



# Fuzzy control

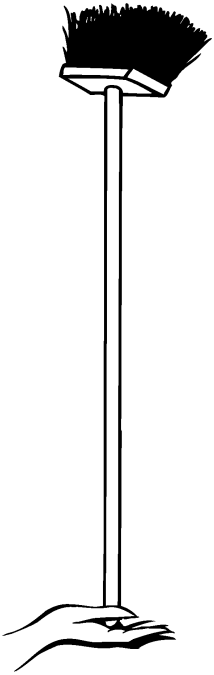
## **1 Inleiding**

### **1.1 Bij dit hoofdstuk**

In de voorgaande hoofdstukken gingen we er steeds van uit dat we een mathematisch model van het te regelen proces konden opstellen. In de meeste gevallen lukt dat ook wel. Sommige processen moeten echter met wat kunst en vliegwerk (bijv. lineariseren) worden beschreven in een bepaald werkgebied. Maar er zijn ook processen die zich moeilijk in een – normaliter gebruikt – mathematisch model laten vastleggen. Het ontwerp van een automatische regelaar is dan ook onmogelijk. Toch blijken wij als individu vaak prima in staat zo'n proces te besturen en zelf als regelaar te fungeren. Kennelijk kunnen wij via waarneming met onze zintuigen ('meten') en verwerking hiervan met onze hersenen ('regelen') dit soort processen goed de baas ('sturen').

Als voorbeeld zou genoemd kunnen worden het terugslaan van een tennisbal die op ons afkomt. Wij zijn in staat als individu dat heel behoorlijk te doen. Dit automatiseren (met gebruik van een robot) is vrijwel onmogelijk omdat de modelvorming vrijwel onmogelijk is. Hetzelfde geldt voor het verticaal gebalanceerd in de lucht houden van een bezemsteel op de palm van een hand in figuur 1.

De mathematische beschrijving van het dynamisch gedrag van zo'n bezemsteel in de driedimensionale ruimte is behoorlijk ingewikkeld (rotatie én translatie in drie vrijheidsgraden). Inzicht in het procesgedrag, zoals in dit boek steeds als voorwaarde is gesteld voor het kunnen ontwerpen van een regelaar, hebben we niet. Toch zijn we vaak in staat om ook zonder deze kennis het proces te regelen! Het lijkt alleszins de moeite waard deze manier van 'menselijk regelen' te bestuderen en vast te leggen in een algoritme. Dat kan vervolgens softwarematig worden geïmplementeerd in een computer, zodat – in tegenstelling tot de juiste gedane uitspraak – toch een 'automatisch regeling' totstandkomt. We noemen dat een 'fuzzy' regeling. Fuzzy duidt hier op vaagheid, de procesgegevens zijn minder strikt en de regeling wordt meer 'gevoelsmatig'. De met behulp van fuzzy control ontwikkelde regeling is echter niet vaag in de betekenis van onvoorspelbaar. De opgelegde regels voor de regeling



**Figuur 1** Balanceren van een bezemsteel

veroorzaken gedrag van het proces dat voor 100% voorspelbaar is.

## 1.2 Introductie

We hebben in dit boek steeds gewerkt met een speciaal type proces: lineair, analoog, continu, tijdonafhankelijk en SISO. Alleen processen die tot deze categorie behoren, kunnen worden beschreven met een overbrengingsfunctie  $H(s)$ , en dan nog slechts voor het geval we kijken naar de *verandering* die ontstaat ten opzichte van het werkpunt waarin het proces geacht wordt zich op het moment van aansturing in statisch evenwicht te bevinden. Hetzelfde geldt voor de regelaar. Voor het verwerven van elementaire inzichten zijn deze (ideale) processen heel waardevol.

Wat nu in een fuzzy geregeld systeem gebeurt, is in feite het nabootsen van het *menselijk regelen*. Omdat mensen geen mathematische wezens zijn, ontmoeten we bij deze aanpak geen mathematische analyses en zelfs niet eens écht numerieke waarden; de inhoud van een glas bier bijvoorbeeld duiden wij mensen aan met een zogenaamde linguïstische waarde als 'vol', 'leeg', 'half vol' of 'vrijwel leeg', en niet met een numerieke waarde zoals 23, 0, 9 of 2 cl.

Bij een fuzzy aanpak moet bij wijze van spreken de knop volledig om; in dat deel van de regelkring waar de fuzzy control zich afspeelt wordt géén wiskunde bedreven. Dat gebeurt

ook niet in onze hersenen bij het balanceren van de bezemsteel; de hersenen werken meer in de trant van:

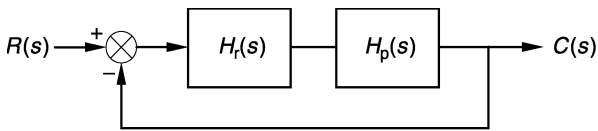
- als de bezemsteel wat naar rechts helt, dan moet de hand langzaam een beetje naar rechts bewogen worden;
- als de steel veel naar rechts helt, dan moet de hand snel en redelijk ver naar rechts bewogen worden;
- als de steel nog meer naar rechts helt of dreigt te vallen, dan moet de hand snel en redelijk ver naar rechts bewogen worden en tegelijkertijd naar beneden;
- als de steel vrijwel verticaal en stil staat, dan moet de hand niet bewogen worden;
- als de steel naar voren en wat naar links helt, dan ...
- enzovoort.

In dit proces zijn vele ‘en/of’-operaties op de ingangssignalen van de regelaar van toepassing, terwijl op de uitgangssignalen van de regelaar in het algemeen alleen ‘en’-operaties van toepassing zijn.

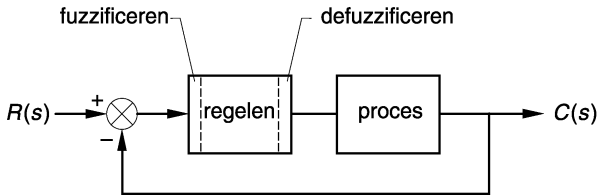
Deze handelingen kunnen geprogrammeerd worden met behulp van ‘if-then’-statements. Als de set van ‘if-then’-statements compleet is, zodat alle mogelijke situaties die zich kunnen voordoen worden afgevangen, en als de set voldoende vaak doorlopen wordt zodat de stuurwaarde voor het proces voldoende vaak wordt bijgesteld, dan hebben we ook voor dit ‘niet te beschrijven proces’ een regelaar ontworpen. Het spreekt voor zich dat zo’n fuzzy regelaar ook ontworpen kan worden voor een proces dat wel te beschrijven is, maar gebruikelijk is dat niet; men geeft er dan meestal de voorkeur aan een ‘mathematische’ regelaar te ontwerpen omdat het gedrag ervan – inclusief bewijsvoering – goed te beschrijven is.

Figuur 2 geeft het blokschema van een fuzzy regelaar weer. Ten opzichte van de mathematische regelaar is er geen fundamenteel verschil. Zoals uit de figuur blijkt, wordt aan de ingang van de fuzzy regelaar altijd nog een numerieke waarde aangeboden, want het naar rechts hellen van de bezemsteel bijvoorbeeld wordt gemeten door een opnemer die zijn meetwaarde in de vorm van een getal afgeeft. Er zal dus een interface aanwezig moeten zijn die deze numerieke meetwaarde omzet in een voor de fuzzy regelaar zinvolle waarde als ‘een beetje’, ‘wat meer’, ‘sterk’ enzovoort. Dit proces heet *fuzzificeren* van de numerieke waarde. In de regelaar zelf wordt alles op basis van deze fuzzy (linguïstische) waarden afgehandeld en wordt een – eveneens fuzzy – stuurwaarde voor het proces bepaald. Aan de uitgang van de fuzzy regelaar zal het omgekeerde moeten plaatsvinden: de fuzzy stuurwaarde moet worden omgezet in een numerieke waarde, want de stuurwaarde kan de bezemsteel natuurlijk niet ‘langzaam’ en ‘een beetje’ naar rechts brengen. Dat moet iets zijn als ‘met een snelheid van 0,6 m/s’ en ‘over een afstand van 23 cm’. Dit proces heet het *defuzzificeren* van de fuzzy stuurwaarde.

In werkelijkheid zal al gauw sprake zijn van méér dan één ingangssignaal (stuursignaal) voor het proces en méér dan één uitgangssignaal. Bovendien zijn er meestal meerdere stoorsignalen in het spel. Bij een mengkraan voor warm en koud water bijvoorbeeld geldt voor het (meng)proces:



a.



b.

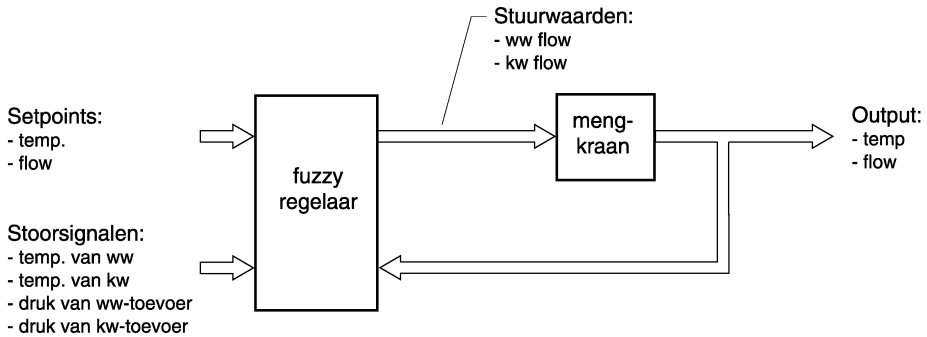
**Figuur 2** Mathematische (a) en fuzzy (b) regeling

- *ingangssignalen* zijn de warmwaterflow en de koudwaterflow;
- *uitgangssignalen* zijn de temperatuur van het mengwater en de flow;
- *stoorsignalen* zijn onder andere variaties in temperatuur en druk van de warm- en koudwatertoevoer.

Als de afhandeling van al deze signalen met een fuzzy regeling geschiedt, dan ontstaat het schema van figuur 3 en hebben we een automatische regeling gemaakt. Deze werkt goed als wij in staat zijn in de fuzzy regelaar een complete set van zinvolle ‘if-then’-statements op te nemen die per tijdseenheid voldoende vaak wordt doorlopen om de aansturing van het proces bij te stellen. De kunst is natuurlijk die set inderdaad *compleet* en *zinvol* te krijgen. We zijn weliswaar verlost van mathematische perikelen, maar daarvoor in de plaats zijn andere problemen gekomen: wat is zinvol en wanneer is de set compleet (dus vergeten we niets)? Zo is het statement ‘als de toevoerdruk van het warme water een beetje stijgt, dan moet de warmwaterflow wat groter en de koudwaterflow wat kleiner worden’ niet zinvol (aannemende dat de setpoints onveranderd zijn gebleven, dus flow en temperatuur aan de uitgang van het proces hetzelfde moeten blijven). Dit statement mag dus niet voorkomen. Het statement ‘als het temperatuur-setpoint fors verhoogd wordt, dan moet de koudwaterflow fors verlaagd en de warmwaterflow fors verhoogd worden’ mag natuurlijk niet ontbreken (aannemende dat het flow-setpoint hetzelfde is gebleven).

## 2 Terminologie

Fuzzy systemen kennen natuurlijk hun eigen terminologie. We zullen nu eerst een aantal gangbare termen uit dit vakgebied toelichten alvorens aan de hand van een voorbeeld de werking van een fuzzy-geregeld proces te demonstreren. We kiezen in het algemeen de En-

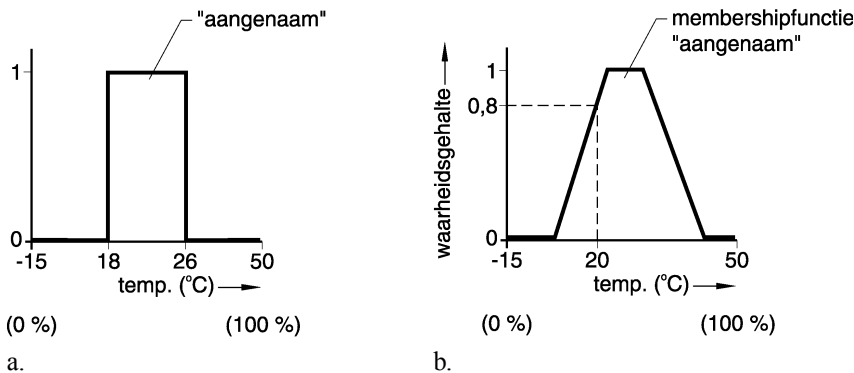


**Figuur 3** Fuzzy regeling van een MIMO-proces

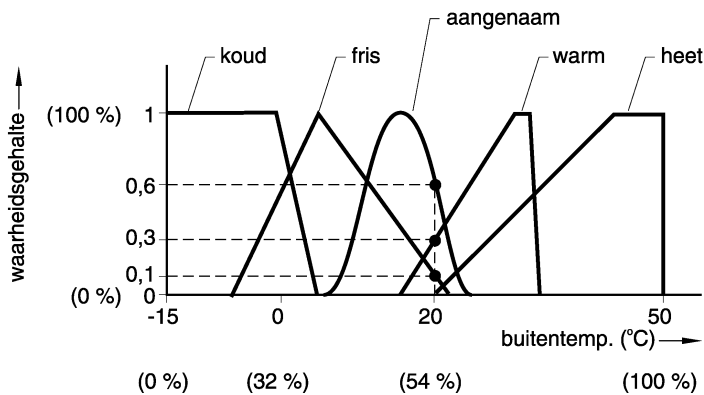
gelse benamingen; vaak zijn er varianten in omloop waarmee dus hetzelfde wordt bedoeld.

- *Fuzzy* betekent letterlijk zoiets als vaag, onduidelijk, en moet bijvoorbeeld in het vakgebied van de logica worden gezien als tegenhanger van het woord ‘crisp’ (hard, tweewaardig, binair); we spreken in dat geval van fuzzy logic versus crisp logic. Bij crisp logic is een uitspraak waar óf niet waar, een signaal hoog óf laag, een deur open óf dicht; dit wordt geassocieerd met de binaire waarde 0 of 1. Bij fuzzy logic staan we toe dat ook tussenliggende waarden mogelijk zijn. Dat spooft ook met de werkelijkheid, want een uitspraak kan immers ook een beetje waar zijn (in plaats van waar of niet-waar), een signaal kan half hoog zijn en een deur kan vrijwel dicht zijn. Dat betekent dat we aan een toestand niet meer de waarde 0 of 1 moeten toekennen, maar een zogenaamd ‘waarheidsgehalte’ tussen 0 en 1, waarmee de overgang van de ene naar de andere toestand geleidelijk wordt gemaakt. In figuur 4b is dat als voorbeeld weergegeven voor de uitspraak ‘het is aangenaam weer’ op basis van de buitentemperatuur. In de figuur lezen we af dat de uitspraak ‘het is aangenaam weer’ bij een buitentemperatuur van 20 °C een waarheidsgehalte van 0,8 (= 80%) heeft. In figuur 4a zien we dat deze uitspraak de (crisp) waarde 1 heeft.
- *Linguïstische* waarden vormen, zoals we gezien hebben, de tegenhanger van numerieke waarden. In ons dagelijks leven werken we meestal met linguïstische waarden, terwijl we in de fysica voornamelijk numerieke waarden hanteren. Een fuzzy waarde kan geassocieerd worden met een linguïstische waarde.
- *Membershipfuncties* geven aan hoe de overgang van de ene toestand naar de andere toestand verloopt (zie fig. 4b). De vorm en plaats van deze overgangen wordt door de ontwerper van het fuzzy systeem bepaald en het is de kunst van de ontwerper hierin de juiste keuzen te maken. Bij een bepaalde grootheid/uitspraak kunnen meerdere (zinvolle) membershipfuncties worden bedacht. Als voorbeeld zijn in figuur 5 bij de variabele buitentemperatuur de membershipfuncties ‘koud’, ‘fris’, ‘aangenaam’, ‘warm’ en ‘heet’ weergegeven.

De vorm en plaats van de membershipfuncties zijn naar eigen inzicht gekozen. We



**Figuur 4** Crisp (a) versus fuzzy (b)



**Figuur 5** Voorbeelden van membershipfuncties

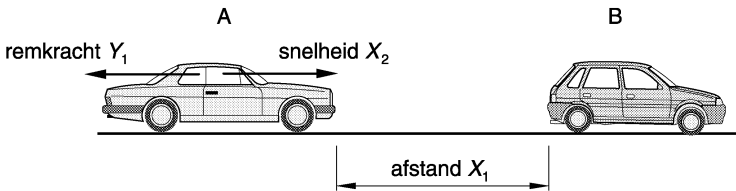
zien dat membershipfuncties elkaar kunnen overlappen. Dat is heel normaal, want een buitentemperatuur van  $20$  °C wordt niet door iedereen hetzelfde beoordeeld; wat voor de een aangenaam is kan voor de ander fris zijn of zelfs al warm.

- *Fuzzificeren* is zoals eerder gezegd het omzetten van numerieke ingangswaarden van de regelaar in voor het fuzzy systeem te hanteren fuzzy waarden. In feite zijn deze fuzzy waarden dus op te vatten als linguïstische waarden, die intern in het fuzzy systeem toch worden voorgesteld als een numerieke waarde (dat kan in een computer natuurlijk niet anders), en wel als de waarde van het waarheidsgehalte van een of meer membershipfuncties behorend bij dat ingangssignaal. De membershipfuncties vormen dus de interface tussen numeriek en fuzzy. In figuur 5 betekent dit dat een numerieke waarde van  $20$  °C wordt vertaald in een fuzzy waarde met een waarheidsgehalte van 10% voor de membershipfunctie 'fris', 60% voor de membershipfunctie 'aangenaam' en 30% voor de membershipfunctie 'warm'; 'koud' en 'heet' scoren beide 0%.

- De *rule-base* is de verzameling ‘if-then’-statements waarmee van de gefuzzificeerde ingangssignalen van de regelaar fuzzy uitgangssignalen van de regelaar worden gemaakt. Het zal duidelijk zijn dat elke statement uit de rule-base een fuzzy waarde van een of meer uitgangssignalen kan opleveren; dit manifesteert zich in de vorm van een bepaald waarheidsgehalte van een of meer membershipfuncties van elk uitgangssignaal. Hoe dat precies werkt zien we het beste aan het hierna te behandelen voorbeeld.
- *Fuzzy inferentie* is in elk fuzzy geregeld systeem de op één na laatste stap die in de fuzzy regelaar wordt uitgevoerd. Dat wil zeggen dat van alle membershipfuncties van alle uitgangssignalen van de regelaar (dit zijn dus de stuursignalen voor het proces) wordt bepaald hoe groot hun uiteindelijk aandeel is in het betreffende uitgangssignaal. Hoe dit werkt wordt eveneens het beste duidelijk aan de hand van het voorbeeld.
- *Defuzzificeren* is de laatste stap die wordt uitgevoerd in de fuzzy regelaar. Zoals eerder gezegd, worden nu alle fuzzy stuurwaarden van de regelaar omgezet in een numerieke waarde waarmee het proces daadwerkelijk kan worden aangestuurd. Deze operatie vormt dus de interface tussen de uitgang van de fuzzy regelaar en de ingang van het proces. Ook deze operatie wordt het best duidelijk in het voorbeeld.

### 3 Verkennend voorbeeld

Als voorbeeld zullen we een regeling beschouwen waarmee kop-staart-botsingen op autosnelwegen voorkomen kunnen worden. Indien auto A in figuur 6 hiermee is uitgerust, dan kunnen de remmen automatisch worden aangestuurd zodat A niet op B botst. We zullen stap voor stap nagaan hoe daarvoor een fuzzy regelaar kan worden ontworpen.



**Figuur 6** Regeling om kop-staart-botsingen te voorkomen

#### Stap 1

Kies zinvolle grootheden als input voor de regelaar. De grootheid ‘afstand’ tot B en ‘snelheid’ van A zijn zinvol en zelfs noodzakelijk. We zullen deze respectievelijk  $X_1$  en  $X_2$  noemen. Andere zinvolle grootheden zijn onder andere verkeersdruk, toestand van het wegdek, fitheid van de bestuurder en weersomstandigheden. Ten behoeve van de eenvoud laten we die echter buiten beschouwing. Het spreekt voor zich dat als dergelijke grootheden worden meegenomen, deze ook door middel van een sensor moeten worden gemeten. Niet-zinvolle grootheden zijn bijvoorbeeld de temperatuur van de wiellagers of de dag van

de week, ofschoon hierover niet het laatste woord gezegd hoeft te zijn! Het is een hele kunst alle zinvolle/noodzakelijke grootheden te vinden die als input voor de regelaar moeten worden genomen. Naarmate die beter gekozen worden, zal de regeling beter werken (dit geldt in feite ook voor een niet-fuzzy regeling). Uit de praktijk van de fuzzy regelingen blijkt dat – vanaf een zeker aantal – het toevoegen of weglaten van een minder belangrijke ingangsgrootheid vaak niet eens meer zoveel effect heeft (waarschijnlijk omdat door allerlei correlaties die ingangsgrootheid dan toch al vertegenwoordigd is in andere ingangsgrootheden).

### *Stap 2*

Kies ook zinvolle en vooral noodzakelijke uitgangsgrootheden voor de output van de regelaar. Een noodzakelijke grootheid zal de remkracht zijn (voetrem); we geven die in ons voorbeeld aan met de letter  $Y_1$ . Een zinvolle grootheid is de brandstoftoevoer naar de motor (gaspedaal) of de remkracht via de handrem, respectievelijk aan te duiden met  $Y_2$  en  $Y_3$ . Voor de eenvoud en het gemak beperken we ons in dit voorbeeld tot het uitgangssignaal  $Y_1$ . Duidelijk zal zijn dat ook het kiezen van de juiste uitgangssignalen van de regelaar een hele kunst is.

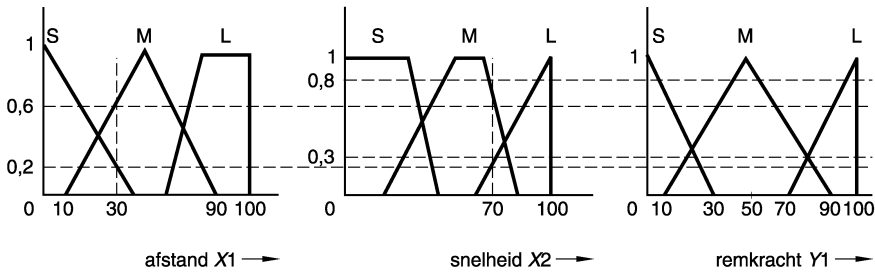
### *Stap 3*

Fuzzificeer de gekozen in- en uitgangsgrootheden van de regelaar door per grootheid geschikte membershipfuncties te kiezen qua vorm, aantal en plaats. Op deze manier kunnen linguïstische waarden die wij mensen aan in- en uitgangsgrootheden toekennen, gekoppeld worden aan fuzzy waarden. In figuur 7 is voor elke in- en uitgangsgrootheid weergegeven welke keuzen voor de membershipfuncties zijn gedaan. Het aantal membershipfuncties is voor elke grootheid hetzelfde gehouden en beperkt tot drie, met de algemene benamingen: ‘small’ (S), ‘medium’ (M) en ‘large’ (L). Dit aantal kan natuurlijk voor elke grootheid gevarieerd worden (‘very small’ kan bijv. worden toegevoegd), ofschoon de praktijk van fuzzy control leert dat het opsplitsen van een grootheid in meer dan vijf membershipfuncties in het algemeen niet leidt tot een betere regeling. Het spreekt voor zich dat ook het kiezen van de membershipfuncties (vorm, aantal en positie) een hele kunst is, waarvoor de ontwerper van de regelaar gevoel moet hebben.

### *Stap 4*

Een volgende belangrijke stap die gezet moet worden is het creëren van een rule-base. In de verzameling ‘if-then’-statements die deze rule-base bevat mag een aantal statements voor een goede werking van de regelaar absoluut niet ontbreken (zie bijv. regel 1 (statement 1) van de volgende rule-base). Sommige statements zijn minder belangrijk en zullen relatief weinig invloed hebben op het effect van de regelaar; uit de praktijk van de fuzzy regelingen blijkt ook hier dat het al dan niet toevoegen van een minder belangrijke statement – vanaf een bepaalde omvang die de rule-base dan al heeft – weinig effect meer heeft. (Dat komt waarschijnlijk door de correlatie met andere statements die wel zijn opgenomen). Het op-





**Figuur 7** Keuze van membershipfuncties

nemen van statements of regels die niet zinvol zijn, dus betrekking hebben op situaties die zich niet voordoen, moet natuurlijk worden vermeden omdat het gevolg hiervan is dat het doorlopen van de totale cyclus onnodig veel computertijd kost. Het vullen van de rule-base is weer een hele kunst, wat in de praktijk vaak pas na een aantal iteraties lukt. In dit voorbeeld beperken we ons tot vijf regels. Deze set is zeker niet compleet, maar wel bruikbaar als het gaat om de vraag hoe een en ander in z'n werk gaat. De gekozen regels zijn (waarden die buiten het gedefinieerde bereik liggen zijn natuurlijk niet van toepassing):

1. if  $\{X_1 = S \text{ and } X_2 = L\}$  then  $\{Y_1 = L\}$
2. if  $\{X_1 = L \text{ and } X_2 = S\}$  then  $\{Y_1 = S\}$
3. if  $\{X_1 = M \text{ or } X_2 = M\}$  then  $\{Y_1 = S\}$
4. if  $\{X_1 = S\}$  then  $\{Y_1 = M\}$
5. if  $\{X_1 = M \text{ and } X_2 = M\}$  then  $\{Y_1 = M\}$

*Stap 5*

Elke regel uit de rule-base levert een bepaalde bijdrage aan het uitgangssignaal van de regelaar. Het totale resultaat van al die bijdragen komt tot stand via de fuzzy inferentie. Die verloopt in ons voorbeeld als volgt:

Stel dat bij aanvang van een regelcyclus geldt:  $X_1 = 30\%$  en  $X_2 = 70\%$ . Dit levert voor de waarheidswaarde van de membershipfuncties van  $X_1$  en  $X_2$  (zie fig. 7):

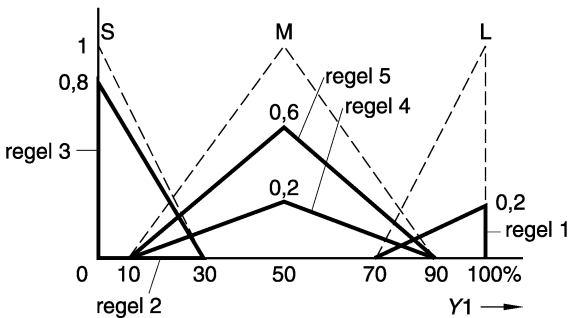
- $X_1$ :     $S = 0,2$      $M = 0,6$      $L = 0,0$
- $X_2$ :     $S = 0,0$      $M = 0,8$      $L = 0,3$

Hanteren we deze waarden voor de vijf regels uit de rule-base, dan levert dat per regel het volgende op:

1. if  $\{X_1 = S(= 0,2) \text{ and } X_2 = L(= 0,3)\}$  then  $\{Y_1 = L\}$ : evenals bij de 'and'-operator uit de logica geldt ook hier dat deze operatie als resultaat de laagste waarde van de twee operanden oplevert, dus 0,2. (Hierop zijn varianten in omloop, maar daar gaan we verder niet op in.) Het gevolg ('inferentie') hiervan is dat de membershipfunctie  $L$  van de uitgangsvaariabele  $Y_1$  voor 20% meedoet in de opbouw van het stuursignaal  $Y_1$ . We

verwezenlijken dat door in figuur 8 de membershipfunctie  $L$  met een schaalfactor 0,2 op te nemen. (Ook hierop zijn weer varianten mogelijk waar we niet op ingaan.)

2. Het 'if'-statement van regel 2 levert:  $(0,0 \text{ and } 0,0) = 0,0$ , zodat de membershipfunctie  $S$  in  $Y_1$  voor 0% meedoet (zie fig. 8).
3.  $(0,6 \text{ or } 0,8)$  levert in analogie met de binaire logica de hoogste waarde van de twee op, dus 0,8. (Ook hierop bestaan weer varianten die we ook nu weer buiten beschouwing laten.) Membershipfunctie  $S$  van  $Y_1$  doet voor 80% mee (fig. 8).
4.  $X_1$  is voor 20% small ( $S$ ), dus doet  $M$  in  $Y_1$  op basis van deze regel voor 20% mee (fig. 8).
5. Het resultaat van het 'if'-statement is 0,6;  $M$  in  $Y_1$  doet dus voor 60% mee (fig. 8).

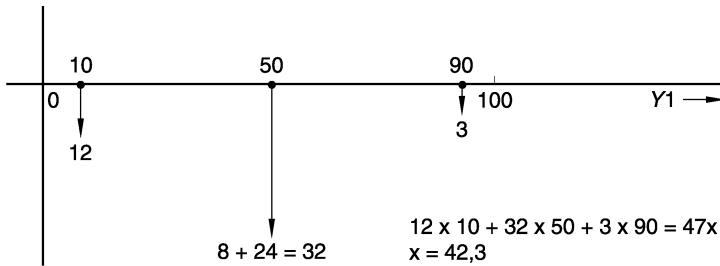


**Figuur 8** Fuzzy interferentie

### Stap 6

In figuur 8 is de fuzzy inferentie voor  $Y_1$  weergegeven op basis van de rule-base. Als laatste stap moet nu de stuurwaarde  $Y_1$  defuzzificeerd worden. We hebben immers behoefte aan een numerieke stuurwaarde voor de remkracht. Voor dit defuzzificeren zijn ook weer verschillende methoden mogelijk, maar wij beperken ons tot een – veel gebruikte – methode, de zogenaamde zwaartepuntsmethode. Die komt erop neer dat de ligging van de resultante van alle zwaartepunten van de membershipfuncties in figuur 8 bepalend is voor de remkracht. Uit figuur 9 volgt na enig rekenwerk dat deze resultante in het punt 42,3 aangrijpt, zodat de remkracht met dit percentage van de maximale remkracht wordt geactiveerd. (Let op: de plaats van de resultante is bepalend, niet de grootte.)

Nadat stap 1 tot en met 6 zijn doorlopen, worden ze opnieuw doorlopen enzovoort. De uitgangswaarde van de regelaar (dit is de stuurwaarde voor het proces) wordt dus doorlopend bijgesteld. Aan het tempo waarin dit moet gebeuren zullen natuurlijk bepaalde eisen moeten worden gesteld. Als de regelcyclus te langzaam doorlopen wordt, betekent dat een te grote dode tijd, waardoor de kans op instabiel gedrag sterk toeneemt. Te snel doorlopen heeft geen zin en kost onnodig veel computercapaciteit. Zoals gezegd zal een optimale fuzzy regeling meestal via 'trial and error' totstandkomen. Bij het ontwerp moet getracht worden



**Figuur 9** Zwaartepuntsmethode

de handelwijze van een (geroutineerde) automobilist zo goed mogelijk in kaart te brengen en na te bootsen in de fuzzy regelaar.

Fuzzy regelingen worden in de praktijk veel toegepast. Een klassiek voorbeeld is de automatisering van cementmortelcentrales; door de ervaring van een goede operator in een rule-base vast te leggen lukte dat. Het bleek dat zaken als kleur, korreligheid en dofheid van de mortel belangrijke indicatoren waren voor de kwaliteit van het product. De visuele waarneming van de operator moet wel kunnen worden overgenomen door sensoren. Vaak ligt daar een groot probleem.

Een ander populair voorbeeld van een fuzzy regeling is die in een videocamera, waarbij kleine (onbedoelde) verplaatsingen van het apparaat (door bibberingen van de hand) niet leiden tot een verplaatsing van het beeld; alleen grotere (bedoelde) verplaatsingen worden gevolgd.

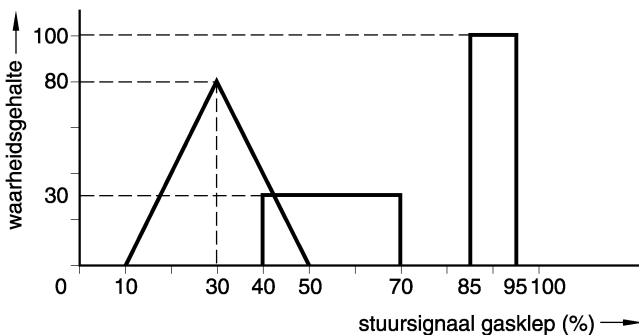
Ook de volautomatische besturing van metro's is een bekend voorbeeld uit de wereld van de fuzzy regelingen. Zelfs auto's heeft men – zonder bestuurder en met hoge snelheid – met succes over de autobaan laten rijden, maar de psychologische impact van dit soort toepassingen is groot. Medeweggebruikers accepteren geen voertuigen zonder chauffeur en voor de inzittenden van dat voertuig geldt misschien hetzelfde.

#### 4 Samenvatting

Van veel processen blijkt het gedrag niet op een wiskundige wijze beschreven te kunnen worden. Het ontwerp van een conventionele regelaar (bijv. PID) is dan onmogelijk. Een fuzzy aanpak biedt echter een oplossing voor dit probleem. In feite wordt dan de wijze waarop wij als mens een proces regelen zo goed mogelijk nagebootst door een computer. De kunst is om alle informatie die wij daar als mens bij gebruiken in kaart te brengen, evenals de manier waarop wij die verwerken en vertalen in handelingen.

## 5 Opgaven

1. Wat wordt in een fuzzy geregeld systeem bedoeld met:
  - a. het fuzzificeren van een signaalwaarde?
  - b. een membershipfunctie behorend bij een in- of uitgangssignaal van de regelaar?
  - c. de rule-base van de regelaar?
  - d. de operatie 'fuzzy inferentie' die in de regelaar wordt uitgevoerd?
  - e. het defuzzificeren van de uitgangssignalen van de regelaar?
2. We beschouwen een gasgestookte tunneloven voor de productie van baksteen. De gasklep die de toevoer van gas naar de oven bepaalt, wordt gestuurd door een signaal dat van een fuzzy regelaar afkomstig is. Stel dat de fuzzy inferentie in de regelaar voor dat signaal een resultaat oplevert zoals in figuur 10. Hoe groot is dan de gedefuzzificeerde waarde van het stuursignaal voor de gasklep op basis van de zwaartepuntsmethode?



**Figuur 10** Resultaat van de fuzzy inferentie

3. In een grote mengbak (proces) worden diverse ingrediënten (inputs) vermengd tot brooddeeg met een aantal kenmerken (outputs). Dit mengproces willen we automatiseren met behulp van een fuzzy regelaar, die de toevoer van ingrediënten completeert nadat vanuit een basisvulling is gestart.
  - a. Kies van het proces twee zinvolle uitgangsgrootheden en drie zinvolle ingangsgrootheden op basis waarvan de regelaar kan worden ontworpen.
  - b. Voorzie elk van deze grootheden van een aantal membershipfuncties en bedenk ten minste vijf zinvolle regels voor de rule-base van de regelaar.
  - c. Wat zou in dit geval de beste keuze zijn voor de tijd die ligt tussen twee opeenvolgende regelcycli: 1 seconde of 1 minuut? Motiveer het antwoord.