CODEN: LUTFD2/(TFRT-5222)/1-084/(1979)

AUTOMATISK INSTÄLLNING AV PID-REGULATORER BASERAD PÅ OPTIMERING OCH IDENTIFIERING

POUL KONGSTAD

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA
AUGUSTI 1979

AUTOMATISK INSTÄLLNING AV PID-REGULATORER BASERAD PÅ OPTIMERING OCH IDENTIFIERING

POUL KONGSTAD
520911-4072

EXAMENSARBETE VID
INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA

1979

Handledare:
Björn Wittenmark

ad Nilo

Dokumentutgivare
Lund Institute of Technology
Department of Automatic Control

Handläggare
Björn Wittenmark

Författare
Poul Kongstad

Dokumentnamn
REPORT LUTFD2/(TFRT-5222)/1-084/(1979)

Utgivningsdatum
August 1979

Dokumentbeteckning

Ärendebeteckning
06T6

10T4

Dokumenttitel och undertitel
Automatisk inställning av PID-regulatorer baserad på optimering och identifiering.

(Automatic tuning of PID-controllers using optimization and identification)

Referat (sammandrag)
This master thesis presents a general structure for the optimization of regulator parameters. It is assumed that control structure and control aim (given as a loss function) are chosen. The process that is controlled need not be known; identification and optimization are performed simultaneously. Processes containing pure time delay can be optimized with only minor program changes.

This search method has been applied to PI- and PID-regulators and processes of moderate order and have been programmed in Simnon, a simulation language. Actual results from the computer simulations using different loss functions and programs are included. The report also contains discussions of the choice of loss function and identification in closed loop systems.

Referat skrivet av
author

Förslag till ytterligare nyckelord
44T0

Klassifikationssystem och -klass(er)
50T0

Indextermer (ange källa)
52T0

Omfång
84 pages

Övriga bibliografiska uppgifter
56T2

Språk
Swedish

Sekretessuppgifter
60T0

ISSN
60T4

ISBN
60T6

Dokumentet kan erhållas från
Department of Automatic Control
Lund Institute of Technology
Box 725, S-220 07 LUND 7, Sweden

Mottagarens uppgifter
62T4

Pris
66T0

DOKUMENTDATABLAD enligt SIS 62 10 12

SIS-DB 1

Blankett LU 11:25 1976–07

INNEHÅLLSFÖRTECKNING

1. PROBLEMFORMULERING 1

2. PID-REGULATORN 4

3. INSTÄLLNING AV REGULATORER - ÖVERSIKT 7

3.1 Metodiker och problem 8
3.2 Några inställningsmetoder 11

4. SÖKSYSTEMETS UPPBYGGNAD 16

4.1 Blockschema 16
4.2 Synkronisering av process- och minloop 18

5. IDENTIFIERING AV PROCESSEN 21

5.1 Identifieringsproblemet 21
5.2 Identifieringsrutinen 23

6. FÖRLUSTFUNKTIONER 26

6.1 Principer 27
6.2 Några vanliga förlustfunktioner 30
6.3 Andra typer av förlustfunktioner 33
6.4 Parametrisering av regulatorn 34

7. OPTIMERINGSRUTINER 35

7.1 Allmänt 35
7.2 Några använda rutiner 36

8. SÖKSYSTEMETS FÖRVERKLIGANDE 39

8.1 Simuleringsprogrammen 39
8.2 Val av parametrar 49

9. SIMULERINGAR 55

9.1 Dator och processer 55
9.2 Ett optimeringsexempel 56
9.3 Några simuleringsresultat 59
9.4 Några praktiska synpunkter 65

10. SLUTSATSER OCH UTVECKLINGSRIKTNINGAR 67

10.1 Sammanfattning och resultat 67
10.2 Modifieringar och utvecklingsriktningar 69

"Ord använder man till tre ting: att uttrycka tankar; att dölja tankar; att dölja ett det inte finns några tankar." ANTIK FILOFOF

1. PROBLEMFORMULERING

Syftet med denna rapport är att beskriva ett system som söker lämpliga inställningar för de tre regulatorparametrarna i en PID-regulator.

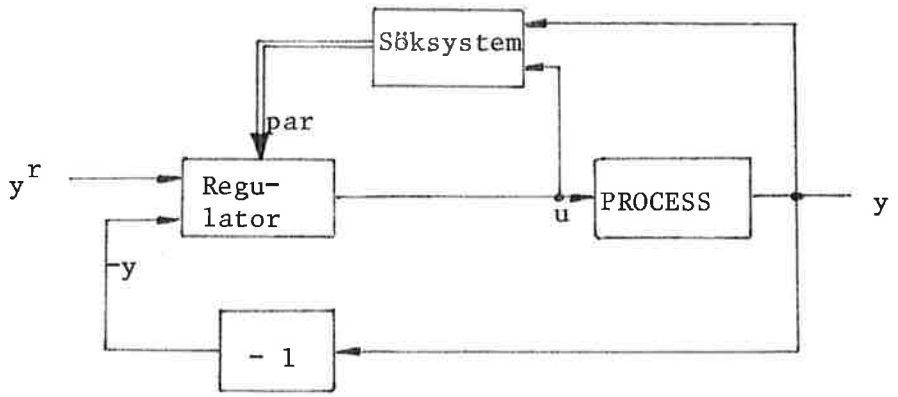


Fig 1.1 Skiss av ett söksystems omgivning

Den aktuella processen regleras hela tiden av sin PID-regulator. Under inställningsproceduren görs några ändringar i börvärdet y^r, i övrigt arbetar söksystemet utan att störa processen.

Sökning efter bra regulatorparametrar sker genom att minimera en förlustfunktion (som är ett mått på hela systemets prestanda) genom snabba simuleringar i en modell av process + regulator.

De processer som förekommer antas ha måttligt oscillativt stegsvar och en insignal, en utsignal. De förutsätts styr- och observerbara. Processens överföringsfunktion antas ej känd, men bör kunna approximeras med en högst 5. ordningens linjär process, eventuellt innehållande en ren tidsfördröjning. Omgivning och process antas konstanta eller bara ändra sig långsamt i tiden. Verkliga processer är sällan linjära, men det antas att en linjariserad modell kan användas vid den aktuella arbetspunkten. Byte av arbetspunkt kan då medföra att optimeringen får upprepas om man har höga prestandakrav. Måttliga störningar och brus

får förekomma.
Det system som här skisseras kommer att kunna vara till hjälp då man justerar in en regulator för en okänd process, en vanlig situation.
Då man funnit regulatorparametrar som ger tillfredsställande processuppförande fixeras parametrarna och söksystemet bortkopplas. Eftersom söksystemet realiseras som datorprogram är förnyad optimering enkel: programavsnitt laddas och aktiveras. Söksystemet bör kunna köras på ett mindre datorregleringssystem som "offline tuning" medan reglering pågår.
För att bli en adaptiv regulator skulle systemet behöva vara aktivt ständigt.

Förlustfunktionen ska ange vad som är önskat beteende hos processen. Den beräknas under en bestämd tid för en känd insignal med hjälp av t ex processens in- och utsignaler. Om t ex styringrepp orsakar mycket slitage, och alltså blir dyra i längden, ska förlustfunktionen öka ju större styringreppen är. Genom att operatören får tillgång till förlustfunktionens värde får han eller hon ett relativt mått på hur lyckad regulatorinställningen blivit.

Med termen 'process' kommer att avses den verkliga (industriella) apparat som ska regleras, och 'system' syftar dels på datormodeller och programblock, dels på en process med regulator.

Kapitel 2 orienterar om den vanliga PID-regulatorn som en bakgrund till kommande kapitel. I kapitel 3 ges en översikt över problemet att ställa in regulatorer. De huvudsakliga dragen i det aktuella söksystemet ges i fjärde kapitlet tillsammans med information om hur delsystemen samarbetar.
Vissa delproblem behandlas därefter separat. Kapitel 5 ger synpunkter på identifieringsproblemet, särskilt för återkopplade processer och den använda identifieringsrutinen beskrivs. Det är viktigt att olika förlustfunktioner kan användas i olika situationer. Detta område behandlas relativt utförligt i kapitel 6. Där ges också exempel på omparametrisering av regulatorn. Sjunde kapitlet ger synpunkter på optimeringsrutiner.
I kapitel 8 är det dags att beskriva hur sökmetoden implementerats i dator. Såväl program som parametrar diskuteras utförligt. I nionde kapitlet redovisas simuleringsresultat och några erfarenheter. Avslutande kommentarer och förslag för framtiden ges i kapitel 10.

"Ni kan få vilken färg ni vill - bara den är svart"
H FORD

2. PID-REGULATORN

De idag mest använda regulatorerna i industrin är PI- och PID-regulatorerna. Det är numera vanligt att de realiseras som algoritmer i en dator och in- och utsignaler överföres som elektriska strömmar. På datorns in- och utgångar sitter omvandlare mellan strömmar och tal i datorn.

Insignaler till regulatorn är dels information om processens nuvarande utsignal y, dels information om önskat beteende y^r. Av skillnaden mellan dessa bildas en felsignal e.

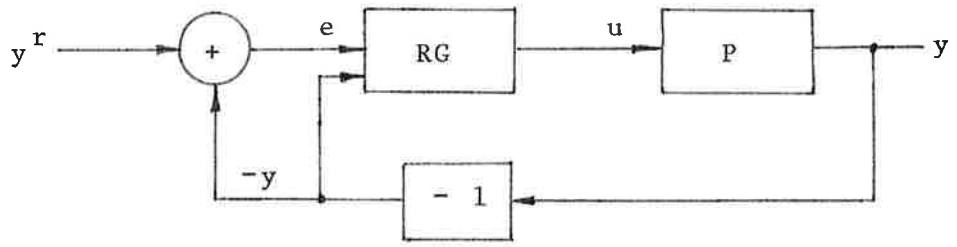


Fig 2.1 Slutet system. RG=processregulator, P=process.

Felet $e = y^r - y$.
En idealiserad PID-regulator ges av

$$u(t) = K\{e(t) + \frac{1}{T_i}\int_0^t e(s)ds - T_d\frac{dy}{dt}\} \qquad (2.1)$$

eller i samplad form (DT = samplingsintervallet):

$$u(t) = K\{e_t + \frac{DT}{T_i}\sum_0^t e_k - T_d\frac{y_t - y_{t-1}}{DT}\}$$

K kallas förstärkningsfaktor, faktorerna T_i och T_d integraltid resp derivatatid. Sätts $T_d = 0$ har man en PI-regulator vilken används då fördröjningstiden är minst ca dubbelt så stor som stigtiden för processen.

(Fördröjningstiden är den tid det tar för processens utsignal att nå halvvägs till slutvärdet då insignalen är ett steg. Stigtiden är tiden för utsignalen att ändras från 10 - 90% av slutvärdet.)

Större K ger snabbare system men stabiliteten minskar. Integraltermen gör att felet så småningom blir noll för konstant insignal. Med integralterm blir systemet långsammare och stabiliteten minskar. Integraltermen påverkar endast systemets lågfrekventa egenskaper (början av stegsvaret påverkas t ex inte). Minskande T_i gör integraltermen större. Integraltermen är viktigt vid reglering av flöden, tryck och koncentrationer.
Då referensvärdet ändras kraftigt blir feltermen e stor under några ögonblick och integraldelen växer kraftigt. Regulatorn kommer därför tvingas styra utsignalen förbi stationära värdet för att ett motsatt fel e ska minska integraltermen igen. Stegsvaret blir mkt oscillativt. Eftersom integraltermen egentligen bara behövs när felet är litet begränsar man ofta integraldelens belopp. Detta kallas ibland reset windup och är speciellt enkelt på datoriserade regulatorer. Villkorlig integration innebär att integraldelen bara kopplas in då felet är litet.

För att skydda processen begränsar man gärna regulatorns utsignal, ev också dess ändringshastighet. Samtidigt som begränsningen inträder bör man "räkna ner" integraltermen så att inte den försiktigare styrningen straffas genom att integraldelen växer kraftigt.

Derivatatermen gör systemet snabbare och i allmänhet stabilare om fördröjningen är liten. Man föredrar att endast derivera processens utsignal y i stället för felet e eftersom stegändringar i referenssignalen annars skulle orsaka mycket stora styrsignaler till processen. Derivering innebär att signaler med hög frekvens förstärks kraftigt. Detta ger ofta problem när högfrekvent brus finns på mätsignalen y. Man tvingas därför att modifiera derivatatermen så att förstärkningen upphör att stiga öven brytfrekvensen $\omega = G_d/T_d$. Regulatorn blir:

$$u(t) = K\left(e_t + \frac{DT}{T_i}\sum_k e_k - K\sum_k\left(\frac{G_dT_d/K}{T_d+G_dDT}\right)^{t-k}(y_{k+1}-y_k)\right) \quad (2.2)$$

Alternativt kan man begränsa derivatatermen då dess amplitud överskrider ett maxvärde.
Derivatatermen är väsentlig vid reglering av temperaturer och ångtryck.

I PID-regulatorer med dödzon sätts utsignalen = 0 då felets belopp är litet och därigenom sparas ställdon.

Processer med ett stegsvar som inte oscillerar nämnvärt regleras vanligen utmärkt med en PI- eller PID-regulator. Innehåller processen ren tidsfördröjning begränsas möjligheterna till höga prestanda. Om regulatorn har tillgång till en modell av processen kan utmärkt reglering uppnås (Otto Smith-regulatorn, Åström (1976)). PID-regulatorn tar inte speciell hänsyn till brus eller störningar (se minimalvariansreglering Åström (1970) resp framkoppling Åström (1976)).

En PI-regulator kan flytta två poler för det slutna systemets överföringsfunktion. Dvs dynamiken för processer med högst två poler kan kontrolleras fullständigt. För högre ordningens processer kan man studera Bodediagrammet för det öppna systemet regulator+process, se Åström (1976) s 69-72.

PID-regulatorn är enkel och billig. Regulatorer med få parametrar är ganska okritiska vid justeringen, se Goodwin (1978). Små variationer i processens dynamik får ingen större inverkan.

3. INSTÄLLNING AV REGULATORER - ÖVERSIKT

Vid nykonstruktion har man störst möjlighet att åstadkomma god anpassning mellan regulator och reglerad process, både vad gäller struktur och inställning. När en befintlig process med kända egenskaper (matematisk modell) ska regleras kan kompenseringsmetoder med fördel användas, se Åström (1976).

Det normala är att man ska reglera en process vars dynamik inte är känd i detalj och där det inte är lönsamt att ta fram en "exakt" modell ur grundformler.
Den möda man lägger ner på att förfina inställningen (och välja regulator) beror på hur höga prestandakrav man har. En mycket stor grupp av reglerproblem är inte svårare än att en standardregulator kan användas. Insatsen består då i att få reda på så mycket (litet) information om processen att regulatorn kan ställas in.

Avsnitt 1 ger några förutsättningar och problem vid detta informationssökande. Avsnitt 2 redovisar några inställningsmetoder som beskrivits i litteraturen.

"I never read books,
just look at pictures"
ANDY WARHOL

3.1 METODIKER OCH PROBLEM

Det finns två grundläggande vägar till regulatorinställning. Efter ansatser om struktur hos process och regulator kan man på många olika sätt beräkna parametrar eller söka sig fram iterativt.

Man behöver kriterier som entydigt talar om dels när man funnit en tillräckligt bra matematisk modell av processen, dels vad som är "bra" regulatorinställning.

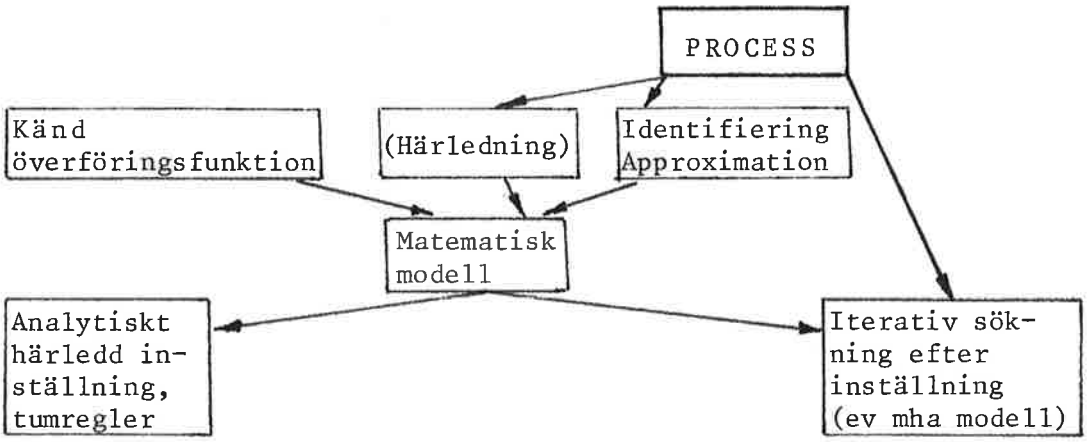


Fig 3.1 Vägar att finna regulatorparametrar

Med hänvisning till fig 3.1 ska några vägar kommenteras.

HÄRLEDNING

Den matematiska modell man får fram ur mätdata vid identifiering av en process speglar bara uppförandet vid den aktuella arbetspunkten för provet medan en modell härledd ur grundformler kan ha stort giltighetsområde och ge god fysikalisk insyn.

Att utgå från fysikaliska (,biologiska,...)samband och härleda en modell utifrån den aktuella processen vållar många problem. Arbetet blir mycket omfattande - ibland omöjligt - utom i enstaka enkla godartade fall.

I litteraturen förekommer många tumregler att trimma in en PID-regulator. Dessa recept har härletts utifrån en ganska enkel modell som antas vanlig, t ex en tidskonstant och en tidsfördröjning. Receptets användbarhet beror helt på om den aktuella processen låter sig approximeras så.

IDENTIFIERING som underlag för matematisk modell

Den testsignal man använder vid insamlig av mätdata måste få alla intressanta egenskaper hos processen att "visa sig". Man föredrar signaler som motsvarar många olika frekvenser t ex stegfunktioner och brus.

Testsignalen måste med nödvändighet störa processen, i vissa fall bringas den att självsvänga. Detta kan bli dyrt (slitage, spill) för processer med långa svängningstider.

Många metoder tar inte hänsyn till att brus, störningar och mätfel förekommer. Noggrannheten försämras och man vet inte hur bra modellen blir. Praktisk erfarenhet får vägleda. Ibland kan man skatta en sökt parameter indirekt genom att mäta en annan storhet som inte är så störd av brus. Vissa metoder kan sortera ut störfaktorerna och ge en separat beskrivning av dessa, till priset av mer omfattande beräkningar.

På identifieringsmetoderna ställer man vidare kravet att beräkningarna ska vara rekursiva så att nya mätdata kan införas utan att beräkningarna måste göras om. Metoden bör kunna kompensera en hög störnivå genom längre mättid.

ITERATIV SÖKNING

Man kan tänka sig varje regulatorparameter som en koordinataxel i ett parameterrum. Iterativ sökning innebär att man börjar med ett startvärde för varje parameter, motsvarande en punkt i parameterrummet.

Man provkör processen med regulator för denna inställning och noterar hur bra uppförandet blir mha en förlustfunktion. Därefter uppsökes en annan punkt i parameterrummet och mäter uppförandet där. En jämförelse mellan dessa resultat ger anvisningar vart man ska förflytta sig härnäst. Genom att beräkna gradienter kan man flytta sig i den riktning i parameterrummet där kriteriet blir bäst. Man får kompromissa mellan att vara försiktig och snabb i sökningen. Med små steg får man beräkna förlustfunktionen många gånger (=många provkörningar av process+regulator) medan man med stora förflyttningar i parameterrummet helt kan missa målet.

Sökning görs ofta mha gradienter, dvs en kombination av parameterändringar som ger störst förbättring av förlustfunktionen. De numeriskt beräknade gradienterna kan störas kraftigt av brus. I sådana fall kan de skattas med korrelationsmetoder.

En nackdel med sökmetoder är att man inte kan vara säker på att de konvergerar mot bra regulatorparametrar. I praktiken kan man starta med några olika utgångsvärden och sortera bort avvikande resultat mha förlustfunktonens värden. Kapitel 7 behandlar optimeringsrutiner.
En annan svårighet är att processen kan försättas i instabila lägen under sökproceduren. Eftersom processen inte är känd vet man inte var instabilitetsgränserna ligger. I de fall man har erfarenhet av vad förlustfunktionen bör bli kan man lägga in tester som avbryter en simulering om förlustfunktionen antar abnorma värden.
Betydande produktionsspill kan uppstå om processen är långsam så att förlustfunktionen behöver utvärderas under lång tid i varje försök.

En del av dessa svårigheter kan övervinnas genom att man först skapar sig en matematisk modell av processen och därefter gör alla simuleringar i modellen. Dessa kan göras mycket snabbt i dator.

3.2 NÅGRA INSTÄLLNINGSMETODER

3.2.1 BERÄKNAD INSTÄLLNING BASERAD PÅ IDENTIFIERAD MODELL

3.2.1.1 Tumregler

Vid inställning av enkla regulatorer används ofta enkla recept. Det betonas att de värden man får fram ska betraktas som utgångspunkt för en manuell vidarejustering. Normalt krävs det att processens förstärkning sjunker för höga frekvenser.

1)
Det mest välkända receptet för PID-regulatorn utarbetades av Ziegler-Nichols 1942, se Harriott (1964, s 178). Det har tagits fram för 2.ordningens processer med tidsfördröjning.
Enligt ett av Z-N recept ställes regulatorn in som en proportionell regulator ($T_i \to \infty, T_d=0$) och förstärkningen K ökas så mycket att process+regulator <u>självsvänger</u> med konstant amplitud. Ur denna förstärkning K_{sv} och svängningstid T_{sv} fås regulatorinställningen.

	K/K_{sv}	T_i/T_{sv}	T_d/T_{sv}
P	0.5	-	-
PI	0.45	0.833	-
PID	0.6	0.5	0.125

<u>Tab 3.1</u> Rekommenderade inställningar enligt Z-N 1.

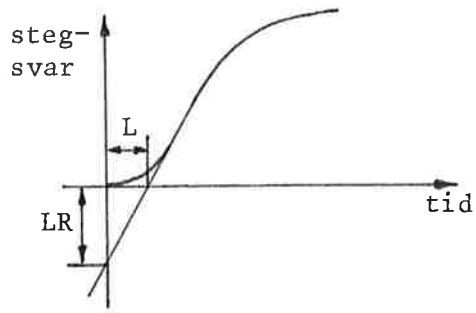
En variant har föreslagits i Harriott , s. 179.

Enligt ett andra recept av Ziegler-Nichols ritar man upp stegsvaret för processen och ur fördröjning L och maximal lutning R fås parametrarna, se fig 3.2.

Tab 3.2 Rekommenderade parametrar enligt Z-N 2.

P:	$K=1/LR$	-	-
PI:	$K=0.9/LR$	$T_i = 3L$	-
PID:	$K=1.2/LR$	$T_i = 2L$	$T_d = 0.5L$

Fig 3.2



En nackdel med denna metod är att störningar kan göra lutningsskattningen osäker.

Ziegler-Nichols metoder ger "måttliga" överslängar och utsignalsvängningarna dämpas någorlunda. För en del 1. och 2.ordningens system kan metoden ge fel resultat.

2)

Även Lopez m fl (1967) beaktar stegsvaret. Man påstår att många industriella processer kan approximeras med 1.ordningens modeller med en tidsfördröjning då det gäller stegsvaret. Ur stegsvaret kan L, R och G_o (processförstärkningen) beräknas enligt fig 3.2. Utgående från denna enkla modell provar man sedan olika recept: Ziegler-Nichols, "4/1 decay ratio" för felet e samt minimering av felsummor: $\int t|e|dt$ (ITAE) resp $\int e^2 dt$ (ISE).

Man finner dels att ITAE-kriteriet ger bättre stegsvar med mindre översläng och snabbare konvergens än t ex Ziegler-Nichols, dels att en PIDregulator ska användas då $L > 0.4(G_o/R)$, annars PI.

Tab 3.3 Rekommenderade parametrar enligt ITAE.

PI	$K=(0.859/G_o)(RL/G_o)^{-0.977}$	$T_i=1.484(G_o/R)(RL/G_o)^{0.68}$
PID	$K=(1.357/G_o)(RL/G_o)^{-0.947}$	$T_i=1.176(G_o/R)(RL/G_o)^{0.738}$ $T_d=0.381(G_o/R)(RL/G_o)^{0.995}$

Man bör observera att alla slutsatserna gäller modellen och att det verkliga resultatet beror av överensstämmelsen mellan modell och verklig process.

3)
Symmetrisches Optimum ger ett svängningsfritt stegsvar och används av Frank.

3.2.1.2 Självinställare

Självinställande regulatorer kan göras med polplacering och rekursiv identifiering, se Åström m fl (1978), (1976 s. 246-270), men man kan också använda algoritmerna till ett system som på begäran går in och anger nya parametrar.
Man identifierar en högre ordningens modell och bestämmer en regulator som placerar systemets poler i förutbestämda lägen. Regulatorn innehåller vanligtvis rätt många parametrar. Det är emellertid också möjligt att använda en PID-struktur och göra polplacering om ens styrda process kan approximeras med ett andra ordningens system. Algoritmer för självinställande PID-regulatorer baserade på polplacering beskrivs av Wittenmark (1979).

3.2.1.3

Davies (1966) ger exempel på inställning av en PI-regulator på en 3.ordningens process. Inget hindrar att PID används. Algoritmen minimerar felets kvadratiska medelvärde med en gradientmetod (steepest-descent). För att beräkna regulatorparametrarna behöver han konstruera en härledd funktion (signal) ur felsignalen mha en modell av processen. Davies metod får i det allmänna fallet kompletteras med en identifiering av processen. Hela konstruktionen kan realiseras digitalt. Systemet påstås relativt okänsligt för brusstörningar och konvergerar i provkörda fall på högst 10 tidskonstanter för processen då en analogimaskinkoppling används.

Vill man optimera ett annat kriterium får metoden härledas på nytt.

3.2.2 INSTÄLLNING BASERAD PÅ SÖKNING

1)
Pendlebury et al visar på en möjlighet att skriva ett mycket kort program för optimering under drift. Regulatorn är en digital PID och målet är att minimera en viktad summa av felet, $\Sigma t|e_t|$. Sökmetoden är enkel. Reglersystemet utsätts för en liten stegstörning uppåt i referensvärdet, en paus, återgång med referensvärdet och en paus. Under denna tid beräknas förlustfunktionen. En regulatorparameter i sänder varieras ±h och varje gång beräknas förlustfunktionen. Blir den bättre behåller man den nya parametern, annars inte. Sökriktningarna blir hela tiden parallella med axlarna i parameterrummet.

En dator kan lätt exekvera flera PID-regulatorer med sådan inställningsutrustning parallellt (varje loop behöver ca 20 parametrar). I sammansatta system riskerar man naturligtvis att ändringar i ett delsystem påverkar förlustfunktionen i ett annat.
Förlustfunktionen kan lätt skiftas.

Genom att stegstörningarna görs både uppåt och neråt hoppas författaren att inställningen blir lagom med tanke på olinjariteter hos processen.
Likaså påstås anpassningen till rådande processbrus blir bättre än i modellsimuleringar.
Den enkla proceduren tar inte till vara så mycket information från tidigare sökningar som t ex gradientmetoder gör. Därför behövs fler försök för att nå acceptabelt resultat. Det ställs också större krav på hyfsade startvärden för regulatorparametrarna.
En allvarlig nackdel med metoden är att man tvingas störa processen många gånger då ganska lite information dras ut ur varje störning. Ojämn produktkvalitet under långa provtider kan orsaka större förlust än en snabb regulatorinställning med få prov och mindre optimalt beteende.

2)

Enligt Mills (1965) kan söktiden nerbringas om ändringarna i regulatorparametrar samordnas då man försöker minimera en förlustfunktion. När flera parametrar sökes ska förlustfunktionen utvärderas för varje ändring utan att tidigare ändringar får störa mätningen. Processens dynamik och störningar avgör hur lång tid detta minst tar.

Mills korrelationsmetod mäter samvariationen mellan en parameterändring och förlustfunktionen. Då flera regulatorparametrar varieras måste parameterändringarna vara okorrelerade. Detta begränsar mätsnabbheten ytterligare.

Genom att låta ändringsfrekvenserna förhålla sig som 1 : 2 : 4 blir inställningstiden ca 1/3 av tiden då ändringarna ej synkroniserats (vid en viss noggrannhet i förlustfunktionen).

Mills föreslår att varje parameter ändras i förhållande till förbättringen av förlustfunktionen, dvs en steepest-descent metod.

Mills synpunkter synes mest relevanta då inga störeller provsignaler får matas in på processen.

"Gee, Adam, this was fun.
How did you ever come
to think of it?"

EVE

4. SÖKSYSTEMETS UPPBYGGNAD

Detta kapitel beskriver huvuddragen i den föreslagna metoden och diskuterar dess delar översiktligt. Detaljerna ges i kapitel 8.

4.1 BLOCKSCHEMA

Söksystemet uppgörs av rutiner i en dator, exempelvis den processdator som sköter regleringen.
Den reglerade processen P styrs hela tiden av sin regulator RG under sökproceduren. Vi kallar denna slinga för PROCESSLOOPEN, fig 4.1.

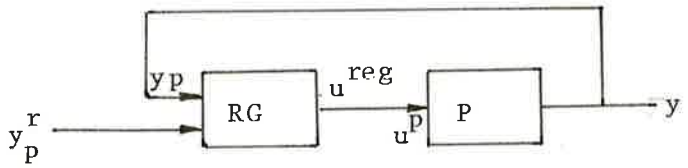


Fig 4.1 Processloopen

Metoden bygger på minimering av en förlustfunktion som beror på beteendet hos en modell av processen. Sökningen efter regulatorparametrar och utvärderingen av förlustfunktionen LOSS sker med hög hastighet i en datormodell. Denna del kallas MINLOOPEN, fig 4.2. Den innehåller en modell MOD av processen P och en likadan regulator som processloopen. Vidare innehåller den en sökrutin OPTA och en rutin FKN som beräknar förlustfunktionen.

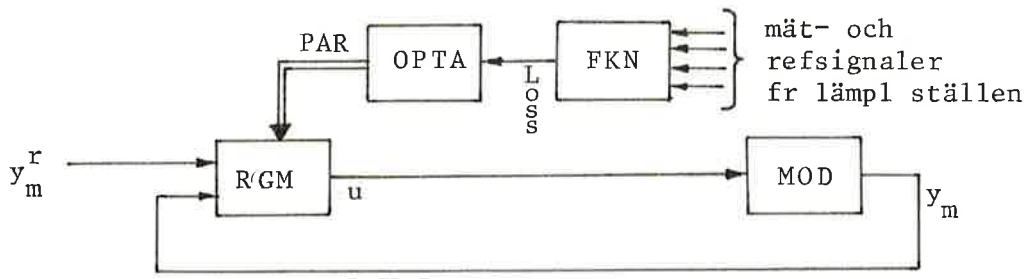


Fig 4.2 Minloopen

Sökrutinen OPTA beräknar succesivt nya förslag till regulatorparametrar PAR. Dessa matas i första hand till regulatorn RGM i minloopen. Därefter görs en datorsimulering i minloopen med en testsignal och förlustfunktionen LOSS beräknas. Med ledning av LOSS beräknar minimeringsrutinen återigen ett nytt förslag PAR.

Den verkliga processen antas inte känd. Därför innehåller söksystemet en identifieringsrutin LS som har processens in- och utsignaler som indata och en skattning TH av processens parametrar som utdata. Vid flera tillfällen under sökningens gång matas regulatorparametrar PAR över från minloopen till processloopen för att denna ska ha så bra parametrar som möjligt. Ett block LATCH tillhandahåller på sin utgång de parametrar PAR som processregulatorn för tillfället ska använda.

Söksystemet kommer alltså att få följande blockschema:

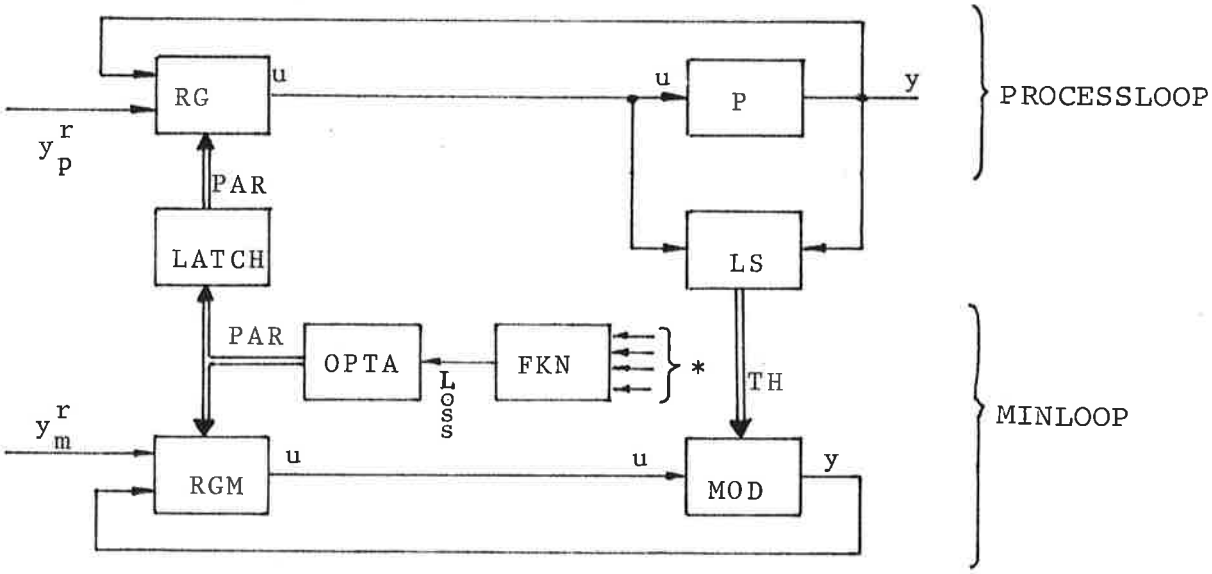


Fig 4.3 Söksystemets blockschema (* mätpunkter för förlustfkn)

Naturligtvis kan rutinen FKN tillfälligtvis kopplas över till processloopen i stället för att mäta förlustfunktionens värde där.

4.2 SYNKRONISERING AV PROCESS- OCH MINLOOP

Hela processloopen arbetar med ett samplingsintervall DT som passar den reglerade processen P. Det innebär att utsignalerna hålls konstanta tills ett intervall förflutit. Därefter hämtas nya mätvärden in och nya styrsignaler beräknas och läggs ut.

Då man ska bedöma ett system tekniskt ser man ofta hur utsignalen blir för en stegändring i referenssignalen. Denna signal har valts här. Ingenting hindrar att andra provsignaler används.
Minloopen ska ha den styrsignal som förlustfunktionen bestämmer. Processloopen ska, åtminstone i början, ha en signal som medger god identifiering.

Vid varje samplingstidpunkt i processloopen beräknar identifieringsrutinen LS en ny skattning av processen P. Ett exempel på hur regulator RG och process P arbetar ges i fig 4.4, där även skattningarna från LS lagts in.

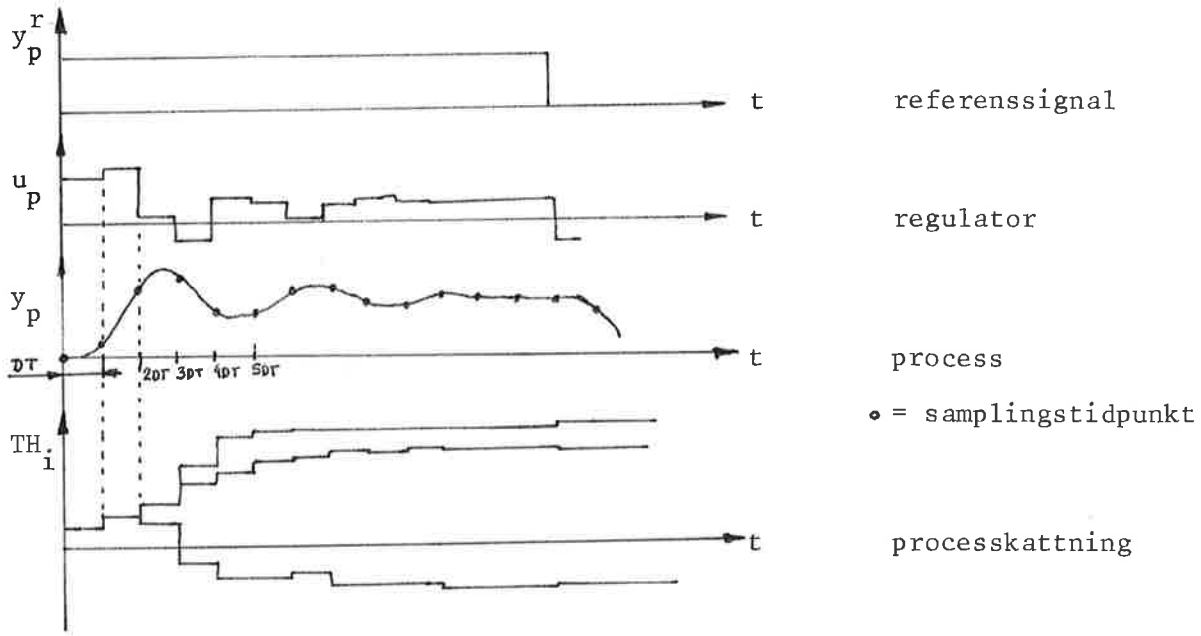


Fig 4.4 Några signaler vid ett processtegsvar. Samplingsintervallet DT har markerats.

Minloopen arbetar med större hastighet.
Under UPER (=litet heltal) samplingsintervall i processloopen görs en simulering av ett helt provstegsvar i minloopen, se fig 4.5 nedan.
För varje stegsvar i minloopen utvärderas förlustfunktionen och därefter beräknas nya regulatorparametrar för regulatorn RGM (i OPTA). Samplingsintervallet i minloopen betecknas DTM.

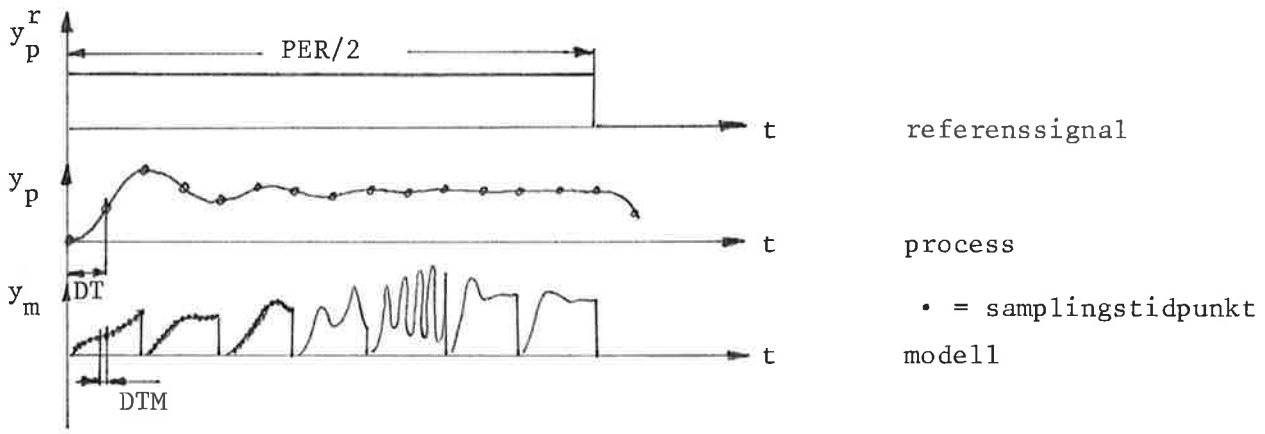


Fig 4.5 Processens stegsvar och succesiva provstegsvar i minloopen (här UPER = 2).

Alla system i minloopen nollställs före varje provstegsvar. Se vidare avsnitt 8.2.1.

Följande parametrar väljes då tidsskalan för de olika delsystemen fastlägges:

Parameter	Namn	Tolkning	Fastlägges m h t
DT	samplings-intervall för proc-loopen	←	processkrav
PROVL	Provlängd	Antal termer i förlustfunk.	hur snabbt proc svänger in sig, se även 8.2.1.

- forts -

PER	period för procloopens stegsvar	period i antal samplingsintervall	identifiering (se 5.1, 8.2.3) 2·PROVL
UPER	uppdaterings-period	antal sampl.interv. i procloopen innan ny uppsättning reg.parametrar matas över till proc-regulatorn RG	förväntad konvergens-snabbhet för identifiering+sökning.

Om man sätter samplingsintervallet DTM i minloopen

$$DTM = \frac{UPER}{PROVL} DT \qquad (4.1)$$

kommer nya regulatorparametrar att matas över till processloopen efter UPER samplingsintervall i processloopen. Genom att öka UPER körs minloopen "långsammare". Skulle det visa sig att den aktuella maskinen inte hinner med alla beräkningar med litet DTM bör man alltså öka UPER. (se även 8.2.3 och bilaga 3)

Sammanfattningsvis kan sägas att denna metod tar ut mycket information ur ett begränsat antal (störande) processtegsvar. Sökprogrammet bör kunna köras med lägre prioritet parallellt med andra regulatorprogram. Pga datorns snabbhet förkortas söktiden avsevärt jämfört med prov direkt på processen.

"Cheer up, the worst is yet to come"
PHILANDER JOHNSON

5. IDENTIFIERING AV PROCESSEN

5.1 IDENTIFIERINGSPROBLEMET

Som påpekas i Söderström (1974) är de flesta teoretiska resultat om skattningars egenskaper endast giltiga för öppna system, dvs system utan återkoppling (reglering).
Söderström samt Elvgren, Krantz (1974) ger några olika fall då det är möjligt att identifiera ett linjärt tidsinvariant samplat system med linjär regulator:

1) referenssignalen (insignalen y^r) ska vara "tillräckligt varierande" (persistingly exciting, se def i Söderströms bok)
2) regulatorn är tidsvariabel (t ex under sökning efter parametrar eller då man avsiktligt skiftar mellan två olika uppsättningar regulatorparametrar)
3) konstant regulator av tillräckligt hög ordning
4) oberoende normalfördelat brus adderas i regulatorn
5) tillräckligt hög ordning n har ansatts i identifieringsrutinen.

Alltså gynnas vi av villkor 2 men missgynnas av 3 då vi söker parametrar för t ex en enkel PI-regulator. Kort sagt hoppas man att identifieringen blir klar före sökningen efter regulatorparametrar.

Eftersom processloop och minloop har separata insignaler kan vitt brus användas som referenssignal för processloopen i början av sökningen om man finner att processen är svår att identifiera (1 och 4).

Villkoret 1 på insignalen ger t ex att periodtiden inte får vara kort om processen exciteras med steg.

Att en konstant regulator måste vara av hög ordning då många processparametrar ska identifieras är lätt att se.

Antag att systemet $y_t = -a_1y_{t-1}-a_2y_{t-2} + b_1u_{t-1}+ b_2u_{t-2}$

med en proportionell regulator $u_t = K(y_t^r - y_t)$

ska identifieras (jämför fig 4.1).

Då blir $y_t = -a_1y_{t-1}-a_2y_{t-2} +b_1K(y_{t-1}^r - y_{t-1})+b_2K(y_{t-2}^r - y_{t-2}) =$

$$= c_1y_{t-1} + c_2y_{t-2} + c_3y_{t-1}^r + c_4y_{t-2}^r$$

För en konstant insignal men exciterad process "visar sig" bara c_1, c_2 och c_3+c_4 och därur kan inte alla a_i, b_i bestämmas (3 ekvationer, 4 obekanta). Genom att variera tillräckligt på insignalen visar sig alla processparametrarna, 1.

För en komplexare regulator fås processparametrarna med insignalen y^r:

Regulator: $u_t = k_1y_t^r - k_2y_t - k_3y_{t-1}$

Då blir $y_t = (-a_1-b_1k_2)y_{t-1} -(a_2+b_2(k_2+k_3))y_{t-2} - b_2k_3y_{t-3} +$

$$+ b_1k_1y_{t-1}^r + b_2k_2y_{t-2}^r =$$

$$= d_1y_{t-1} + d_2y_{t-2} + d_3y_{t-3} + d_4y_{t-1}^r + d_5y_{t-2}^r$$

Parametrarna d_1,d_2,d_3,d_4+d_5 kan identifieras för en icke-konstant process med konstant insignal och därur bestämmes a_i,b_i (4 ekv, 4 obekanta).

Det är alltså <u>återkopplingen</u> som bestämmer hur många parametrar som kan identifieras.

Då processen är okänd och regulatorn enkel och konstant kan det alltså vara lönt att börja med att identifiera processen själv utan någon återkoppling.

5.2 IDENTIFIERINGSRUTINEN

Då de störningar som förekommer på processens in- och utgångar antas vara vitt brus används en minsta-kvadrat metod. Denna är rekursiv vilket innebär att nya mätdata kan tillfogas utan att alla räkningar behöver göras om då nästa skattning ska avges.

Man försöker skatta parametrarna $a_1,\ldots,b_n$ i följande linjära modell:

$$y_t + a_1 y_{t-1} + \ldots + a_n y_{t-n} = \\ = b_1 u_{t-1-L} + \ldots + b_n u_{t-n-L} + v_t \qquad (5.1)$$

där y är processens utsignal, u insignalen och v en störning (vitt brus). Modellens ordning är n, antalet tidsfördröjningar L. Samplingsintervallet har valts som tidsenhet.

Inför följande beteckningar:

$$\theta_o = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_n \\ b_1 \\ \vdots \\ b_n \end{pmatrix} \quad \text{(sökt parametervektor)} \qquad \Phi_t = \begin{pmatrix} -y_{t-1} \\ -y_{t-2} \\ \vdots \\ -y_{t-n} \\ u_{t-1-L} \\ \vdots \\ u_{t-n-L} \end{pmatrix} \quad \text{(tidigare processvärden)}$$

Då blir (5.1):

$$y_t = \theta_o^T \Phi_t + v_t \qquad (5.2)$$

Minsta-kvadrat skattningen $\hat{\theta}(N)$ av parametrarna, baserad på N mätningar ges i princip av det värde θ som minimerar

$$V(\theta) = \frac{1}{N} \sum^{N} [y_t - \theta^T \Phi_t]^2 \qquad (5.3)$$

dvs en summa av kvadrater på skillnaden mellan verklig signal och signal beräknad enligt (5.2).

För att metoden ska fungera krävs att man känner ordningstalet n, att insignalsekvensen u innehåller tillräckligt många frekvenser och att v är vitt brus. För återkopplade system hänvisas till avsnitt 5.1.

Beräkningarna görs enligt följande formler:

$$\begin{cases} K(N+1) = \dfrac{P(N)\,\Phi(N+1)}{1+\Phi^T(N+1)P(N)\,\Phi(N+1)} & (5.4) \\ P(N+1) = [P(N) - K(N+1)\,\Phi^T(N+1)P(N)]/WT(N) & (5.5) \\ \hat{\theta}(N+1) = \hat{\theta}(N) + K(N+1)\left[y(N+1) - \hat{\theta}^T(N)\,\Phi(N+1)\right] & (5.6) \\ WT(N+1) = (1 - WTM)WT(N) + WTM \;; \quad WT(0)=WTI & (5.7) \end{cases}$$

Argumentet N+1 hänför sig till nya mätvärden coh skattningar.

Den <u>nya skattningen</u> fås alltså genom att lägga till en korrektion till föregående skattning. Korrektionen är proportionell mot skillnaden mellan det verkliga mätvärdet y(N+1) och det värde vi kan beräkna mha (5.2) utan brus. Vektorn K talar om hur korrektionen ska viktas.

P är en symmetrisk matris 2n×2n, påminnande om en kovariansmatris för in- och utsignaler. Dess begynnelsevärde är 0 utom i diagonalen (100). P kommer att bli ett mått på osäkerheten i skattningen.

WT är en viktfaktor ("<u>glömskefaktorn</u>") $\leq$ 1 och den medför att gamla processvärden viktas mindre än färska. WT < 1 används då processen är tidsvariabel eller svåridentifierad. Den innebär att man i stället minimerar

$$V_i(\theta) = \frac{1}{N}\sum^{N}[y_t - \theta^T\Phi_t](WT)^{N-t} \qquad (5.8)$$

(I början N-t stort och WT<1 ⇒ avvikelse spelar liten roll. Se även N, avsnitt 8.2.3).

WT<1 gör att P-matrisen växer då in- och utsignaler är konstanta. Därför måste man excitera processen då WT<1.

Vanliga värden på WTI är 0.9 och WTM 0.01 - 0.001. Fördelarna med en glömskefaktor som går mot 1 är att den behövs mest i början av skattningen då värdena är dåliga,och att WT<1 medför att skattningsparametrarna varierar något hela tiden (skattningsfelet avtar inte som 1/N vilket är fallet annars).

Om färgat brus förekommer ger minsta-kvadrat skattningen felaktiga resultat. Vid misstanke om färgat brus använder man antingen en rekursiv maximum likelihood metod eller utökar ordningstalet n väsentligt i minsta-kvadrat metoden.
Söderström (1974) påstår att konvergensen är snabb, dock mot fel värden om bruset är färgat eller processen är svåridentifierbar.

Det är bra om datorn arbetar med många signifikanta siffror, gärna minst 7, eftersom P-matrisen bildas genom skillnader mellan ungefär jämnstora tal då metoden börjat konvergera.
Beträffande val av n och L hänvisas till avsnitt 8.2.3.

6. FÖRLUSTFUNKTIONER

"Förlustfunktioner antar speciellt stora värden vid krogbordet"

En av de viktigaste delarna i detta system som söker regulatorparametrar är förlustfunktionen. Den införs för att avgränsa problemet och göra det entydigt lösbart.

Syftet med regleringen anges i förlustfunktionen som baseras på mätbara data. Funktionen anger precis vad man ska ta hänsyn till när man optimerar och den ger allt vad söksystemet ska uppfatta som bra och dålig reglering.

Genom att konstruera sin förlustfunktion sinnrikt kan man ta hänsyn till flera faktorer och vikta dem olika. Sådana kan vara ojämnheter i produktionskvaliteten, energiåtgång, "mjukhet" i styrningen så att inte reglerdon slits i onödan etc.

Förlustfunktionens värde för en viss process och regulatorinställning används av minimeringsrutinen.
Det är egalt om man minimerar en förlustfunktion eller maximerar en "vinstfunktion".

Avsnitt 1 ger några allmänna principer och synpunkter. I andra avsnittet behandlas några vanliga förlustfunktioner och avsnitt 3 ger förslag på förlustfunktioner av annan typ. Kapitlet avslutas med synpunkter på regulatorparametrisering.

6.1 PRINCIPER

Förlustfunktionens värde beräknas genom att processen (eller en modell av den) styrs av en regulator med viss struktur och parameteruppsättning och utsätts för en representativ provsignal.
Det kan t ex vara en 3.ordningens modell som styrs av en PI-regulator medan testsignalen är en stegändring i referensvärdet. Förlustfunktionen uppmäts kanske under 50 samplingsvall.

Förlustfunktionen ska ha indata i form av processvärden som kan påverkas genom styrning. Primärdata kan räknas om så att "verklig" vinst eller förlust fås.

Naturligtvis måste det bli avvägningar mellan önskvärda processdata och kostnader att uppnå dessa.
I praktiken kan endast suboptimeringar göras, bl a beroende på att slutlig förlust eller fördel är för svår att förutse och att man av praktiska skäl sällan kan optimera mer än en eller två reglerkretsar åt gången.

Stegreferens - Modellreferens - Signalreferens

Genom begreppet förlustfunktion har man ett mycket allmänt redskap i sin hand. Metoden kan användas på alla system. Man måste välja de storheter som ska vara indata. Det är vanligt att man bildar skillnaden mellan en signal som man anser "bra" (börvärde) och processens värde (ärvärde). Det enklaste är att välja processens referensvärde som den ena insignalen (stegvisa ändringar, stegreferens). Ett annat bra val är utsignalen från ett känt system med önskvärda egenskaper då detta utsätts för samma referenssignal som själva processen med dess regulator (modellreferens). Se avsnitt 9.3 och 10.1.

Antag att en process kommer att utsättas för en periodisk styrsignal under drift. En tredje variant är då att låta den andra insignalen till förlustfunktionen vara denna styrsignal. Då kan förlusten för just denna signal minimeras, en extremt "kundanpassad individuell regulatorinställning" (signalföljning, se 10.2.3).

Det valda kriteriet på god styrning ska vara så selektivt att optimeringsrutinen kan hitta ett minimum. Detta är svårt att veta på förhand och dessutom inverkan faktorer som regulatorns parametrisering, skalning av regulatorparametrarna och processens typ.

Man kan ha hjälp av att provköra regulator+process för olika regulatorinställningar och med ledning av förlustfunktionens värde skissa nivåkurvorna (punkter med samma funktionsvärde, jämför väderkartans isobarer). Därur kan man se om en annan skalning eller parametrisering är nödvändig.

Om den verkliga processen innehåller en ren tidsfördröjning är det lämpligt att denna mäts upp och en kompensering införs så att denna systemegenskap inte straffas.

Anm. 1

Om man av någon anledning inte finner globalt minimum av förlustfunktionen finns det flera uppsättningar regulatorparametrar som ger samma värde.
Nedan visas stegsvar för 3 olika regulatorinställningar för en PI-regulator på en 3.ordningens process RTH (se 9.1). Nivåkurvor för detta system med förlustfunktionen ISE ($\Sigma e^2 dt$, se 6.2.1) visas i fig 7.2. Minimum är ca 2.3. De tre realiseringarna ger alla värdet 3 för förlustfunktionen och de varierar som synes.

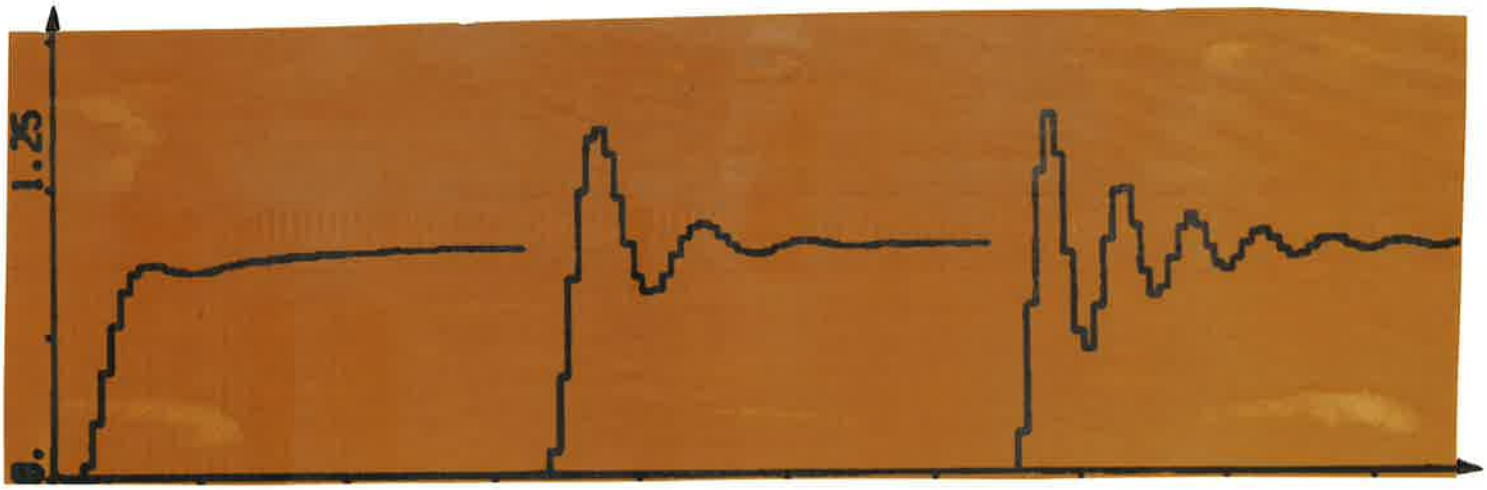


Fig 6.1 Tre olika realiseringar som ger LOSS=3 med förlustfunktionen ISE på PI+RTH.
Regulatorinställningar K/T_i:
1.75/9.7, 3.06/4, 5.58/12.

Anm. 2

Om regulatorn innehåller begränsning av utsignalens u^r belopp, kommer detta att fungera som ett extra tillägg till förlustfunktionen (nämligen obegränsat stort tillskott då $|u^r|$>ULIM, 0 annars).

6.2 NÅGRA VANLIGA FÖRLUSTFUNKTIONER

I samtliga fall nedan bildas felsignalen som skillnaden mellan ett önskat värde och processens värde: $e_t = y_t^d - y_t^p$. Insignalen kan vara stegändringar eller brus. I Ogata (1970) finner man ytterligare detaljer.

6.2.1 ISE (Integral squared error)

Förlustfunktionen $LOSS_1 = \sum^N e_t^2$ där N är ett så stort tal att systemet hinner stabilisera sig (e_t försumbar). Tiden räknas i antal samplingsintervall för regulatorn.

Stora fel viktas kraftigt, små fel obetydligt. Resultatet blir ofta en snabbt system med lång insvängningstid. Överslängen blir måttlig. Styrsignalerna blir små utom i början.
Detta kriterium är ett av de vanligaste, beroende på dess enkelhet och möjligheten att beräkna ISE på olika sätt. Det kan med modifikationer ges tolkning som energiförbrukning eller varians. Optimum är inte så skarpt.
Kriteriet tycker att alla tidpunkter är lika viktiga då det gäller att följa den önskade signalen. Denna egenskap kan vara trevlig då man använder modellreferens (se 6.1) då modellens beteende i stort ska efterliknas (se 6.1 och 9.3).

Nedan ges ett exempel på stegsvar för ett 2.ordningens system med en PI-regulator inställd efter detta kriterium. Med ISE-kriteriet tvingas processen stiga snabbt i början då felet är stort. Några svängningar runt stationära värdet straffas ej hårt.
För jämförelsens skull ges även stegsvaret för en inställning efter tumregel (Ziegler-Nichols 1, 3.2.1.1.1).

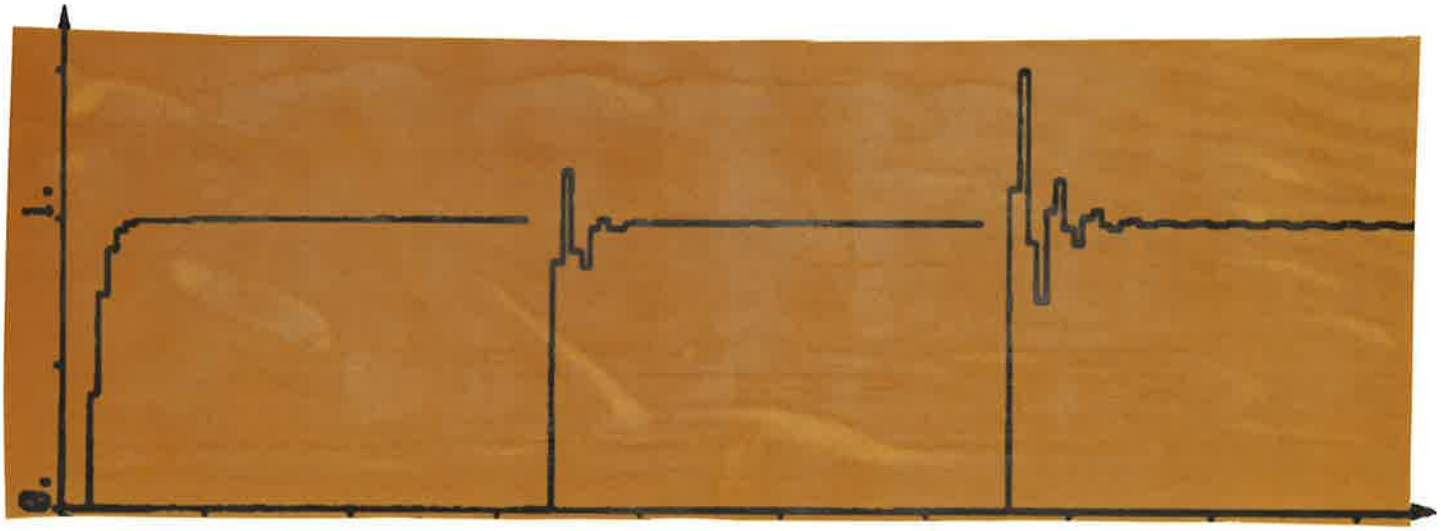


Fig 6.2 a. Stegsvar för 2.ordningens process SEC
b. Stegsvar med PI-regulator opt. enligt ISE
c. Stegsvar med PI-regulator enligt Z-N 1.

6.2.2 ITAE (integral of time mulitiplied error)

$$LOSS_2 = \sum^{N} t|e_t|$$

Detta kriterium är intressant eftersom det viktar tidpunkter olika. Ett stort fel i början betyder inte så mycket som svängningar längre fram. Processen tvingas uppnå stationaritet snabbt. Överslängen i början blir enligt litteraturen normalt något större än för ISE men stegsvaret blir väl dämpat. Dessutom är kriteriet selektivt och väl lämpat för inställning av PI- och PID-regulatorer. "Krypande stegsvar" elimineras.

Varianter är $LOSS_3 = \sum^{N} f_t|e_t|$ där man låter viktfunktionen f bli stor för viktiga tidpunkter, resp $LOSS_4 = \sum^{N} t\left(|e_t| + \left|\frac{de}{dt}\right|\right)$ som har ännu större selektivitet än ITAE.

Stegsvar med ITAE visas i fig 6.3. Jämfört med ISE får stigningen i början ta längre tid men överslängarna är mer dämpade, eftersom de straffas mer ju längre tiden går. Här blev överslängen i början tom mindre än för ISE.

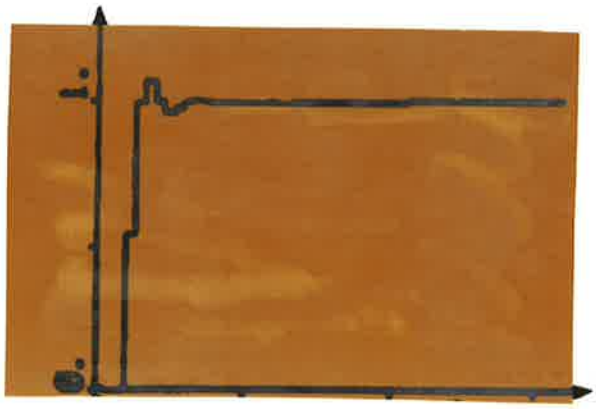


Fig 6.3 Stegsvar för 2.ordningens process SEC med PI-regulator optimerad enligt ITAE.

6.2.3 Hänsyn till stora styrsignaler

$$LOSS_5 = LOSS_i + \lambda \sum^{N} (\Delta u_t)^2$$

där $LOSS_i$ är t ex något av de tidigare redovisade felsignalkriterierna och Δu är ändringen i styrsignal från ett samplingsintervall till ett annat.

Detta kriterium adderar straff på stora styrsignaler och får styrningen att bli mjuk. Nackdelar med kriteriet är att kravet på små styrsignaler ofta är motstridigt det ursprungliga kriterium man ställt upp och att det är svårt att förutse hur lugnt/långsamt stegsvaret blir för ett visst λ.

Kriteriet ISE fungerar bra med detta tillägg, se fig 6.4, medan ITAE redan är ett så specialiserat kriterium att $LOSS_5$ inte bör användas.
I fig 6.4 ser man att förloppet blir lugnt och det är Σe^2-termen som gör att rätt slutvärde uppnås.

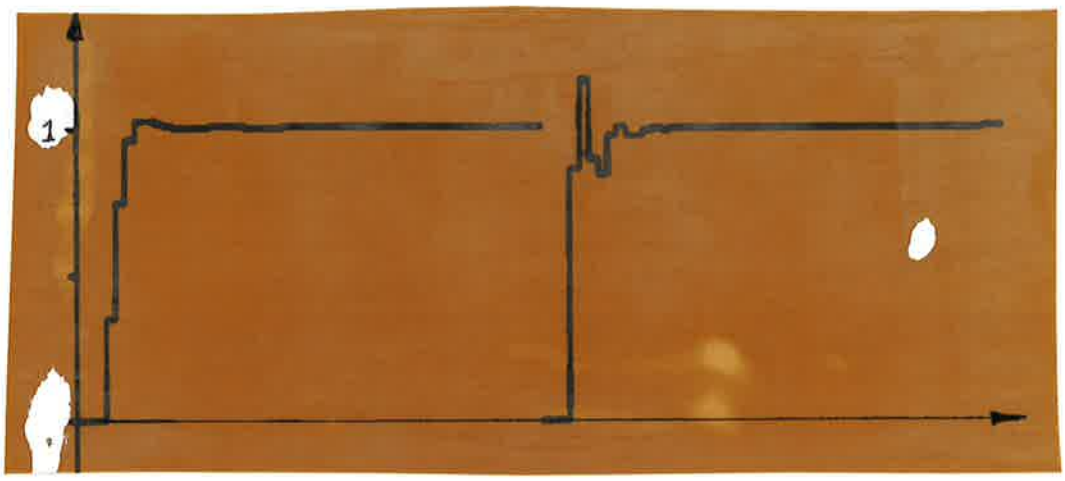


Fig 6.4 Stegsvar för 2.ordningens process SEC med PI-regulator
a. optimerad enligt ISE + $\lambda\sum\Delta u^2$, $\lambda=0.5$
b. optimerad enligt ISE

6.3 ANDRA TYPER AV FÖRLUSTFUNKTIONER

6.3.1 Otillåtna parametervärden

En metod att lägga begränsningar på de värden som regulatorparametrarna antar är att införa stora tillskott till förlustfunktionen då otillåtna värden provas av optimeringsrutinen.
Ex. Antag att optimeringsrutinen under sökningen börjar prova negativ förstärkning K och att man aldrig vill ha det. Genom ett tillägg till LOSS kommer optimeringsrutinen snabbare ut i tillåtet område.

6.3.2 Ungefärlig dynamikspecifikation

Om målet bara är att uppnå en viss ungefärlig dynamik kan man specificera ett område i y-t-planet där förlustfunktionen sätts lika med noll. Utanför området stiger förlustfunktionen monotont.

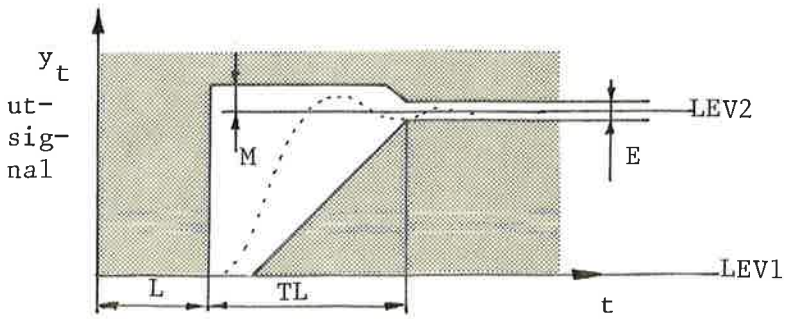


Fig 6.5
Exempel på tillåtet område, "BOX".

6.4 PARAMETRISERING AV REGULATORN

En och samma regulator kan framställas på olika sätt. För dessa antar förlustfunktionen naturligtvis samma värden men (nivåyte-)landskapet ser annorlunda ut. Detta kan ha betydelse för minimeringsrutinen.
Här ges två exempel beträffande PI-regulatorn för att belysa fenomenet.

Regulator 1: $$u_t = K[e_t + \frac{DT}{T_i} \sum_k e_k]$$

Regulator 2: $$u_t = K_1 e_t + K_2 \, DT \sum_k e_k$$

Det är troligen så att regulator 2 är lämpligare vid minimering i en rutin utan skalfaktorer. Jämförbara nivåkurvor för förlustfunktionen ISE och processen RTH (se 9.1) visas nedan.

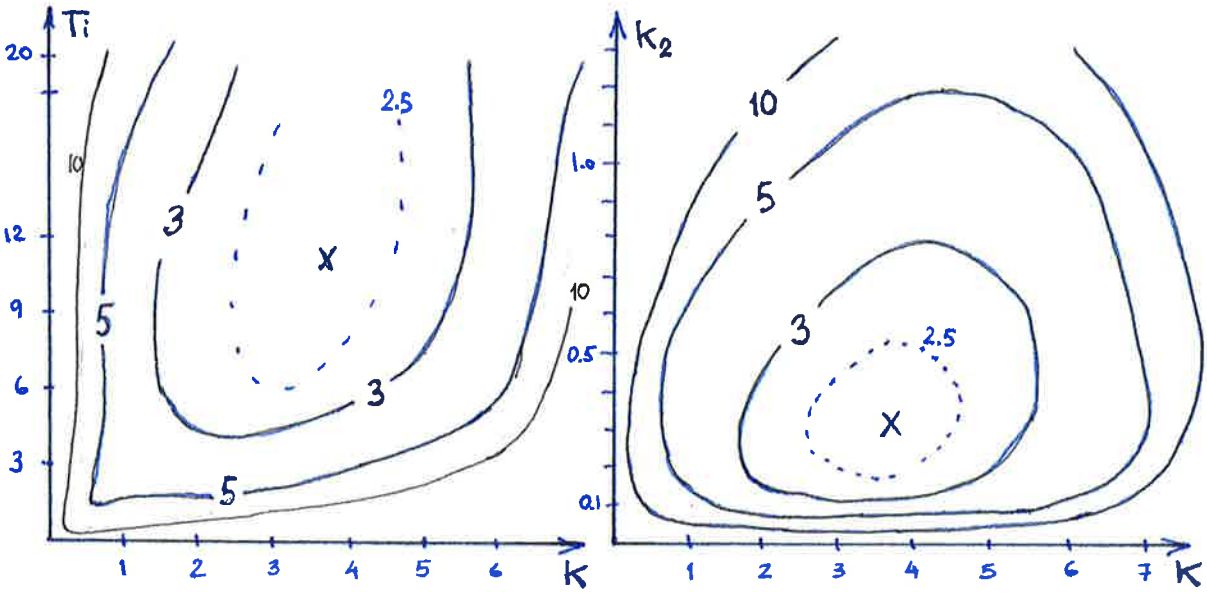


Fig 6.6 Nivåkurvor för 3.ordn processen RTH och förlustfunktionen ISE för två olika PI-regulatorer.

Nivåkurvor och därmed lämpligaste parametrisering beror starkt på aktuell process.

Beskriv er själv med mindre än 25 ord.

-Koncis.

7. OPTIMERINGSRUTINER

Optimeringsrutinen i söksystemet har till uppgift att söka regulatorparametrar så att förlustfunktionen blir minimal. För varje regulatorinställning som föreslås måste förlustfunktionen uppmätas, t ex genom att ett stegsvar körs. Med ledning därav föreslås nya parametrar. Effektiviteten hos optimeringsrutinen avgör hela söksystemets effektivitet.
Avsnitt 1 diskuterar optimeringsrutiner allmänt, medan avsnitt 2 ger några exempel på rutiner.

7.1 ALLMÄNT

Uppgiften består alltså i att minimera en skalär funktion av några regulatorparametrar θ_1, θ_2,..., processen P och regulatortypen R: min LOSS(θ,P,R) då endast information av funktionsvärden finns tillgängliga.

Numeriska minimeringsrutiner lider alltid av bristen på <u>överblick</u> - var ungefär finns (den djupaste) dalen i funktionsvärdet?
Olika rutiner använder olika mycket av den hittills skaffade informationen för att gissa nästa steg. Detta är både på gott och ont. En enkel snabb rutin med "kort minne" glömmer lyckosamma försök och felsteg lika fort. En avancerad rutin kan bygga upp goda skattningar av både första- och andraderivator och dessutom rikta in sig efter "topologin".

En annan fara är alltför automatiserade "idiotsäkra" rutiner. Ju fler avgöranden som överlåts på rutinen, ju större är risken att hamna i sökningar som ständigt upprepas. Samtidigt finns mindre information presenterad för användaren för att åtgärda misslyckandet.

Många minimeringsrutiner gör andragradsapproximationer

av funktionen och fungerar bättre ju mer parabellik funktionen LOSS är. Ibland är det nödvändigt att skala om parametrarna så att nivåkurvorna (-ytorna) blir mer cirkelformade.

Metoderna är iterativa, dvs vi har en parameterpunkt x_k och finner nästa x_{k+1} genom $x_{k+1} = x_k + s_k r_k$. r_k är sökriktningen och s_k är ett tal som anger steglängden.
Vi har alltså två problem: 1) välj riktning, 2) välj steglängd.

I de flesta fall tjänar man på att utnyttja beprövade programpaket skrivna av experter på området. Några rutiner medger minimering under bivillkor av samma typ som i 6.3.1.

För alla metoder får man räkna med en del provkörningar med olika startvärden och parametrar för optimeringsrutinen innan den fungerar bra för det aktuella problemet.

7.2 NÅGRA ANVÄNDA RUTINER

7.2.1 ABSOLUTE BIAS

Denna metod tas med som exempel på metoder med "kort minne". Sökritningen fås här genom att generera en slumpvektor $r_k \in N(0,1)$. Nya punkten $x_{k+1} = x_k + r_k$. Sökningen i en lyckosam riktning fortsättes tills en sämre punkt påträffas. När störningar kan förväntas accepteras endast förbättningar i funktionsvärde om skillnaden är större än $\varepsilon > 0$.
För funktioner av få variabler är metoden effektiv. Programmet är mycket kort och går snabbt att exekvera. I gengäld krävs fler iterationer, vilket i vårt fall kan ta tid om mycket räknearbete fordras för varje förlustfunktionsberäkning.
Algoritmen innehåller en hel del parametrar som ska justeras in.

7.2.2 KONJUGERADE RIKTNINGAR

Två vektorer x och y säges vara konjugerade med avseende på en matris Q om $x^TQy=0$.
Metoder som använder konjugerade riktningar har egenskapen att anpassa sina sökningar till "dalarnas" riktning, se skrivbordsexemplet i fig 7.1.

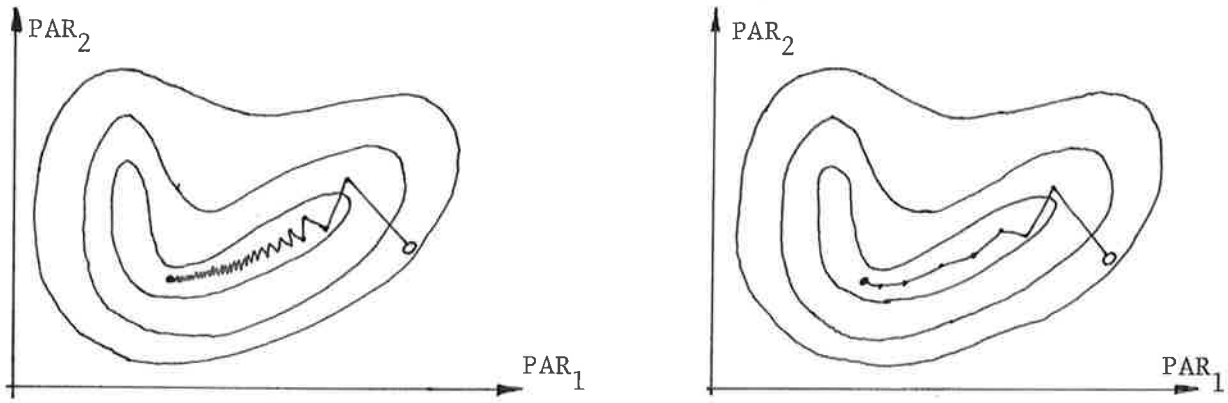


Fig 7.1 Exempel på sökningsväg för en enkel gradientmetod resp. konjugerad riktning metod.

Metoder som används flitigt är Davidon-Fletcher-Powell (1959), (1963) resp. Fletcher-Reeves (1964).

Metoderna har vanligen mycket snabb konvergens. Det rekommenderas att de återstartas från x_k då drygt n stycken sökriktningar provats. Om de får in felaktiga värden pga störningar märks dessa längre än för metoder med "kort minne".

7.2.3 PROGRAMPAKETET OPTA

Vid institutionen för reglerteknik vid Lunds tekniska högskola har framtagits ett programpaket för optimering av dynamiska system med avseende på vissa parametrar, Glad (1974). Det används under simuleringspaketet Simnon från samma inst. Metoden är baserad på en

kvasi-Newton metod och skattar derivatorna numeriskt.

I ett yttre skal innehåller OPTA rutiner för att kunna vara ett samplat system bland andra och för att kunna minimera under bivillkor. I en inre del finns själva minimeringsrutinen VA10A, Fletcher (1972). Denna rutin har även omfattande strategier för att finna bra steglängd s_k. Här kan nämnas ökande s_k vid lyckade försök, prov åt andra hållet vid miss, parabelskattning. Rutinen väljer också derivataberäkning med bakåt- eller centrala differenser. Det måste naturligtvis finns en mängd uthoppsställen så att inte rutinen fastnar i en sökprocedur.

Observera alltså att varje försök med nya regulatorparametrar innebär att man simulerar ett helt stegsvar så att förlustfunktionen kan beräknas. Trots alla finesser når man ofta bättre resultat genom att återstarta OPTA någon gång.

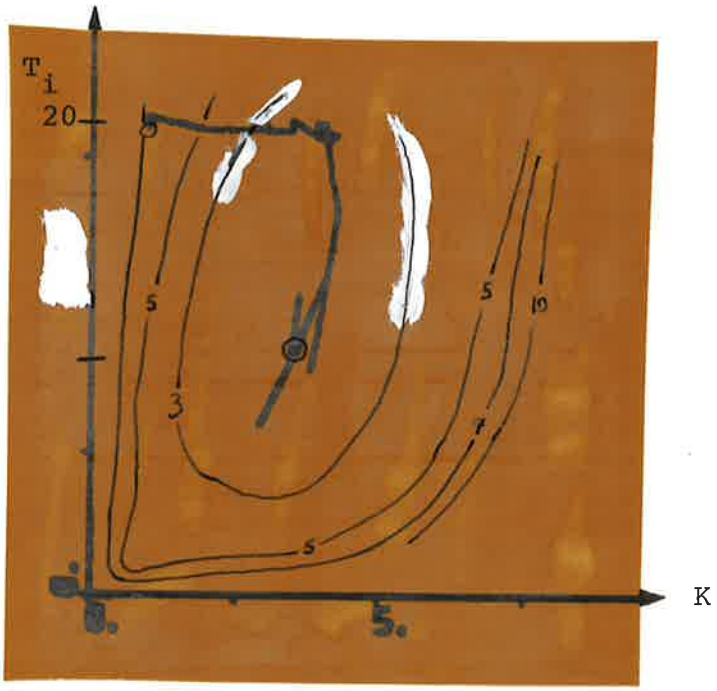


Fig 7.2 Konkret exempel på hur OPTA sökt sig fram då en PI-regulator optimerades för en 3.ordn. process. Nivåkurvorna (jämför 6.4) har lagts in i efterhand.

I denna rapport har termerna optimeringsrutin och minimeringsrutin använts synonymt.

8. SÖKSYSTEMETS FÖRVERKLIGANDE

"It's clever, but is it art?"
KIPLING

I kapitel 4 skissades söksystemets uppbyggnad. Både process och söksystem förverkligas som datorprogram i institutionens dator (jfr torrsim).
Efter den teoretiska exposén i kapitel 5-7 ges här en närmare redogörelse för programmen.
I avsnitt 1 beskrivs simuleringsprogrammen. Avsnitt 2 diskuterar programmens samverkan och valet av startparametrar.

8.1 SIMULERINGSPROGRAMMEN

8.1.1 SIMNON

Överordnat simuleringsprogram är SIMNON, Elmqvist (1975), utvecklat vid institutionen för reglerteknik, LTH. Programmet är interaktivt och medger snygg presentation av valda simuleringsvariabler. Systembeskrivningen görs som separata delprogram och ett speciellt "connecting system" kopplar samman dem. Varje delsystem beskrivs med differensekvationer (samplade system) eller ordinära differentialekvationer (kontinuerliga system). Det finns också möjlighet att inkludera egna program i Fortran. Syntaxen påminner om Algol-60.
För programlistorna nedan kan nämnas att Simnon räknar med flyttal och villkorsuttryck ges värdet 1 om de är sanna, 0 annars.

Bilden på nästa sida visar söksystemet så det provats (fig 8.1). Samtliga system har gjorts samplade för att göra simuleringstiden kortare.

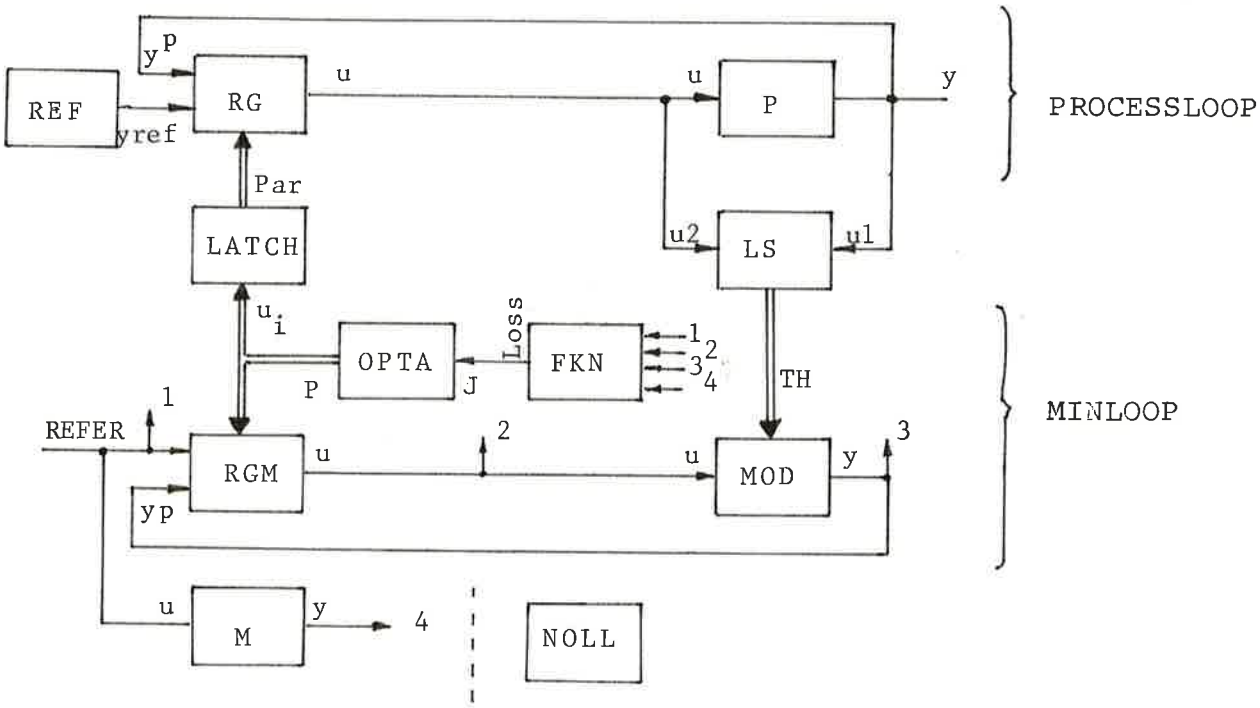


Fig 8.1 Söksystemets blockschema

8.1.2. PROCESSLOOPEN

■ REF Levererar en kantvåg med periodtiden PER till processloopen. Signalnivåerna valbara.

```
DISCRETE SYSTEM REF
TIME T
OUTPUT  Y
TSAMP  TS

OUTPUT
Y=IF MOD(T,PER*DT)<(.5*PER*DT-.000001) THEN NIV1 ELSE NIV2

DYNAMICS
TS=T+DT

NIV1:  1
NIV2:  0
DT:    1
PER: 100
END
```

■ RG PID-regulator med begränsning ULIM på utsignalens u belopp, en värdefull skyddsåtgärd. Integraldelen I approximerar integralen i (2.1) med $\frac{DT}{T_i}\sum_{o}^{t-1} e_k$ där e är felet, DT samplingstiden och T_i integraltiden. Vid utsignalbegränsning justeras I (reset windup). Derivatadelen fås som en

enkel bakåtdifferens $\frac{TD}{DT}(y_t - y_{t-1})$ där y är processens utsignal. Då processignalen är brusig kan derivatadelens belopp begränsas till DMAX genom att sätta parametrarna SYSL och SN. SYSL är processens stegsvarslutning och SN en faktor som anger hur mycket "snabbare" derivationen får vara. Förslag SN: 1-5. Eftersom regulatorparametrarna kan varieras utifrån är de ingångar till RG.
Anm. En variant är att uppdatera I redan i outputsection i RG så att även senaste felet e_t inkluderas i summan på förra sidan.

```
DISCRETE SYSTEM RG
"DISKRET PID-REG MED BEGR AV UTSIGNAL OCH RESET WINDUP,
"BEGR AV DERIVATATERMEN.
INPUT  YREF  YP K TI TD
OUTPUT U
STATE   I   X
NEW    NI  NX
TIME    T
TSAMP   TS

INITIAL
DMAX = SYSL*SN*DT

OUTPUT
E = YREF-YP
DSL = -TD * (YP-X)/DT
D = SIGN(DSL) * MIN( ABS(DSL), DMAX)
UT= K*(E+I+D)
U=IF ABS(UT)>ULIM THEN SIGN(UT)*ULIM ELSE UT
"ATERSTALL I-DEL EFTER BEGR.  (1+DT/TI) PGA SUMMERING T, T-1,
I1=IF ABS(UT) > ULIM THEN U/K - (1+DT/TI)*E - D ELSE I

DYNAMICS
NI = I1 + E/TI*DT
NX = YP
TS = T+DT

SYSL:  3
SN:    5
ULIM: 10
DT:  1
END
```

■ P Den aktuella processen (se 9.1)
Till processen räknas även ev. brusfiltrering om sådan är nödvändig.

■ LS Rekursiv minsta-kvadrat identifiering i Fortran. Den klarar upp till 5.ordningens system med tidsfördröjning $L \leq 15$ samplingsintervall. Ordningstalet N ska vara processens dubbla ordning. Programlistan i bilaga 1 innehåller skattningarna av parametrarna $a_1,..,a_n,b_1,..,b_n$ i (5.1) där 2n=N. Glömskefaktorn WT sättes WTI strax under 1 vid starten. Den ändras enligt (5.7), dvs om WTM är något större än 0 kryper WT exponentiellt mot 1.

8.1.3 MINIMERINGSLOOPEN

■ RGM PID-regulator i minloopen. Denna regulator ska i princip vara identisk med RG i processloopen. För att slippa omskalning av regulatorparametrarna ska DTRG sättas till samplingstiden för processregulatorn. Samplingstiden DTM för minloopen är normalt betydligt kortare än i processloopen.
För varje provstegsvar i minloopen ska vissa tillstånd och variabler nollställas. Detta sker för RGM del direkt i programmet.

```
DISCRETE SYSTEM RGM
"DISKRET PID-REG MED BEGR AV DERIVATATERMEN FÖR ANV I MINLOOPEN.
"OBS ATT DTRG SKA SATTAS = DT[RG] (SAMMA TIDSSKALA I REGULATORERNA).

INPUT  YREF  YP K TI TD
OUTPUT U
STATE   I   X
NEW    NI  NX
TIME    T
TSAMP   TS

INITIAL
DMAX = SYSL*SN*DTRG

OUTPUT
E = YREF-YP
DSL = -TD * (YP-X)/DTRG
D = SIGN(DSL) * MIN( ABS(DSL), DMAX)
U= IF MOD(T/DTM,PROVL)>PROVL-.999 THEN 0 ELSE K*(E+I+D)
"NOLLST FÖR ATT HALLA U=0 DA T/DTM=HELTAL*PROVL.
DYNAMICS
"NOLLST VID T/DTM=HELTAL*PROVL, DVS PROVSTEGSVAR SLUT.
NI=IF MOD(T/DTM,PROVL)>PROVL-.999 THEN 0 ELSE I + E/TI*DTRG
NX=IF MOD(T/DTM,PROVL)>PROVL-.999 THEN 0 ELSE YP
TS = T+DTM
SYSL:  3
SN:    5
DTM:  1
PROVL:50
DTRG: 1      "SAMPLINGSTID I PROCESSLOOPEN
"  OBS ATT I SIMNON BLIR MOD(X,X)=X, EJ 0.
END
```

■ MOD 5.ordningens diskret system på styrbar kanonisk form med koefficienterna a_i, b_i i (5.1) som ingångar. Koefficienterna levereras av identifieringsrutinen LS och gör att minloopen hela tiden använder den mest aktuella skattningen av processen. Om ordningstalet i LS har valts lägre än 5 (N<10) sätts de högre koefficienterna till 0 i connecting system KON. Då ett provstegsvar är klart nollställs tillstånden dels av systemet NOLL, dels i MOD själv.

```
DISCRETE SYSTEM MOD
"5.ORDN DISKR SYST. INSTAB.FARA OM >3.MULT POL LIGGER NARA ENHETSCIRKELN
"NOLLSTALLNING DELS I MOD, DELS AV NOLL DA PROVSTEGSVAR KLART.

INPUT  U A1 A2 A3 A4 A5 B1 B2 B3 B4 B5
OUTPUT Y
STATE  X1  X2  X3  X4  X5
NEW   NX1 NX2 NX3 NX4 NX5
TIME   T
TSAMP  TS

OUTPUT
Y=B1*X1+B2*X2+B3*X3+B4*X4+B5*X5

DYNAMICS
NN=-A1*X1-A2*X2-A3*X3-A4*X4-A5*X5 + U
"NOLLST VID T/DTM=HELTAL*PROVL, DVS PROVSTEGSVAR SLUT.
NX1=IF MOD(T/DTM,PROVL)>PROVL-.999 THEN 0 ELSE NN
NX2=X1
NX3=X2
NX4=X3
NX5=X4
TS=T+DTM

DTM:  1
PROVL:50
END
```

■ FKN Beräkning av förlustfunktionen för PROVL punkter på ett provstegsvar i minloopen.
Med parametern NOM väljer man vilket ingång som ska användas vid beräkningen av felet e. NOM=0 ⇒ ⇒ e = börvärdet REFER - y^{MOD}, NOM=1 ⇒ e = y^M - y^{MOD}, där y^M är utsignalen från en känt system med önskvärd dynamik och samma insignal som minloopen.
Med parametern SQ väljes mellan förlustfunktionerna

$$(1): \sum_{t=2}^{PROVL-1} e_t^2 \quad (ISE) \quad \text{resp} \quad (0): \sum_{t=2}^{PROVL-1} t|e_t| \quad (ITAE).$$

Önskar man dessutom lägga till straff på <u>ändringar i styrsignalen</u> till systemet (MOD) sättes parametern (λ) LA>0 [$\lambda \cdot \sum(\Delta u_t)^2$ adderas]. Variabeln STRAF ger tillägg till förlustfunktionen för <u>negativa värden</u> på regulatorparametrarna och adderas automatiskt. FKN innehåller en variabel R (även utgång RR) som räknar hur många samplingsintervall provstegsvaret i minloopen pågått (används för nollställning och ber av ITAE). Själva förlustfunktionen summeras i variabeln <u>SUM</u> och just innan provstegsvaret ska sluta laddas det uppnådda värdet över till variabeln <u>LOSS</u> som håller värdet tillgängligt under nästa provstegsvar. Genom få omkopplingar och byte av samplingsintervall kan FKN användas att mäta upp förlustfunktionen i processloopen i stället.
Nollställning sköts dels av FKN själv, dels av systemet NOLL.

```
DISCRETE SYSTEM FKN
"BERAKNING AV FÖRLUSTFUNKTION
INPUT U Y YNOM YREF  K TI TD
OUTPUT  RR
TSAMP  TS
TIME   T
STATE SUM UOLD  R LOSS
NEW   NS  NOLD NR NL
"LOSS AR SLUTVARDET FÖR FÖRLUSTFKN (PROVL-2 TERMER).

OUTPUT
RR=R

DYNAMICS
E=(IF NOM THEN YNOM ELSE YREF) - Y
DU=U-UOLD
"STRAFF PA SMA ELLER NEGATIVA REGULATORPARAMETRAR
STRA=(IF K<.01 THEN 10*(.01-K) ELSE 0)+(IF TI<.01 THEN 10*(.01-TI) ELSE 0
STRAF = STRA + (IF TD<0 THEN -10*TD ELSE 0)
SNS=SUM + LA*DU*DU + (IF SQ THEN E*E ELSE (R-1)*ABS(E)) + STRAF
NR=R+1
NL=IF R>PROVL-1.999 AND R<PROVL THEN SNS ELSE LOSS
NS=IF R<1.99 THEN 0 ELSE SNS
NOLD=IF R<1.99 THEN 0 ELSE U
TS=T+DTM

LA:  0
SQ:  1
NOM: 0
PROVL: 50
DTM: 1
END
```

■ OPTA Optimeringsrutin som kallas in då varje provstegsvar är klart och beräknar nya förslag till regulatorparametrar (utsignalverktor P, PD), se även 7.2.3. Insignal är enbart värdet på förlustfunktionen LOSS från förra uppsättningen regulatorparametrar. Mindre än 10 parametrar och startvärden behöver ändras manuellt före körningens början.

■ M M är ett andra ordningens system $\frac{\omega_0^2}{s^2+2\zeta\omega_0 s+\omega_0^2}$ som samplats med intervallet DTM. ζ är relativa dämpningen och ω_0 anger systemets snabbhet. Se fig 8.2. M matas med samma insignal som minloopen och dess utsignal används vid beräkningen av förlustfunktionen då modellreferens förekommer (se 6.1 och 9.3). Genom att ange Z (ζ) och ONOLL (ω_0) kan man mha fig 8.2 specificera önskat beteende för sitt system. Samplingstiden DTM är samma som för systemet MOD och nollställning vid nytt provstegsvar sköts av NOLL.
Programmet listas i bilaga 2.

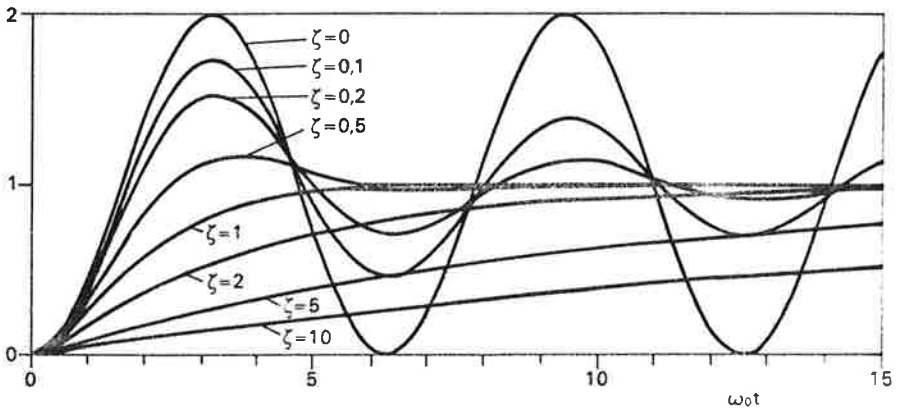


Fig 8.2 Stegsvar för ett system med överföringsfunktionen

$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 + \omega_0^2}.$$

■ NOLL Nollställningsrutin. Detta system aktiveras varje gång ett provstegsvar i minloopen är klart. Det nollställer de tillstånd som delsystemen inte själva

ställer. Samplingstiden DTN beror av minloopens provtid och samplingsintervall, se avsnitt 8.2.1.

```
DISCRETE SYSTEM NOLL
"DETTA SYSTEM NOLLSTALLER TILLSTANDEN I MINLOOPEN VARJE GANG
"   ETT NYTT PROVSTEGSVAR KÖRS DAR. RESET[OPTA]=0.
"VISSA TILLSTAND OCH SIGNALER NOLLSTALLS DIREKT I P31,D5,FKN.

TIME  T
TSAMP TS

OUTPUT
S1[M]=0
S2[M]=0
X1[MOD]=0
X2[MOD]=0
X3[MOD]=0
X4[MOD]=0
R[FKN]=0

DYNAMICS
TS=T+DTN
DTN:  1
END
```

8.1.4 ÖVRIGA SYSTEM

■ LATCH Detta system överför regulatorparametrar från minloopen (OPTA) till regulatorn RG i processloopen. Vart UPER samplingsintervall DT i processloopen uppdateras processregulatorns RG parametrar.
Då hela simuleringen startar matas startvärdena (1,20,0 vid PID) över till regulatorn RG.
Startvärdena ändras med kommandot INIT. Det kan vara värdefullt vid återstarter. Parametervärdena hålls konstanta tills förlustfunktionen börjar understiga LMAX. Därefter byts utparametrarna bara om motsvarande förlustfunktion är bättre än de närmast två föregående (då LATCH exekverade). Kort minne används eftersom LOSS kan bli litet någon gång innan identifieringsrutinen LS funnit en bra bild av processen P. Det är viktigt att fördröjda parametrar från OPTA (PD1,..) matas in till Latch eftersom förlustfunktionen inte finns tillgänglig förrän provstegsvaret är klart.
Samplingsintervall: DT som i processloopen.

```
DISCRETE SYSTEM LATCH
"ÖVERFÖRING AV REG.PAR TILL PROCESSLOOPEN VART UPER SAMPL.INTERVALL.
"ENDAST BASTA VARDE AV T, T-1, T-2 ÖVERFÖRES.
INPUT U1 U2 U3
OUTPUT PAR1 PAR2 PAR3
STATE L1  L2  L3 LOL  LEOL
NEW   NL1 NL2 NL3  NO  NE
TIME T
TSAMP TS

INITIAL
L1:1
L2:20
L3:0
LOL=LMAX
LEOL=LMAX

OUTPUT
PAR1 = L1
PAR2 = L2
PAR3 = L3

DYNAMICS
"PA INGANGARNA FORUTS PAR.VARDEN SAMMANHÖRANDE M AKT LOSS[FKN].
BRA = IF LOSS[FKN]<LEOL AND LOSS[FKN]<LOL AND LOSS[FKN]<LMAX THEN 1 ELSE 0
NL1 = IF BRA THEN U1 ELSE L1
NL2 = IF BRA THEN U2 ELSE L2
NL3 = IF BRA THEN U3 ELSE L3
NE = LOL
NO = LOSS[FKN]
TS = T + DT*UPER

UPER: 2
DT:   1
LMAX:  20
END
```

■ KON Ett sk connecting system som beskriver hopkopplingen av delsystemen.

En sats ser vanligen ut som X(A)=Y(B). Den innebär att variabeln X i system A sätts lika med värdet Y i system B. Hela KON genomsökes varje gång <u>något</u> delsystem ska exekveras. <u>Om</u> det är dags att exekvera B görs detta <u>innan</u> X får sitt värde.
Det måste naturligtvis vara så att Y(B) kan beräknas just då. Om nämligen beräkningen av Y kräver att X har ett värde har vi fått en algebraisk loop, vilket är förbjudet (jämför hönan och ägget).
Algebraiska loopar undvikes genom att införa avsiktliga fördröjningar i transport av värden mellan delsystem och genom att alla Output-sektioner i delsystemen exekveras för Dynamics-sektionerna.
I detta fall medverkar följande system att bryta möjliga algebraiska loopar: FKN, LATCH, MOD samt de använda processystemen P.

I avsnitt 9.2 kan samverkan mellan blocken studeras i ett optimeringsexempel som genomgås i detalj.

```
CONNECTING SYSTEM KON
TIME T

"PROCESSLOOPEN
YREF[RG]=Y[REF]
YP[RG]=Y[RTH]      "PROC
K[RG]=PAR1[LATCH]
TI[RG]=PAR2[LATCH]
TD[RG]=PAR3[LATCH]

U[RTH]=U[RG]     "PROC

U2[LS]=U[RG]
U1[LS]=Y[RTH]    "PROC

"MINIMERINGSLOOPEN
REFER = IF RR[FKN]<1.999 OR RR[FKN]>PROVL-.999 THEN 0 ELSE NIV
YREF[RGM]=REFER
LOSS[OPTA]=LOSS[FKN]
YP[RGM]=Y[MOD]
K[RGM]=P1[OPTA]
TI[RGM]=P2[OPTA]
TD[RGM]=P3[OPTA]  " =0 SA LANGE NPAR.OPTA=2.

A1[MOD]=TH1[LS]
A2[MOD]=TH2[LS]
A3[MOD]=TH3[LS]
A4[MOD]=0
A5[MOD]=0
B1[MOD]=TH4[LS]
B2[MOD]=TH5[LS]
B3[MOD]=TH6[LS]
B4[MOD]=0
B5[MOD]=0
U[MOD]=U[RGM]

U[M]=REFER
YNOM[FKN]=Y[M]
YREF[FKN]=REFER
U[FKN]=U[RGM]
Y[FKN]=Y[MOD]
K[FKN]=P1[OPTA]
TI[FKN]=P2[OPTA]
TD[FKN]=P3[OPTA]       "=0 DA NPAR.OPTA=2

U1[LATCH]=PD1[OPTA]
U2[LATCH]=PD2[OPTA]
U3[LATCH]=PD3[OPTA]  "=0 SA LANGE NPAR.OPTA=2

NIV:   1
PROVL: 50
END
```

8.2 VAL AV PARAMETRAR

8.2.1 PROVTIDENS LÄNGD I MINLOOPEN

PROVL är antalet samplingar per provstegsvar i minloopen. Antag att PROVL=50. Då fås följande tidsschema i minloopen:

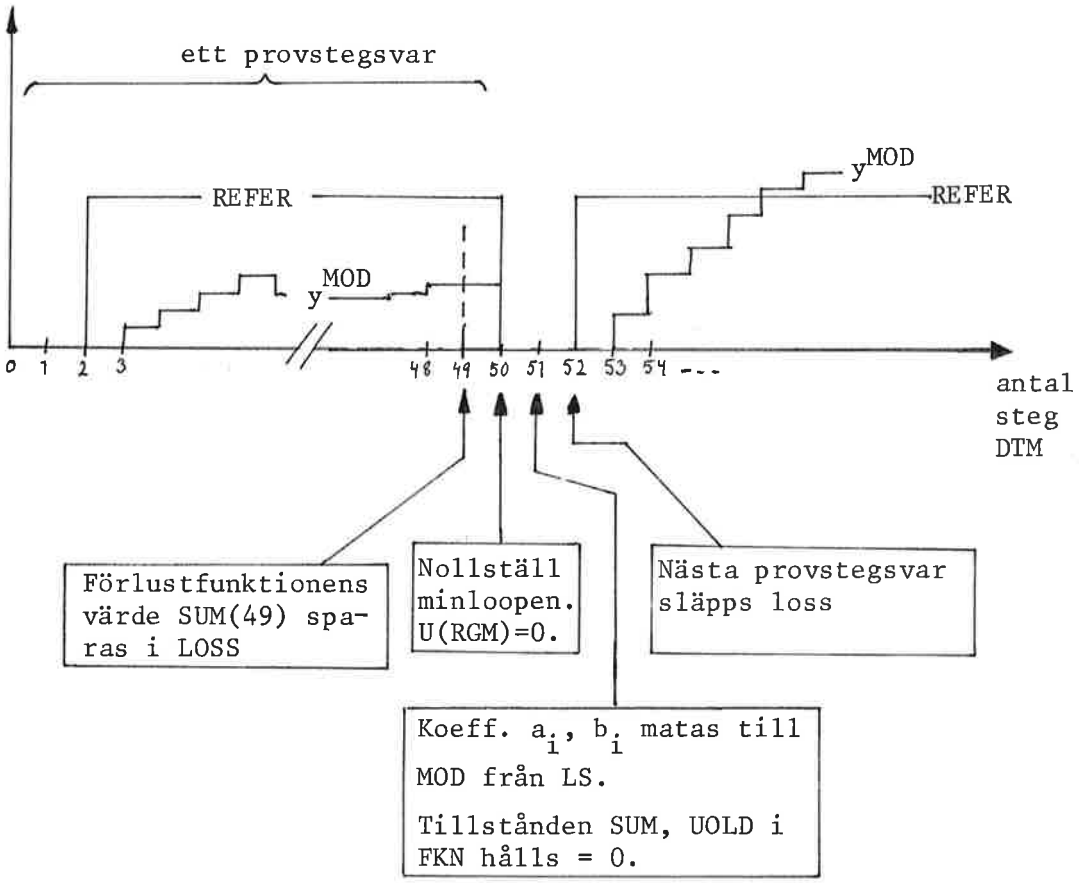


Fig 8.3 Tidsschema över minloopens händelser

Antalet termer i förlustfunktionen blir alltså PROVL-2 eftersom referenssignalen i minloopen (REFER) stiger från 0 vid andra samplingstidpunkten och LOSS i FKN får värdet SUM(PROVL-1). Viloperioden ovan gör att man blir oberoende av exekveringsordningen för de olika delsystem som ska exekveras vid samma tidpunkt.

Under provkörningarna har PROVL valts till 50 (stegsvaret har lugnat sig), UPER=2 och DT (samplingsinter-

vall i processloopen) = 1. Enligt (4.1) blir DTM=0.04. Detta innebär att ett provstegsvar med 50 samplingar i minloopen tar samma tid som 2 samplingar i processloopen. Därför ska DTN[NOLL]= TINC[OPTA]=2 då DT=1. Se även bilaga 3.

8.2.2 STARTVÄRDEN

Tabellen nedan är en utskrift av parameterfilen PAR. Den innehåller startvärden för tillstånd och parametrar för en simulering med 3.ordningens processen RTH, optimering av en PID-regulator enligt kriteriet ISE. Med kommandot GET PAR laddas värdena till Simnonsystemen.

Tabell 8.1 Parameterfilen PAR

```
[REF]
PER:100.
DT:1.
NIV1:1.
NIV2:0.
[RG]
I:0.
X:0.
SYSL:3.
SN:5.
DT:1.
ULIM:10.
[RTH]
X1:0.
X2:0.
X3:0.
B1:0.03178
B2:0.06085
B3:6.451E-3
A1:-1.322
A2:0.4618
A3:-0.04076
DT:1.
[LS]
L:0.
N:6.
WTI:0.9
WTM:0.01
DT:1.
PP:100.
IT1:0.1
IT2:0.1
IT3:0.1
IT4:0.1
IT5:0.1
IT6:0.1
IT7:0.1
IT8:0.1
IT9:0.1
IT10:0.1
[RGM]
I:0.
X:0.
SYSL:3.
SN:5.
DTRG:1.
DTM:0.04
PROVL:50.
[MOD]
X1:0.
X2:0.
X3:0.
X4:0.
X5:0.
DTM:0.04
PROVL:50.
[M]
S1:0.
S2:0.
ONOLL:40.
Z:0.8
DT:0.04
[FKN]
SUM:0.
UOLD:0.
R:0.
LOSS:0.
NOM:0.
LA:0.
SQ:1.
PROVL:50.
DTM:0.04
[NOLL]
DTN:2.
[LATCH]
L1:1.
L2:20.
L3:0.
LOL:0.
LEOL:0.
LMAX:60.
DT:1.
UPER:2.
[OPTA]
PI1:0.3
PI2:20.
PI3:0.
XM1:1.
XM2:10.
XM3:1.
DFN:-0.5
TINC:2.
HH:0.01
EPS:0.
PRIN:0.
EVMAX:1.E4
CEQ:0.
C:1.
DELTA:0.01
RESET:0.
DARK:0.
MODE:1.
LPLOT:0.
[KON]
PROVL:50.
NIV:1.
```

8.2.3 VAL AV VISSA PARAMETRAR (jämför tab 8.1)

PER[REF]: Vid simulerings början tjänar man på att hålla PER måttlig, tex 40, för att underlätta identifieringen. Mot slutet kan med fördel PER ökas, t ex 100, så man tydligare ser hela stegsvaret för processen (samma längd som i minloopen). Då kan man göra en motsvarande mätning av förlustfunktionen i processloopen. Om PER är litet bör det vara udda.

ULIM[RG]: Nyttig försiktighetsåtgärd att hindra regulatorn att ge extrema styrsignaler. Detta kan ske någon gång då sökningen inte är klar (se även Ll[LATCH]).

N[LS]: N sättes =2n. Praktisk erfarenhet är värdefull när det gäller att gissa processens ordningstal n. Man kan börja måttligt, säg N=4, och se på V[LS].
$V_{N+1} = 0.9V_N$ + RNORM, där RNORM är kvadratsumman på ändringen i varje komponent i skattningsvektorn TH. V är alltså en sorts glidande "medelvärde" av korrektionen av skattningen. (Med viktfaktor 0.9 kommer det att dröja ca 20 steg innan $RNORM_k$ har dämpats 90%.) V bör minska märkbart då man ökar N och träffar rätt ordningstal (om inte bruset är för starkt). V ger dock inte allt.
Ett annat symptom på att N är för litet är att parameterskattningarna gör små språng varje gång referensvärdet ändras stegvis.
Ett incitament att öka N är om de styrsignaler som processen möter "i verkliga livet" avviker starkt från minloopens styrsignal eller om man misstänker att färgat brus förekommer.
Enligt avsnitt 5.1 bör ökande N åtföljas av ökande PER[REF],kanske PER>5N.

L[LS]: Processens rena tidsfördröjning mätt i antalet samplingsintervall (DT). Nuvarande rutin LS medger L upp till 15. En yttre rutin skulle kunna variera L. Det normala är att man mäter upp fördröjningen och sätter $L = \frac{\text{fördröjn i sek}}{\text{DT i sek}}$.

WTI[LS]: Startvärde för glömskefaktorn i identifieringsrutinen. Värden 0.9-1 används. 0.9 används vid svåridentifierade eller tidsvariabla processer.
WTI<1 ställer vissa krav på exciteringen, se avsn 5.1.

WTM[LS]: Konvergensfaktor för glömskefaktorn; anger hur snabbt denna går mot 1. Normala värden är 0.01-0.001. Det dröjer ca 1/WTM steg innan WT nått 1.

IT1 - IT10[LS]: Startvärden för processkattningen som matas ut från identifieringsrutinen. Dessa kan vara 0, men har satts till 0.1 för att minimeringsrutinen ska få några värden att arbeta med.
Om man återstartar optimeringen bör naturligtvis IT sättas till bästa skattning av processen.

DTRG[RGM]: Sätts lika med samplingsintervallet DT i processloopen och används för att få samma skalning av regulatorparametrarna i min- och processloopen.

Z, ONOLL[M]: Motsvarar ζ, ω_0 i ett 2.ordningens system. Förekommer då man vill specificera en "idealprocess" vid modellreferens. Mha fig 8.2 väljes parametrarna ω_0 och $\zeta \neq 1$.

DTN[NOLL]: Tid för ett helt provstegsvar i minloopen, se avsnitt 8.2.1.

LMAX[LATCH]: Gränsvärde för förlustfunktionen för att regulatorparametrar ska få överföras till processregulatorn. LMAX bör sättas liberalt: Dels vet man inte alltid hur stor förlustfunktionen kommer att bli. Dels kan ju processen vara ofullständigt identifierad (förlustfunktionen beräknas ju mha skattningen) men ändå så riktig att processregulatorn kan ha glädje av parametrarna.
Under proven sattes LMAX=60 då förlustfunktionerna kunde krypa ner till 2-10.

L1,L2,L3[LATCH]: Startvärden för de parametrar (K, T_i, T_d) som matas ut till processregulatorn RG och används tills optimeringsrutinen funnit någorlunda bra (se

LMAX) parametrar. Om man inte vet rimliga värden får man sätta L1 och L3 (K och T_d) små och L2 (T_i) stor för en PID-regulator.
Vid återstarter av optimeringen och produktionsomställningar kan man använda senaste regulatorvärden. Genom att sätta utsignalbegränsningen ULIM i processregulatorn snålt, kan man våga dra på mer "startförstärkning" (L1) i LATCH och därmed få en snabbare identifiering av processen.

För optimeringsrutinen OPTA kan följande parametrar vara av intresse:

PI1, PI2,.. Startvärden för regulatorparametrarna (K, T_i, T_d) i minloopen, alltså vid start av optimeringen. Vet man inte vad man ska sätta, ansätter man en "försiktig" regulator eller använder tumreglerna. Vid återstarter ska bästa värden från tidigare användas. Genom att göra körningar från några olika startpunkter finner man lätt optimala regulatorparametrar. Om man råkar få startvärden som resulterar i en mycket stor förlustfunktion LOSS kan i besvärliga fall numeriska svårigheter uppstå pga att OPTA ser på skillnader i LOSS och antalet signifikanta siffror är högst 8 i PDP-15 datorns Simnon.

XM1, XM2,.. Skalfaktorer som påverkar steglängden vid derivataberäkningarna. Om ändringar i en viss regulatorparameter nästan inte påverkar förlustfunktionens värde vid aktuell regulatorversion bör man öka motsvarande XM_i.
Ex: Då PI-regulatorn skrives $u_t = K(e_t + \frac{DT}{T_i}\sum e_k)$ är känsligheten i T_i-parametern ofta låg.
XM2=20 kan då vara ett lämpligt värde.

TINC: Samplingsintervall för OPTA.
Detta ska vara lika med tiden för ett provstegsvar i minloopen, jämför avsnitt 8.2.1.

HH: Steglängd vid beräkning av derivator med differenser. Steglängden i varje riktning är $XM_i \cdot HH = \Delta x_i$. Allmänt ger ett litet HH en god bild av gradienten i den aktuella punkten, men om förlustfunktionens värde där är stor räcker möjligen inte antalet siffror för att ge signifikans åt skillnaden:

$$\text{grad(LOSS)} = \left(\frac{LOSS(\bar{x}+\Delta\bar{x}_1)-LOSS(\bar{x})}{|\Delta\bar{x}_1|}, \frac{LOSS(\bar{x}+\Delta\bar{x}_2)-LOSS(x)}{|\Delta\bar{x}_2|}, \ldots \right)$$

där $\bar{x}$ är koordinaterna för punkten och $\Delta\bar{x}_i$ är en kort riktningsvektor i riktningen i.

Det visar sig ibland bra att börja med ett rejält HH, säg 0.1 och sedan göra en återstart med HH mindre, t ex 0.01. Med ledning av hur regulatorparametrerna ändrat sig under körningen kan man justera skalfaktorerna XM_i om så behövs.

RESET: OPTA innehåller rutiner som automatiskt återställer alla tillstånd som ingår i simuleringen till startvärden efter varje provstegsvar om RESET>0.
Eftersom blocket NOLL mm finns ska RESET=0.

EPS: Uthoppsvillkor ur minimeringsrutinen OPTA, sättes 0 så att inte optimeringen stannar annat än på kommando.

9. SIMULERINGAR

'Aldrig förr i historien har det varit möjligt att frambringa så många felaktiga resultat på så kort tid.'

Söksystemet har simulerats på dator för olika kombinationer av processer och regulatorer. Här ska redovisas några erfarenheter av dessa prov, medan sammanfattande synpunkter ges i nästa kapitel.
Avsnitt 1 orienterar om förutsättningarna vid simuleringen. Andra avsnittet beskriver en optimerings förlopp mera i detalj. I avsnitt 3 visas några konkreta simuleringsresultat i form av stegsvar. Kapitlet avslutas i avsnitt 4 med några praktiska synpunkter, bl a vad man gör om sökningen inte lyckas första gången.

9.1 DATOR OCH PROCESSER

Datorn PDP-15 är en större minidator med en mängd kringutrustning som plotter, skivminne osv. Ordlängden är 18 bitar och reella variabler upptar 2 ord, vilket ger ca 8 signifikanta siffror. Simuleringspaketet Simnon har använts, Elmqvist (1975). Det är bara meningsfullt att köra Simnonpaketet på datorer med allra minst 32 k ord kärnminne samt avancerad filhantering och overlaymöjlighet. Att skriva en liten interpreterare för kommandon och administrationsrutin som är skräddarsydd för simuleringar med detta söksystem uppskattas att behöva högst 3 k ord minne.

Som "verklig process" har använts 2. och 3.ordningens processer utan svängande stegsvar. Programlistningarna ges i bilaga 2. För att snabba upp simuleringarna realiserades dessa som samplade system.

<u>SEC</u>: $G_2 = \frac{1}{(1+0.5s)(1+s)}$ ger vid sampling med DT=1:

$$H_2 = \frac{0.399577z^{-1}+0.146996z^{-2}}{1-0.503215z^{-1}+0.049787z^{-2}} = \frac{b_1z^{-1}+b_2z^{-2}}{1+a_1z^{-1}+a_2z^{-2}} \qquad (9.1)$$

<u>RTH</u>: $G_3 = \frac{1}{(1+0.5s)(1+s)(1+5s)}$ som med DT=1 ger:

$$H_3 = \frac{0.031776z^{-1}+0.0608495z^{-2}+0.0064509z^{-3}}{1-1.32195z^{-1}+0.46178z^{-2}-0.040762z^{-3}} \qquad (9.2)$$

H_3 visade sig något svårt att sidentifiera, men lyckades med glömskefaktor WT=0.9.
z^{-1} betecknar skiftoperatorn bakåt: $z^{-1}u(t) = u(t-1)$.

Även en process IMFAS $= \frac{2(0.5-s)}{(1+0.5s)(1+s)(1+5s)}$ (icke-minimum fas) har provats med gott resultat.

9.2 ETT OPTIMERINGSEXEMPEL

Hur en optimering startas upp visas i bilaga 4.
Nedan visas några signaler vid optimering av en PI-regulator RG enligt förlustfunktionen ISE på processen SEC.
Hela söksystemets blockschema visas i fig 8.1.

Samplingsintervallet DT för processen SEC och identifieringsrutinen LS är 1 varför dessa system kan ändra utsignaler för t=1,2,3,... Minimeringsrutinen OPTA exekveras efter varje provstegsvar i minloopen dvs t=2,4,6,..., så att regulatorparametrarna (1, 2) i fig 9.1 ändrar sig bara då. Likaså ändrar sig värdet av förlustfunktionen LOSS (3) vid jämna tidpunkter.

Med hänvisning till fig 9.1 på nästa sida kan man med någon möda urskilja följande ur kurvornas virrvarr:

a) REGULATORPARAMETRARNA (1, 2)
Startvärdena K, T_i är 0.5, 20. De ses i t=0-2.
Härur evalueras LOSS för denna regulatorinställning: presenteras i t=2-4. Nu ska OPTA beräkna gradienten i denna punkt (K, T_i). Därför ändras först K (se 1) i t=2-4 och sen T_i (2) i t=4-6. Nu har OPTA tre värden på LOSS i tre näraliggande punkter (K, T_i) och (-1)·gradienten beräknas vid t=6. Detta blir första sökriktningen.

LOSS (3) är fortfarande stor (utanför figuren), men OPTA tar allt längre steg i denna riktning (se 1, 2 för t=10-16) och nu börjar LOSS minska märkbart. För t=16,18,20 görs som synes en ny gradientberäkning (se LOSS 3, 1, 2). Nära minimum tas bara små steg i regu-

latorparametrarna. Efter t=48 sker inget viktigt mer; man kan skönja några gradientberäkningar.

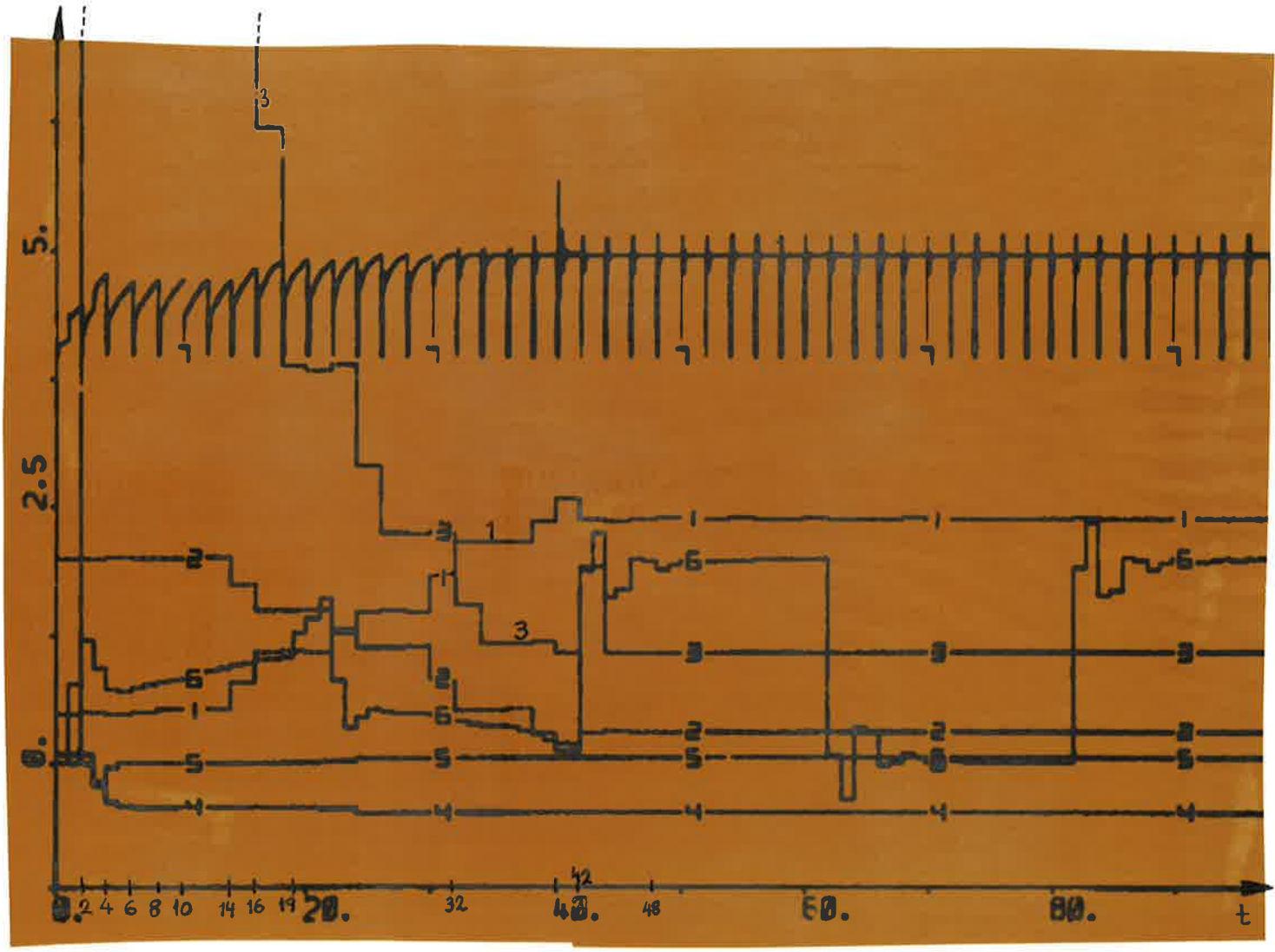


Fig 9.1 Optimeringsförlopp för SEC + PI-regulator, ISE.
(HH=0.01, XM2=10, WTI=0.9, WTM=0, N=4, SQ=1, ULIM=10, PER=40)
Signalerna är:

1 - förstärkning K för regulatorn i minloopen
2 - integraltid $T_i/10$ f s. (nerskalad av plottningsskäl)
3 - förlustfunktionen LOSS Σe_t^2 från föreg. provstegsvar
4-5 - ex på skattning från LS: a_1, a_2 i SEC, se (9.1)
6 - utsignalen i processloopen, y^P
7 - utsignalen i minloopen (provstegsvar), y^{MOD}.

b) IDENTIFIERINGEN AV PROCESSEN (4, 5)

Bara två av de fyra koefficienterna a_i, b_i ritas ut. Identifieringen av a_1, a_2 i SEC (se H_2, (9.1)) går ganska fort. Båda kurvorna (4, 5) utgår från 0.1. När några utsignaler (6) från SEC visat sig kommer de första skattningarna och vid t=4 kan slutvärdena anas.
Varje gång processen styres ut har LS en chans att förbättra skattningen, t ex vid t=20,40,.. då referensvärdet i processloopen ändras. Processen styrs också ut om regulatorn får nya parametrar från minloopen, vilket man kan se vid t=19 i utsignalen (6). LOSS (3) har blivit liten och därför släpps de fram till processregulatorn.
Varje gång identifieringsrutinen är klar med en ny skattning matas koefficienterna upp till modellen MOD och det sker vid t=1,2,3,... Se t ex på utsignalen (7) hur modellen blivit en annan vid t=1.

c) UTSIGNALEN I MINLOOPEN, y^{MOD}, (7)

Av plottningsskäl har nollpunkten förlagts till 4.
Ett provstegsvar körs under två tidsenheter DT och samplas under denna tid 50 gånger. I början är stegsvaren försiktiga eftersom startvärdena i OPTA för regulatorn RGM är "lågt satta". Från t=16 och framåt stiger förstärkningen en stund och man ser hur stigtiden för (7) minskar. Till slut kommer överslängarna vid t=32. Vid t=40 försökes en ganska svängig regulatorinställning i minloopen vilket tydligt syns.
Mycket värre instabiliteter kan ibland uppträda under några steg. Det gör inget eftersom OPTA snart finner stabilare värden igen och blocket LATCH hindrar i de flesta fall att processloopen får dessa stora värden.

d) UTSIGNALEN I PROCESSLOOPEN, y^{SEC}, (6)

Som synes är redan tredje stegsvaret bra och avviker inte mycket från det slutliga (fig 6.2.b). Några våldsamheter förekommer aldrig.
Regulatorn i processloopen får sina startvärden från

L1,... i LATCH (försiktigt satta till 1, 20). Först när förlustfunktionen LOSS börjar bli acceptabel (<LMAX =60 i LATCH) släpps värdena från minloopen igenom (t=14). Med periodtiden 40 blir varje stegsvar i processloopen 20 samplingsintervall långt.

e) FÖRLUSTFUNKTIONEN LOSS (3)

Startpunktens parametrar gav tydligen ett mycket stort värde på LOSS. Längre fram minskar LOSS mycket angenämt utom i t=40, LOSS(42). Då detta inträffar plockar OPTA fram bästa föregående punkt och "glömmer misstaget". Värdet av LOSS vid en viss tidpunkt hänför sig till stegsvar (7) och regulatorparametrar(1, 2) två tidsenheter DT bakåt, dvs senast fullbordade provstegsvar.

Denna gång räckte det tydligen med en körning. Hur man återstartar optimeringen beskrivs i avsnitt 9.4.

9.3 NÅGRA SIMULERINGSRESULTAT

Nedan visas några slutliga stegsvar tillsammans med vissa simuleringsdata. I avsnitt 6.2 om förlustfunktioner finns fler stegsvar uppritade för 2.ordningens processen SEC.

I den mån optimeringarna konvergerar beror stegsvarets utseende på förlustfunktonen. För jämförelsens skull medtas även kurvor på processens stegsvar enligt tumregel (Ziegler-Nichols 1).

I en del fall har optimeringen fått återstartas. Antalet får inte uppfattas alltför typiskt eftersom optimeringens förlopp bl a beror på valda startvärden och steglängd HH. När man använder ett färdigt optimeringsprogram kan man inte påverka optimeringens gång (t ex återställa andraderivatmatrisen då rutinen inte hittar bättre värden längre).
Ungefärliga antalet provstegsvar i minloopen anges. Mot-

svarande antal i processloopen är 5-10% därav.
Skrivsätt: (aa.o) där aa=ungefärlig simuleringslängd i antal provstegsvar, o =antal återstarter.
Ex: för optimeringen i avsnitt 9.2: (30,0).

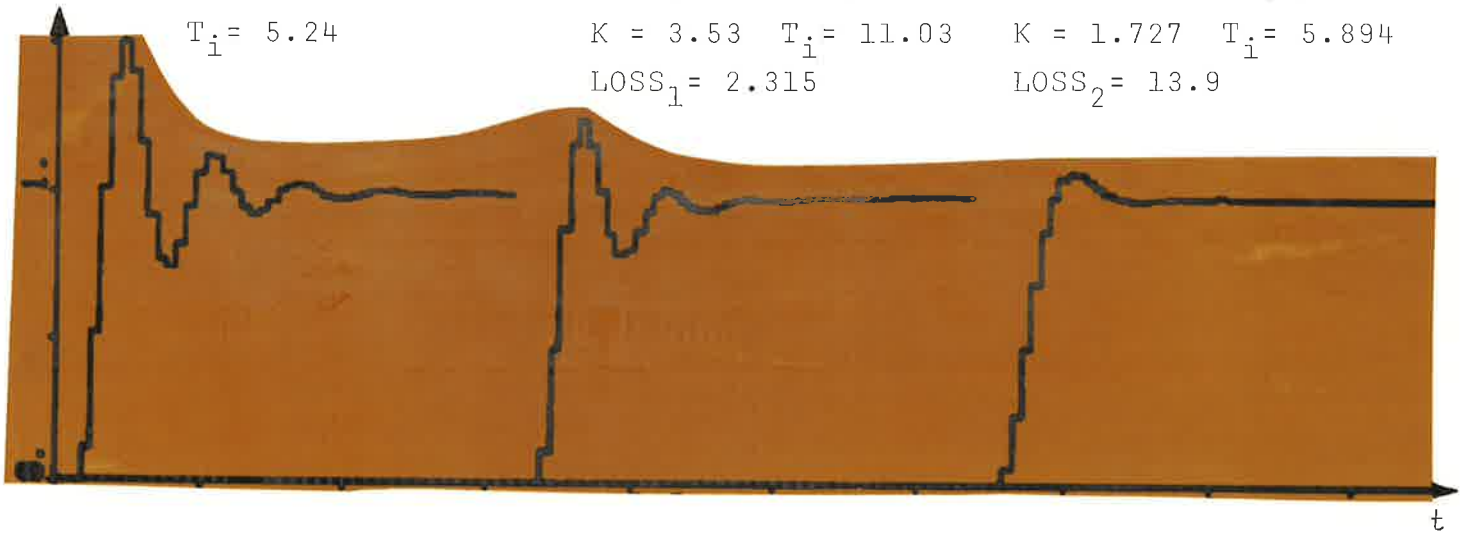
PI-regulator på 3.ordningens process RTH (formel (9.2))

Fig 9.2

a. tumregel
$K = 3.87$
$T_i = 5.24$

b. förlustfunktion ISE. (175,2)
$K = 3.53$ $T_i = 11.03$
$LOSS_1 = 2.315$

c. förlustfunktion ITAE. (130,2)
$K = 1.727$ $T_i = 5.894$
$LOSS_2 = 13.9$



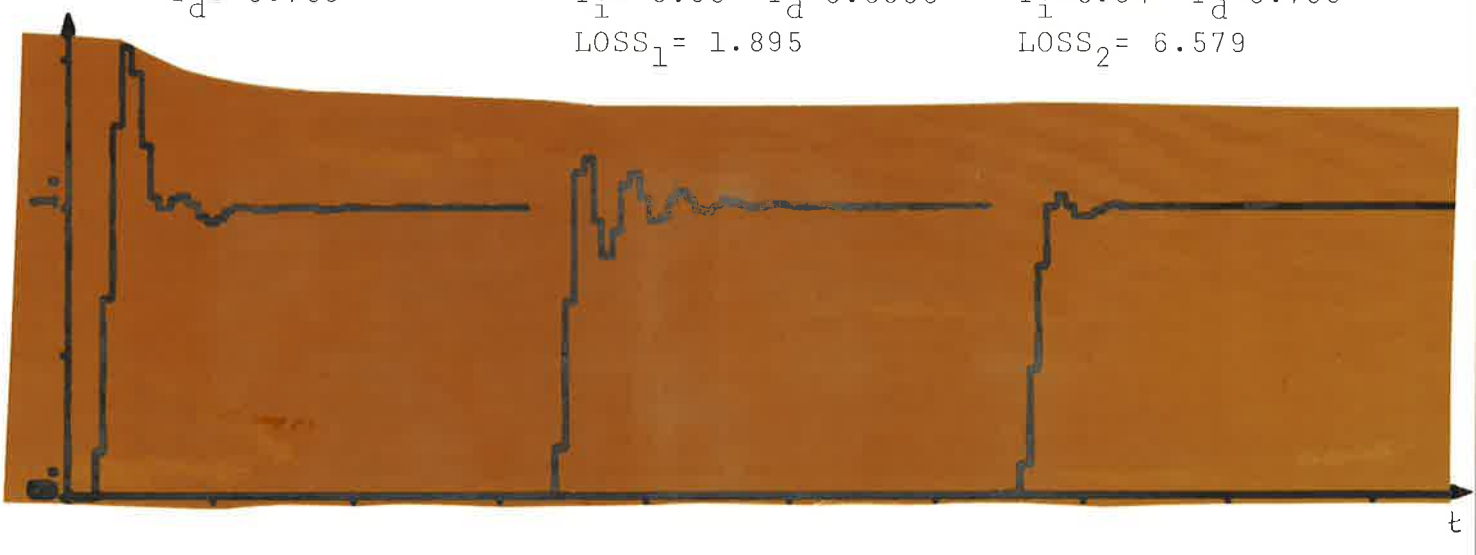
PID-regulator på RTH

Fig 9.3

a. tumregel
$K = 5.16$
$T_i = 3.143$
$T_d = 0.786$

b. förlustfunktion ISE (130,2)
$K = 5.322$
$T_i = 9.95$ $T_d = 0.8905$
$LOSS_1 = 1.895$

c. förlustfunktion ITAE (100,1)
$K = 3.343$
$T_i = 8.94$ $T_d = 0.769$
$LOSS_2 = 6.579$

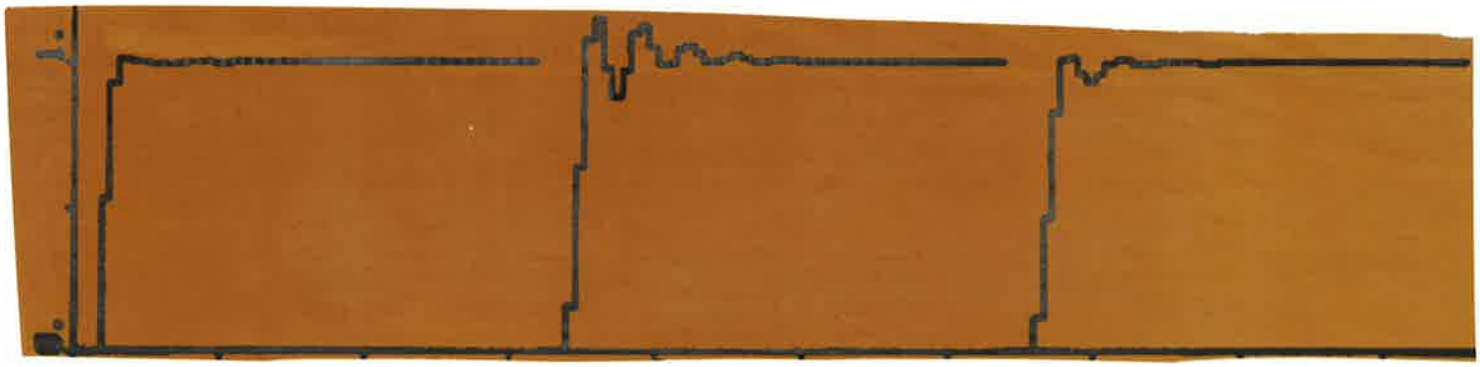


Försök med modellreferens

Här körs en PID-regulator på RTH. Felet e i förlustfunktionen är $e = y^M - y^p$. y^M är utsignalen från en 2.ordningens referensprocess M (se avsnitt 8.1.3 och bilaga 2) med $\zeta = 0.8$ och $\omega_0 = 40$, en snabb process, se 9.4.a nedan.
y^p är utsignalen från den styrda processen. I fig 9.4.b visas y^p då förlustfunktionen ISE använts. Eftersläpningen i starten är obetydlig, sedan fås en del oscillationer.
När referensprocessen är så pass lik ett steg kunde det ju vara intressant att se om man med kriteriet ITAE kunde slippa oscillationerna. Man minns ju fig 9.3.c ovan. Jämför 9.4.c. Starten blir lite långsammare men i övrigt är uppförandet bättre än b.

Fig 9.4 Stegsvar (NOM[FKN]=1)

a. y^M för M(0.8/40)	b. Modellreferens, RTH + PID + ISE.	c. Modellreferens, RTH + PID + ITAE.
	K = 5.038	K = 3.579
	T_i=8.154 T_d=1.007	T_i=9.638 T_d=0.875
	$LOSS_1$= 0.293	$LOSS_2$=4.63



Normalt använder man annars kriteriet ISE vid modellreferens.
Man kan notera att även om felsignalen i förlustfunktionen $e_{FKN} = y^M - y^p$, så har regulatorn bara tillgång till $e_{RG} = y^r - y^p$. y^r är den yttre styrsignalen.

PI-regulator och SEC: modellreferens.

I detta fall vill man få utsignalen från en styrd process SEC av 2.ordningen att efterlikna en långsam referenssignal y^M (ζ=0.9, ω_0=5), se fig 9.5.a nedan. Den styrda processens signal y^p visas i fig 9.5.b. Skillnaderna mot a. är obetydliga. Systemet är något snabbare i början och aningen långsammare i kröken. Stegsvaret för processen SEC ensam visas i 9.5.c.

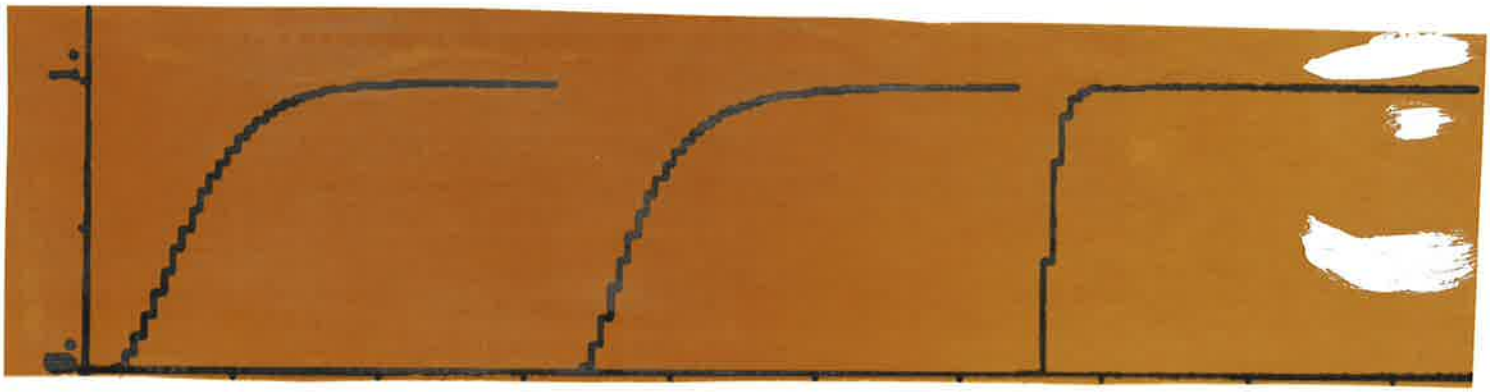


Fig 9.5 Stegsvar

a. y^M för M(0.9/5)

b. SEC + PI + ISE vid modellreferens.
K = 0.0592
T_i = 0.575
$LOSS_1$ = 0.041

c. Processen SEC utan regulator.

Exempel på optimering vid för lågt gradtal för identifieringsrutinen LS, samt återstart.

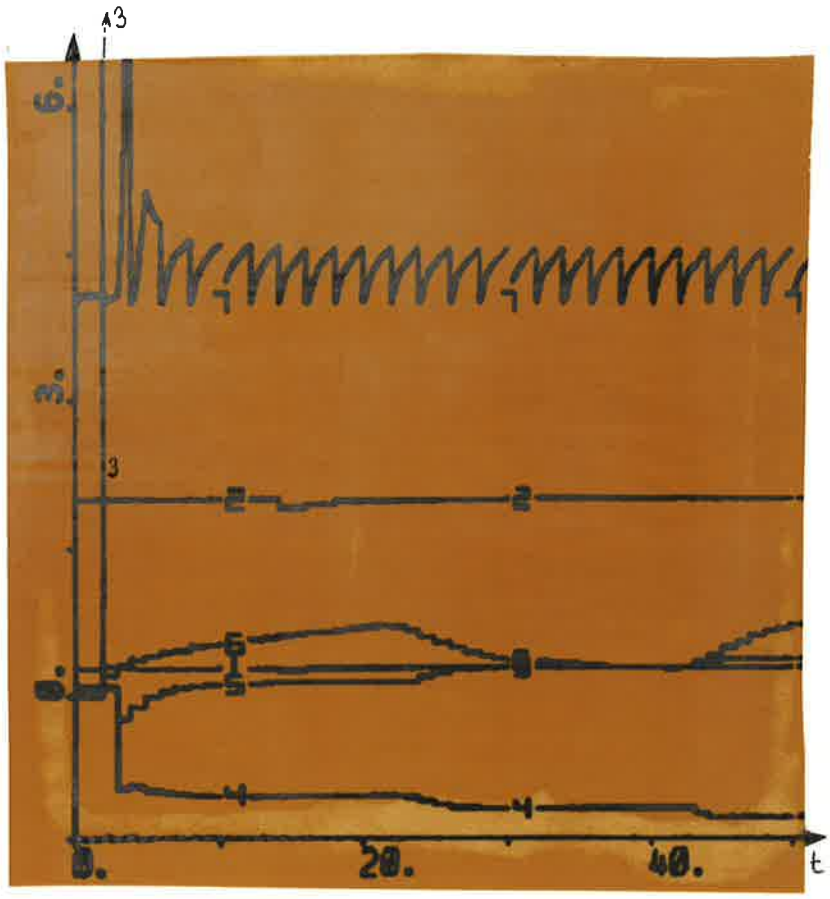
Fig 9.6 visar början av en optimering av en PI-regulator på processen RTH. N[LS] har avsiktligt satts för lågt, 4 i stället för 6. Det är då naturligt att de skattade parametrarna avviker från processens. Två av parametrarna visas i figuren (4, 5). Skattningen förbättras vid varje förändring i processignalerna. Processloopens stegsvar visas i (6) och minloopens i (7). Förlustfunktionen LOSS (ISE) rymms inte inom diagrammet.

Fig 9.6 Första delen av optimeringen av PI + RTH + ISE.

1 - förstärkn K[RGM]
2 - integraltid Ti/10
3 - förlustfkn LOSS
4,5 -två av processkattn: a_1, a_2.
6 - proc utsignal y^P
7 - modellutsignal y^{MOD}.

Parametrar:
SQ=1 NOM=0 LMAX=60
HH=0.01 XM2=5 WT=0.9

Resultat denna del:
LOSS=20.6
K =0.298
T_i=20 (=startvärde!)
TH=-1.29/0.379/0.032/0.066

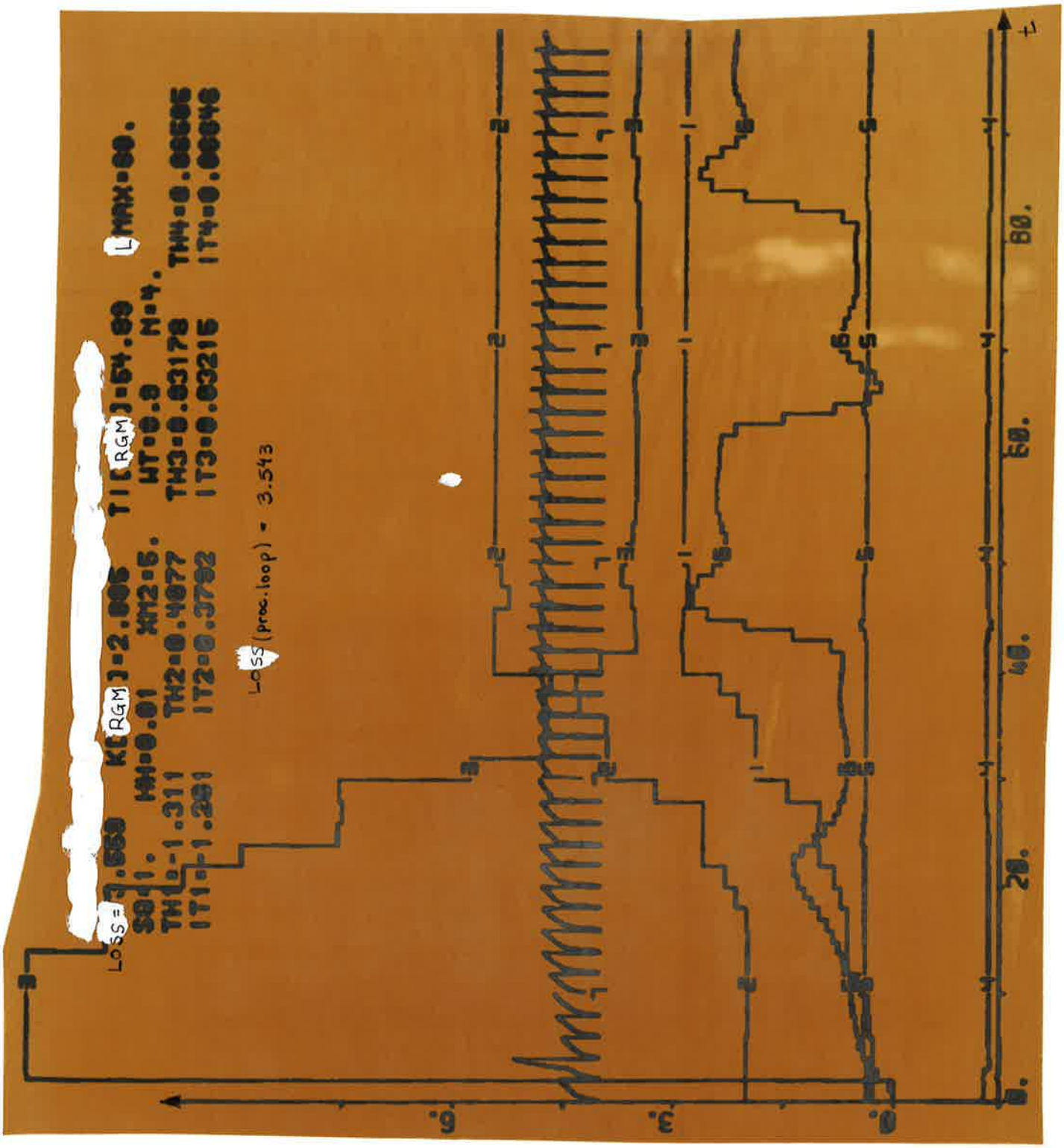


Regulatorparametrarna K, T_i (1, 2) ändras inte mycket varför optimeringen återstartas med uppnådda parametrar och skattningar, se fig 9.7. Nu när OPTA kan starta med en bättre modell uppnås bättre värden, något som syns både på förlustfunktionen (3) och stegsvaret (6). Då referenssignalens PER[REF]=40 är stegsvaret (6) bara 20 samplingspunkter långt.
Ytterligare en återstart hade eventuellt gett en liten förbättring.

I fig 9.7 ser man att slutresultatet blev riktigt bra trots att bilden av processen var ofullständig. Minloopens förlustfunktion blev 3.60 mot 3.54 (!) i processloopen. Identifieringsrutinen lyckades åstadkomma en modell som var bra för just stegändringar, samma signal som förlustfunktionen baserades på. (Stegsvaren ser praktiskt taget lika ut för RTH och den skattade 2.ordningens modellen.) För andra signaler med stora utslag hade resultatet antagligen blivit sämre.

Parameterskattningarna (4, 5) gör de karaktäristiska små hoppen varje gång en stor förändring inträffar för processen, ett tecken på att modellens ordning är för låg.

Fig 9.7 Andra delen av optimeringen PI + RTH + ISE. Kurvors beteckning och parametrar: se fig 9.6. De uppnådda resultaten framgår av figuren.



9.4 NÅGRA PRAKTISKA SYNPUNKTER

Vid optimeringen är det bra att kunna plotta regulatorparametrar i minloopen och förlustfunktionens värde, utsignaler från verklig process och modellen samt i början skattningarna av processparametrarna. På då sätt ser man lätt om någon del av söksystemet "stagnerar".

Går identifieringen dåligt kan man variera på referenssignalen (t ex kortare period eller lägga på brus) till processloopen samt glömskefaktorn WTI. Ser man på förlustfunktionen och provstegsvaret i minloopen att optimeringsrutinen inte finner bra regulatorparametrar kan man återstarta optimeringen. En återstart har fördelen att sökningen börjar i en nyberäknad gradientriktning från den uppnådda punkten.

Under simuleringens början kör man gärna med begränsning av processregulatorns utsignal: ULIM[RG]. Eftersom man är intresserad av att snabbt få processen att köra bra kan man öka ULIM när regulatorparametrarna blivit bra.

Om ett par sökningar med olika startvärden ger olika slutresultat använder man den uppsättning som ger lägst förlustfunktion såvida inte utsignalen i minloopen har visat sig orimlig (jfr anm. 1 i avsnitt 6.1).
Som slutkontroll kan förlustfunktionens värden i processloopen och minloopen jämföras för lika insignaler.

Det går inte att ge allmänna regler för vad som är en godtagbart låg förlustfunktion. Ofta är det svårt att ge en enkel tolkning av denna så att man kan sätta absoluta gränser. Värdet beror naturligtvis starkt av process och -brus. Då provtiden i minloopen är begränsad måste man kolla att provstegsvaret hunnit konvergera någorlunda för att förlustfunktionen ska säga något intressant.

Förlustfunktionen ISE = Σe_t^2, e = $y^r - y^p$ har vid simuleringarna givit lägre värden än ITAE = $\Sigma t|e_t|$ då referenssignalen varit ett steg. Typiska värden för 2. och 3.ordningens processerna SEC och RTH har varit 1.5-2.5 resp 3-7 med PI- och PID-regulatorer.
Då y^r var utsignalen från en referensprocess med rimlig snabbhet i förhållande till den verkliga processen har ISE ofta givit värden <0.1, medan ITAE givit betydligt större värden.

Återstart

I avsnitt 8.2.2 ges exempel på startparametrar vid en optimering. Då man återstartar en simulering i Simnon sätts normalt alla startvärden in igen, dvs allt börjar om från början. Man måste alltså ändra startvärdena för att kunna utnyttja hittills uppnådda resultat.
Då man vill återstarta en optimering för att förbättra eller kontrollera resultatet ska man:
a) sätta startvärdena PI1,... i OPTA till nuvarande parametervärden PD1,... Dessa motsvarar aktuell LOSS,
b) sätta startvärdena IT1,... i identifieringsrutinen LS till nuvarande skattning TH1,... c) samtidigt kan man ändra steglängd HH och ev skalfaktorer XM_i i OPTA samt initialvärden L1,... i LATCH till PD1,....

I bilaga 4 ges ett exempel på hur en optimering i Simnon startas upp.

10. SLUTSATSER OCH UTVECKLINGSRIKTNINGAR

"Cogito, ergo sum."
DESCARTES

10.1 SAMMANFATTNING OCH RESULTAT

Det beskrivna söksystemet är en mycket generell struktur för att optimera regulatorparametrar. Man är på inget sätt bunden till just PID-regulatorn som har valts här. T ex kan man prova olika regulatortyper på en process och se vilken som ger bäst resultat. Processloopen och minloopen (där optimeringen sker) har separata styrsignalingångar varför man kan välja signaler som är lämpligast för resp loop. För processloopen är det signaler som ger god möjlighet att identifiera processen. När identifieringen är klar kan processloopen återgå till helt normal körning medan optimeringen fortsätter i minloopen. Minloopens styrsignal beror av den valda förlustfunktionen.

Söksystemet kan även behandla tidsfördröjning i processen. Då sättes fördröjningen i antal samplingsintervall L i identifieringsrutinen, och ett block med lika många tidsfördröjningar införes just efter modellen i minloopen. Om fördröjningen varierar med någon annan variabel kan man införa en separat rutin som beräknar L.

Vid simuleringarna har inget brus använts. Brus är inte farligt annat än på så sätt att det kan störa identifieringsrutinen i vissa fall som nämns i avsnitt 5.2. Naturligtvis kommer även "felmåttet" V i rutinen att öka pga bruset (se även N, avsnitt 8.2.3).

Slutresultatet vid en optimering beror av många faktorer. Här kan nämnas vald förlustfunktion, optimeringsrutinens effektivitet, processens styr- och observerbarhetsegenskaper. Optimeringen omfattar endast en processloop åt gången. Man kan kanske tänka sig att klara enklare kaskadkopplade loopar genom att låta minloopens regulator innefatta båda de yttre regulatorerna.

Tack vare de säkerhetsmekanismer som finns inbyggda i söksystemet kan man ge sig på en process nästan utan att veta något om den utan risker. Man behöver veta processens utstyrningsområde och ungefärliga snabbhet för att kunna sätta ULIM i processregulatorn. För att kunna ansätta regulatorstruktur (t ex PI) behöver man ha sett t ex ett stegsvar för processen. Kan man därav också gissa ungefärligt ordningstal spar man tid genom att få identifieringen "ur vägen" snabbare.

Simuleringar på 2. och 3.ordningens processer har visat att det oftast räcker med mindre än 200 provstegsvar i minloopen för att hitta bra regulatorparametrar, förutsatt att identifieringen är någorlunda effektiv. Optimeringsrutinens startvärden spelar liten roll. Mjukast förlopp får man oftast genom att ansätta försiktiga regulatorparametrar som startvärden, jfr avsnitt 8.2.3.

För standardfallen tycks förlustfunktionen ITAE = $\Sigma t|e_t|$ vara bäst.
För optimeringsresultat hänvisar jag i övrigt till avsnitt 9.2, 9.3 och 6.2, särskilt ITAE.

Försöken med modellreferens var speciellt lovande. Då låter man skillnaden mellan utsignaler från ett känt önskvärt system och minloopens styrda modell vara felsignal till förlustfunktionen. Om bara referensprocessen är någorlunda rimlig får man mycket god likhet i beteende och god konvergens med förlustfunktionen ISE = Σe_t^2 (som försöker efterlikna beteendet i stort). I stället för att byta mellan olika förlustfunktioner kan man alltså byta mellan olika referensprocesser. Den tidigare varianten med ett steg direkt till förlustfunktionens referensingång kan ses som ett specialfall av modellreferens (systemet = 1, ingång→utgång). För en annan tolkning, se signalreferens i nästa avsnitt.

10.2 MODIFIERINGAR OCH UTVECKLINGSRIKTNINGAR

1)

Till existerande system vill jag föreslå att man studerar området förlustfunktioner ytterligare för att använda optimeringen för specifika mål såsom energibesparing, minimering av spill osv.

För varje använd regulatortyp för en viss process kan välja mellan att anpassa skalfaktorer i optimeringsrutinen eller parametrisera om regulatorn. Kanske visar det sig att en viss parametrisering av PID-regulatorn konvergerar bättre i genomsnitt trots att processerna varierar.

2)

Det finns alternativ till att beräkna förlustfunktionen ISE. Man kan visa att $\sum_0^\infty e_t^2$ kan omformas till

$\frac{1}{2\pi i}\oint |E(z)|^2 \frac{dz}{z}$, E polynomiell, över enhetscirkeln $z = \exp(i\omega)$, om processen konvergerar. I tillämpningen utvärderar man ISE för ett ändligt antal steg, men skillnaden blir försumbar om man avbryter efter så många steg att modellen nått konstans.

Integralen kan skrivas $\frac{1}{2\pi i}\oint \frac{B(z)B(z^{-1})}{A(z)A(z^{-1})} \frac{dz}{z}$ som existerar utom då A(z) har nollställen på enhetscirkeln och kan i sin tur beräknas snabbt i numeriska rutiner av liten längd.

Om man alltså ur processkattningar, regulatorstruktur och -parametrar samt insignal i minloopen kan beräkna polynomen A och B kan beräkningen av ISE troligen göras snabbare på detta sätt och ersätta minloopen.

3) Signalreferens

Om den styrda processen i drift kommer att styras av en ungefärligt känd periodisk referenssignal kan man optimera regulatorn just med tanke på denna.
Då låter man felsignalen i förlustfunktionen vara skillnaden mellan styrsignalen och modellens utsignal (i minloopen). För varje optimeringssteg upprepas signalsekvensen till minloopen.
Genom att lägga till en tidsberoende faktor (viktfaktor) i förlustfunktionen kan signalföljsamheten för processen ökas markant i vissa kritiska punkter.
I ett sådant fall kan man alltså använda en standardregulator och uppnå ganska specifika resultat.
Ett konstruerat exempel visas nedan.

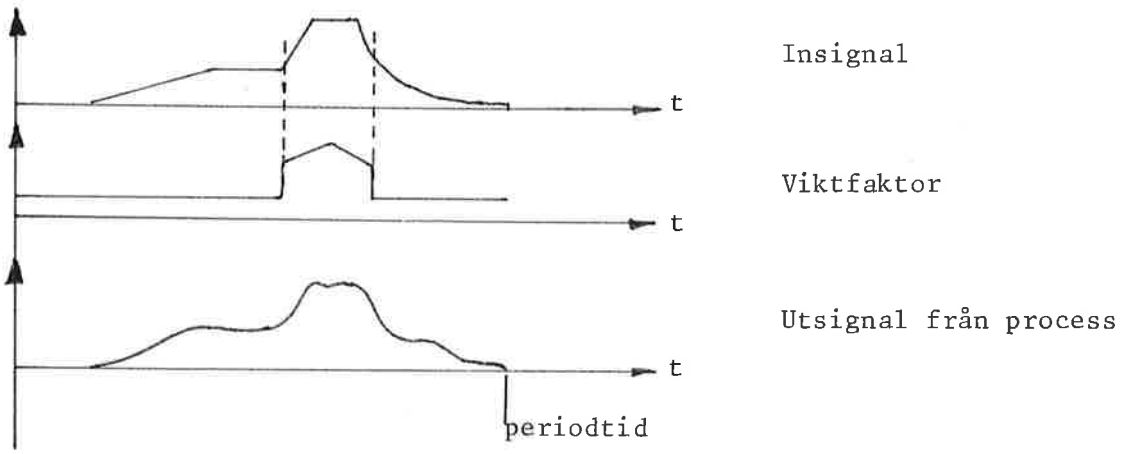


Fig 10.1 Påhittat exempel på signalreferens

4)
Det ligger sas om hörnet att göra detta söksystem till en adaptiv regulator. Men för att bli en effektiv sådan fordras troligen att nuvarande exekveringstid bantas ner eftersom processdatorn ska hinna exekvera många regulatorer på kort tid.

5) Styrspråk

Vid en realisering av ett söksystem som beskrivits här har man nytta av ett interaktivt, operatörsvänligt styrspråk. Några funktioner hos ett sådant styrspråk skulle kunna vara:

- koppla in och ur optimering av en viss reglerkrets
- dirigera optimeringsrutinen
- ändra värden under körning
- "trimma in" och prova olika förlustfunktioner
- koppla om styrsignaler i söksystemet under drift.

11. EFTERSKRIFT 72

12. BILAGOR

Bilaga 1: programlista för identifierings-rutinen

Bilaga 2: programlistor för använda processer (SEC, RTH, M)

Bilaga 3: byte av provtid i minloopen

Bilaga 4: uppstart av en optimering i Simnon

13. REFERENSER

EFTERSKRIFT

Denna rapport är resultatet av ett examensarbete vid institutionen för reglerteknik vid Lunds tekniska högskola.
Många anser att ett examensarbete bör ses som en tillämpningsuppgift och förstudie till ett undersökningsområde eftersom tiden är begränsad.

En av lärdomarna från arbetet vill jag speciellt framhålla. Det är mycket tidsbesparande att testa varje block eller enhet pedantiskt innan det "bygges ihop" med tidigare block. Lika noggrann provning och kontroll måste göras så snart blocket anslutits innan andra delar adderas.

Jag vill framföra mitt tack till Björn Wittenmark vid institutionen som handlett arbetet och föreslagit strukturen. Jag vill även rikta ett varmt tack till Jan Sternby vid institutionen som bidragit med många intressanta diskussioner och råd vid utprovningen. Likaså har Ivar Gustavsson, Tomas Schöntal och Tommy Essebo vid institutionen gett råd vid datoranvändningen. Därtill kommer att Claes Ahlerup, Bengt Hansson och Nils Johansson vid SATT-Elektronlund AB samt Östen Nilsson vid Hugo Tillqvist AB givit värdefulla synpunkter.

Min lättnad överträffas nog bara av läsarens, när jag nu förklarar denna rapport avslutad.

Lund Ulrikdagen 1979

Poul Kongstad

```
      SUBROUTINE LS
C     THIS SUBROUTINE ESTIMATES THE COEFFICIENTS A(I), B(I) IN A
C     MODEL A(Q-1)Y(T) = Q↑(-L)*B(Q-1)U(T) + E(T) BY THE METHOD OF
C     RECURSIVE LEAST SQUARES UNDER SIMNON.
C     REF: SÖDERSTRÖM: KOMPENDIUM I PROC.IDENTIFIERING  UPPSALA 1974.
C     AUTHOR:  POUL KONGSTAD
C
C     PARAMETERS:
C     N     THE NUMBER OF ESTIMATED PARAMETERS = TWICE THE
C            ORDER OF PROCESS, MAX 10, SEE DECL BELOW (DEFAULT VALUE: 0)
C     L     THE NUMBER OF EXTRA DELAYS, MAX 15. (DV: 0)
C     DT    SAMPLING TIME (DV: 1).
C     PP    INITIAL VALUE FOR P-MATRIX DIAG.ELEMENTS (DV: 100).
C     IT(10)   VECTOR WITH INITIAL VALUES FOR TH (DV: ALL 0.1).
C     WTI   INITIAL VALUE FOR DATA FORGETTING FACTOR (DV: 0.9).
C     WTM   FORGETTING PROFILE  (DV: 0.01)
C
C     INPUTS:
C     U1    PROCESS OUTPUT Y(T)
C     U2    PROCESS INPUT  U(T)
C     OUTPUT:
C     TH    VECTOR WITH ESTIMATED PARAMETERS A1,...,B1,...
C          (DV:  SEE ABOVE: IT)
C
C     VARIABLES OF INTEREST:
C     FI(10)    OLD PROCESS VALUES (-Y(T-1),...,U(T-1),..,U(T-5) )
C     UL(15)    DELAYED PROCESS INPUT VALUES
C     RIT       NUMBER OF ITERATIONS SINCE START
C     NOR       ROOT MEAN SQUARE OF K(N+1)*(Y(N+1)-TH*FI), THE LAST
C               CORRECTION TO PARAMETER ESTIMATION VECTOR TH
C     V         WEIGHTED LOSS FUNCTION, V(T)=0.9*V(T-1)+(NOR)↑2.
C
C     SUBROUTINES REQ.D:  NONE.
C
      REAL K(10), IT(10), KT(4)
      DIMENSION AV(10),P(10,10),TH(10),FI(10),UL(15),PT(16)
      COMMON /DESTIN/IDUM,IPART
      COMMON /TIME/T
      COMMON PP,DT,IT,RNORM,WTI,WTM,WT,V,USAVE
C
      GO TO (1,2,3,4,5,6,7,8), IPART
C     IDENTIFICATION
1     CALL IDENT(5HDISCR, 2HLS)
      RETURN
C
C     DECLARATIONS OF VARIABLES
2     CALL INPUT(U1,2HU1)
      CALL INPUT(U2,2HU2)
      CALL OUTPUV(TH,10,2HTH)
      CALL VAR(RITER,3HRIT)
      CALL VARV(FI,10,2HFI)
      CALL VARV(UL,15,2HUL)
      CALL VAR(RNORM,3HNOR)
      CALL VAR(V,1HV)
      CALL PAR(RL,1HL)
      CALL PAR(RN,1HN)
      CALL VAR(WT,2HWT)
      CALL PAR(WTM,3HWTM)
      CALL PAR(WTI,3HWTI)
      CALL PAR (DT,2HDT)
      CALL PAR(PP,2HPP)
      CALL PARV(IT,10,2HIT)
      CALL TSAMP(TS,2HTS)
```

```
064             RETURN
065       C
066       C     ASSIGNMENT OF CONSTANTS (PAR, INIT);  SAVED IN COMMON BLOCK.
067       3     PP=100.
068             DT=1.
069             WTM=0.01
070             WTI=0.9
071             DO 300 I=1,10
072       300   IT(I)=0.1
073             RETURN
074       C
075       C     INITIAL SECTION
076       4     N = INT(RN+.01)
077             L = INT(RL+.01)
078             WT=WTI
079             DO 401 I=1,N
080             TH(I)=IT(I)
081             FI(I)=0.
082             DO 400 J=1,N
083       400   P(I,J)=0.
084       401   P(I,I)=PP
085             DO 402 I=1,L
086       402   UL(I)=0.
087             RITER=0.
088             USAVE=0.
089             YSAVE=0.
090             RETURN
091       C
092       C     OUTPUT SECTION
093       C     UPDATE FI
094       5     I=INT(RN/2.+.01)
095             J=I+I
096             IF (N.LT.4) GO TO 505
097       500   FI(I)=FI(I-1)
098             FI(J)=FI(J-1)
099             I=I-1
100             J=J-1
101             IF(I.GT.1) GO TO 500
102       505   FI(1)=-YSAVE
103       C
104       C     UPDATE UL VECTOR WHEN L>0
105             IF (L.EQ.0) GO TO 503
106             FI(J)=UL(L)
107             IF(L.EQ.1) GO TO 502
108             I=L
109       501   UL(I)=UL(I-1)
110             I=I-1
111             IF(I.GT.1) GO TO 501
112       502   UL(1)=USAVE
113             GO TO 504
114       503   FI(J)=USAVE
115       504   YSAVE=U1
116       C
117       C     K VECTOR
118             DO 510 I=1,N
119             SLASK=0.
120             DO 511 J=1,N
121       511   SLASK=SLASK + P(I,J)*FI(J)
122             K(I)=SLASK
123       510   AV(I)=SLASK
124       C
125             SLASK=1.
126             DO 520 I=1,N
127       520   SLASK=SLASK + AV(I)*FI(I)
```

```
      DO 521 I=1,N
521   K(I)=K(I)/SLASK
C
C     P MATRIX
      DO 522 I=1,N
      DO 522 J=1,I
      P(I,J) = ( P(I,J) - K(I)*AV(J))/WT
522   P(J,I)=P(I,J)
C
C     ESTIMATION AND PREDICTION ERROR
      RNORM=0.
      SLASK=YSAVE
      DO 530  I=1,N
530   SLASK=SLASK-TH(I)*FI(I)
      DO 531 I=1,N
      DEL = K(I)*SLASK
      RNORM=RNORM+DEL*DEL
531   TH(I) = TH(I)+DEL
      RITER=RITER+1.
      RETURN
C
C     DYNAMICS SECTION
6     TS = T + DT
      WT=(1.-WTM)*WT + WTM
      USAVE=U2
      RETURN
C
C     COMP. OF ACCEPTED VALUES.
7     V=.9*V + RNORM
      RNORM=SQRT(RNORM)
      RETURN
C
C     FINAL COMP
8     RETURN
      END
```

```
DISCRETE SYSTEM SEC
"SAMPLING AV 1/((1+.5S)(1+S)) MED DT=1.
INPUT  U
OUTPUT Y
STATE  X1 X2
NEW   NX1 NX2
TIME   T
TSAMP  TS

OUTPUT
Y = B1*X1 + B2*X2

DYNAMICS
NX1= -A1*X1 - A2*X2 + U
NX2= X1
TS= T + DT

A1: -0.503215
A2:  0.049787
B1:  0.399577
B2:  0.146996
DT:  1
END
```

- -

```
DISCRETE SYSTEM RTH
"SAMPLING AV 1/((1+.5S)(1+S)(1+5S)) MED DT=1.
INPUT  U
OUTPUT Y
STATE  X1 X2 X3
NEW   NX1 NX2 NX3
TIME   T
TSAMP  TS

OUTPUT
Y = B1*X1 + B2*X2 + B3*X3

DYNAMICS
NX1= -A1*X1 - A2*X2 - A3*X3 + U
NX2= X1
NX3= X2
TS= T + DT

A1: -1.32195
A2:  0.46178
A3: -0.040762
B1:  0.031776
B2:  0.0608495
B3:  0.00645093
DT:  1
END
```

```
DISCRETE SYSTEM M
"DISKRET MODELL AV REFERENSPROCESS. LOSNINGSTID <1.5 DT, OVERSLANG 15%
"Z FAR EJ VARA 1 ELLER NEGATIV.
INPUT U
OUTPUT Y
TIME   T
STATE  S1  S2
NEW   NS1 NS2
TSAMP  TS

INITIAL
OM=ONOLL*SQRT(ABS(1-Z*Z))
A=EXP(-Z*ONOLL*DT)
B=IF Z>1 THEN .5*(EXP(OM*DT)+EXP(-OM*DT)) ELSE COS (OM*DT)
D=Z*ONOLL/OM*(IF Z>1 THEN .5*(EXP(OM*DT)-EXP(-OM*DT)) ELSE SIN(OM*DT))
B1=1-A*(B+D)
B2=A*(A+D-B)
A1=-2*A*B
A2=A*A

OUTPUT
Y=S1

DYNAMICS
NS1=-A1*S1 + S2 + B1*U
NS2=-A2*S1 + B2*U
TS= T + DT

Z:     .2672
ONOLL: 10

DT:    1

END
```

BYTE AV PROVTID I MINLOOPEN

Formel (4.1) ger sambandet mellan de parametrar som avgör provstegsvarets längd i minloopen.
Då denna ändras ska följande variabler ändras (x):

System	Parameter	PROVL	UPER	DTM	DTN	TINC
RGM		x		x		
MOD		x		x		
FKN		x		x		
M				x		
KON		x				
LATCH			x			
NOLL					x	
OPTA						x

UPPSTART PÅ PDP-15 (BILAGA 4)

Antag att vi på PDP-15 datorn har tillgång till ett Simnonpaket innehållande LS samt på skivminnet våra delsystem och parameterfilen PAR (se tabell 8.1). Som process önskar vi använda RTH och vi vill köra exemplet i fig 9.4.c, dvs modellreferens på M(0.8/40) med förlustfunktion ITAE för en PID-regulator. Kommandon:

```
$ A RK 3,4/NON 5,7,15,16                operativsystemkommandon
$ BUFFS 6                               för uppstart
$ E SIMNON

> LET NPAR.OPTA=3                       Här startar Simnon
> LET NCONS.OPTA=0                      Globala variabler (OPTA, PID-reg.)
> SYST REF RG RTH LS RGM MOD M FKN NOLL LATCH OPTA KON
                                        Def av systemet.
                                        Utskrift som varnar för ev
                                        algebraiska loopar; kollade.
> GET PAR                               Startvärden från parameter-
                                        filen laddas.
> PAR PER:40                            Ändra enstaka parametrar:
> PAR WTM:.001                                god identifiering
> PAR NOM: 1                            Modellreferens
> PAR SQ: 0                             ITAE
> INIT PI1: 4                           Andra startvärden för OPTA
> INIT PI2: 20
> INIT PI3: .1
> PLOT K[RGM] LOSS TH1 TH2 Y[RTH] Y[MOD]  Uppritning av variabler under sim.
> AXES v -1  6  H 0 150                 Definition av axlar
> SIMU  0 150 - MARK                    Simuleringen startas.

> AXES H 150 300                        Nya axlar ritas
> SIMU - CONT MARK                      Sim fortsättes (150-300).
> DISP LOSS PD1 PD2 PD3                 Utskrift av LOSS och reg.par.
> STOP                                  Uthopp ur Simnon
```

At hvide, hvad man ikke véd,
er dog en slags Alvidenhed.

KUMBEL KUMBELL

REFERENSER

Davidon — Variable metric method for minimazation, AEC Research & Development Report ANL5990, 1959.

Davies — A self-optimizing PI-controller, Control, nov 1966, pp 594-598.

Elvgren,Krantz — Identifiering av återkopplade system, examensarbete vid reglerteknik, LTH, Rapport TRFT-5142.

Elmqvist — Simnon, an interactive simulation program for nonlinear systems, user's manual. Report TRFT-3091, Dept of Autom Controll, LTH, Lund, 1975.

Fletcher — Fortran subroutines for minimazation by quasi-Newton methods, Harwell report AERE-R7125, 1972.

Fletcher, Powell — A rapidly convergent descent method for minimazation, Computer Journal vol 6, pp 163-168, 1963.

Fletcher, Reeves — Function minimazation by conjugate gradients, Computer Journal vol 7, pp 149-154, 1964.

Glad — A program for the interactive solution of parametric optimization problems in dynamic systems, report TRFT-3084, Dept of Autom Control, LTH, Lund 1974.

Frank — Beitrag zum Technischen Aufbau adaptiver Regler, Universität Karlsruhe, BRD, inst für Elektro- und Regelungstechnik.

Goodwin et al — Design of restricted complexity adaptive regulators, IEEE Transactions on automatic controll, August 1978.

Harriott — Process Control, New York 1964.

Leonhard — Regelkreise mit symmetrischer übertragungsfunktion, Regelungstechnik 13 s 4-12, 1965.

Lopez et al — A comparison of controller tuning techniques, Control Engineering, dec 1967, pp 72-75.

Mills — An adaptive controller with synchronus perturbations, Elec.Engineering Transactions, March 1965, pp 23-26.

Ogata — Modern Control Engineering, New Jersey 1970.

Pendlebury et al Digital control of a process using on-line auto-tuning. Automation prp2, paper 3.3, Kodak, Middlesex, United Kingdom.

Söderström Kompendium i processidentifiering, Teknikum, Uppsala universitet, 1974.

Wittenmark Self-tuning PID-controller, based on pole-placement, Dept of Autom Control, LTH, Lund 1979 (to appear).

Åström Introduction to stochastic control, Academic Press, New York 1970.

Åström et al Theory and applications of self-tuning regulators, Automatica vol 13, 1977, pp 457-476.

Åström et al Self-tuning controllers based on pole-placement design, report TRFT-3148, may 1978, Dept of Autom Control, LTH, Lund.

Åström Reglerteori, Uppsala 1976.