

CODEN: LUTFD2/(TFRT-5415)/1-56/(1989)

Implementering av en fuzzyregulator

Ulf Andersson

Institutionen för Reglerteknik
Tekniska Högskolan i Lund
Oktober 1989

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		<i>Document name</i> Master Thesis	
		<i>Date of issue</i> October 1989	
		<i>Document Number</i> CODEN: LUTFD2/(TFRT-5415)/1-56/(1989)	
<i>Author(s)</i> Ulf Andersson		<i>Supervisor</i> Lars Pernebo, SATTControl, Karl-Erik Årzén, LTH	
		<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Implementering av en fuzzyregulator. (Implementation of a fuzzy controller.)			
<i>Abstract</i> <p>The report refers to a master thesis project which has been done at SattControl AB and the Institute of Technology in Lund. The purpose of the project was to implement a fuzzy controller in SattLine, a new system of SattControl.</p> <p>The controller has been tested on two simulated processes. The first process was a cream separator where the percentage of fat in the cream was controlled. The other process was a nonlinear watertank where the level of the liquid was controlled.</p> <p>The results show that the fuzzy controller in general can be constructed to perform better than a usual fix PID controller. However, the tuning of the fuzzy controller needs a lot more work than the PID controller.</p>			
<i>Key words</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i>			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 56	<i>Recipient's notes</i>	
<i>Security classification</i>			

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

REFERAT

Rapporten avser ett examensarbete utfört på SattControl AB och Lunds Tekniska Högskola. Arbetet gick ut på att implementera en fuzzyregulator i SattControls nya system SattLine.

Regulatorn har testats på två simulerade processer. Den ena processmodellen var en gräddseparator där fetthalten på grädden skulle regleras. Den andra processmodellen var en olinjär vattentank med fritt utlopp där vätskenivån skulle regleras.

Resultaten visar att den implementerade fuzzyregulatorn generellt sett kan göras bättre än en fix PID-regulator. Inställningen av fuzzyregulatorn kräver dock större arbetsinsatser än motsvarande inställning av PID-regulatorn.

INNEHÅLLSFÖRTECKNING

1	INLEDNING	4
2	HISTORIK	5
3	TEORI	7
	3.1 Oskarp logik	7
	3.2 Lingvistiska variabler.....	8
	3.3 Oskarpa mängder	9
	3.4 Villkorliga relationer mellan oskarpa mängder.....	11
	3.5 Generering av utsignal	15
4	IMPLEMENTERING	17
	4.1 Grafiska byggblock	17
	4.1.1 Insignalinterface	19
	4.1.2 Tillhörighetsfunktioner i villkorsdelen.....	20
	4.1.3 Logiska operatorer	21
	4.1.4 Regelblock.....	22
	4.1.5 Tillhörighetsfunktioner i slutsatsdelen	22
	4.1.6 Utsignalberäknare	25
	4.1.6.1 Tyngpunktsmetoden	25
	4.1.6.2 Ytdelningsmetoden	27
	4.1.7 Utsignalinterface	29
	4.1.8 Presentatörer	29
	4.2 Konstruktion av en fuzzyregulator.....	29
	4.2.1 Regelval.....	30
	4.2.2 Sammankoppling av byggblocken	31
5	SIMULERINGAR	33
	5.1 Gräddseparator	33
	5.1.1 Fuzzyregulatorn	34
	5.1.2 Resultatet	35
	5.2 Vattentank.....	36
	5.2.1 Fuzzyregulatorn	37
	5.2.2 Resultatet	39

6	SAMMANFATTNING OCH KOMMENTARER	40
7	REFERENSER	41

BILAGOR

BILAGA 1	Tillhörighetsfunktionernas initialutseende	42
BILAGA 2	Presentatörernas utseende.....	43
BILAGA 3	Reglering av gräddseparatorn	44
BILAGA 4	Reglering av vattentanken	49

1 INLEDNING

Det finns många processer som är svåra att automatisera. Skälet till detta är att dessa processer är så olinjära och komplexa att det är svårt att ställa upp bra matematiska modeller för dem. Trots detta klarar mänskliga operatörer av att styra dessa processer på ett godkänt sätt. Ofta använder dessa operatörer tumregler för hur styringreppen skall göras. Dessa tumregler har tagits fram av personer med stor kännedom om processen. En sådan kännedom bygger mer på en ren känsla för hur processen reagerar än på fysikaliska lagar. Detta gör det svårt att översätta tumreglerna till matematiska ekvationer. Det man skulle behöva är en regulator som använder människans komplicerade sätt att resonera.

En fuzzyregulator är en regelbaserad regulator. Dess regler relaterar styringreppen hos regulatorn med processens variabelvärden. Reglerna bygger inte på klassisk binär logik utan på oskarp (fuzzy) logik och oskarpa mängder. Reglerna får därmed den egenskapen att de kan vara uppfyllda till en viss grad. Hur mycket reglerna sedan bidrar till slutresultatet beror just på denna grad av uppfyllnad.

Om man nu intervjuar den mänskliga operatören och översätter hans tumregler för hur processen skall styras till lingvistiska regler av typen

"Om A är hög och B är låg så gör C stor"

och definierar uttrycken hög, låg etc. som oskarpa mängder, så får man en uppsättning regler till en fuzzyregulator. Dessa regler får man givetvis justera efter hand som man testkör den nya regulatorn. Svårigheten med en fuzzyregulator är dels att få fram regler som täcker alla eventuella processhändelser och dels att definiera de lingvistiska uttrycken hög, låg, stor, liten etc. som oskarpa mängder.

Teorin bakom det hela uppkom på 60-talet och sedan dess har många försök med fuzzyregulatorer presenterats. Trots att många av dessa fungerat mycket väl är det endast ett fåtal som nått den industriella marknaden.

Syftet med detta examensarbete har varit att undersöka om det är möjligt att implementera fuzzyregulatorer i SattControls nya system SattLine. Om det gick att implementera en sådan fuzzyregulator så skulle den testas på två simulerade processer. Den ena processmodellen är en gräddseparator där fetthalten hos grädden skall regleras. Den andra är en olinjär vattentank där vätskenivån skall regleras.

2 HISTORIK

Utvecklingen av fuzzyregulatorer har sitt ursprung i den matematiska läran om oskarpa mängder (Fuzzy Set theory), som publicerades av *Zadeh* [1] år 1965. Denna lära kan tolkas som ett slags gränssnitt mellan heuristik och numerik.

Ideén att använda oskarpa mängder i algoritmer för datorer växte fram i början på sjuttioalet. En detaljerad beskrivning av dessa fuzzy-algoritmer publicerades år 1973 av *Zadeh* [2].

Den första fuzzyregulatorn presenterades år 1975 av *Mamdani* och *Assilian* [3]. Regulatorn konstruerades på Queen Mary College i London och reglerade hastigheten och ångtrycket hos en liten ångmaskin. Kort därefter undersökte *Kickert* och *Lemke* [4] hur en fuzzyregulator klarade av att reglera temperaturen i en varmvattentank. Dessa resultat ledde till att även andra började undersöka fuzzyregulatorer och många positiva resultat rapporterades. Trots att de positiva resultaten visade att fuzzyregulatorer uppförde sig bättre än konventionella PID-regulatorer vid processreglering så stannade deras regulatorer kvar på forskningsstadiet och slog inte igenom på den industriella marknaden.

Försök med en adaptiv fuzzyregulator presenterades först av *Procyk* och *Mamdani* [5] år 1977. Den hade förmågan att förändra sina lingvistiska regler under exekveringens gång.

Först 1980 kom den första industriella fuzzyregulatorn. Det var det danska företaget F.L.Smith som kom med en regulator för förbränningen i cementugn [6]. Försök att styra cementugnar med konventionella metoder hade tidigare visat sig ge klena resultat. Detta beror på att en cementugn är en så komplex process att det är svårt att ställa upp en generell modell för den. Problemet har varit att antingen har modellen man fått fram varit för enkel för att fungera eller för exakt för att kunna gälla för andra cementugnar.

De två brittiska företagen Blue Circle Industries PLC och Sira Ltd. utvecklade tillsammans en fuzzyregulator för processindustrin. Regulatorn kallades LINKman och utvecklades framförallt för cementindustrin.

Den första integrerade kretsen med oskarp logik utvecklades av *Togai* och *Watanabe* vid Bell Telephone Laboratories år 1985. Den har en kapacitet på 250 000 FLIPS (Fuzzy Logic Inferences Per Second) när den drivs med en 16 MHz klocka.

I april 1987 presenterades en fuzzy-dator, konstruerad av *Yamakawa* vid Kumamoto University. Dess beräkningskapacitet är 10 megaFLIPS.

I september 1988 släpptes en fuzzyregulator av det japanska företaget Micom Inc. Regulatorn är baserad på ett höghastighets CMOS LSI som gör det möjligt att göra beräkningar med 60 megaFLIPS.

Den 1:e april 1989 startades LIFE (Laboratory for Internationell Fuzzy Engineering Research) i Japan. Det är 46 privata företag som tillsammans med nationella institutioner och universitet kommer att syssla med utveckling inom områden där fuzzy-teknik kan användas.

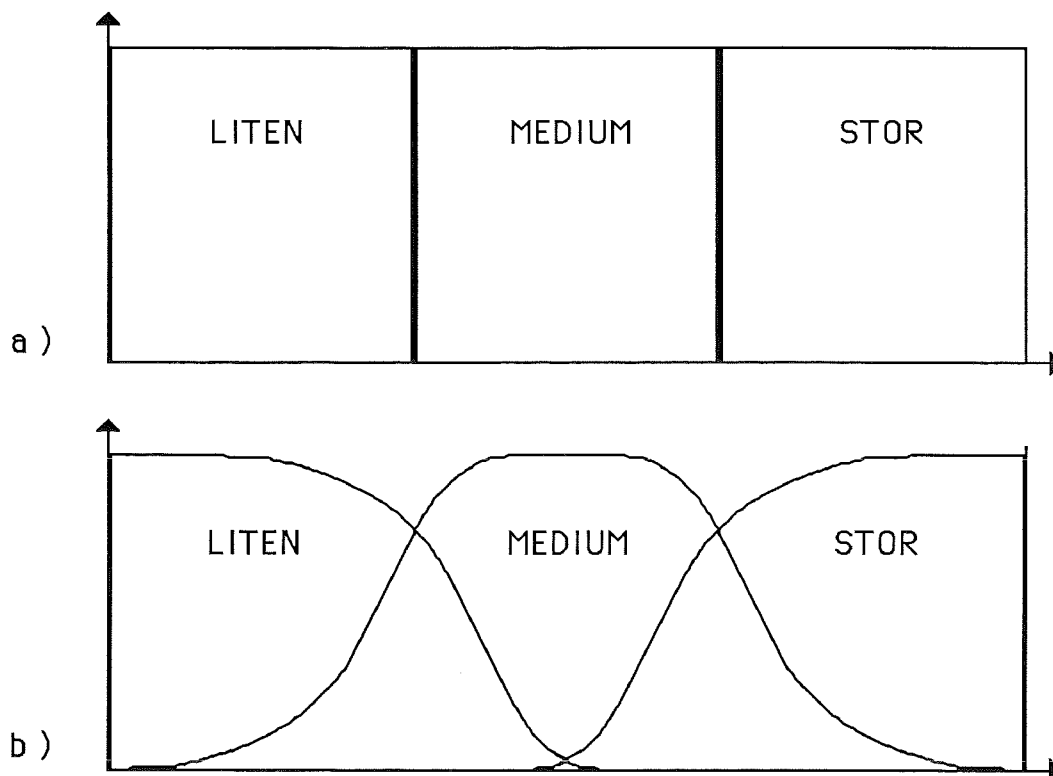
Den snabba utvecklingen på hårdvarusidan kommer säkerligen att öka användningsområdet för den oskarpa logiken. Japanerna satsar för närvarande hårt inom området och kommer troligen att inom en snar framtid visa upp sina resultat.

3 TEORI

Teorin bakom fuzzyreglering bygger helt och hållet på den oskarpa logiken. Enligt den oskarpa logikens upphovsman *Zadeh* [2] så definieras den oskarpa logiken som "logiken som ligger till grund för den approximativa bevisföringens metoder".

3.1 Oskarp logik

Det som främst karakteriserar binär logik är att ett påstående antingen är sant eller falskt. Den oskarpa logiken har i stället en kontinuerlig övergång mellan de två värdena. Om man i den binära logiken låter 0 och 1 motsvara falskt respektive sant så kan variabler i den oskarpa logiken anta vilka värden som helst i intervallet $[0,1]$. Se även figur 3.1 som grafiskt visar skillnaden mellan binär och oskarp logik för de tre kvantitativa termerna "Stor", "Medium" och "Liten".



Figur 3.1. Figuren visar ett exempel på hur de tre kvantitativa nivåerna "Liten", "Medium" och "Stor" kan representeras i binär logik (a) respektive oskarp logik (b).

Den oskarpa logikens approximativa resonemang lämpar sig väl inom många applikationer. Man kan bl.a. nämna "icke fysiska" system som de ekonomiska och medicinska, men även inom området processreglering är den oskarpa logiken användbar. Detta gäller speciellt processer som är så komplexa eller olinjära att det är svårt att ställa upp bra modeller för dem.

Framför allt närmar sig den oskarpa logiken människans sätt att tänka, resonera och uttrycka sig. Människan använder sig ofta av lingvistiska uttryck som "stor", "gammal", "lång" etc. för att beskriva saker. Sådana termer har oftast suddiga eller diffusa gränser vilket passar den oskarpa logiken, där till och med "sanningen" är en fråga om värdering.

3.2 Lingvistiska variabler

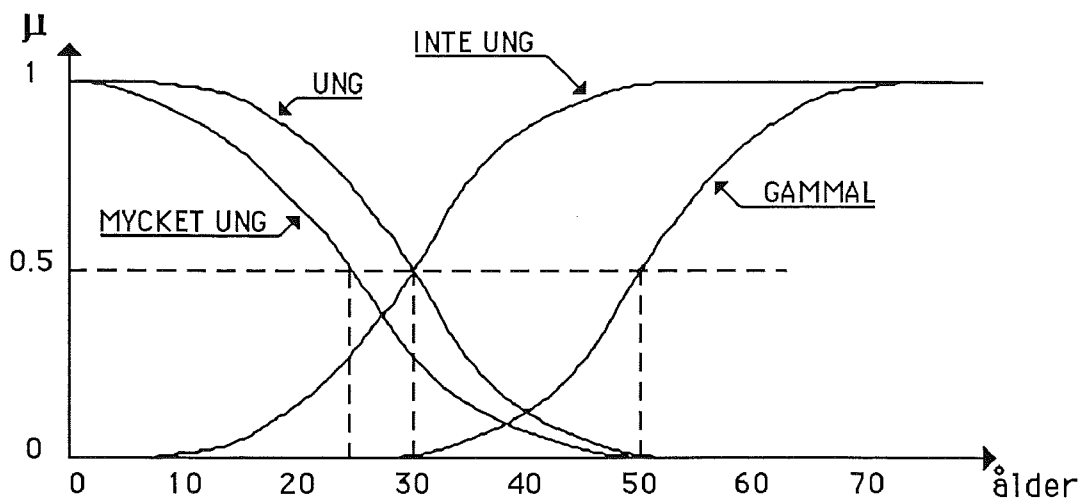
En lingvistisk variabel är en variabel vars värden är ord eller uttryck i naturligt språk. Ett vanligt exempel i det här sammanhanget är den lingvistiska variabeln "Ålder" vars värden kan vara "ung", "inte ung", "mycket ung", "ganska ung", "gammal" etc. Med lingvistiska variabler kan man därför beskriva saker som annars är matematiskt svårdefinierade.

De lingvistiska variablernas värden representeras av en sorts sannolikhetsfördelningar som brukar kallas tillhörighetsfunktioner, se figur 3.2. Tillhörighetsfunktionernas värde ligger i intervallet $[0,1]$ där värdet 1 motsvarar full tillhörighet och 0 avsaknad av tillhörighet. När man använder sig av lingvistiska variabler i reglersammanhang är det oftast i så kallade oskarpa algoritmer (fuzzy-algoritmer). Dessa algoritmer består då oftast av ett antal lingvistiska regler av typen

```
IF "temperaturen är mycket hög" THEN "minska trycket väsentligt"  
ELSE  
IF "temperaturen är hög" THEN "minska trycket en aning"  
ELSE  
IF "temperaturen ...
```

där temperaturen och trycket är lingvistiska variabler.

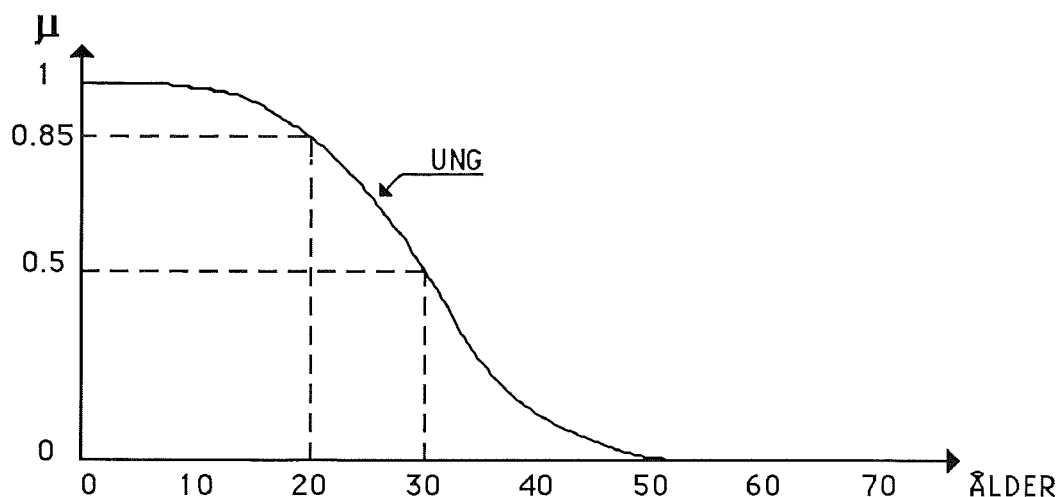
Ett annat sätt att se på de lingvistiska variablernas värden är som oskarpa delmängder av den lingvistiska variabelns definitions-mängd.



Figur 3.2. Exempel på hur olika tillhörighetsfunktioner kan se ut för den lingvistiska variabeln ålder.

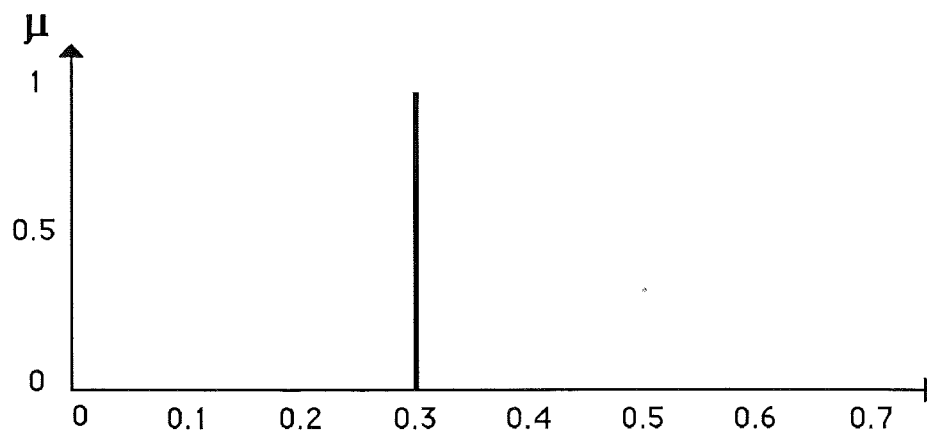
3.3 Oskarpa mängder

En oskarp delmängd A av mängden U karakteriseras av sin tillhörighetsfunktion μ_A . Denna ger varje element u i U ett värde $\mu_A(u)$ i intervallet $[0,1]$. Detta värde motsvarar graden av tillhörighet för elementet u i A . Se även figur 3.3 som visar ett exempel på en tillhörighetsfunktion och dess värden.



Figur 3.3. Om den oskarpa delmängden "Ung" av mängden "Ålder" har utseendet enligt figuren så är en 20-årig person ung till graden 0,85. Det vill säga $\mu_{UNG}(20) = 0,85$.

Även deterministiska värden kan anges som oskarpa delmängder. Ett exempel på detta är en mätsignal från en process. När signalen är uppmätt så har signalens tillhörighetsfunktion värdet 1 för just detta värde och 0 för alla övriga värden, se figur 3.4. En sådan oskarp delmängd kallas på engelska för en singleton.



Figur 3.4. Tillhörighetsfunktionen för ett deterministiskt värde kallas för en singleton och kan se ut som figuren visar.

Följande tre basoperatorer används på oskarpa delmängder

Unionen mellan två oskarpa delmängder A och B av samma mängd U betecknas A+B. Dess tillhörighetsfunktion är definierad enligt

$$\mu_{A+B}(u) = \max[\mu_A(u); \mu_B(u)] : u \text{ är ett element i } U$$

Unionen motsvaras av den logiska operatören ELLER (eng. OR).

Snittet mellan två oskarpa delmängder A och B av samma mängd U betecknas AB. Dess tillhörighetsfunktion är definierad enligt

$$\mu_{AB}(u) = \min[\mu_A(u); \mu_B(u)] : u \text{ är ett element i } U$$

Snittet motsvaras av den logiska operatören OCH (eng. AND).

Komplementet till den oskarpa delmängden A av mängden U betecknas $\neg A$. Dess tillhörighetsfunktion är definierad enligt

$$\mu_{\neg A}(u) = 1 - \mu_A(u) : u \text{ är ett element i } U$$

Komplementet motsvaras av den logiska operatören INTE (eng. NOT).

3.4 Villkorliga relationer mellan oskarpa mängder

När man använder sig av den oskarpa logiken i reglersammanhang måste man kunna beräkna följande slutsats

Regel: If x is A then y is B
Givet: x is A'
Slutsats: y is B'

där A och A' är oskarpa delmängder till U och B och B' är oskarpa delmängder till V. Till exempel kan det finnas en regel som säger "if x is High then y is Low". Om det då i stället är så att "x is Zero", vad blir då y.

För att kunna beräkna detta B' behövs först och främst en tolkning av det villkorliga uttrycket "if A then B", som även kan skrivas " $A \Rightarrow B$ ". I binär logik skulle detta motsvarats av en sanningstabell, se figur 3.5. Den första tanken är då att göra en direkt översättning till den oskarpa logiken. Denna tolkning har emellertid visat sig ge dåliga egenskaper hos den resulterande regulatorn och en annan tolkning är nödvändig. Den tolkning av " $A \Rightarrow B$ " som gett bäst resultat är att se det som en oskarp relation mellan två oskarpa variabler. Detta betyder att " $A \Rightarrow B$ " är en oskarp mängd av elementpar (u,v), där u är element i U och v är element i V. Graden av tillhörighet för (u,v) ges som tidigare av en tillhörighetsfunktion $\mu_{A \Rightarrow B}(u,v)$.

Resultatet av detta är att " $A \Rightarrow B$ " skall ses som en oskarp relation från mängden U till V. Denna tolkning har i sin tur gett upphov till flera tämligen likvärdiga uttryck för relationens tillhörighetsfunktion. Av dessa kommer endast två att tas upp här. Den som vill läsa mer om olika oskarpa relationer hänvisas till Mizumoto [7].

		B	
		FALSK	SANN
A	SANNINGS VÄRDE		
	FALSK	SANN	SANN
SANN	FALSK	SANN	

Figur 3.5. I binär logik kan " $A \Rightarrow B$ " representeras i form av en sanningstabell.

De flesta existerande regulatorer använder sig av *Mamdani's* tolkning av relationen "A => B". Han definierar relationen som den kartesiska produkten mellan A och B, som betecknas AxB. Enligt denna definition ges relationens tillhörighetsfunktion av

$$\mu_{A \Rightarrow B}(u,v) = \mu_{A \times B}(u,v) = \min[\mu_A(u); \mu_B(v)]$$

Den andra tolkningen som tas upp här är den som används i F.L.Smith's fuzzyregulator för cementugnar. Relationen definieras där som den aritmetiska produkten i stället för den kartesiska och dess tillhörighetsfunktion ges av

$$\mu_{A \Rightarrow B}(u,v) = \mu_A(u) \cdot \mu_B(v)$$

Då det nu finns möjlighet att beräkna den oskarpa relationens tillhörighetsfunktion $\mu_{A \Rightarrow B}(u,v)$ kan även slutsatsen B' beräknas. Detta görs med hjälp av max-min sammansättningen (eng. *compositional rule of inference*) av den oskarpa delmängden A' och den oskarpa relationen "A => B". Detta skrivs som

$$B' = A' \circ (A \Rightarrow B)$$

och motsvarande tillhörighetsfunktion ges av

$$\mu_{B'}(v) = \max_u(\min[\mu_{A'}(u) ; \mu_{A \Rightarrow B}(u,v)])$$

För tydlighets skull visas i figurerna 3.6 och 3.7 ett exempel på hur tillhörighetsfunktionen för slutsatsen B' grafiskt kan bildas för givna A,A' och B.

Oftast finns det många regler

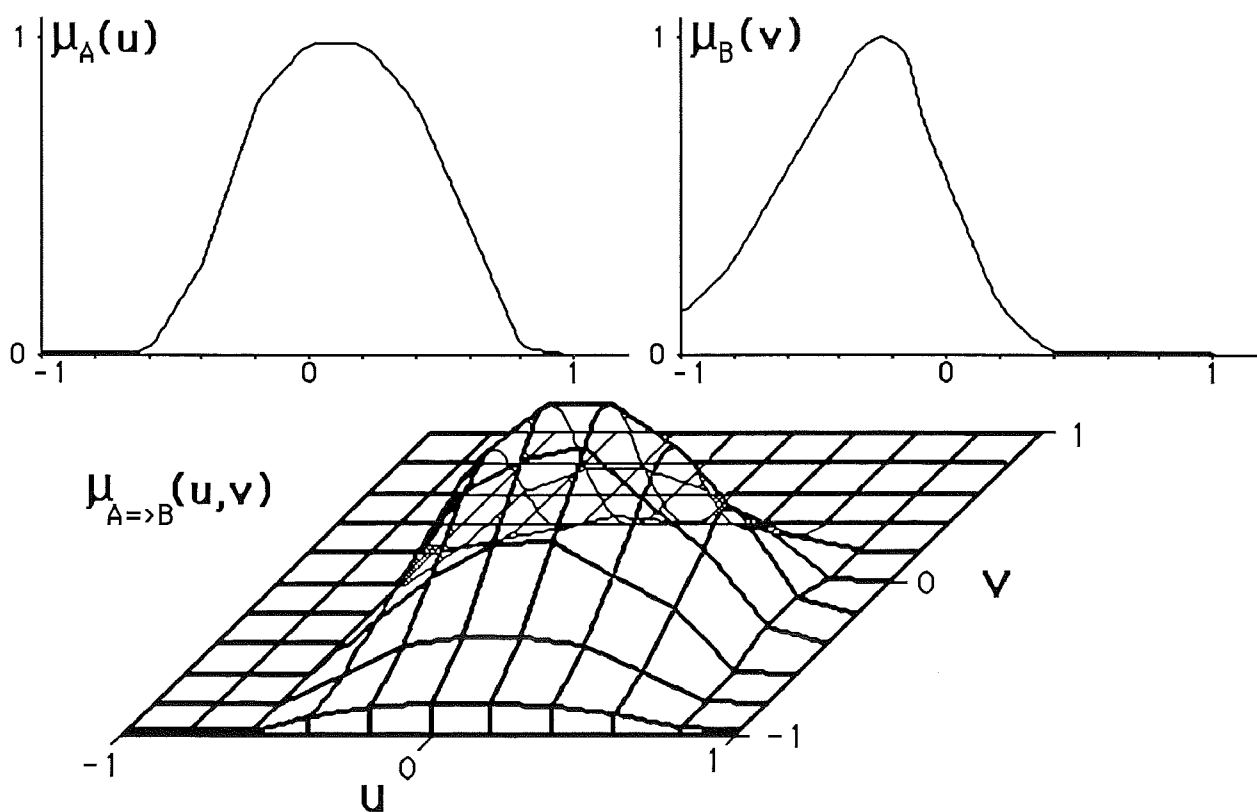
$$\text{if } A_1 \text{ then } B_1 \quad \text{else} \quad \text{if } A_2 \text{ then } B_2 \quad \text{else} \quad \dots \quad \text{else} \quad \text{if } A_n \text{ then } B_n$$

Den totala slutsatsen bildas då genom att ta maximum över samtliga delslutsatser. Tillhörighetsfunktionen för den totala slutsatsen $\mu_{B'_{Tot}}$ beräknas då enligt

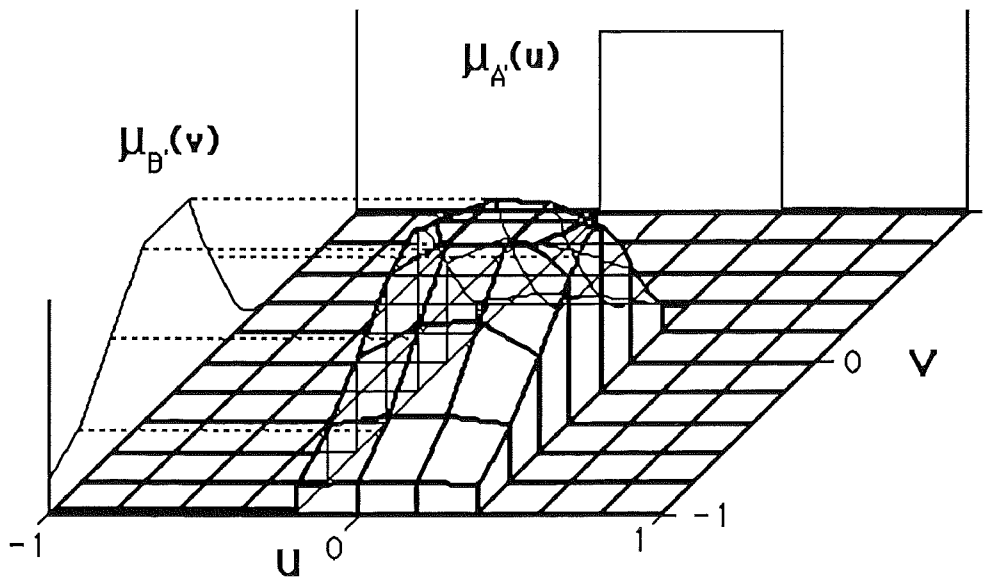
$$\mu_{B'_{Tot}}(v) = \max \left[\begin{aligned} &\max_u(\min[\mu_{A'}(u) ; \mu_{A_1 \Rightarrow B_1}(u,v)]) ; \\ &\max_u(\min[\mu_{A'}(u) ; \mu_{A_2 \Rightarrow B_2}(u,v)]) ; \\ &\dots \\ &\max_u(\min[\mu_{A'}(u) ; \mu_{A_n \Rightarrow B_n}(u,v)]) \end{aligned} \right]$$

När den oskarpa delmängden A' är en singleton u_0 (dvs. $\mu_{A'}(u_0) = 1$ och $\mu_{A'}(u) = 0$ för $u \neq u_0$) så förenklas beräkningarna av slutsatsen B' väsentligt, ty för fallet med en enda regel beräknas tillhörighetsfunktionen för B' enligt

$$\begin{aligned} \mu_{B'}(v) &= \max_u (\min[\mu_{A'}(u) ; \mu_{A \Rightarrow B}(u,v)]) = \\ &= \max[\max_{u \neq u_0} (\min[\mu_{A'}(u) ; \mu_{A \Rightarrow B}(u,v)]) ; \\ &\quad (\min[\mu_{A'}(u_0) ; \mu_{A \Rightarrow B}(u_0,v)])] = \\ &= \max[\max_{u \neq u_0} (\min[0 ; \mu_{A \Rightarrow B}(u,v)]) ; \\ &\quad (\min[1 ; \mu_{A \Rightarrow B}(u_0,v)])] = \\ &= \mu_{A \Rightarrow B}(u_0,v) \end{aligned}$$



Figur 3.6. Figuren visar hur tillhörighetsfunktionen för den oskarpa relationen $\mu_{A \Rightarrow B}(u,v)$ grafiskt kan bildas enligt F.L.Smidth's tolkning.



Figur 3.7. Figuren visar hur max-min sammansättningen mellan $\mu_{A \Rightarrow B}(u,v)$ från figur 3.6 och $\mu_{A'}(u)$, definierad i figuren, kan tolkas grafiskt. $\mu_{A'}(u)$ skär ut ett snitt i $\mu_{A \Rightarrow B}(u,v)$ och allt som är utanför snittet kastas. $\mu_{B'}(v)$ bildas sedan genom att projicera den resterande ytan på v -planet till vänster i figuren.

Det vill säga att när A' är en singleton beräknas slutsatsen B' enligt

$$\mu_{B'}(v) = \mu_{A \Rightarrow B}(u_0, v)$$

Detta är slutsatsen till en regel. Skillnaden mellan de ovan givna tolkningarna av " $A \Rightarrow B$ " visas i figur 3.8, där A' är en singleton u_0 .

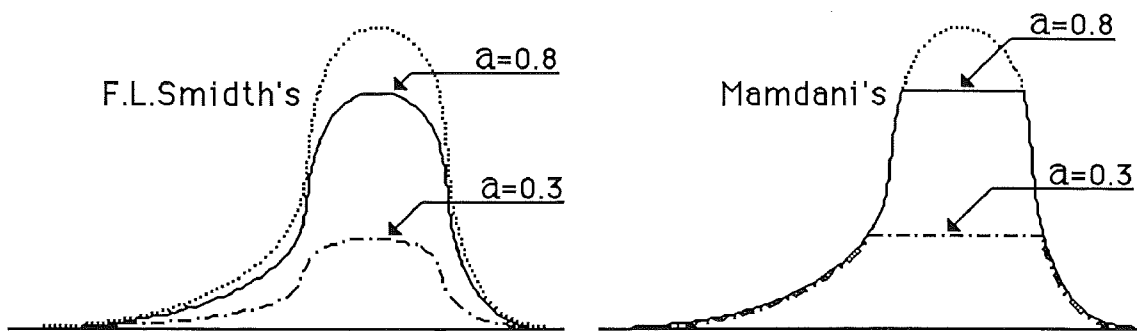
Resultatet kan sedan utvidgas till fallet med många regler

if A_1 then B_1 else if A_2 then B_2 else ... else if A_n then B_n

Om A' är en singleton så ges tillhörighetsfunktionen för B' av

$$\mu_{B'}(v) = \max[\mu_{A_1 \Rightarrow B_1}(u_0, v) ; \dots ; \mu_{A_n \Rightarrow B_n}(u_0, v)]$$

I reglersammanhang motsvaras den oskarpa delmängden A' av ett mätvärde eller en sammansättning av mätvärden (reglerfelet). Oftast anser man att detta värde är tillräckligt tillförlitligt och att A' skall vara en singleton. Men om mätsignalen är brusig finns det ett alternativ. Då skulle A' t.ex. kunna ha ett utseende liknande en normalfördelningskurva. Bredden på A' skulle motsvara brusets standardavvikelse och man skulle därför ta hänsyn till osäkerheten i mätvärdet. Tyvärr ökar detta antalet beräkningar kraftigt och alternativet mister därför sin eventuella betydelse.



Figur 3.8. Skillnaden mellan F.L.Smidth's och Mamdani's tolkningar av relationen "A => B". De streckade kurvorna i figuren visar $\mu_B(v)$ och de andra kurvorna visar $\mu_{B'}(v)$ då A' är en singleton u_0 och $\mu_A(u_0)=a$

3.5 Generering av utsignal

Det enda som återstår är att ur en given tillhörighetsfunktion $\mu_{B'}(v)$ plocka ut ett enda element v , som då skall representera hela den oskarpa delmängden B' . Med andra ord skall man konvertera B' till en singleton v_0 .

Även här finns det många varianter varav endast ett fåtal kommer att beskrivas här. Den enklaste går ut på att som utsignal välja det v som maximerar $\mu_{B'}(v)$, dvs. välja v_0 så att

$$\mu_{B'}(v_0) = \max_v [\mu_{B'}(v)]$$

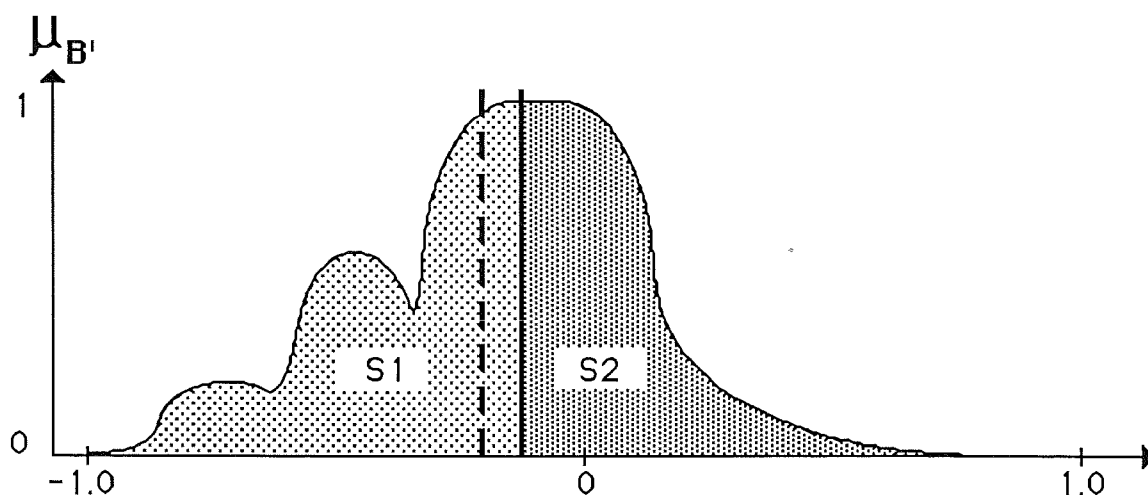
Om det skulle finnas flera maxima eller maximat är en plåtå bildas i stället ett medelvärde dessa och används som utsignal. Detta alternativet ger dock inget kontinuerligt förhållande mellan in- och utsignal utan man får i stället ett slags multirelä. Den metod som då är att föredra enligt Braae och Rutherford [8] är att välja utsignalen v_0 som tyngdpunkten av B' . Utsignalen beräknas enligt

$$v_0 = \int v \cdot \mu_{B'}(v) dv / \int \mu_{B'}(v) dv$$

Ett tredje alternativ är att göra som F.L.Smidth, nämligen att som utsignal välja det v_0 som delar $\mu_{B'}(v)$ i två lika stora delar, se figur 3.9.

Av dessa tre alternativ är de två sistnämnda att föredra även om de medför tyngre beräkningar än det första alternativet. Skillnaden mellan de två sistnämnda alternativen illustreras lättast genom ett exempel. Antag att man har en kurva med utsignalerna beräknade enligt de båda metoderna. Om man nu lägger till två lika

stora ytbidrag på var sin sida om utsignallinjen i de båda fallen så kommer utsignalen från ytdelningsmetoden att förbli densamma. Om de båda ytbidragen inte placeras på samma avstånd från den gamla tyngdpunkten kommer denna att förändras. På så sätt blir tyngpunktsmetoden känsligare för förändringar långt i från tyngdpunkten.



Figur 3.9. Vid beräkning av utsignalen finns det två metoder som kan rekommenderas. Den första går ut på att välja tyngdpunktslinjen (streckade linjen) för den skuggade ytan i figuren som utsignal. Den andra går ut på att dela upp ytan i två lika stora delar S1 och S2 och välja delningslinjen (heldragna linjen) som utsignal.

4 IMPLEMENTERING

All implementering har gjorts i SattLine på SattControl AB i Lund. Systemet är baserat på grafisk programmering i separata grafiska moduler. Dessa moduler innehåller all den grafik som skall presenteras för användaren. Eventuell programkod ligger inkluderad i grafiken som ekvationsboxar. För att nå information som ligger "djupt" inne i modulerna har man som hjälpmedel ett zoomverktyg. På detta sätt finns all nödvändig information lätt tillgänglig för användaren.

4.1 Grafiska byggblock

Konstruerandet av en fuzzyregulator görs med hjälp av färdiga byggblock som består av en eller flera grafiska moduler. Byggblocken ligger i ett bibliotek och man hämtar lätt fram de block man behöver. Följande byggblock har implementerats

- Insignalinterface
- Tillhörighetsfunktioner i villkorsdelen
- Logiska operatorer
- Regelblock
- Tillhörighetsfunktioner i slutsatsdelen
- Utsignalberäknare
- Utsignalinterface
- Presentatörer

Sammankopplingen av byggblocken görs med hjälp av grafiska kopplingar. En grafisk koppling är en överföringslänk mellan modulerna och kan överföra en variabel. Denna variabel kan dock vara en record som i sin tur kan innehålla många variabler.

För att visa hur byggblocken används i en fuzzyregulator visas ett enkelt exempel med en enda regel.

Regel: If e is High AND Δe is Zero THEN Δu is Low

där e är reglerfelet, Δe är mätvärdets förstadifferens och Δu är styrsignalens inkrement. Mätvärdets förstadifferens representerar egentligen felets förstadifferens men då man inte vill ha med börvärdesförändringar i differentieringen används istället mätvärdet.

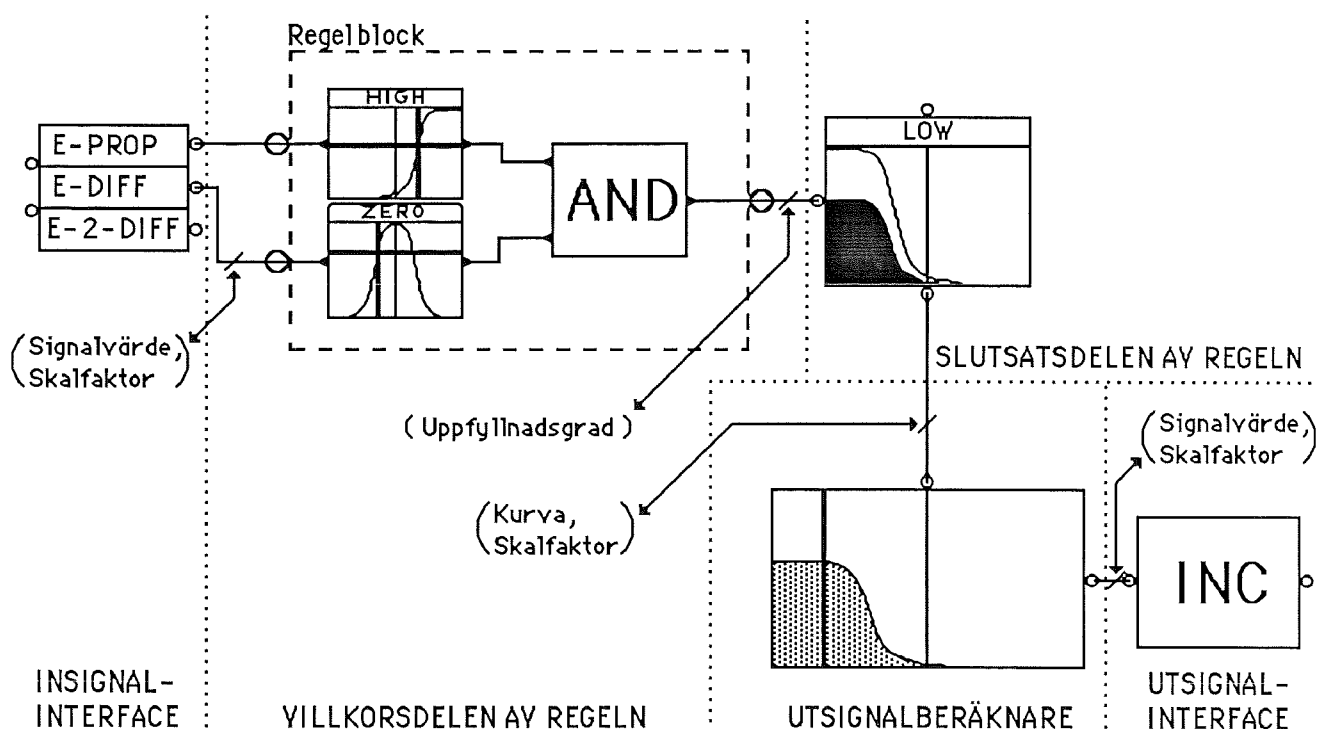
Man börjar då med att plocka in börvärde och mätvärde från processen till insignalinterfacet till vänster i figur 4.1. Interfacet skickar reglerfelet samt processvärdets första- och andradifferens

in i regulatorn till villkorsdelen av regeln. I detta exempel används bara felet och mätvärdets förstadifferens i regeln. I regelblocket kopplas villkorsdelens tillhörighetsfunktioner ihop med hjälp av logiska operatörer. Resultatet skickas sedan vidare till slutsatsdelens tillhörighetsfunktioner. Finns det många regler länkas slutsatsdelens tillhörighetsfunktioner samman och dess slutresultat skickas vidare till utsignalberäknaren. Den utsignal man får ut av utsignalberäknaren skickas till utsignalinterfacet, som i sin tur skickar ut den till det övriga systemet

Om man har flera utsignaler kan reglerna se ut som

If e is High AND Δe is Zero THEN Δu_1 is Low AND Δu_2 is Medium

Eftersom villkorsdelen är densamma för båda utsignalerna är det bara slutsatsdelen som behöver modifieras. Detta görs lättast genom att man för varje utsignal har en uppsättning med slutsatsdelens tillhörighetsfunktioner, utsignalberäknare och ett utsignalinterface. På det viset får man två parallella slutsatsdelar men bara en villkorsdel.



Figur 4.1. Den grafiska beskrivningen av en regulator består av fem olika delar. Dessa är markerade i figuren med hjälp av punktlinjer. I figuren visas en regulator som bara består av en enda regel. "If e is High AND Δe is Zero THEN Δu is Low". Skalfaktorn i figuren består av max- och mingränser till tillhörighetsfunktionerna.

4.1.1 Insignalinterface

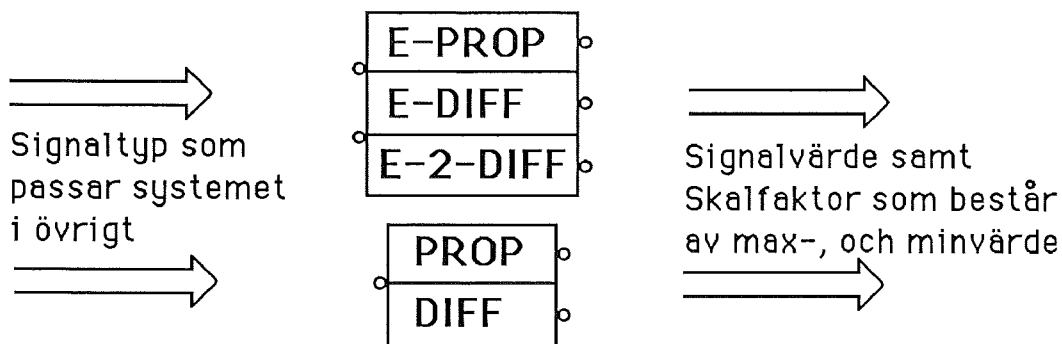
Det finns två byggblock som tar hand om insignalsidans gränssnitt mellan fuzzyregulatorn och det övriga systemet, se figur 4.2. I Signalsnittet in mot regulatorn ingår förutom signalvärdet även en skalfaktor till tillhörighetsfunktionerna. Denna skalfaktor bestämmer max- och mingränserna för villkorsdelens tillhörighetsfunktioner i x-led.

Det ena insignalinterfacet har som yttre insignaler börvärdet och processens mätvärde. Interfacet bildar felet e , mätvärdets förstadifferens Δe och mätvärdets andradifferens $\Delta^2 e$.

$$\begin{aligned} e_t &= \text{börvärde}_t - \text{mätvärde}_t \\ \Delta e_t &= (\text{mätvärde}_{t-1} - \text{mätvärde}_t) / T_{\text{samp}} \\ \Delta^2 e_t &= (\Delta e_t - \Delta e_{t-1}) / T_{\text{samp}} \end{aligned}$$

samt skickar vidare dessa till fuzzyregulatorn. Att mätvärdets första- och andradifferens betecknas med Δe respektive $\Delta^2 e$ beror på att de representerar felets första- och andradifferens. Skälet till att det är processens mätvärde och inte felet som differentieras är att annars ger stegförändringar i börvärdet för stora utslag i regulatorns styrsignal. Det andra insignalinterfacet har bara en yttre insignal. In till fuzzyregulatorn skickas denna signal och dess förstadifferens.

Eftersom regulatorn är implementerad på inkrementell form så kan man med hjälp av de tre insignalerna kunna få något som påminner om PID-struktur.



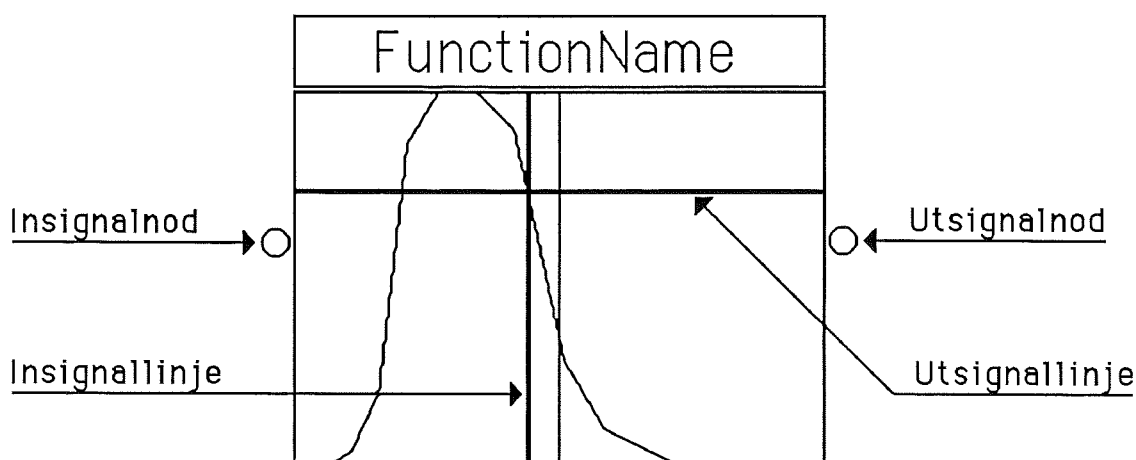
Figur 4.2. På insignalsidan finns det två olika interfaceblock. Det vänstra bildar reglerfelet samt börvärdets första- och andradifferens. Det högra blocket skickar vidare insignalen och dess förstadifferens.

4.1.2 Tillhörighetsfunktioner i villkorsdelen

Det finns två uppsättningar av tillhörighetsfunktionerna. Den första används i villkorsdelen och den andra i slutsatsdelen av reglerna. Båda uppsättningarnas tillhörighetsfunktioner består av 10 punkter. Mellan punkterna används linjär interpolering. Man kan ändra punkternas lägen och intervallet för tillhörighetsfunktionen under exekvering. Punkternas lägen ändrar man genom att göra en musklickning i byggblocket. Då får man upp ett fönster där punktparens numeriska värde visas. Dessa kan man sedan enkelt ändra. Intervallet för tillhörighetsfunktionen ändras via interfacen. För villkorsdelens tillhörighetsfunktioner ändras intervallet via insignalinterfacen och för slutsatsdelens via utsignalinterfacet.

Det finns 12 olika fördefinierade tillhörighets-funktioner med initialutseende enligt bilaga 1.

Low	Large negative	Large positive
High	Medium negative	Medium positive
Medium	Small negative	Small positive
Zero	Zero negative	Zero positive



Figur 4.3. Tillhörighetsfunktionerna i villkorsdelen har två grafiska kopplingsnoder som används vid sammankoppling med andra byggblock. In- och utsignalerna visas grafiskt i tillhörighetsfunktionens graf under exekveringen med hjälp av varsin linje. Under exekveringen visas även tillhörighetsfunktionens namn (t.ex. "High") där det i figuren står "FunctionName"

Den uppsättning av tillhörighetsfunktionerna som används i villkorsdelen av reglerna har som insignal en post som består av ett signalvärde och en skalfaktor. Skalfaktorn anger tillhörighetsfunktionens max- och mingränser i x-led. Som utsignal lämnas ett värde mellan 0 och 1. Detta värde motsvarar graden av tillhörighet för insignalen i tillhörighetsfunktionen. Om t.ex. tillhörighetsfunktionen är "High" och insignalens värde är x, då beräknas utsignalens värde y enligt

$$y = \mu_{\text{High}}(x)$$

Om insignalvärdet ligger utanför max- eller mingränsen så ger tillhörighetsfunktionen gränsvärdet som utsignal. Om till exempel insignalens värde x är mindre än mingränsen x_{min} så beräknas utsignalens värde enligt

$$y = \mu_{\text{High}}(x_{\text{min}})$$

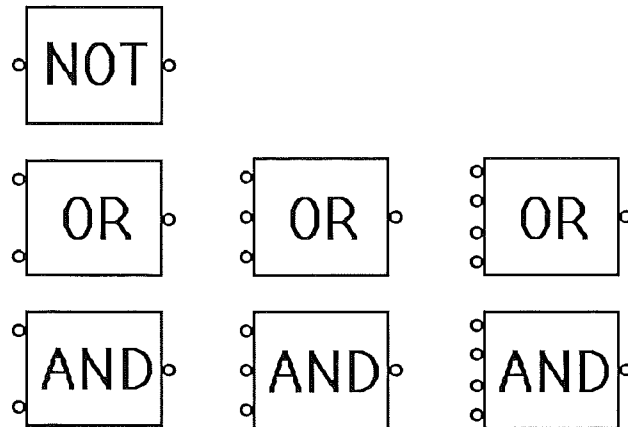
För tydlighets skull markeras in- och utsignal grafiskt under exekvering med hjälp av en vertikal respektive en horisontell linje i tillhörighetsfunktionens graf, se figur 4.3.

4.1.3 Logiska operatorer

Det finns 3 olika logiska operatorer. De används i villkorsdelen av reglerna. De har följande betydelse

$$\begin{array}{lcl} x \text{ AND } y & = & \text{Min} (x , y) \\ x \text{ OR } y & = & \text{Max} (x , y) \\ \text{NOT } x & = & 1 - x \end{array}$$

Det finns även, som figur 4.4 visar, byggblock för AND och OR med fler än två ingångar. Både in- och utsignalernas värden ligger mellan 0 och 1. Under exekvering visas utsignalens värde grafiskt genom att hela det grafiska byggblocket för operatoren blir blåfärgad. Intensiteten på den blåa färgen är proportionell mot utsignalvärdet. För värdet 0 är blocket svart och för värdet 1 är det helt blått.



Figur 4.4. Det finns tre olika logiska operatörer. Operatörerna AND och OR finns i tre versioner. Det enda som skiljer versionerna åt är antalet ingångar.

4.1.4 Regelblock

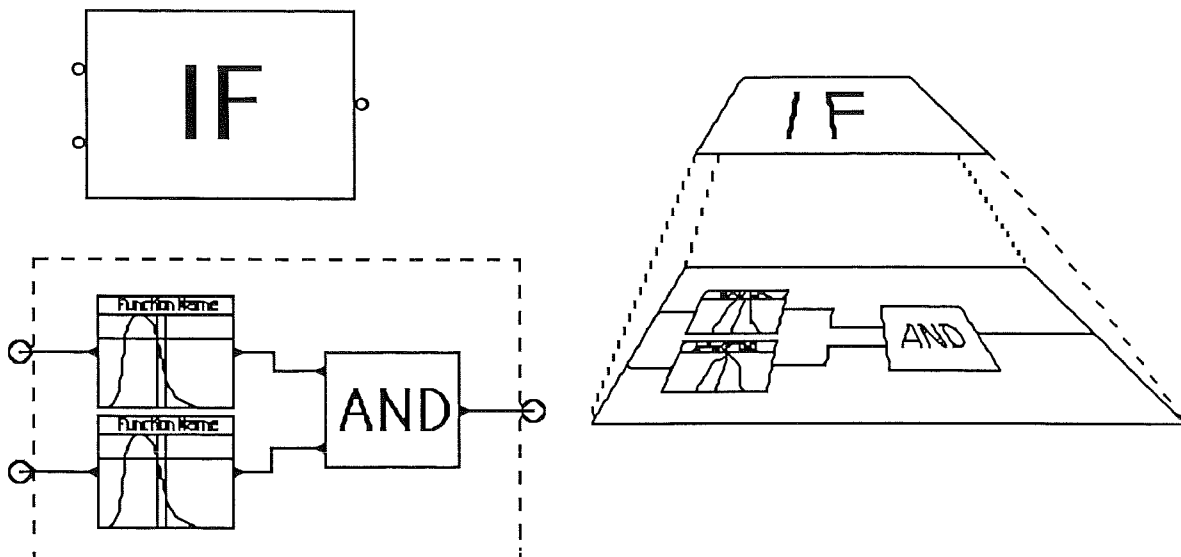
Regelblocken har utseende enligt figur 4.5 och har en enda egentlig funktion. Genom att använda dessa block separerar man nämligen tydligt reglerna från varandra. Det blir då lättare att under exekvering se vilken regel som är mest uppfylld. Om man inte zoomar upp blocket så syns endast if-boxen, se figur 4.4. Denna visar då, precis som operatörerna, utsignalens värde grafiskt med hjälp av blå färg. Intensiteten på den blå färgen är även här proportionell mot utsignalvärdet.

4.1.5 Tillhörighetsfunktioner i slutsatsdelen

De tillhörighetsfunktioner som används i slutsatsdelen av reglerna består också av 10 punkter. Funktionen kvantiseras dock internt i 21 punkter

$$[(x_1, y_1) ; (x_2, y_2) ; \dots ; (x_{21}, y_{21})]$$

och bildar 20 intervall. Detta görs för att man lättare skall kunna länka samman reglerna. Den första av de två insignalerna till tillhörighetsfunktionen, insignalnod1 i figur 4.6, är ett värde som betecknas med k och som ligger mellan 0 och 1. Detta värde motsvarar uppfyllnadsgraden för villkorsdelen av regeln och används för att skala ner tillhörighetsfunktionens kvantiserade punkter till



Figur 4.5. Ett regelblock har två olika utseende beroende på hur mycket man har zoomat upp sin regulator. Initialutseendet är "If-boxen" uppe till vänster i figuren. När man sedan zoomar upp bilden enligt skissen till höger kommer den verkliga regeln fram. Den består av två eller flera (ej i figur) tillhörighetsfunktioner kopplade till den logiska operatorn AND.

$$[(x_1, k \cdot y_1) ; (x_2, k \cdot y_2) ; \dots ; (x_{21}, k \cdot y_{21})]$$

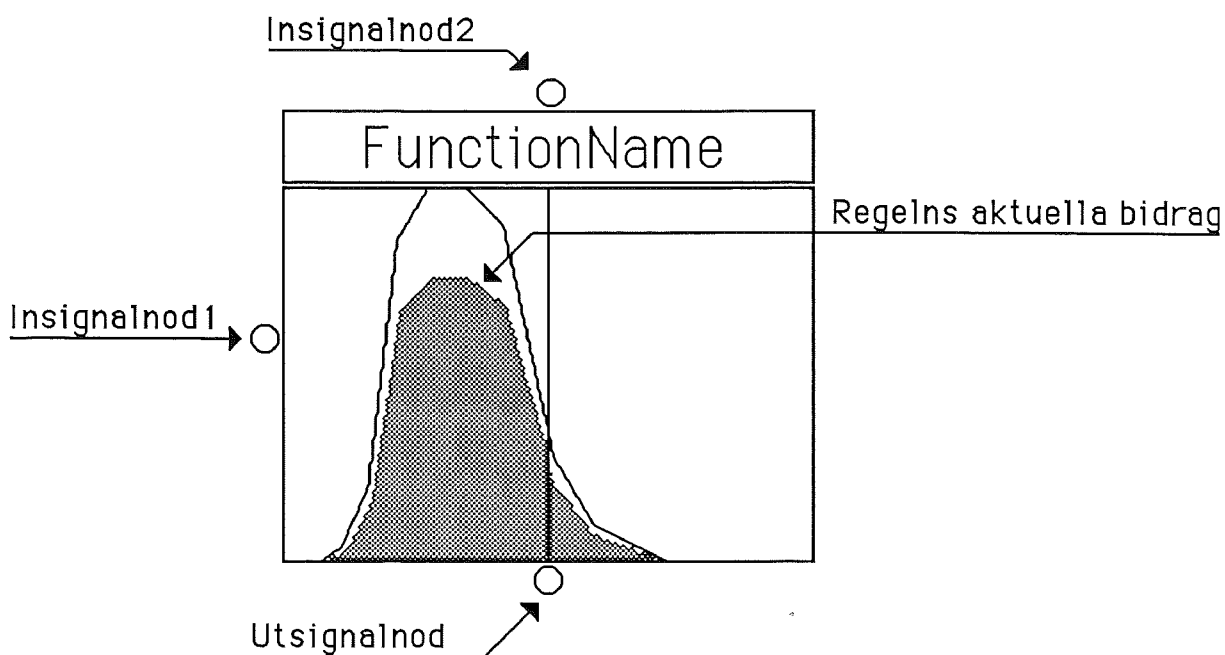
Denna nedskalning motsvarar F.L.Smidth's tolkning av den villkorliga relationen mellan två oskarpa mängder, se kapitel 3.5.

Under exekvering visas den nedskalade funktionen grafiskt som en blåfärgad yta, se figur 4.6. Den andra insignalen till tillhörighetsfunktionen består av en annan uppsättning med 21 punkter.

$$[(x'_1, y'_1) ; (x'_2, y'_2) ; \dots ; (x'_{21}, y'_{21})]$$

Dessa två olika uppsättningar med punkter har gemensamma x-värden, men givetvis olika y-värden. Som utsignal skickas de 21 punkter som bildas genom att ta maximum över båda uppsättningarna med punkter. Det betyder att utsignalen ser ut som

$$[\begin{array}{l} (x_1, \max[k \cdot y_1 ; y'_1]) ; \\ (x_2, \max[k \cdot y_2 ; y'_2]) ; \\ \dots ; \\ (x_{21}, \max[k \cdot y_{21} ; y'_{21}]) \end{array}]$$



Figur 4.6. Tillhörighetsfunktionen i slutsatsdelen har tre grafiska kopplingsnoder. Två av dessa är insignalnoder. Den skuggade ytan i figuren visar hur mycket tillhörighetsfunktionen skalats ner. Den nedskalade kurvan är den aktuella regelns bidrag i algoritmen.

På detta sätt länkas reglerna samman och de resulterande maximala 21 punkterna skickas vidare till nästa regel. Den sista regeln skickar sina resulterande 21 punkter vidare till utsignalberäknaren.

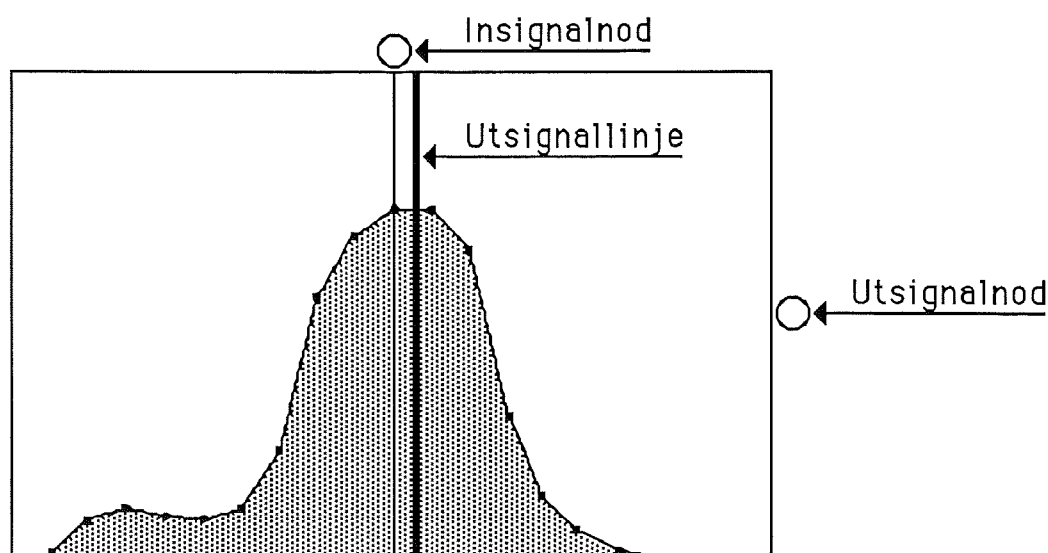
Precis som för villkorsdelens tillhörighetsfunktioner finns det en skalfaktor som anger tillhörighetsfunktionens max- och mingränser i x-led. Denna skalfaktor kommer från utsignalinterfacet och går via utsignalberäknaren in till tillhörighetsfunktionen genom utsignalnoden i figur 4.6. Detta görs för att man genom att ändra på ett enda ställe skall kunna ändra alla tillhörighetsfunktionernas max- och mingränser på en gång. Detta innebär att både insignalnod2 och utsignalnoden innehåller dels de 21 ovan diskuterade punkterna och dels skalfaktorn till tillhörighetsfunktionen. Signalflödesmässigt går de 21 punkterna framåt och skalfaktorn bakåt i regulatorstrukturen.

4.1.6 Utsignalberäknare

Det finns två olika byggblock för att beräkna utsignalen. Båda metoderna för utsignalberäkning har 21 punkter som insignal. Av dessa punkter bildas en kurva genom att använda linjär interpolering mellan punkterna. Båda metoderna använder sig av ytan under kurvan vid beräkningen, se figur 4.7. Denna yta består av 20 delytor som var och en har formen av en parallelltrapets.

Den ena metoden för utsignalberäkning ger som utsignal tyngdpunkten för kurvan. Den andra metoden delar ytan i två lika stora delytor med en vertikal linje. Som utsignal ges delningslinjens x-värde. För båda metoderna gäller det att om ingen regel är uppfylld till mer än 0,2 så ges utsignalen värdet 0.

Utsignalen markeras grafiskt under exekveringen med hjälp av en gul vertikal linje.



Figur 4.7. Utsignalberäknaren visar insignalkurvans utseende, samt utsignalens värde i form av en värdelinje.

4.1.6.1 Tyngdpunktsmetoden

Metoden ger som namnet säger tyngdpunkten för kurvan som utsignal. Tyngdpunkten beräknas lättast genom att först beräkna tyngdpunkt och yta för varje delyta under kurvan. Den totala tyngdpunkten fås sedan genom

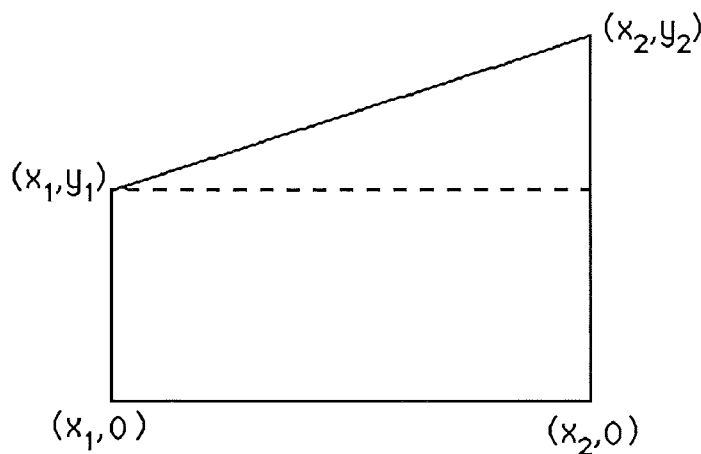
$$TP_{\text{Total}} = \sum_i (TP_i \cdot Y_{ta_i}) / \sum_i Y_{ta_i} \quad ; \quad i = 1..21$$

Eftersom varje delyta har formen av en parallelltrapets med två räta vinklar kan den delas upp i en triangel och en rektangel. Detta förenklar härledningen av beräkningsformlerna. Med beteckningar enligt figur 4.8 får man då följande uttryck

$$\begin{aligned} TP_{\text{Triangel}} &= (2x_2+x_1) / 3 \\ TP_{\text{Rektangel}} &= (x_2+x_1) / 2 \\ Yta_{\text{Triangel}} &= (x_2-x_1) \cdot (y_2-y_1) / 2 \\ Yta_{\text{Rektangel}} &= (x_2-x_1) \cdot y_1 \\ (TP \cdot Yta)_{\text{Triangel}} &= (y_2-y_1) \cdot (2x_2+x_1) \cdot (x_2-x_1) / 6 \\ (TP \cdot Yta)_{\text{Rektangel}} &= y_1 \cdot (x_2+x_1) \cdot (x_2-x_1) / 2 \end{aligned}$$

Adderar man ihop uttrycken parvis samt förenklar så erhåller man

$$\begin{aligned} Yta_{\text{Parallelltrapets}} &= (x_2-x_1) \cdot (y_1+y_2) / 2 \\ (TP \cdot Yta)_{\text{Parallelltrapets}} &= (x_2-x_1) \cdot (2x_1y_1+x_1y_2+x_2y_1+2x_2y_2) / 6 \end{aligned}$$



Figur 4.8. Beteckningar för en parallelltrapets till utsignalberäkning enligt tyngdpunktsmetoden.

Eftersom intervallet mellan x-värdena för den totala kurvan är konstant kan termen (x_2-x_1) brytas ut och förkortas bort. Det slutliga resultatet blir då

$$TP_{\text{Total}} = \sum_i (2x_1y_1+x_1y_2+x_2y_1+2x_2y_2)_i / 3 \cdot \sum_i (y_1+y_2)_i \quad ; \quad i = 1..21$$

Detta resultat kan emellertid förenklas ytterligare. Punktparet (x_2, y_2) för en delyta blir nämligen nästa delytas punktpar (x_1, y_1) . Detta betyder att alla punktpar utom det första och det sista ingår i två delytor.

Om insignalens 21 punktpar anges som

$$(x_1, y_1) ; (x_2, y_2) ; \dots ; (x_{21}, y_{21})$$

så beräknas tyngdpunkten TP som

$$TP = \frac{[x_1 \cdot (2y_1 + y_2) + x_2 \cdot (y_1 + 4y_2 + y_3) + x_3 \cdot (y_2 + 4y_3 + y_4) + \dots + x_{20} \cdot (y_{19} + 4y_{20} + y_{21}) + x_{21} \cdot (y_{20} + 2y_{21})]}{[6 \cdot (y_1/2 + y_2 + y_3 + \dots + y_{20} + y_{21}/2)]}$$

4.1.6.2 Ytdelningsmetoden

Metoden går ut på att man delar den totala ytan under kurvan i två lika stora delytor med en vertikal linje. X-värdet för denna delningslinje lämnas som utsignal.

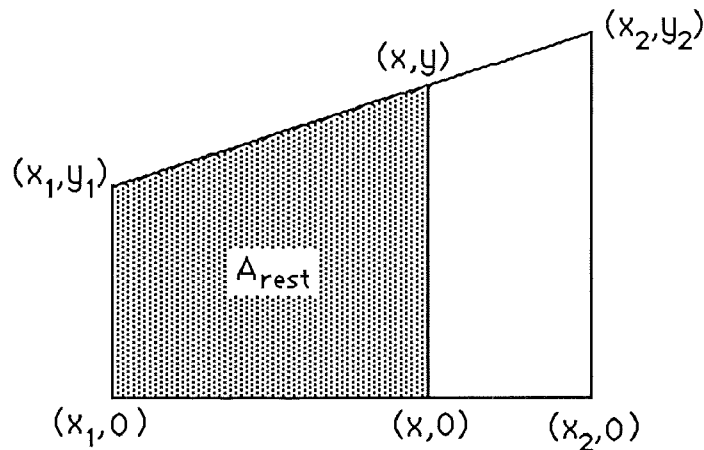
Man börjar med att beräkna den totala ytan. Sedan letar man upp den parallelltrapets som delninglinjen skall gå igenom. Därefter beräknar man linjens exakta position inuti parallelltrapetsen. En parallelltrapets med beteckningar enligt figur 4.9 har följande yta

$$Y_{\text{Parallelltrapets}} = (x_2 - x_1) \cdot (y_1 + y_2) / 2$$

Eftersom intervallet mellan x-värdena är konstant över hela den totala ytan kan parallelltrapetsytan representeras av summan av sina y-värden.

Förutom beteckningarna i figuren kommer följande beteckningar att användas i fortsättningen

- Y_{sum_i} = Summan av y-värdena för den i:te parallelltrapetsen. Detta värde är proportionellt mot ytan för parallelltrapetsen.
- A_{sum_i} = $Y_{\text{sum}_1} + Y_{\text{sum}_2} + \dots + Y_{\text{sum}_i}$.
- $A_{\text{sum}_{\text{Halv}}}$ = $A_{\text{sum}_{20}}/2$. Ett värde som är proportionellt mot halva totala ytan.



Figur 4.9. Beteckningar för en parallelltrapets till utsignalberäkning enligt ytdelningsmetoden

Den parallelltrapets där delningslinjen skall gå markerar vi med i . Då gäller följande

$$A_{rest} = (A_{sum_{Halv}} - A_{sum_{i-1}}) \cdot (x_2 - x_1) / 2$$

Där A_{rest} är det verkliga ytbidrag som behövs från den i :te parallelltrapetsen för att ha delat den totala ytan i två lika delar. Denna yta har även den formen av en parallelltrapets och beräknas enligt

$$A_{rest} = (x - x_1) \cdot (y + y_1) / 2$$

där

$$y = y_1 + (x - x_1) \cdot (y_2 - y_1) / (x_2 - x_1)$$

Om man nu sätter de båda uttrycken för A_{rest} lika med varandra och löser ut x får man

$$x = x_1 + \frac{\sqrt{[y_1^2 + 2 \cdot A_{rest} \cdot (y_2 - y_1) / (x_2 - x_1)]} - y_1 \cdot (x_2 - x_1)}{y_2 - y_1}$$

Det finns dock ett läge när denna ekvation ej kan användas. Det är när $y_2 = y_1$. Då består ytan av en rektangel och x -värdet beräknas i stället som

$$x = x_1 + A_{rest} / y_1$$

Det x -värde man nu har fått fram är det x -värde en vertikal linje skall ha för att dela den totala ytan i två lika stora delar.

4.1.7 Utsignalinterface

Det finns bara ett utsignalinterface. Detta tar fuzzyregulatorns utsignal och använder detta som inkrement till en tillståndsvariabel. Som yttre utsignal skickas denna tillståndsvariabel. Detta gör att fuzzyregulatorn blir implementerad på inkrementell form.

I signalsnittet in mot regulatorn ingår i utsignalinterfacet förutom signalvärdet även en skalfaktor till tillhörighetsfunktionerna. Denna skalfaktor bestämmer max- och mingränserna för slutsatsdelens tillhörighetsfunktioner i x-led.

4.1.8 Presentatörer

Det finns tre olika sorters presentatörer, vilkas utseende visas i bilaga 2

- Stapeldiagram
- Historikdiagram
- xy-diagram

Både stapeldiagrammet och historikdiagrammet presenterar tre olika mätvärden. Dessa är börvärdet, processens mätvärde samt styrsignalen. Skalningen av axlarna i dessa två diagram kan lätt ändras under exekvering.

I xy-diagrammet visas reglerfelet och dess förstadifferens i form av en kurva. Kurvan består av 30 punkter. Dessa punkters värden motsvarar de 30 senast samplade värdena på felet e och dess förstadifferens Δe .

Utöver dessa presentatörer kan man få en viss uppfattning om processens dynamik via regulatorkonfigurationen. Detta genom att graden av uppfyllnad för reglerna visas grafiskt under exekveringen.

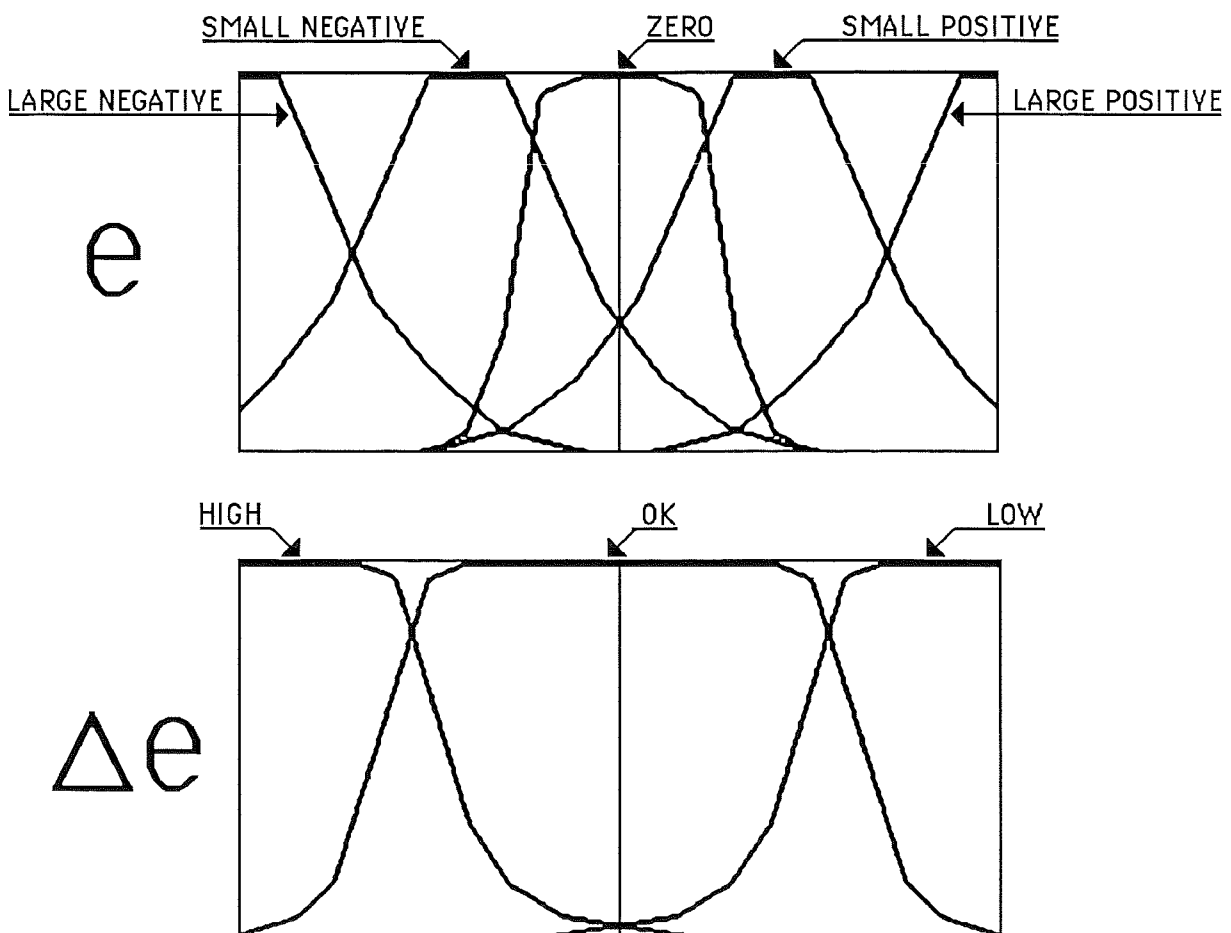
4.2 Konstruktion av en fuzzyregulator

Konstruktionen av en fuzzyregulator görs i två steg. Det första steget består i att bestämma vilka regler man vill ha med. När man väl gjort det kan man börja med steg nummer två. Detta steg består av att bygga själva regulatorn. Detta görs med hjälp av de färdiggjorda byggblocken. Till sist är det "bara" att testa och eventuellt justera regulatorn.

4.2.1 Regelval

Det svåraste momentet när man konstruerar en fuzzyregulator är att komma fram till vilka regler man skall använda. Det från början tänkta sättet med att översätta den mänskliga operatörens tumregler och använda dessa i fuzzyregulatorn fungerar bara om det finns en väl insatt operatör. Det tillvägagångssätt som använts här är helt annorlunda. Om det är möjligt bör man kombinera båda metoderna. Då får man en bra kontroll på att reglerna täcker alla driftsfall.

För enkelhets skull går vi igenom ett exempel där vi bara använder två insignaler, nämligen reglerfelet e och dess förstaderivens Δe . För varje insignal vill man i villkorsledet ha en uppsättning tillhörighetsfunktioner som överlappar varandra, se figur 4.10.



Figur 4.10. När man väljer sina regler skall man se till att villkorsdelens tillhörighetsfunktioner överlappar varandra. I figuren visas ett exempel på hur tillhörighetsfunktionerna kan väljas för de två insignalerna e och Δe .

När man väl är nöjd med detta skapar man en tabell av dessa tillhörighetsfunktioner. För fallet med två insignaler gör man en tabell i matrisform. I tabellen fyller man i slutsatsdelens tillhörighetsfunktioner för varje regel, se figur 4.11. När denna tabell är ifylld kan man börja bygga sin regulator.

		Δe		
		LOW	OK	HIGH
e	LNEG	LNEG	MNEG	—
	SNEG	MNEG	ZNEG	ZPOS
	ZERO	SNEG	ZERO	SPOS
	SPOS	ZNEG	ZPOS	MPOS
	LPOS	—	MPOS	LPOS

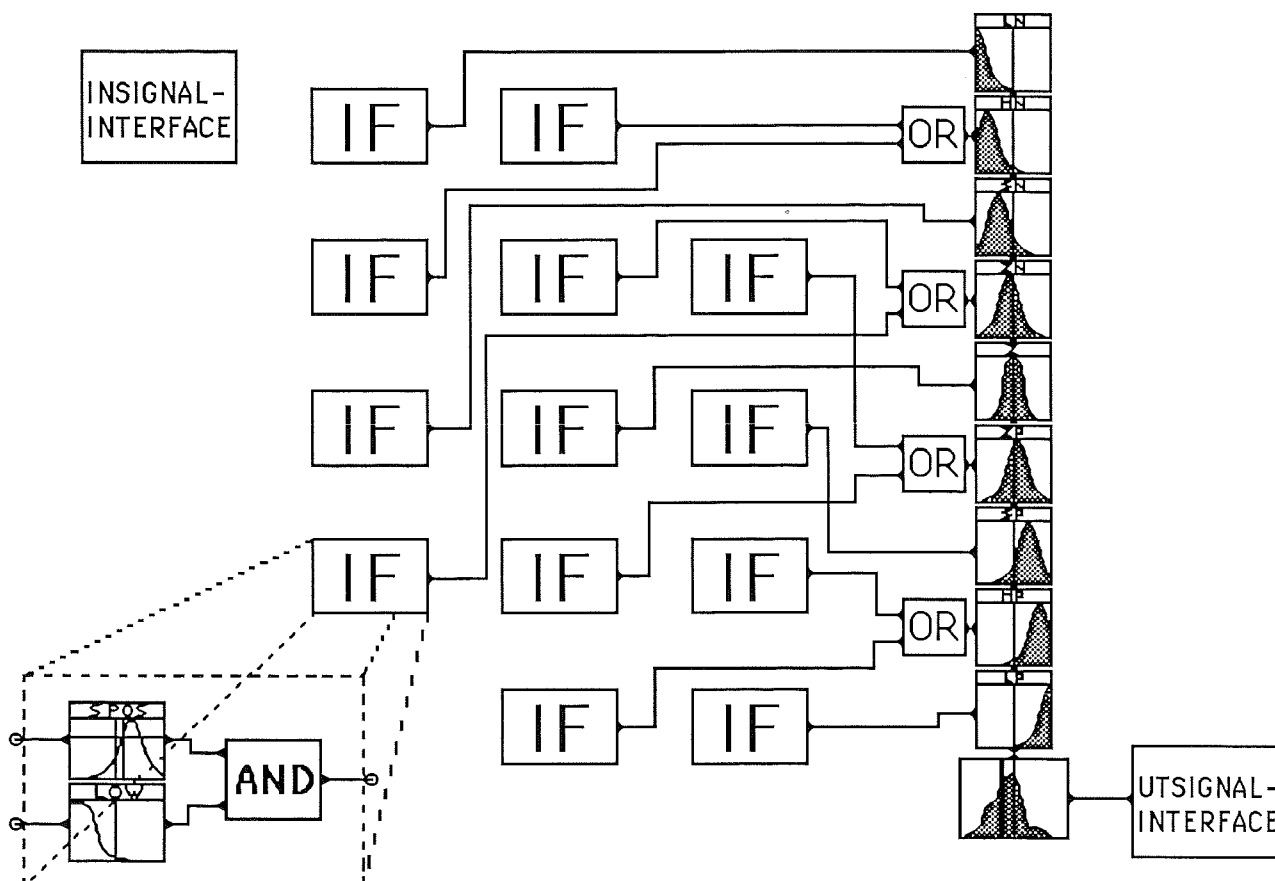
LNEG = Large Negative
 MNEG = Medium Negative
 SNEG = Small Negative
 ZNEG = Zero Negative
 ZPOS = Zero Positive
 SPOS = Small Positive
 MPOS = Medium Positive
 LPOS = Large Positive

Figur 4.11. När man väl valt sina tillhörighetsfunktioner i villkorsdelen av reglerna ställer man upp en tabell för dem. Sedan fyller man i slutsatsdelens tillhörighetsfunktioner. I figuren visas ett exempel på hur en tabell kan se ut för en regulator med två insignaler

4.2.2 Sammankoppling av byggblocken

Det första man gör är att plocka fram en regulatormall för en fuzzyregulator. Denna innehåller de presentatörer och interfaceblock som behövs. Nästa steg är att sätta ihop reglerna inuti regulatorn. Ur biblioteket plockar man fram ett regelblock vilket man sedan duplicerar tills rätt antal block finns tillgängliga. När man sedan placerar ut regelblocken är det viktigt att placera dem så att man lätt kan identifiera sina regler. I fallet med två insignaler kan man lämpligen behålla tabellstrukturen från figur 4.11. Därefter måste man gå in i varje regelblock och precisera reglerna. Detta gör man genom att se till att rätt tillhörighetsfunktioner finns inuti blocken. När alla regelblocken är utplacerade och preciserade plockar man fram slutsatsdelens tillhörighetsfunktioner samt utsignalberäknaren. Eftersom signalflödet i blocken antingen går från vänster till höger eller uppifrån och ner placeras lämpligen byggblocken enligt figur 4.12. Slutligen kopplar man ihop byggblocken via de grafiska kopplingarna.

När byggblocken är sammankopplade är regulatoren färdig att testas. Det finns ett par möjligheter att justera regulatoren under exekvering. Man kan som tidigare nämnts ändra utseendet på tillhörighetsfunktionerna. Man kan även ändra skalan i x-led för dessa. För tillhörighetsfunktionerna i villkorsdelen ändras skalan i x-led via insignalinterfacet och för tillhörighetsfunktionerna i slutsatsdelen ändras den via utsignalinterfacet.



Figur 4.12. Figuren visar hur byggblocken till en fuzzyregulator med regler enligt exemplet från kapitel 4.2.1 bör placeras ut. Regelblocken är placerade i en tabellstruktur efter tabellen i figur 4.11. Eftersom signalflödet antingen går från vänster till höger eller uppifrån och ner bör strukturen användas även för andra fuzzyregulatorer.

5 SIMULERINGAR

Fuzzyregulatorn har testats genom att försöka reglera två olika simulerade processer. Den första processmodellen är en gräddseparator där fetthalten skall regleras. Den andra processmodellen är en olinjär vattentank där vätskenivån skall regleras.

5.1 Gräddseparator

Separatortar emot råmjölk och separerar bort fett från denna. Den lämnar ifrån sig dels skummjölk med konstant fetthalt och dels grädde med varierbar fetthalt. Det som skall regleras hos separatort är fetthalten hos grädden. Detta görs genom att styra utflödet från separatort enligt figur 5.1. När flödet på grädden ökar så spås grädden ut med mer mjölk och fetthalten minskar. Processen regleras med en fuzzyregulator i kaskad med en PID-regulator. Fuzzy-regulatorn lämnar som utsignal ett börvärde på flödet till PID-regulatorn, som i sin tur styr detta flöde.

Med följande beteckningar

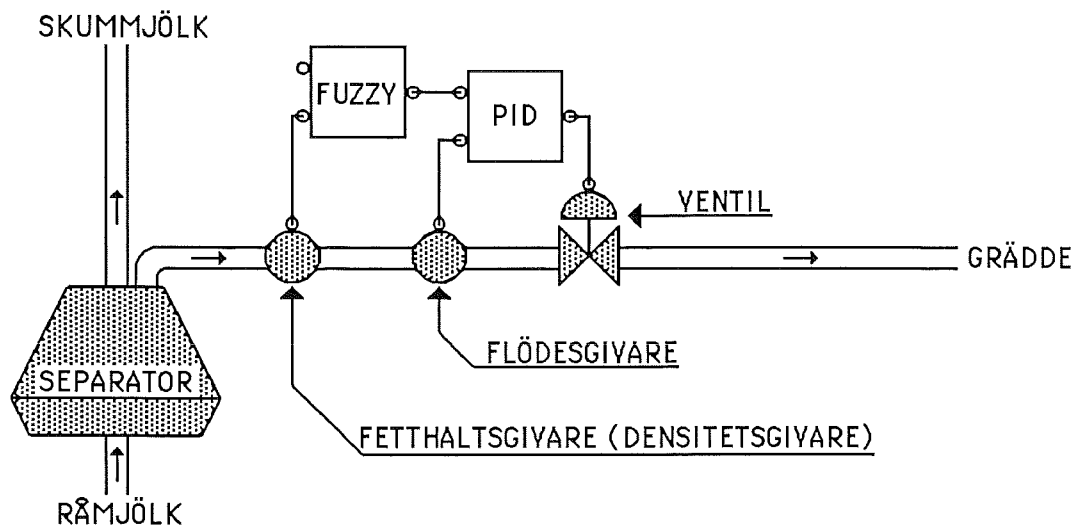
Fett _t	=	Gräddens fetthalt vid tiden t
Flöde _t	=	Gräddflödet vid tiden t
Råfett	=	Fetthalten på råmjölken
Råflöde	=	Flödet på råmjölken
Ventilvärde _t	=	Utsignalen från PID-regulatorn

kan separatort beskrivas matematiskt på dess diskreta tillståndsform enligt

$$\begin{aligned} \text{Fett}_{t+1} &= 0,9 \cdot \text{Fett}_t + 0,1 \cdot \text{Råfett} \cdot \text{Råflöde} / \text{Flöde}_t \\ \text{Flöde}_{t+1} &= 0,5 \cdot \text{Flöde}_t + 0,5 \cdot \text{Ventilvärde}_t \end{aligned}$$

Båda givarna implementerade så att de ger en tidsfördröjning på ett sampelintervall. Detta innebär att värdena på fetthalt och flöde för grädden vid tiden t finns tillgängliga för regulatorerna vid tiden $t+1$. Med hänsyn tagen till processens dynamik innebär detta att ett styrgrepp vid tiden t kan märkas i värdet från flödesgivaren vid tiden $t+2$ och i värdet från fetthaltsgivaren vid tiden $t+3$.

Samtliga signalvärden till och från processen är angivna i procent och ligger i intervallet 0-100%.



Figur 5.1. Schema över gräddseparatorn. Det finns två givare. Den ena mäter flödet och den andra mäter densiteten på grädden. Fetthalten fås genom att värdet från densitetsgivaren är omräknat till fetthalt redan i givaren.

5.1.1 Fuzzyregulatorn

Fuzzyregulatorn som styr fetthalten använder sig av två yttre insignaler, processens mätvärde och börvärdet. Av dessa bildas reglerfelet e och dess förstadifferens Δe . Regulatorn ger som utsignal en styrsignal implementerad på inkrementell form.

Max- och mingränserna för tillhörighetsfunktionerna i x-led har valts till

e	: felet	: [-20 , 20]
Δe	: felets förstadifferens	: [-2,5 , 2,5]
Δu	: styrsignalens inkrement	: [-2 , 2]

För båda insignalerna används i villkorsdelen av reglerna fem överlappande tillhörighetsfunktioner. Dessa är "Large negative", "Small negative", "Zero", "Small positive" samt "Large positive". Insignalernas tillhörighetsfunktioner har kombinerats ihop till 21 regler. Dessa regler finns presenterade i tabellform i figur 5.2. Utifrån dessa regler har sedan själva fuzzyregulatorn byggt upp. Den grafiska presentationen av regulatorn visas i bilaga 3.

Fuzzyregulatorn är kopplad i kaskad med en PID-regulator. Denna PID-regulator styr själva flödet hos grädden och får sitt börvärde från fuzzyregulatorn. PID-regulatorns parametrar är

$$\begin{aligned} K &= 0,25 \\ T_i &= 1 \\ T_d &= 0 \end{aligned}$$

Simuleringsresultaten från denna regulatorkombination visas även den i bilaga 3. Som jämförelse har motsvarande simuleringar gjorts med två kaskadkopplade PID-regulatorer. Deras parametrar är

$$\begin{array}{ll} \text{Inre loopen: } K = 0,25 & \text{Yttre loopen: } K = -0.5 \\ T_i = 1 & T_i = 10 \\ T_d = 0 & T_d = 0 \end{array}$$

Samtliga parametervärden till PID-regulatorerna är väl genomtestade sen tidigare simuleringar.

Δe

	LNEG	SNEG	ZERO	SPOS	LPOS
LNEG	LPOS	MPOS	SPOS	—	—
SNEG	MPOS	SPOS	ZPOS	ZERO	ZNEG
e ZERO	SPOS	ZPOS	ZERO	ZNEG	SNEG
SPOS	ZPOS	ZERO	ZNEG	SNEG	MNEG
LPOS	—	—	SNEG	MNEG	LNEG

Figur 5.2. De regler som används vid reglering av gräddseparatorn ges i figuren på tabellform. Beteckningarna på tillhörighetsfunktionerna är desamma som i figur 4.11.

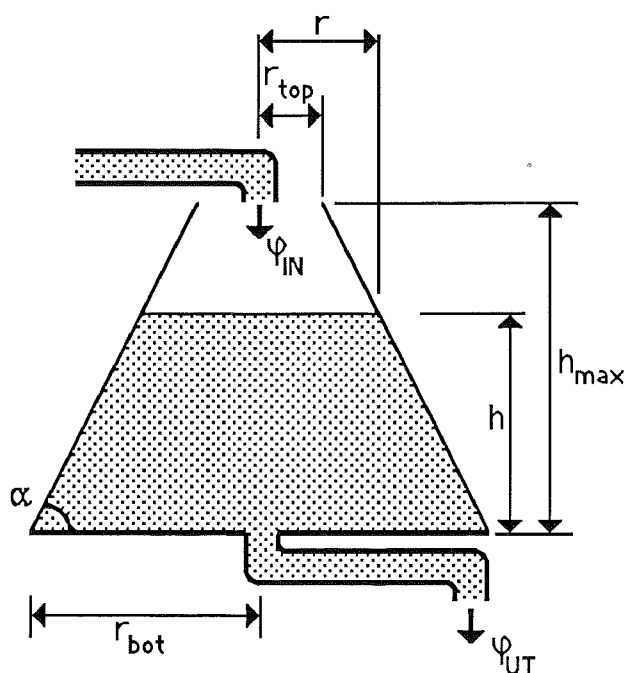
5.1.2 Resultatet

Simuleringsresultatet i bilaga 3 visar att fuzzyregulatorn klarar av att reglera fetthalten ungefär lika bra som en fix PID-regulator. Då fetthalten skall hållas hög har både fuzzyregulatorn och PID-regulatorn svårigheter att hålla fetthalten konstant. Detta beror på att olinjäriteten hos processen förhåller sig som 1/flödet samt att processen innehåller en viss tidsfördröjning. Med detta i åtanke skulle fuzzyregulatorn kunna förbättras genom att använda även flödet som insignal.

Vid en jämförelse mellan fuzzyregulatorn och PID-regulatorn kan man konstatera att PID-regulatorn oftast är en aning snabbare än fuzzyregulatorn vid stegförändringar i börvärdet.

5.2 Vattentank

Vätskenivån i tanken skall regleras genom att styra inflödet till tanken. Tanken har fritt utlopp vilket ger den en viss olinjäritet. Denna olinjäritet har ökat genom att tanken har formen av en stympad kon. Utseendet på tanken kan styras med tre parametrar. Dessa är tankens höjd, tankens bottenradie samt cosinus för vinkeln mellan tankens botten och dess sida, se figur 5.3.



Figur 5.3 Formen på tanken kan styras med parametrarna h_{max} , r_{bot} , $\cos \alpha$. Tanken har fritt utlopp och vätskenivån regleras genom att styra inflödet.

Tanken är implementerad på diskret form. Volymförändringen antas ha ett cylindriskt utseende. Med beteckningar enligt figur 5.3 ges då volymförändringen enligt

$$\Delta V = \Delta h \cdot \pi \cdot r^2$$

Detta är en rimlig approximation så länge Δh är liten. Tankens dynamik kan då anges på sin tillståndsform enligt

$$h_{t+1} = h_t + (\varphi_{in_t} - \varphi_{ut_t}) \cdot T_{samp} / (\pi \cdot r_t^2)$$

där

$$\varphi_{\text{ut}} = \text{utflödet ur tanken} = a \cdot \sqrt{h_t}$$

$$r_t = \text{tankens radie vid vätskenivån} = (r_{\text{bot}} \cdot \sin \alpha - h_t \cdot \cos \alpha) / \sin \alpha$$

I simuleringarna har följande konstanter använts

$$a = 8 \qquad h_{\text{max}} = 30$$

$$r_{\text{bot}} = 25 \qquad \cos \alpha = 0,55$$

5.2.1 Fuzzyregulatorn

Fuzzyregulatorn som reglerar vätskenivån använder sig av två yttre insignaler. Dessa är processens mätvärde h och börvärdet. Av dessa bildas tre signaler, reglerfelet e , dess förstadifferens Δe samt vätskenivån h . Regulatorn ger som utsignal en styrsignal implementerad på inkrementell form.

Max- och mingränserna i x-led för tillhörighetsfunktionerna har valts till

$$e : \text{felet} \qquad \qquad \qquad : [-5, 5]$$

$$\Delta e : \text{felets förstadifferens} \qquad : [-0,15, 0,15]$$

$$h : \text{vätskenivån} \qquad \qquad \qquad : [0, 30]$$

$$\Delta u : \text{styrsignalens inkrement} \qquad : [-20, 20]$$

För vätskenivån används tre tillhörighetsfunktioner för att bilda två överlappande intervall, "Low" och "Medium OR High". För felet används fem överlappande tillhörighetsfunktioner. Dessa är "Large negative", "Small negative", "Zero", "Small positive" samt "Large positive". För felets förstadifferens används sju överlappande tillhörighetsfunktioner. Dessa är "Large negative", "Small negative", "Zero negative", "Zero", "Zero positive", "Small positive" samt "Large positive".

I tankens nedre del är tidskonstanten längre än i den övre. Därför har tillhörighetsfunktionerna för felets förstadifferens en känsligare uppdelning för tankens nedre del, se figur 5.4. Med känsligare uppdelning menas i det här fallet att de ligger närmare varandra och överlappar varandra mer. Dessutom ligger deras tyngdpunkt närmare värdet noll.

h : HIGH
or
MEDIUM

Δe

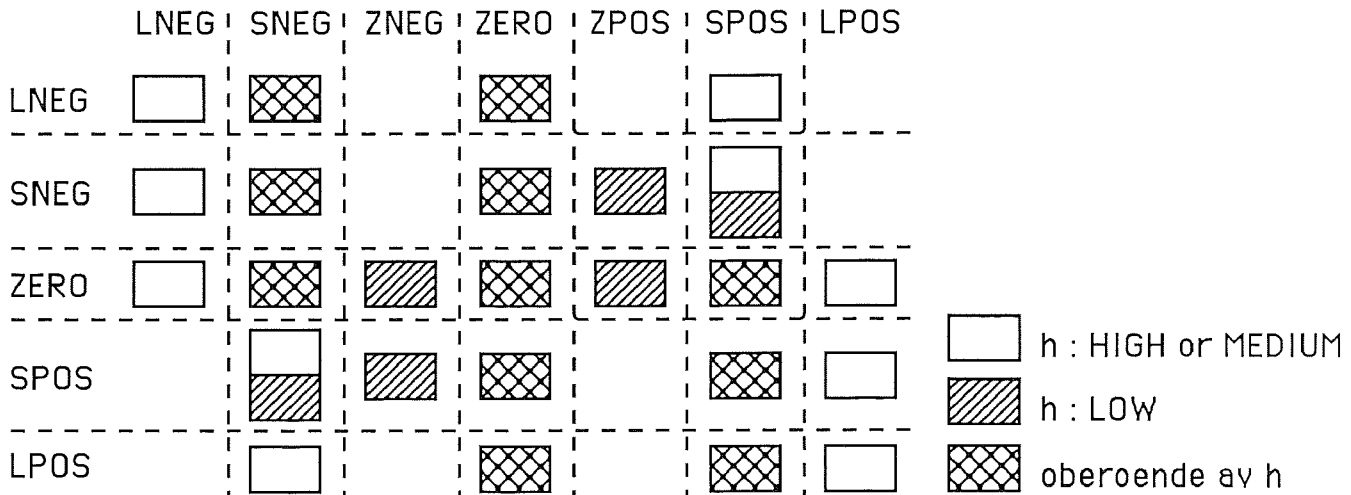
	LNEG	SNEG	ZNEG	ZERO	ZPOS	SPOS	LPOS
LNEG	LNEG	LNEG*	—	MNEG*	—	SNEG	—
SNEG	LNEG	MNEG*	—	SNEG*	—	ZERO	—
ZERO	MNEG	SNEG*	—	ZERO*	—	SPOS*	MPOS
SPOS	—	ZERO	—	SPOS*	—	MPOS*	LPOS
LPOS	—	SPOS	—	MPOS*	—	LPOS*	LPOS

h : LOW

	LNEG	SNEG	ZNEG	ZERO	ZPOS	SPOS	LPOS
LNEG	—	LNEG*	—	MNEG*	—	—	—
SNEG	—	MNEG*	—	SNEG*	ZPOS	ZPOS	—
ZERO	—	SNEG*	ZNEG	ZERO*	ZPOS	SPOS*	—
SPOS	—	ZNEG	ZNEG	SPOS*	—	MPOS*	—
LPOS	—	—	—	MPOS*	—	LPOS*	—

* Regler som är lika i båda tabellerna.

Figur 5.4. Figuren visar reglerna för fuzzyregulatorn. Det finns 27 olika regler. De regler som inte är beroende av vätskenivån h har skrivits in i båda tabellerna och markerats med en asterisk.



Figur 5.5. Figuren visar regelblockens inbördes placering inuti regulatorn

Insignalernas tillhörighetsfunktioner har kombinerats ihop till 27 olika regler. Dessa visas i tabellform i figur 5.4. Samtliga regler har placerats i separata regelblock. Deras inbördes placering inuti regulatorn visas i figur 5.5. Själva regulatorn visas i bilaga 4 där även simuleringsresultaten visas. Som jämförelse har motsvarande simuleringar även gjorts med en PID-regulator.

PID-regulatorn har ställts in med hjälp av en automatinställare för nivån 8 cm. Dess parametrar är

$$\begin{array}{lcl} K & = & 40 \\ T_i & = & 47 \\ T_d & = & 0 \end{array}$$

5.2.2 Resultatet

Simuleringsresultatet från de fyra simuleringarna visar att fuzzyregulatorn generellt är bättre än en fix PID-regulator. I den nedre delen av tanken ger PID-regulatorn kraftiga överslängar i stegsvaret medan fuzzyregulatorn lägger sig på rätt nivå direkt. Generellt för hela tanken är att fuzzyregulatorn har ett snabbare insvängningsförlopp än PID-regulatorn.

Fuzzyregulatorn har även jämförts med en parameterstyrd PID-regulator. Parameterstyrningen gjordes med höjden som styrvariabel och parametrarna ställdes in med hjälp av en automatinställare för fyra olika vätskenivåer. Resultatet av denna jämförelse visar att den parameterstyrda PID-regulatorn är snabbare i den övre delen av tanken medan de båda regulatorerna är ungefär likvärdiga i den undre delen. Fuzzyregulatorn skulle kunna förbättras för den övre delen av tanken genom att använda en finare uppdelning av villkorsdelens tillhörighetsfunktioner för höjden. I så fall skulle fuzzyregulatorn förmodligen prestera lika bra resultat som den parameterstyrda PID-regulatorn. Emellertid skulle denna förbättring kräva en hel del arbete och kommer därför inte att göras.

6 SAMMANFATTNING OCH KOMMENTARER

Tidigare resultat visar att fuzzyregulatorer uppför sig väl så länge reglerna är väl valda. Den största fördelen med en fuzzyregulator är flexibiliteten med att direkt kunna översätta mänsklig operatörkunskap till regler i regulatorn. En annan fördel är dess enkla struktur som inte kräver några större kunskaper inom reglerteknik. Dess nackdelar är dels svårigheterna med att analysera den färdiga regulatorn och dels svårigheterna med att definiera tillhörighetsfunktionerna och välja regler till regulatorn.

I jämförelse med en fix PID-regulator kan man nämna att en fuzzyregulator är betydligt jobbigare att ställa in än en PID-regulator, men när fuzzyregulatorn väl är inställd kan den ofta ge bättre resultat än PID-regulatorn. Av detta kan man dra slutsatsen att då det finns en existerande PID-regulator som fungerar bra för den aktuella processen finns det ingen anledning att lägga ner en massa jobb på att konstruera en fuzzyregulator för att reglera processen. De fall där det lönar sig att satsa på en fuzzyregulator är processer där vanliga metoder har misslyckats. Typiska processer av det slaget är sådana som har många mätsignaler som på något sätt är kopplade till varandra.

Den implementerade fuzzyregulatorn uppför sig väl vid reglering av de två simulerade processerna trots att processerna egentligen inte är av den typ man normalt försöker reglera med fuzzyregulatorer. Processerna kan nämligen regleras med normala PID-regulatorer på ett klart godkänt sätt.

Fuzzyregulatorn har inte testats på någon verklig process då den kräver ganska långt samplingsintervall. Det kortaste samplingsintervallet som mätts upp är ca 0,5 sek.

Då tiden för ett examensarbete är tämligen kort saknas en del detaljer för att den skall kunna användas direkt i systemet. Som exempel kan nämnas att den ej är helt anpassad för att användas som inre regulator i kaskadkoppling med andra regulatorer.

Fuzzyregulatorn består i sitt nuvarande läge av en enkel struktur och medger inte så stora valmöjligheter vid konstruktionen. Till exempel skulle följande saker kunna implementeras

Utsignalinterface där man kan välja om styrsignalen för regulatorn skall vara på inkrementell form eller inte.

För processer med stora tidsfördröjningar skulle man kunna införa begränsningar på inkrementet till styrsignalen då regulatorn vill göra ett flertal förändringar åt samma håll.

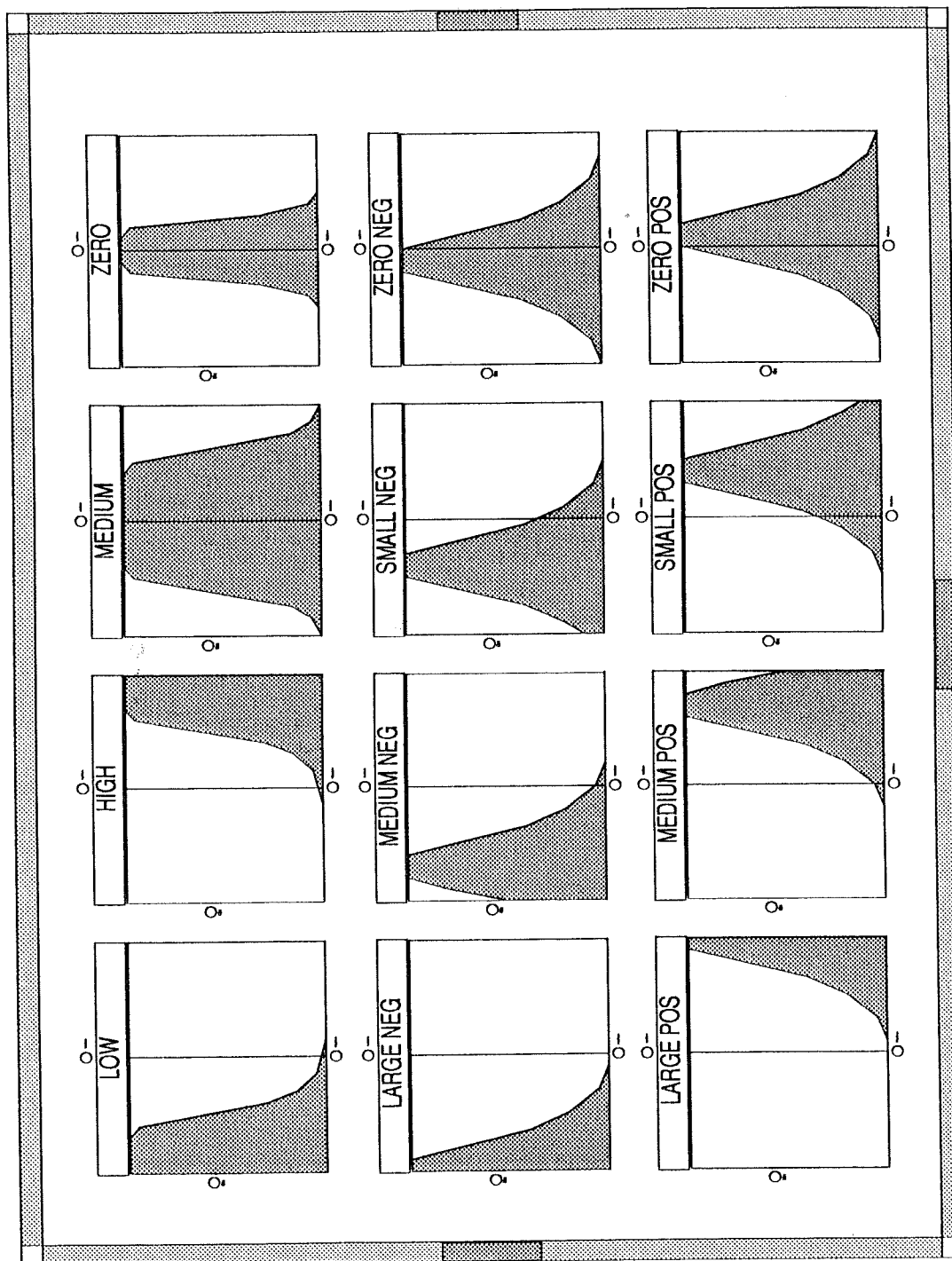
Tillståndsvariabler för processen skulle kunna användas. Till exempel skulle man kunna göra en finare uppdelning av villkorsdelens tillhörighetsfunktioner då processen närmar sig sitt stationära läge.

7 REFERENSER

1. L. A. Zadeh : *Fuzzy Sets* . Information and Control, vol. 8, sid. 338-353, 1965
2. L. A. Zadeh : *Outline of a New Approach to the Analysis of Complex Sysytems and Decision Processes* . IEEE Transactions on Systems, Man and Cybernetics, vol. SMC-3, No. 1, sid. 28-44, 1973
3. E. H. Mamdani, S. Assilian : *A Fuzzy Logic Controller for a Dynamic Plant* . International Journal of Man-Machine Studies, vol. 7, sid. 1-13, 1975
4. W. J. M. Kickert; H. R. Van Nauta Lemke : *Application of a Fuzzy Controller in a Warm Water Plant* . Automatica, vol 12, sid. 301-308, 1976
5. T. J. Procyk; E. H. Mamdani : *A Linguistic Self-Organizing Process Controller* . Automatica , vol 15, sid 15-30, 1979
6. L. P. Holmblad; J. J. Ostergaard : *Control of a Cement Kiln by Fuzzy Logic* . Fuzzy Information and Decision Processes, M. M. Gupta and E. Sanches eds. North Holland, Amsterdam, 1982
7. M. Mizumoto : *Fuzzy Controls Under Various Fuzzy Reasoning Methods* . Information Sciences, vol. 45, No. 2, sid. 129-151, Juli 1988
8. M. Braae; D. A. Rutherford : *Fuzzy Relations in a Control Setting*. Kybernetes, vol. 7(3), sid 185-188, 1978

BILAGA 1

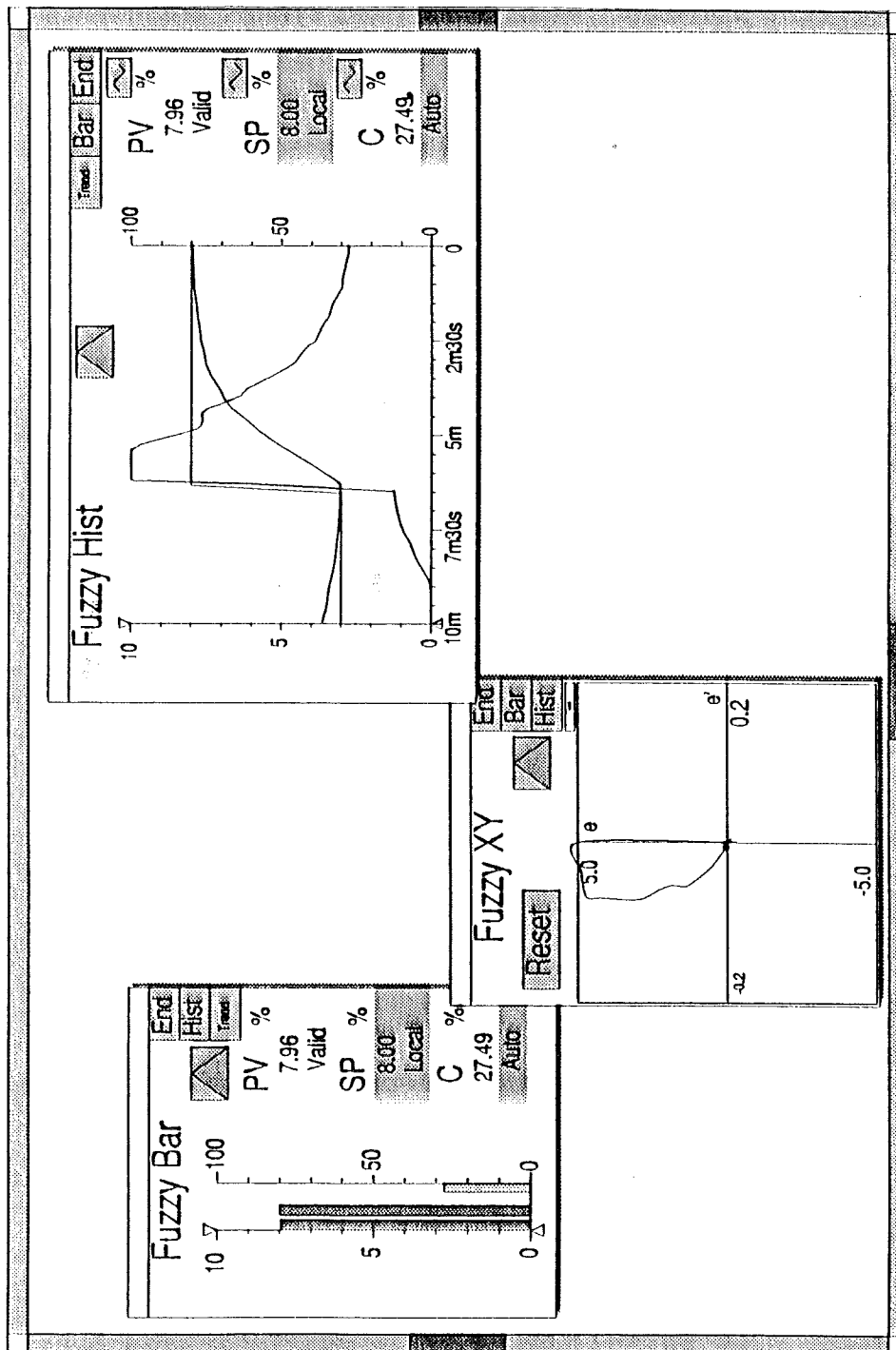
Det finns 12 fördefinierade tillhörighetsfunktioner. I denna bilaga visas deras initialutseende. Detta kan ändras under exekvering. Det finns två uppsättningar tillhörighetsfunktioner. De som visas här är den uppsättning som används i slutsatsdelen av reglerna.



BILAGA 2

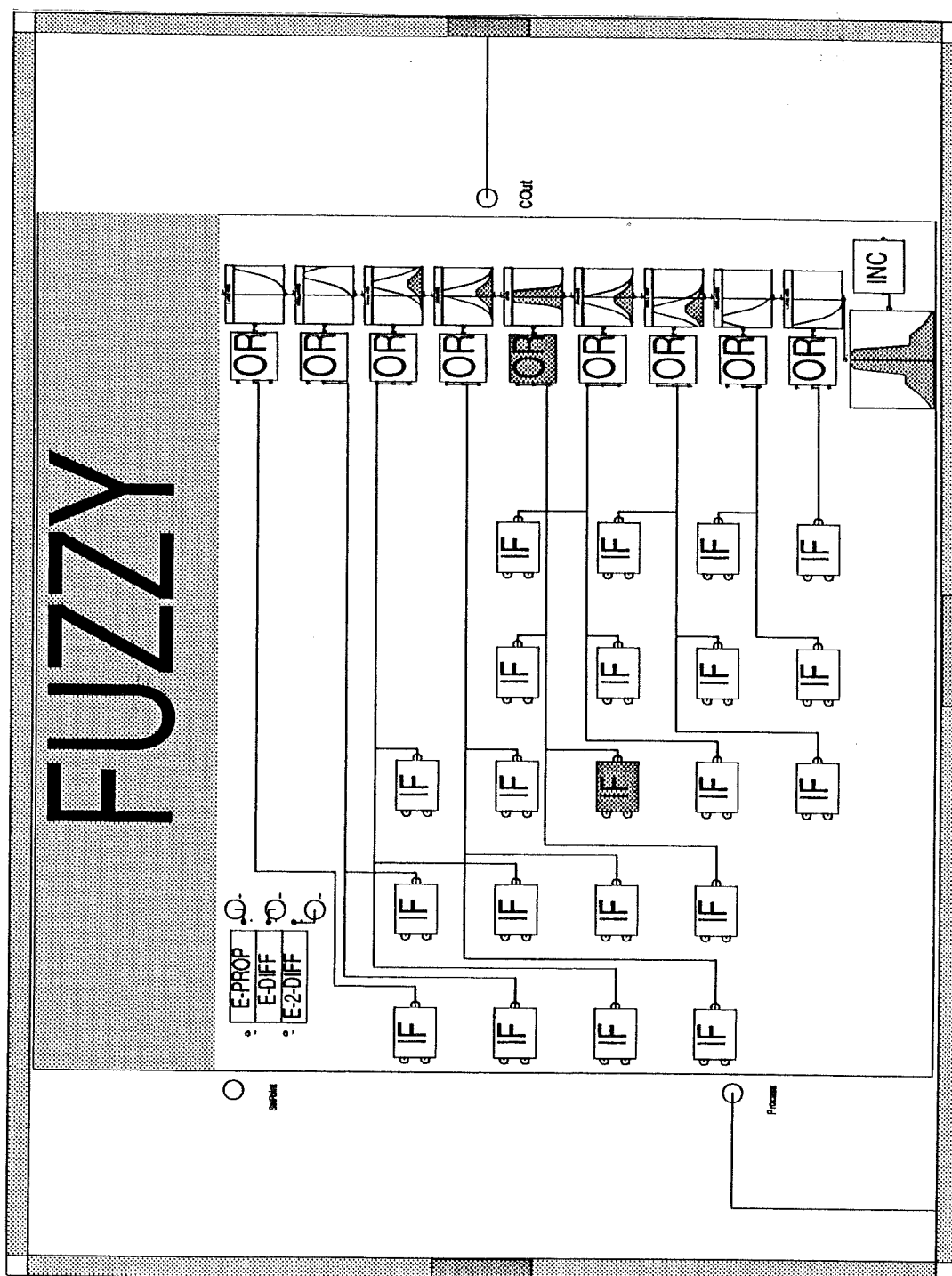
Det finns tre olika sorters presentatörer för fuzzyregulatorn. Dessa är Stapeldiagram (Fuzzy Bar), Historikdiagram (Fuzzy Hist) samt xy-diagram (Fuzzy XY). I xy-diagrammet visas hur felet e och dess förstadifferens e' förändras under exekveringens gång. I stapeldiagrammet och historikdiagrammet visas tre signalvärden

- SP = börvärdet (eng. Setpoint)
- PV = processvärde (eng. Processvalue)
- C = styrsignalen (eng. Controlsignal)



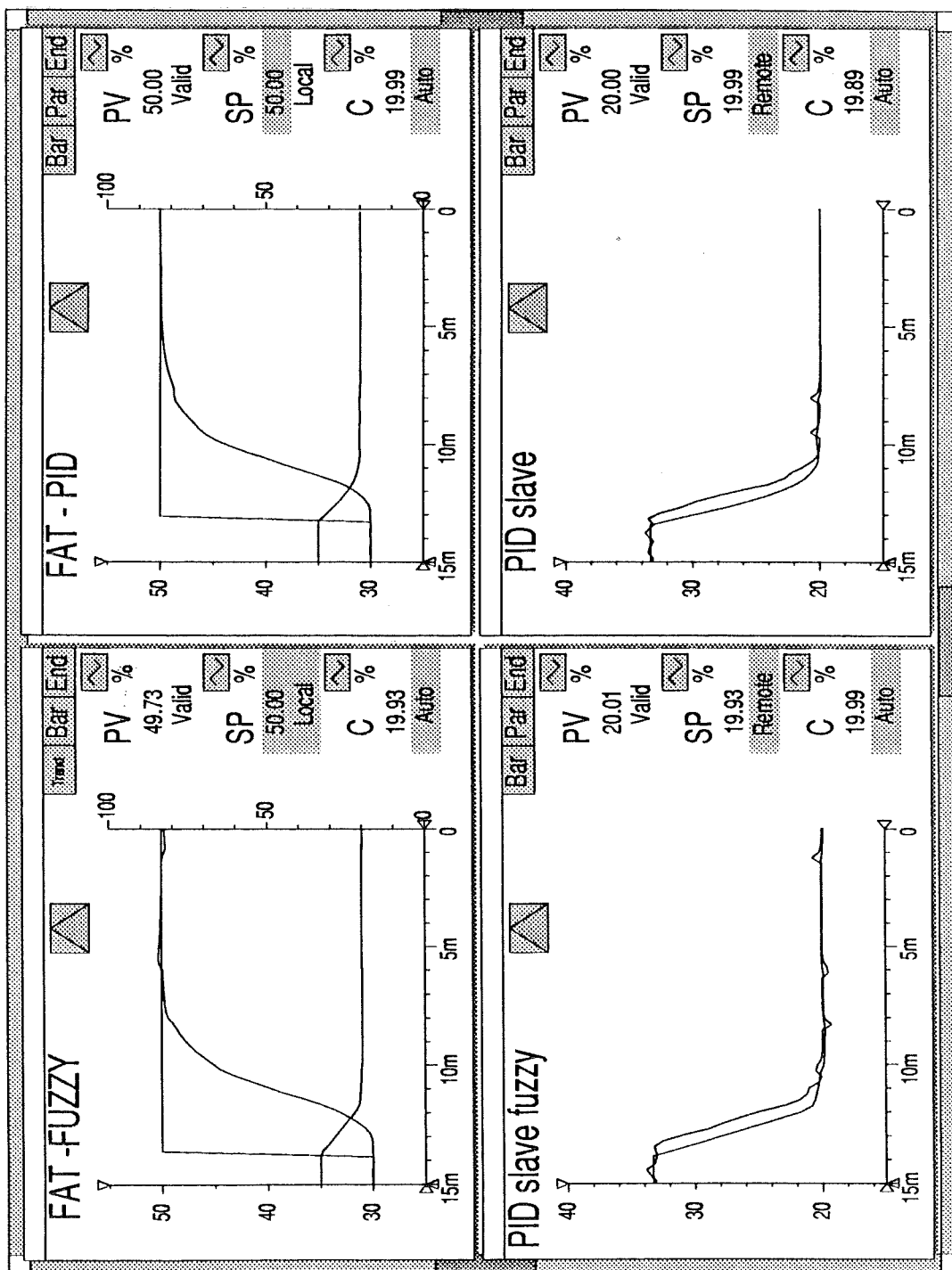
BILAGA 3

Bilaga 3 innehåller fuzzyregulatorn till gräddseparatorn samt simuleringsresultat från simuleringar dels med fuzzyregulatorn i kaskad med en PID-regulator och dels med två kaskadkopplade PID-regulatorer. Här nedan visas den grafiska representationen av själva fuzzyregulatorn.



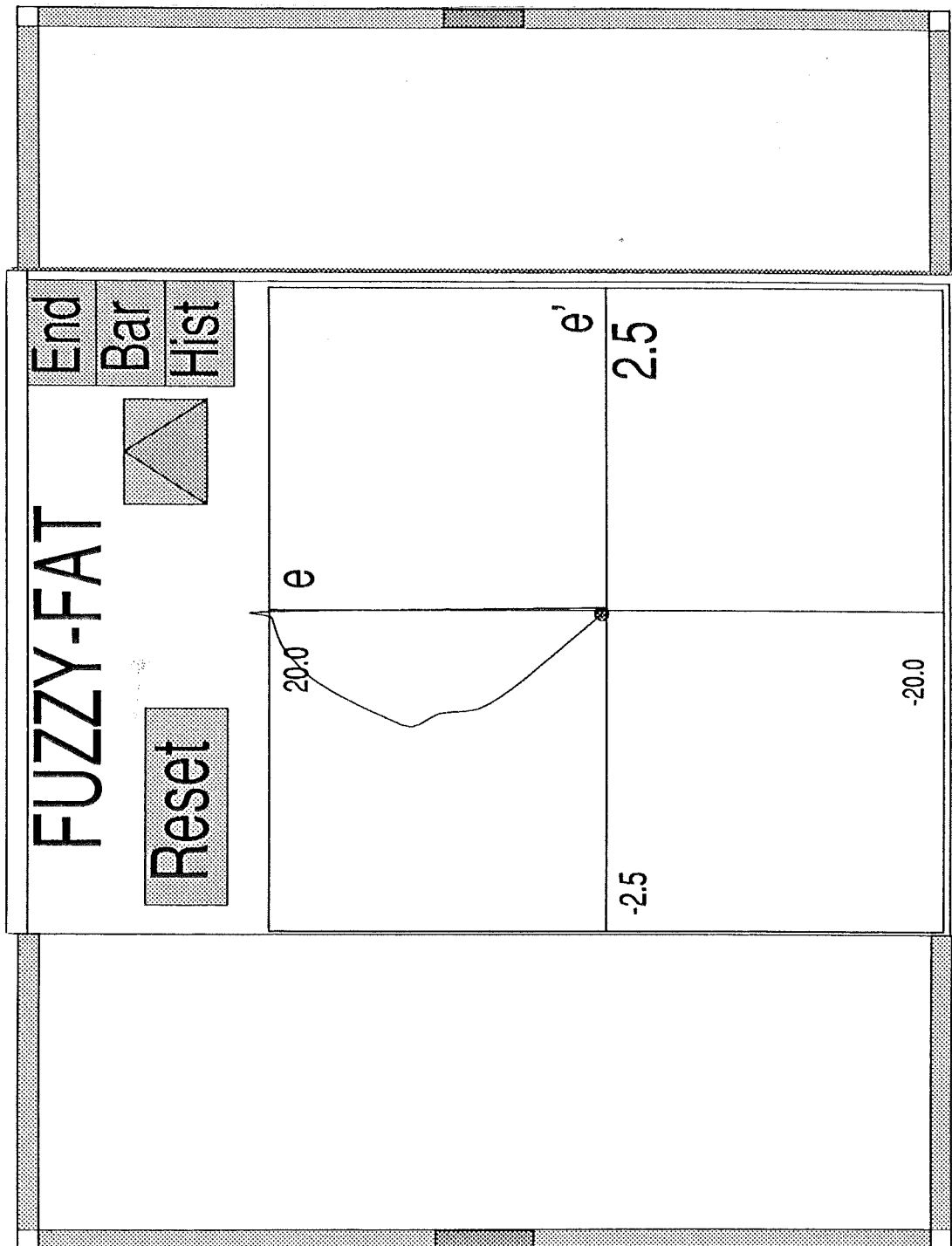
Simuleringsresultat 1a.

Här nedan visas stegsvaren dels för fuzzyregulatorn i kaskad med en PID-regulator och dels för de två kaskadkopplade PID-regulatorerna vid en börvärdesändring från 30 till 50 %.



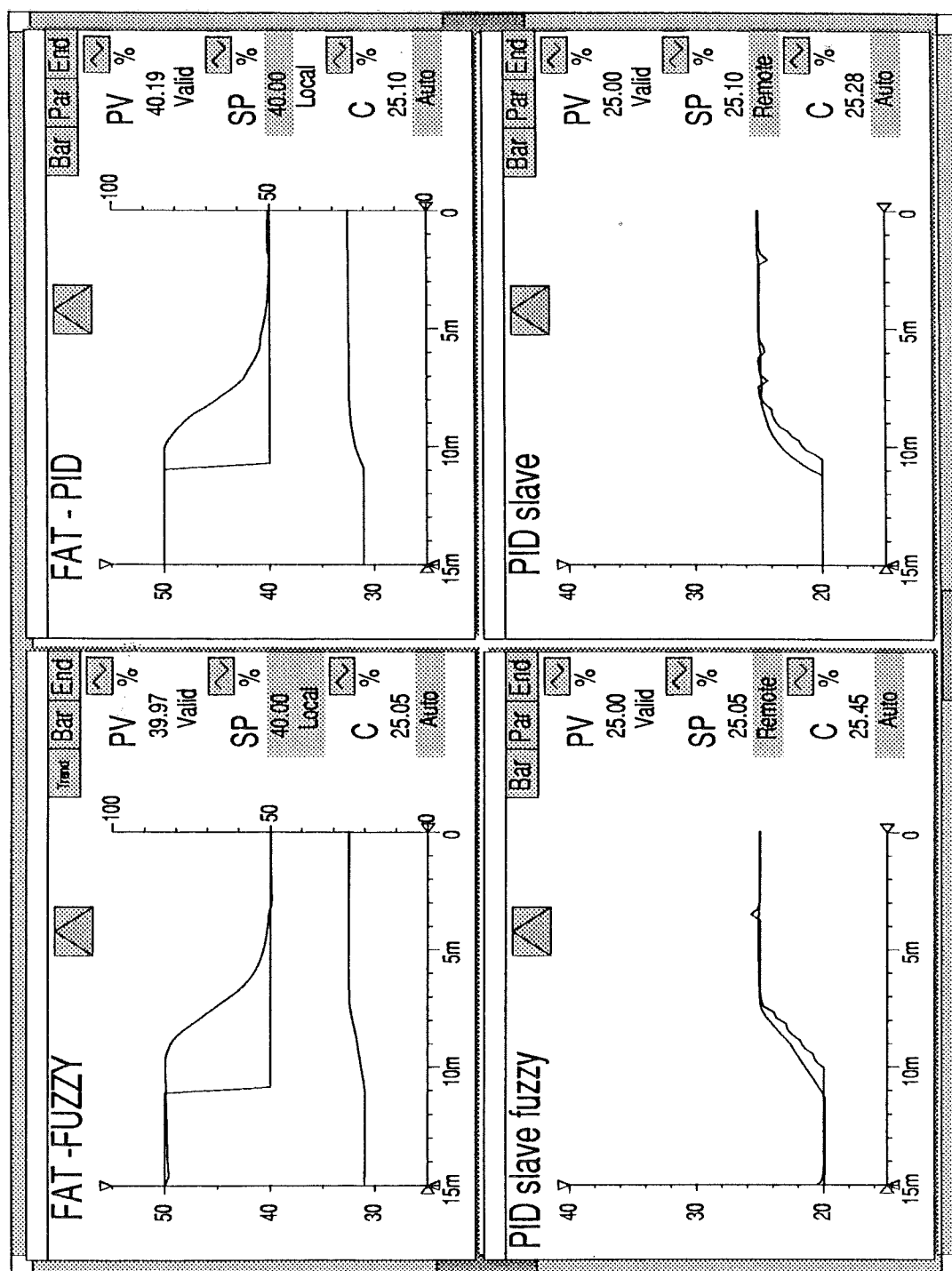
Simuleringsresultat 1b.

Här nedan visas xy-diagrammet över felet och dess förstadiifferens för fuzzyregulatorn vid en börvärdesändring från 30 till 50 %.



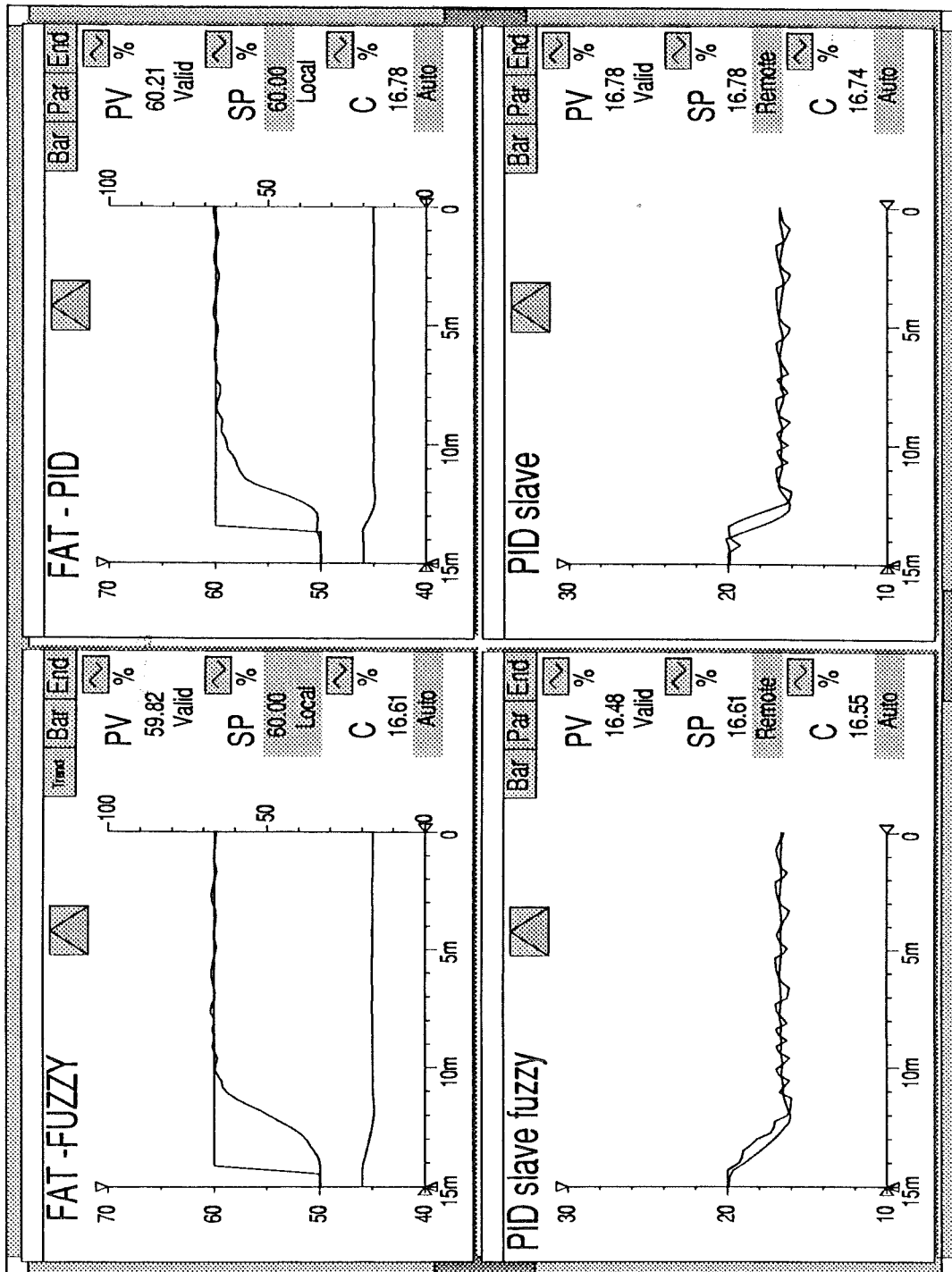
Simuleringsresultat 2.

Här nedan visas stegsvaren dels för fuzzyregulatorn i kaskad med en PID-regulator och dels för de två kaskadkopplade PID-regulatorerna vid en börvärdesändring från 50 till 40 %.



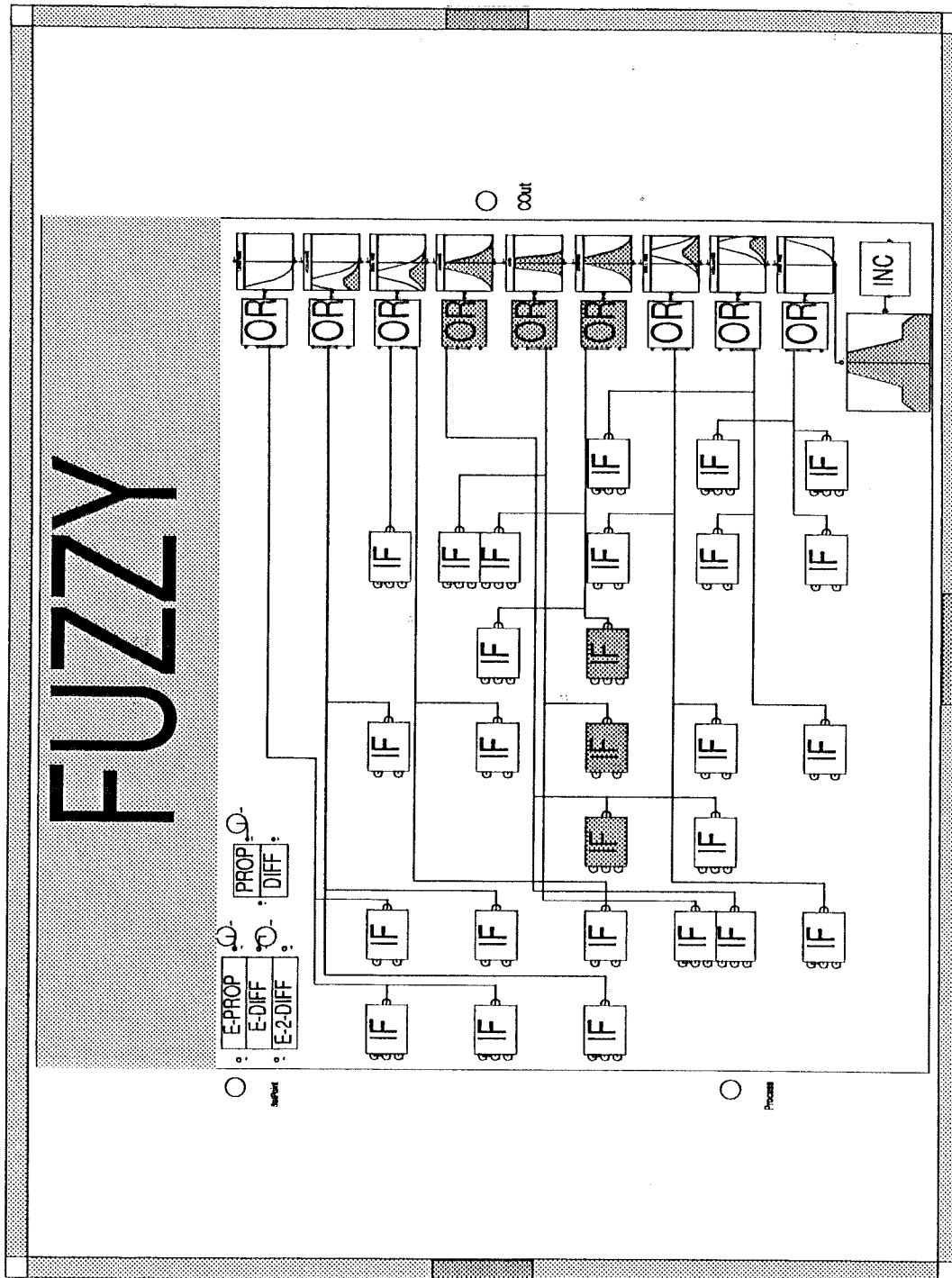
Simuleringsresultat 3.

Här nedan visas stegsvaren dels för fuzzyregulatorn i kaskad med en PID-regulator och dels för de två kaskadkopplade PID-regulatorerna vid en börvärdesändring från 50 till 60 %.



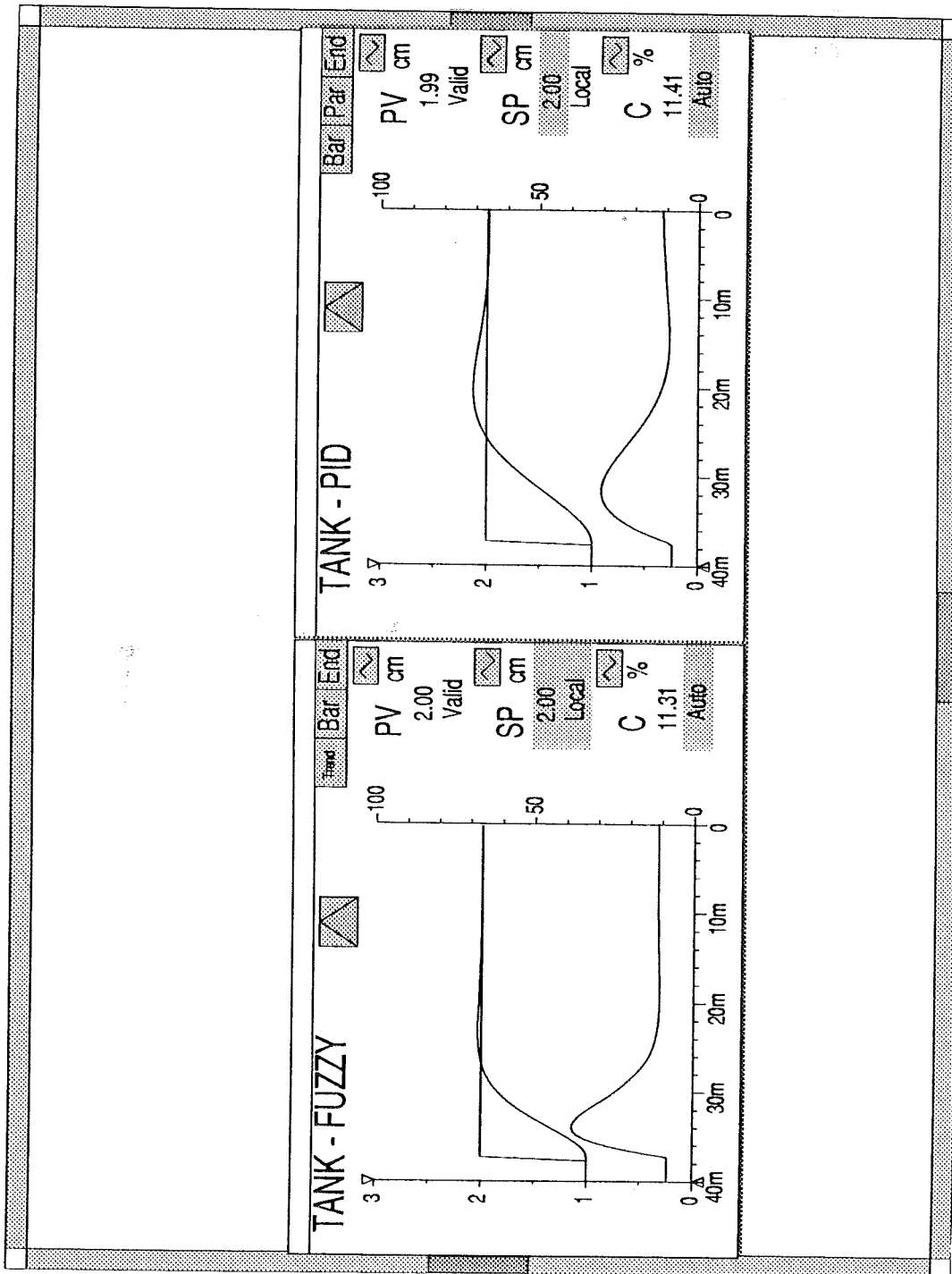
BILAGA 4

Bilaga 4 innehåller fuzzyregulatorn till vattentanken samt simuleringsresultat dels från simuleringar med fuzzyregulatorn och dels simuleringar med en fix PID-regulator. Här nedan visas den grafiska representationen av själva fuzzyregulatorn.



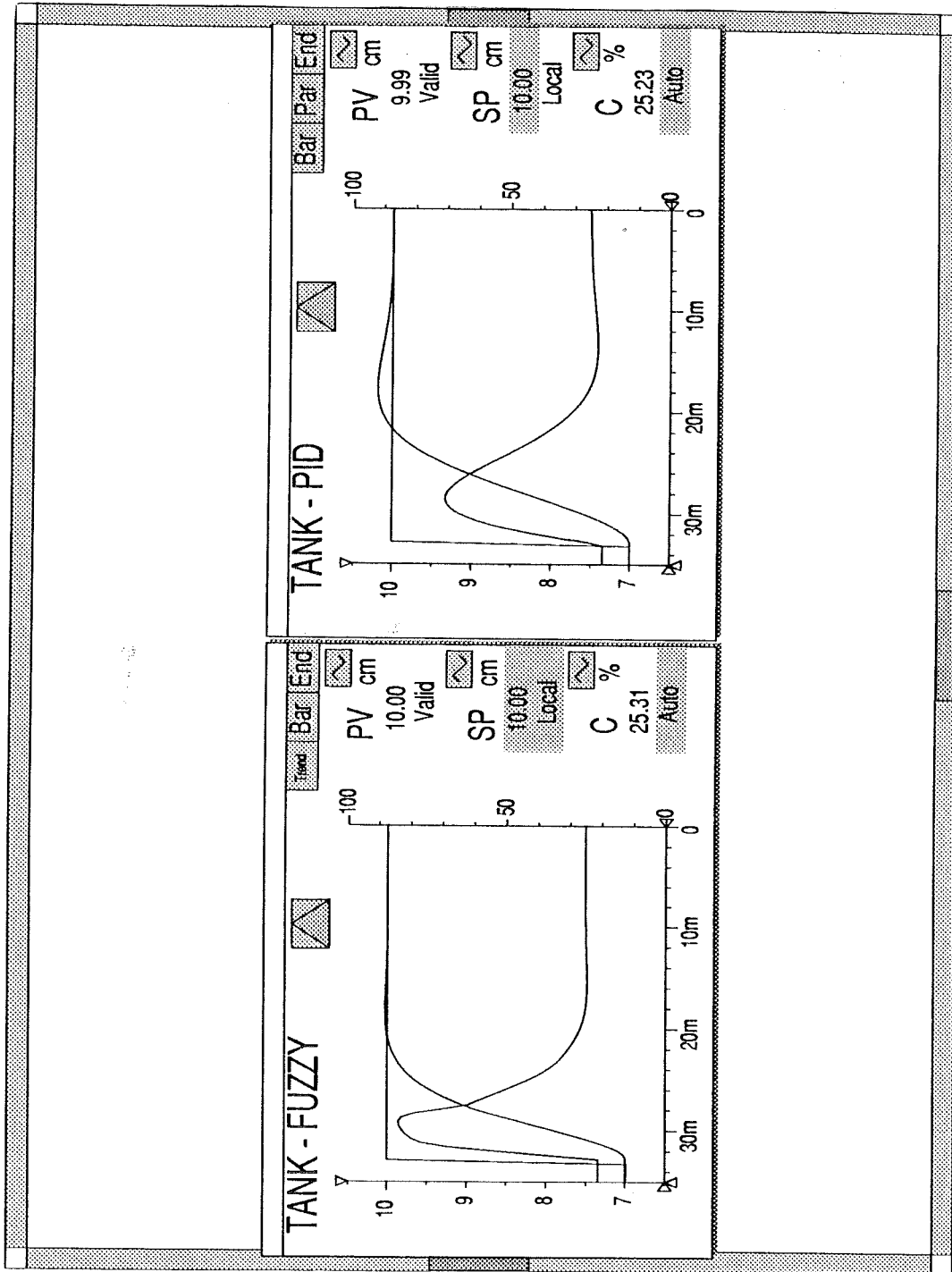
Simuleringsresultat 1.

Här nedan visas stegsvaren dels för fuzzyregulatorn och dels för PID-regulatorn vid en börvärdesändring från 1 cm till 2 cm.



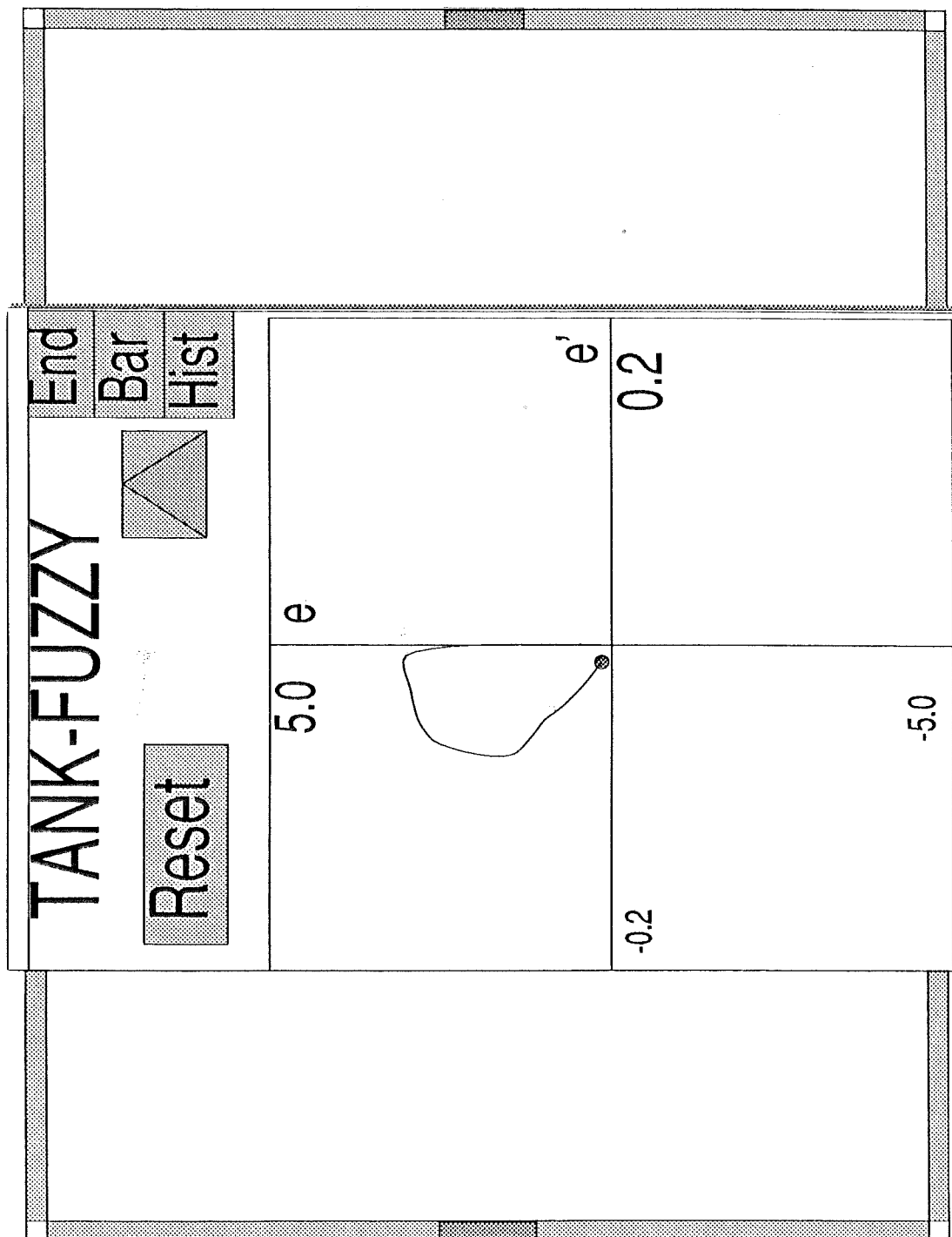
Simuleringsresultat 2a.

Här nedan visas stegsvaren dels för fuzzyregulatorn och dels för PID-regulatorn vid en börvärdesändring från 7 cm till 10 cm.



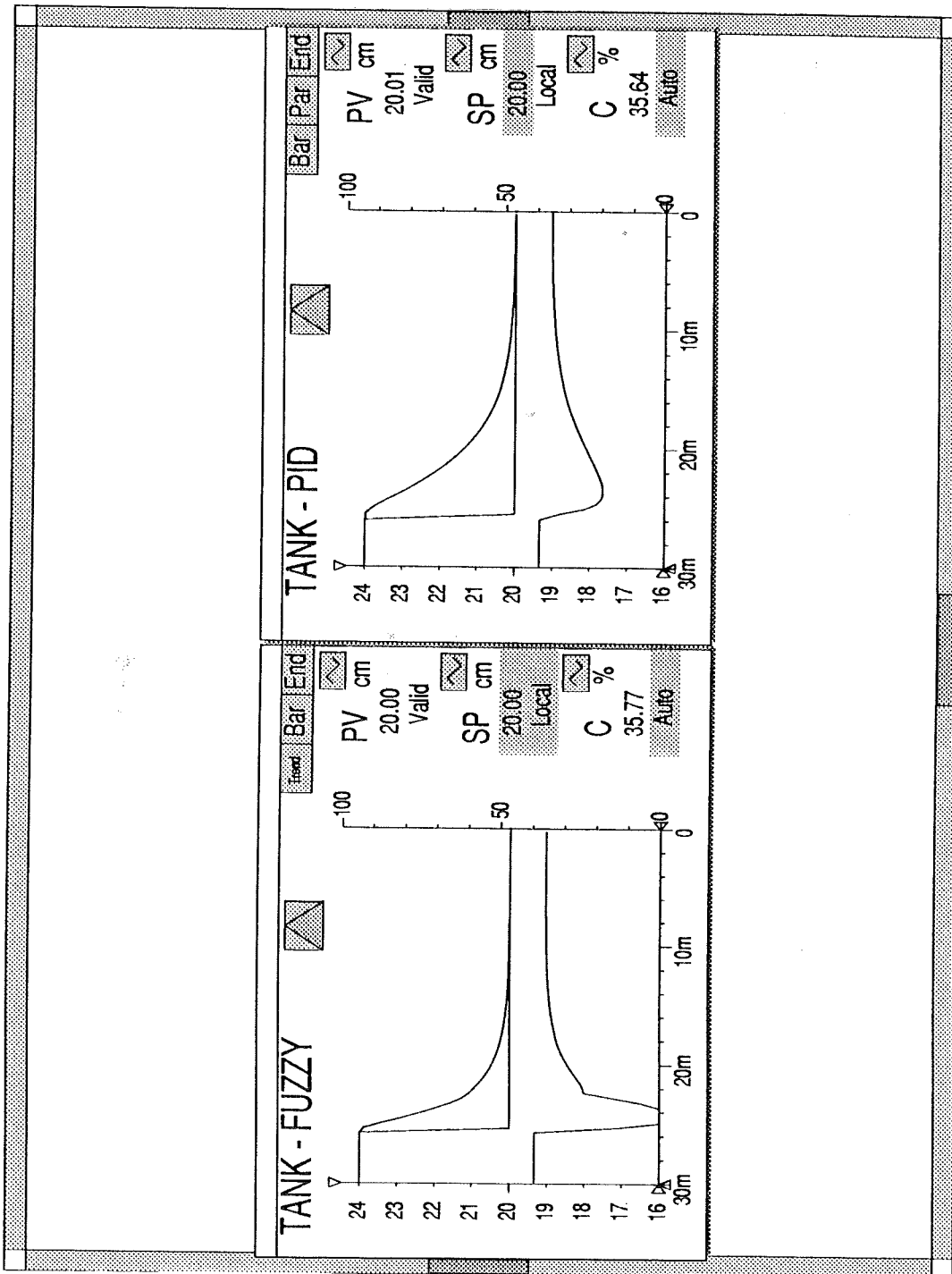
Simuleringsresultat 2b.

Här nedan visas xy-diagrammet över felet och dess förstadiendifferens för fuzzyregulatorn vid en börvärdesändring från 7 cm till 10 cm.



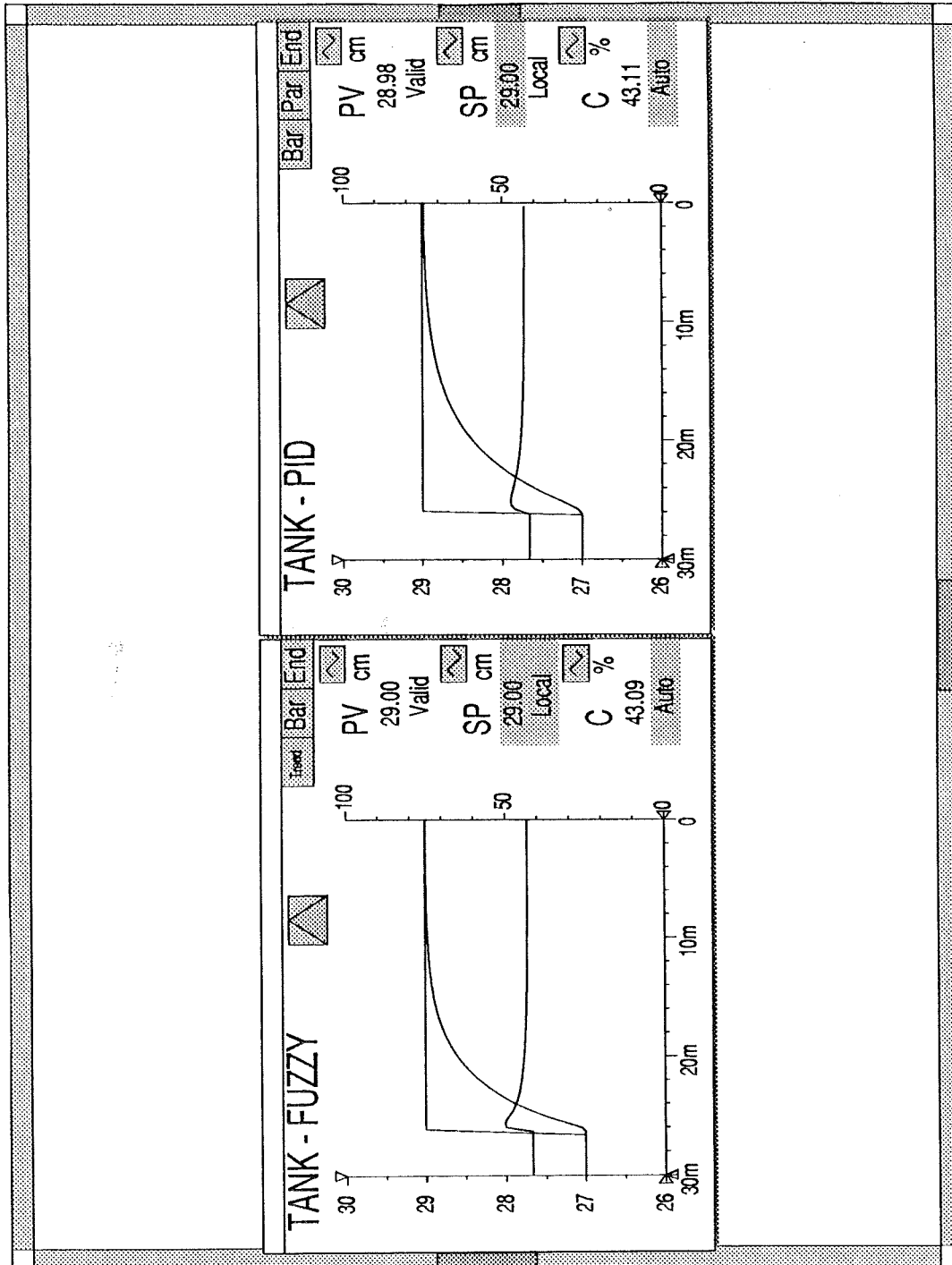
Simuleringsresultat 3.

Här nedan visas stegsvaren dels för fuzzyregulatorn och dels för PID-regulatorn vid en börvärdesändring från 24 cm till 20 cm.



Simuleringsresultat 4a.

Här nedan visas stegsvaren dels för fuzzyregulatorn och dels för PID-regulatorn vid en börvärdesändring från 27 cm till 29 cm.



Simuleringsresultat 4b.

Här nedan visas xy-diagrammet över felet och dess förstadiendifferens för fuzzyregulatorn vid en börvärdesändring från 27 cm till 29 cm.

