$$

Turn this into TF by taking $y(s)/u(s)$.

Doing this gives

$$s\underline{x}(s) = A\underline{x}(s) + B\underline{u}(s)$$

Which implies that

$$\underline{x}(s) = (sI - A)^{-1}B\underline{u}(s)$$

We also have

$\underline{y}(s) = C\underline{x}(s) + D\underline{u}(s)$

And substituting for $\underline{x}(s)$ gives

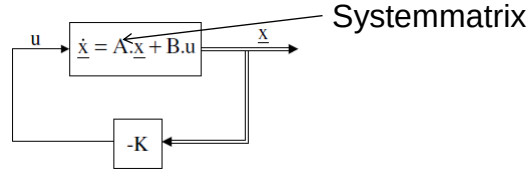
$$TF = \frac{\underline{y}(s)}{\underline{u}(s)} = C(sI - A)^{-1}B + D$$

From state-space to transfer function: example

- TF=$\frac{\alpha}{s+\alpha}$, $\alpha > 0$ and take the initial conditions to be zero ($y(t) = 0$ for $t = 0$)
- Differential equation is $\frac{dy(t)}{dt} + \alpha y(t) = \alpha u(t)$
- Take state variable $x(t) = y(t)$, then $A = -\alpha, B = \alpha$, $C = 1$ and $D = 0$
- TF=$C(sI - A)^{-1}B + D = 1\big(s - (-\alpha)\big)^{-1} + 0 = \frac{\alpha}{s+\alpha}$

1.1 Control Law

- The simplest form of state-feedback



- Presumed that all of the states can be measured
- The dynamics of the sensors and actuators are either assumed to be negligible or are included in the state-space description of the process and/or plant

1.1 Control Law

- Suppose SISO system:
  - Input u is a scalar
  - Controller matrix K is a row vector of gains
  - State vector $\underline{x}$

$$u = -K\underline{x} = -\begin{bmatrix} k_1 & k_2 & \cdots & k_n \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

- Why do we need the minus sign?

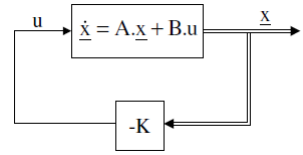
1.1 Control Law

- With blockdiagram algebra we have

$$\dot{\underline{x}} = A\underline{x} + Bu = A\underline{x} - B.K\underline{x}$$

$$s\underline{x}(s) - \underline{x}(0) = A\underline{x}(s) - BK\underline{x}(s)$$

$$\underline{x}(s) = (sI - (A - BK))^{-1} .\underline{x}(0)$$



- The characteristic equation for the closed-loop system is thus

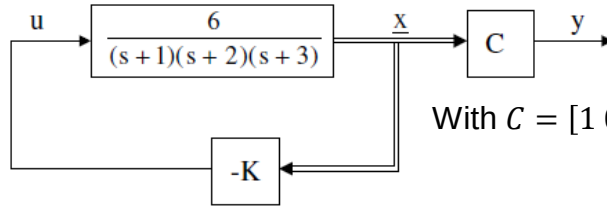
$$\det [sI - (A - BK)] = 0$$

- For n-th order system this is a n-th order polynomial with n closed loop poles and n controller gains

- **Objective**: Determine gains that produce poles with specified closed loop performance → pole placement
- In MATLAB: `K = place(A,B,p)`

1.1 Control Law

❑ Example:



With $C = [1\ 0\ 0]^T$

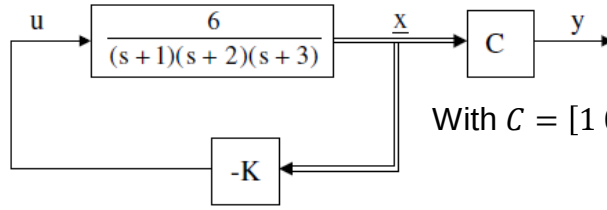
Companion form

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} .u$$

$$y = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

1.1 Control Law

❑ Example:



With $C = [1\ 0\ 0]^T$

$$\det\left[\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ 6 \end{bmatrix} \begin{bmatrix} k_1 & k_2 & k_3 \end{bmatrix}\right)\right] = 0$$

which expands to give

$$s \begin{vmatrix} s & -1 \\ 11 + 6k_2 & s + 6(1 + k_3) \end{vmatrix} + 6(1 + k_1) \begin{vmatrix} -1 & 0 \\ s & -1 \end{vmatrix} = 0$$

which in turn results in

$$s^3 + 6(1 + k_3)s^2 + (11 + 6k_2)s + 6(1 + k_1) = 0$$

1.1 Control Law

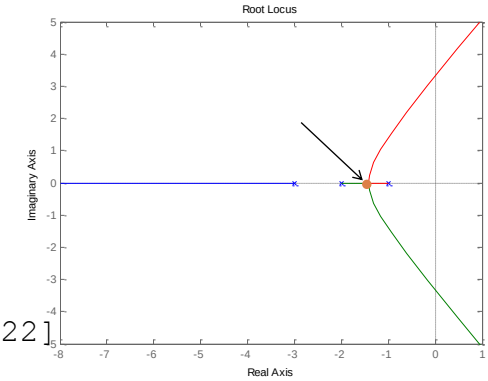
- Pole placement
  - a) Choose critically damped response. How?

```
>> sys=tf([6],[1 6 11 6])

Transfer function:
           6
-----------------------
s^3 + 6 s^2 + 11 s + 6

>> [K,poles]=rlocus(sys);

poles=[-3.155 -1.422 -1.422]
```



1.1 Control law

- Since critically damped response is desired, the following characteristic equation is needed

$$(s + 3.155)(s + 1.423)(s + 1.422) = 0$$
$$s^3 + 6.0s^2 + 11.0s + 6.384 = 0$$

Comparing with

$$s^3 + 6(1 + k_3)s^2 + (11 + 6k_2)s + 6(1 + k_1) = 0$$

gives

$$6(1 + k_1) = 6.384$$
$$11 + 6k_2 = 11 \qquad \text{hence } K = [\,0.064 \;\; 0.0 \;\; 0.0\,]$$
$$6(1 + k_3) = 6$$

Why only one K element non-zero? Why is k_1 non-zero?

1.1 Control law

- b) suppose underdamped response is required with roots not lying on the root-locus of the original closed-loop system. Choose poles with damping factor of about 0.8. How?

$$(s + 5.0)\left(s + 0.4 + j0.3\right)\left(s + 0.4 - j0.3\right) = 0$$

$$s^3 + 5.8s^2 + 4.25s + 1.25 = 0$$

Comparing with

$$s^3 + 6\,(1 + k_3)\,s^2 + (11 + 6k_2)\,s + 6\,(1 + k_1) = 0$$

gives

$$6(1 + k_1) = 1.25$$
$$11 + 6k_2 = 4.25 \quad \text{hence } K = -\begin{bmatrix} 0.792 & 1.125 & 0.033 \end{bmatrix}$$
$$6(1 + k_3) = 5.8$$

1.1 Control law

- Comments:
  - Not essential to use companion form. However, in a different form a set of linear equations need to be solved
  - Magnitude of elements of controller matrix K → indication of appropriateness of chosen pole locations
  - Straight forward generalisation to MIMO, e.g.

$$\begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = K\underline{x} = \begin{bmatrix} k_{11} & k_{12} & \cdots & k_{1n} \\ k_{21} & k_{22} & \cdots & k_{2n} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

From state-space to transfer function

$$\begin{cases} \dot{\underline{x}}(t) = A\underline{x}(t) + B\underline{u}(\mathrm{t}) \\ \underline{y}(t) = C\underline{x}(t) + D\underline{u}(\mathrm{t}) \end{cases}$$

Turn this into TF by taking $y(s)/u(s)$.

Doing this gives

$$s\underline{x}(s) = A\underline{x}(s) + B\underline{u}(s)$$

Which implies that

$$\underline{x}(s) = (sI - A)^{-1}B\underline{u}(s)$$

We also have

$\underline{y}(s) = C\underline{x}(s) + D\underline{u}(s)$

And substituting for $\underline{x}(s)$ gives

$$TF = \frac{\underline{y}(s)}{\underline{u}(s)} = C(sI - A)^{-1}B + D$$

From state-space to transfer function: example

- TF=$\frac{\alpha}{s+\alpha}$, $\alpha > 0$ and take the initial conditions to be zero ($y(t) = 0$ for $t = 0$)
- Differential equation is $\frac{dy(t)}{dt} + \alpha y(t) = \alpha u(t)$
- Take state variable $x(t) = y(t)$, then $A = -\alpha, B = \alpha$, $C = 1$ and $D = 0$

- TF=$C(sI - A)^{-1}B + D = 1\big(s - (-\alpha)\big)^{-1} + 0 = \frac{\alpha}{s+\alpha}$

1.2 Observer Design

- The control law assumed that all the states were available for feedback
- Generally not the case, largely due to the limitations of measurement
- In such situations it is necessary to estimate the states and to use the estimate of the state vector for feedback purposes
- An observer is the name given to an estimator that is used for estimating the states of a system

1.2 Observer Design

- From a knowledge of the A and B matrices of the system, a state-space model of the process based upon estimates of its states is of the form:

  $\dot{\underline{\hat{x}}} = A.\underline{\hat{x}} + B.u$

  where ˆ denotes an estimate
- Knowing the real inputs to the process, and assuming the initial conditions to be zero, an estimate of its states could be found directly by solving the model:

  $\underline{\hat{x}}(t) = L^{-1}\left\{(sI - A)^{-1} B.u(s)\right\}$ $\quad \hat{x} = \int_0^t e^{A(t-\tau)} Bu(\tau) d\tau$
- In practice, the inputs vary with time and the solution has to be found by numerical integration

1.2 Observer Design

- Provided that the A matrix is accurate and the system is stable, then the estimates of the states will converge numerically on their true values. This can be demonstrated by considering the errors on the estimates of the states, their residuals. Let:

$$\tilde{\underline{x}} = \underline{x} - \hat{\underline{x}}$$

where ˜ denotes the residual. Differentiating gives

$$\begin{aligned}\dot{\tilde{\underline{x}}} &= \dot{\underline{x}} - \dot{\hat{\underline{x}}} \\ &= A.\underline{x} + B.u - \left(A.\hat{\underline{x}} + B.u\right) \\ &= A.\left(\underline{x} - \hat{\underline{x}}\right) \\ &= A.\tilde{\underline{x}}\end{aligned}$$

in time domain this becomes

$$\tilde{\underline{x}} = e^{At}.\tilde{\underline{x}}(0) \quad \text{where} \quad e^{At} = L^{-1}\left\{[sI - A]^{-1}\right\}$$

- Thus, provided the roots are negative real, or have negative real parts the solution is stable and converges.

$$\det[sI - A] = 0$$

1.2 Observer Design

- Some details about $\underline{x} = e^{At}.\underline{x}(0)$

$$\Phi(t) = e^{At} = I + A.t + \frac{A^2t^2}{2} + \frac{A^3t^3}{3!} + \ldots$$

(Taylor expansion, also def. matrix exponential)

$$\begin{aligned}\Phi(s) &= L\left\{e^{At}\right\} \\ &= \frac{I}{s} + \frac{A}{s^2} + \frac{A^2}{s^3} + \frac{A^3}{s^4} + \ldots \\ [sI - A]\,\Phi(s) &= [sI - A] \\ &\quad \times \left[\frac{I}{s} + \frac{A}{s^2} + \frac{A^2}{s^3} + \frac{A^3}{s^4} + \ldots\right] \\ &= I^2 + \frac{A}{s} + \frac{A^2}{s^2} + \frac{A^3}{s^3} + \ldots \\ &\quad - \frac{A}{s} - \frac{A^2}{s^2} - \frac{A^3}{s^3} - \ldots \\ &= I\end{aligned}$$

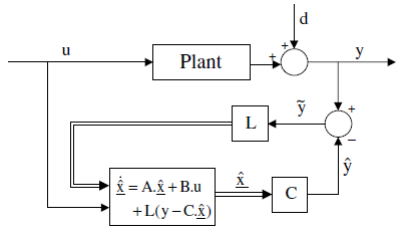
$$\Phi(s) = [sI - A]^{-1}$$

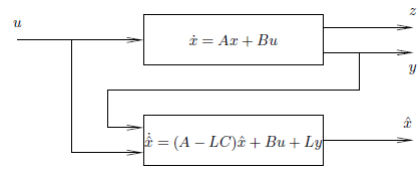
hence

$$\Phi(t) = e^{At} = L^{-1}\left\{[sI - A]^{-1}\right\}$$

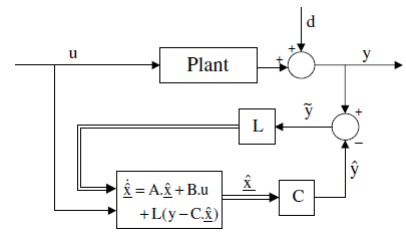
1.2 Full order observer

- The problem with the previous predictor is that the inputs are seldom known accurately and there is inevitably some plant/model mismatch, i.e. errors in the A and B matrices
- These always result in offset and can lead to numerical instability
- Improve upon convergence using the predictor-corrector approach
- General structure (using output feedback) of a so-called "full order observer" for a SISO system is





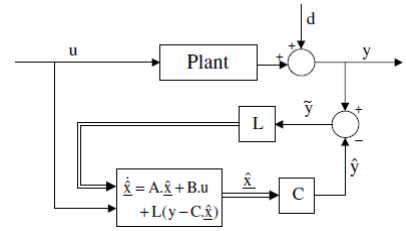
1.2 Full order observer



- The predictor-corrector is comprised of the state space model of the plant, i.e. its system and input matrices A and B respectively, with an extra term for correction purposes
- The corrector is in proportion to the size of the prediction error. Remember that the dynamics of the instrumentation, i.e. of the actuator and of the measurement must be included in the model:

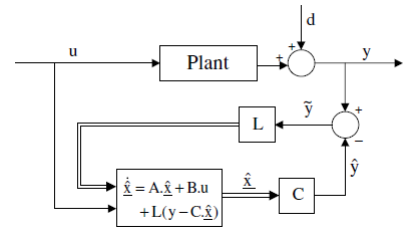
$$\begin{aligned}\dot{\hat{\underline{x}}} &= A.\hat{\underline{x}} + B.u + L.\tilde{y} \\ &= A.\hat{\underline{x}} + B.u + L.(y - \hat{y}) \\ &= A.\hat{\underline{x}} + B.u + L.(y - C.\hat{\underline{x}})\end{aligned}$$

1.2 Full order observer



- The observer matrix L is a column vector of gains, or weighting factors, and is the principal design parameter of the observer.
- Basic philosophy:
  - estimate all the states, whether they can be measured or not.
  - Then, by driving the residual on that state which is measured to zero (e.g. y), the estimates of all the states will be driven towards their actual values

1.2 Full order observer



- Let the residuals converge

$$\tilde{x} = x - \hat{x}$$

$$\dot{\tilde{x}} = \dot{x} - \dot{\hat{x}}$$

Substituting from $\dot{\hat{x}} = A.\hat{x} + B.u + L.(y - C.\hat{x})$

$$\dot{\tilde{x}} = A.x + B.u - (A.\hat{x} + B.u + L.(y - C.\hat{x}))$$

$$= A.(x - \hat{x}) - L.(C.x - C.\hat{x})$$

$$= (A - L.C).\tilde{x}$$

with time-domain eq.

$$\tilde{x}(t) = L^{-1}\left\{(sI - (A - LC))^{-1}\right\}\tilde{x}(0)$$

for which the characteristic equation is: $\det[sI - (A - L.C)] = 0$

- Provided the observer's eigenvalues are negative real, or have negative real parts, stable convergence is assured.

1.2 Full order observer

- Example:
  - Consider previous example
  - 3th order system
  - For the observer's response to be stable choose poles on the negative real axis: this will give exponential decay of the residuals.
  - For the decay to be fast choose pole positions that correspond to lags that are of similar magnitude to the smallest lags of the system itself.

1.2 Full order observer

- Example:
  - Aim at obtaining $(s+4)(s+3)^2 = 0$

$$s^3 + 10s^2 + 33s + 36 = 0$$

the characteristic polynomial for the observer design is

$$\det\left[\begin{bmatrix} s & 0 & 0 \\ 0 & s & 0 \\ 0 & 0 & s \end{bmatrix} - \left(\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ -6 & -11 & -6 \end{bmatrix} - \begin{bmatrix} l_1 \\ l_2 \\ l_3 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}\right)\right] = 0$$

which yields

$$s^3 + (6 + l_1).s^2 + (11 + 6.l_1 + l_2).s + (6 + 11.l_1 + 6.l_2 + l_3) = 0$$

comparing coefficients

$$\begin{aligned} 6 + l_1 &= 10 \\ 11 + 6.l_1 + l_2 &= 33 \\ 6 + 11.l_1 + 6.l_2 + l_3 &= 36 \end{aligned} \rightarrow L = \begin{bmatrix} 4 \\ -2 \\ -2 \end{bmatrix}$$

1.3 Reduced Order Observer

- It is not necessary to estimate all the states of a system if there are good quality, noise free measurements of some of the states available. In such cases estimating a reduced order observer is sufficient.

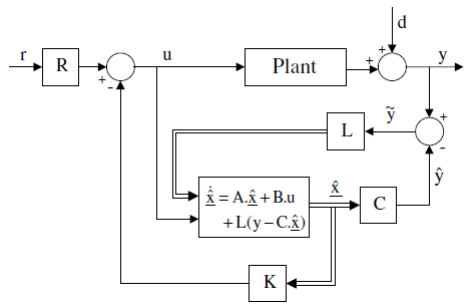
$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} B_1 \\ B_2 \end{bmatrix} .u$$

where subscript 1 denotes known (measured states) and 2 denotes unknown states

- Saves on the computation effort involved in the estimation process.
- However, if there is any doubt about the quality of the measurements, it is probably best to use a full order observer. It is also easier to understand!

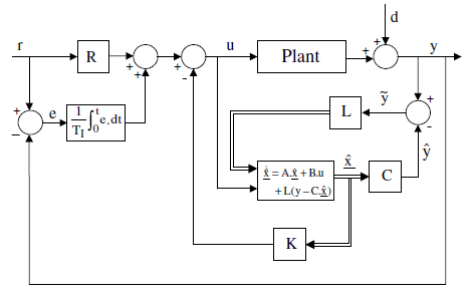
1.4 Integration of observer and controller

- Integration of the controller, set-point compensation and observer



1.4 Integration of observer and controller

- It is inevitably the case that plant/model mismatch occurs, *i.e.* there are errors in the A and B matrices. These result in offset. This can be countered by introducing some integral action



- In general, only small integral action

1.5 Implementation

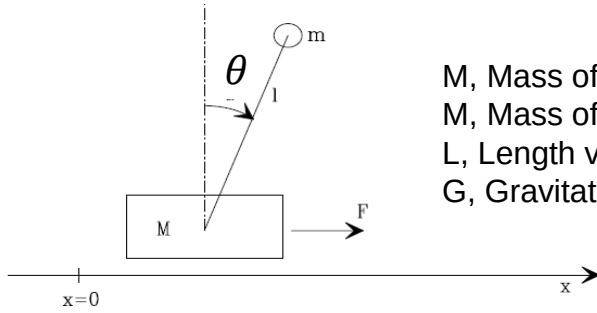
- All signals are presumed in deviation form. Thus, for realisation, steady state biases have to be added to or subtracted from the various signals as appropriate.
- Solution of the observer is found by numerical integration (e.g. Eulers implicit method) where $f_s \approx \Delta t$ (Δt is step length for numerical integration)
- At each sampling instant, knowing y and u of plant, the numerical integration would be executed once to predict the states. These are then used to calculate the next value of the controller output signal.

1.5 Implementation

- Four time-scales are involved:
  - *predictor-corrector*, for stability, $\Delta t < \frac{smallest\ lags\ of\ system}{10}$
  - *observer L*, dominant pole positions are chosen consistent with the fastest dynamics (smallest lags) of the system. A stable overdamped response is essential. Observer's dynamics > 5* controller's dynamics
  - *controller K*, dominant pole positions are chosen that are consistent with the slowest dynamics (largest lags) of the system. A stable underdamped response is typically the requirement.
  - *Integrator,* integral time T_I must be consistent with the dynamics of the integral action being even more sluggish than the largest lags of the system.

3. EXAMPLE: INVERTED PENDULUM

3 Example: inverted pendulum



M, Mass of cart = 0.5 kg
M, Mass of pendulum = 0.2 kg
L, Length van het pendulum = 0.3 m
G, Gravitation = 9.8 m/s2

- Non-linear differential equation

$$(M+m)\ddot{x} + m\ddot{\theta}lcos\theta - ml\dot{\theta}^2\sin\theta = F \quad (1)$$
$$mlcos\theta\ddot{x} + m\ddot{\theta}l^2 - mglsin\theta = 0 \quad (2)$$

2.5 Example: inverted pendulum

- Single out $\ddot{x}$ in eq. (2) and substitute in (1)

$$\ddot{\theta} = \frac{F + ml\dot{\theta}^2 \sin\theta - \frac{(M+m)g\sin\theta}{\cos\theta}}{ml\cos\theta - \frac{(m+M)l}{\cos\theta}}$$

After linearisation around set-point (werkingspunt) $\theta \approx 0$ ($\sin\theta \approx \theta$ en $\cos\theta \approx 1, \dot{\theta}^2 \approx 0$) this gives

$$\ddot{\theta} = \frac{(M+m)g\theta - F}{Ml} \quad (3)$$

2.5 Example: inverted pendulum

- Single out $\ddot{\theta}$ in eq. (2) and substitute in (1)

$$\ddot{x} = \frac{-mg \sin\theta \cos\theta + ml\dot{\theta}^2 \sin\theta + F}{M + m - m(\cos\theta)^2}$$

After linearisation around the setpoint (werkingspunt) $\theta \approx 0$ ($\sin\theta \approx \theta$ en $\cos\theta \approx 1, \dot{\theta}^2 \approx 0$) geeft dit

$$\ddot{x} = \frac{F - mg\theta}{M} \qquad (4)$$

2.5 Example: inverted pendulum

- With (3) and (4) the TFs $\frac{X(s)}{F(s)}, \frac{\theta(s)}{F(s)}$ can be found

$$\frac{\theta}{F} = -\frac{1}{Mls^2 - (M+m)g}$$
$$\frac{X}{F} = \frac{ls^2 - g}{Mls^4 - g(M+m)s^2}$$

2.5 Example: inverted pendulum

- Problem setting: make a controller such that the following constraints are fulfilled:
  - 10% max. overshoot on position of cart x
  - Max. 33° (0.58 rad) overshoot on pendulums angle
  - Rise time on x must be smaller than 5s
  - The settling time on x and θ must be smaller than 10 s.
  - The static error must be smaller than 2%

References

- Jonathan Love, *Process Automation Handbook: A systematic approach*, Springer, 2007.
- G. F. Franklin, J. D. Powell, and A. Emami-Naeini. *Feedback Control of Dynamic Systems*, Prentice Hall, Upper Saddle River, NJ, 4th edition, 2002.
- J. V. Vegte. *Feedback Control Systems*. Prentice Hall, New Jersey, 3rd edition, 1994.
- Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control. Prentice Hall*, New Jersey, 1996.

Fuzzy Logic

P. Karsmakers

KATHOLIEKE UNIVERSITEIT LEUVEN

KHK
KATHOLIEKE HOGESCHOOL KEMPEN

Outline

- Introduction
- Applications
- Basic concepts
- Extensions

1. Introduction

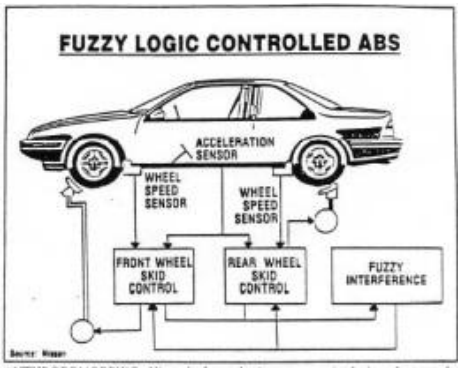
- Fuzzy-logic = extension of set theory (verzamelingenleer)
- In contrast to binary logic, concepts such as ’large’, ’warm’, ’a little bit more’ or ‘much too fast’ can be described
- No explicit definition of the transition from ’true’ to ’false’
- Humans can operate correctly using such a fuzzy description. E.g. driving a car
- Underlying theory isn’t fuzzy but well founded
- \+ Cases where Fuzzy logic might be used in control theory
  - non-linear control loops,
  - in cases where processes can’t be exactly modelled,
  - in applications where measurements are unreliable,
  - in case adaptive and learning systems are needed

1. Introduction

- \+ Further benefits:
  - Controller is adapted to human type of acting and reasoning
  - Robust against small deviations of the characteristics of a system
- \- Disadvantages:
  - No unique methodology to define a fuzzy controller
  - Practical experience required

2. Applications

- ABS Brakes
- Expert Systems
- Control Units: air conditioners, washing machines, traffic control systems
- Bullet train between Tokyo and Osaka
- Video Cameras
- Automatic Transmissions
- Temperature control of Viessmann heating devices
- see also: http://www.fuzzytech.com/e/e_a_esa.html



ANTHROPOMORPHIC: Nissan's fuzzy logic program is designed to apply human reasoning characteristics to the control module of its patented ABS design.

2. Applications



- **Temperature control in heating devices**

  "Thanks to Fuzzy Logic even small influences of the normal heating operation and peaks in the heat are quickly recognized. This allows a flexible scheme. The Fuzzy Logic system bases its conclusions on:

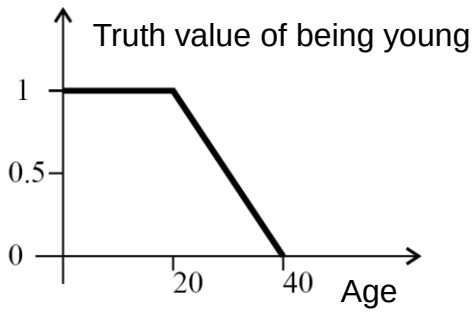
  - the course of the decrease in heat the day before
  - the course of the current heat demand
  - solar radiation

  Another advantage is the significantly simpler installation and operation. In order to set the desired room temperature, only one knob to be operated: warmer or colder. A setting of heating curve and the fitting of an outside temperature sensor are no longer needed. Fuzzy Logic is used in the Viessmann system Vitotronic 150.

3.1 Fuzzy set

❑ 3.1.1 Concept “fuzzy”

- In contrast to classical set theory also partial membership of a set possible
- Example: definition of “young”
  - ❍Exact definition:
    IF age<20 THEN young
  - ❍Fuzzy definition using the relation



3.1 Fuzzy set

❑ 3.1.2 Linguistic variable

- Variables in fuzzy logic are usually concepts
- e.g.: concept:
  ‘the ambient temperature is hot’ or shortly ‘hot’
- Concept ‘hot’ depends on the situation
- User has to define in which manner a temperature is hot
  e.g. using the figure on the previous slide

❑ 3.1.3 Membershipsdegree and -function

- Membershipsdegree
  - ❍Denotes the trustworthiness of a certain hypothesis
  - ❍Truth value that an elements belongs to a set
- Membershipsfunction
  - ❍Function that gives membershipsdegrees for all possible elements

- Definitions:
  - ❍*U*: universum, population of considered elements
  - ❍*u*: arbitrary element from *U*
  - ❍μ: membershipsdegree

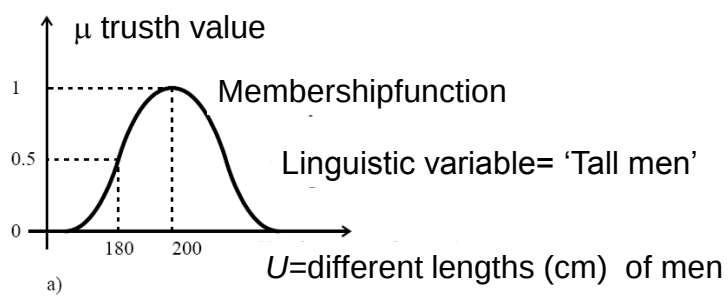
❍Example: concept “tall men”

A man with a length of 180 cm belongs with a truth value of 90% to a collection of tall men. A man of two meters belongs with a 100% truth value to the collection of tall men. However, a man of 210 cm long, has a truth of 70%. In the category of 'extreme' tall people he scored 100%.

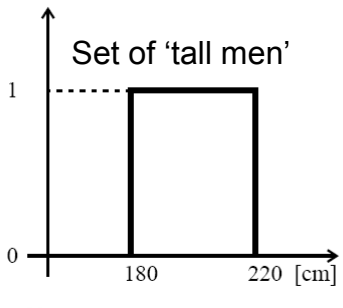
❍No exact rule for defining membershipsfunctions

- Must be a reflection of human experience

❍Example membershipfunctions



❍Compare previous fuzzy membershipfunction with a crisp set (from binary logic)



❑ 3.1.4 Formalism

- Fuzzy set A is defined as

  $A = \{[u, \mu_A(u)], \forall u \in U\}\ with\ 0 \leq \mu_A(u) \leq 1$

  A is an ordered set, has 2 quantities:
  - ❍order of element
  - ❍corresponding truth value
- A discrete population A is written as

  $A = \mu_A(u_1)/u_1 + \mu_A(u_2)/u_2 + ... + \mu_A(u_n)/u_n.$

  / and + do not refer to division and sum
- A continuous population

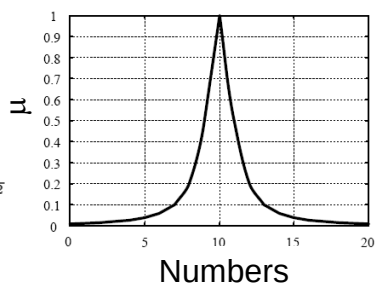
  $A = \int_A \mu_A(u)/u.$

  e.g.: concept '*almost equal to 10*'

  $A = \int_{\cong 10} \frac{1}{1+[u-10]^2}/u\ with\ \mu_A(u) = \frac{1}{1+[u-10]^2}$

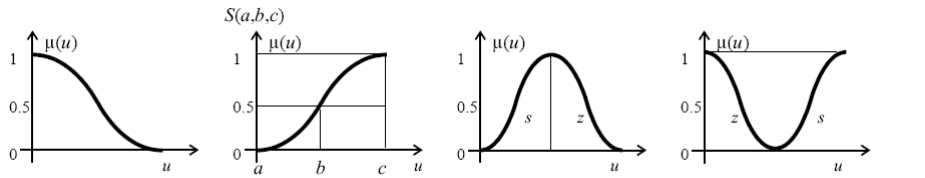
  in case

  $\mu_A(0) = 1/101, \mu_A(1) = 1/82, \mu_A(10) = 1, ...$



3.1 Fuzzy set

- 3.1.5 Memberschipfunction shapes
  - User-defined, common examples are



Z-shape S-shape Gaussian Inverse Gaussian

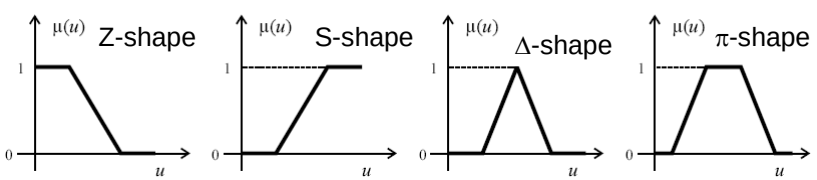
  - All 4 shapes can be derived from a general S-form

$$\begin{cases} S(u) = 0 & u \in [-\infty, a] \\ S(u) = 2(u-a)^2/(c-a)^2 & u \in [a, b] \\ S(u) = 1 - 2(u-c)^2/(c-a)^2 & u \in [b, c] \\ S(u) = 1 & u \in [c, \infty] \end{cases},$$

with $b = (a+c)/2$

3.1 Fuzzy set

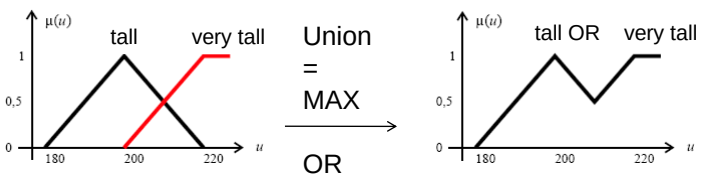
- Continuous membershipfunctions might result in complex formula's→ long computation time
- Solution: use simple fuzzy sets such as



- Piecewise linear nature of functions gives computational advantages and still gives 'good' results

3.2 Basic operations

- 3.2.1 Union $A \cup B$
  - = or-function
  - Create new concept using a combination of others
    e.g. ‘tall’ OR ‘very tall’
  - The truth value of this new concept is
    $\mu_{A \cup B}(u) = \max[\mu_A(u), \mu_B(u)]$.
  - EXAMPLES:
    - continuous



3.2 Basic operations

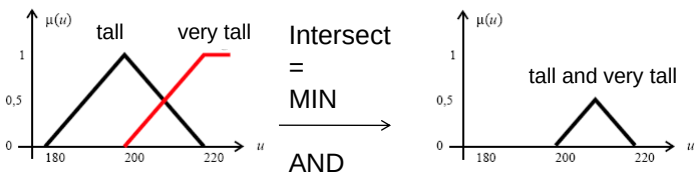
- EXAMPLES:
  - discrete
    ‘tall’ = 0/170 + 0/180+0.5/190+1/200+0.5/210+0/220
    ‘very tall’=0/170 + 0/180+0/190+0/200+0.5/210+1/220
    the union then gives
    ‘tall or very tall’=0/170 + 0/180+0.5/190+1/200+0.5/210+1/220
- Note that this max operator also applies on binary logic. Check truthtable of OR function…

❑ 3.2.2 Intersection $A \cap B$

- =and-function
- The truth value of this new concept is

$$\mu_{A\cap B}(u) = \min[\mu_A(u), \mu_B(u)].$$

- EXAMPLES:
  - ❍Continuous



  - ❍Discrete

'tall and very tall'=0/170 + 0/180+0/190+0/200+0.5/210+0/220

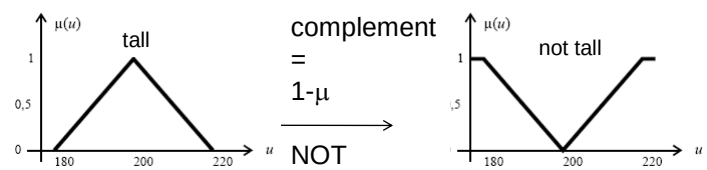
3.2 Basic operations

❑ 3.2.3 Complement $\neg A$

- The truth value of this new concept is

$$\mu_{\neg A}(u) = 1 - \mu_A(u).$$

- EXAMPLES:
  - ❍Continuous



  - ❍Discrete

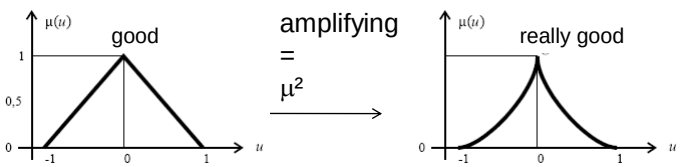
'not tall'=1/170 + 1/180+0.5/190+0/200+0.5/210+1/220

❑ 3.2.4 Amplifying

- An amplifying degree is defined as

$$REALLY\,A = \int_A \mu_A^2(u)/u$$

- EXAMPLES:
  - ❍ Continuous



  - ❍ Discrete

‘small’=1/1 + 0.7/2+0.2/3+0/4+0/5

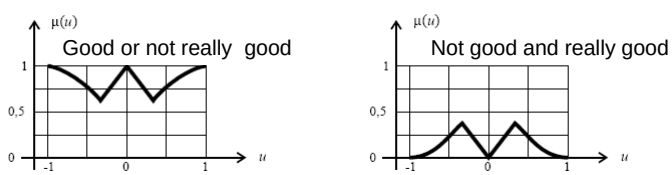
‘really small’=1/1 + 0.49/2+0.04/3+0/4+0/5

❑ 3.2.5 Application

Compute membershipfunctions for

1. ‘Small, but not really small’ (with a discrete population (integer elements [1-5] from previous slide)
2. ‘Good or not really good’ (with a continuous population from [-1,1] from previous slide)
3. ‘Not good, but really good’ (same population as in 2.)

‘small, but not really small’=0/1+0.51/2+0.2/3+0/4+0/5.



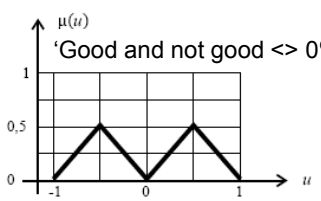
❑ 3.2.6 Remarks

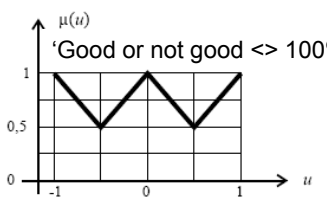
- Not all Boolean rules can be used

~~$A \cup^{\neg} A = U$~~
~~$A \cap^{\neg} A = \emptyset$~~

- Instead in fuzzy-set theory this becomes (suppose *A(u)=0.7)*

$$A(u) \cup^{\neg} A(u) = \max(0.7, 0.3) = 0.7 < 1$$
$$A(u) \cap^{\neg} A(u) = \min(0.7, 0.3) = 0.3 > 0$$





- Also other union and intersection operators possible (note that generally the union-operator → s-norm, intersection-operator → t-norm)
  - ❍EXAMPLES:
    - Probability operators

$$\mu_{A \cap B}(u) = \mu_A(u).\mu_B(u), \text{ algebraic multiplication}$$
$$\mu_{A \cup B}(u) = \mu_A(u) + \mu_B(u) - \mu_A(u).\mu_B(u), \text{ algebraic sum}$$

    - Giles operators

$$\mu_{A \cap B}(u) = \max\{0, \mu_A(u) + \mu_B(u) - 1\}, \text{ bounded difference}$$
$$\mu_{A \cup B}(u) = \min\{1, \mu_A(u) + \mu_B(u)\}, \text{ bounded sum}$$

  - ❍This course only uses min and max as explained before

3.3 Fuzzy relations

- Define relation between input- and output information
- Input and output mostly have different dimensions
- In this relation, there is knowledge about the desired control action
- Example: suppose fuzzy-set *A* operates in universum *U* and *B* in *V.* To relate them use

  'IF A THEN B'

  this rule assumes a relation R

  $$R = A * B$$

  when $\mu_A(u) = 0.5$ then action *B* will be only applied for 50%

3.3 Fuzzy relations

- Example (cont'd): relation *R* ('IF input large THEN output large') in matrix form (also called control matrix)

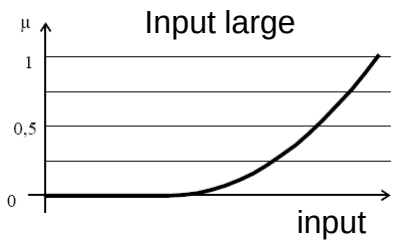
Linguistic vars for input

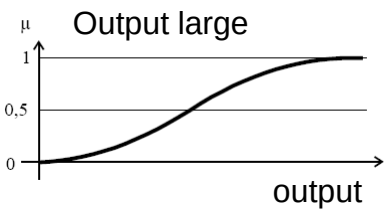
	Large	Medium	Small
Large	1	0	0
Medium	0	0	0
Small	0	0	0

Linguistic vars for output

3.3.1 More details on continuous populations

- Assume fuzzy-set for A ‘input large’, B ‘output large’ those as in the figures



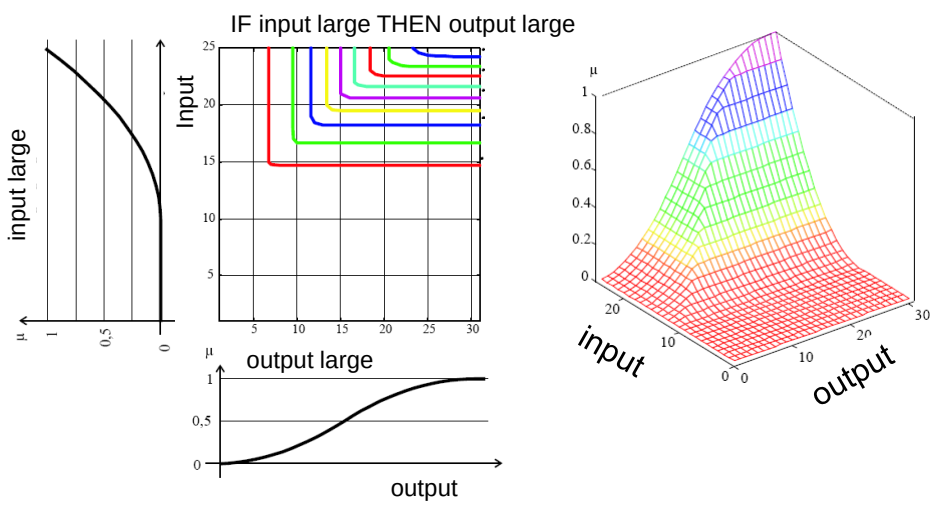


- Relation is defined as

$$R = A * B = \int_{uv} \mu_A(u) \cap \mu_B(v)/(u,v)$$

3.3.1 More details on continuous populations

- The matrix formulation as before extends to a 3-D figure where x-axis= input population, y-axis= output population, z-axis=truth value



3.3.2 More details on discrete populations

- In case of discrete sets a relation can be defined as

$$\mu_R(u,v) = \mu_{A*B}(u,v) = \min\left[\mu_A(u), \mu_B(v)\right], \forall u \in U, v \in V.$$

- In matrix form this gives

$$A * B = \begin{bmatrix} \min\left[\mu_A(u_1), \mu_B(v_1)\right] & \dots & \min\left[\mu_A(u_1), \mu_B(v_m)\right] \\ \vdots & \ddots & \vdots \\ \min\left[\mu_A(u_n), \mu_B(v_1)\right] & \dots & \min\left[\mu_A(u_n), \mu_B(v_m)\right] \end{bmatrix}$$

- Example: 'IF too fast THEN hit brakes'

  Define : 'too fast' with velocity in km/h

  'hit brakes' with NP, LP, HP respectively no, low and high pressing as

  A='Too fast'=0/20 + 0/40 + 0.1/60 + 0.6/90 + 1/120

  B='Hit brakes'=0/ NP + 0.5/LP + 1/HP

3.3.2 More details on discrete populations

- The fuzzy relation *R* is then

$$R = \begin{matrix} 0 \\ 0 \\ 0.1 \\ 0.6 \\ 1 \\ \uparrow \\ \text{A} \end{matrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0.1 & 0.1 \\ 0 & 0.5 & 0.6 \\ 0 & 0.5 & 1 \end{bmatrix} \begin{matrix} \\ \\ \\ \\ \\ 0 \quad 0.5 \quad 1 \leftarrow \text{B} \end{matrix}$$

3.3.3 Extension

- Rule ‘IF A THEN B, THEN C’ is defined as

  $A * B * C = \int_{u,v,w} \mu_A(u) \cap \mu_B(v) \cap \mu_C(w)/(u,v,w)$

  or

  $A * B * C = \sum_{u,v,w} \mu_A(u) \cap \mu_B(v) \cap \mu_C(w)/(u,v,w)$

- Notice that for rule ‘IF A and B THEN C’ the same result is obtained

3.4 Inference (or combination) rule

- Extend the ‘hit brake’ example with rules
  - ‘IF too fast THEN low brake pressing’
  - ‘IF ok THEN do nothing’
  - ‘IF slow THEN accelerate’

$$\begin{array}{c} \\ \text{MTF} \\ \text{TF} \\ \text{G} \\ \text{TS} \end{array} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{array}{c} \text{HB} \\ \text{LB} \\ \text{DN} \\ \text{AC} \end{array}$$

with MTF=much too fast; TF=too fast; G=good; TS=too slow; HB=hard brake; LB=low brake; DN=do nothing; AC= accelerate

3.4 Inference (or combination) rule

- If true speed is known then rules can be
  - 'true speed'=0/MTF+0.4/TF+0.6/G+0/TS
  - 'true action'=0/HB+0.4/LB+0.6/DN+0/AC

  'true speed' is a fuzzy-set over the linguistic input variables

  'true action' results from the inference rule and is a fuzzy-set over the output variables
- Generally: for an output set *B'*, an input set *A'* and a relation *R* ('IF A THEN B') we have

  $B' = A' \circ R = A' \circ A * B.$

  operator $\circ$ = maximum-minimum product
- Hence membershipfunction of *B'* equals

  $\mu_{B'}(v) = \max_v \min_u [\mu'_A(u), \mu_R(u, v)]$

3.4 Inference (or combination) rule

$$\min_u[\mu_{A'}, \mu_R(u, v)] = \begin{bmatrix} \min(0,1) & \min(0,0) & \min(0,0) & \min(0,0) \\ \min(0.4,0) & \min(0.4,1) & \min(0.4,0) & \min(0.4,0) \\ \min(0.6,0) & \min(0.6,0) & \min(0.6,1) & \min(0.6,0) \\ \min(0,0) & \min(0,0) & \min(0,0) & \min(0,1) \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0.4 & 0 & 0 \\ 0 & 0 & 0.6 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\mu_{B'} = \begin{bmatrix} \max_v \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} & \max_v \begin{bmatrix} 0 \\ 0.4 \\ 0 \\ 0 \end{bmatrix} & \max_v \begin{bmatrix} 0 \\ 0 \\ 0.6 \\ 0 \end{bmatrix} & \max_v \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} =$$

$$0/HB + 0.4/LB + 0.6/DN + 0/AC$$

3.4 Inference (or combination) rule

- EXAMPLE 2:
  - A=0/1+0.2/2+0.5/3+1/4
  - B=1/a+0.8/b+0.4/c+0/d

$$R = \begin{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 0.5 & 0.5 & 0.4 & 0 \\ 1 & 0.8 & 0.4 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \\ \begin{matrix} a & b & c & d \end{matrix} & \end{matrix}$$

Take as fuzzy set $A' = 0/1 + .3/2 + .6/3 + .7/4$. The relation $\min[\mu_{A'}(u), \mu_R(u,v)]$ then gives

$$\begin{bmatrix} \min(0,0) & \min(0,0) & \min(0,0) & \min(0,0) \\ \min(0.3,0.2) & \min(0.3,0.2) & \min(0.3,0.2) & \min(0.3,0) \\ \min(0.6,0.5) & \min(0.6,0.5) & \min(0.6,0.4) & \min(0.6,0) \\ \min(0.7,1) & \min(0.7,0.8) & \min(0.7,0.4) & \min(0.7,0) \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 0.5 & 0.5 & 0.4 & 0 \\ 0.7 & 0.7 & 0.4 & 0 \end{bmatrix}$$

$$\mu_{B'} = \begin{bmatrix} \max_v \begin{bmatrix} 0 \\ 0.2 \\ 0.5 \\ 0.7 \end{bmatrix} & \max_v \begin{bmatrix} 0 \\ 0.2 \\ 0.5 \\ 0.7 \end{bmatrix} & \max_v \begin{bmatrix} 0 \\ 0.2 \\ 0.4 \\ 0.4 \end{bmatrix} & \max_v \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \end{bmatrix} =$$

$0.7/a + 0.7/b + 0.4/c + 0/d.$

3.4 Inference (or combination) rule

$$R = \begin{matrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0.2 & 0.2 & 0.2 & 0 \\ 0.5 & 0.5 & 0.4 & 0 \\ 1 & 0.8 & 0.4 & 0 \end{bmatrix} & \begin{matrix} 1 \\ 2 \\ 3 \\ 4 \end{matrix} \\ \begin{matrix} a & b & c & d \end{matrix} & \end{matrix}$$

- Remark:
  - Control matrix implies that *d* will never be used
  - Also the case when
    - ‘IF hot THEN heating off‘

      This rule says nothing about the case when it is cold

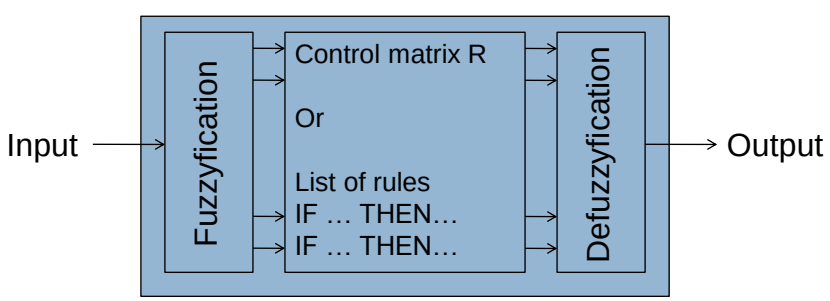
      but can be used in case only a little hot: e.g. in case only 30% hot, using the inference rule, results in heating 30% off → this is the minimum operator
    - In case there exists a second rule on output ‘heating off’ which is 50% true then the maximum is retained → 50% heating off
    - Combine the above operations results in a min-max-operator

3.4 Inference (or combination) rule

- Remark 2:
  - some programs do not use the control matrix *R*
  - but instead use 1 rule which combines all individual rules per linguistic output variable
  - E.g
    IF (A and B) or (C and D) or (A and not D) THEN action 1.
  - Again max-min operator

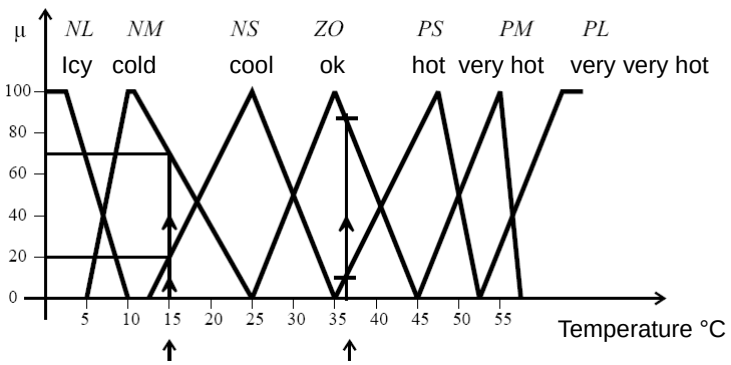
3.5 Fuzzy control algorithm

- A fuzzy controller must convert non-fuzzy input to fuzzy set → fuzzyfication step
- Idem for conversion of fuzzy to non-fuzzy output → defuzzyfication
- Architecture



3.5.1 Fuzzyfication step

- Compare measurements with predefined fuzzy sets



15°C=0.7/cold + 0.2/cool

36°C=0/NL+0/NM+0/NS+0.87/ZO+0.12PS+0/PM+0/PL. (using standard abbreviations)

3.5.1 Fuzzyfication step

- standard abbreviations

NL	Negative Large
NM	Negative Medium
NS	Negative Small
ZE	Zero
PS	Positive Small
PM	Positive Medium
PL	Positive Large

3.5.1 Fuzzyfication step

- Membershipfunctions are stored in the proces control system using equations
- Given an input measurement, the memberschipsdegree is determined using these equations

3.5.2 Rules

- IF … THEN …
- Are application-specific
- Example relation scheme (2 inputs error *e* and derivative of error *de*, 1 output *u*)

de,e	NL	NM	NS	ZE	PS	PM	PL
NL	ONL	ONL	ONL	ONL	ONM	ONS	OZE
NM	ONL	ONL	ONL	ONM	ONS	OZE	OPS
NS	ONL	ONL	ONM	ONS	OZE	OPS	OPM
ZE	ONL	ONM	ONS	OZE	OPS	OPM	OPL
PS	ONM	ONS	OZE	OPS	OPM	OPL	OPL
PM	ONS	OZE	OPS	OPM	OPL	OPL	OPL
PL	OZE	OPS	OPM	OPL	OPL	OPL	OPL

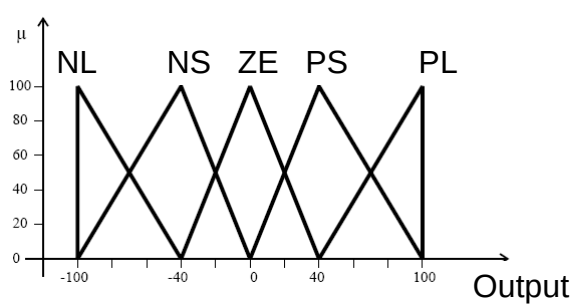
- Remark: not all cells need to be filled, the above scheme relates with a PD-controller, e.g. OPL= output positive large

3.5.2 Rules

- Example rules:
  - IF e=PL and de=ZE THEN u=OPL
  - IF e=PM and de=ZE THEN u=OPM
  - IF e=ZE and de=PL THEN u=OPL
  - IF e=ZE and de=ZE THEN u=OZE
  - IF e=NL and de=PL THEN u=OZE
- If some rules are more reliable than others → use weighting factors

3.5.3 Defuzzyfication step

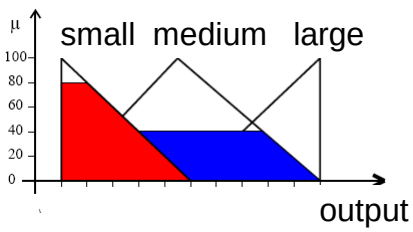
- Defuzzyfication = processing 1 output signal based on different truth values



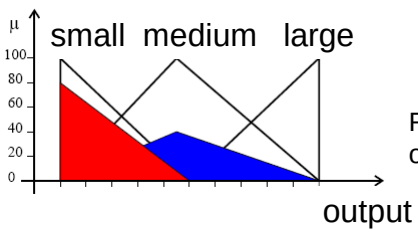
- Different (implication) methods to combine output variables, one mostly uses:
  - Minimum inference
  - Product inference

3.5.3 Defuzzyfication step

- Example: set=0.8/small + 0.4/medium + 0/large
  - Minimum inference (bounds on the minimum value)



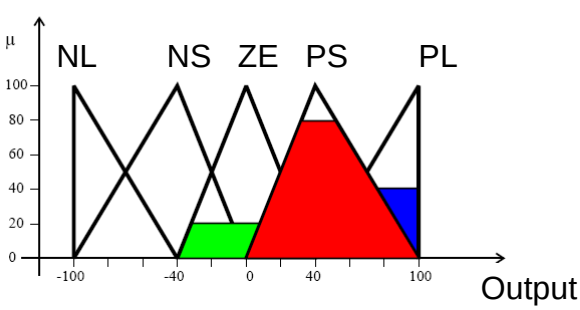
  - Product inference (bounds product of original μ() and truth value)



Remark: combination of each rule outcome is called aggregation (we use max)

3.5.3 Defuzzyfication step

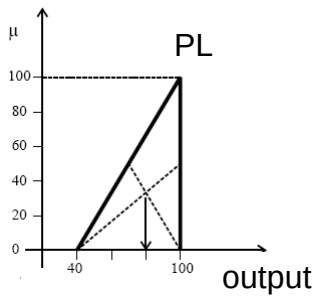
- Assume output set equals
  u=0/NL+0/NS+0.2/ZE+0.8/PS+0.4/PL
- Then according to minimum-inference rule the results looks like



- Positive small has highest truth value
- Conclusion might be: output = ±40

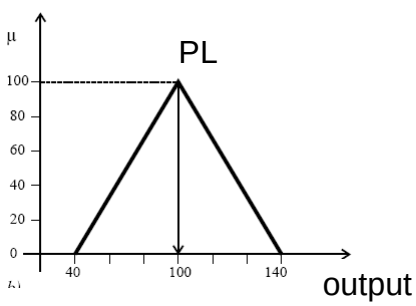
3.5.3 Defuzzyfication step

- Do nothing (ZE) and high control (PL) have trust values
- Total shaded area affects controller output
- → use center of gravity or median (line of gravity) method
- The location of median or center of gravity of the total shaded area (aggregated output) → controller output (mamdani-type controllers)
- Problems: figure will never output a value of 100



3.5.3 Defuzzyfication step

- Solution: extended center of gravity method
  - Mirror memberschipfunctions on the utmost left and right → in case PL = 100%, output=100



3.5.3 Defuzzyfication step

- Approximation to speed-up defuzzyfication
  - E.g. singleton method:
    - Define linguistic variables as singletons
    - Output becomes weighted mean of fuzzy output set

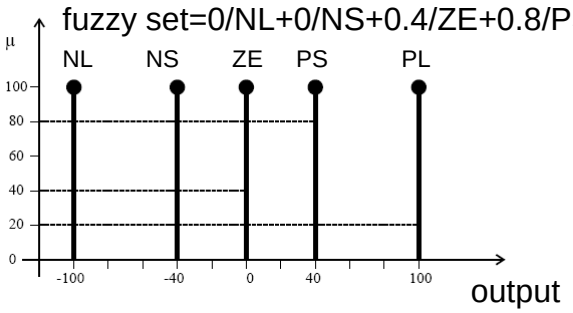
$$output = \frac{\sum_i v_i \mu(v_i)}{\sum_i \mu(v_i)}$$

    - Also called Sugeno-type models

3.5.3 Defuzzyfication step

- Example

  start from previous fuzzy output set and redraw as



$$\frac{40*0+80*40+20*100}{40+80+20} = 37.14\%$$

4 Extensions

- Extension to fuzzy controllers/control algorithms
- +/- compared to classical PID controllers
- Hybrid (fuzzy and classical) controllers

4.1 +/- compared to classical PID controllers

- +
  - Directly input experience;
  - Directed influencing of process;
  - Good match with human reasoning;
  - Flexible control behavior;
  - Good robustness against process deviations;
  - Good operation with non-linearity's
- -
  - Can’t compute properties since no mathematical model of closed control loop is present (simulation is necessary);
  - Experience of designer is necessary;
  - Implementation of classical I- and D- actions are still required;
  - Basics of control theory are still important for the design process

4.1 +/- compared to classical PID controllers

	Fuzzy	PID	State(-space) controller
Mathematical model	not necessary	desired	Required
Properties of system	Linear, non-linear	Mainly linear	More difficult for non-linear
# input/#output	SISO, MIMO	SISO	SISO,MIMO
Application domain	Systems where experience and intuition is important	linear	Systems with mathematical model
Control structure	Separate rules	One formula	One formula
Simplicity rules	Simple, natural language	Experience and knowledge necessary	Difficult, knowledge required
Adaptation of rules	Based on experiences	Based on system knowledge	Based on computations and simulations
Design methodology … based on	Qualitative Natural language	Quantitative Logical formulas	Quantitative Formulas and matrices
Reliability	Can be simulated	Can be computed	Can be computed
Robustness	Good	Moderate	Moderate up till good

4.2 Fuzzy-PID controller

- Derive standard rules analogous to those for classical PID controllers
- Recall PD-controller transfer function

$$TF_{PD} = Kr(1 + \tau_d p)$$

- PD control signal is proportional to
  - error signal and;
  - derivative of error signal

4.2 Fuzzy-PID controller

- Fuzzy controller simulates behavior using e.g.
  - IF e=PL AND de=ZE THEN u=PL
    IF e=PM AND de=ZE THEN u=PM

  …
  - IF e=ZE AND de=PL THEN u=PL

  …
  - IF e=NL AND de=PL THEN u=ZE

  …

de,e	NL	NM	NS	ZE	PS	PM	PL
NL				ONL			OZE
NM				ONM		OZE	
NS				ONS	OZE		
ZE	ONL	ONM	ONS	OZE	OPS	OPM	OPL
PS			OZE	OPS			
PM		OZE		OPM			
PL	OZE			OPL			

4.2 Fuzzy-PID controller

- Use memory to simulate I-action using rules
- IF e=NL THEN du=NL
  IF e=ZE THEN du=ZE
  IF e=PL THEN du=PL
- The total output signal (after defuzzification)
  Output = u+du1(k)
  du1(k)=du1(k-1)+du(k)

4.2 Fuzzy-PID controller

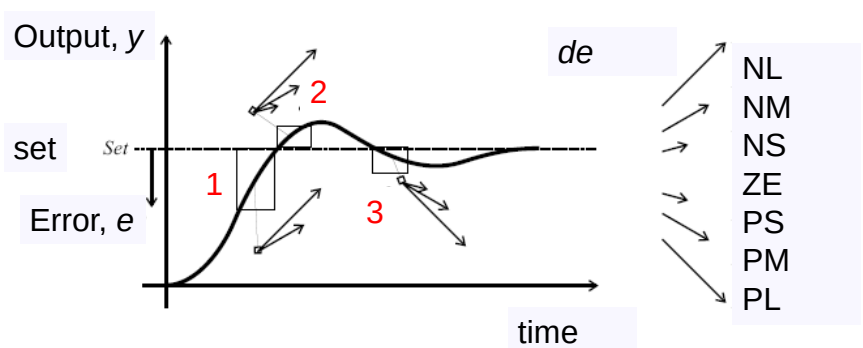
de,e	NL	NM	NS	ZE	PS	PM	PL
NL			DUNL		DUNL	DUNL	
NM			DUNM		DUNM	DUNM	
NS			DUNS				
ZE							
PS					DUPS		
PM					DUPM		
PL					DUPL		

1

2

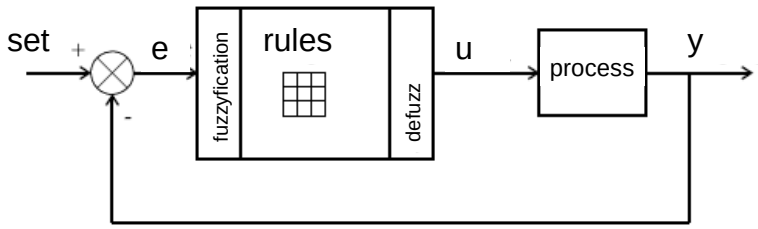
3

4.2 Fuzzy-PID controller

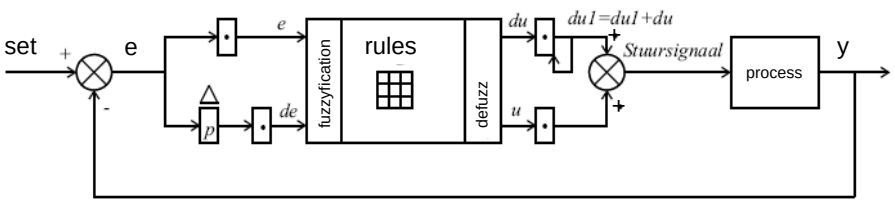


4.3 Control structures

- Simplest form

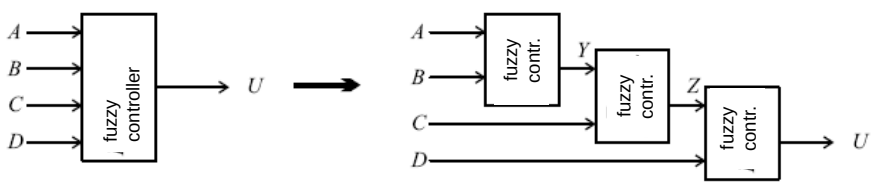


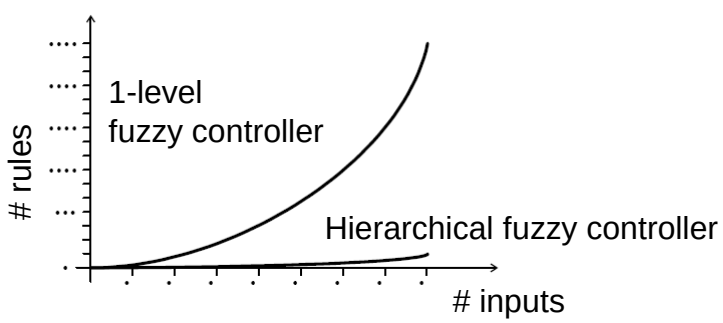
- Fuzzy PID implementation



4.3 Control structures

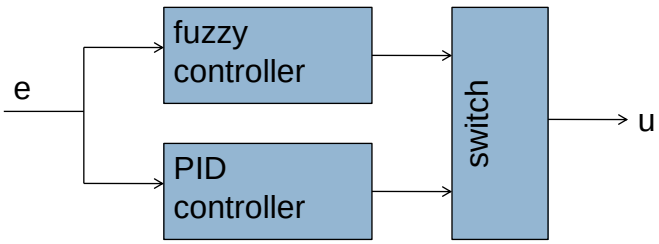
- Hierarchical structure



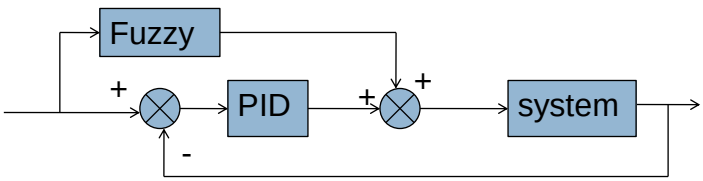


4.3 Control structures

- Simplest hybrid structure

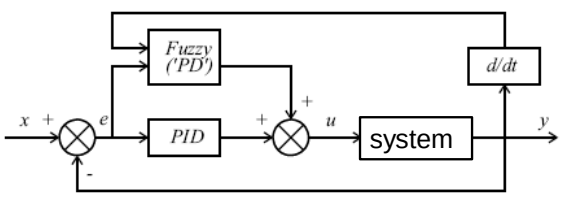


- Fuzzy controller as forward coupling for fast and stable start-up

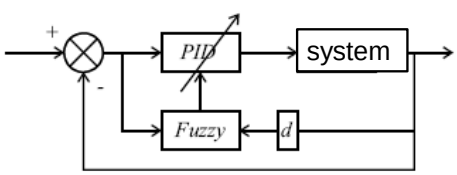


4.3 Control structures

- Hybrid structure

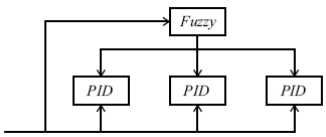


- Adaptive hybrid structure



4.3 Control structures

- Supervision structure



4.4 Industrial applications

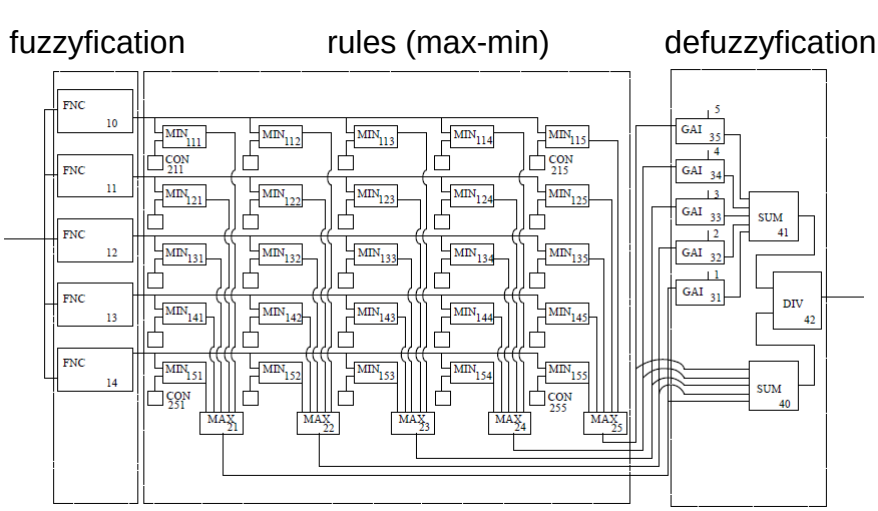
- Examples include:
  - structure switching in control loops
  - parameter adaptation in control loops
  - feed-forward coupling with fuzzy controllers
- Not limited to industrial apps also applies to
  - residential devices such as washing machines, micro-wave ovens, vacuum cleaners, … ;
  - image recognition;
  - or in financial engineering

4.4 Industrial applications

- Practical realizations
  - fuzzy control for metro of Sendai → increased comfort when accelerating and hitting brake, less energy consumption, better planning
  - fuzzy control in water treatment installation → better quality
  - fuzzy control in air-conditioning systems → increased comfort, less energy consumption
  - fuzzy control in elevators → less waiting times
  - …
- Hardware/software application depends on problem

4.4 Industrial applications

- Example implementation



References

- J. Baeten, “Deel 3: fuzzy regeltechniek”, http://193.190.56.244/~jbaeten/
- L. Zadah, “Fuzzy sets as a basis of possibility”, Fuzzy Sets Systems, Vol. 1, pp3-28, 1978.
- T. J. Ross, “Fuzzy Logic with Engineering Applications”, McGraw-Hill, 1995.
- K. M. Passino, S. Yurkovich, "Fuzzy Control" Addison Wesley, 1998.
- W. Pedrycz, “Fuzzy control and fuzzy systems,” sec. ed., New York: John Wiley, 1993
- Siemens AG, “Fuzzy Logic,” december 1993

Fuzzy Logic: design examples

P. Karsmakers

KATHOLIEKE UNIVERSITEIT LEUVEN

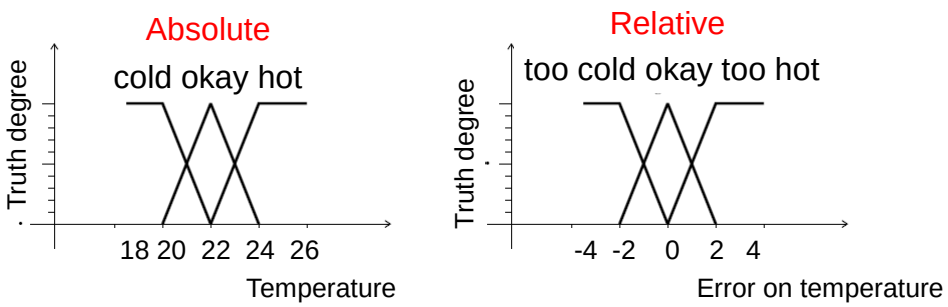
KHK
KATHOLIEKE HOGESCHOOL KEMPEN

Outline

- Room temperature control
- Water level control
- Cart and Pole Simulation

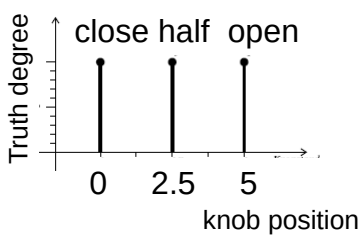
1. Room temperature control

- Simple didactical example
  - Output: room temperature
  - Input: knob position of heating
- Design flow:
  - First step: define linguistic variables
    - Although it's better to work with relative concepts, we first start with absolute definitions



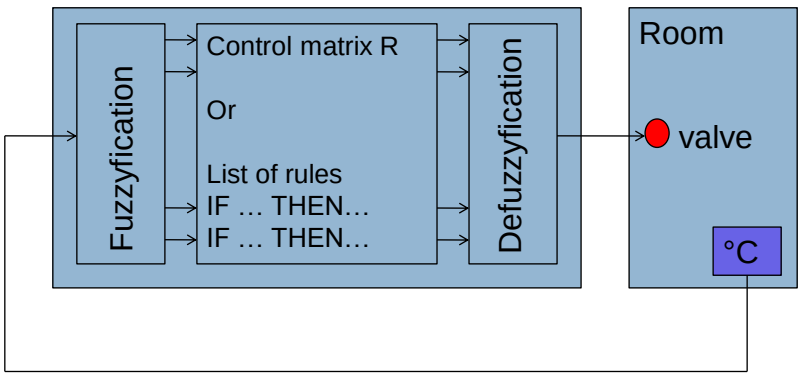
1. Room temperature control

- Sugeno –type controller is used → output concepts are singletons
- No thermostatic but simple valve
- Notice that all linguistic variables are chosen arbitrarily, other options could be valid too.



1. Room temperature control

- Second step: control loop and rules



1. Room temperature control

- Possible rules:
  - 1: IF cold THEN open
  - 2: IF okay THEN half
  - 3: IF hot THEN close
- Sounds logical that controller needs to strive to okay
- Rule 2 says if okay then knob position = 2.5, this assumes an academic correct dimensioned system that finally evolves to 22° when knob position = 2.5

- Remark: practical setup can be improved by
  - using relative definitions
  - Integrating rules such as “IF too cold THEN increase heating” would be better but are a little bit more difficult

1. Room temperature control

❑ Control matrix

$$R = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

open half close

cold okay hot

1. Room temperature control

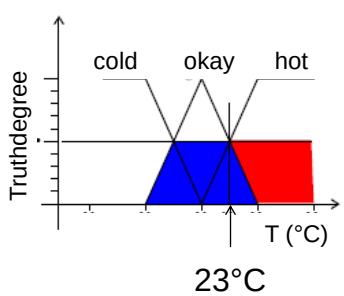
❑ Controller operation

- 1: Fuzzyfication step

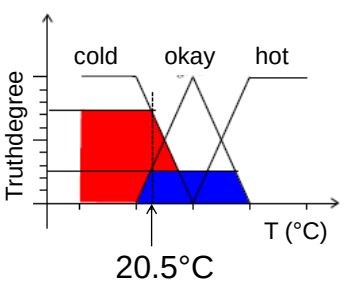
  Suppose roomtemperature = 23°C , 20.5°C

  Fuzzyfy: 23=0/cold+0.5/okay+0.5/hot

  20.5=0.75/cold+0.25/okay+0/hot



23°C

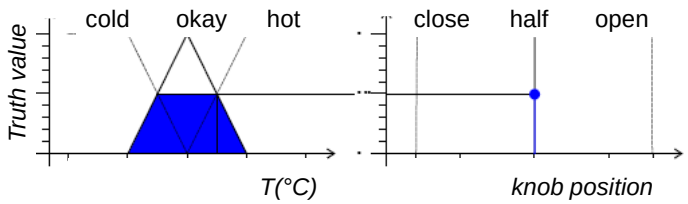


20.5°C

1. Room temperature control

- 2: Compute outputset
  - ❍Apply rules 2 and 3 to inputset
  - ❍Both rules have truth degree 0.5 for 23°C hence output values are only half true

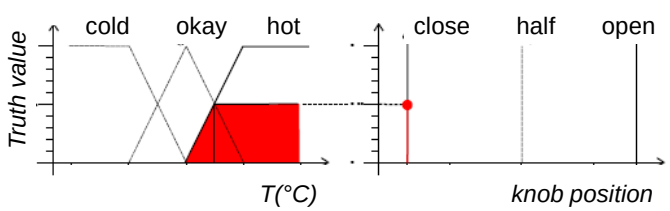
IF okay THEN half



1. Room temperature control

- 2: Compute outputset
  - ❍Apply rules 2 and 3 to inputset
  - ❍Both rules have truth degree 0.5 for 23°C hence output values are only half true

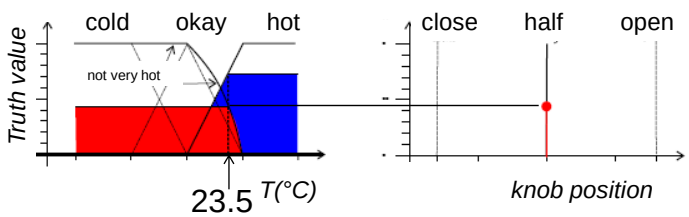
IF hot THEN close



1. Room temperature control

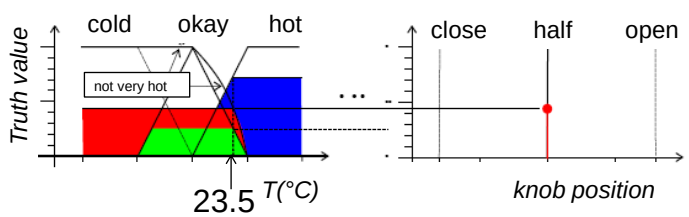
- 3: Combination of rules using OR/AND-function

example: *6: IF hot AND NOT very hot THEN half*



- Combine rule 3 with 6 (latter) using OR-function (e.g. max) which gives

IF hot AND NOT very hot OR okay THEN half



1. Room temperature control

- 4: defuzzyfication step

$$output = \frac{\sum_i v_i \mu(v_i)}{\sum_i \mu(v_i)}$$

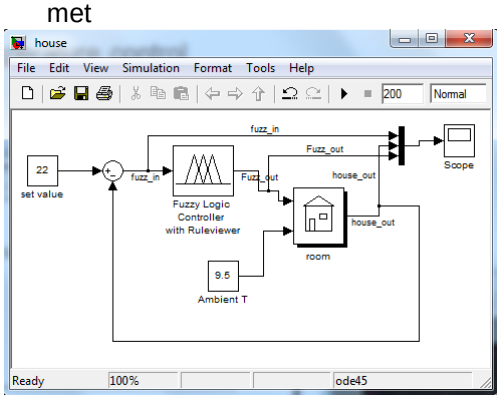
for 23°C=0.5/close+0.5/half+0/open this gives

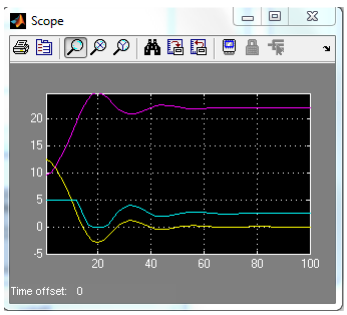
output=(0.5*0+0.5*2.5+0*5) / (0.5+0.5+0)=1.25

- Questions:
  - ❍What if it is desired to control room for other T?
  - ❍What if due to some kind of “load” on the room temperature a steady state error (D:standfout) exists?

1. Room temperature control

- Simulation
  - Room is modelled as second order system (ω_n=0.1 r/min, ζ=0.7)
  - Room heats 5°C / unit of knob position
  - When ambient temperature = 9.5°C then set temperature is exactly met

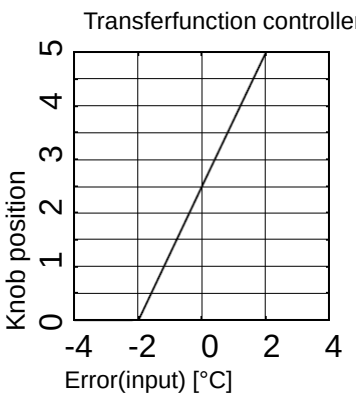


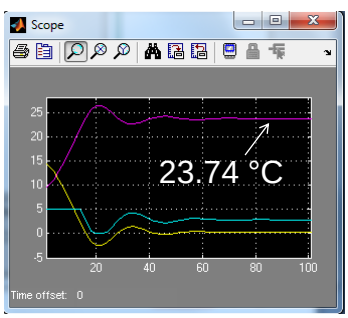


pink line = T room
blue line = controller output

1. Room temperature control

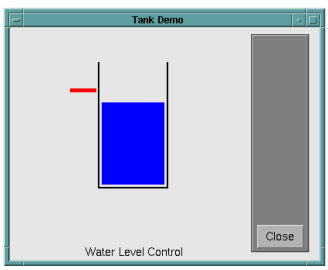
- Changing set value and/or ambient temperature will give a steady state error → proportional controller (with saturation → knobposition [0-5])
- Example: set value = 24°C
- Can the steady-state-error be obtained using the controllers' transferfunction?





2. Water level control

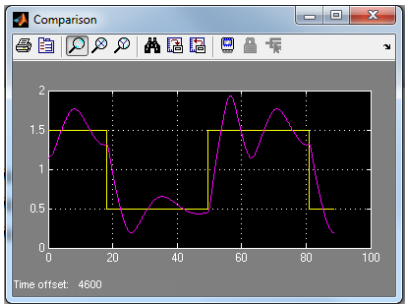
- After creation of fuzzy system (E.g. using the GUI tools) you are ready to embed your system directly into a simulation
- Problem description
  - tank with a pipe flowing in and a pipe flowing out
  - input flow adjustable change the valve controller
  - outflow rate constant (depends on diameter of the outflow pipe)
  - varying pressure in tank (depends on water level)
  - the system has some very nonlinear characteristics.



2. Water level control

- Design controller:
  - Controller input: desired water level - actual water level
  - Controller output: rate at which the valve is opening or closing.
- A first simple fuzzy controller design might be:
  1. *If (level is okay) then (valve is no_change)*
  2. *If (level is low) then (valve is open_fast)*
  3. *If (level is high) then (valve is close_fast)*
- Simulate in simulink using block called the Fuzzy Logic Controller block.
- Demo: `sltank` (shows simulink model and loads FIS (Fuzzy Inference System) structure)

2. Water level control



1. If (level is okay) then (valve is no_change)

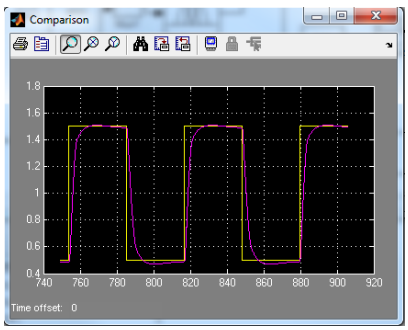
2. If (level is low) then (valve is open_fast)

3. If (level is high) then (valve is close_fast)

- Solution: slow down the valve movement close to the right level by adding

4. If (level is good) and (rate is negative), then (valve is close_slow)

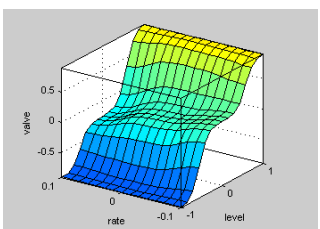
5. If (level is good) and (rate is positive), then (valve is open_slow)



2. Water level control

Interesting property:

- ❑ tank empties more slowly than it fills up (because of the specific value of the outflow diameter pipe)
- ❑ Change this by making close_slow valve membership function slightly different from open_slow setting.
- ❑ A PID controller does not have this capability. The valve command versus the water level change rate (depicted as water) and the relative water level change (depicted as level) surface looks like this.

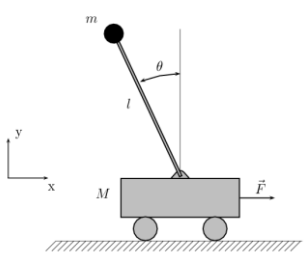


2. Water level control

- General remarks:
  - Make your own controller using the fuzzy logic block set in simulink. (`fuzblock`)
  - Connect fuzzy controller with a FIS configuration which can be prepared using the FIS editor (`fuzzy`)
  - If changes are made using the FIS editor → do not forget to export changes to workspace

3. Cart and Pole simulation

- Also called inverted pendulum
- Problem:
  - a pendulum which has its mass above its pivot point.
  - a pole is mounted with the pivot point on a cart that can move horizontally.
  - inverted pendulum is inherently unstable,
  - must be actively balanced in order to remain upright,
    - In this case by moving the pivot point horizontally as part of a feedback system



3. Cart and Pole simulation

- Demo: `slcp`
- look at simulink implementation of the controller

References

- J. Baeten, “Deel 3: fuzzy regeltechniek”, http://193.190.56.244/~jbaeten/
- The MathWorks, “Fuzzy logic Toolbox”, http://www.mathworks.com/help/toolbox/fuzzy/exampleindex.html