

CODEN: LUTFD2/(TFRT-5297)/1-54/(1983)

TEMPERATURREGLERING AV TORKPROCESS MED
DIGITAL REGULATOR

MIKAEL SALMI

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA
MAY 1983

LUND INSTITUTE OF TECHNOLOGY DEPARTMENT OF AUTOMATIC CONTROL Box 725 S 220 07 Lund 7 Sweden		Document name Report	
		Date of issue May 1983	
		Document number CODEN:LUTFD2/(TFRT-5297)/1-54/(1983)	
Author(s) Mikael Salmi		Supervisor Carl Mannerfeldt, Matz Lenells	
		Sponsoring organization	
Title and subtitle Temperature Control of Dryingprocess with a Digital Controller. (Temperaturreglering av torkprocess med digital regulator.)			
Abstract In this report, the dynamics of a <u>drying drum</u> in a <u>fertilizer</u> industry have been studied, and a digital temperature controller have been designed. A number of step responses were measured, to get an idea on how the process worked. From the step responses the conclusion were drawn that a cascads-controller, feed-backed from both the temperature into and out from the drying drum, would control the temperature sufficiently. To find the best parameter values two different strategies were used: the first one is to optimize the step response of the temperature and the other one is to simulate a physical model of the drying drum with a simulation-program. For both the strategies a sampling-interval of one minute were used. The program was implemented in the computer and the controller were tested on the real process. The parameter values were adjusted and the final result was: $K_p = 0.3$, $K_{pi} = 31 \text{ l/}^\circ\text{C}$, $T_i = 8.5$ minutes. Although the simulation were made with a rough physical model, its parameter values were much closer to the final ones than the values from the optimization, This shows that one can get good results even with a simple model. However, it is not always that easy to put up a physical model. In the last chapter the control system has been examined. The meaning is to show how a working system in industry can look like, and to show how the actual control design is put into the computer.			
Key words			
Classification system and/or index terms (if any)			
Supplementary bibliographical information			
ISSN and key title			ISBN
Language Swedish	Number of pages 54	Recipient's notes	
Security classification			

In this report, the dynamics of a drying drum in a fertilizer industry have been studied, and a digital temperature controller have been designed.

A number of step responses were measured, to get an idea on how the process worked. From the step responses the conclusion were drawn that a cascade-controller, feedbacked from both the temperature into and out from the drying drum, would control the temperature sufficiently. To find the best parameter values two different strategies were used: the first one is to optimize the step response of the temperature and the other one is to simulate a physical model of the drying drum with a simulation-program.

For both the strategies a sampling-interval of one minute were used. The program was implemented in the computer and the controller were tested on the real process. The parameter values were adjusted and the final result was: $K_p = 0.3$, $K_{pi} = 31 \text{ } 1/^\circ\text{C}$, $T_i = 8.5 \text{ minutes}$.

Although the simulation were made with a rough physical model, its parameter values were much closer to the final ones than the values from the optimization. This shows that one can get good results even with a simple model. However, it is not always that easy to put up a physical model.

In the last chapter the control system has been examined. The meaning is to show how a working system in industry can look like, and to show how the actual control design is put into the computer.

Mikael Saluni

Handledare: Carl Mannerfeldt
Matz Leunels

Temperaturreglering av torckprocess
med digital regulator

Temperature control of drying process
with a digital controller

SAMMANFATTNING

Problemet som är behandlat och löst i denna rapport är en reglering av uttemperaturen från en torktrumma hos SUPRA PK:s fabrik i Landskrona (PK står för fosfor och kalium, d.v.s. gödslets beståndsdelar).

Först utfördes ett antal stegsvarsmätningar på det öppna systemet, för att få en uppskattning om hur processen fungerade. Från processens stegsvar drogs slutsatsen att en kaskadregulator, återkopplad från in- och uttemperaturerna från torktrumman skulle räcka för att få en bra reglering. Parametrar beräknades dels med hjälp av en stegsvarsmetod, dels genom att ställa upp en fysikalisk modell och simulera på dator. Vid stegsvarsmetoden fick man följande

parametervärden: $K_p = 0.55^\circ\text{C}^{-1}$, $K_{pi} = 8.5$ (dimensionslös),

$T_i = 2$ min. och vid simuleringen fick man följande:

$K_p = 0.4^\circ\text{C}^{-1}$, $K_{pi} = 31$, $T_i = 8$ min. Båda beräkningarna

gjordes för samplingsintervallet en minut. Regleringen gjordes med PK-fabrikens processdator och flera olika tester gjordes på den verkliga processen. Parametervärdena justerades och det bästa resultatet fick man med följande

värden: $K_p = 0.3^\circ\text{C}^{-1}$, $K_{pi} = 31$, $T_i = 8.5$ minuter.

Fast att simuleringen utfördes med en grov fysikalisk modell, kom den slutliga parameterinställningen att ligga närmast de parametervärdena som erhöles vid simulering, vilket visar att man kan få bra resultat även med en enkel modell. Det är dock inte alltid så lätt att få fram en fysikalisk modell.

I kapitel 3 har PK-fabrikens processdatorsystem beskrivits ganska ingående, med tonvikt på IPCL-språket och reglerpaketet. Det finns två syften med detta. Det första är att visa hur ett fungerande system i industrin kan se ut och vilka möjligheter och begränsningar det har. Det andra är att beskrivningen ska kunna vara till hjälp för den som vill sätta sig in i hur den aktuella regulatorn är förverkligad med hjälp av datorn.

INNEHALLSFÖRTECKNING

	sid
Kap1. PROBLEMBESKRIVNING.....	4
1.1 Processystem.....	4
1.2 Reglerproblemet.....	4
Kap2. REGULATORFRAMTAGNING.....	7
2.1 Fysikaliska överväganden.....	7
2.2 Beräkning av parametrar med hjälp av stegsvar..	10
2.3 Simulering.....	12
Kap3. PROCESSDATORSYSTEM.....	18
3.1 Inledning.....	18
3.2 Operativsystemet.....	18
3.3 Allmänt om IPCL.....	20
3.4 Operationer och Instruktioner.....	22
3.5 On-line programmering.....	26
3.6 Reglering.....	27
3.7 Programmet.....	31
Kap4. UTVÄRDERING.....	34
REFERENSER.....	40
APPENDIX	
1. Syntax för flerordsinstruktioner.....	41
2. Instruktionsuppsättning IPCL.....	42
3. Operatörskommando.....	46
4. SIMNON-program.....	49

KAPITEL 1

kap 1 PROBLEMBESKRIVNING1.1 Processystem

PK-fabrikens process beskrivs av processschemat på nästa sida. De råvaror som används är superfosfat, kaliumklorid, magnesiumsulfat, kopparsulfat, borat och kopparklorid. Råvarorna fördelas med hjälp av ett råvaruintransportsystem till olika fickor. Därifrån tages de ut med ett doseringssystem som, beroende på vilket recept som används, transporterar rätt mängd råvaror till ett samlingsband. Råvarorna blandas med returgoods från sikten, och transporteras därefter till granulatoren.

Granulatoren är en 11 meter lång roterande trumma med 2,8 meters diameter där råvarorna blandas. Genom en dysa tillsätts tvättvätska i början av granulatoren. Regleringen av mängden tvättvätska beräknas utifrån hur mycket material som finns i granulatoren och hur mycket av det som är råvaror, finretur eller krossgods. Med ett kommando på tangentbordet anger man vilken fukthalt materialet i granulatoren skall ha. Ju högre fukthalt desto kraftigare granulering och större granuler. Från granulatoren faller materialet genom en tratt ner i torktrumman, där det möter varma rökgaser från brännkammaren.

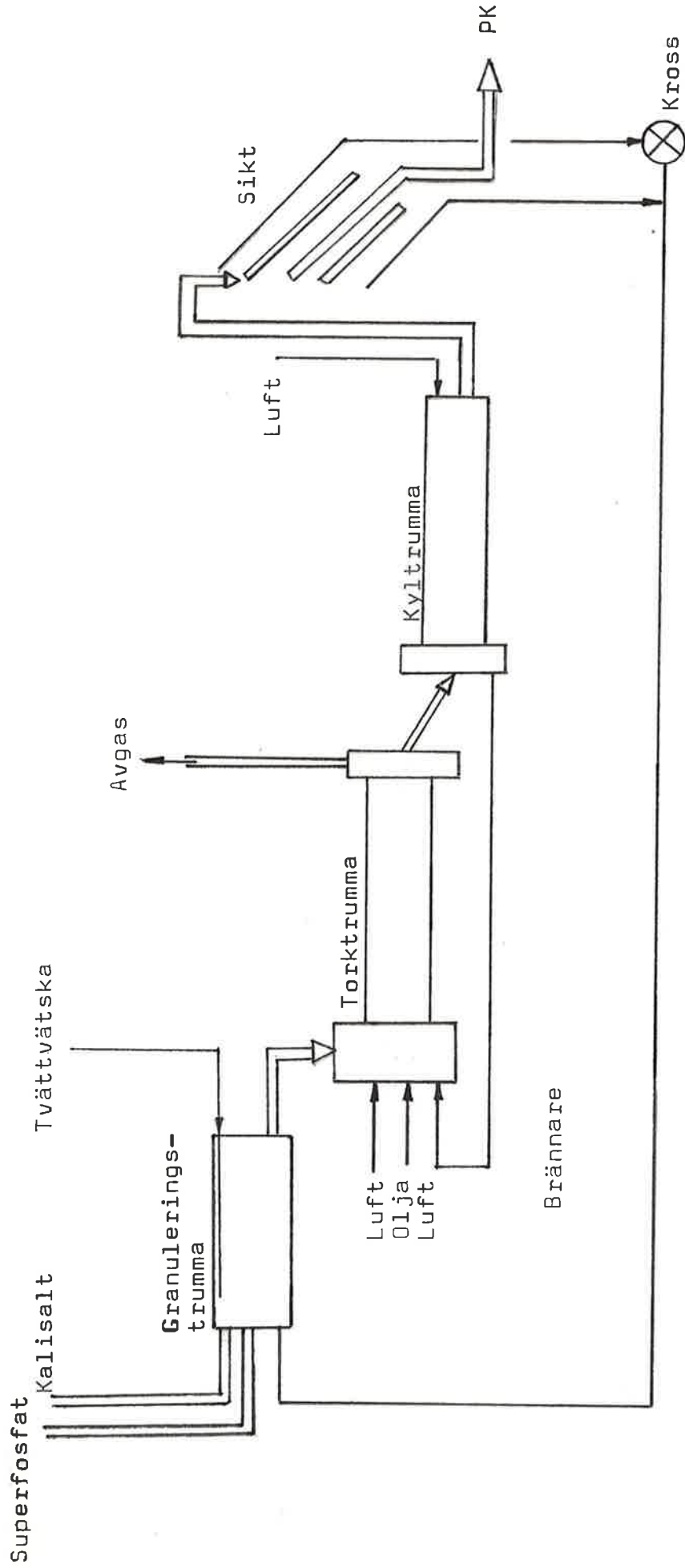
Torktrumman är 22 meter lång och 3 meter i diameter, och roterar med ca 3,5 varv/minut. Uppehållstiden i torktrumman är ca 13 minuter men varierar med cirkulationsmängden. Sekundärluften till brännaren tas från kyltrumman för att använda värmeinnehållet i kylavgaserna. I inloppet på torktrumman har rökgaserna en temperatur på mellan 500 och 700°C, beroende på driftsförhållanden, och vid utloppet en vald temperatur kring 110°C.

Efter torktrumman transporteras det varma godset till kyltrumman. Kyl luften blåses in i kyltrummans utloppsända. När gödseln lämnar kyltrumman har den kylts till 35-40°C. Därefter transporteras den kylda gödseln till sikten, som består av en överkorns- och en underkornssikt sammanbyggda till en enhet. Överkornen krossas och förs tillsammans med underkornen tillbaka till granulatoren.

1.2 Reglerproblemet

Problemet jag fått mig tilldelat att lösa är att få uttemperaturen från torktrumman konstant. Genom uttemperaturen påverkar man fukthalten i den färdiga produkten, vilken man vill ha till 2%. Om fukthalten är högre kan man inte lagra produkten p.g.a. att gödselgranulerna klumpar ihop sig, och en lägre fukthalt innebär att man torkar materialet onödigt mycket vilket betyder ökade bränslekostnader. Därför vill man ha

processchema



KAPITEL 1

fukthalten så nära 2% som möjligt. Fukthalten mäts upp i ett laboratorium. Det är därför omöjligt att reglera på fukthalten, utan man får istället försöka hålla en så jämn torkning som möjligt genom att hålla uttemperaturen konstant, ca 110°C.

Med en bättre reglering kan man eventuellt minska oljekostnaderna. Det skulle även innebära en större möjlighet att påverka granulatorstorleken, och därigenom kunna få mindre mängd returgoods, d.v.s. få en snabbare produktgenomströmning.

KAPITEL 2

Kap 2 REGULATORFRAMTAGNING

I detta kapitel skall regulatorns utseende samt dess parametervärden bestämmas. Först kommer ett avsnitt om fysikaliska överväganden, i vilket regulatorstrukturen bestämmas. Därefter kommer ett avsnitt där parametervärdena bestämmas. I det tredje och sista avsnittet beräknas en fysikalisk modell för torktrumman, som därefter simuleras på dator.

2.1 fysikaliska överväganden

I detta avsnitt undersöks de fysikaliska förhållandena för torktrumman genom mätning av olika storheter som kan tänkas påverka uttemperaturen. Därefter bestämmas en lämplig regulatorstruktur för regleringen.

Torkningen är direkt proportionell mot skillnaden i in- och uttemperaturerna, eftersom avgiven värme från luften är

$$Q = L \cdot C_p \cdot (T_{IN} - T_{ut})$$

där L är luftflödet.

Om fukthalten i materialet som kommer in i torktrumman ökar, eller om produktmängden ökar, avkyls luften i torktrumman och uttemperaturen sjunker. Om man i något av dessa fall ökar intemperaturen så att uttemperaturen förblir konstant avges mer värme och materialet får ungefär samma fukthalt ut från torktrumman.

Fukthalten i materialet och mängden produkt från granulatore till torktrumman är två faktorer som påverkar uttemperaturen. En annan faktor som kan tänkas påverka är sekundärluftens temperatur som kommer från kyltrumman. Intemperaturen till torktrumman är givetvis den faktor som påverkar uttemperaturen mest, och det är genom denna man kan styra uttemperaturen. Genom att öka oljemängden till brännaren, genom att ändra oljeventilöppningen, ökar man intemperaturen som i sin tur ökar uttemperaturen. För närvarande styrs uttemperaturen genom att man ändrar ventilläget med en handstation i instrumentpanelen. Ventilläget styrs över en reglermodul med ström utgång 4-20 mA. Modulen genererar en ström som styrs av en digital-analog-omvandlare med 8 bitars upplösning. Vid oljeventilen sitter en lägesregulator som med hjälp av en kam kompenserar oljeventilens olinjäritet, så att förhållandet mellan utsignalen och volymströmmen blir approximativt linjärt. Nollläget justeras med två länkarmar som sitter vid lägesregulatorn. Inställningen är för närvarande inte korrekt, d.v.s. stängd ventil vid 54 bitar, utan ventilen blir inte helt stängd förrän utsignalen sjunkit till strax under 50 bitar. Vid en utsignal på 115 bitar är oljevolymin 10,2 liter/minut och med utsignalen 129 bitar är oljevolymin 11,7 liter/minut.

För att få reda på hur temperaturen påverkades gjordes ett antal stegändringar i ventilöppningen. Torktrummas in- och uttemperatur, utsignalen till oljeventilen, mängden tvättvätska och cirkulerande mängd gods mättes var tionde sekund. Tanken med detta var att få klarhet i hur de olika mätvariablerna påverkade uttemperaturen. Om någon av mätvariablerna visade en klart mätbar påverkan på uttemperaturen skulle man kunna göra en framkoppling och på så sätt eliminera störningen. Resultatet av en stegändring visas i fig 2.1.

I kontinuerlig drift varierar mängden gods förhållandevis långsamt och någon påverkan på uttemperaturen av variationen i cirkulerande mängd kunde inte urskiljas.

Vid riktigt stora stegändringar ändras andelen återcirkulerande underkorn. Därmed ändras även tvättvätskemängden genom processregleringen. Resultatet blir att torktrummas temperatur påverkar tvättvätskemängden. Vid små stegändringar kunde man inte observera något samband överhuvudtaget.

Sekundärluftens uttemperatur steg 10 grader på en timme då

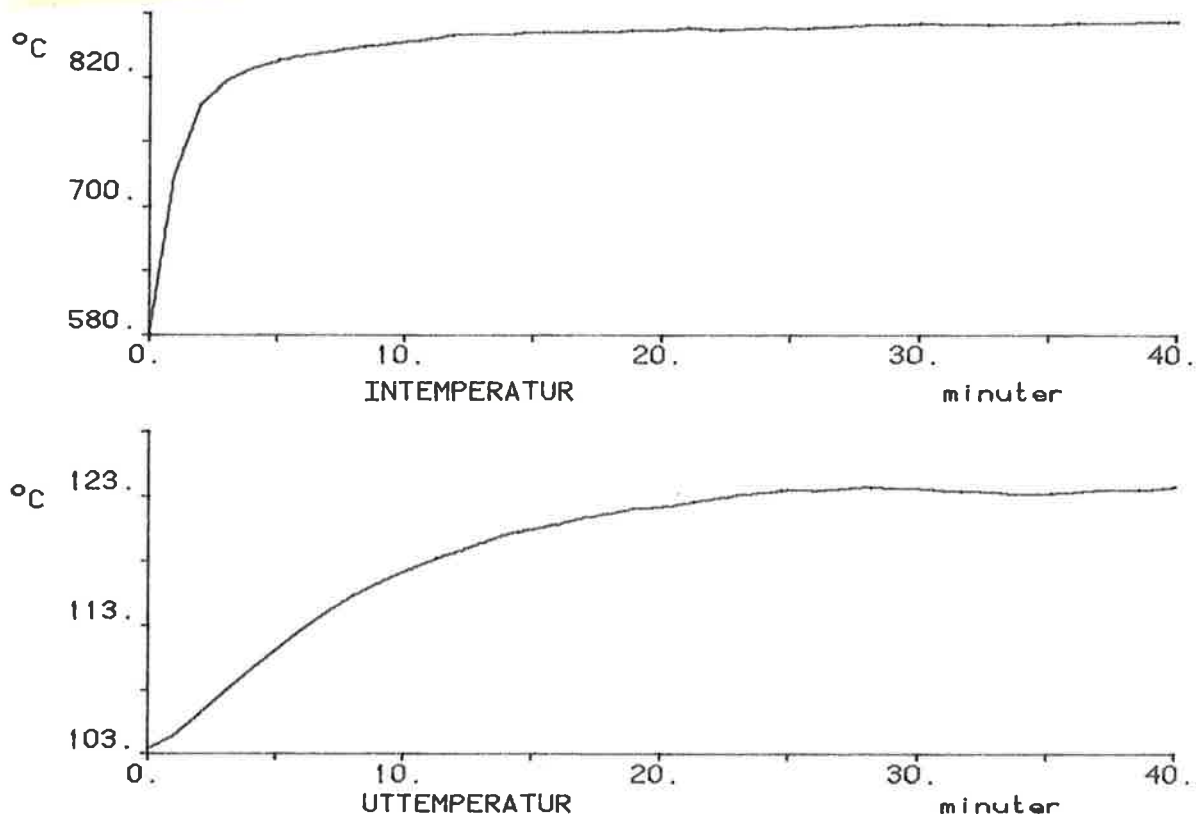


fig 2.1 stegsvar då ett steg sker i oljetillförseln till brännaren.

intemperaturen till torktrumman ökades 280 grader. Av detta kan man dra slutsatsen att sekundärluften inte påverkar torktrummans temperatur i någon mätbar omfattning.

Torktrummans intemperatur är däremot intressant. Den når inte sitt slutvärde förrän efter i storleksordningen 10 minuter, och uttemperaturen är direkt beroende av intemperaturen.

Resultatet blev således att den enda faktor som man bör ta hänsyn till är intemperaturens dynamik; man reglerar uttemperaturen genom att reglera oljeflödet till brännaren på basis av mätning av in- och uttemperaturen i torktrumman, s.k. kaskadreglering, se fig 2.2.

Den ena regulatorn som styr reglerventilen har intemperaturen som insignal. Börvärdet till denna regulator erhålls som utsignal från den andra regulatorn, som har uttemperaturen som insignal. Regulatorn i den inre kretsen minskar inverkan av intemperaturens dynamik. Denna regulator bör inte ha någon I-del. Dess börvärde är ju den andra regulatorns utsignal, och en eventuell I-del hade fasförskjutet signalerna och kanske minskat regleringens hastighet.

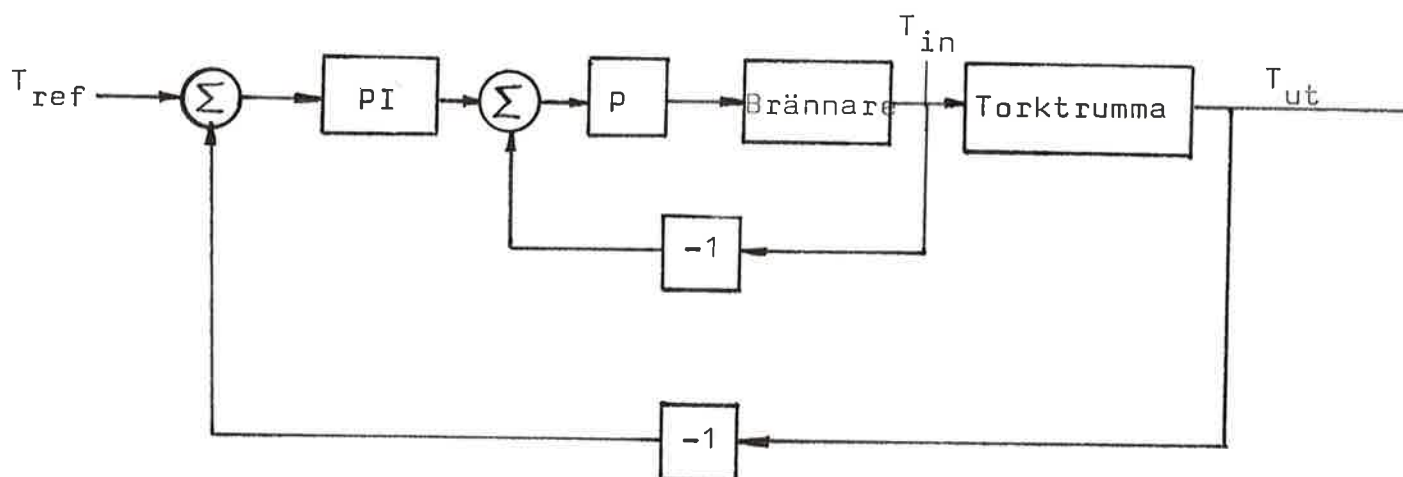


Fig 2.2 Kaskadreglering av torkprocessen.

KAPITEL 2

2.2 Beräkning av parametrar med hjälp av stegsvar

Brännaren

Med ledning av stegsvaren för intemperaturen kan man anse sig kunna approximera brännaren med ett första ordningens system.

Om $y(t)$ = intemperaturen vid tiden t och $u(t)$ = ventilläget vid tiden t får man att:

$$y(t+1) = ay(t) + bu(t)$$

där samplingsintervallet är satt till en tidsenhet, och a och b är okända konstanter. För att få reda på konstanterna a och b användes stegsvarmätningarna (se fig 2.1).

Anta att $y(0) = 0$ och $u(t) = 171 - 54 = 117$. Då får man:

$$\begin{array}{l} y(1) = 730 - 581 = 149^\circ\text{C} \\ y(\infty) = 870 - 581 = 289^\circ\text{C} \end{array} \quad \left. \vphantom{\begin{array}{l} y(1) \\ y(\infty) \end{array}} \right\} \Rightarrow 149 = b \cdot 117 \quad \Rightarrow \quad b = 1.2735^\circ\text{C}$$

$$289/117 = b/(1-a) \quad \Rightarrow \quad 1-a = 149/289 \quad \Rightarrow \quad a = 0.4844$$

Statiska förstärkningen är $b/(1-a) = 289/117 = 2.47^\circ\text{C}$

Överföringsfunktionen för det öppna systemet är

$$H_0(z) = b/(z-a)$$

Återkoppling med en P-regulator med förstärkningen K ger

$$H(z) = H_0(z)/(1+H_0(z)) = bK/(z-a+bK)$$

Systemets pol är $z = a - bK$

För att systemet skall vara stabilt måste polerna i ett samplat system vara inom enhetscirkeln i z -planet. Stabilitetsgränserna för systemet blir:

$$K_{\min} = (a-1)/b = -0.4 \text{ } 1/^\circ\text{C}$$

$$K_{\max} = (a+1)/b = 1.166 \text{ } 1/^\circ\text{C}$$

Snabbaste stegsvaret utan översläng får man om man, enligt Dead-Beat strategin, placerar polen i origo. Då ska stegsvaret nå sitt slutvärde på ett samplingsintervall.

$$a - bK = 0 \quad \Rightarrow \quad K_{\text{nom}} = 0.4844/1.2735 = 0.38 \text{ } 1/^\circ\text{C}$$

KAPITEL 2

Två andra stegsvarsmätningar ger följande:

		<u>1</u>	<u>2</u>
a	=	0.68	0.46
b	=	0.97°C	1.27°C
b/(1-a)	=	3.0°C	2.35°C
K _{min}	=	-0.33 1/°C	-0.425 1/°C
K _{max}	=	1.73 1/°C	1.15 1/°C
K _{nom}	=	0.7 1/°C	0.362 1/°C

Observera att systemet har approximerats med ett första ordningens system utan tidsfördröjning.

P-regulatorn stoppades in och testades för att få ett snyggt stegsvar. $K = 0.45$ tycktes vara ett bra val. Man fick en liten översläng, men samtidigt skulle det ju snabba upp uttemperaturens stegsvar. Det tog i storleksordningen 4 minuter för intemperaturen att nå sitt slutvärde, vilket bevisar att brännaren inte är ett första ordningens system.

Torktrumman

Torktrumman kan knappast approximeras med ett första ordningens system. Tidsfördröjningen är alltför stor. Istället får man tillgripa någon sorts inställningsregel. Från stegsvaren beräknas a , b , T_d och T enligt fig 2.3.

Enligt kriteriet ITAE (se Åström 5.7) så minimerar man det integrerade, tidsmultiplicerade absolutfelet, d.v.s. man minimerar:

$$\int t * |e(t)| dt$$

Detta gör man genom att sätta, för en PI-regulator, förstärkningen $K = 0.86/a$ och integraltiden $T_i = T_d * 1.5 * r^{-0.3}$, där $r = T_d/T$. Detta gäller när processförstärkningen = 1. Därför måste K multipliceras med $b/u(t)$. Två olika stegsvar gav följande:

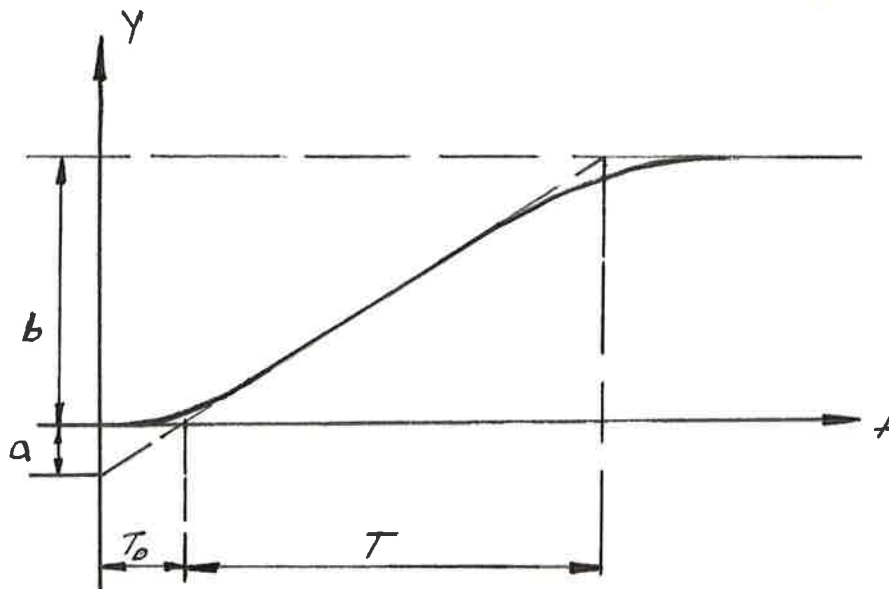


fig 2.3 beräkning av a , b , T_0 och T från stegsvar.

	1	2
K	10.0	8.44
T_i	1.8 min	3.57 min

I en PI-regulator skall integraltiden vara ungefär lika med stigtiden, vilken är 9,2 respektive 12 minuter i exemplena ovan. Skillnaden är markant, men eftersom båda beräkningarna bara är riktlinjer för parametervalet har det inte så stor betydelse.

2.3 Simulering

I detta avsnitt beräknas en fysikalisk modell för torktrumman, som därefter simuleras på dator. Programmen finns i appendix 4.

Som tidigare nämnts är förhållandet mellan insignalen till D/A-omvandlaren och oljevolymsströmmen approximativt linjär. En insignaländring på 1 bit (1 bit = 1/256 av fullt utslag) motsvarar en volymändring på ca 0.11 liter/minut. Med oljans värmevärde $H = 42.5$ MJ/kg och densiteten $\rho = 0.9$ kg/liter får man att 1 bits insignaländring motsvarar

$$V \cdot \rho \cdot H = 0.11 \cdot 0.9 \cdot 42500 / 60 = 70 \text{ kW}$$

Luftflödet i torktrumman är ungefär 700 kg/min och luftens värmekapacitivet $C_p = 1.01 \text{ kJ/kg}$. Detta ger att 1 bits signaländring höjer luftens temperatur med

$$\Delta T = E / (C_p \cdot m) = 70 \cdot 60 / (1.01 \cdot 700) = 5.94^\circ\text{C}$$

Alltså en statisk förstärkning på 5.94°C/bit . I föregående avsnitt beräknades den statiska förstärkningen med utgångspunkt från mätningar till 2.47°C/bit . Detta innebär att endast 42 % av tillgänglig energi används till att värma luften. Resterande mängd är förluster, utom en mindre del som värmer upp granulerna direkt. De framräknade siffrorna är dock behäftade med viss osäkerhet. Ventilkaraktistikan är inte perfekt linjär och temperaturgivaren mäter inte den exakta genomsnittliga intemperaturen.

I stället för att räkna på volymströmmen antas att förhållandet mellan insignalen till D/A-omvandlaren och intemperaturen är det första ordningens system som räknades fram i föregående avsnitt. Omvandlat till kontinuerlig tid fås:

$$dT/dt = A \cdot T + B \cdot U$$

där $U =$ ventilläget

$$A = -0.716 \text{ min}^{-1}$$

$$B = 1.77^\circ\text{C/min}$$

Om man delar in torktrumman i tre sektioner med sektionsvis konstanta storheter får man med en energibalans för luften

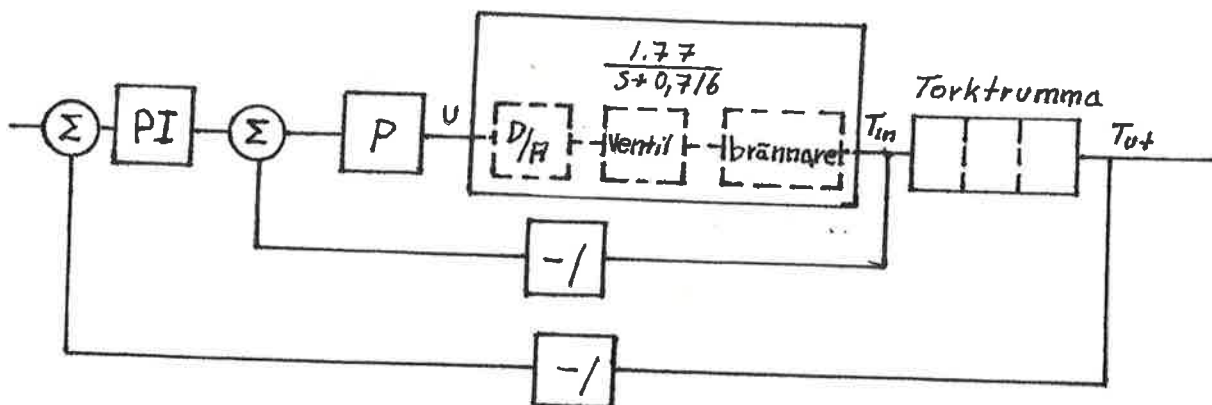


fig 2.4 Det simulerade systemets blockschema

KAPITEL 2

(se fig 2.5):

$$M \cdot C_p \cdot dT_i / dt = m \cdot (C_{p,i-1} \cdot T_{i-1} - C_{p,i} \cdot T_i) - K \cdot (T_i - T_{g,i}) + \\ + (m_{v,i-1} \cdot i_{i-1} - m_{v,i} \cdot i_i) + (m_{v,i-1} - m_{v,i}) \cdot \Delta i$$

På motsvarande sätt får man med en energibalans för granulerna:

$$G \cdot C_{pg} \cdot dT_{g,i} / dt = m_g \cdot C_{pg} \cdot (T_{g,i-1} - T_{g,i}) + (i_{li-1} \cdot m_{li-1} - i_{li} \cdot m_{li}) + \\ + (K - K_v) \cdot (T_i - T_{g,i})$$

$$i_i = 2.087 \cdot T_i + 2467.5 \text{ J/(kg} \cdot \text{°C)} \quad (= \text{vattenångans entalpi})$$

$$\Delta i = 2.087 \cdot (T_i - T_{g,i})$$

K = Värmeövergångskoefficient mellan luft och granuler

T_i = Luftens temperatur i sektion i

$T_{g,i}$ = Granulernas temperatur i sektion i

M = Mängd luft i sektion i (antas konstant = 89.4 kg)

m = massflöde luft i sektion i (antas konstant = 700 kg/min)

K_v = Värmeövergångskoefficient för ångan

G = Mängd granuler i sektion i (antas konstant = 6500 kg)

m_g = massflöde granuler i sektion i (antas konstant = 1500 kg/min)

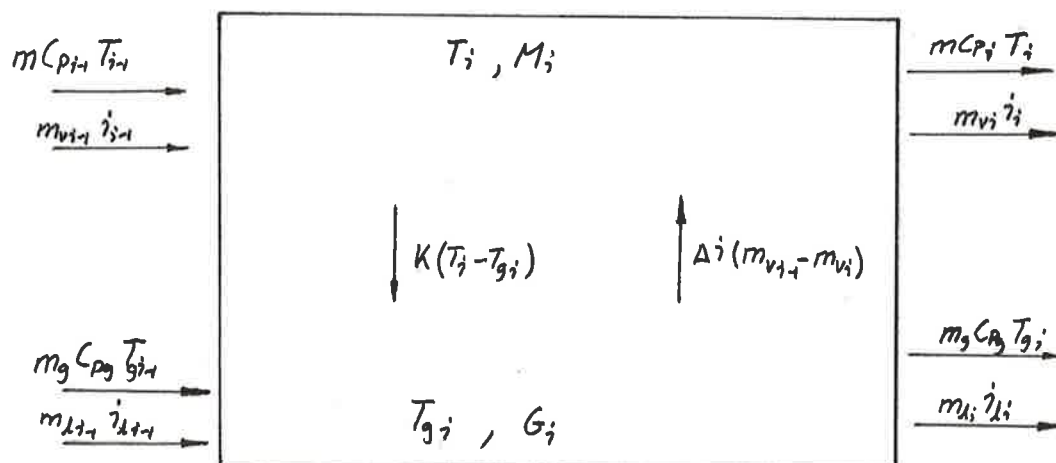


fig 2.5 energibalans för sektion i

KAPITEL 2

$$i_{li} = 4.189 * T_{gi} \quad \text{J/(kg}^{\circ}\text{C)} \quad (\text{ = vattnets entalpi})$$

m_{li} = massflöde vatten i sektion i

m_{vi} = massflöde ånga i sektion i

Det största problemet är att få en ekvation för hur vattnet i granulerna förångas. En viss del av vattnet finns inne i granulkornen. Att få fram hur detta förångas, och speciellt hur mycket energi som åtgår, är mycket svårt. Dessutom vet man inte hur stor del av vattnet som är i kornen. Därför gjordes det förenklande antagandet att ångbildningen är direkt proportionell mot temperaturskillnaden. Ett annat antagande som gjorts är att mängden luft är lika stor i de tre sektionerna. P.g.a. temperaturskillnaderna är detta inte helt korrekt. Proportionalitetskonstanten K_v och K beräknas som:

$$K_v = \frac{Q}{\Delta T_m * Z}$$

$$K = Q / (\Delta T_m * Z)$$

Z = antalet sektioner = 3

ΔT_m = logaritmiska medeltemperaturdifferensen, d.v.s.

$$\Delta T_m = \frac{(\Delta T_{in} - \Delta T_{ut})}{\ln(\Delta T_{in} / \Delta T_{ut})}$$

Q_v = energi/minut till förångning, som man får från:

$$Q_v = m * (C_p * T_{in} - C_p * T_{ut}) - m_g * C_{pg} * (T_g_{ut} - T_g_{in})$$

Q = avgiven energi/minut från luften

$$Q = m * (C_p * T_{in} - C_p * T_{ut})$$

Energibalansen för ångan blir:

$$m_{vi} * i_{vi} = m_{vi-1} * i_{vi-1} + K_v * (T_i - T_{gi})$$

Kontinuitetsekvation för vatten:

$$m_{li-1} + m_{vi-1} = m_{li} + m_{vi}$$

Värmeövergångskoefficienterna har stor inverkan på resultatet. Dessa beräknades först med följande antaganden:

$$T_{in} = 580^{\circ}\text{C}$$

$$T_{ut} = 103,4^{\circ}\text{C}$$

$$T_{gin} = 25^{\circ}\text{C}$$

$$T_{gut} = 100^{\circ}\text{C}$$

$$\Rightarrow \Delta T_m = 108.2589^{\circ}\text{C}$$

$$Q = 377556 \text{ kJ/min}$$

$$Q_v = 267306 \text{ kJ/min}$$

KAPITEL 2

$$m1_{in}/G = 11\%$$

$$m1_{ut}/G = 3\%$$

Resultatet blev då:

$$K = 1162,5 \text{ kJ}/(^{\circ}\text{C}\cdot\text{min})$$

$$Kv = 823,05 \text{ kJ}/(^{\circ}\text{C}\cdot\text{min})$$

För att få samma statiska förstärkning som verkligheten ändrades Kv till 950 kJ/($^{\circ}\text{C}\cdot\text{min}$). Programmet ses i sin helhet i bilaga 4. Modellen är grovt gjord med idealiserade förhållanden, så därför kan man inte begära att den ska vara exakt som verkligheten. Om man delat in torktrumman i fler delar, 5 eller kanske till och med 10, hade man fått en lite annorlunda tidsfördröjning, men man kan ifrågasätta om man hade kommit så mycket närmare verkligheten, på grund av de förenklade förutsättningarna. Ett stegsvar är uppritat i fig 2.6. Verkligheten, d.v.s. kurvorna från figur 2.1, är inritad i samma figur som jämförelse.

Modellen kopplades ihop med en kaskadregulator och den testades med olika parametervärde. Det bästa resultatet erhöles för $K = 35$, $T_i = 7.5$ minuter för PI-regulatorn och $K = 0.3$ 1/ $^{\circ}\text{C}$ för P-regulatorn. Svaret på en börvärdesändring är uppritat i fig 2.7.

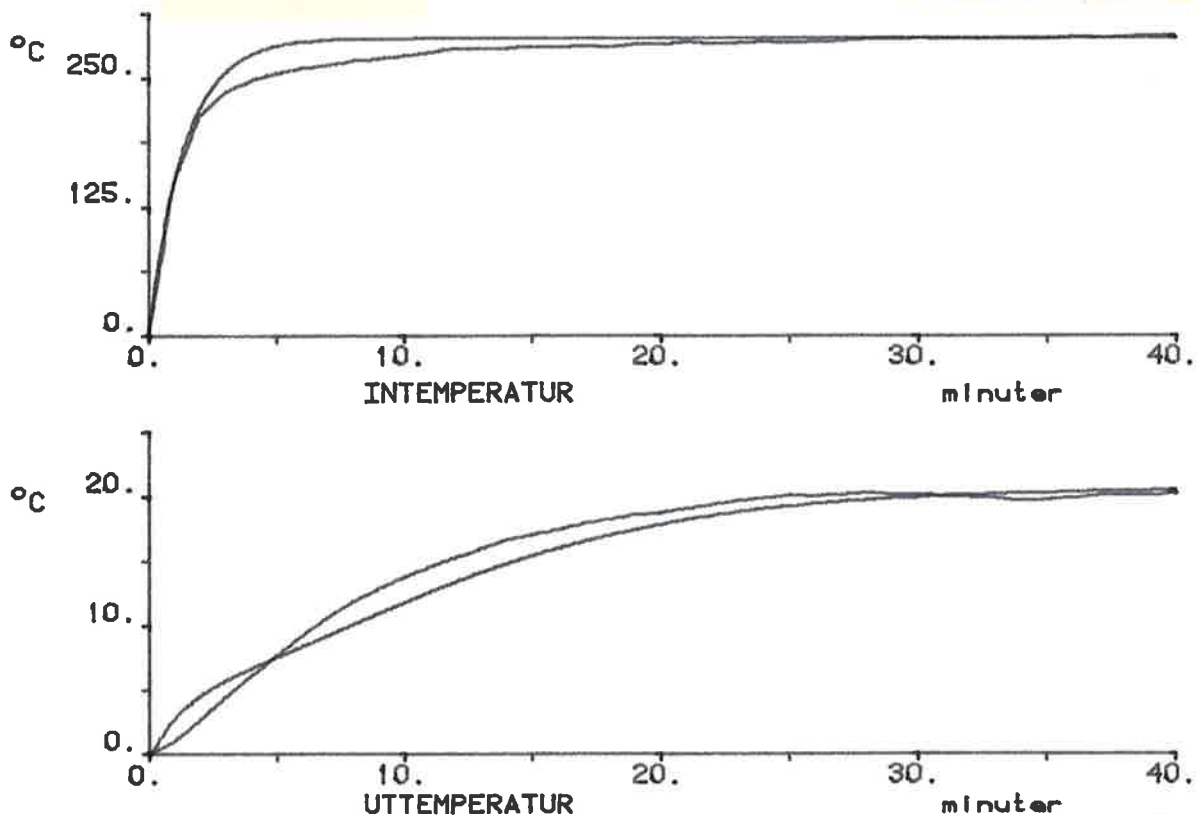


fig 2.6 En jämförelse mellan den verkliga processen (1) och den simulerade modellen (2)

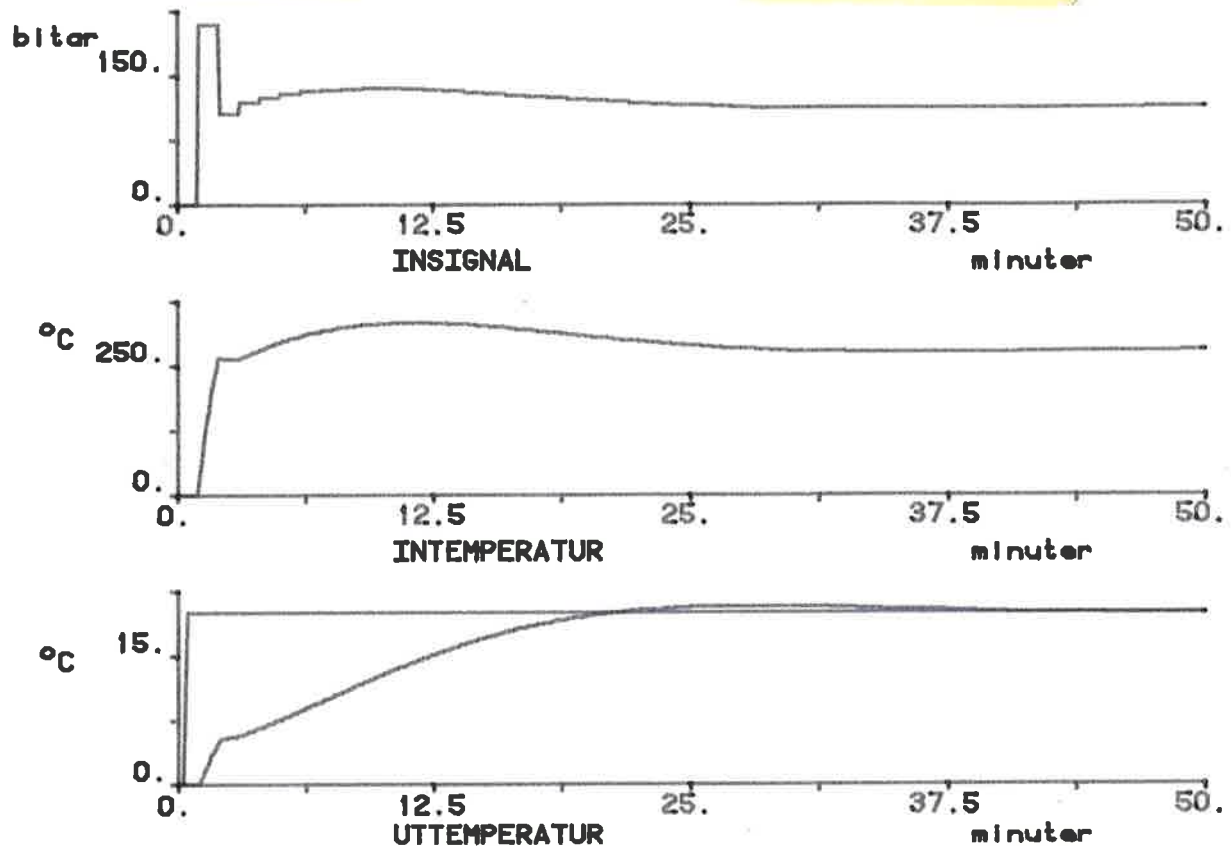


fig 2.7 Det simulerade systemets svar på en börvärdesändring $T_{ref} = 20^{\circ}\text{C}$

KAPITEL 3

Kap. 3 PROCESSDATORSYSTEM3.1 Inledning

PK-supras processdatorsystem är uppbyggt med en Nova minidator med 64 k primärminne med 16 bitars ordlängd. Till datorn finns anslutet en dubbel floppy-disc med vardera 512 k bytes kapacitet samt 4 terminaler. Programmet är av DDC-typ, vilket innebär att alla regulatorer finns inne i datorn. Programmeringsspråket är ett interpretativt processtyrningsspråk, IPCL, som egentligen utvecklats för reglering av klimatprocesser. Fördelen med ett interpretativt språk jämfört med ett språk som är baserat på kompilering till maskinkod är att man får en hög packningsgrad och en relativt liten skillnad mellan källkod och absolut kod. Detta innebär att det är relativt enkelt att ändra eller lägga in ny kod on-line. Eftersom datorsystemet inte använder massminnet on-line är detta ytterst viktigt. Nackdelen är att exekveringshastigheten blir lägre, men eftersom processerna på Supra är långsamma har detta mindre betydelse.

3.2 Operativsystemet

Primärminnets uppdelning på olika program, tabeller och IPCL-kod framgår av figuren på nästa sida. Bint innehåller definitioner för alla digitala ingångsmoduler. Tabellen består av: entrynummer, värde, rack, position, förgreningsmodul, modultyp och forcerat värde. Program DIGI har lägre prioritet än DIGO och ANAI och går så fort den får tillgång till förgreningsmodulen, dock bara en modul åt gången. För att DIGI inte ska få vänta alldeles för länge finns det en långsamhetsspärr på 10 ms.

DIGO har högre prioritet än DIGI och ANAI och jobbar 16 gånger varje halvsekund. Den jobbar i tre faser:

1. Ser efter om det finns modultyp med "alltid utställning".
2. Kollar om det finns ändringar, d.v.s. gammalt och nytt bitmönster stämmer ej överens.
3. Cyklisk genomgång som börjar där den förra cykliska genomgången slutade.

DIGO har även hand om de analoga utgångarna, vilka oftast faller under fall 2.

ANAI arbetar med ADCT och lägger resultaten i ARV-tabellen. ADCT består av 6 ord/värde som anger:

0. Fas och period för inläsning av mätvärde.
1. FGM nr, rack (AD-omvandlare), position.

MINNESDISPOSITION

EXEKUTIV									
DIGI PROGR	DIGO PROGR	ANAI PROGR	LARM PROGR	UTSKRIFTS- PROGR	GRÄNSVÄRDES- PROGR	IPCL- PROGR	UT-MOVE PROGR	OPERA- TÖRS KOM.- PROGR	
BINT: DIGITAL INGÅNGS TABELL	UTM: UTGÅNGS TABELL	ADCT: ANALOG INGÅNGS TABELL	LUST: LARM TABELL	CYKLISK UTSKRIFT- STACK	ARV: MÄT- VÄRDEN	LIMT: GRÄNS- VÄRDES TABELL	UT- TABELL	REGU- LATOR- PARA- METRAR	
				TEXT: TEXT- TABELL	BORV: BÖR- VÄRDEN	IPCL KOD			TID- KANALER
				UTSKRIFTS- TABELL					

KAPITEL 3

2. Kanal, rack (multiplexor), position.
3. Skalfaktor, filterfaktor, låggräns, höggräns, maxändring.
4. Inläst bitmönster.
5. Nollpunkt.

AD-omvandlingen tar 20 ms.

Skillnaden mellan inläsning av analoga och digitala ingångar är att endast en kanal läses in åt gången för analoga medan en hel modul läses vid digitala ingångar.

LARM-programmet arbetar med LUST. LUST's entry nr korresponderar direkt med BINT's entry nr. Varje sekund går LARM-programmet igenom BINT. När den påträffar en ändring börjar den räkna. När den har räknat färdigt överlämnas informationen till en utskriftsstack. Om 2 larm kommer samma sekund kommer de ut i nummerordning. Tidpunkten anges som den tidpunkt som de stoppas in i stacken. Denna information tar 4 ord: entry nr i LUST, terminaler, tid, datum. 75 sådana rymts i stacken som är cyklisk, d.v.s. de 75 senaste larmutskrifterna ligger där.

Terminalhanteringsprogrammen har över dygnet lite arbetstid men har hög prioritet p.g.a. att man inte vill sitta och vänta. Om dessa program kräver mycket tid finns risk att program med kort cykeltid inte hinner med, t.ex. DIGO.

IPCL-koden har lägst prioritet och får all tid som blir över. För att öka IPCL's tillgång till CPU:n kan man minska andra programs tillgänglighet.

3.3 Allmänt om IPCL

För att källkoden skall likna absolutkoden så mycket som möjligt och för att konverteringen skall bli enkel har alla paranteser utlämnats och all tilldelning sker från vänster till höger i källkoden, och uppifrån och nedåt i absolutkoden. Skillnaden mellan källkod och absolutkod är att den horisontella notationen ändras till den vertikala och för ett fåtal källkodsinstruktioner byter en del instruktioner plats. Några räknarceller läggs också till i vissa fall. Upprepad tilldelning är tillåten. Ett beräkningsuttryck får omfatta flera rader. Dubbla operationer är dock ej tillåtna.

All kod skrivs en tab in på raden. Alla operationer skall åtskiljas med blanktecken, kommatecken, vänster- eller högerparanteser. Hela rader med kommentarer kan läggas in genom att skriva ett C som första tecken på raden. IPCL-programmet börjar alltid med åtta kommentarrader och exekveringen avbryts när END påträffas i huvudprogrammet.

KAPITEL 3

Programmet får innehålla flera END.

Alla beräkningsparametrar läggs utanför IPCL-programmet i externa tabeller för att ändringar ska kunna ske enkelt. Ett undantag är vissa konstanter, som interpretatorn genererar från en instruktion. Man kan ange egna gränser för största index för standardvariablerna med en DIMENSION sats.

Exempel:

```
DIMENSION MV(100),PV(200)
```

Den normala programfilen IPCL.TB kompileras först med NIPCL.TB NIPCL IPCL.TB IPCL.SR

Inga fel får förekomma.

Därefter sker assemblering med definitionsfilen MIPCL.SR. ASM/L MIPCL/S IPCL .

Även här får inga fel förekomma. För att rätta fel on-line så skall man använda sig av listan IPCL.LS.

Vid den första exekveringen erhålls oftast en del systemfelutskrifter. Felutskrifter erhålles vid felaktigt index för de flesta indexerade variablerna, vid dubbla flyttalsoperationer och vid dubbla logiska operationer.

Standardvariabler:

<u>beteckning</u>	<u>max_index</u>	<u>klartext</u>	<u>tabell</u>
MV	2047	mätvärde	ARV
SV	2047	börvärde	BORV
PV	2047	parametervärde	REAL
NUM	511	heltal	INTT
IN	16383	ingång	BINT
AIN	511	analog ingång	BINT
OUT	16383	utgång	UTM
AOUT	511	analog utgång	UTM
IOUT	511	öka/minska utgång	UT
TCL	511	tidkanal	TKNT

KAPITEL 3

3.4 Operationer och Instruktioner

Språkets instruktioner är snarlika FORTRAN's instruktioner och operationer. P.g.a. dess speciella användningsområde förekommer det dock en del tillägg. Skillnader och väsentliga tillägg anges nedan (se även bilaga 2).

Eftersom dubbla operationer inte är tillåtna förekommer det några extra logiska operationer:

FAT	False and true
TAF	True and false
FOT	False or true
TOF	True or false

Exempel:

IPCL	FORTRAN
A TAF B	A .AND. .NOT. B

Tilldelning av en logisk variabel med ett logiskt uttryck sker med instruktionerna:

SET	tilldela
NET	negera och tilldela

Binära ingångar nollställs med instruktionerna STOP, ZERO, CLOSE, ISTART, IOPEN och ettställs med START, ONE, OPEN, ISTOP, ICLOSE. I står för inverterad.

Vid start och stopp av flera objekt vill man ofta att dessa sker med olika tidsintervall. Detta gör man med följande operationer:

DOFF(1-511)	Fördröjt frånslag
DON(1-511)	Fördröjt tillslag
DOFFON(1-511)	Fördröjt till- och frånslag

Maximal fördröjning är 511 sekunder.

Drift av olika processdelar kan styras med tidkanaler med instruktionen:

TCL(1-511)

där siffran anger entrynummer i tabellen TKNT, som i sin tur innehåller numret på en tidkanal. Instruktionen är en logisk variabel.

KAPITEL 3

Även instruktionen TIME(FAS,PER) är en logisk variabel som endast är sann då fasen stämmer med aktuell räknare, och period är rätt under en tidsenhet. PER är ett heltal 1-8 vilket motsvarar en samplingsperiod på respektive 5, 10, 20, 40 sek, 2, 5, 10 minuter och 1 timme. Fasen är i sekunder och måste vara mindre än perioden och mindre än 255. Instruktionen används med fördel tillsammans med en IF THEN ELSE-sats. En IF THEN ELSE-sats avslutas alltid med IFEND.

För att kunna hoppa i programmet finns följande satser:

GOTO LAB	ovillkorligt hopp
IF (LAC) CGOTO LAB	hopp om sann
IF (LAC) NCGOTO LAB	hopp om falsk

där LAB anger ett läge och LAC är ett logiskt uttryck eller innehållet i den logiska ackumulatorn. Hopp kan ske både uppåt och nedåt i programmet. Interpretatorn testar alltid att hoppadressen refererar till ett läge som finns i IPCL-koden.

Ställdon kan öppnas, stängas eller köras efter ett annat ställdons utgång.

Öka/minska utgång:

IOUT(1-511)	utgång nr
MAXIOUT(1-511)	maxutgång nr
MINIOUT(1-511)	minutgång nr

Indexet anger entrynummer i UT-tabellen.

Analog utgång:

AOUT(1-511)	utgång nr
MAXAOUT(1-511)	maxutgång nr
MINDOUT(1-511)	minutgång nr

Indexet anger entrynummer i UTM-tabellen.

För att köra ett absolut ställdon i ett visst läge eller för att köra ett integralställdon en viss sträcka används instruktionen:

NUM(1-511)

eller

IC(0-511)

För NUM så anger indexet entrynummer i INTT-tabellen. IC står för integer constant och ger ett heltal lika med angivet index.

Följande flyttalsvariabler finns:

MV(1-2047)	Mätvärde
SV(1-2047)	Börvärde
PV(1-2047)	Parametervärde
RC(0-511)	Reel konstant

RC ger ett flyttal lika med angivet index och kan ej tilldelas ett värde. 2 ord används för flyttalen.

För att omvandla flyttal till heltal och tvärtom, används följande symboler:

FLOAT

IFIX

När IFIX påträffas omvandlas flyttalsuttrycket till ett heltal och därefter sätts heltalsvariabeln i högerledet. Motsvarande gäller för FLOAT.

För att begränsa en variabel till ett givet intervall kan man använda sig av en max- och minfunktion. Dessa kan emellertid ersättas med funktionen MIX.

MIX(YMIN, Y, YMAX)

där YMIN och YMAX är gränserna och Y är variabeln som skall begränsas. Resultatet lagras i flyttalsackumulatorn.

För att begränsa en variabeländring används funktionen:

DMAX(CMAX, OLD, NEW)

Är skillnaden NEW-OLD absolut större än CMAX begränsas NEW, annars är NEW oförändrad. DMIN är omvändningen till DMAX, d.v.s. alltför små ändringar tillåts ej.

För att åstadkomma rampändringar av olika variabler används:

RAMP(DELTA, OLD, NEW)

Bortsett från funktionsnamnet är funktionen RAMP identisk med DMAX.

För att filtrera en variabel används funktionen

FILT(FF, OLD, NEW)

KAPITEL 3

Filtret är ett förenklat första ordningens filter som kan skrivas som

$$\text{NEW} + \text{FF} * (\text{OLD} - \text{NEW})$$

Resultatet lagras i flyttalsäckumulatorn.

För att klara av paranteser finns följande stackinstruktion:

FPOP	FPUSH	För flyttal
IPOP	IPUSH	För heltal
LPOP	LPUSH	För bitar

Mellanresultatet lagras med PUSH-instruktionen och hämtas med POP-instruktionen. Högst åtta mellanresultat får vara stackade i var stack. Alla stackarna nollställs var sekund, vilket medför att stacken skall tömmas varje exekvering. Observera att ordningsföljden är viktig. Vid osymmetriska operationer såsom FAT, TAF, FOT, TOF, -, och / är det lätt hänt att man får felaktigt resultat.

För att få en utskrift används funktionen:

$$\text{PRINT}(\text{DEV}, \text{NR})$$

där

DEV	utskriftsenheten	heltal 1-15
NR	utskriftsnummer	heltal 1-4095

Subrutiner kan användas i IPCL-koden. Anrop sker enligt följande:

$$\text{CALL TEST} (\quad)$$

där TEST är subrutinens namn. Inom parantesen anges antalet argument som ett heltal och de data som används av subrutinen. Subrutinen definieras genom att ange raden

$$\text{SUBROUTINE TEST} (\text{na}, \text{TYP}_1, \dots, \text{TYP}_{\text{na}})$$

Därefter följer koden. Alla operander som används anges med ARG(1-511) istället för de normala operanderna. Indexet anger vilket argument som skall läsas in eller läggas ut. Följande argumenttyper finns:

- 1 IN , OUT
- 2 TCL och grupp 1

KAPITEL 3

- 3 NUM, AIN, AOUT, IOUT
- 4 IC och grupp 3
- 5 MV, PV, SV
- 6 RC och grupp 5
- 0 övrigt

Udda typer anger tilldelningsbara variabler. Jämna typer anger icke tilldelningsbara variabler. Exekveringen av subrutinen avbrytes med RETURN.

Mer än ett hundratal assemblersubrutiner finns tillgängliga i biblioteket NIPCL.LB . Anrop sker med raden

CASM ANAME

följt av eventuella argument på samma rad. De assemblersubrutiner som skall användas och laddas med i systemet deklarereras med DASM som nedan

DASM rutin1, rutin2,.....rutinN

För att kunna kunna programmerar stora system med parallella processer kan programmet delas upp i olika underprogram, som exekveras med följande anrop:

RUN PNAME (PSN,PWN)

där

PSN programstatus

PWN programräknare, väntesatser

Interpretatorn klarar av max åtta nivåer av underprogram.

3.5 On-line programmering

Generellt gäller att: inmatning skall skrivas på samma sätt som utmatning, interna adresser i IPCL-program anges oktalt och relativt entrypointern IPCL externa adresser anges absolut och decimala heltal anges alltid utan decimalpunkt. Efter kommandoordet IPCL, skriver man den relativa programadress som skall undersökas. Genom upprepad LF (linefeed) listas programmet vertikalt. Avslutning sker med CR (carriage return). ESC (alt mode) ger backning en cell

KAPITEL 3

vid en tom rad och vid inmatning förändras ej läget. Detta kan utnyttjas för att kontrollera en inmatning. Vid backning kan tolkningen bli felaktig, därför är det viktigt att start sker från rätt ställe. Utmatning består av en rad med IPCL relativ adress, oktala kod, klartext och värdet. För flerordsinstruktioner skrivs endast den oktala koden för den första cellen. Inmatning kan ske efter att utmatning skett. Flera instruktioner eller hela uttryck skrives på samma rad. Mellan varje instruktion krävs ett mellanslag. Om raden avslutas med CR kan inmatning fortsätta omedelbart efter den nyinlagda koden.

Antag att utmatning består av m ord IPCL kod och som börjar på cell n har skrivits ut. Antag att inmatningen består av p ord IPCL kod.

- | | |
|----------------|----------------|
| 1. tom rad CR | ger uthopp |
| 2. tom rad LF | ger IPCL $n+m$ |
| 3. tom rad ESC | ger IPCL $n-1$ |
| 4. kod CR | ger IPCL $n+m$ |
| 5. kod LF | ger IPCL $n+p$ |
| 6. kod ESC | ger IPCL n |

3.6 Reglering

Start och stopp av en regulator sker genom att man lägger regulatoranropet i IPCL-programmet. Regulatoranropet är följande:

REG(1-511)

Indexet anger numret på reglarentryt. Alla regulatorer beräknas av ett program PREG, som är underordnat programmet PIPCL. Ett reglarentry innehåller följande:

Regulatortyp
 Utsignaltyp
 DYMIX
 DUMIX
 UMIK
 D1
 D2
 D3
 D4
 P1
 P2
 P3

Regulatortyp

anges med ett tal 0-15 vilka betyder:

1	PD
2	PI
3	D
4	P
5	DP
6	I
7	MAN
8	självinställande PI
9	PID
10	till förfogande
11	Q kvotregulator
12	DQ deltakvotregulator
13	DS dynamiskt system
14	SM summering
15	ESM utvidgad summering

Utsignaltyp

anges med ett tal 0-7. Följande finnes:

kod	beteckning	förklaring
1	UK	kaskadutgång
2	UI	öka/minska utgång
3	UP	pulslängdmodulerad utgång
4	UA	analog utgång
5	UB	analog utgång (backup)
6	US	stegutgång

DYMIX

Anger ett av sju flyttal, som används som dödzon. Vid ett reglerfel mindre än dödزونen sker ingen ändring av utsignalen.

DUMIX

Anger ett av sju flyttalspar, som anger den minsta respektive den största utsignaländring som tillåts.

UMIX

Anger ett av sju flyttalspar, som anger utsignalens definitionsområde. Denna skall alltid användas vid kaskadregulatorer, då en eventuell integration sker i regulatorns utsignal.

D1

Anger det värde som skall regleras, vilket kan anges på samma sätt som vid gränsvärdesövervakningen. Bokstaven M för ärvärde och B för börvärde följt av ett heltal utan mellanslag.

D2

Anger det värde som regulatorn skall följa (börvärde). Anges

KAPITEL 3

på samma sätt som för D1.

D3

Används av backupregulatorn för att hämta in aktuell utsignal från modul. Denna parameter används för att definiera stegutsignaler.

D4

Anger utsignalens nummer. Dess typ anges av parametern UT. För kaskadregulatorn anges D4 som för D1 och D2. Öka/minska utgångar och analoga utgångar anges bara som ett heltal (entry nr i UTM). För stegutsignaler så anges den första utsignalen.

P1,P2,P3

För PID-regulatorn gäller

P1	Förstärkning
P2	Integrationskonstant
P3	Derivationskonstant

Efter börvärdet lagras tre flyttal vilka är:

reglerfelet	$e(t-1)$
föregående mätvärde	$y(t-1)$
förföregående mätvärde	$y(t-2)$

Detta innebär att de tre efterföljande börvärdena måste vara lediga. Detta har gjorts för att spara kärnminne, då de flesta reglerfall klaras av med PI-reglering.

För regulatorerna PD, D, P, PI, I och DP så gäller att

P1	Förstärkning
P2	Derivationskonstant/integrationskonstant
P3	Mellanresultatet $e(t-1)$

För kvotregulatorn gäller att

P1	dynamisk kompensering
P2	relaxation
P3	mellanresultatet $e(t-1)$

O<P1<1
O<P2<1

PID-regulatorer

$e(t)$	reglerfel	$(y - y_{\text{bör}})$
$u(t)$	utsignal	(styrsignal)
$y(t)$	insignal	(ärvärde)
$y_{\text{bör}}$	insignal	(börvärde)

KAPITEL 3

P-regulatorn

$$u(t) = P_1 * e(t)$$

D-regulatorn

$$u(t) = P_1 * (e(t) - e(t-1))$$

PD-regulatorn

$$u(t) = P_1 * (e(t) + P_2 * (e(t) - e(t-1)))$$

I-regulatorn

$$u(t) = u(t-1) + P_1 * e(t)$$

PI-regulatorn

$$u(t) = u(t-1) + P_1 * (e(t) - e(t-1) + e(t) / P_2)$$

DP-regulatorn

$$u(t) = u(t-1) + P_1 * (e(t) - e(t-1))$$

PID-regulatorn

$$u(t) = u(t-1) + P_1 * (e(t) - e(t-1) + e(t) / P_2 + \\ + P_3 * (y(t) - 2y(t-1) + y(t-2)))$$

Observera att integraltid och derivationstid är angivna i tidsenheten samplingsintervall. Samplingsintervallet anges inte i reglertabellen REG, utan bestäms av exekveringssättet i IPCL-programmet.

Självinställande PI-regulatorer

Dessa regulatorer kan användas för att finna lämpliga parametrar till en PI-regulator med fixa parametrar. De kan också användas för att reglera processer vars dynamik varierar så att en regulator med fixa parametrar ej kan användas.

Först beräknas två modellparametrar till en första ordningens modell. Därefter dimensioneras en PI-regulator så att det slutna systemet erhåller en dubbelrot eller endast en enkelrot med önskat värde. Vilka reglerparametrar som används och hur ofta de eventuellt skall ändras, styrs av användaren. Användaren skall ange fyra flyttalsvariabler, komihågfaktorn RF, transformationsparametrarna PK och PL och ändringsperioden PP som anger antalet samplingsintervall mellan varje uppdatering.

De självinställande PI-regulatorerna arbetar med 19 extra

KAPITEL 3

flyttalsvariabler, som läses efter börvärdet som anges med regulatorparametern D2.

kvotregulatorer

Kvotregulatorn beräknas enligt algoritmen:

$$u(t-1) = P_1 * u(t-2) + (1-P_1) * u(t-1)$$

$$u(t) = P_2 * u(t-1) + (1-P_2) * u(t-1) * y_{\text{bör}} / y(t)$$

deltakvotregulatorn

Denna beräknas i två steg:

Förstärkningen: $g(t) = P_1 * g(t-1) + (1-P_1) * \Delta u(t-1) / \Delta y(t)$

$$u(t) = -g(t) * (y(t) - y_{\text{bör}})$$

3.7 Programmet

Vid övergång från handstation till automatik, d.v.s. till regulatorn, kan integralen ha ett sådant värde att utsignalen gör ett språng, om man inte vidtager speciella åtgärder. PI-regulatorn ser ut på följande sätt:

$$u(t+1) = u(t) + P1(e(t) - e(t-1) + e(t) / P2)$$

där P1 är förstärkningen och P2 är integraltiden uttryckt i samplingsintervall. I vårt fall är utsignalen börvärde till P-regulatorn, och P-regulatorns utsignal styr oljeventilen. Denna blir:

$$u(t) = (T - Y_{\text{ref}}(t)) * K$$

T är torktrummans intemperatur och Yref är PI-regulatorns utsignal. I handläge får man då att:

$$y_{\text{ref}}(t) = T - u(t) / K$$

Instoppat i IPCL-koden uppdaterar man på så sätt regulatorutgången i handläge. Ett problem kvarstår dock; i PI-regulatorn läggs det föregående felet $e(t-1)$ i P3 och denna kan man inte nå inifrån IPCL-koden. I handläge är ju inte regulatorn inkopplad och man har ett gammalt, eventuellt stort, fel i P3. För att få det gamla felet = 0 får man sätta börvärdet lika med ärvärdet och göra en skenreglering, d.v.s. låta regulatorn arbeta men inte ställa ut svaret.

KAPITEL 3

Regulatorprogrammet

```

C TC240 TEMPERATURREGLERING BRÄNNARE
C REGULATORN STARTAS MED IN1211 = 1 OCH STOPPAS MED
C IN1211 = 0
C BÖRVÄRDET RAMPAS OM IN1212 = 1
C IN1212 ANVÄNDS FÖR TEST OM NY START

```

```

CD SV(809)          BÖRVÄRDE TC240 GRADER CELSIUS
CD SV(810)          STEGHÖJD RAMP
CD SV(802)          UTSIGNAL FRÅN RAMP = INSIGNAL TILL REG 22
CD SV(808)          UTSIGNAL FRÅN REG 22 = INSIGNAL TILL REG 43
CD SV(819)          PARAMETERVÄRDE: P1 I REG 43 MED OMBYTT TECKEN
                    1
CD SV(811)          UTSIGNAL REG 43
CD MV(74)           ÄRVÄRDE INTEMPERATUR
CD MV(75)           ÄRVÄRDE UTTEMPERATUR
CD MV(63)           ÅTERLÅST UTSIGNAL HANDSTATION
C IN(1211)          START REGULATOR OCH RAMP
C IN(961)           INDIKERING HANDSTATION
C IN(1212)          FIKTIV NYSTART
C AOUT(66)         UTENTRY

```

```

CD REG(22)
CD PI+UK+113      M75      B802      0      B808      -31.0      8.5      0.0
CD REG(43)
CD P+UK+114      M74      B808      0      B811      -0.3      0.0      0.0

```

```

C OM NY START SÄTT BÖRVÄRDET = ÄRVÄRDET

```

```

IF IN(1211) TAF IN(1212) THEN
  MV(75) = SV(802)
IFEND
IN(1211) SET IN(1212)

```

```

C OM HANDSTATION LIGGER I HAND UPPDATERA REGULATORUTGANG.
C VID ÖVERGANG FRÅN HAND TILL AUTO LÄGGS AKTUELLT MÄTVÄRDE
C SOM STARTVÄRDE FÖR BÖRVÄRDES RAMPEN.

```

```

IF NOT IN(961) THEN
  MV(75) = SV(802)
  REG(22)
  MV(63) = SV(811) IFIX AOUT(66)
  SV(811)/SV(819) + MV(74) = SV(808)
ELSE

```

```

ANNARS RAMP OCH REGLERA OM IN1211 = 1 ANNARS STÅNG VENTIL

```

```

IF TIME(1,4) THEN
  IF IN(1211) THEN
    RAMP(SV(810),SV(802),SV(809)) = SV(802)

```

KAPITEL 3

```

                REG(22)
                REG(43)
                SV(811) IFIX AOUT(66)
            ELSE
                RC(0) = SV(811)
                MINAOUT(66)
            IFEND
        IFEND
    IFEND

```

DUMIX, UMIK och DYMIX har satts så de inte begränsar regleringen. DUMIX1 = (0,1000), DYMIX1 = (54,255), UMIK3 = (0,1500), UMIK4 = (0,1000)

Mellan 5 och 10 samplingsintervall är lagom för ett stegsvar. Därför valdes samplingsintervallet till en minut. TIME(1,4) är sann den första sekunden i varje minut, d.v.s. samplingsintervallet är 1 minut.

Mätvärdena M74 och M75 samplas var tionde sekund med en filterfaktor på 0.8, d.v.s.

Mätvärde = gammalt värde + 0.2*(nytt värde - gammalt värde)

Filtret måste man ha för att inte ett felaktigt värde ska förstöra regleringen. Eftersom filtret varit detsamma under alla mätningar och tester, har det ingen praktisk inverkan på regleringen. Viktigt att notera är att om man ändrar parametern p_1 i REG 43 måste man även ändra B819 till p_1 med

ombytt tecken. Minustecknet i p_1 beror på att regulatorn

beräknar felet $e(t)$ som $y - y_{bör}$, vilket kan tyckas vara lite

bakvänt.

KAPITEL 4

Kap 4 UTVÄRDERING

Programmet infördes i IPCL-koden. Regulatorparametrarna var $K = 8.5$ och $T_i = 2$ min för PI-regulatorn och $K = 0.55^\circ\text{C}^{-1}$ för P-regulatorn. ⁱ Då börvärdet ändrades från 110°C till 114°C fick man ett resultat enligt fig 4.1.

Stegsvaret är alldeles för långsamt och förstärkningen borde varit högre.

Vid en börvärdesändring blir utsignaländringen, under förutsättning att både in- och uttemperaturerna har varit samma det senaste samplingsintervallet,

$$K_P * K_{PI} * (1 + 1/T_i)$$

per grad börvärdesändring.

För första försöket var den:

$$0.55 * 8.5 * (1 + 1/2) = 7.0125^\circ\text{C}^{-1}$$

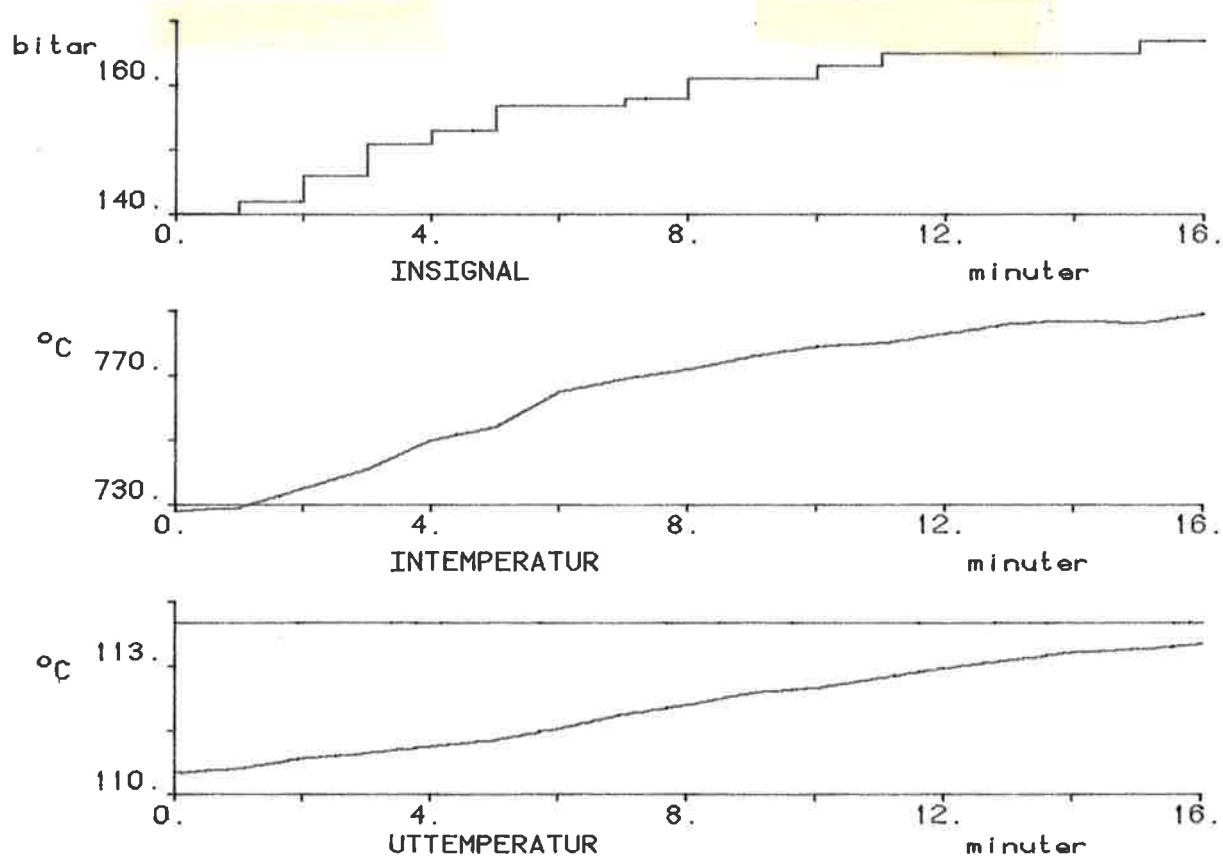


fig 4.1 Börvärdesändring med $K_{PI} = 8.5$, $K_P = 0.55^\circ\text{C}^{-1}$, $T_i = 2$ min

KAPITEL 4

Parametrarna ändrades till 14.0 och 1.8 för PI-regulatorn och $K_p = 0.4$ vilket ger en utsignaländring på 8.71 per grad

börvärdesändring. Ett steg gjordes. Systemet kom i svängning och var på gränsen till instabilitet.

Ett tredje försök gjordes med $K_{PI} = 11.0$ $T_i = 1.5$ min och $K_p = 0.45^\circ\text{C}^{-1}$ d.v.s. en lite lägre förstärkning:

$$0.45 * 11 * (1+1/1.5) = 8.25^\circ\text{C}^{-1}$$

Stegsvaret syns i figur 4.2. Systemet svängde fortfarande alltför mycket. Felet antogs vara integraltiden som var för låg. Integraltiden ändrades till 8.5 min och $K_{PI} = 22$ vilket

ger:

$$0.45 * 22 * (1+1/8.5) = 11.06^\circ\text{C}^{-1}$$

Svängningarna har nästan helt försvunnit fastän förstärkningen är kraftigare. Stegsvaret börjar se hyfsat ut, men temperaturen sjunker sakta efter ett tag.

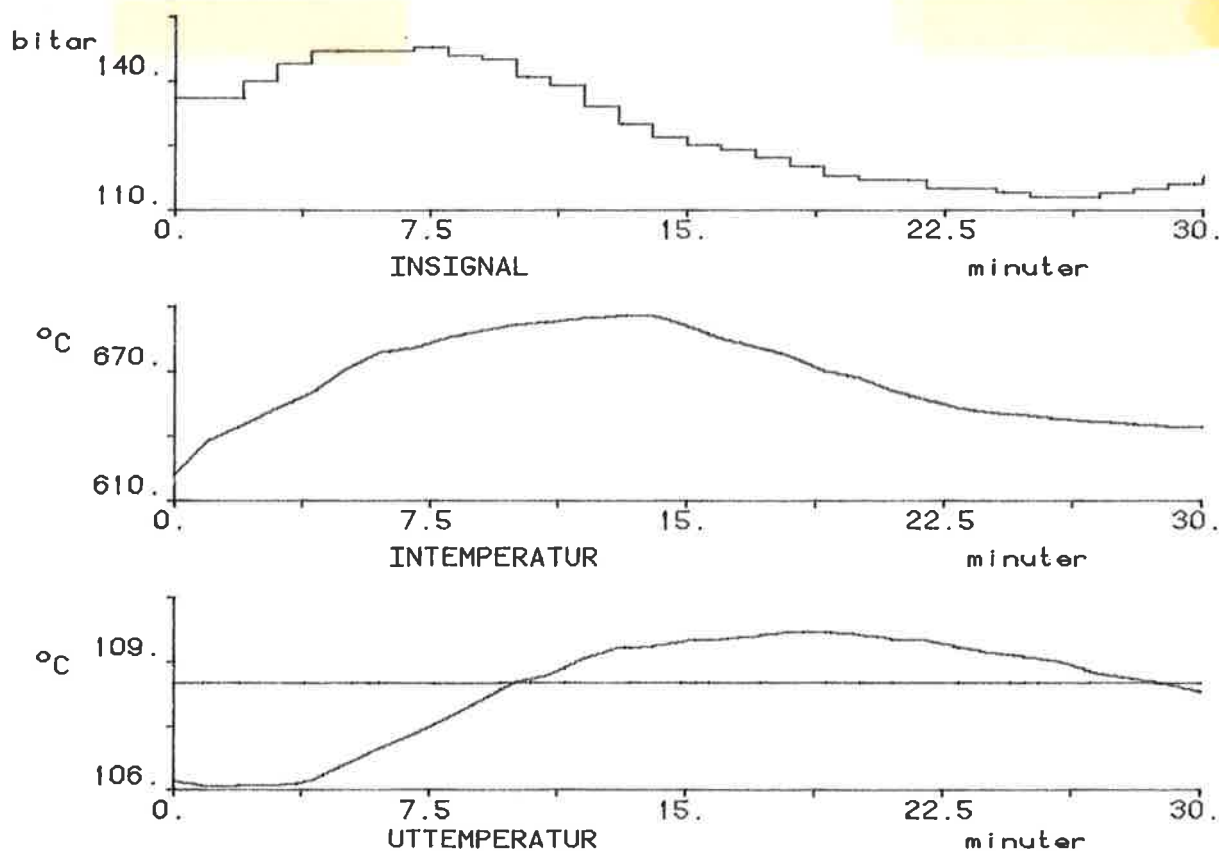


fig 4.2 Börvärdesändring då $K_{PI} = 11.0$, $T_i = 1.5$ min, $K_p = 0.45^\circ\text{C}^{-1}$

KAPITEL 4

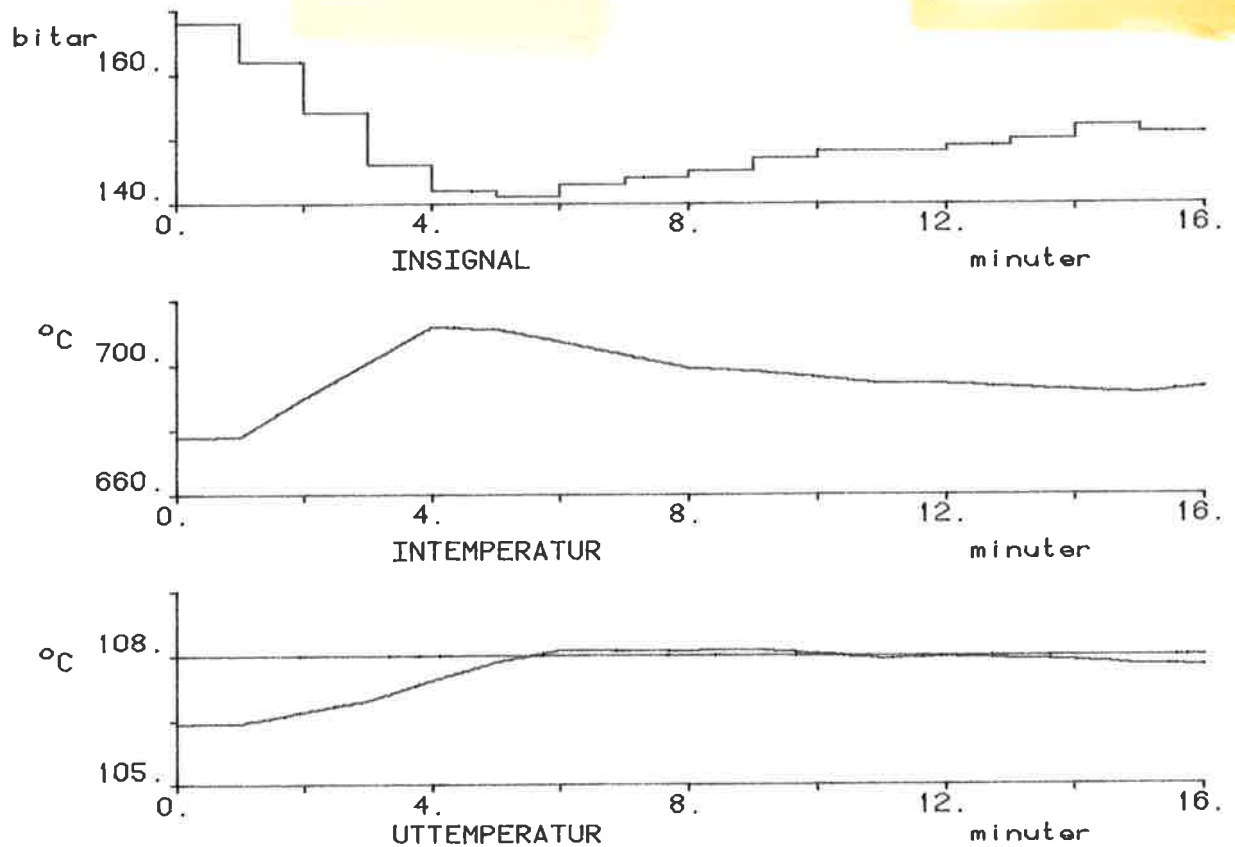


Fig 4.3 Börvärdesändring då $K_{PI} = 22.0$, $T_i = 8.5$ min, $K_P = 0.45^\circ\text{C}^{-1}$

Intemperaturen stiger till ett toppvärde efter ca 4 minuter för att därefter minska alldeles för mycket. Problemet avhjälptes genom att minska utsignalens beroende av intemperaturen, d.v.s. minska K_P . K_P minskades först till

0.4 och därefter till 0.3. Samtidigt måste ju K_{PI} ökas i

ungefär motsvarande grad. Den slutliga inställningen blev:

$$K_{PI} = 31.0; T_i = 8.5 \text{ min}; K_P = 0.3^\circ\text{C}^{-1}$$

vilket ger:

$$0.3 * 31.0 * (1 + 1/8.5) = 10.4^\circ\text{C}^{-1}$$

Regulatorns primära uppgift är ju inte att följa ideliga börvärdesändringar så bra som möjligt, utan att vid kontinuerlig drift hålla uttemperaturen så nära ett och samma börvärde under längre tid. Därför gjordes det mätningar när regulatorn försökte hålla samma temperatur hela tiden. Den största avvikelser som uppmättes var $\pm 2^\circ\text{C}$ i uttemperaturen, då processen gick normalt. En mätning i

KAPITEL 4

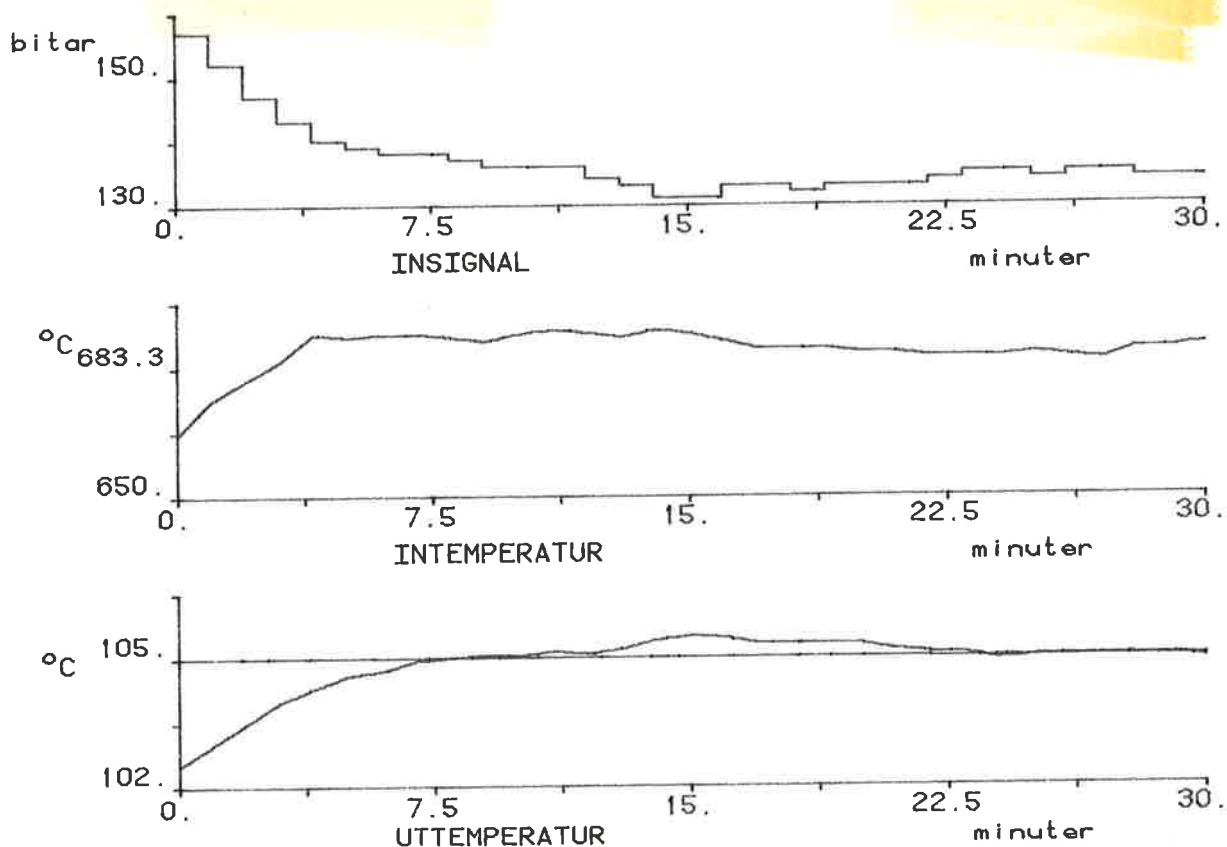


fig 4.4 Börvärdesändring med $K_{PI} = 31.0$, $T_i = 8.5$ min, $K_P = 0.3^\circ\text{C}^{-1}$

kontinuerlig drift ses i fig 4.5. Observera den annorlunda tidsskalan. Som jämförelse har temperaturvariationen utan regulatorn inkopplad (d.v.s. oljebrännaren i handläge) ritats upp i fig 4.6. Observeras bör att oljebrännarens ventilläge har ändrats endast en gång under 7.5 timmar. Temperaturen svänger då ca $\pm 4^\circ\text{C}$. Att uttemperaturen är jämnare med regulatorn inkopplad är alltså helt uppenbart.

Slutresultatet visar att parameterframtagningen med hjälp av SIMNON var klart bättre än med hjälp av en stegvarsmetod även om den fysikaliska modellen var grov. En noggrannare modell hade i det här fallet knappast varit meningsfull; i så fall skulle man använt en annorlunda regulator där man utnyttjat den kunskap man fått av modellen, men frågan är om det hade varit av någon större praktisk nytta.

Eftersom även andra faktorer spelar in; produktionsmängd, receptbyte, råvarornas skiftande kvalitet och andra processtörningar, är det svårt att dra någon generell slutsats om huruvida fukthalten i den färdiga produkten har blivit jämnare. Men genom att ha om inte eliminerat så åtminstone minskat inverkan av en av processtörningarna

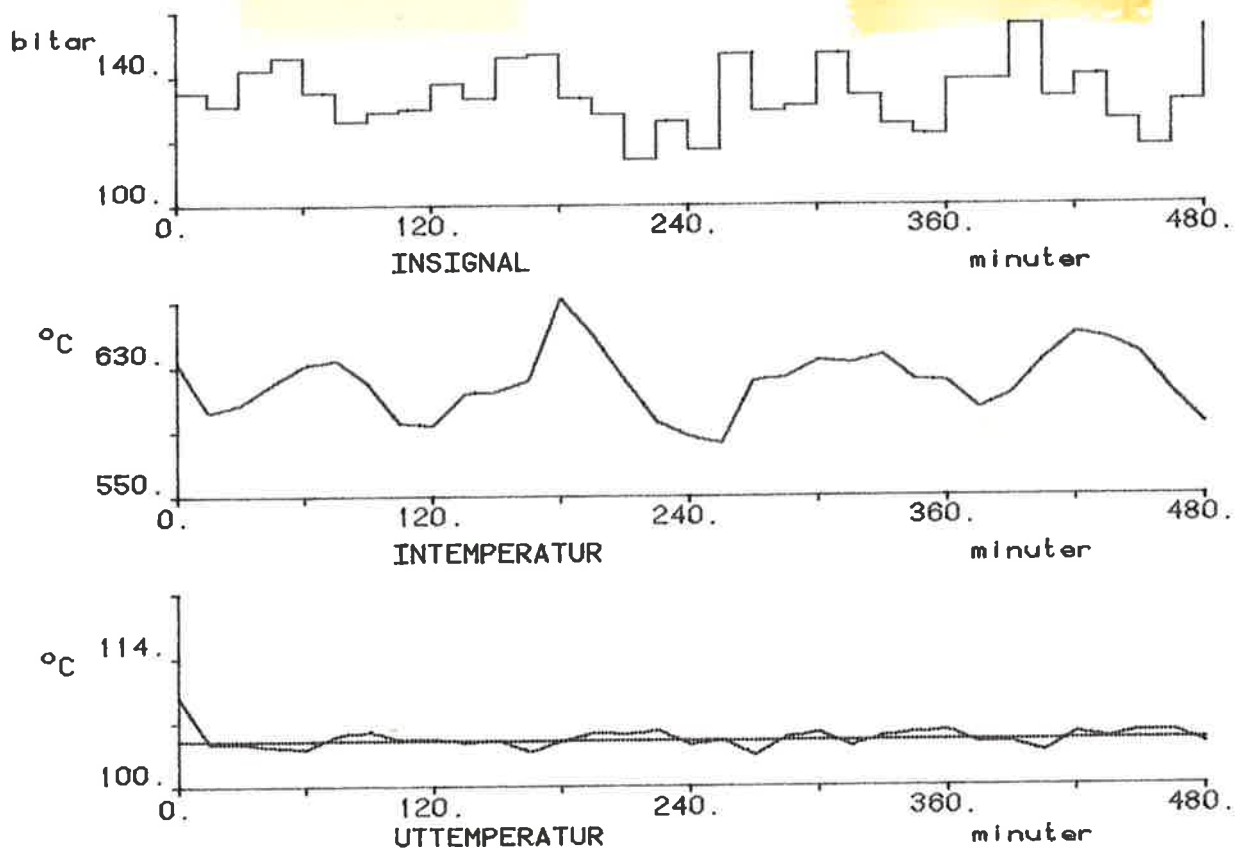


fig 4.5 Mätning i kontinuerlig drift med
 $K_{PI} = 31.0$, $T_i = 8.5 \text{ min}$, $K_p = 0.3^\circ\text{C}^{-1}$

föreligger större möjligheter att få ut en bättre produkt med jämnare fukthalt.

KAPITEL 4

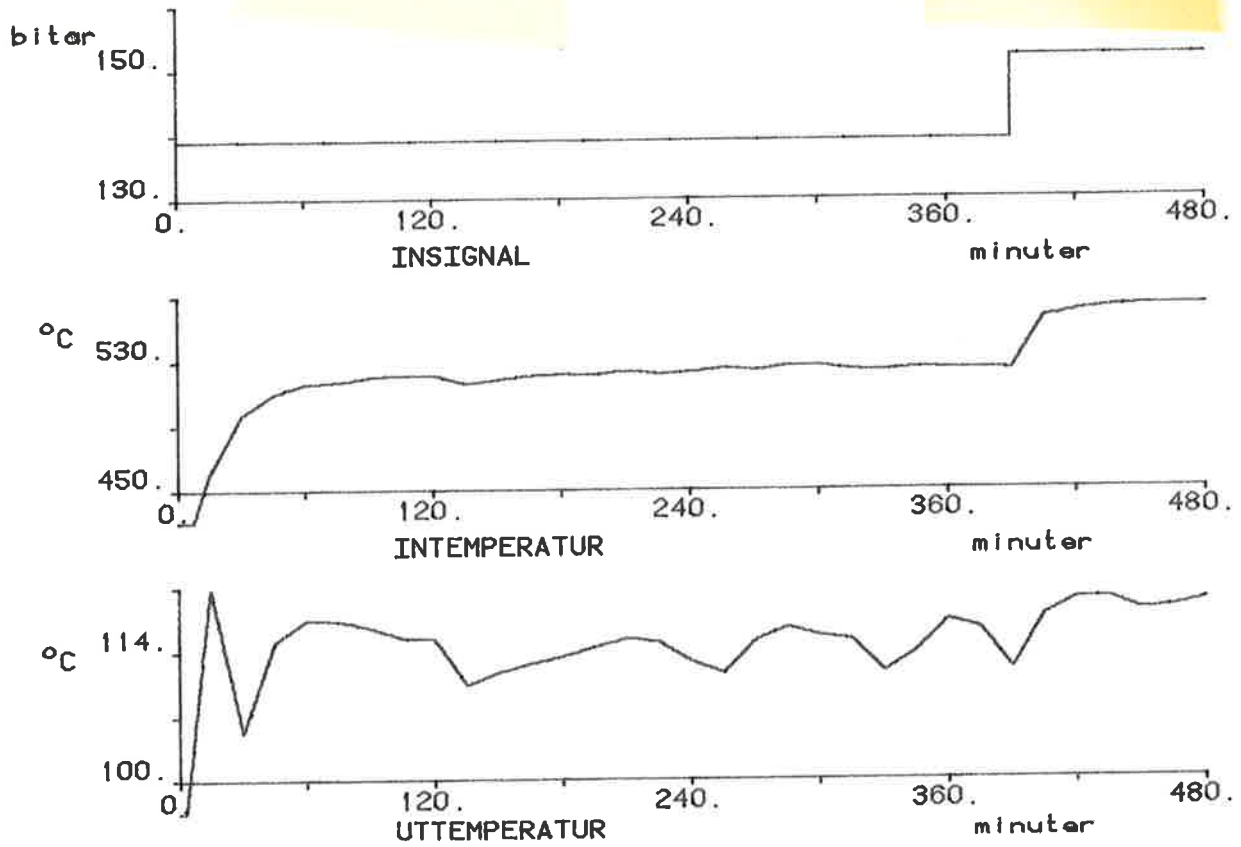


fig 4.6 Mätning i kontinuerlig drift med manuell temperaturreglering.

REFERENSER

Jensen Lars. :IPCL ett interpretativt processtyrningspråk.

Jensen Lars. :Reglering med DDC-6.

Aström K.J. (1982):Reglerteknik - en elementär introduktion.
Kapitel 5. PID reglering.
CODEN:LUTFD2/(TFRT-3166)/1-050/(1982)

Appendix_1

syntax_för_flerordsinstruktioner

För att kunna ändra ett IPCL-program on-line så är det viktigt att känna till vilka instruktioner som ej översätts direkt eller som blir flerordsinstruktioner. De celler som ändras har strukits under.

<u>Källkod</u>	<u>Absolutkod</u>
TIME(X,Y)	TIME X,B7+Y
RUN PNAME(PSN,PWN)	RUN PNAME PSN. PWN.
EXIT(ns)	EXIT ns.
CEXIT(ns)	som EXIT
NCEXIT(ns)	" "
INIT(ns)	" "
CINIT(ns)	" "
NCINIT(ns)	" "
WAIT(ns,vt)	WAIT ns. vt.
CEWAIT(ns,vt)	som WAIT
NCEWAIT(ns,vt)	" "
DOFF(X)	DOFF+X
DON(X)	som DOFF
DOFFON(X)	" "
PRINT (dev,nr)	PRINT dev.B3+nr
SQRT(A,B)	A SQRT B
FILT(A,B,C)	A B FILT C
MIX(A,B,C)	som FILT
DMIN(A,B,C)	" "
DMAX(A,B,C)	" "
RAMP(A,B,C)	" "
VAR(I)	ARRAY VAR I
VAR(I,J,K)	MATRIX VAR I J K

Appendix_2Instruktionsuppsättning_IPCL

Logiska variabler

IN(1-16383)	ingång
OUT(1-16383)	utgång
TCL(1-511)	tidkanal ej tilldelning
TIME(fas,per)	tidspuls ej tilldelning

Heltalsvariabler

AIN(1-511)	analogingång
AOUT(1-511)	analogutgång
MAXAOUT(1-511)	" " maxställ
MINAOUT(1-511)	" " minställ
IOUT(1-511)	öka/minska utgång
MAXIOUT(1-511)	" " maxställ
MINIOUT(1-511)	" " minställ
NUM(1-511)	heltal

Heltalskonstant

IC(0-511)	heltal ej tilldelning
-----------	-----------------------

Flyttalsvariabler

MV(1-2047)	mätvärde
SV(1-2047)	börvärde
PV(1-2047)	parametervärde

Flyttalskonstant

RC(0-511)	flyttal ej tilldelning
-----------	------------------------

Regulatoranrop

REG(1-511)	regulatoranrop
------------	----------------

Utskriftsinstruktion

PRINT(1-15,1-4095)	
--------------------	--

Logiska operationer

		0 0	1 0	0 1	1 1
AND	och	0	0	0	1
NAND	icke och	1	1	1	0
OR	eller	0	1	1	1
NOR	icke eller	1	0	0	0
EQ	lika	1	0	0	1
NE	icke lika	0	1	1	0
FAT	falsk och sann	0	0	1	0
TAF	sann och falsk	0	1	0	0

FOT	falsk eller sann	1	0	1	1
TOF	sann eller falsk	1	1	0	1
NOT	negera	1	1	0	1

Logisk tilldelning

SET	tilldela
NET	negera och tilldela
DOFF(1-511)	fördröjt fränslag
DON(1-511)	" tillslag
DOFFON(1-511)	" till- och fränslag
IFFTF	if false force to false else auto
IFFTT	if false force to true else auto
ITFTF	if true force to false else auto
ITFTT	if true force to true else auto
AUTO	forcera till auto
FOFF	forcera till från
FON	forcera till till
STOP, ZERO, CLOSE, ISTART, IOPEN	nollställning
START, ONE, OPEN, ISTOP, ICLOSE	ettställning

Hoppsatser och villkorssatser

GOTO LAB	ovillkorligt hopp
IF (LAC) GOTO LAB	hopp om sann
IF (LAC) NOGOTO LAB	hopp om falsk
(IAC) SGOTO ATAB	beräknad hoppsats
IF (LAC) THEN kod IFEND	
IF (LAC) THEN kod ELSE kod IFEND	

Heltalsoperationer

INC	öka med ett
DEC	minska med ett
MAXA	tilldela 255 till nästa heltal
MINA	tilldela 0 till nästa heltal
MAXI	tilldela 16384 till nästa heltal
MINI	tilldela -16384 till nästa heltal
INVERT	byter tecken vid heltalstilldelning
MASK	utför "och" mellan två heltals bitmönster
SHIFT	Vänster operand skiftas åt vänster vid positiv högeroperand.

Heltals- och flyttalsoperationer

*	multiplikation
/	division
+	addition
-	subtraktion
=	tilldelning (från vänster mot höger)
MAX	maxoperation
MIN	minoperation
ABS	absolutbelopp
NABS	absolutbelopp med minustecken

Omvandlingsfunktioner

FLOAT tilldelning av heltalsvärde till flyttal
 IFIX tilldelning av flyttalsvärde till heltal

speciella funktioner

SQRT(A,B) kvadratrotsiteration
 MIX(A,B,C) minmaxoperation
 FILT(A,B,C) filtrering
 DMAX(A,B,C) maxändringsbegränsning
 DMIN(A,B,C) minändringsbegränsning
 RAMP(A,B,C) maxändringsbegränsning

relationsoperationer

RGE större än eller lika med
 RGT större än
 RLE mindre än eller lika med
 RLT mindre än
 REQ lika
 RNE olika

stackoperationer

FPUSH float push
 FPOP float pop
 IPUSH integer push
 IPOP integer pop
 LPUSH logical push
 LPOP logical pop

DO-loops

DO (IAC)=I1,I2,I3 loop index är ett heltal
 I1 start index
 I2 stop index
 I3 index ökning
 DOEND slut på DO-loop

programkontroll

RUN PNAME(PSN,PWN) anrop av subprogram PNAME
 PROGRAM PNAME börja subprogra PNAME
 PSN subprogram status number
 PWN subprogram wait number
 END subprogram slut status oförändrad

EXIT

Deklarationer

DASM assembler subrutiner

DIMENSION	största dimensioner
ENTRY	internt läge som refereras externt
EXTERNAL	externt läge som refereras internt
TITLE	programdelsnamn för mapp

Appendix 3
Operatörskommando

För att inmatning av kommando ska kunna ske måste datorns operatörsprogram aktiveras, vilket sker genom inknappning av Ctrl/A. Datorn svarar då med >, varefter kommandot kan börja knappas in.

<u>Kommando</u>	<u>Betydelse</u>
ADCT	Analog till digital omvandlingstabell
ADFF	Fiterfaktorer till AD-omvandlingen
ADHÖG	Höggränser till AD-omvandlingen
ADLAG	Låggränser till AD-omvandlingen
ADMAX	Maxändringar till AD-omvandlingen
ADSKF	Skalfaktorer till AD-omvandlingen
AVBLOCKERA	Avblockering av utskrifter
B	Börvärde
BINT	Definitioner för digitala ingångar
BLOCKERA	Utskrift sker ej automatiskt
BREAK	Stopp av larm och tillståndsutskrifter
BÖRVARDE	Börvärde
COPY	Kopierar från disc i fack 0 till disc i fack 1
DATA	Tabell med texter som ingår i larmutskrifter
DATUM	Inmatning av datum
DEB	Lämna DDC-6 och gå till debuggern
DRFT	Drifttidsräkningstabell
DUMIX	Max och min utsignaländring
DUMP	Dump av tabellerna till diskette
DYMIX	Dödzon
HELG	Förändra typ av veckodag under vissa perioder
HELP	Inhibering av behörighetskontroll

I	Ingång
ID	ID-begrepp. Korta textsträngar
INGANG	Ingång
INTT	Heltalsvärde som används av IPCL
IPCL	Kontroll och ev. ändring av IPCL-kod
KLOCKSLAG	Inmatning av klockslag
KVITTERA	Kvittering av utskrifter
LIMT	Gränsvärdesövervakningstabell
LOAD	Laddning av minne
LUOD	Tabell för beskrivning av larmutskriftsterminale
LURT	Räknartabell till larmen
LUST	Larmutskriftstabell
LUTB	Tidsbeskrivningar till larmen
M	Mätvärde
MUTSK	Start av ett flertal utskrifter
MÄTVARDE	Mätvärde
O	Ändring av minnestabell oktalt
PDUMP	Dump av programmet till diskette
PERUT	Igångsättning av periodiska utskrifter
PLTB	Plottningstabell
PLOT	Igångsättning av plottning
PUTB	Periodisk utskriftstabell
REG	Regulatorstabell
S	Start av de olika programmen
STATE	Utskrift av intern programstatusinformation
TEXT	Textsträngar för larm- och tillståndskrifter
TID	Utskrift av datum och klockslag

TILT	Beskriver utskrifternas utseende
TKNT	Tidkanalnummertabell
TTTT	Tidkanaltabell
U	Utgång
UMIX	Max och min utsignal
UT	Utställningstabell
UTGÅNG	Utgång
UTM	Definitioner av digitala utgångsmoduler
UTSKRIFT	Start av tillstånds eller larmutskrift
VDAG	Utskrift eller inmatning av veckodag
VERIFIERA	Verifiering av utskrifter

Appendix 4

APPENDIX_4Program_i_SIMNON_för_simulering_och_figurgenerering

```

macro figgen
"macro för generering av alla figurer
syst uttemp
par U:117
split 2 1
axes hx 0 40 v 0 250
store T0 T3
simu/a
axes h 0 40 v 0 25
simu 0 40
m2
hcopy 0.85
syst uttemp ut pi p
split 3 1
store u[uttemp] T0[uttemp] T3[uttemp] tref
traj 31 8 0.4
ashow u
ashow u
ashow T0
ashow T3 Tref
m3
hcopy 0.85
fig1
m2
hcopy 0.85
fig2
m3
hcopy 0.85
fig3
m3
hcopy 0.85
fig4
m3
hcopy 0.85
fig5
m3
hcopy 0.85
fig6
m3
hcopy 0.85
fig7
m3
hcopy 0.85
end

```

"Endast fig1 med tillhörande
"program är redovisade längre
"fram. Övriga fig-macro skiljer
"sig endast med initialvärde
"och indata.

Appendix 4

```

continuous system uttemp
state t0 t1 t2 t3 tg0 tg1 tg2 tg3
der dt0 dt1 dt2 dt3 dtg0 dtg1 dtg2 dtg3
input U

m11=m10 -(mv1-mv0)
m12=m11 -(mv2-mv1)
m13=m12 -(mv3-mv2)

Q1 = K*(t1-tg1)
Q2 = K*(t2-tg2)
Q3 = K*(t3-tg3)

mv0=0
mv1=(mv0*iv0+kv*(t1-tg1))/iv1
mv2=(mv1*iv1+kv*(t2-tg2))/iv2
mv3=(mv2*iv2+kv*(t3-tg3))/iv3

dt0=-0.716*t0 +1.77*U
dt1=(ma*(cp0*t0-cp1*t1)+mv0*iv0-mv1*iv1-Q1+(mv1-mv0)*(t1-tg1)*A)/M*cp1
dt2=(ma*(cp1*t1-cp2*t2)+mv1*iv1-mv2*iv2-Q2+(mv2-mv1)*(t2-tg2)*A)/M*cp2
dt3=(ma*(cp2*t2-cp3*t3)+mv2*iv2-mv3*iv3-Q3+(mv3-mv2)*(t3-tg3)*A)/M*cp3

dtg0=0
dtg1=(mg*cpg*(tg0-tg1)+il0*m10-il1*m11+Q1-kv*(t1-tg1))/G*cpg
dtg2=(mg*cpg*(tg1-tg2)+il1*m11-il2*m12+Q2-kv*(t2-tg2))/G*cpg
dtg3=(mg*cpg*(tg2-tg3)+il2*m12-il3*m13+Q3-kv*(t3-tg3))/G*cpg

iv0=A*t0+B
iv1=A*t1+B
iv2=A*t2+B
iv3=A*t3+B
A=2.087
B=2467.5

il0=4.189*tg0
il1=4.189*tg1
il2=4.189*tg2
il3=4.189*tg3

G=6500
M=89.4
m10=167
ma=700
mg=1500
K=1162.5
kv=950

cp0=1.11
cp1=1.08
cp2=1.04
cp3=1.01
cpg=0.98
end

```

"t = lufttemperatur
 "tg = granul temperatur
 "U = insignal till brännare
 "m1 = massflöde vatten
 "mv = massflöde ånga
 "Q = Värme flöde från luft
 " till granuler
 "iv = ångans entalpi
 "il = vattnets entalpi
 "G = mängd granuler i en del
 " av torktrumman
 "M = mängd luft
 "ma = massflöde luft
 "mg = massflöde granuler
 "K = värmeövergångskoeff.
 "kv = värmeövergångskoeff.
 " för ånga

Appendix 4

```

connecting system ut
time t
tr=if t<0.5 then 0 else 10
tref[pi] = tr
t3[pi] = t3[uttemp]
t0[p] = t0[uttemp]
out[p] = out[pi]
U[uttemp] = U[p]
end

```

```

discrete system p
"P-regulator
"P=0.3
input out t0
output U
state P
new dP
time t
tsamp ts
output
U=P*(out-t0)
P:0.3
ts=t+dt
dt:1
dP=P
end

```

```

discrete system pi
"PI-regulator
"K=31 och Ti=8.5
input tref t3
output out
state I k Ti
new NI dk dTi
time t
tsamp ts
output
E=tref-t3
out=k*(E+I)
dynamics
NI = I+E/Ti
ts=t+dt
dt:1
Ti:8.5
k:31
dk=k
dTi=Ti
end

```

Appendix 4

```

macro fig1
let n.ifile=3
let fname.ifile=data2
syst diffdata integ ifile figur1 "data2 måste först konverteras
par dt[ifile]:1 "i idpac, vilket göres med
par dt[diffdata]:1 "kommandot conv data2 > data2 3 1
par k1:1
init x:581
split 2 1
axes hx 0 40 vx 580 880
plot x
simu 0 40 0.01
axes hx 0 40 vx 103 125
par k1:0
par k2:1
init x:103.4
plot x
simu 0 40 0.01
end

```

```

connecting system figur
time t
u[diffdata]=k1*(c2[ifile]-581)+K2*(c3[ifile]-103.4)
u[integ]=y[diffdata]
k1:0
K2:0
end

```

```

continuous system integ
input u
state x
der dx
time t
dx=u/15
end

```

```

discrete system diffdata
input u
output y
state diff
new ndiff
time t
tsamp ts
y=u - diff
ndiff = u
ts = t + dt
dt:1
end

```

Appendix 4

```

discrete system indata
input u
output y
time t
tsamp ts
y=u
ts=t+dt
dt:1
end

```

1	730.2	104.4
2	794.6	106.1
3	817.5	107.9
4	828.6	109.55
5	835.6	111.1
6	840.6	112.7
7	843.9	114.11
8	847.8	115.3
9	850	116.3
10	853	117.2
11	856	118
12	860.3	118.7
13	860	119.4
14	860.2	120.1
15	862.1	120.5
16	861.8	120.9
17	862.6	121.4
18	861.7	121.7
19	864	122.1
20	864.3	122.2
21	865.9	122.5
22	864.4	122.8
23	865	123.1
24	865.9	123.3
25	865	123.5
26	865.9	123.4
27	867.5	123.6
28	868.6	123.7
29	868.8	123.6
30	870.3	123.6
31	869.5	123.4
32	869	123.4
33	869	123.3
34	868.8	123.1
35	869.5	123.2
36	870.7	123.35
37	870.5	123.5
38	870.6	123.5
39	872.1	123.56
40	871.2	123.7
41	871.7	123.8
42	872.2	123.8
43	872.8	123.76
44	872.1	123.85
45	871.2	124

```

>Data2
"Kolumn 1 = minuter
"Kolumn 2 = intemperatur
"Kolumn 3 = uttemperatur
"Tabellen måste konverteras
"i idpac innan den används

```

Appendix 4

```
macro traj x y z
init k:x
init Ti:y
init P:z
simu 0 45
end
```

```
macro m2
mark a 6 0
mark "UTTEMPERATUR
mark a 15 0
mark "minuter
mark a 6 6.5
mark "INTEMPERATUR
mark a 15 6.5
mark "minuter
mark a 0.5 5
mark "o
mark a 0 -0.2
mark "C
mark a 0.5 12
mark "o
mark a 0 -0.2
mark "C
end
```

```
macro m3
mark a 6 0
mark "UTTEMPERATUR
mark a 15 0
mark "minuter
mark a 6 4.5
mark "INTEMPERATUR
mark a 15 4.5
mark "minuter
mark a 6 9
mark "INSIGNAL
mark a 15 9
mark "minuter
mark a 0 12.5
mark "bitar
mark a 0.5 3
mark "o
mark 0 -0.2
mark "C
mark a 0.5 8
mark "o
mark 0 -0.2
mark "C
end
```