

CODEN: LUTFD2/(TFRT-5338)/1-53/(1985)

Reglerpaket för digital reglering med dödtidskompensering och framkoppling

Ingemar Ljungdahl
Anders Holmberg

Institutionen för Reglerteknik
Lunds Tekniska Högskola

November 1985

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> November 1985	
		<i>Document Number</i> CODEN: LUTFD2/(TFRT-5338)/1-53/(1985)	
<i>Author(s)</i> Ingemar Ljungdahl Anders Holmberg		<i>Supervisor</i> B. Johansson, G. Persson, B. Wittenmark	
		<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Reglerpaket för digital reglering med dödtidskompensering och framkoppling. (Digital control algorithms for deadtime compensation and feed forward control)			
<i>Abstract</i> <p>Process- and control informationssystem is often a necessary tool for the industry of today. COMATOR PROCESS AB have develop a system called PROVIEW that contains several of well worked through functions. This report discusses some functions more perspicuous while other more in detail, such as digital control (DC). The control-packet contains several algorithms to make it easy to get as good control of the process as possible. The program have PID-absolut, PID-delta feedforward and deadtimecompensation. The deadtimecompensation is implemented with Smith's predictor.</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> 53	<i>Recipient's notes</i>	
<i>Security classification</i>			

FÖRORD

Examensarbetet som avslutning på en civilingenjersutbildning skall eller bör vara ett steg ut i arbetslivet, lite krasst uttryckt, ut i verkligheten.

Klarar vi det här? Har skolan gett en riktig bild av en civilingenjers arbete eller...?

Frågorna var många och spänningen stor när vi fick pärmarna med information från COMATOR och fick se definitionen av ex-jobbet. En del förstod man medan mycket verkade oklart.

Sommaren kom och vi satte igång. Termerna och språket var till en början främmande men efterhand som vi jobbade oss in i systemet klarnade begreppen och "vår" del kändes motiverad. Körde vi fast, och det gjorde vi ibland, fanns det alltid någon som kunde bistå med tips och goda råd.

Därför vill vi tacka alla på COMATOR för en intressant och trevlig tid och för att vi fått göra vårt exjobb hos er.

Särskilt vill vi tacka vår handledare på COMATOR, Göran Persson, för bra handledning och gott samarbete, Bertil Johansson på COMATOR för enkla lösningar på svåra problem, men kanske främst Marie Andersson som enkelt och klarsynt förklarade frågor av programmeringsmässig natur. Vidare skall vår handledare vid reglertekniska institutionen, på LTH, Björn Wittenmark ha tack för att funnits till hands när vi behövt reglerteknisk rådgivning.

Helsingborg, Oktober - 85
Ingemar Ljungdahl
Anders Holmberg

Innehållsförteckning

1	INLEDNING	1
2	COMATOR PROCESS AB OCH PROVVIEW	2
2.1	Presentation av COMATOR PROCESS AB	2
2.2	PROVIEW	3
3	DIGITAL REGLERING	4
3.1	Allmänt	4
3.1.1	Programbeskrivning	4
3.1.1.1	1 : PID - regulator, delta	5
3.1.1.2	2 : PID - regulator, absolut	5
3.1.1.3	3 : Max PID - regulator, delta	6
3.1.1.4	4 : Summa PID - regulator, delta	6
3.1.1.5	5 : Summan av 2 insignaler-generell signal	6
3.1.1.6	6 : Min av 2 insignaler-generell signal	6
3.1.1.7	7 : Max av 2 insignaler-generell signal	6
3.1.1.8	8 : Diff. av 2 insignaler-generell signal	6
3.1.1.9	9 : Lead / Lag - reglering	6
3.1.1.10	Dödtidskompensering	7
3.1.1.11	Framkoppling	7
3.1.1.12	Utmatning	8
3.2	Reglerteori	8
3.2.1	PID-regulator, absolut	8
3.2.2	PID-regulator, delta	10
3.2.3	Lead/Lag - reglering	11
3.3	Dödtidskompensering	11
3.3.1	Allmänt	11
3.3.2	Smiths' regulator	12
3.4	Framkoppling	13
3.4.1	Allmänt	13
3.4.2	Framkoppling - återkoppling	14
4	HANDHAVANDE	15
4.1	Operatörskommunikation	16
4.2	Programuppbyggnad för systemet	18
4.2.1	DATABAS	19
4.2.2	DCDCB1,2	19
4.2.3	DCHALA	19
4.2.4	DCVALI	19
5	TESTNING OCH RESULTAT	21
5.1	Testning	21
5.1.1	Kärning av testuppkopplingen	22
5.2	Presentation av kurvor	23
6	APPENDIX	30
6.1	Appendix 1:Huvud och pseudokod för programmen	30
6.2	Appendix 2:DATABAS SUBROUTINER för DC	39
6.2.1	DCR : hämta reell parametrar i DC-tabell	43
6.2.2	DCRLA : ändra reell parameter i DC-tabell	44
6.2.3	DCI : hämta integer parameter i DC-tabell	45
6.2.4	DCILA : ändra integer parameter i DC-tabell	46
6.2.5	DCS : hämta datasträng parameter i DC-tabell	46
6.2.6	DCSLA : ändra datasträng parameter i DC-tabell	47
6.3	Appendix 3:Verifiering av lead/lag - reglering	48
6.4	Appendix 4:Dödtidskompensering, härledning	49
7	REFERENSER	50

1 INLEDNING

Från Comator Process AB fick vi ett programpaket skrivet i FORTRAN för digitalreglering. Uppgiften bestod i att anpassa dessa FORTRAN-rutiner till Comators nya processdatorsystem, PROVIEW. Dessutom skulle två nya funktioner implementeras, framkoppling och dættidskompensering.

Vi utgick från två äldre program för digital reglering när vi skrev vårt program, byggstenarna: PID-delta och PID-absolut reglering, användes i det stora hela som de stod, likaså de andra algoritmerna. Vårt program ändrades så att det kunde utnyttja befintliga subrutiner för hämtning och ändring i databasen, vidare kom vi fram till en bra lösning till problemet med dættider där Otto Smiths' regulator utnyttjas. En enkel framkoppling infogades också i programmet.

Vi fick bestämma och implementera en helt ny uppsättning av databasparametrar för digital reglering. Se APPENDIX 2 där dels själva databasparametrarna och dels rutinerna för manipulering av dessa beskrivs.

Arbetet utfördes på Comators husdator PDP 11/44 under operativsystemet RSX11M+ med hjälp av Comators programutvecklingssystem, rapportgenerator, databashanterare och filhanterare.

Förutom själva regleralgoritmerna skulle även operatörs-kommunikationen ingå: formulärhantering för ifyllande av reglerparametrar.

Arbetet avslutades med att testa regleralgoritmerna. Ett simuleringsprogram fick skrivas som simulerade en första ordningens process. Den simulerade processen styrdes med ovan nämnda hjälpmedel, resultatet studerades och redovisas också i denna rapport.

2 COMATOR PROCESS AB OCH PROVIEW

2.1 Presentation av COMATOR PROCESS AB

Comator Process Ab har sedan företagets grundande 1973 verkat inom området för industriella datorsystem. Företaget har succesivt byggts upp genom rekrytering av personal främst från processindustrin.

Verksamheten är idag till största delen inriktad på kvalificerade programprodukter för industritillämpningar. Som exempel kan nämnas styrning av vindtunnlar, styr- och övervakningssystem för kabelutläggningsfartyg, samt processdatorsystem för industrin.

Comators flesta leveranser sker med funktionsansvar för hela datorsystemet, d v s från programvara till körklar anläggning.

Exempel på produkter kan nämnas olika kommunikationsprogram, datainsamling, digitalreglering, bildskärmspresentation (svartvit/färg), kurvpresentation, X/Y-plotter, vägnings- och doseringssystem, produktionsrapporter samt ytterligare ett stort antal moduler, alltifrån små enkla program till kompletta system för en viss uppgift eller process.

2.2 PROVIEW

Proview är för det första ett processinformationssystem som kan anslutas som ett överordnat informationssystem till existerande styr- och regler-system och för det andra ett kombinerat process- och reglerinformationssystem, där det sist nämnda systemet till en del är resultatet av vårt examensarbete. Proview kan på så sätt bidra till modernisering av äldre, väl fungerande styr- och regler-system eller byggas in redan från början i nya processanläggningar.

Proview är baserat på DEC Micro-PDP men är körbart på alla DEC-datorer under operativsystemet RSX11M och RSX11M-PLUS.

Proview är uppbyggt av ett grundsystem omfattande:

- maskinvara (dator, bildskärm och skrivare)
- programvara (operativsystem och PROVIEW-grundmodul)
- övrigt (garanti, dokumentation, utbildning/assistans)

Till detta grundsystem kan fogas en eller flera av följande basfunktioner:

- DIGIN (databashantering av DI ,(Digital In))
- DIGUT (databashantering av DO ,(Digital Ut))
- ANAIN (databashantering av AI ,(Analog In))
- ANAUT (databashantering av AO ,(Analog Ut))
- DDC (databashantering av DC ,(Digital Reglering))
- ALARM (alarmhantering)
- TREND (trendkurvor)
- SCHEMA (process-scema)

Systemet är också utbyggbart med ett stort antal tillägg-funktioner: maskinoptioner och programoptioner. Likaså är det enkelt för den programmeringskunnige användaren att lägga till egna applikationsprogram, editera och länka dessa till systemets databas.

3 DIGITAL REGLERING

I delkapitel 3.1 ges en översikt av det digitala reglerprogrammet DDCP - med kommentarer och anvisningar till de olika algoritmerna. I 3.2 följer en genomgång ur regler-teknisk synvinkel av PID-regleralgoritmerna och annan reglerteori. Dödtidskompensering och framkoppling behandlas separat i delkapitel 3.3 och 3.4.

3.1 Allmänt

Detta dokument innehåller en allmän beskrivning av program-systemet DDCP, ett generellt styrsystem.

Programmet har alla ändringsbara parametrar, såsom börvärde, P-, I- och D-konstanter mm, samlade i en central tabell uppdelad per mätpunkt. Dessa tabeller är ändringsbara från operatörsterminalen på ett enkelt och operatörsvänligt sätt. Ingen programmering krävs varken vid implementering eller ändring. En ny mätpunkt skapas helt från operatörsterminalen. Mer om detta i kapitel 4.

Syftet med denna beskrivning är att ge en orienterande information om reglerprogrammets uppbyggnad och funktion.

3.1.1 Programbeskrivning

Programmet är gjort för 16 regulatorer, regleringen börjar med regulator 16 och avslutas med regulator 1. För att reglering skall kunna ske måste blockering vara i läge från, "F", för aktuell regulator.

Till varje regulator finns ett individuellt reglertidsintervall, som anger tid mellan regleringreppen.

Hur lång tid detta är i sekunder är beroende av hur ofta programmet körs, om programmet exempelvis går var trettionde sekund blir samplingsintervallet:

30 * reglertidsintervallet.

När man funnit en regulator med blockeringen i läge från och räknaren för tid mellan ingrepp har räknats ner till noll hämtas data från databasen som möjliggör reglering.

En kontroll av börvärdet görs så att det håller sig innanför givna gränser, ett för högt eller för lågt värde medför att börvärdet sätts till respektive gränsvärde.

Via rutinen IBADE hämtas signalen som skall regleras från AI(Analog In)- eller AO(Analog Ut)-tabell, vid något fel på insignalen sker ingen reglering. IBADE används för alla insignaler.

Programmet innehåller 9 stycken regleralgoritmer, numrerade

1 - 9 och möjlighet att kombinera dessa med framkoppling och/eller dædtidskompensering. Framkoppling och dædtidskompensering har ingen verkan vid användande av algoritmerna 5 - 9.

Beteckningar använda i programmet och i beskrivningen av algoritmerna nedan :

VAL1 : insignal 1
 OLD1 : færegående insignal 1
 OLD2 : fære færegående insignal 1
 VAL2 : insignal 2
 BORV : bærværdet
 FBORV : færegående bærværdet
 ERROR : BORV - VAL1
 TERM : reglertidsintervall
 PK : proportionalitetskonstant
 IK : integrationskonstant
 DK : derivatakonstant
 YP : reglerfel P-del
 YI : reglerfel I-del
 OLDYI : færegående reglerfel I-del
 YD : reglerfel D-del
 YF : reglerfel framkoppling
 YM : modellens utsignal vid dædtidskompensering
 DN : delresultat vid lead/lag-reglering
 DN1 : færegående DN
 CN : delresultat vid lead/lag-reglering

3.1.1.1 1 : PID - regulator, delta

$$\begin{aligned} YP &= PK * (ERROR - (FBORV - OLD1)) \\ YI &= PK * TERM * (FBORV - OLD1) / IK \\ YD &= PK * DK * (VAL1 - 2OLD1 + OLD2) / TERM \\ YY &= YP + YI - YD + YF \end{aligned}$$

Om framkoppling ej utfæres sættes YF = 0.

3.1.1.2 2 : PID - regulator, absolut

$$\begin{aligned} YP &= PK * ERROR \\ YI &= PK * TERM * (FBORV - OLD1) / IK + OLDYI \\ YD &= PK * DK * (VAL1 - OLD1) / TERM \\ YY &= YP + YI - YD + YF \end{aligned}$$

Om framkoppling ej utfæres sættes YF = 0. Fæer att undvika wind-up pæ YI fæer denna værdet av max eller min signal om den gæer utanfæer dessa grænser.

Fæer bæde algoritm 1 och 2 gæller, att om skillnaden BORV - VAL1 æverstiger "grænser fæer I-undertryckning" sættes YI = 0.

3.1.1.3 3 : Max PID - regulator , delta

Max av 2 insignaler regleras med algoritm 1.

$$VAL1 = \text{Max} (VAL1, VAL2)$$

3.1.1.4 4 : Summa PID - regulator , delta

Summan av 2 insignaler regleras med algoritm 1.

$$VAL1 = VAL1 + VAL2$$

3.1.1.5 5 : Summan av 2 insignaler-generell signal

$$YY = VAL1 + VAL2 - \text{generell signal}$$

3.1.1.6 6 : Min av 2 insignaler-generell signal

$$YY = \text{Min} (VAL1, VAL2) - \text{generell signal}$$

3.1.1.7 7 : Max av 2 insignaler-generell signal

$$YY = \text{Max} (VAL1, VAL2) - \text{generell signal}$$

3.1.1.8 8 : Diff. av 2 insignaler-generell signal

$$YY = VAL1 - VAL2 - \text{generell signal}$$

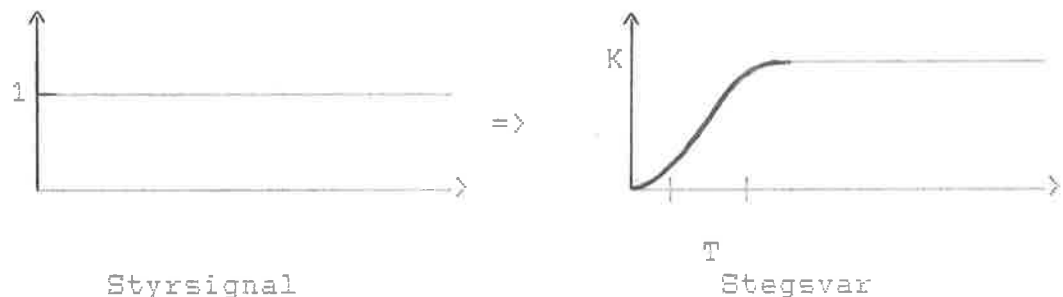
För algoritmerna 5 - 8 gäller att adressen till den generella signalen anges av användaren vid parameter-namnet "insignal 3", i bilden för mätpunktstabellen.

3.1.1.9 9 : Lead / Lag - reglering

$$\begin{aligned} DN1 &= \text{föreg. DN} \\ DN &= DN1 + \text{TERM} * (VAL1 - DN1) / IK \\ CN &= DK * (DN - DN1) + DN \\ YY &= PK * CN \end{aligned}$$

3.1.1.10 Dædtidskompensering

Dædtidskompensering används för att minska inverkan av tidsfördröjningar hos processen. Kompenseringen har implementerats med Otto Smiths' regulator som förenklat innebär att man ansätter en processmodell och återkopplar styrsignalen genom denna, en del fördröjs motsvarande processens tidsfördröjning och en del utan tidsfördröjning, skillnaden mellan dessa signaler adderas till felsignalen ERROR och PID-reglering sker sedan med den modifierade ERROR-signalen enligt algoritm 1 eller 2. Operatören ansätter en första ordningens processmodell genom att ange en förstärkningsfaktor och en tidskonstant. Vidare uppskattas processens tidsfördröjning (dædtiden). För att kunna ansätta en processmodell studerar man processens stegsvar; man ger processen ett enhetssteg på styrsignalen och får då tidskonstant och förstärkning för processen enligt figurer.



Figur 1 Uppskattning av förstärkningsfaktor och tidskonstant.

T definieras som 10/90-stigtiden för signalen.

Dædtidskompensering ger ingen verkan med algoritmerna 5-9.

OBS! Dædtidskompensering kan ej användas om systemet är instabilt eller om systemet innehåller integratorer.

3.1.1.11 Framkoppling

Anledningen till att använda framkoppling är att man snabbare kan kompensera för ingrepp i processen än vanlig PID - reglering klarar.

Den signal som skall framkopplas anges av operatören och absolut PID - reglering görs på skillnaden mellan föregående och nuvarande framkopplingsignal.

3.1.1.12 Utmatning

2 kaskadkonstanter, K1 och K2, används på följande sätt, är K1 = 0 normal utgång enligt nedan, annars räknas styrsignalen YY om som $YY = K1 * YY + K2$.

Man kan välja mellan att ha inkrementell ändring av styrsignalen eller att styrsignalen får ett nytt värde varje gång.

Vid kaskadreglering läggs den beräknade styrsignalen i en AI-tabell och används som insignal till en annan regulator.

När K1 = 0, alltså normal utgång gäller följande för YY:

YY måste vara större än "min regleringrepp" om utmatning skall ske, om ej lagras YY och adderas till styrsignalen vid nästa reglertillfälle.

YY får ej vara större än "max pos.ändring av utsignal", om så är fallet sätts YY till detta värde.

YY får ej vara större än "max neg.ändring av utsignal" (absolutvärden), om så är fallet sätts YY till detta värde.

Vid normal utgång adderas YY till nuvarande styrsignal och läggs i en AO-tabell.

3.2 Reglerteori

Teorin för kontinuerliga system har följts och sedan approximerats till den digitala motsvarigheten.

3.2.1 PID-regulator, absolut

För en PID-regulator råder följande samband mellan reglerfelet e och styrsignal u :

$$u(t) = P * (e(t) + D * de/dt + 1/I * \int_0^t e(s) ds)$$

Reglerfelet är differensen mellan referensvärdet Yr och utsignalen Y, dvs $e = Yr - Y$.

Styrsignalen är alltså sammansatt av tre termer där P betecknar proportionaltermen, vilken är proportionell mot felet, D derivatatermen, som är proportionell mot felets derivata och I integraltermen, som är proportionell mot felets tidsintegral.

Ofta kan referensvärdet Yr ej predikteras. Detta är t.ex. fallet då referensvärdesändringarna är stegformade.

Derivatatermen de/dt ersättes därför ofta med $-dy/dt$ för

att undvika för stora styrsignaler in till processen.
Reglerfunktionen kan då beskrivas med:

$$u(t) = P * (e(t) - D * dy/dt + 1/I * \int^t e(s) ds)$$

Om man låter derivataterman realiseras med approximationen:

$$D * de/dt = pD / (1 + pD/N) e(s)$$

där $N = 3-30$ och p är differentialoperatorn, får PID-regulatorns överföringsfunktion följande utseende:

$$G(s) = P * (1 + 1/sI + sD / (1 + sD/N)) , \quad U(s) = G(s) * E(s)$$

Styrsignalen från proportionaltermen blir:

$$U_p = P * E = P * (Y_r - Y)$$

För att åstadkomma integralverkan approximeras integralen med summan:

$$I(t) = P * h / I * e(t - kh) , \quad h \text{ är samplingsintervallet.}$$

Summan kan beskrivas med rekursionsformeln:

$$I(t) = I(t-h) + P * h / I * e(t-h)$$

Styrsignalen från integraltermen blir då:

$$U_i = P * h / I * E(n-1) + U(n-1)$$

För att åstadkomma derivataverkan så används bakåtapproximation på derivatadelen enligt $s = (1 - q^{-1}) / h$:

$$\frac{(1 - q^{-1}) * D / h}{1 + (1 - q^{-1}) * D / Nh} * P * E(n) = \frac{1 - q^{-1}}{h/D + (1 - q^{-1}) / N} * P * E(n)$$

(om N är tillräckligt stort) $= D/h * (1 - q^{-1}) * P * E(n)$

och med $D * de/dt$ ersatt med $-D * dy/dt$ får vi styrsignalen från derivatatermen:

$$U_d = -P * D / h * (1 - q^{-1}) * Y(n) = -P * D / h * (Y(n) - Y(n-1))$$

där $Y(n)$ är nuvarande insignal och $Y(n-1)$ närmast föregående.

Om vi summerar styrsignalerna får vi:

$$U = U_p + U_i + U_d$$

Den beskrivna regulatorstrukturen används i algoritm 2.

3.2.2 PID-regulator, delta

Låt oss som tidigare skriva upp den kontinuerliga PID-regulatorn på Laplaceform:

$$U(s) = P * (1 + 1/sI + sD/(1+sD/N)) * E(s)$$

med beteckningar enligt tidigare. Om man använder bakåt-skiftapproximation på det kontinuerliga systemet erhåller vi en digital PID-regulator med följande P, I och D -delar:

$$P, \quad P*h/I(q-1), \quad P*D(q-1)/h(q+w) \quad \text{där } w = - \exp(-hN/D).$$

Vi får alltså en digital PID-regulator med utseendet:

$$u(kh) = P * (1 + h/I(q-1) + D(q-1)/h(q+w)) * e(kh)$$

Om man istället jämför ändringar i styrsignalen $u(kh)$, kan man skriva upp dessa ändringar som:

$$\begin{aligned} \Delta u(kh) &= u(kh) - u(kh-h) = \\ &P * (1-q + hq /I + D(1-2q + q^2)/h(1+wq)) * e(kh) \end{aligned}$$

Styrsignalen från proportionaltermen blir:

$$U_p = P * (E(n) - E(n-1))$$

där $E(n)$ och $E(n-1)$ är nuvarande och föregående reglerfel.

Styrsignalen från intergraltermen blir:

$$U_i = P*h/I * E(n-1)$$

Med de/dt som tidigare ersatt av $-dy/dt$ och om $N=3-30$ blir $w \approx 0$ så kan derivatatermen approximeras med:

$$U_d = -P*D/h * (Y(n) - 2*Y(n-1) + Y(n-2))$$

där $Y(n)$, $Y(n-1)$ och $Y(n-2)$ är nuvarande, föregående och före föregående insignal (ärvärde). Den totala styrsignalen är precis som i det föregående summan av U_p , U_i och U_d .

Den beskrivna regulatorstrukturen används i algoritm 1.

Anmärkning: Som den uppmärksamme läsaren ser har, vid programbeskrivningen för delta och absolut PID - reglering, D-delen angetts med negativt tecken, detta minustecken återfinns här i formeln för U_d .

3.2.3 Lead/Lag - reglering

Algoritmen för lead/lag - reglering (fasavancerande och fasretarderande) är tagen från ett äldre program för digital reglering. En "bakåträkning" ger det kontinuerliga systemets motsvarighet - se appendix 3, för övrigt behandlas ej lead/lag - reglering i vår rapport.

3.3 Dödtidskompensering

Först behandlas problemet med dödtider i processer i ett vidare perspektiv och sedan beskrivs mera ingående den lösning vi har använt oss av, Smiths' regulator.

Beskrivningen följer i stora drag K J Aströms behandling av ämnet, se ref.

3.3.1 Allmänt

Reglering av processer med långa dödtider är ett exempel på reglerproblem där de enkla regulatorerna ej räcker till.

Dödtider förekommer allmänt vid reglering av industriella processer. Tidfördröjningar uppträder alltid i samband med materialtransporter i rör eller på band. Mätning av sammansättning kan för det mesta endast göras i speciella analysatorer. Det är då nödvändigt att ta ett prov, vilket tar tid. En ytterligare tidsfördröjning uppstår genom att själva mätningen tar en viss tid. I många fall finns också fördröjningar i samband med styrdonen. En tidsfördröjning kan ofta vara en bra approximation till ett system med högre ordningens dynamik.

System med dödtider är svåra att reglera. Ett skäl är att den prediktion som erhålles med hjälp av linjär trendextrapolation är mycket dålig. För system med dödtider är det mycket viktigt att känna styrsignalens gamla värden för att kunna prediktera utsignalens framtida värden.

Problematiken belyses klarare genom ett exempel:

Antag att reglerfelet hos ett regler-system är EO vid tiden T_0 . Styrvariabeln ökas då med $k \cdot EO / T_i$, k och T_i är proportional- respektive integralkonstant. För ett system utan tidsfördröjning skulle detta medföra en minskning av reglerfelet. Men på grund av dödtiden slår styringreppet ej igenom och reglerfelet ändras ej så mycket. Samma förändring som tidigare görs nu i styrvariabeln. Förloppet upprepas sedan i 3 steg. Vid tiden T_4 har den ändring som gjordes vid T_0 slagit igenom och reglerfelet blir litet.

Vid tiden T5 uppträder effekten av den ändring som gjordes vid T1 och reglerfelet blir ungefär $-E_0$, vid T6 blir det ungefär $-2*E_0$ osv.

Av resonemanget följer att svängningar kommer att uppstå om regulatorns förstärkning bibehålles vid det värde som passar en process utan tidsfördröjning.

Om man lagrar gamla styringrepp och tar hänsyn till dessa vid beräkning av styrsignalens nya värden skulle resultatet avsevärt kunna förbättras. Detta är ide'n bakom reglering av system med stora döttider.

3.3.2 Smiths' regulator

Reglering av processer med långa döttider kan göras med Smiths' prediktor eller Smiths' regulator efter sin upphovsman Otto Smith. En studie av regulatorn avslöjar en ide' av central betydelse. Det visar sig att Smiths' regulator innehåller en matematisk modell av den process som skall regleras. Regulatorn har alltså kunskap om processen inbyggd i sig.

Smiths regulator har en del begränsningar, den kan ej användas för instabila system och den kan vara känslig för mätbrus.

Antag att den reglerade processen har överföringsfunktionen: $\exp(-sT)*G_p$, där T är döttiden.

Bestäm först överföringsfunktionen G_r , för en regulator som skulle ge bra reglering av motsvarande process utan tidsfördröjning.

Det slutna systemets överföringsfunktion:

$$G(s) = G_p * G_r / (1 + G_p * G_r)$$

Antag att processen med tidsfördröjning regleras med en regulator med överföringsfunktionen G_{rd} . Det slutna systemet blir då:

$$G_d(s) = \exp(-sT) * G_p * G_{rd} / (1 + \exp(-sT) * G_p * G_{rd})$$

På grund av tidsfördröjningen kan vi naturligtvis ej kräva att $G(s) = G_d(s)$. Däremot skulle det vara möjligt att bestämma G_{rd} så att $G_d(s) = \exp(-sT) * G(s)$, dvs:

$$\exp(-sT) * G_p * G_{rd} / (1 + \exp(-sT) * G_p * G_{rd}) =$$

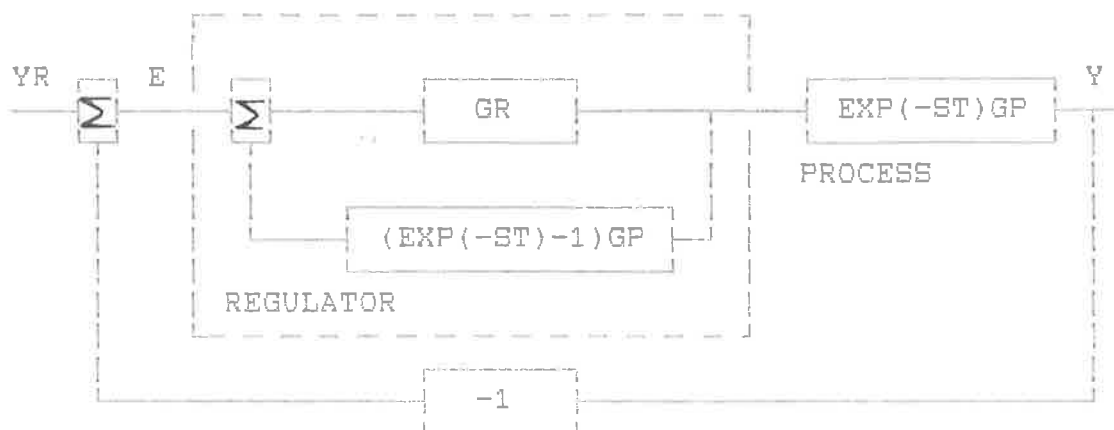
$$\exp(-sT) * G_p * G_r / (1 + G_p * G_r)$$

alltså: $G_{rd} = G_r / (1 + (1 - \exp(-sT)) * G_p * G_r)$.

Detta är Smiths' formel.

För att bestämma en regulator för en process med döttid bestämmas först en regulator G_r , som ger bra reglering för motsvarande process utan döttid. Regulatorn G_{rd} för processen med döttid kan sedan beräknas ur Smiths' formel.

Gr kan t.ex. väljas som en PID-regulator. Figuren nedan visar ett blockschema över en process som regleras med en Smith-regulator.



Figur 2 Blockschema över Otto Smith-regulator.

Sambandet mellan insignal och utsignal kan skrivas som:

$$u(t) = Gr * (Yr(t) - Y(t) - Ym(t) + Ym(t-T))$$

där Ym står för den ansatta modellens utsignal och T för dödtiden.

Vi har valt att ansätta en processmodell av första ordningen enligt: $G_m = \exp(-sT) * K / (1+sT_1)$ där T som tidigare betecknar dödtiden, K är modellens förstärkningsfaktor och T1 dess tidskonstant. Beräkningar, se appendix 4, överför det kontinuerliga systemet till den digitala motsvarigheten. Följande styrlag erhålles då, antag samplingstiden $h = 1$.

$$Y(k+1) = AM*Y(k) + BM2*U(k-d+1) + BM1*U(k-d)$$

AM, BM1 och BM2 är konstanter beräknade i appendix 4, d är det närmast större hela samplingsintervall jämfört med dödtiden.

3.4 Framkoppling

Först ges en allmän beskrivning av framkoppling, och varför det kan vara motiverat att använda denna, sedan kommer en beskrivning av hur vi har löst det.

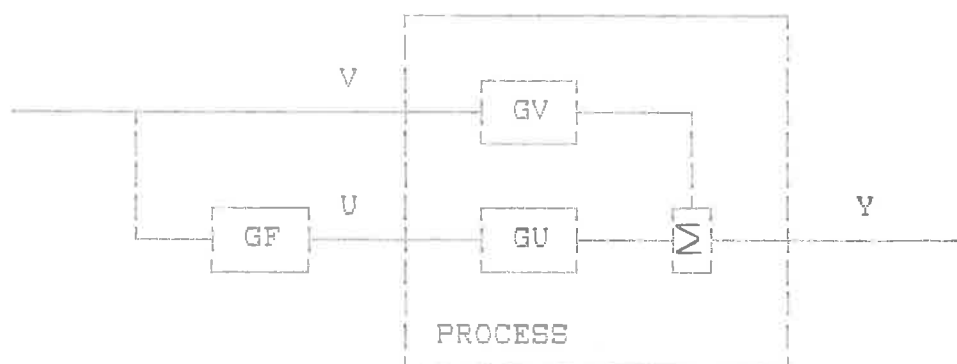
3.4.1 Allmänt

Det vanligaste sättet att eliminera störningar i processer är återkoppling, men nackdelen med ett återkopplat system är att först måste ett reglerfel uppstå för att regulatorn

skall kunna eliminera störningen.

Ibland är det möjligt att mäta störningen innan den påverkar utsignalen och i så fall ändra styrsignalen så att störningens inverkan minskar. Detta förfarande kallas framkoppling, på engelska feedforward.

Vid användning av framkoppling beskrivs den reglerade processen som ett system med två insignaler: störningen V och styrsignalen U , och en utsignal Y , (se fig).



Figur 3 Reduktion av störningar med hjälp av framkoppling.

Är systemet linjärt och tidsinvariant kan det beskrivas med sina överföringsfunktioner G_u , G_v och G_f som har följande betydelse:

- G_u : överföringsfunktion mellan styrsignalen U och en utsignal Y .
- G_v : överföringsfunktion mellan störning och styrsignal.
- G_f : överföringsfunktion för framkopplingslänk sådan att G_u och G_f ger motsatt verkan till G_v (se nedan).

Framkopplingen går ut på att finna en kompenseringslänk sådan att den sammansatta verkan av G_u och G_f ger motsatt verkan mot G_v .

Alltså, framkopplingsvillkoret : $G_u * G_f = -G_v$.

3.4.2 Framkoppling - Återkoppling

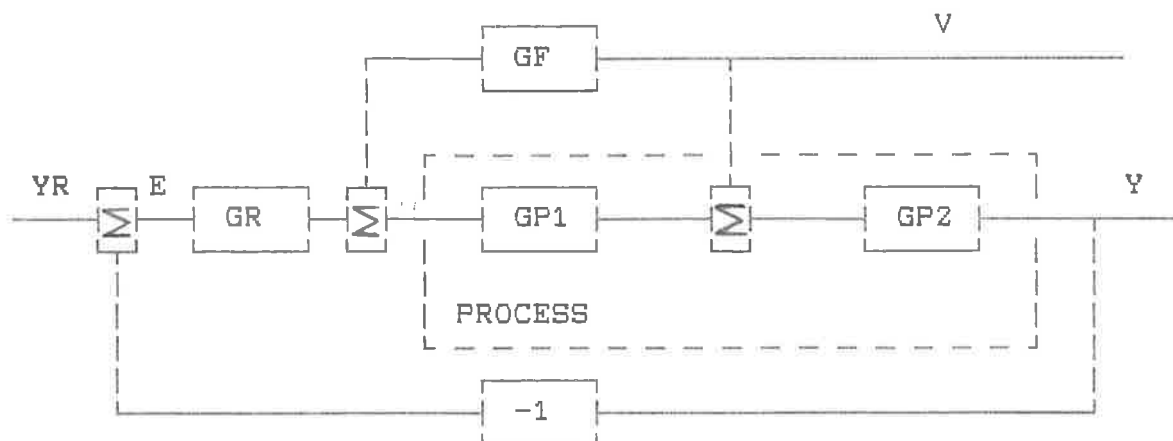
Framkoppling och återkoppling kompletterar varandra och det är ofta fördelaktigt att använda dem tillsammans.

Framkopplingen tar hand om stora och frekventa störningar medan återkopplingen tar resten; oexakthet hos framkopplingen och andra ej mätbara störningar.

Två nackdelar med den i det föregående beskrivna varianten av framkoppling är att man måste kunna detektera störningen och man måste veta hur systemet påverkas av den.

Men även en oprecis framkoppling kan vara effektiv. Vidare kan nämnas att återkoppling kan leda till instabil-

itet medan framkoppling ej ger några stabilitetsproblem.



Figur 4 Blockschemata över framkoppling - återkoppling.

Figur 3 visar hur återkoppling och framkoppling kan kombineras. Processen har delats upp i två delsystem, GP1 och GP2 annars samma beteckningar som tidigare.

Enkel räkning ger ett uttryck på reglerfelet:

$$E = 1/(1+GR*GP) * YR - GP2*(1+GP1*GF)/(1+GR*GP) * V$$

$$\text{där } GP1*GP2 = GP$$

Om framkopplingen väljs så att $GP1*GF = -1$ blir överföringsfunktionen för reglerfelet noll och störningens inverkan kan elimineras.

Vi har använt oss av den beskrivna framkopplingsprincipen.

En utvald signal : "framkopplingssignal" , samplas och nuvarande värde jämförs med närmast föregående och en framkoppling görs på denna skillnad med en absolut PID-regulator. Ofta är det fullt tillräckligt med enbart proportionell reglering.

4 HANDHAVANDE

I detta kapitel redogörs för operatörsdialogen samt för de ändringar som gjorts där. En beskrivning av kopplingen mellan de olika programmen i systemet redovisas också.

4.1 Operatörskommunikation

Operatörskommunikationen består i princip av två delar:

- kommandoexekutiven (KEX).
- dialogprogrammen (DCMD, DCIOMD).

Basen för operatörskommunikationen kallas för KEX och består av ett antal kommando som operatören kan välja på: AN/(VI)/LI DC n. AN står för ändra, VI för visa, LI för lista, DC för digital reglering och n för punktnummer inom DC. (VI står inom parantes eftersom detta kommando ej behöver skrivas).

Utgångsläget för KEX är "Kommando" - frågan till operatören. Denna läggs ut nederst till vänster på bildskärmen. Om operatören anger ett korrekt kommando lämnar KEX över kontrollen till dialogprogrammen, i vilket operatören i detalj anger vad han vill få gjort bla. genom att trycka på funktionstangenterna PF1-4. Vissa enklare kommandon kräver ingen operatörsdialog utöver KEX - kommandot, varför "Kommando" - frågan omedelbart uppträder igen. Om operatören anger ett felaktigt kommando skrivs "Felaktigt kommando" på raden ovanför.

När operatören angivit ett korrekt KEX - kommando går enligt ovan kontrollen över till dialog programmet. Operatören har via dialogprogrammen möjlighet att under drift, från bildskärmen programmässigt konstruera en ny reglerpunkt eller ändra på tabellkonstanter för en redan existerande reglerpunkt. Vid inläggning av en ny reglerpunkt skrives samtliga parametrar in via tangentbordet. De fält som operatören kan modifiera indikeras med ljus bakgrund. Fältets längd indikeras också på detta sätt. Fält som operatören ej kan modifiera saknar denna bakgrund. I vissa fält kan utgångsdata vara ifyllda, så att operatören endast behöver ändra ett fåtal fält.

Vid digital reglering visas/listas/ändras data med hjälp av två bilder. Bild 1 innehåller i stort sett samma parametrar som ingår i AI och AO medan bild 2 innehåller parametrar som bärvärde, ärvärde, styrsignal, P-, I-, D-konstanter mm. För att reglering skall kunna ske måste blockering vara i läge från, "F", för aktuell reglerpunkt. Nedan visas exempel på AN/VI/LI DC n kommandon samt tillhörande definition av funktionstangenterna.

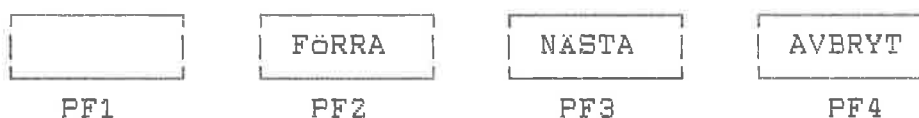
Ett lista kommando (ex. LI DC 14) medför att databasparametrarna för DC skrivs ut på en skrivare. Bildskärmen raderas ej med detta kommando.

Ett visa kommando (ex. VI DC 14 eller bara DC 14) medför att databasparametrarna för DC 14 kommer upp på bildskärmen. Samtidigt definieras de fyra funktionstangenterna:



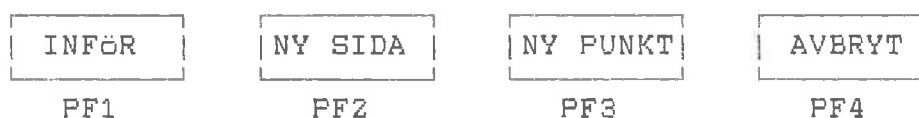
Operatören har då möjlighet att lista (LISTA) aktuell punkt på skrivare, att bläddra fram (NY SIDA) till nästa sida eftersom databasparametrarna för DC inte får plats på en sida, att välja ny punkt (NY PUNKT) eller att avbryta (AVBRYT) visning av databasparametrarna för punkter av vald typ (DC).

När operatören valt ny punkt ändras definitionen av funktionstangenterna till:



Operatören har då möjlighet att bläddra tillbaka (FÖRRA) till föregående punkt av typ DC, att bläddra fram (NÄSTA) till nästa punkt av typ DC eller att avbryta (AVBRYT) visning av databasparametrarna för punkter av vald typ DC.

Ett ändra kommando (ex. AN DC 14) medför att databasparametrarna för utpekad punkt kommer upp på bildskärmen. Samtidigt definieras de fyra funktionstangenterna:



Operatören har då möjlighet att avbryta (AVBRYT) ändring av punkter inom samma typ DC. Ingen lagring av ändrade data på den bild som just är uppe kommer att ske.

Tangenten (INFÖR) används när man vill avsluta ändringsdialogen för en viss punkt och lagra inmatade data. Lagringen sker inte direkt utan operatören måste kvittera en extra gång på de omdefinierade funktionstangenterna enligt nedan:



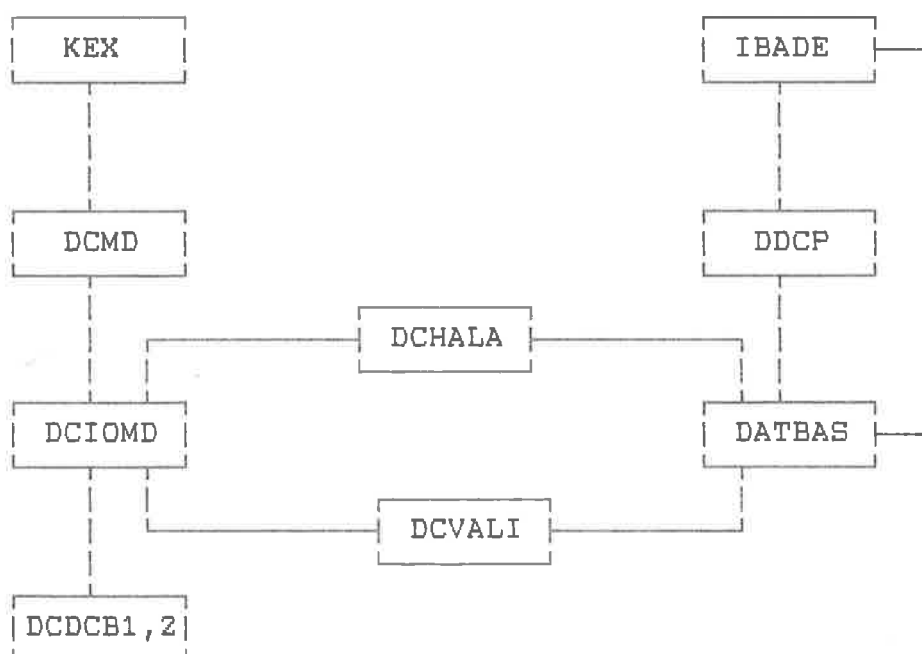
Vill operatören lagra ändrade data trycker han på (LAGRA), vill han inte lagra trycker han på (LAGRA EJ). Angrar han sig och vill ändra mer trycker han på (TILLBAKA) varefter han åter befinner sig i ändringsmod på vald punkt.

Operatören kan också använda tangenterna (NY SIDA) och (NY PUNKT) vars funktion har definierats tidigare under kommandot visa. I båda fallen ges möjlighet till lagring av nya data genom samma kvitteringsförfarande som vid (INFÖR).

Pågående ändringsdialog kan avbrytas genom att trycka på (AVBRYT). Ingen data från aktuell punkt kommer då att lagras.

Samma kommando (ÄN/(VI)/LI) finns naturligtvis också för AI, AO, DI, DO, men här skiljer sig definitionerna av funktions-tangenterna från DC.

4.2 Programuppbyggnad för systemet



Figur 4 Blockschemata över programuppbyggnaden.

Blocken KEX, DCMD och DCIOMD har beskrivits tidigare. (Se Appendix 1 för programuppbyggnaden av DCMD och DCIOMD). DDCP är huvudreglerprogrammet och IBADP är en funktion till DDCP som testar om signalen är riktig för reglering samt plockar upp signalen. (Se Appendix 1).

4.2.1 DATABAS

PROVIEW - systemet samlar kontinuerligt in mätdata från processen. Dessa mätdata lagras i den så kallade databasen. I databasen lagras således uppmätta temperaturer och flöden men även information om huruvida en ventil är öppen eller stängd.

Databasen består av ett antal parametrar som beskriver varje mätpunkt (analog, digital, beräknad eller manuellt inmatad, digital reglering). Dessa parametertabeller lagras normalt i datorns internminne för att snabbt vara tillgängliga för programmen. Endast vid mycket stora system lagras vissa av tabellerna på externminnet. Parametrarna talar om hur signalen skall behandlas. Det är mycket viktigt att dessa parametrar är riktiga. Normalt bestäms deras värde vid idrifttagningen för att sedan ej ändras om inte något oförutsett händer. (Se Appendix 2 DATABAS SUBROUTINER för DC)

Efter inmatning behandlar systemet det inmatade värdet. Reglerprogrammet (DDCP) som önskar tillgång till någon av mätpunktens parametrar, t ex senast inlästa mätvärde, för att utföra en beräkning, anropar databasen med internpunkt-nummer (kretsnummer) och parameterindex som argument. Efter beräkningen återskrives det beräknade värdet i databasen och är där tillgängligt för alarmhantering, kurvpresentation, process, process - schema, rapporter m m.

4.2.2 DCDCB1,2

Databasparametrarna presenteras på bildskärmen med hjälp av menyfilen DCDCB1.MNY eller DCDCB2.MNY beroende på vilken sida som visas på skärmen. Filen *.MNY innehåller "cursor" - hantering och fältmarkeringar och är därför terminal-beroende. Operatörsdialogens layout definieras av dess definitionsfil (*.RDF) se figur 4.

4.2.3 DCHALA

Hämtar data till operatörsdialogens layout eller lagrar data från operatörsdialogens layout i databasen. Vid lagring lagras endast ändrade data. (Se Appendix 1).

4.2.4 DCVALI

Validerar numeriska parametrar och sträng parametrar. Hämtar strängar och numeriska värden från bilden och undersöker om de är rätt strängar eller om de numeriska värdena är inom rätt gränser när man ändrar i bilden. (Se Appendix 1).

Mätpunktstabell reglerparametrar (1)

Namn	REG001		
Internnr	1		
Beskrivande text	REGULATOR 1		
Typ av börvärde (LO,BE,MA) ...	LO	Algoritmnummer (1-9)	1
Börvärde	2.000	Tid mellan regleringgrepp	1
Nytt värde (JA/NEJ)	NEJ	Alarmskrift, operatörs-	
Blockering (T=Till,F=Från) ...	F	plats (1-6, A=Alla)	A
Blockering grupp (TILL/FRAN)..	FRAN	Alarmskrift, skrivare	
Antal teck. i schema (1-8) ...	1	(T=Till, F=Från)	F
Antal dec. i schema (0-3)	0	Kvitterad (JA/NEJ)	NEJ
Fysikalisk enhet	VOLT	Tidsförd. alarm i sek. (0-255).	0
Börvärde hög gräns	10.000	Alarmgrupp (0-31)	0
Börvärde låg gräns	0.000	Börvärde=ärvärde (0-normal,1-	
Max regleravvikelse i %	5.000	utföres om init=1)	0
Rådande alarmstatus	NORMAL	Uttyp vid kaskad (0-inkremental,	
Min regleringgrepps signal	0.000	utsignal ändring,1-nytt värde)	0
Riktning (0-pos., 1-neg.)	0		

Mätpunktstabell reglerparametrar (2)

Namn	REG001		
Internnr.....	1		
Beskrivande text.....	REGULATOR 1		
Börvärde	2.000	Initiering (0-normal, 1-första)	0
Insignal 1	PUMP01 2.000	Dåtidskompensering (JA/NEJ) ..	NEJ
Insignal 2	0.000	Modellens förstärkningsfaktor .	1.000
Insignal 3	0.000	Modellens tidskonstant	1.000
Utsignal (styrsignal) VENT01	2.000	Dåtiden (tidsfördröjning)	0.000
P-konstant	0.100	Framkoppling (JA/NEJ)	NEJ
I-konstant	1.000	Framkopplingssignal PUMP01	2.000
D-konstant	1.000	P-konstant (framkoppling)	1.000
Kaskadkonstant 1	0.000	I-konstant (framkoppling)	0.000
Kaskadkonstant 2	0.000	D-konstant (framkoppling)	0.000
Gräns för I-undertryckning ..	20.000		
Nedre gräns reglerfel I-del .	20.000		
Övre gräns reglerfel I-del ..	20.000		
Max neg. ändring av utsignal	20.000		
Max pos. ändring av utsignal	20.000		

Figur 4 Mätpunktstabell för reglerparametrar, sid 1 och 2.

5 TESTNING OCH RESULTAT

Här redovisar vi de arbete som utförts lite mera praktiskt. I 5.1 beskrivs testuppkopplingen och hur testningen gick till. 5.2 innehåller resultatet dvs. kurvor på utsignalen vid olika inställningar på regulatorn.

5.1 Testning

För att kunna testa regleralgoritmerna och operatörs-kommunikationen behövde vi en process. Processen simulerades som en första ordningens process med eller utan tidsfördröjning. Den simulerade processmodellen är av samma modell som används i algoritmen för döttidskompensering:

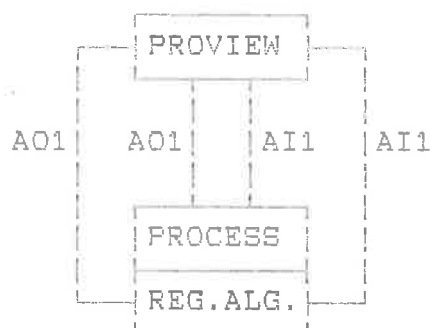
$$\exp(-st)*K/(1+sT)$$

K = processens förstärkning

T = processens tidskonstant

t = processens tidsfördröjning

Processen skrevs som ett huvudprogram som i sin tur anropade reglerprogrammet. Det är bara under testningen som reglerprogrammet fungerade som en subrutin. När reglerprogrammet ingår i COMATOR PROCESS AB:s PROVIEW system är det ett huvudprogram, eftersom man då har tillgång till en process i t ex industrin.



Figur 5 Blockschemat över testuppkopplingen.

Blocket PROCESS och REG.ALG. innehåller den simulerade processen och reglerprogrammet. Detta programpaket kördes med ett intervall på en sekund. PROVIEW innehåller funktioner som tidigare beskrivits i kap. 2.2. Vid testningen användes funktionerna analog ingång (AI), analog utgång (AO), digital reglering (DC), alarm och kurvor. Kurvpaketet i PROVIEW kördes med ett intervall på fem sekunder.

Vid igångkörningen nollställdes insignal (AI1) och styr-signal (AO1). Därefter ändrades DC1:s parametrar till

önskade strängar och numeriska värden, bärvärde och algoritnummer ställdes in, P, I och D - konstanter fick värden osv.

Från processen genererades ärvärdet som lades på AI1 i PROVIEW. Reglerprogrammet använde AI1 som insignal till tidigare vald algoritm. Reglerprogrammet beräknade en styr-signal som lades på AO1 i PROVIEW. Processen använde AO1 som styr-signal till processen varefter nytt ärvärde genererades.

För att bättre se hur ärvärdet ställde in sig vid bärvärdesändringar, stegstörningar och tidsfördröjningar användes under testningen funktionen snabbkurva i PROVIEW. På snabbkurvan ritades ärvärdet (AI1) in.

5.1.1 Körning av testuppkopplingen

Starta PROVIEW med hjälp av `EXEN` på VIC A201,10A och svara ja på följande frågor: Skall dummy ärvärden läggas in i DAT-BAS cykliskt?, Skall larpaketet installeras och startas?, Skall trendkurvepaketet installeras o startas?. Starta kommandofilen DCDEMO.CMD med hjälp av kommandot `DCDEMO` (VIC A201,10A). Denna kommandofil byter ut PROVIEW:s TESCAN mot "vårt" TESTDC (processen). Därefter är det bara att skriva RUN KEXn och demonstrationen kan börja.

5.2 Presentation av kurvor

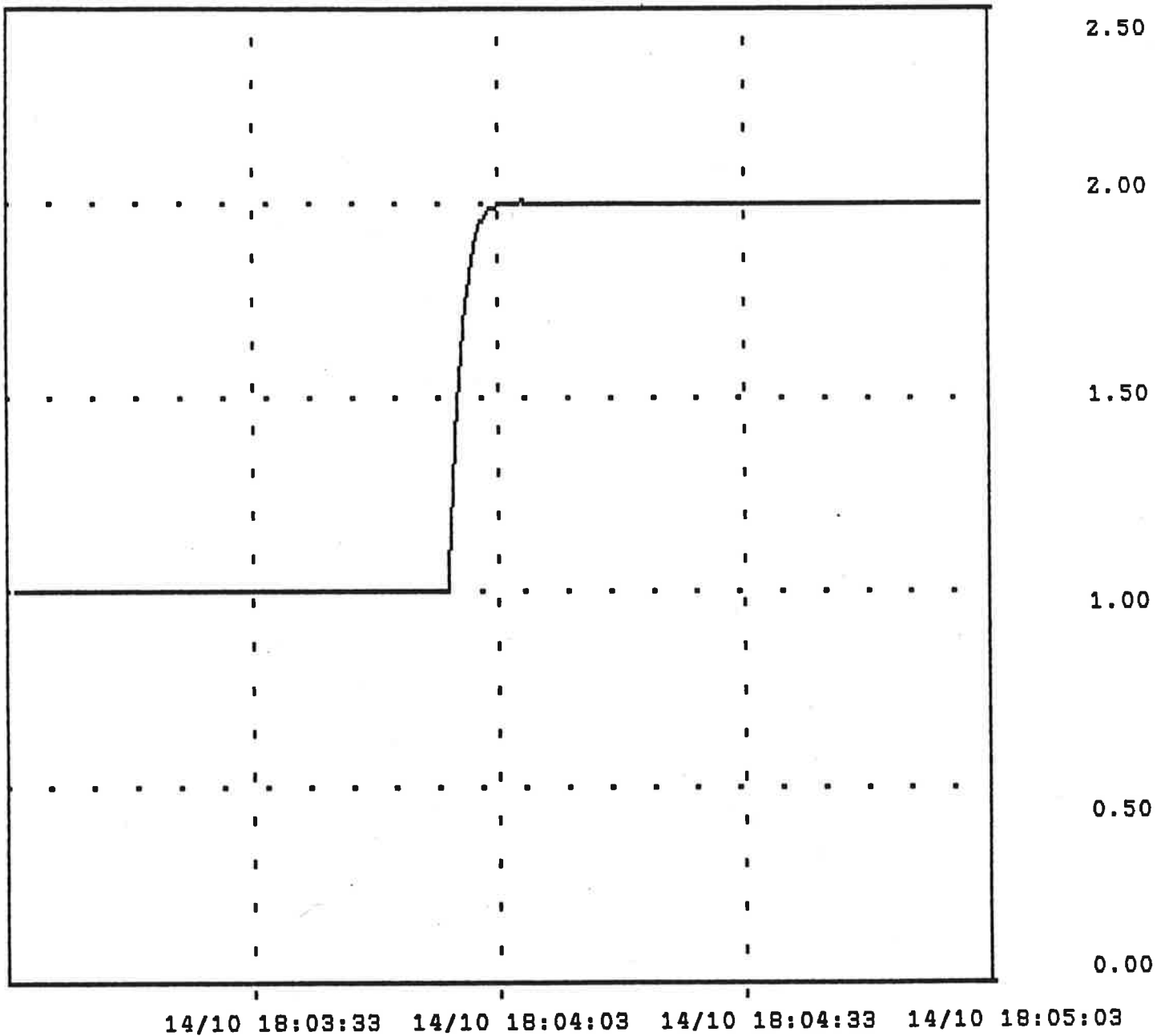
På följande sidor redovisas resultatet av körningar med den simulerade processen styrd av reglerprogrammet. Utsignalen visas för bärvärdesändringar och stegstärningar på utsignalen.

För samtliga försök har samplingsintervallet varit 1 s. Till varje kurva har angivits aktuell inställning av reglerparametrarna, tidsfördröjning och om döttidskompensering använts, vidare ges en kommentar till inställningen av regulatorn.

Kurvgrupp=xxx Tidbas=5 0 tim 2 min

PUMP01

2.00

m/s²

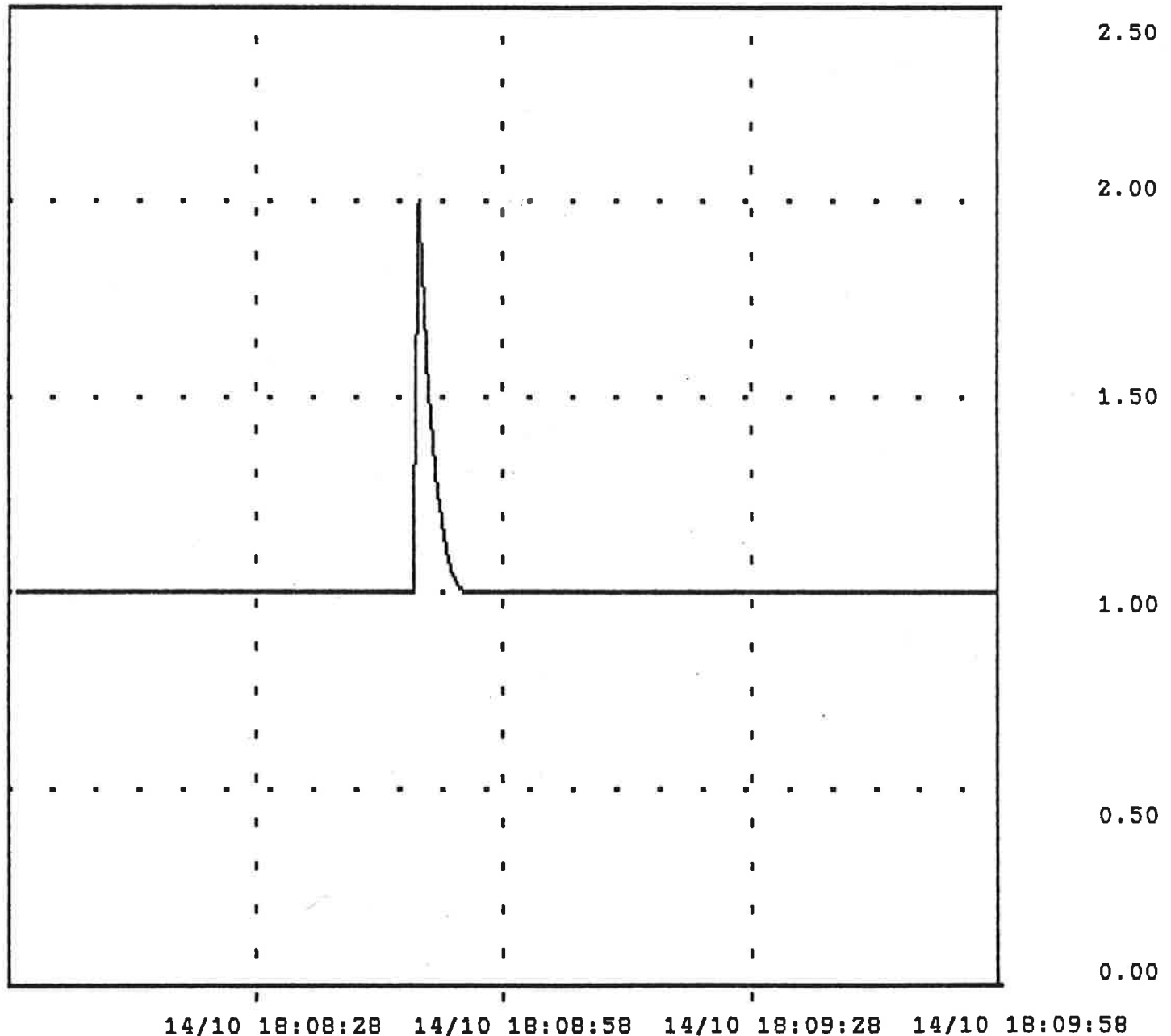
KURVA 1 : algoritm 1 - PID-delta, bärvärdesändring
P = 0.7 , I = 1.4 , D = 0
ingen tidsfördröjning

Bärvärdesändring där största snabbhet med minsta möjliga
översläng har använts som kriterium på bra reglering.

Kurvgrupp=xxx Tidbas=5 0 tim 2 min

PUMP01

1.00

m/s²

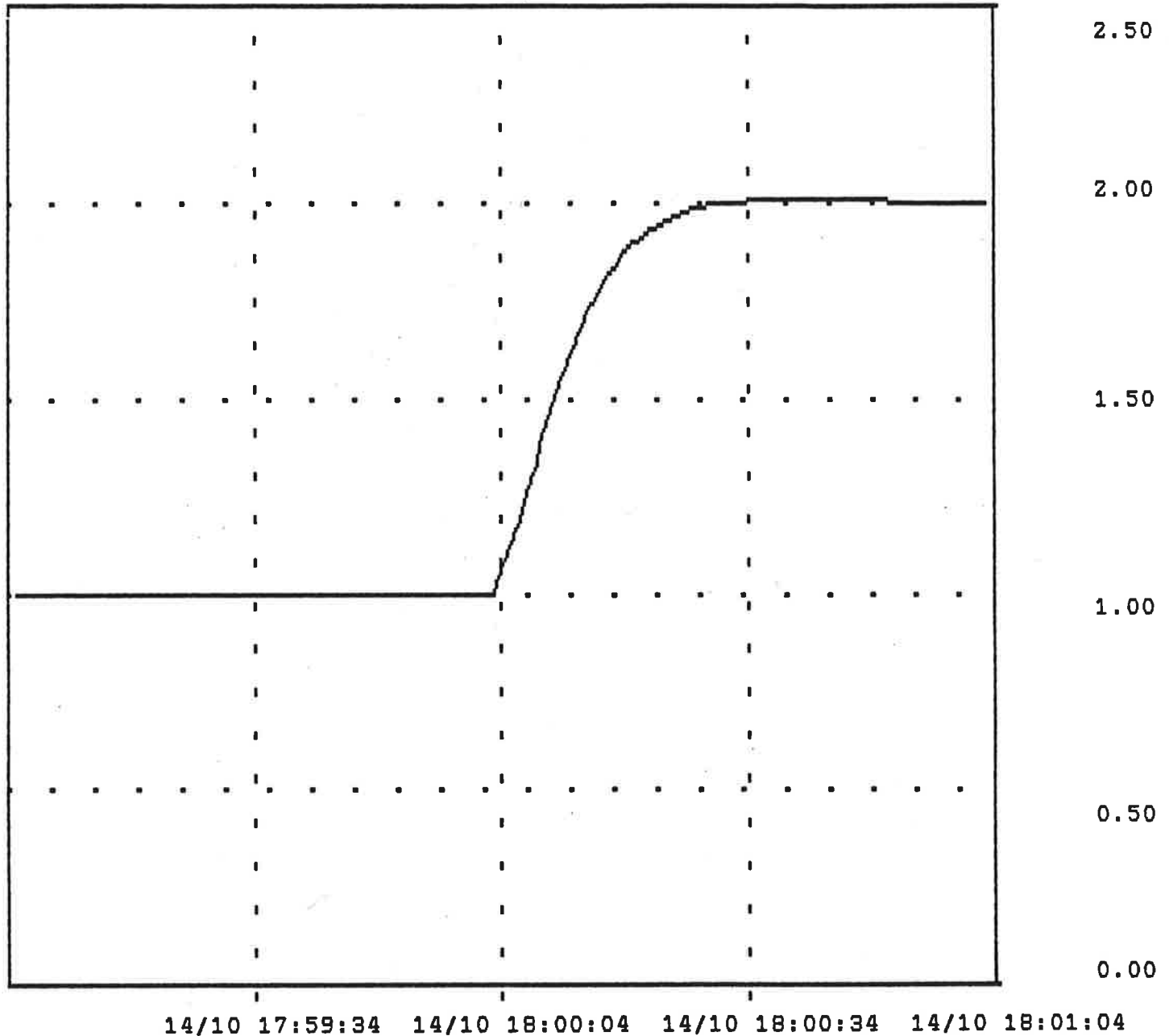
KURVA 2 : algoritm 1 - PID-delta, stegstärning på utsignal
 P = 0.7 , I = 1.4 , D = 0
 ingen tidsfördröjning

Här står vi styrsignalen med ett steg och försöker få en så snabb inställning med minsta möjliga översläng som möjligt.

Kurvgrupp=xxx Tidbas=5 0 tim 2 min

PUMP01

2.00

m/s²

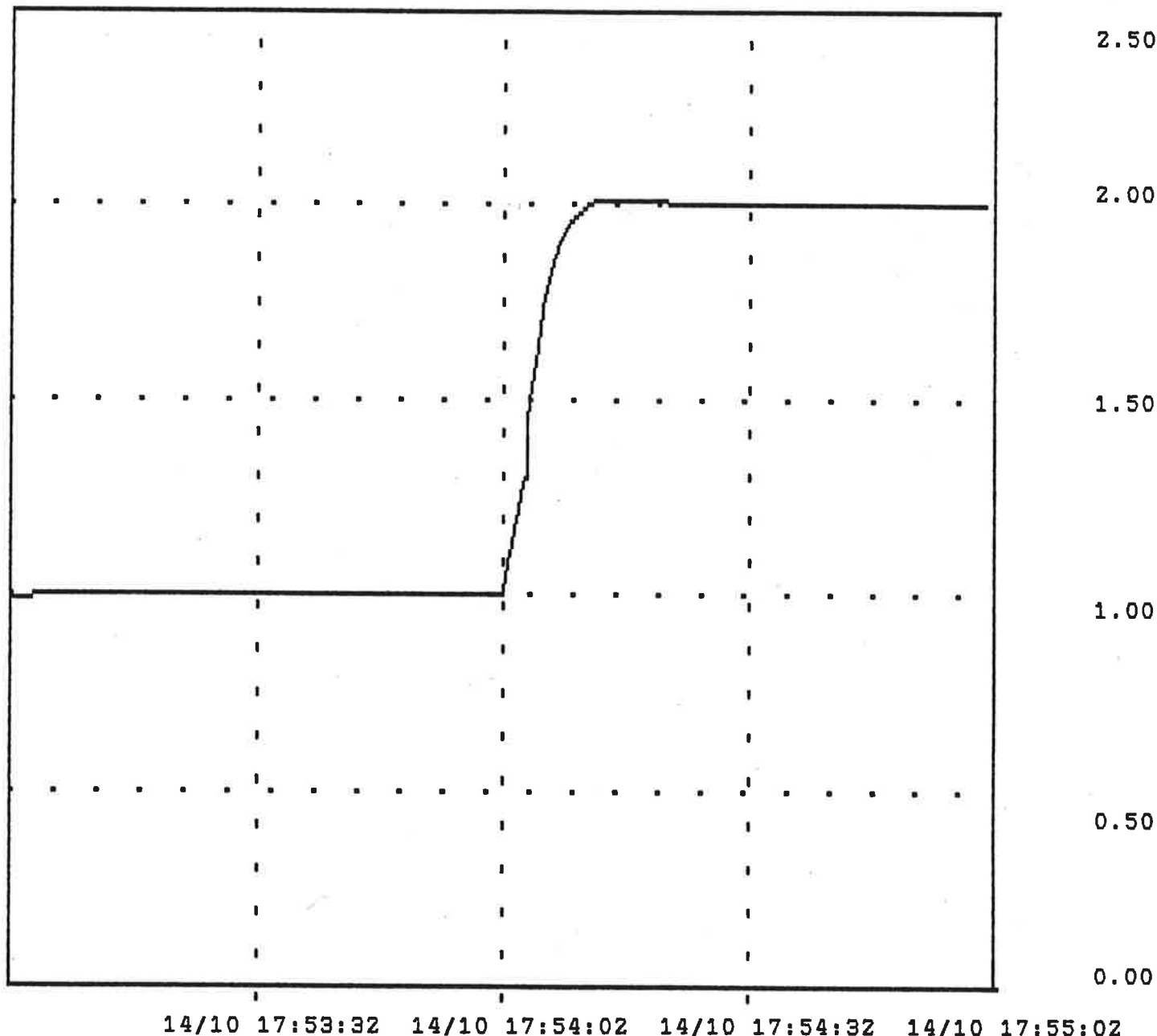
KURVA 3 : algoritm 1 - PID-delta, börvärdesändring
 P = 0.085 , I = 1.2 , D = 0
 tidsfördröjning: 5s
 ingen döttidkompensering

Börvärdesändring men nu med en tidsfördröjning på 5s hos processen. Här har ingen döttidkompensering använts. Förstärkningen måste minskas för att överslängen skall bli så liten som möjligt.

Kurvgrupp=xxx Tidbas=5 0 tim 2 min

PUMP01

2.00

m/s²

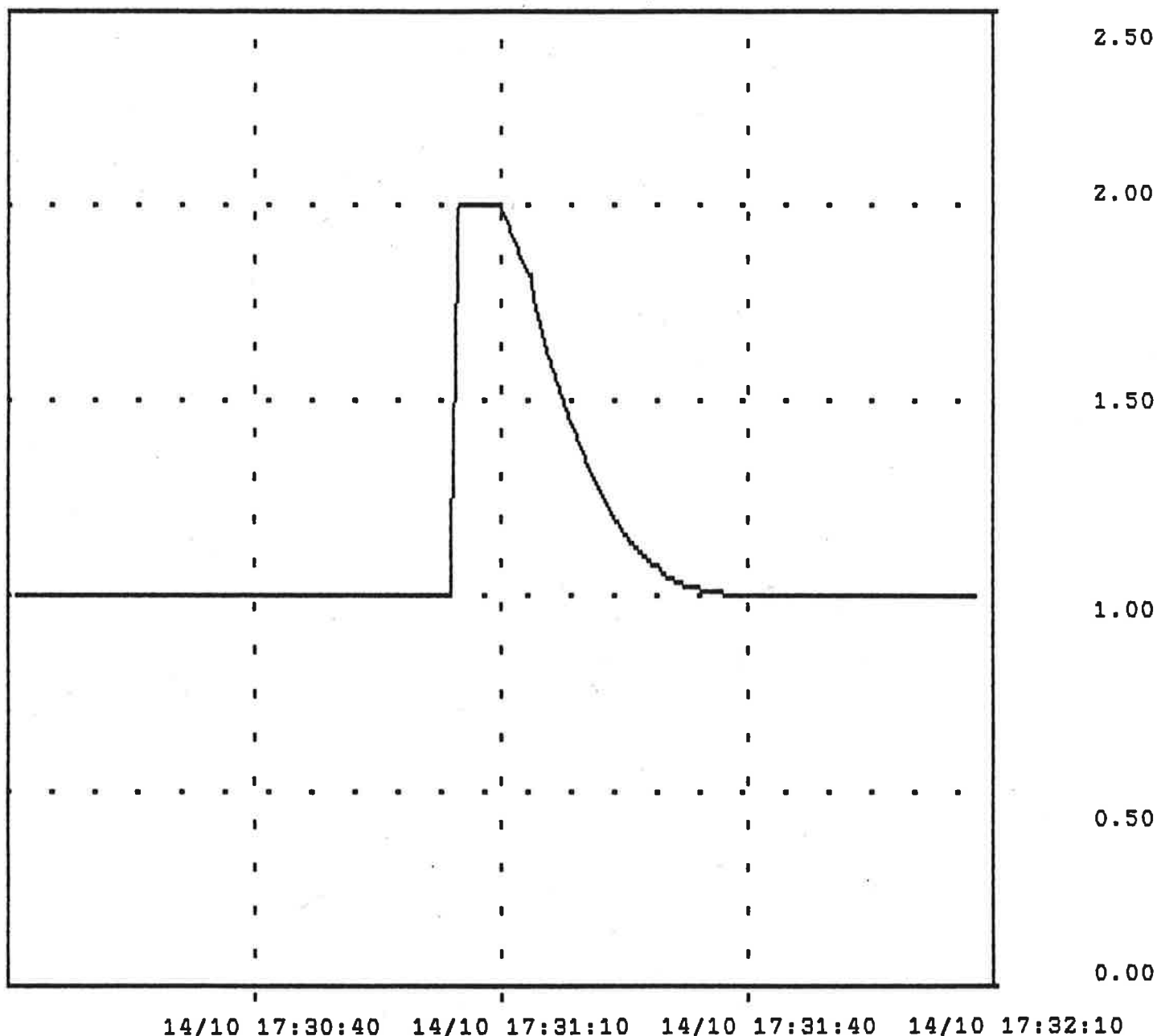
KURVA 4 : algoritm 1 - PID-delta, bärvärdesändring
 P = 0.2 , I = 1.0 , D = 0
 tidsfördröjning: 5s
 dædtidskompensering

Även här en bärvärdesändring med tidsfördröjningen 5s.
 Nu används dædtidskompenseringen och för att få en jämför-
 bar kurva med föregående (samma översläng) har förstärkning-
 en ökats. Vi ser att snabbheten har ökat åtminstone med
 en faktor 2.

Kurvgrupp=xxx Tidbas=5 0 tim 2 min

PUMP01

1.00

m/s²

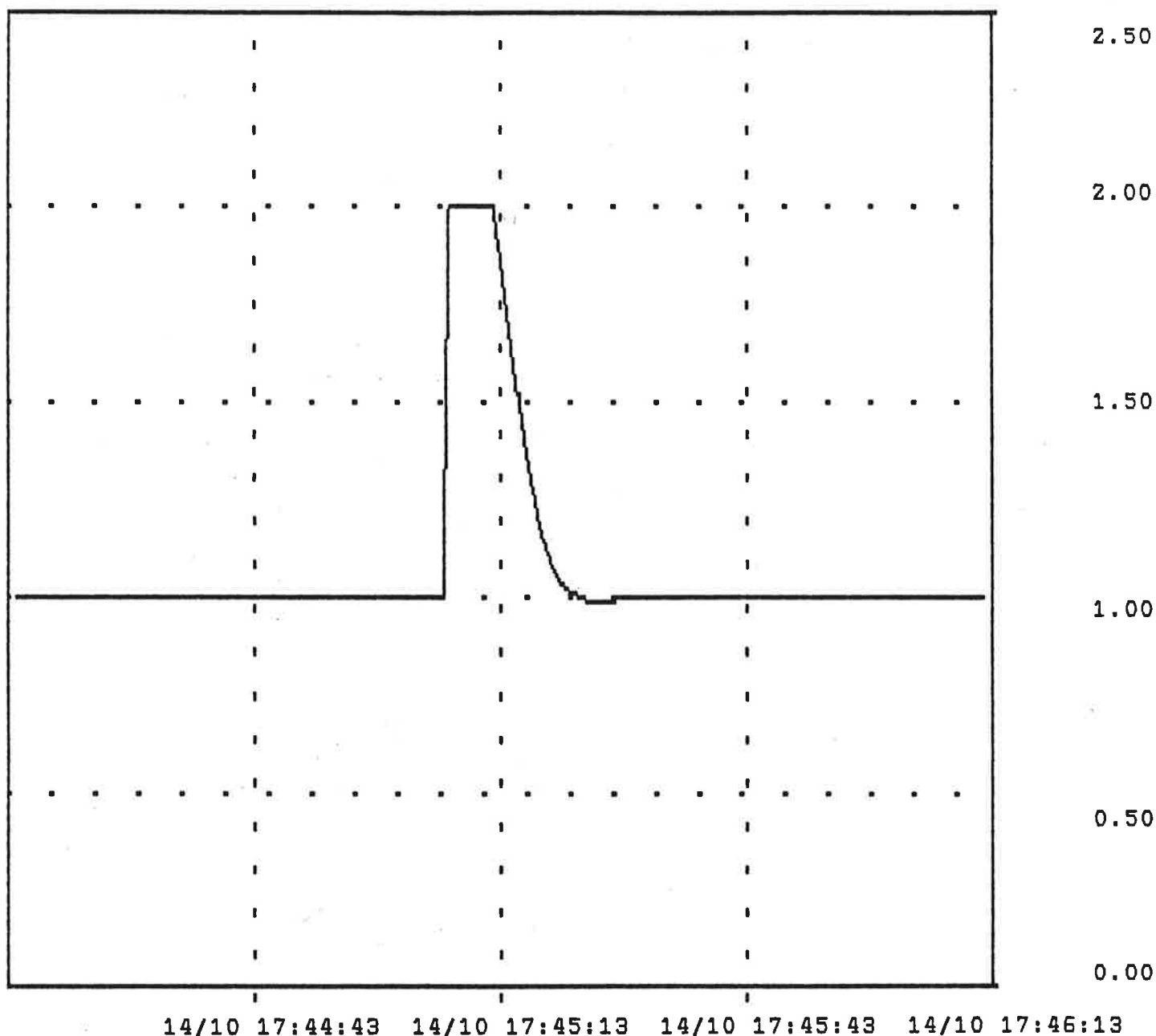
KURVA 5 : algoritm 1 - PID-delta, stegstörning på utsignal
P = 0.085 , I = 1.2 , D = 0
tidsfördröjning: 5s
ingen döttidkompensering

Stegstörning på utsignalen med en tidsfördröjning hos processen på 5s. Här har ingen döttidkompensering använts.

Kurvgrupp=xxx Tidbas=5 0 tim 2 min

PUMP01

1.00

m/s²

KURVA 6 : algoritm 1 - PID-delta, stegstyrning på utsignal
 P = 0.2 , I = 1.0 , D = 0
 tidsfördröjning: 5s
 dædtidskompensering

Som föregående men med dædtidskompensering, precis som i fallet med bærværdesændring har øverslængen studerats vid instællningen. Førstærkningen har øskats jæmført med fallet dæ ingen dædtidskompensering anvænds, en økning av snæbbheten med en faktor 2 kan iækttagas æven hær.


```

CCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCCC
CC
CC      PSEUDOKOD
CC
Cé DO  900  Reglerlooparna genomlöps
Cé  IF ej blockering THEN  fortsätt
Cé  ELSE nästa regulator
Cé  IF första gång börja
Cé  ELSE räkna ner reglertiden
Cé  IF signalen klar att reglera , läs in denna
Cé  ELSE börja om
Cé  PID konstanterna ,utadressen och reglertiden läses in
Cé  IF borvardet större än max
Cé  THEN  BORV = MAX
Cé  IF borvardet mindre än min
Cé  THEN  BORV = MIN
Cé  Test om I-undertryckning
Cé  Packar upp intyp och inadress
Cé  Algoritnumret läses in
Cé  Nollställning av dottidsvariabler
Cé  Tidsfördröjningen kontrolleras
Cé  Hopp till rätt algoritm
Cé  ALG:1  PID - regulator delta
Cé      Test om dottidskompensering
Cé      I-delen  :   $I = P * T / I$ 
Cé      IF I-undertryckning utför denna
Cé      D-delen  :   $D = D * P / T$ 
Cé      Test om framkoppling
Cé      Utsignalen byggs samman
Cé  ALG:2  PID - regulator absolut
Cé      Test om dottidskompensering
Cé      I-delen
Cé      IF I-undertryckning utför denna
Cé      D-delen
Cé      Test om framkoppling
Cé      Utsignalen byggs samman

```


6.2 Appendix 2: DATABAS SUBROUTINER för DC

1	= Externnummer	ASCII	6
2	= Beskrivning	ASCII	30
3	= Börvärde	REAL	
4	= Blockering	BIT	1
5	= Blockering (grupp)	BIT	1
6	= Alarmgrupp	BIT	5
7	= Tidsfördröjning	BYTE	
8	= Alarmutskrift rapport	BIT	1
9	= Kvitterad	BIT	1
10	= Ingenjörsenhet (6)	ASCII	6
11	= Operatörsplats	BIT	3
12	= Alarmstatus	BIT	3
13	= Nytt värde	BIT	1
14	= Antal decimaler i schema	BIT	2
15	= Initiering	BIT	1
16	= Antal tecken i schema	BIT	3
17	= Börvärde = ärvärde	BIT	1
18	= Uttyp vid kaskad	BIT	1
19	= Nedre gräns reglersignal I-del	REAL	
20	= Övre gräns reglersignal I-del	REAL	
21	= Börvärde hög gräns	REAL	
22	= Börvärde låg gräns	REAL	
23	= P-konstant	REAL	
24	= I-konstant	REAL	
25	= D-konstant	REAL	
26	= Kaskadkonstant 1	REAL	
27	= Kaskadkonstant 2	REAL	
28	= Gräns för I-undertryckning	REAL	
29	= Typ och internindex för utsignal	INTEGER	
31	= Minsta regleringrepps signal	REAL	
32	= Tid mellan regleringrepp	BYTE	
33	= Algoritmnummer	BIT	4
34	= Typ av mätpunkt	BIT	2
35	= Max negativ ändring	REAL	
36	= Max positiv ändring	REAL	
37	= Riktning	BIT	1
38	= Typ och internindex för insignal 3	INTEGER	
41	= Framkoppling	BIT	1
43	= P-konstant (framkoppling)	REAL	
44	= I-konstant (framkoppling)	REAL	
45	= D-konstant (framkoppling)	REAL	
46	= Typ och internindex för framkopplings- signal	INTEGER	
48	= Dödtidskompensering	BIT	1
49	= Förstärkningsfaktor	REAL	
50	= Tidskonstant	REAL	
51	= Dödtid (tidsfördröjning)	REAL	
52	= Typ och internindex för insignal 1	INTEGER	
54	= Typ och internindex för insignal 2	INTEGER	
56	= Max regleravvikelse i %	REAL	

1 Externnummer

6 stycken alfanumeriska tecken lagrade i ASCII-format

2 Beskrivning

30 stycken alfanumeriska tecken lagrade i ASCII-format som fungerar som en verbal beskrivning av punkten.

3 Bärvärde

Gällande bärvärde

4 Blockering

Individuell blockering av punkten.

0 = Reglering

1 = Ingen reglering

5 Blockering (grupp)

Gruppvis blockering av punkten. All alarmövervakning kopplas bort och ingen omvandling av inläst värde till ingenjörstorheter sker så länge gruppen är blockerad.

6 Alarmgrupp

0 = Punkten tillhör ingen alarmgrupp

1-16 = Alarmgrupp som punkten är kopplad till. Vid blockering av alarmgruppen blockeras denna punkt och vid deblockering av gruppen plockas blockerings-signalen bort. Den individuella blockeringen påverkas ej.

7 Tidsfördröjning

Tidsfördröjning i antal sekunder som ett alarm måste vara aktivt innan ett alarm-meddelande genereras (0-255).

8 Alarmutskrift rapport

0 = Alarmmeddelande går både till bildskärm och skrivare.

1 = Alarmmeddelande går endast till bildskärm.

9 Kvitterad

0 = Okvitterat alarm

1 = Kvitterat alarm

10 Ingenjörsenhet (6)

6 stycken alfanumeriska tecken lagrade i ASCII-format.

11 Operatörsplats

0 = Alarmmeddelande går ut till samtliga operatörsplatser.

1-6 = Alarmmeddelandet går endast till anvisad operatörsplats.

12 Alarmstatus

0 = Normal status

1-4 = Reserv

5 = Alarm hög-hög gräns

6-7 = Reserv

13 Nytt värde

1 = Nytt värde har lästs in

14 Antal decimaler i schema

Antal decimaler som börvärdet skall presenteras med i process-schema (0-3).

15 Initiering

Kan ej påverkas

0 = Normaldrift

1 = Första gång efter omkoppling ej dator till dator eller efter fel på insignal.

16 Antal tecken i schema

Antal tecken som börvärdet skall presenteras med i process-schema (1-8).

(Inkl. decimalpunkt och eventuellt inledande tecken)

17 Börvärde = ärvärde

0 = Ingen effekt

1 = Sättes börvärde = ärvärde när initiering sätts till 1

18 Uttyp vid kaskad

Endast vid kaskad-loopar

0 = Inkremental, utsignaländring

1 = Ersätter tidigare värde

19 Nedre gräns reglersignal I-del

För att eliminera wind-up på reglersignalen från I-delen så sättes den till min signal om den skulle försöka gå förbi denna gräns (gäller alg. 2).

20 Övre gräns reglersignal I-del

För att eliminera wind-up på reglersignalen från I-delen så sättes den till max signal om den skulle försöka gå förbi denna gräns (gäller alg. 2).

21 Börvärde hög gräns

Max.-värde för börvärde

22 Börvärde låg gräns

Min.-värde för börvärde

23 P-konstant

Proportionalkonstant vid reglering

24 I-konstant

Integralkonstant vid reglering

25 D-konstant

Deriveringskonstant vid reglering

26 Kaskadkonstant 1

Kaskadkonstant vid kaskadreglering

Om denna parameter är noll, ingen kaskad (jämför 29).

27 Kaskadkonstant 2

Kaskadkonstant vid kaskadreglering

28 Gräns för I-undertryckning

Om avvikelserna mellan är- och bärvärde är större än denna parameter, kopplas I-termen bort från regleringen.

29 Typ och internindex för utsignal

K1 = 0 = Utmatningsadress till AO, ej kaskad

K1 ej 0 = Adress till AI där resultatet skall läggas, kaskad

31 Min regleringreppssignal

Minsta regleringreppssignal som skall matas ut

Det avlästa råa värdet före omvandling och filtrering.

32 Tid mellan regleringrepp

Tid mellan regleringrepp

33 Algoritmnummer

Ett tal mellan 1 och 9

34 Typ av bärvärde

Typ av bärvärde.

0 = LO bärvärdet ställs in lokalt

1 = BE bärvärdet beräknas externt

2 = MA bärvärdet ställs in manuellt

35 Max negativ ändring

Största negativa regleringreppssignal

36 Max positiv ändring

Största positiva regleringreppssignal

37 Riktning

0 = Positiv

1 = Inverterad riktning

38 Typ och internindex för insignal 3

Adress för generell insignal

41 Framkoppling

0 = Ingen framkoppling

1 = Framkoppling

43 P-konstant (framkoppling)

Proportionalkonstant vid framkoppling

44 I-konstant (framkoppling)

Integralkonstant vid framkoppling

45 D-konstant (framkoppling)

Deriveringskonstant vid framkoppling

46 Typ och internindex för framkopplingssignal

Adress för signal som skall framkopplas

48 Dödtidskompensering

0 = Ingen dödtidskompensering

1 = Dödtidskompensering

49 Förstärkningsfaktor

Modellens förstärkningsfaktor vid dödtidskompensering

50 Tidskonstant

Modellens tidskonstant vid dödtidskompensering

51 Dödtid (tidsfördröjning)

Tidsfördröjning i processen vid dödtidskompensering

52 Typ och internindex för insignal 1

Adress för insignal 1

54 Typ och internindex för insignal 2

Adress för insignal 2

56 Max regleravvikelse i %

Högst tillåtna regleravvikelse i % innan ett *A*-alarm genereras

6.2.1 DCR : hämta reell parametrar i DC-tabell

Namn: DCR

Uppgift: Hämta parameter i DC-tabell som reellt tal

Anrop: VALUE = DCR (INTNR,PARAM)

INTNR = internpunktnummer

PARAM = önskad parameter

3	=	Börvärde	REAL
19	=	Nedre gräns reglersignal I-del	REAL
20	=	Övre gräns reglersignal I-del	REAL
21	=	Börvärde hög gräns	REAL
22	=	Börvärde låg gräns	REAL
23	=	P-konstant	REAL
24	=	I-konstant	REAL
25	=	D-konstant	REAL
26	=	Kaskadkonstant 1	REAL
27	=	Kaskadkonstant 2	REAL

28 = Gräns för I-undertryckning	REAL
31 = Minsta regleringrepps signal	REAL
35 = Max negativ ändring	REAL
36 = Max positiv ändring	REAL
43 = P-konstant (framkoppling)	REAL
44 = I-konstant (framkoppling)	REAL
45 = D-konstant (framkoppling)	REAL
49 = Förstärkningsfaktor	REAL
50 = Tidskonstant	REAL
51 = Dödtid (tidsfördröjning)	REAL
56 = Max regleravvikelse i %	REAL

VALUE = svaret som reellt tal

Funktion: Rutinen hämtar begärt värde i reellt format

6.2.2 DCRLA : ändra reell parameter i DC-tabell

Namn: DCRLA

Uppgift: Lagra ett reellt tal i DC-tabell

Anrop: CALL DCRLA (INTNR,PARAM,VALUE)

INTNR = internpunktnummer

PARAM = önskad parameter

3 = Bärvärde	REAL
19 = Nedre gräns reglersignal I-del	REAL
20 = Övre gräns reglersignal I-del	REAL
21 = Bärvärde hög gräns	REAL
22 = Bärvärde låg gräns	REAL
23 = P-konstant	REAL
24 = I-konstant	REAL
25 = D-konstant	REAL
26 = Kaskadkonstant 1	REAL
27 = Kaskadkonstant 2	REAL
28 = Gräns för I-undertryckning	REAL
31 = Minsta regleringrepps signal	REAL
35 = Max negativ ändring	REAL
36 = Max positiv ändring	REAL
43 = P-konstant (framkoppling)	REAL
44 = I-konstant (framkoppling)	REAL
45 = D-konstant (framkoppling)	REAL
49 = Förstärkningsfaktor	REAL
50 = Tidskonstant	REAL
51 = Dödtid (tidsfördröjning)	REAL
56 = Max regleravvikelse i %	REAL

VALUE = det nya värdet som reellt tal

Funktion: Rutinen lagrar värdet i reellt format

6.2.3 DCI : hämta integer parameter i DC-tabell

Namn: DCI

Uppgift: Hämta parameter i DC-tabell som heltal

Anrop: IVALUE = DCI (INTNR,PARAM)

INTNR = internpunktnummer

PARAM = önskad parameter

4	= Blockering	BIT 1
5	= Blockering (grupp)	BIT 1
6	= Alarmgrupp	BIT 5
7	= Tidsfördröjning	BYTE
8	= Alarmutskrift rapport	BIT 1
9	= Kvittebad	BIT 1
11	= Operatörsplats	BIT 3
12	= Alarmstatus	BIT 3
13	= Nytt värde	BIT 1
14	= Antal decimaler i schema	BIT 2
15	= Initiering	BIT 1
16	= Antal tecken i schema	BIT 3
17	= Bärvärde = ärvärde	BIT 1
18	= Uttyp vid kaskad	BIT 1
29	= Typ och internindex för utsignal	INTEGER
32	= Tid mellan regleringrepp	BYTE
33	= Algoritnummer	BIT 4
34	= Typ av mätpunkt	BIT 2
37	= Riktning	BIT 1
38	= Typ och internindex för insignal 3	INTEGER
41	= Framkoppling	BIT 1
46	= Typ och internindex för framkopplingssignal	INTEGER
48	= Dödtidskompensering	BIT 1
52	= Typ och internindex för insignal 1	INTEGER
54	= Typ och internindex för insignal 2	INTEGER

IVALUE = svaret som heltal

Funktion: Rutinen hämtar begärt värde i integer format

6.2.4 DCILA : ändra integer parameter i DC-tabell

Namn: DCILA

Uppgift: Lagra ett heltal i DC-tabell

Anrop: CALL DCILA (INTNR,PARAM,IVALUE)

INTNR = internpunktnummer

PARAM = önskad parameter

4	= Blockering	BIT 1
5	= Blockering (grupp)	BIT 1
6	= Alarmgrupp	BIT 5
7	= Tidsfördröjning	BYTE
8	= Alarmutskrift rapport	BIT 1
9	= Kvitte	BIT 1
11	= Operatörsplats	BIT 3
12	= Alarmstatus	BIT 3
13	= Nytt värde	BIT 1
14	= Antal decimaler i schema	BIT 2
15	= Initiering	BIT 1
16	= Antal tecken i schema	BIT 3
17	= Börvärde = ärvärde	BIT 1
18	= Uttyp vid kaskad	BIT 1
29	= Typ och internindex för utsignal	INTEGER
32	= Tid mellan regleringrepp	BYTE
33	= Algoritnummer	BIT 4
34	= Typ av börvärde	BIT 2
37	= Riktning	BIT 1
38	= Typ och internindex för insignal 3	INTEGER
41	= Framkoppling	BIT 1
46	= Typ och internindex för framkopplingssignal	INTEGER
48	= Dätidskompensering	BIT 1
52	= Typ och internindex för insignal 1	INTEGER
54	= Typ och internindex för insignal 2	INTEGER

IVALUE = det nya värdet som heltal

Funktion: Rutinen lagrar värdet i integerformat

6.2.5 DCS : hämta datasträng parameter i DC-tabell

Namn: DCS

Uppgift: Hämta parameter i DC-tabell som datasträng.

Anrop: CALL DCS (INTNR,PARAM,AREA)

INTNR = internpunktnummer

PARAM = önskad parameter

1	= Externnummer	ASCII 6
2	= Beskrivning	ASCII 30
10	= Ingenjörsenhet (6)	ASCII 6

AREA = area där datasträngen skall lagras.

Funktion: Rutinen hämtar begärd datasträng.

6.2.6 DCSLA : ändra datasträng parameter i DC-tabell

Namn: DCSLA

Uppgift: Lagra en datasträng i DC-tabellen

Anrop: CALL DCSLA (INTNR,PARAM,AREA)

INTNR = internpunktnummer

PARAM = önskad parameter

1	= Externnummer	ASCII 6
2	= Beskrivning	ASCII 30
10	= Ingenjörsenhet (6)	ASCII 6

AREA = den nya datasträngen

Funktion: Rutinen lagrar datasträngen i DC-databasen.

5.3 Appendix 3: Verifiering av lead/lag - reglering

En härledning av de i programmet använda satserna för lead- respektive lag-reglering görs nedan.

$DN = DN1 + TERM/I*(VAL1-DN1)$ i programmet skrives som:

$D(n) = D(n-1) + h/I(u(k) - D(n-1))$ där h och TERM står

för samplingsintervallet, I för integrationskonstant och D(index) för delresultat vid beräkningar.

$$\Rightarrow D(n) (1 - q^{-1} + h/I * q^{-1}) = h/I * u(k)$$

$$\Leftrightarrow D(n) ((1 - q^{-1})/h + h/I * (q^{-1} - 1)/h + h/I) = 1/I * u(k)$$

övergång till kontinuerligt system \Rightarrow

$$D(n) = 1/(s(I-h) + h) * u(k)$$

$CN = D*(DN-DN1) + DN$ i programmet skrives som:

$$C(n) = D (1 - q^{-1}) D(n) + D(n) \text{ där } D \text{ är derivatakonstanten}$$

och C(index) är delresultat vid beräkningar.

övergång till kontinuerligt system \Rightarrow

$$C(n) = (sDh + 1) D(n)$$

utsignal : $YY = P * C(n) \Rightarrow$

$$YY = P * (sDh + 1) / (s(I-h) + h) * u(k).$$

6.4 Appendix 4: Dödtidskompensering, härledning

Vi har ansatt en kontinuerlig processmodell av första ordn:

$$GM(s) = \exp(-sT) * K / (1 + sT_1)$$

K : modellens förstärkning

T₁ : modellens tidskonstant

T : dödtiden (tidsfördröjningen)

Det kontinuerliga systemet blir då:

$$\begin{cases} \dot{x}(t) = -1/T_1 \cdot x(t) + u(t-T) \\ y(t) = K/T_1 \cdot x(t) \end{cases}$$

Systemets digitala motsvarighet blir:

$$\begin{cases} x(kh+h) = \Phi \cdot x(kh) + \Gamma_0 \cdot u(kh-dh+h) + \Gamma_1 \cdot u(kh-dh) \\ y(kh) = C \cdot x(kh) \end{cases}$$

där h är samplingsintervallet och d beräknas enligt:

$$T = (d-1)h + T' \quad , \quad 0 < T' \leq h$$

$$C = K/T_1$$

$$\Phi = \exp(-h/T_1)$$

$$\begin{aligned} \Gamma_1 &= \exp(-(h-T')/T_1) * \int_0^{T'} \exp(-s/T_1) ds = \\ &= T_1 * (\exp(-(h-T')/T_1) - \exp(-h/T_1)) \end{aligned}$$

$$\Gamma_0 = \int_0^{h-T'} \exp(-s/T_1) ds = T_1 * (1 - \exp(-(h-T')/T_1))$$

$$\Rightarrow y(kh) = C (qI - \Phi) (\Gamma_0 q^{-1} + \Gamma_1 q^{-d}) u(kh)$$

$$\Leftrightarrow y(kh) (qI - \Phi) = C (\Gamma_0 q^{-d} + \Gamma_1 q^{-1}) u(kh)$$

$$\Leftrightarrow y(kh+h) = \Phi * y(kh) + C \Gamma_0 * u(kh-dh+h) + C \Gamma_1 * u(kh-dh)$$

med följande beteckningar använda vid beskrivningen:

$$AM = \Phi$$

$$BM1 = C \Gamma_1$$

$$BM2 = C \Gamma_0$$

7 REFERENSER

Teorin till rapporten har hämtats ur litteratur från institutionen för reglerteknik och från universitets biblioteket 2:

Aström, Karl, Johan, Reglerteori 1976

Aström, Karl, Johan, REGLERTEKNIK en elementär introduktion
kap. 5, 6, 8

Aström, Karl, Johan, Wittenmark, Björn, COMPUTER CONTROLLED
SYSTEM Theory and Design

Luyben, Process Modelling, Simulation and control for
chemical engineers, kap. 13

För inlärnin g av datorsystemet och operativsystemet användes följande litteratur:

Digital Equipment Corporation,

PDP-11 FORTRAN-77
Language Reference Manual

PDP-11 FORTRAN-77
User's Guide

Introduction to RSX-11M and RSX-11M-PLUS

RSX-11M/M-PLUS Utilities and Commands
A Self-Paced Course
Volume 1,2, Exercises and Solutions

Inom COMATOR PROCESS AB finns speciella rutiner för att underlätta arbetet dokumenterat:

COMATOR PROCESS AB, Interna Dokument

Slutligen skall också nämnas att vid programskrivningen användes:

Ekman Torgil, Eriksson Göran, Programmering i Fortran 77
1980