# Projekt i Adaptiv Reglering

Åström, Karl Johan

1990

*Document Version:*
Förlagets slutgiltiga version

Link to publication

*Citation for published version (APA):*
Åström, K. J. (Red.) (1990). *Projekt i Adaptiv Reglering*. (Technical Reports TFRT-7444). Department of Automatic Control, Lund Institute of Technology (LTH).

*Total number of authors:*
1



LUND UNIVERSITY

PO Box 117
221 00 Lund
+46 46-222 00 00

**Department of Automatic Control**
**Lund Institute of Technology**
P.O. Box 118
S-221 00 Lund Sweden

*Document name*
Report

*Date of issue*
January 1990

*Document Number*
CODEN: LUTFD2/(TFRT-7444)/1–315/(1990)

*Supervisor*

*Sponsoring organisation*

*Author(s)*
Karl Johan Åström

*Title and subtitle*
Projekt i Adaptiv Reglering (Project in Adaptive Control)

*Abstract*

This report is a collection of project reports in the course Adaptive Control during 1987/88 and 1988/89. Each student in the course makes a one week project. The projects range from theoretical to practical projects. Some projects give a discussion of the applicability of adaptive control to different processes.

*Key words*

*Classification system and/or index terms (if any)*

*Supplementary bibliographical information*

*ISSN and key title*

*ISBN*

*Language*
Swedish, English

*Number of pages*
315

*Security classification*

*Recipient's notes*

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

2.



# Innehåll

# 1. Inledning

Denna rapport innehåller de projektrapporter som utförts i kursen Adaptiv Reglering under läsåren 1987/88 och 1988/89. Idéen med att sammanställa materialet är dels att ha ett material att visa teknologerna och dels att visa vad som kan åstadkommas på ett kort projekt. Målet är givetvis att projekten skall bli bättre och bättre. Hittills har projekten och rapportering gjorts i fritt format. Vi finner således alla varianter från handskrivna rapporter till vackert ordbehandlade sådana.

Kvaliteten på innehållet varierar också kraftigt. I fortsättningen är det nog bra att ställa formella krav åtminstone på rapporternas uppläggning. Under läsåret 1988/89 gjorde vi också experimentet att kombinera projektuppgifter i kursen Adaptiv Reglering med kurserna i Datorimplementering av Reglersystem och Tillämpad Artificiell Intelligens. Detta visade sig vara mycket lyckosamt. Bra exempel är t.ex. projekten Automatinställning med parameter styrning, Auto-tuners för bomprocess och PID-Expert.

Rapporten är organiserad på följande sätt. Anvisningar för projekten ges i avsnitt 2. Därefter följer de olika projekten organiserade på följande sätt. Teoriorienterade projekt finns i avsnitt 3. Avsnitt 4 behandlar speciella tillämpningar av adaptiva system. Där finns adaptiva auto-piloter för båtar, adaptiva system för bilmotorer och diverse servon. Avsnitt 5 behandlar experiment med kommersiella adaptiva system. Avsnitt 6 innehåller olika experiment med automatinställning och avsnitt 7 presenterar undersökning av en design utmaning från American Control Conference 1988 och 1989.

# 2. Projektuppgifter i Adaptiv Reglering

Här är förslag till några projektuppgifter. Uppgifterna skall normalt genomföras i grupper. Två till fem personer är en bra gruppstorlek. Ideen är att Du skall arbeta med ett större problem på egen hand. Du får gärna ta hjälp genom att fråga lärare och kamrater. Jourtider då Du kan få hjälp kommer att anslås senare. Uppgiften skall gå att klara av på en veckas heltidsarbete inklusive dokumentation. Den skall redovisas i en kort men välskriven rapport. Uppgifterna är av varierande svårighetsgrad. Flera kan utvidgas till examensarbeten.

### Teoriuppgifter

Det finns många olösta problem med adaptiva regulatorer. För den som är teoretiskt intresserad finns det mycket att ta i tu med. Dessa diskuteras på individuell basis. Exempel: stabilitetsanalys av parameterstyrning, bevisa stabilitet av MRAS för låga förstärkningar.

### Litteraturstudier

Läs någon eller några centrala artiklar. Kontrollera resultat med räkning och simulering. Redovisa Dina resultat. Exempel: stabilitet för stokastiska självinställare, medelvärdesteori, betydelsen av exitation i adaptiva system, Parks och Monopoli m.fl. Åström och Wittenmark (1973), Ljungs doktorsavhandling, Källströms avhandling.

4.



## Speciella tillämpningar

Dessa uppgifter består i att undersöka en speciell tillämpning. I uppgiften ingår att göra en liten litteraturstudie, att läsa in problemet och att simulera. Exempel på tillämpningar är adaptiva autopiloter för båtstyrning, adaptiv robotstyrning, adaptiv reglering av verktygsmaskiner eller adaptiv reglering av bilmotorer.

## Produktstudier

Välj ut en kommersiell adaptiv regulator. Lär Dig hur den fungerar. Tänk igenom vad dess styrka och svagheter kan vara. Föreslå några experiment som kan belysa dess goda och dåliga sidor. Prova den på några processer i labbet, t.ex. simulerade processer på analogimaskin, bommen, fläkten, servon, Schmitbauer-processen. En bra början kan vara att upprepa Laboration 1 och 3 med den kommersiella hårdvaran. Genomför experimenten. Redovisa Dina resultat. Du kan välja mellan Asea Novatune och First Control SattControl ECA 400. Trevligt vore om det kunde bli fyra grupper. En på vardera av Novatune, First Control, ECA 400 och en fjärde grupp som jämför resultaten.

## Produktarkelogi

Flera av tillverkarna talar ej exakt om vad dom gör. Försök att tänka ut experiment som kan utföras för att man skall kunna komma på hur systemen fungerar. Metoden kan kallas *reverse engineering* eller identifiering av regulatorer.

## Special 1

Implementera automatinställning med relämetoden på Novatune eller First Control. Använd den sedan för att automatiskt välja samplingsintervall och önskad pol hos det slutna systemet.

## Special 2

Tänk ut hur man skall göra en automatisk övervakning av en självinställare baserad på polplacering. Prova resultatet på någon labprocess.

## Special 3

Prova med a) återkoppling med hög förstärkning b) MRAS och c) STR baserad på polplacering på det andra ordningens system som delats ut. Genomför simulering med Simnon för att visa vad som händer.

## Egna ideer

Bäst av allt. Om Du har några egna ideer som Du vill prova så hjälper vi gärna till med att försöka göra en projektupgift av dem.

**Lycka till med val och genomförande**

Karl Johan Åström

## 3. Teoriuppgifter

6.

# Stabilitet för System med Parameterstyrning
# Projekt i Adaptiv Reglering

Anders Persson

27 mars 1989

7.

# Innehållsförteckning

8.

# 1 Inledning

Detta projekt i adaptiv reglering handlar om parameterstyrning (*gain scheduling*) och stabilitetsundersökning av parameterstyrda regulatorer. Projektet baserar sig i stora delar på en doktorsavhandling av J.S. Shamma vid MIT [3]. Delar av avhandlingen finns publicerade i två artiklar i *IEEE Conference on Decision and Control*, [4] och [5]. Projektet ingår som ett delmoment i kursen Adaptiv Reglering vid LTH. Rapporten är organiserad på följande sätt: Kapitel 2 handlar om parameterstyrning i allmänhet, kapitel 3 om Shammas resultat, kapitel 4 innehåller en del kommentarer till Shammas avhandling, kapitel 5 är en avslutande sammanfattning och sist finns referenser angivna.

# 2 Kort om parameterstyrning

Parameterstyrning är en teknik som används då dynamiken hos det system som skall regleras ändras på ett känt sätt. Tekniken kan användas både om systemet är tidsberoende, olineärt samt om systemets dynamik beror på någon yttre mätbar parameter. Idén är att välja ett antal arbetspunkter som täcker hela systemets arbetsområde, linearisera systemet kring dessa punkter och därefter dimensionera regulatorer till de olika lineariserade systemen. Sedan växlar man mellan regulatorerna då systemet passerar vissa i förväg utsatta gränser eller interpolerar mellan dem. Metoden fungerar bra i praktiska tillämplingar t.ex. flygplan och båtar, där dynamiken varierar både med hastigheten och höjden/djupet. Dock finns ingen teoretisk analys som garanterar stabilitet hos det slutna parameterstyrda systemet.

# 3 Shammas resultat

## 3.1 Inledning

I detta kapitel redogöres för Shammas resultat, vilka garanterar stabilitet för parameterstyrda system. Shamma har i princip behandlat tre olika fall:

- Lineära system med tidsberoende parameter (t. ex flygplan)
- Olineära system med referenssignalen som parameterstyrande variabel
- Olineära system med utsignalen som parameterstyrande variabel

Det är framför allt det första fallet jag kommer att kommentera i projektet, ty de båda andra kräver en hel del matematiska kunskaper i bl.a. Volterra integrodifferential ekvationer (*Volterra Integrodifferential Equations, VIDE*).

Kapitlet börjar med en del grundläggande definitioner och matematiska resultat vilka används senare. Därefter behandlas parameterstyrning av lineära system med tidsberoende parameter.

9.

## 3.2 Grundläggande definitioner och satser

Betrakta följande almänna system av differentialekvationer:

$$\begin{cases} \dot{x} = f(x(t), t) \\ x(t_0) = x_0 \in \mathcal{R}^n \\ t_0 \in \mathcal{R}^+ \end{cases} \tag{1}$$

Vi antar att följande förutsättningar gäller:

1. $\mathbf{f}(\mathbf{x}, \mathbf{t}) = \mathbf{0} \quad \forall t \geq 0$.
2. $\mathbf{f}$ är sådan att det finns en entydig lösning i intervallet $[t_0, \infty]$ samt att lösningen beror kontinuerligt på initialvilkoren $x_0, t_0$.

Beteckna den entydiga lösningen till (1) med $s(t; x_0, t_0)$, där beroendet av initialvillkoren angetts explicit.

Nu är vi klara att definiera lokal exponentiell stabilitet:

**Definition 1:** Det olineära systemet (1) under förutsättningarna 1 och 2 sägs vara **lokalt exponentiellt stabilt** om det finns konstanter $m, \lambda$ och $\epsilon > 0$ så att:

$$|s(t; x_0, t_0)| \leq m e^{\lambda(t-t_0)} |x_0|, \quad t \geq t_0, \quad \forall x_0 \in \mathcal{B}(x_0; \epsilon) \tag{2}$$

där $\mathcal{B}(x_0; \epsilon)$ är det öppna klotet i $\mathcal{R}^n$ med centrum i $x_0$ och radie $\epsilon$ d.v.s:

$$\mathcal{B}(x_0, \epsilon) \equiv \{x \in \mathcal{R}^n \mid |x - x_0| < \epsilon\}$$

Systemet sägs vara **lokalt likformigt exponentiellt stabilt** om konstanterna $m, \lambda$ och $\epsilon$ är oberoende av $t_0$:

$$|s(t; x_0, t_0)| \leq m e^{\lambda(t-t_0)} |x_0|, \quad t \geq t_0, \quad \forall x_0 \in \mathcal{B}(x_0; \epsilon), \ \forall t_0 \in \mathcal{R}^+ \tag{3}$$

Vi definierar även global exponentiell stabilitet:

**Definition 2:** Det olineära systemet (1) under förutsättningarna 1 och 2 sägs vara **globalt exponentiellt stabilt** om det finns konstanter $m$ och $\lambda > 0$ så att:

$$|s(t; x_0, t_0)| \leq m e^{\lambda(t-t_0)} |x_0|, \quad t \geq t_0, \quad \forall x_0 \in \mathcal{R}^n \tag{4}$$

Systemet sägs vara **globalt likformigt exponentiellt stabilt** om konstanterna $m$ och $\lambda$ är oberoende av $t_0$:

$$|s(t; x_0, t_0)| \leq m e^{\lambda(t-t_0)} |x_0|, \quad t \geq t_0, \quad \forall x_0 \in \mathcal{R}^n, \ \forall t_0 \in \mathcal{R}^+ \tag{5}$$

**Anmärkning:** Då systemet (1) är lineärt, är lokal exponentiell och global exponentiell stabilitet ekvivalent. Då systemet (1) är autonomnt (dvs inget explicit tidsberoende : $\dot{x}(t) = f(x(t))$ ) är stabilitet och likformig stabilitet ekvivalent.

Ett annat viktigt begrepp är Lipschitz kontinuitet.

**Definition 3:** En funktion $f : \mathcal{R}^n \mapsto \mathcal{R}^m$ sägs vara **lokalt Lipschitz kontinuerlig** i $x_0$ med konstant L om det finns konstanter L och $\epsilon > 0$ så att

$$|f(x) - f(x_0)| \leq L|x - x_0|, \quad \forall x \in \mathcal{B}(x_0; \epsilon) \tag{6}$$

Funktion $f : \mathcal{R}^n \mapsto \mathcal{R}^m$ sägs vara **globalt Lipschitz kontinuerlig** med konstant L om

$$|f(x) - f(x')| \leq L|x - x'|, \quad \forall x, x' \in \mathcal{R}^n \tag{7}$$

Till sist presenteras Bellman-Grönwalls olikhet.

**Lemma 1 (Bellman-Grönwalls olikhet):** Antag att $c > 0$ och $f : \mathcal{R}^+ \mapsto \mathcal{R}^+$ är lokalt integrerbar och $u \in \mathcal{L}_{\infty e}$, dvs:

$$\mathcal{P}_T u \in \mathcal{L}_\infty, \quad \forall T > 0$$

där $\mathcal{P}_T$ är trunkeringsoperatorn och $\mathcal{L}_\infty$ är rummet av funktioner, $f$, sådana att $\sup_{t \geq 0} |f(t)| \leq \infty$, och

$$u(t) \leq c + \int_0^t f(\tau)u(\tau)d\tau, \quad \forall t \in \mathcal{R}^+ \tag{8}$$

då är:

$$u(t) \leq ce^{\int_0^t f(\tau)d\tau}, \quad \forall t \in \mathcal{R}^+ \tag{9}$$

**Bevis:**

$$u(t) \leq c + \int_0^t f(\tau)u(\tau)d\tau$$

$$\frac{f(t)u(t)}{c + \int_0^t f(\tau)u(\tau)d\tau} \leq f(t)$$

$$\log(c + \int_0^t f(\tau)u(\tau)d\tau) - \log(c) \leq \int_0^t f(\tau)d\tau$$

$$c + \int_0^t f(\tau)u(\tau)d\tau \leq ce^{\int_0^t f(\tau)d\tau}$$

□

## 3.3 Lineära system med tisberoende parameter

Detta kapitel behandlar system av formen:

$$\begin{cases} \dot{x}(t) = A(\theta(t))\, x(t) + B(\theta(t))\, u(t) \\ y(t) = C(\theta(t))\, x(t) \end{cases} \tag{10}$$

Dessa ekvationer representerar ett lineärt system vars dynamik beror på en *tidsvarierande* yttre parameter, $\theta$. Ett exempel på ett verkligt system är ett flygplan, där den tidsvarierande parametern är höjd och fart (eller statiskt och dynamiskt tryck).

För att göra en parameterstyrd regulator till detta system tar man ut en mängd parametervärden, $\{\theta_i\}$, vilka representerar hela området av systemets dynamik och designar en lineär tidsinvariant regulator för varje parametervärde. Mellan dessa parametervärden interpolerar man eller byter vid vissa gränser.

Även om varje lineärt system är stabilt är det inte säkert att det slutna systemet är stabilt. Shamma ger nödvändiga villkor för att systemet skall vara stabilt.

Antag att man har designat en parameterstyrd regulator som är stabil för alla *frusna* parametervärden. Utmed varje parameterbana är det slutna systemet av formen

$$\begin{cases} \dot{x}(t) = A(t)\, x(t) \\ x(0) = x_0 \in \mathcal{R}^n \\ t \geq 0 \end{cases} \tag{11}$$

Observera att $A$ nu representerar det *slutna* systemets dynamik.
Vi förutsätter också följande:

1. Matrisen $A : \mathcal{R}^+ \mapsto \mathcal{R}^{nxn}$ är begränsad.
2. Matrisen $A$ är också globalt Lipschitzkontinuerlig med konstant $L_A$, d.v.s:

$$|A(t) - A(\tau)| \leq L_A|t - \tau| \qquad \forall t, \tau \in \mathcal{R}^+ \tag{12}$$

Till vår hjälp behöver vi också följande lemma:

**Lemma 2:** Betrakta det linjära systemet

$$\begin{cases} \dot{x}(t) = A_0 x(t) + \delta A(t)\, x(t) \\ x(0) = x_0 \in \mathcal{R}^n \\ t \geq 0 \end{cases} \tag{13}$$

Antag att det finns konstanter $m$, $\lambda$ och $k \geq 0$ så att:

$$|e^{A_0 t}| \leq me^{-\lambda t} \tag{14}$$

$$|\delta A(t)| \leq k, \quad \forall t \geq 0. \tag{15}$$

Då är $x(t)$ exponentiellt begränsad av

$$|x(t)| \leq me^{-(\lambda-mk)t}|x_0|, \qquad \forall t \geq 0, x_0 \in \mathcal{R}^n. \tag{16}$$

**Bevis:** Lösningen till (13) ges av:

$$x(t) = e^{A_0 t}x_0 + \int_0^t e^{A_0(t-\tau)}\delta A(\tau)\,|x(\tau)|d\tau$$

Använder man sedan (14) och (15) får man:

$$|x(t)| \leq me^{-\lambda t}|x_0| + \int_0^t me^{-\lambda(t-\tau)}k|x(\tau)|d\tau$$

Multiplicera sedan båda sidor med $e^{-\lambda t}$ och använd Bellman-Grönwalls olikhet,

$$|x(t)| \leq me^{-(\lambda-mk)t}|x_0|.$$

□

**Anmärkning:** Lemma 2 säger att om det ostörda tidsinvarianta systemet ($\delta A(t) \equiv 0$) är exponentiellt stabilt (14) så är också det störda systemet exponentiellt stabilt (16) om störningen är tillräckligt liten (15).

Nu har vi kommit fram till huvudresultatet i detta avsnitt:

**Sats 1:** Betrakta det lineära systemet (11) under förutsättningarna 1 och 2. Antag även att

- $A(t)$ är stabil för alla t
- Det finns konstanter $m$ och $\lambda \geq 0$ sådana att:

$$|e^{A(\tau)t}| \leq me^{-\lambda t}, \qquad \forall t, \tau \geq 0. \tag{17}$$

Under dessa villkor, givet något $\eta \in [0, \lambda]$,

$$L_A \leq \frac{(\lambda-\eta)^2}{4m\ln m} \tag{18}$$

implicerar

$$|x(t)| \leq me^{-\eta t}|x_0|, \qquad \forall t \geq 0, x_0 \in \mathcal{R}^n. \tag{19}$$

**Bevis:** Approximera $A(t)$ i (11) med den styckvis konstanta matrisen

$$A_{pc}(t) \equiv A(nT), \qquad nT \leq t < (n+1)T, \; n = 0, 1, 2, \ldots$$

där $T$ skall väljas lämpligt. Skriv om (11):

$$\dot{x}(t) = A_{pc}(t)\, x(t) + \{A(t) - A_{pc}(t)\}\, x(t).$$

Välj nu

$$T = \frac{2 \ln m}{\lambda - \eta} \tag{20}$$

Då gäller för alla $t \geq 0$:

$$|A(t) - A_{pc}(t)| \leq L_A T \leq \frac{\lambda - \eta}{2m}$$

där $L_A$ är vald från (18). Det följer nu av Lemma 2 att på intervallet $nT \leq t < (n+1)T$,

$$\begin{aligned}
|x(t)| &\leq me^{-\frac{\lambda+\eta}{2}(t - nT)}|x(nT)| \\
&\leq me^{-\frac{\lambda+\eta}{2}(t - nT)}\{me^{-\frac{\lambda+\eta}{2}T}\}^n|x_0| \\
&= me^{-\eta t}\underbrace{e^{-\frac{\lambda-\eta}{2}(t - nT)}}_{\leq 1}\underbrace{\{me^{-\frac{\lambda-\eta}{2}T}\}^n}_{=1}|x_0| \\
&\leq me^{-\eta t}|x_0|
\end{aligned}$$

□

**Anmärkning:** Satsen säger att ett tidsvarierande system (11) behåller dess exponentiella stabilitet för ***frusen tid*** (17) om tidsvariationerna är tillräckligt små (18). Observera att tidsvariationerna hos insignal och utsignal får vara hur stora som helst. Det är bara ändringarna i dynamiken hos systemet som måste vara tidsbegränsade. Om dynamiken ändrar sig snabbt i ett litet område är Lemma 2 mer användbart.

Beviset kan även förklaras i ord: Först approximeras den tidsvarierande dynamiken med styckvis konstant dynamik. På varje styckvis konstant intervall består det lineära systemet av en stabil tidsinvariant del och en tidsvarierande störning. Lemma 2 ger att tillstånden kommer att avklinga exponentiellt om störningen är tillräckligt liten - eller approximationen är godtagbar. När approximationen inte är tillräckligt bra längre gör man en ny.

Sedan behandlar Shamma en del andra metoder för att avgöra stabilitet, Lyaponov och matrisexponentialmetoden, men finner att i verkligheten ger alla dessa metoder ett alldeles för pessimistiskt resultat.

Till sist behandlar han robusthet mot störningar och omodellerad dynamik med hjälp av Volterra Integrodifferentialekvationer.

# 4 Kommentarer

När man nu skall göra en parameterstyrd regulator tar man först reda på hur systemet uppträder vid *frusen* dynamik (17) och får då fram $m$ och $\lambda$. Sedan undersöker man hur dynamiken ändrar sig genom att bestämma Lipschitzkonstanten $L_A$ i (18). Detta ger ett maxvärde på parametern $\eta$, vilken talar om hur snabbt det slutna systemet kan avklinga. Till sist fås $T$, som talar om hur fin indelning man behöver göra för parametervärdena, av (20).

Man kan observera att om det inte ställs så stora krav på det slutna systemets snabbhet ($\eta$ nära noll) blir $T$ litet (20), d.v.s. en fin indelning av parameterintervallen. Om man däremot vill ha så snabbt slutet system som möjligt ($\eta$ nära $\lambda$) blir $T$ stort, d.v.s. en grövre indelning. Detta stämmer inte med den intuitiva uppfattningen att högre krav på systemet borde kräva en finare indelning. Detta verkar konstigt och borde undersökas närmare.

Shamma konstaterar att om $m = 1$ kan man tillåta godtyckligt stora variationer i systemets dynamik (18), d.v.s. instabilitet kan aldrig inträffa p.g.a att systemets dynamik ändras. Man undrar vad som händer då $m$ är mindre än ett, ty då blir både Lipschitzkonstanten $L_A$ och tidsparametern $T$ (20) negativa.

En annan viktig sak är att satsen behandlar bara fallet då man växlar mellan olika regulatorer vid vissa gränser ($nT$), men inte vad som händer om man linjärinterpolerar mellan regulatorparametrarna. Man frågar sig också om det kan bli bättre resultat om man använder ett annat interpolationspolynom.

Shamma testar sina resultat på det amerikanska forskningsflygplanet F-8, men finner att i praktiken ger hans satser alltför pessimistiska resultat, d.v.s. i verkligheten kan man tillåta mycket större tidsvariationer i dynamiken än vad satserna kräver för att garantera global stabilitet. Han anger inga direkta vägar för att komma närmare de verkliga resultaten mer än att simulera sig fram.

Det bör observeras att kravet på Lipschitzkonstanten $L_A$ i (18) hänger samman med valet av $T$ i (19). Shamma har valt dem på ett sätt men det finns andra sätt att kombinera dessa båda.

Sedan består resten av avhandlingen av parameterstyrning av olineära system med först referenssignalen som parameterstyrande variabel och sedan utsignalen. Vid olinära system måste man dock känna till vilka olika typer av referenssignaler som kan komma ifråga samt hur tillstånden utvecklar sig vid dessa referenssignaler. Shamma behandlar också robusthet mot störningar och omodellerad dynamik för dessa olineära system.

/5.

## 5 Sammanfattning

Parameterstyrning är en reglerprincip som fungerar bra i många samanhang, men som saknar en teoretisk analys av stabilitet. Shamma ger i sin avhandling tillräckliga villkor för att det slutna systemet skall vara stabilt samt för att det skall vara robust mot störningar och omodellerad dynamik. Han behandlar också parameterstyrning av olineära system och ger villkor som garanterar stabilitet och robusthet, m.h.a. Volterra integrodifferentialekvationer, *VIDE*.

16.

# Referenser


[1] K.G. Andersson och L.-C. Böiers,*Ordinära Differentialekvationer*, Lunds Universitets Matematiska Institution,1984

[2] R.Bellman,*Stability Theory of Differential Equations*, Mc Graw-Hill, New York, 1953

[3] J.S. Shamma, *Analysis and Design of Gain Scheduled Control Systems*, Ph.D thesis, Laboratory for Information and Decision Systems, MIT, Cambridge,1988

[4] J.S. Shamma och M. Athans,"Stability and Robustness of Slowly Time-Varying Linear Systems", *Proceedings of the 26th IEEE Conference on Decision and Control*, 1987

[5] J.S. Shamma och M. Athans,"Guaranteed Properties for Nonlinear Gain Scheduled Control Systems", *Proceedings of the 27th IEEE Conference on Decision and Control*, 1988

[6] M. Vidyasagar, *Nonlinear Systems Analysis*, Printice-Hall, McGraw-Hill, Englewood Cliffs, NJ, 1978

[7] K.J. Åström och B. Wittenmark, *Adaptive Control*, Addison-Wesley, 1989

17.

# Kommentarer till en rapport av Kumar

projektarbete i adaptiv reglering

Anders Hansson


## Referat

En rapport av Kumar—Convergence of adaptive control schemes using least-squares parameter estimates har kommenterats. Resultaten i rapporten har verifierats med simuleringar.

18.

# Förord

Denna rapport har tillkommit såsom resultat av ett projektarbete i kursen Adaptiv reglering vid Institutionen för Reglerteknik, LTH. Jag vill här passa på att tacka Karl Johan Åström för god och inspirerande handledning. Tack riktas även till Kjell Gustafsson och Bo Bernhardsson.

19.

# Innehåll

20.

# 1. Inledning

Ändamålet med detta arbeta har varit att försöka förstå en rapport av Kumar— Convergence of adaptive control schemes using least-squares parameter estimates. Emellertid har det ej lagts någon större vikt vid att förstå refererade arbeten. För den skull hade det varit nödvändigt med grundligare förkunskaper i sannolikhetsteori och måtteori.

Första delen av arbetet består av en uppräkning av kommentarer och förklaringar till rapporten. Andra delen består av resultat från simuleringar, vilka verifierar Kumars resultat.

# 2. Kommentarer och förklaringar

## 2.1 Introduction

Kumar analyserar indirekta adaptiva regulatorer, där parametrarna skattas med rekursiv minstakvadratmetod. Han inskränker sig till fallet med vitt och normalfördelat brus och visar 1) att parametrarna konvergerar med sannolikhet 1, 2) att det slutna systemet är stabilt om det sanna systemet är minimumfas, 3) att parametrarnas gränsvärde korrekt beskriver det slutna systemets överföringsfunktion från brus till utsignal, 4) att den adaptiva regulatorn är optimal, d.v.s. den reglerar lika bra som en konventionell regulator, 5) att om insignalerna är tillräckligt exiterande, så är parameterskattningarna konsistenta och 6) att de första $d-1$ koefficienterna i $A$-polynomet skattats konsistent för system med fördröjning $d > 1$ även om det inte finns någon insignal.

## 2.2 Convergence of recursive least-squares via bayesian embedding

I det här kapitlet visar Kumar att parameterestimaten konvergerar. Detta bevis bygger på att $\{\hat{\theta}(t), \mathcal{Y}_t\}$ är en konvergent martingal. För att den skall vara det krävs (Hall och Heyde 1980):

1) $\mathcal{Y}_t \subset \mathcal{Y}_{t+1}$

2) $\hat{\theta}(t) \in \mathcal{Y}_t$

3) $E(\mid \hat{\theta}(t) \mid) < \infty$

4) $\hat{\theta}(t) = E\left(\hat{\theta}(t) \mid \mathcal{Y}_t\right)$, a.e.

5) $\sup_t E\left(\mid \hat{\theta}(t) \mid^p\right) < \infty$ för något $p$.

Punkten 1 är trivialt uppfylld. Punkten 2 betyder att skattningen är en mätbar funktion relativt $\mathcal{Y}_t$, vilket den är om den beror kontinuerligt på de stokastiska variabler som genererar $\mathcal{Y}_t$. Punkt 4 är en välkänd egenskap hos Kalmanfiltret. Emellertid visar Kumar inte att punkterna 3 och 5 är uppfyllda. Han hänvisar till en annan artikel skriven av honom själv (Chen m.fl. 1989), som jag tyvärr inte lyckats få tag på. Om dessa punkter är uppfyllda, vet man att skattaren är stabil, och det enda som detta teorem då utesluter är oscillerande lösningar. Det skall också påpekas att konvergensen bara är bevisad så när som på en mängd sanna parametrar med Lebesgue-mått 0.

21.

### 2.3 The normal equations for the limiting parameter estimate

I detta kapitel visar Kumar att två gränsvärde är 0. För detta ändamål refererar han ett teorem om lokal konvergens av martingaler (Chow 1965), som jag tyvärr inte har lyckats tränga igenom. Accepterar man detta resultat, så är resten självklart. Detta resultat har inget självständigt intresse utan utnyttjas i kommande bevis.

### 2.4 The stability of indirect adaptive control laws

I detta kapitel visar Kumar att andramomenten för in- och utsignal är ändliga, om man reglerar ett minimumfassystem med en $RST$-regulator.

### 2.5 Characterization of asymptotic performance of indirect adaptive control laws

Här visar Kumar att skattningarna uppfyller en viss polynomekvation, vilket medför att det slutna systemets överföringsfunktion från brus till utsignal estimeras korrekt. Han visar också att två gränsvärde är 0. Dessa kan användas för att visa att regulatorn är optimal relativt designkriteriet. Slutligen visar han att de $d-1$ första parametrarna i $A$-polynomet skattas konsistent för system med tidsfördröjning $d > 1$. Bevisen för dessa påståenden refererar till Kumar och Praly (1987), vilka jag tyvärr ej heller lyckats tränga igenom. Det sista påståendet har en trevlig tolkning: det går endast att kompensera felaktiga estimat av en parameter med motsvarande felaktigt estimat av en annan parameter om dessa svarar mot samma tidsfördröjning.

Resterande kapitel avhandlar specifika regulatorer. Jag har valt att enbart kommentera det 9:e kapitlet och då med simuleringarna som följer härefter.

## 3. Simuleringar

Simuleringarna är gjorda på fjärde ordningens system med tidsfördröjning 2:

$$Ay = q^{-2}Bu + e,$$

där

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3} + a_4q^{-4}$$

och

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2}.$$

Regulatorn är en minimalvariansregulator med börvärde:

$$BFu + Gy = u_c,$$

där $F$ och $G$ fås som lösning till

$$AF + q^{-2}G = 1,$$

där

$$F(q^{-1}) = 1 + f_1q^{-1}$$

och

$$G(q^{-1}) = g_0 + g_1q^{-1} + g_2q^{-2} + g_3q^{-3}.$$

I de första 4 simuleringarna är $A$ valt så att $a_0 \neq 0$:

$$A(q^{-1}) = 1 - q^{-1} - 0.5q^{-2} - 0.5q^{-3} + 0.25q^{-4}.$$

I de 2 första av dessa har $B$ och $G$ inga gemensamma faktorer:

$$B(q^{-1}) = 1 - q^{-1} + 0.25q^{-2};$$

i de 2 sista har de det:

$$B(q^{-1}) = 1 + q^{-1} + 0.5q^{-2}.$$

I de 4 sista simuleringarna är $A$ valt så att $a_0 = 0$:

$$A(q^{-1}) = 1 + q^{-2} + 0.25q^{-4}.$$

I de 2 första av dessa har $B$ och $G$ inga gemensamma faktorer:

$$B(q^{-1}) = 1 - q^{-1} + 0.25q^{-2};$$

i de 2 sista har de det:

$$B(q^{-1}) = 1 - 0.25q^{-2}.$$

Varannan simulering är gjord med börvärde (fyrkantvåg med amplitud 1 och period 20), varannan utan. Bruset är vitt och normalfördelat med standardavvikelsen 0.01. Resultaten kan ses i figurerna 1–8. Följande skall enligt teorem 8 gälla för de olika simuleringarna:

- Regulatorn är alltid optimal relativt minimalvarianskriteriet.
- Parametern $a_1$ skattas alltid konsistent.
- I simulering 1, 2 och 5 är alla skattningar konsistenta.
- I simulering 4 skattas regulatorns överföringsfunktion rätt.
- I simulering 6 är skattningarna korrekta på en faktor när.

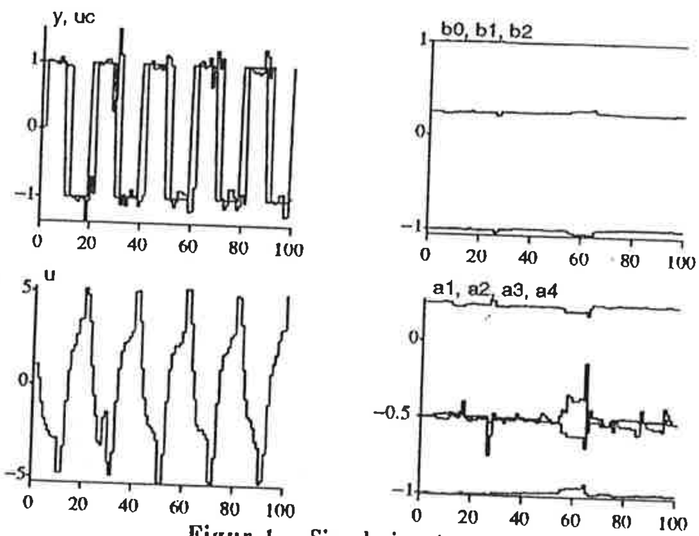
Med god vilja kan det tyckas att det överensstämmer med simuleringarna. Invändas kan att simuleringarna är för korta, men under arbetets gång har observerats att för simuleringar av längd 1000 kan underligheterna vänta med att inträffa tills kring 900. Det slutna systemet är stabilt även om styrsignalen har värdet -20000—det är en lång väg till oändligheten.

## 4. Slutsatser

Kumar ger ett bevis för att parametrarna alltid konvergerar föroutom på en mängd med Lebesguemått 0—exempel på en sådan mängd är alla de tal som går att representera i en dator. Deras väg till gränsvärdet kan vara minst sagt besynnerlig. Styrsignaler kring -20000 kan inte anses vara normalt för ett i vettig mening stabilt system. Man får ej heller glömma att kravet på vitt och normalfödelat brus är hårt, vilket Kumar också påpekar.

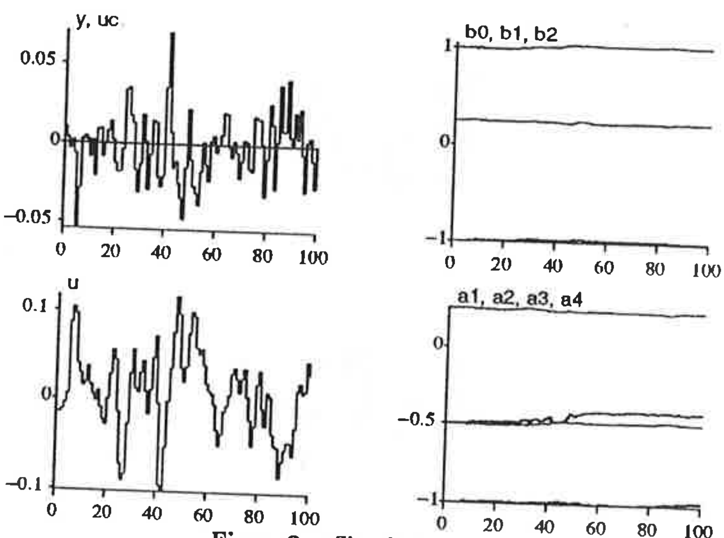
23.

89.05.22 - 20:45:58 nr: 1
hcopy meta/fig1



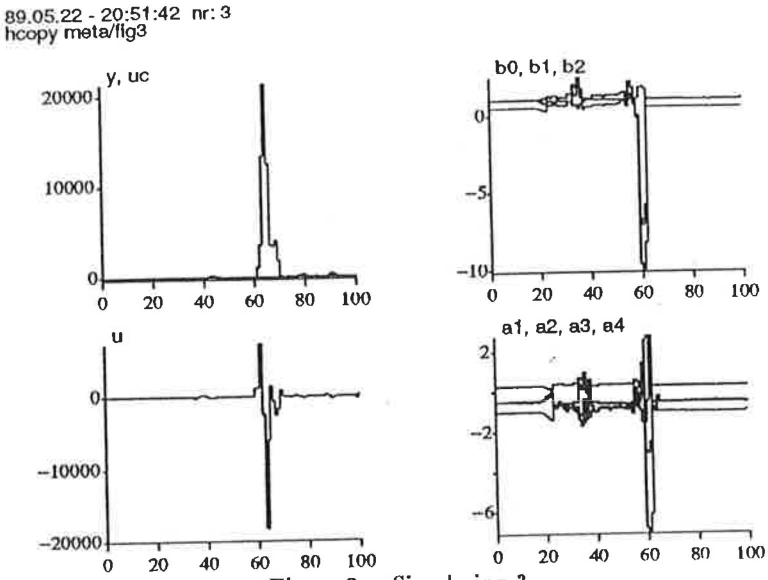

**Figur 1.** Simulering 1

89.05.22 - 20:49:09 nr: 2
hcopy meta/fig2

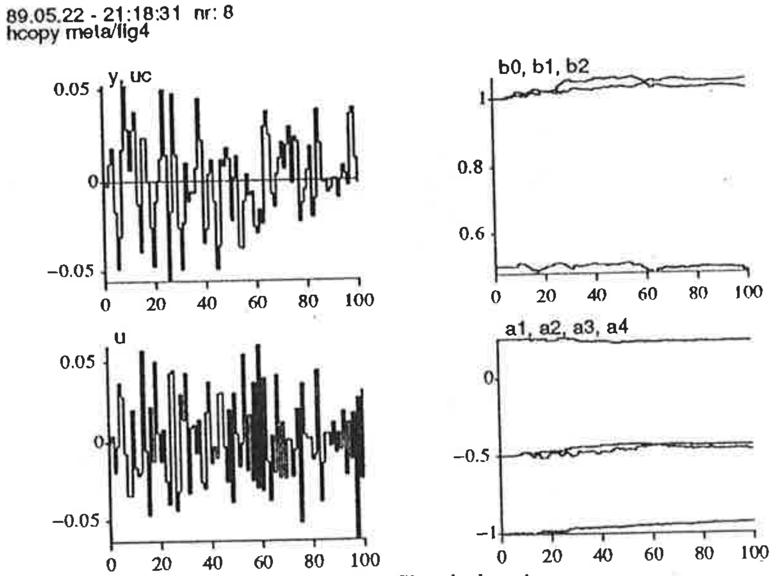



**Figur 2.** Simulering 2

24.

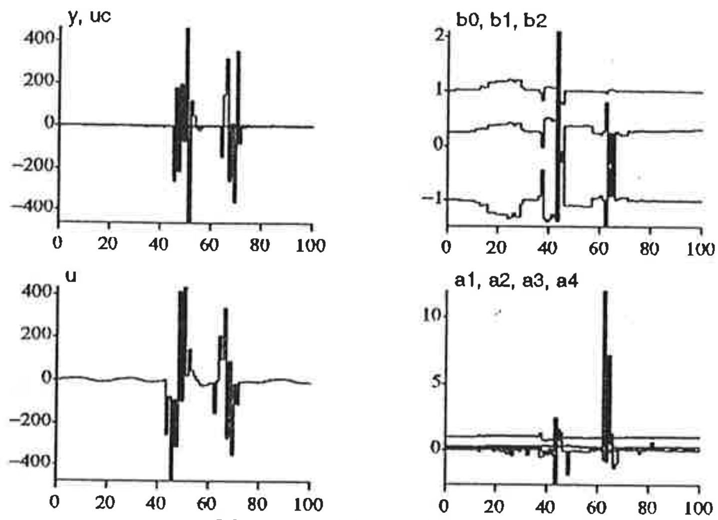



**Figur 3.** Simulering 3



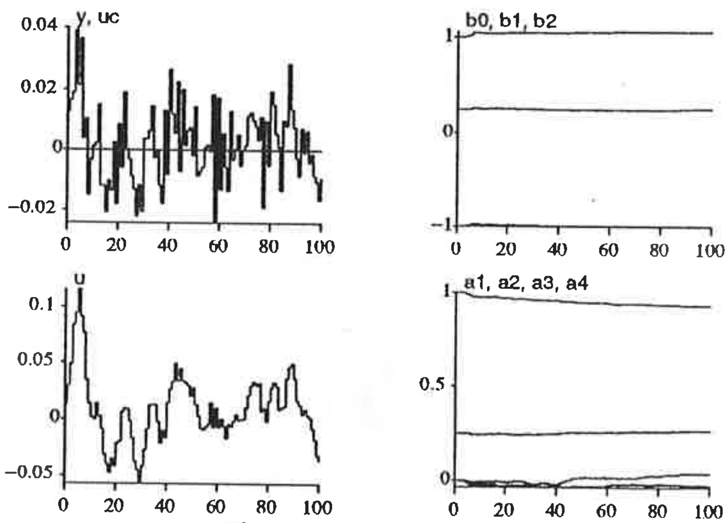

**Figur 4.** Simulering 4

89.05.22 - 20:57:05 nr: 4
hcopy meta/fig5




**Figur 5.** Simulering 5

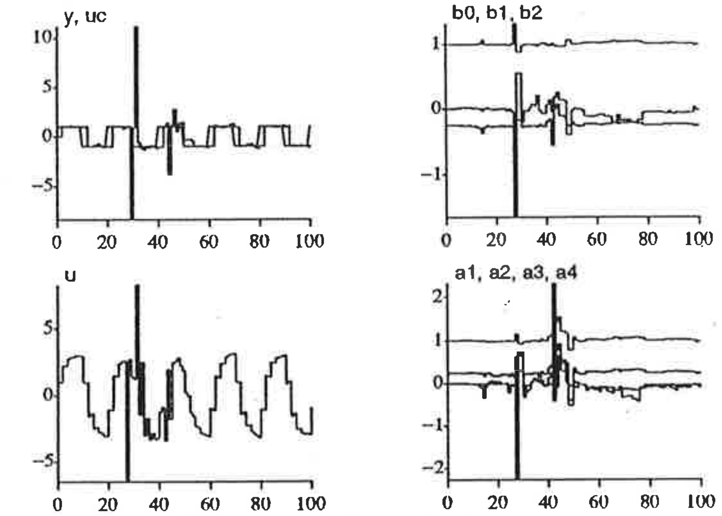
89.05.22 - 21:00:46 nr: 5
hcopy meta/fig6




**Figur 6.** Simulering 6

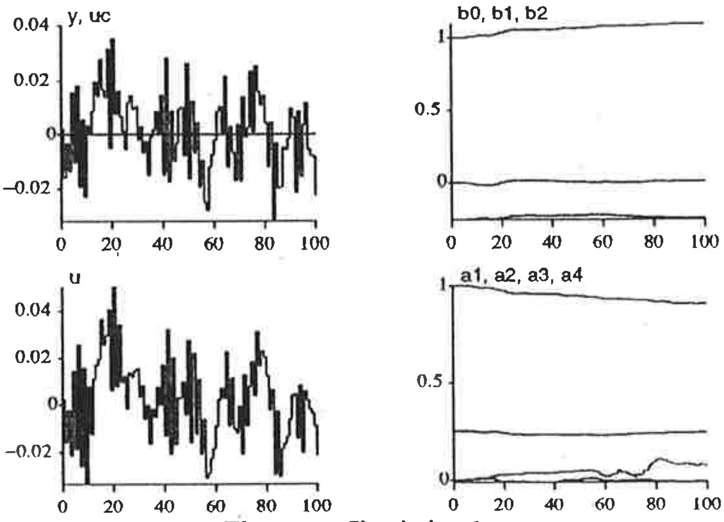
26.

89.05.22 - 21:02:03 nr: 6
hcopy meta/fig7




**Figur 7.** Simulering 7

89.05.22 - 21:04:10 nr: 7
hcopy meta/fig8




**Figur 8.** Simulering 8

27.

## 5. Referenser

CHEN, H. F., P. R. KUMAR, and C. Z. WEI (1989): "On Kalman filtering for conditionally gaussian systems with random matrices," Tech. Rep., Coordinated Science Laboratory, University of Illinois.

CHOW, Y.S. (1965): "Local convergence of martingales and the law of large numbers," *Annals of Mathematical Statistics*, **36**, 552–558.

CHOW, Y. and H. TEICHER (1988): *Probability Theory: Idependence, Interchangeability, Martingales*, Springer-Verlag, New-York.

GOODWIN, G., P. RAMADGE and P. CAINES (1981): "Discrete time stochastic adaptive control," *SIAM Journal on Control and Optimization*, **19**, 829–853.

HALL, P. and C. C. HEYDE (1980): *Martingale Limit Theory and Its Application*, Academic Press, New-York.

KUMAR, P. R. (1989): "Convergence of Adaptive Control Schemes using Lest-squares Parameter Estimates," Department of Electrical and Computer Engineering and the Coordinated Science Laboratory, University of Illinois.

KUMAR, P. R. and L. PRALY (1987): "Self-tuning trackers," *SIAM Journal on Control and Optimization*, **25**, 1053–1071.

LIPTSER, R. S. and A. N. SHIRYAYEV (1977): *Statistics of Random Processes, II: Applications*, Springer-Verlag, New-York.

# A. Simnonprogram

```
macro istrfig
store th1 th2 th3 th4 th5 th6 th7 y[reg] u[reg] uc[reg]
simu 0 100
split 2 2
ashow y[reg] uc[reg]
text 'y, uc'
ashow u[reg]
text 'u'
ashow th1 th2 th3
text 'b0, b1, b2'
ashow th4 th5 th6 th7
text 'a1, a2, a3, a4'
disp th1 th2 th3 th4 th5 th6 th7
END

connecting system conn
time t
c=if mod(t,tp)<tp/2 then ampl else -ampl
tp=20
uc[reg]=c
y[reg]=y[plant]+e1[noise1]
u[plant]=u[reg]
ampl:1
end
```

```
discrete system plant
"file called dplant.t

input u
```

```
output y

state y1 y2 y3 y4
new ny1 ny2 ny3 ny4

time t
tsamp ts

ny1=-a1*y1-a2*y2-a3*y3-a4*y4+u
ny2=y1
ny3=y2
ny4=y3
y=b0*y2+b1*y3+b2*y4

ts=t+h

h:1

a1:-1.0
a2:-0.5
a3:-0.5
a4:0.25
b0:1.0
b1:1.0
b2:0.25

end

discrete system reg
"file called istr7.t

input y uc
output u

state th1 th2 th3 th4 th5 th6 th7
state p11 p12 p13 p14 p15 p16 p17 p22 p23 p24 p25 p26 p27
state p33 p34 p35 p36 p37 p44 p45 p46 p47 p55 p56 p57 p66 p67 p77
state f1 f2 f3 f4 f5 f6 f7 u1
state z11 z12 z13 z21 z22 z23
new nth1 nth2 nth3 nth4 nth5 nth6 nth7
new np11 np12 np13 np14 np15 np16 np17 np22 np23 np24 np25 np26 np27
new np33 np34 np35 np36 np37 np44 np45 np46 np47 np55 np56 np57 np66 np67 np77
new nf1 nf2 nf3 nf4 nf5 nf6 nf7 nu1
new nz11 nz12 nz13 nz21 nz22 nz23
time t
tsamp ts

"residual
e=y-f1*th1-f2*th2-f3*th3-f4*th4-f5*th5-f6*th6-f7*th7

"P(t-1)*Fi(t-1)
pf1=p11*f1+p12*f2+p13*f3+p14*f4+p15*f5+p16*f6+p17*f7
pf2=p12*f1+p22*f2+p23*f3+p24*f4+p25*f5+p26*f6+p27*f7
pf3=p13*f1+p23*f2+p33*f3+p34*f4+p35*f5+p36*f6+p37*f7
pf4=p14*f1+p24*f2+p34*f3+p44*f4+p45*f5+p46*f6+p47*f7
pf5=p15*f1+p25*f2+p35*f3+p45*f4+p55*f5+p56*f6+p57*f7
pf6=p16*f1+p26+f2+p36*f3+p46*f4+p56*f5+p66*f6+p67*f7
pf7=p17*f1+p27*f2+p37*f3+p47*f4+p57*f5+p67*f6+p77*f7

"lambda+Fi(t-1)*P(t-1)*Fi(t-1)
n=lambda+f1*pf1+f2*pf2+f3*pf3+f4*pf4+f5*pf5+f6*pf6+f7*pf7

"K
k1=pf1/n
k2=pf2/n
k3=pf3/n
k4=pf4/n
k5=pf5/n
k6=pf6/n
k7=pf7/n
```

29.

```
"theta(t)
nth1=if abs(th1+k1*e)>0.1 then th1+k1*e else th1
nth2=if abs(th1+k1*e)>0.1 then th2+k2*e else th2
nth3=if abs(th1+k1*e)>0.1 then th3+k3*e else th3
nth4=if abs(th1+k1*e)>0.1 then th4+k4*e else th4
nth5=if abs(th1+k1*e)>0.1 then th5+k5*e else th5
nth6=if abs(th1+k1*e)>0.1 then th6+k6*e else th6
nth7=if abs(th1+k1*e)>0.1 then th7+k7*e else th7

"p
np11=(p11-pf1*pf1/n)/lambda
np12=(p12-pf1*pf2/n)/lambda
np13=(p13-pf1*pf3/n)/lambda
np14=(p14-pf1*pf4/n)/lambda
np15=(p15-pf1*pf5/n)/lambda
np16=(p16-pf1*pf6/n)/lambda
np17=(p17-pf1*pf7/n)/lambda
np22=(p22-pf2*pf2/n)/lambda
np23=(p23-pf2*pf3/n)/lambda
np24=(p24-pf2*pf4/n)/lambda
np25=(p25-pf2*pf5/n)/lambda
np26=(p26-pf2*pf6/n)/lambda
np27=(p27-pf2*pf7/n)/lambda
np33=(p33-pf3*pf3/n)/lambda
np34=(p34-pf3*pf4/n)/lambda
np35=(p35-pf3*pf5/n)/lambda
np36=(p36-pf3*pf6/n)/lambda
np37=(p37-pf3*pf7/n)/lambda
np44=(p44-pf4*pf4/n)/lambda
np45=(p45-pf4*pf5/n)/lambda
np46=(p46-pf4*pf6/n)/lambda
np47=(p47-pf4*pf7/n)/lambda
np55=(p55-pf5*pf5/n)/lambda
np56=(p56-pf5*pf6/n)/lambda
np57=(p57-pf5*pf7/n)/lambda
np66=(p66-pf6*pf6/n)/lambda
np67=(p67-pf6*pf7/n)/lambda
np77=(p77-pf7*pf7/n)/lambda

"u(t-d+1)
nu1=u

"fi
nf1=u1
nf2=f1
nf3=f2
nf4=-y
nf5=f4
nf6=f5
nf7=f6

"process-par
b0=th1
b1=th2
b2=th3
a1=th4
a2=th5
a3=th6
a4=th7

"regulator-par
s0=(a1*a1-a2)/b0
s1=(a1*a2-a3)/b0
s2=(a1*a3-a4)/b0
s3=a1*a4/b0
r1=(b1-a1*b0)/b0
r2=(b2-a1*b1)/b0
r3=-a1*b2/b0
t0=1/b0

"u
```

```
nz11=-r1*z11-r2*z12-r3*z13+uc
nz12=z11
nz13=z12
ucf=-t0*r1*z11-t0*r2*z12-t0*r3*z13

nz21=-r1*z21-r2*z22-r3*z23+y
nz22=z21
nz23=z22
yf=(s1-s0*r1)*z21+(s2-s0*r2)*z22+(s3-s0*r3)*z23

u=t0*uc+ucf-(s0*y+yf)

"update sampling time
ts=t+h

"A and B
th1:1.0
th2:1.0
th3:0.25
th4:-1.0
th5:-0.5
th6:-0.5
th7:0.25

"h
h:1.0

"p
p11:10
p22:10
p33:10
p44:10
p55:10
p66:10
p77:10

"lambda
lambda:1.0

end
```

31.

# Convergence of Adaptive control schemes using least–squares parameter estimates
## P. R. Kumar

Reviewed by: Pablo A. Iglesias
Department of Engineering
Cambridge University

### Introduction

It is well known that the convergence properties of least–squares (LSE) parameter estimators are much better than that of stochastic gradient (SGA) algorithms. For this reason, they are almost always preferred in practical situations. One such use is as the parameter estimator in adaptive control algorithms. Unfortunately, since the LS algorithms are considerably more complicated that SGA, very few theoretical results exist in this context. It is this gap in knowledge that the author aims to fill.

### Bayesian Embedding

The approach used by the author to prove convergence is referred to as "Bayesian embedding." That is, the "true" parameter is assumed to be a random variable with a Gaussian distribution. Then, under the assumption that the system is subject to white, independent gaussian noise, the parameters are shown to converge, almost surely.

The proof is done by showing that $\{\hat{\theta}_t, \mathcal{Y}_t\}$ (where $\{\mathcal{Y}_t\}$ is the $\sigma$–algebra generated by all outputs up to time $t$) is a martingale. Unfortunately, this is really not proven in the present context. The usual notion of martingales would require some integrability condition on the regression vector $\phi(t)$, see [C], page 319. To put such a condition in the adaptive control context, however, assumes that the system's inputs and outputs are bounded. The author states that a proof (where no integrability conditions are needed) is available, and then references a paper co–authored by himself, [CKS]. However, since this reference is to an internal report dated February 1989 (the present paper was received on 09/02/89) one must really take the current results skeptically. In view of the considerable time that is spent showing how the results presented can be used to prove the convergence of some existing algorithms, it seems reasonable to expect that the result should have been presented fully.

### Summary of results

The results obtained will now be summarized. The theorem on the convergence of least squares estimates in indirect adaptive control requires the following assumptions:

1. Minimum phase system (since all zeros must be cancelled).
2. Output additive white, gaussian noise.
3. Perfectly modelled system.

The algorithms to which this result are applied are the following:

1. *Self-tuning regulator of* [ÅW1], *unit delay case.* It is shown that the controller obtained converges to a minimum–variance regulator. Moreover, the system's parameter estimates converge to some scalar multiple of the true parameters. If in addition, the parameter $b_1$ is fixed, then the convergence is to the true parameter.
2. *Self-tuning regulator for general delay case.* The adaptive controller is optimal with respect to minimum variance, and (under certain coprimeness comditions) the parameter estimates converge a scalar multiple of the true parameters.
3. *Self-tuning minimum variance trackers.* Again, the adaptive controller achieves optimal tracking with respect to minimum variance, except that in this case, the parameter estimates only converge if the reference is persistently exciting.
4. *Self-tuning pole-placement algorithms.* The algorithm considers the minimum degree solution to the servo problem. The goal of achieving a desired closed loop tansfer function is achieved, and the system's parameters converge to the true parameters under a persistent excitation condition.

### Simulations

Some of the results stated were simulated. In particular two systems were considered. The first system consists of a normalized model of a motor as in Example 5.1, [ÅW2]. This is a second order system with continuous time transfer function $G(s) = \{s(s+1)\}^{-1}$. When sampled at 2 Hz., the system has a discrete–time transfer function equal to

$$\frac{0.107z + 0.090}{z^2 - 1.607z + 0.61}.$$

The system was then controlled using two type of regulators: a pole–placement algorithm giving the closed loop characteristic equation of $z^2 - 1.32z + 0.5$ and a minimum variance regulator. The initial estimates were chosen at random and the system was simulated for 500 samples. The final estimated parameters can be found on Tables 1 and 2. Sample trajectories (corresponding to the first 100 samples) can be seen in Figures 1 and 2. Notice that the system does converge to multiples of the true parameter, and in fact in most cases this convergence is very close to the true parameters.

The second example is that of an unstable system $G(s) = (s-1)^{-1}$ also sampled twice a second. Again a pole placement algorithm, placing the closed loop pole at $z = 0.607$, and a minimum variance regulator were used. The system's parameter estimates can be found on tables 3 and 4, and sample trajectories in Figures 3 and 4. Notice that despite large initial transients the system was controlled in all cases.

### Conclusions

The paper considered in this project has certainly filled a lot of gaps in the convergence analysis of adaptive control algorithms. In can of course be argued that the additive white noise, since it has non–zero spectrum at all frequencies, achieves the persistent excitation requirement of most indirect adaptive controllers. In fact it gives a persistent excitation of infinite order. It must also be said that the results obtained are, however, purely academic in interest since they still require a complete modelling assumption on the plant. The author suggests, as possible extensions of this work, the relaxation of the independence and the whiteness condition on the noise. On other improvement would be the elimination of the minimum phase assumption. While it is an obvious requirement for the schemes using minimum variance regulation, it should not be necessary for the pole–zero placement regulators.

Although the convergence of adaptive controllers is important, it seems to be only possible in these ideal cases. Perhaps the most important contribution of this paper is that it will redirect the attention of adaptive control research from the task of convergence in idealized cases, to the design and implementation of sensible adaptive controllers.

**References**

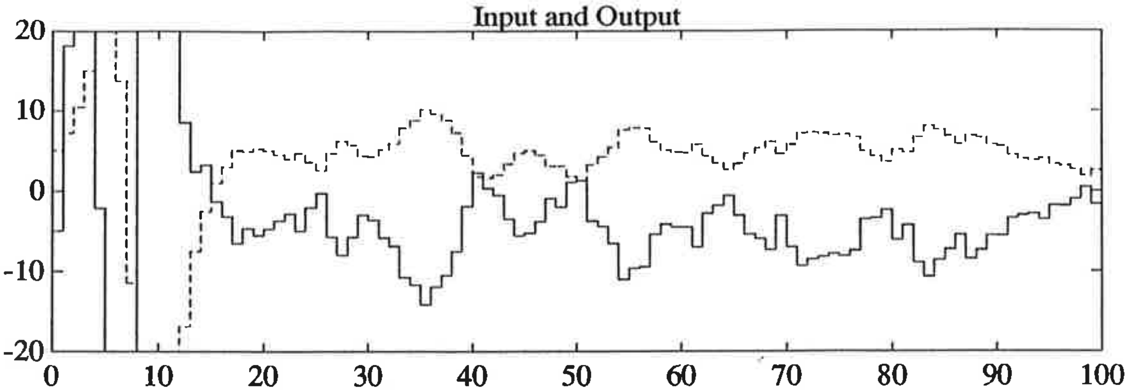

[C] Chung, K. L. *A course in probability theory, 2nd Ed.*, Academic Press, New York, NY, 1974.

[CKS] Chen, H. F., P. R. Kumar and H. H. van Schuppen, "On Kalman filtering for conditionally Gaussian systems with random matrices," Technical report, Coordinated Science Laboratory, University of Illinois, February 1989.

[ÅW1] Åström, K. J. and B. Wittenmark, "On self–tuning regulators," *Automatica*, **9**: 185–199, 1973.

[ÅW2] Åström, K. J. and B. Wittenmark, *Adaptive control*, Addison–Wesley, Reading, MA, 1989.

34.



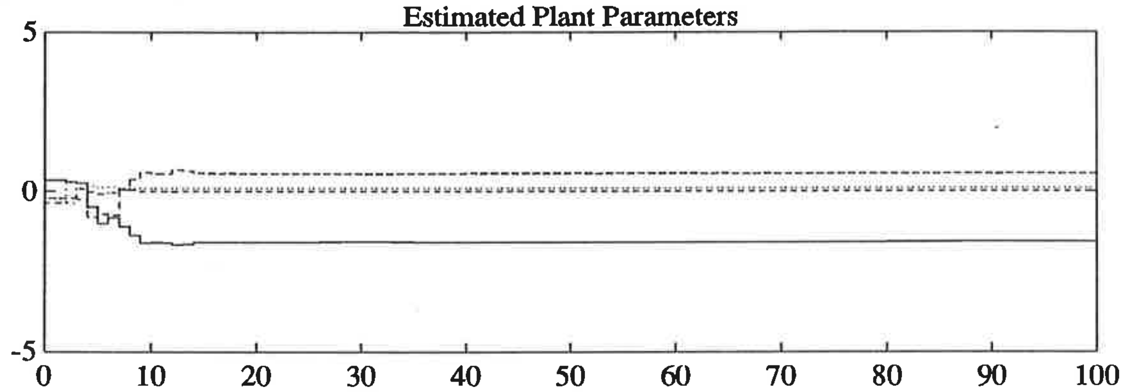
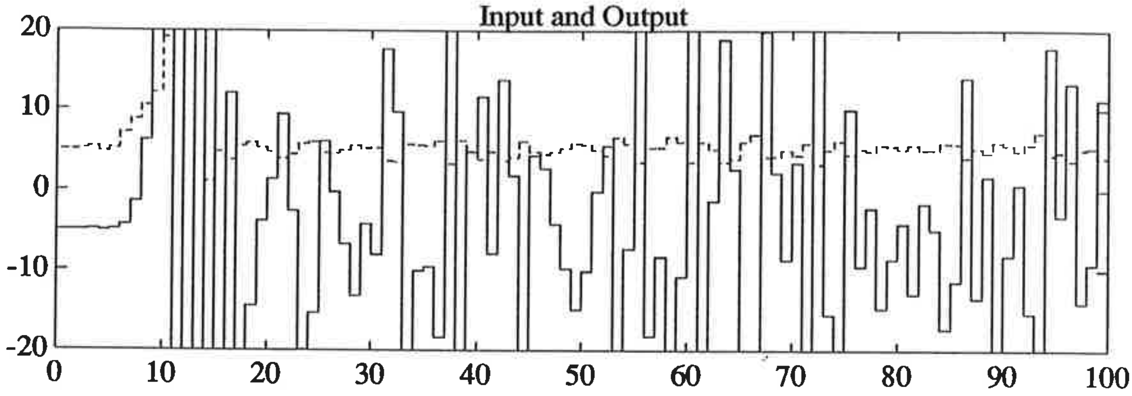




**Figure 1.** Input and output measurements as well as parameter estimates for the second order system using a pole placement regulator. Note that to separate the plots of $y$ and $u$ these have been displaced by 5 and -5 units respectively.

| Simulation # 1 | | | | | | |
|---|---|---|---|---|---|---|
| Parameter | True Value | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 |
| $a_1$ | $-1.607$ | $-1.549$ | $-1.603$ | $-1.609$ | $-1.631$ | $-1.550$ |
| $a_2$ | 0.610 | 0.551 | 0.594 | 0.613 | 0.653 | 0.552 |
| $b_1$ | 0.107 | 0.100 | 0.100 | 0.107 | 0.105 | 0.106 |
| $b_2$ | 0.090 | 0.106 | 0.103 | 0.090 | 0.106 | 0.896 |

**Table 1.** Parameter estimates for second order stable system controlled using a certainty equivalence indirect adaptive controller using pole placement regulator.

35.



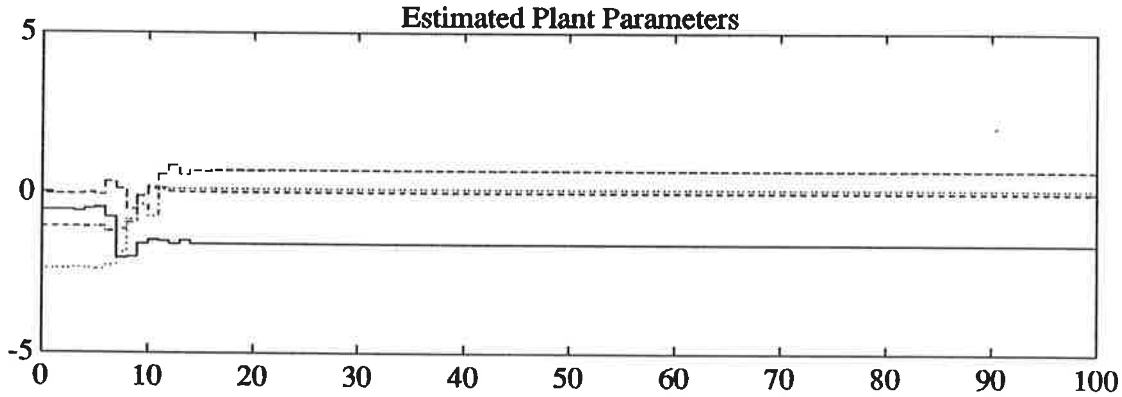
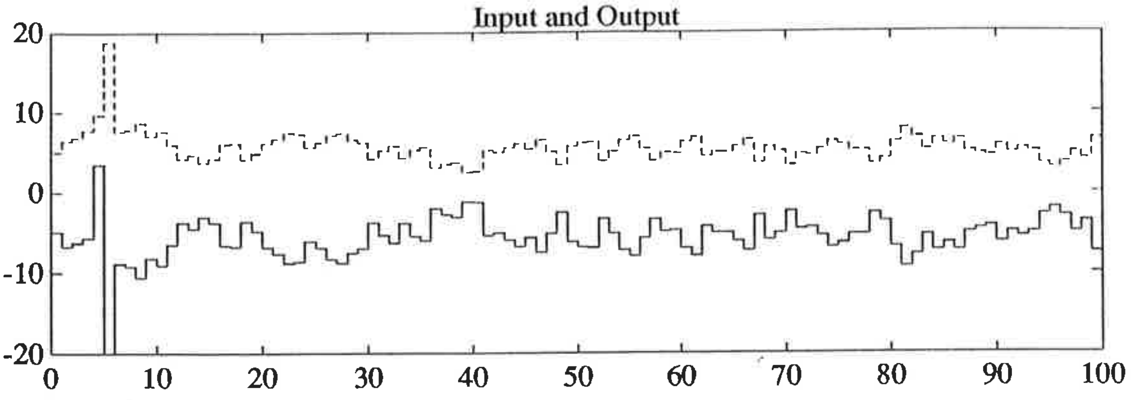




**Figure 2.** Input and output measurements as well as parameter estimates for system of Table 2.

| Simulation # 2 | | | | | | |
|---|---|---|---|---|---|---|
| Parameter | True Value | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 |
| $a_1$ | $-1.607$ | $-1.623$ | $-1.629$ | $-1.597$ | $-1.890$ | $-1.651$ |
| $a_2$ | 0.610 | 0.505 | 0.616 | 0.616 | 0.544 | 0.664 |
| $b_1$ | 0.107 | 0.110 | 0.108 | 0.107 | 0.132 | 0.107 |
| $b_2$ | 0.090 | 0.096 | 0.091 | 0.090 | 0.107 | 0.894 |

**Table 2.** Parameter estimates for second order stable system controlled using a certainty equivalence indirect adaptive controller using a minimum variance regulator.



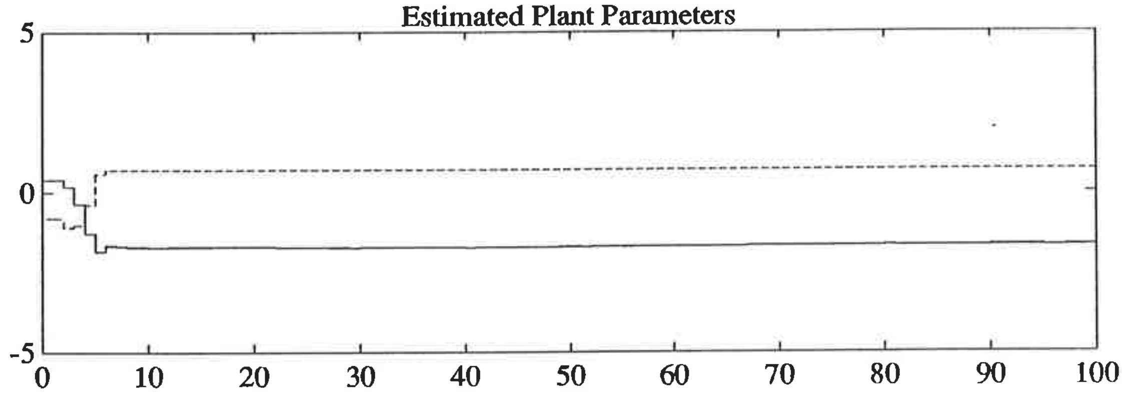
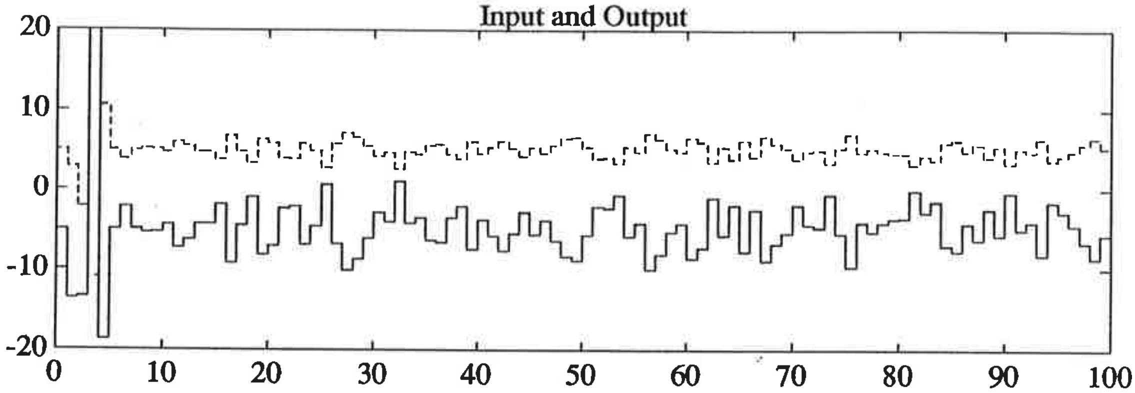




**Figure 3.** Input and output measurements as well as parameter estimates for system of Table 3.

| Simulation # 3 | | | | | | |
|---|---|---|---|---|---|---|
| Parameter | True Value | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 |
| $a_1$ | $-1.649$ | $-1.546$ | $-1.678$ | $-1.678$ | $-1.629$ | $-1.734$ |
| $b_1$ | 0.649 | 0.609 | 0.653 | 0.659 | 0.616 | 0.695 |

**Table 3.** Parameter estimates for first order unstable system controlled using a certainty equivalence indirect adaptive controller using pole placement regulator.

37.



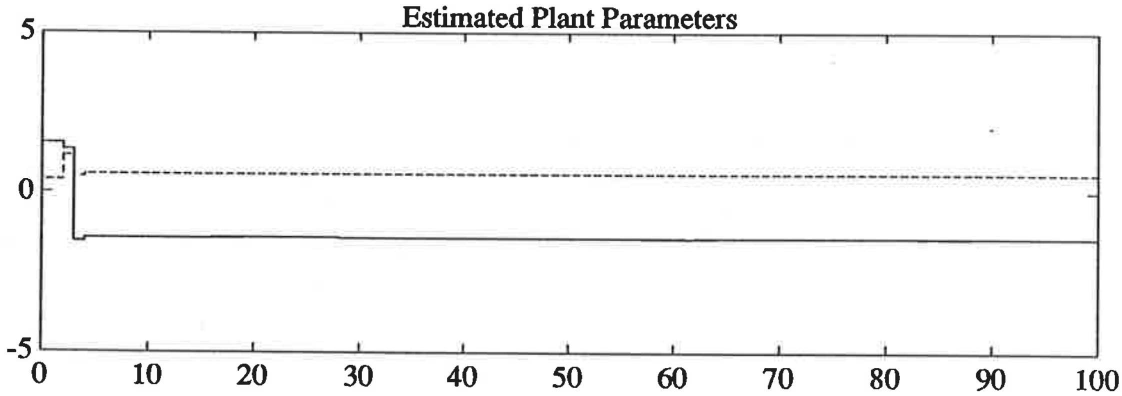




**Figure 4.** Input and output measurements as well as parameter estimates for system of Table 4.

| Simulation # 4 | | | | | | |
|---|---|---|---|---|---|---|
| Parameter | True Value | Run 1 | Run 2 | Run 3 | Run 4 | Run 5 |
| $a_1$ | −1.649 | −1.705 | −1.470 | −1.826 | −1.604 | −1.619 |
| $b_1$ | 0.649 | 0.697 | 0.589 | 0.707 | 0.642 | 0.612 |

**Table 4.** Parameter estimates for first order unstable system controlled using a certainty equivalence indirect adaptive controller using a minimum variance regulator.

38.

# On the use of robust controllers in adaptive control

Pablo A. Iglesias‡
Division of Information Engineering
Cambridge University
Cambridge, CP2 1PZ
England

**Abstract**

Using concepts from non–adaptive robust control, the stability and robustness of a general indirect adaptive control algorithm is analyzed. It is shown that the use of robust controllers, particularly those obtained from $\mathcal{H}^\infty$ optimization techniques, can improve local robustness as long as the controller parameters are sufficiently smooth functions of the estimated parameters.

## 1. Introduction

Ever since the work of Rohrs and co–workers [19], much of the effort of adaptive control research has been centered on the design of robust adaptive controllers. A quick glance at the results available in this field shows that almost all of the algorithms presented achieve their robustness through modifications of the parameter identification scheme, see [11], [14], [17] for examples. During the same period, considerable progress has been made in the analysis and design of non–adaptive, robust controllers. In particular, robust controllers designed on the basis of $\mathcal{H}^\infty$ and $l_1$ optimization techniques have been developed, eg. [4], [9]. It then seems reasonable to study the possible improvements to the robustness of adaptive controllers that could be obtained through the use of one of these robust controllers in adaptive control.

In this paper we examine this question for a discrete–time, indirect adaptive controller. In robust control papers, the plant is assumed to consist of a nominal plant $G_o(z)$, as well as a separate transfer function $\Delta_G(z)$ used to represent plant uncertainty. Two common approaches are to model these uncertainties as additive or multiplicative perturbations on $G_o(z)$; see [6] for a discussion. An alternative expression for the plant uncertainty; where $\Delta_G$ represents additive perturbations to the stable coprime factors of the nominal plant, will be used. This type of expression for model uncertainty has been advocated by Vidyasagar, see [20], where it is shown to have some advantages over other approaches. For example, it allows the number of unstable poles to vary as the plant is perturbed. Glover and McFarlane show in [9], [10], that the design of $\mathcal{H}^\infty$–optimal controllers for this problem is surprisingly explicit, suggesting possible applications to adaptive control.

‡ This work has been supported by NSERC Canada under a 1967 Postgraduate Research Scholarship.

The robustness to stable factor perturbations of a general time–varying system is considered in [15]. This analysis, however, does not include the identification scheme, and so does not truly represent an adaptive controller. A model reference adaptive controller subject to stable factor perturbations of known magnitude is considered by Krause *et al.* [13]; where the robustness of the scheme is achieved by using the known magnitude to introduce a dead zone in the identification scheme.

The results in this paper can be considered as an extension of the work of [15], in that the analysis of the identification algorithm will be included. This will be done using linearization and total stability theory as in Anderson *et al.* [1]. A similar analysis for a pole placement scheme is considered in Phillips *et al.* [16]. It will be shown that provided the mapping from estimated parameters to controller transfer functions is a sufficiently smooth function, then the use of this design techniques will improve the robustness of adaptive controllers.

Section 2 gives some preliminary facts and definitions that will be used throughout the rest of the paper. Section 3 describes the type of systems and of controllers considered. In Section 4 the local stability analysis of the adaptive controller is carried out, and in Section 5 the implications of using robust controllers adaptively, in particular, the robust controller of [10] are discussed. The results are finally summarized in Section 6.

## 2. Preliminaries

In the sequel, $q$ will represent the unit delay operator $qx(k) = x(k+1)$. A causal, linear, time–invariant system will be represented by its transfer function $G(z)$, where the argument $z$ represents the usual $\mathcal{Z}$–transform. The transfer function $G(z)$ is stable if all its poles are in $|z| < 1$. Consequently, every stable transfer function $G$ can be expressed as $G(z) = \sum_{k=0}^{\infty} g_k z^{-k}$. Moreover, if $G$ is strictly causal, then $g_0 = 0$.

We now define some norms which will be used throughout, see [5] for a further discussion. Let $x \in \mathcal{R}^n$; then $\|x\|_2 := \left(\sum_{i=1}^{n} |x_i|^2\right)^{1/2}$, and $\|x\|_\infty := \sup_{1\le i\le n} |x_i|$. If $A \in \mathcal{R}^{m\times n}$, the induced matrix norms will be denoted by $\|A\|_{i2} = \max_{1\le i\le n} \sigma_i(A)$ and $\|A\|_{i\infty} = \max_{1\le i\le m} \sum_{j=1}^{n} |a_{ij}|$ respectively.

A sequence of vectors $\{x(k) : k \ge 0\}$, is said to belong to $l^2$, and $l^\infty$ if they satisfy: $\sum_{k=0}^{\infty} \|x(k)\|_2^2 < \infty$ and $\sup_{k\ge 0} \|x(k)\|_\infty < \infty$ respectively. The norms on $l^2$ and $l^\infty$ will also be denoted by $\|\cdot\|_2$ and $\|\cdot\|_\infty$. There should be no ambiguity with the vector norms, since any vector $x$ can be considered as a sequence $\{x(k) : x(0) = x,\ x(k) = 0, \forall k > 0\}$.

A system with transfer function $G$ is said to be $l^2$–stable if it maps $l^2$ signals into $l^2$. If $G$ is time–invariant, this is equivalent to requiring

$$\|G\|_\infty := \sup_{0\le\omega\le 2\pi} \|G(e^{i\omega})\|_{i2} < \infty.$$

$l^\infty$–stability is defined in a similar manner, and is equivalent to requiring that

$$\|G\|_1 := \sum_{k=1}^{\infty} \|g_k\|_{i\infty} < \infty.$$

We note that $\|G\|_\infty \le \|G\|_1$. Moreover, for time–invariant, finite–dimensional systems, the following relation between the two norms exists.

**Lemma 1** (*Boyd and Doyle* [3])
*If $G(z)$ has a state-space representation of order $n$, then*

$$\|G\|_1 \leq (2n+1)\|G\|_\infty.$$

*If in addition, $G(z)$ is strictly causal, then*

$$\|G\|_1 \leq 2n\|G\|_\infty.$$

□

Finally, if $G$ is time-varying, and its input–output relation can be written as $y(k) = \sum_{j=0}^{k} g(k,j)u(j)$, then the system is $l^\infty$ stable iff

$$\sup_{k\geq 0}\{\sum_{j=0}^{k} \|g(k,j)\|_{i\infty}\} =: \|G\|_{S1} < \infty.$$

It is easy to see that if $G$ is time-invariant, then $\|G\|_{S1} = \|G\|_1$.

# 3. Plant and Controller Structure

This section presents a description of both the system and controller that are to be considered.

## 3.1 Nominal Plant

In order to control the true system, a model of the plant must be available to the designer. For this reason, the input–output behavior of the plant is *approximated* by the model

$$y(k) = G_0(q)u(k). \tag{1}$$

It is assumed that the transfer function $G_0(z)$ can be described by the strictly causal relation

$$G_0(z) = \frac{b_1 z^{n-1} + \ldots + b_n}{z^n + a_1 z^{n-1} + \ldots + a_n} =: \frac{z^n B(z)}{z^n A(z)} \tag{2}$$

where $B(z)$ and $A(z)$ are coprime polynomials in $z^{-1}$. This notation is used so that $A(z)$ and $B(z)$ are both stable transfer functions. The input–output behavior (1) can be expressed as a linear combination of the plant parameters and of filtered inputs and outputs in the usual manner:

$$y(k) = \phi(k)^T \theta_0$$

where

$$\phi(k) := [\, u(k-1) \quad \ldots \quad u(k-n) \quad -y(k-1) \quad \ldots - y(k-n)\,],$$

$$\theta_0 := [\, b_1 \quad \ldots \quad b_n \quad a_1 \quad \ldots \quad a_n \,].$$

In practice the value of $\theta_0$ is not known; the vector $\theta$ will represent an estimate of the plant. Associated with each $\theta$ is the transfer function

$$G(z,\theta) := \frac{B(z,\theta)}{A(z,\theta)},$$

formed by replacing $\theta_0$ with $\theta$ in the obvious manner.

It will be useful to consider coprime factorizations of $G$ over the set of stable transfer functions. Two stable functions $A(z)$ and $B(z)$ are coprime if there exists stable functions $X(z)$ and $Y(z)$ satisfying the Bezout identity

$$A(z)X(z) + B(z)Y(z) = 1.$$

**Definition 1**

*Let $G(z)$ be a transfer function. The pair $[N(z), M(z)]$ is a* coprime factorization *of $G(z)$ if the following three conditions hold:*

*(1) $M(z) \not\equiv 0$.*

*(2) $N(z)$, $M(z)$ are stable, coprime* transfer functions.

*(3) $G(z) = N(z)/M(z)$.*

□

For any given transfer function $G$, the choice of $N$ and $M$ is not unique. For example, since the polynomials in (2) are coprime, then $[B(z), A(z)]$ is a valid coprime factorization. So is any pair $[B(z)/\xi(z), A(z)/\xi(z)]$ where $\xi(z)$ is any polynomial in $z^{-1}$ with no zeros in $|z| \geq 1$. In the sequel it is assumed that such a $\xi(z)$ has been chosen so that $N_\theta$ and $M_\theta$ are unique functions of $\theta$. Note that $\xi$ may also depend on $\theta$. For extensions of Definition 1 the reader is referred to [20].

### 3.2 True System

Even if $\theta_0$ were known, it can not be assumed that the transfer function $G_0(z)$ models the true system exactly, since $G_0$ is only a low order approximation of the system. In the sequel it will be assumed that the true system is linear, causal, time–invariant, and that it can be described by the transfer function

$$G(z) = \frac{N_0(z) + \Delta_N(z)}{M_0(z) + \Delta_M(z)},$$

where the pair $[N_0 + \Delta_N, M_0 + \Delta_M]$ is a coprime factorization of $G$. The perturbation functions $\Delta_N$ and $\Delta_M$ are not known, and can be of arbitrarily high degree. Nevertheless, by property (2) of Definition 1, and the fact that the set of stable transfer functions forms a ring, then both $\Delta_N$ and $\Delta_M$ are stable. This does not imply, however, that $G$ and $G_0$ share the same number of unstable poles.

### 3.3 Controller

In this section the type of controllers to be used are described. With each parameter vector $\theta$, associate a controller with transfer function $K(z,\theta)$, such that $K(z,\theta)$ stabilizes $G(z,\theta)$. *Stabilizes* means that the closed–loop system is internally stable: ie. the transfer functions $S(z,\theta)$, $K(z,\theta)S(z,\theta)$, and $G(z,\theta)S(z,\theta)$ are all stable, where $S(z,\theta) := (1 - G(z,\theta)K(z,\theta))^{-1}$.

Associated with each $K(z,\theta)$, is the coprime factorization $[U(z,\theta), V(z,\theta)]$. There is an unlimited number of possible factorizations. In the sequel, $[U(z,\theta), V(z,\theta)]$, will be required to satisfy the Bezout identity

$$M(z,\theta)V(z,\theta) - N(z,\theta)U(z,\theta) = 1 \tag{3}$$

for all $z$. The control input $u(k)$ is given by

$$V(q)u(k) = U(q)y(k) + W(q)r'(k).$$

In this paper, the design and robustness of the feedforward transfer function $W$ will not be discussed, so $r := Wr'$ will be regarded as the external reference input. The solution to equation (3) is not unique. Nevertheless the choice of $U$, and $V$ can be specified uniquely. The following two assumptions will be made.

**Assumption A1**

*There exists a non empty, open region $\Theta \ni \theta_0$, and a mapping $\mathcal{K} : \theta \to [U(z,\theta), V(z,\theta)]$ such that $\forall \epsilon > 0, \exists \delta > 0$, so that if $\theta_1, \theta_2 \in \Theta$,*

$$\|\theta_1 - \theta_2\|_2 < \delta \Longrightarrow \|[U(z,\theta_1) - U(z,\theta_2), V(z,\theta_1) - V(z,\theta_2)]\|_\infty < \epsilon.$$

□

**Assumption A2**

*$K(z,\theta_0) = U(z,\theta_0)V(z,\theta_0)^{-1}$ stabilizes the true system $G(z)$.* □

Assumption A1 says that the controller is continuous in the *graph metric*, see [20]. Assumption A2 guarantees that the problem is well posed. It implies that the nominal plant models the true system sufficiently well so that the latter can be controlled. This assumption is implicit in any control system, whether adaptive or not. The following lemma presents a sufficient condition for A2 to be satisfied. For notational simplicity, let $K_0 := K(z,\theta_0)$ and $S_0 := S(z,\theta_0)$.

**Lemma 2** (*Vidyasagar* [20])

*Assumption A2 holds provided that*

$$\left\| \begin{bmatrix} S_o M_o^{-1} \\ K_o S_o M_o^{-1} \end{bmatrix} [\Delta_M \ \Delta_N] \right\|_\infty < 1. \tag{4}$$

□

This lemma motivates the design of an $\mathcal{H}^\infty$ optimal robust controller: a controller which, while requiring that $K_0$ stabilize $G_0$, minimizes the expression

$$\left\| \begin{bmatrix} S_o M_o^{-1} \\ K_o S_o M_o^{-1} \end{bmatrix} \right\|_\infty .$$

The solution of this problem for the continuous–time case is derived by Glover and McFarlane in [9], and [10] where the coprime–factorization is specified to be *normalized*; that is:

$$N_0(z)N_0(z^{-1}) + M_0(z)M_0(z^{-1}) = 1.$$

This normalization allows one to bypass the expensive iterative techniques that are normally associated with $\mathcal{H}^\infty$ controllers; see [7] and [8]. A state–space description of the controller can be given explicitly in terms of the system's state–space matrices, and the solution to two uncoupled Riccati equations. More of this will be said in Section 5, where it will be shown that this controller has some other properties beneficial to adaptive controllers. Note that requiring that $[N_0, M_0]$ be normalized is the same as requiring $\xi$ to be the spectral factor: $A(z)A(z^{-1}) + B(z)B(z^{-1}) = \xi(z)\xi(z^{-1})$.

In an adaptive control setting, $\theta_0$ is not known, so $K_0$ can not be implemented directly. Instead, the *certainty equivalence* controller is used. That is, at each time $k$, if the plant estimate is $\theta(k)$, then $K(\theta(k))$ is applied to the system. In order to consider the robustness of the adaptive control algorithm in a unified fashion, assume that the

43.

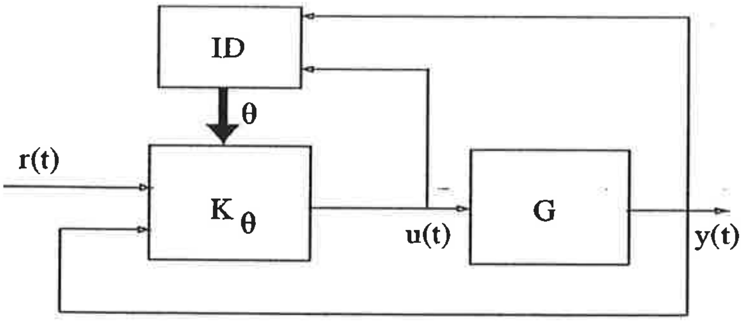



**Figure 1.** General adaptive controller.

difference between the nominal controller, and the frozen parameter controller $K(z,\theta)$; the controller obtained by freezing the parameter estimates at $\theta$, can be represented as

$$K(z,\theta) = \frac{U_0(z) + \Delta_U(z,\theta)}{V_0(z) + \Delta_V(z,\theta)},$$

where $[U_0 + \Delta_U(z,\theta), V_0 + \Delta_V(z,\theta)]$ is a coprime factorization of $K(z,\theta)$. Although the perturbation functions $\Delta_U(z,\theta)$ and $\Delta_V(z,\theta)$ are stable for each time $k$, the time-varying operators need not be stable. Nevertheless, if the parameter estimates vary sufficiently slowly, closed loop stability will be retained, see eg. [5], page 147.

## 4. Local Analysis

The adaptive control problem can be depicted as in Figure 1. The plant is time-invariant. The only time variations in the closed loop system are due to the changing controller, which varies according to the estimated plant parameters. To keep the discussion as simple as possible, assume that the plant estimates change according to a projection identification scheme. That is,

$$\theta(k+1) = \theta(k) + \gamma\frac{\phi(k)e(k,\theta)}{1+\gamma\phi(k)^T\phi(k)}, \tag{5}$$

where $e(k,\theta)$ is the prediction error, defined by

$$e(k,\theta) := y(k) - \phi(k)^T\theta(k).$$

The constant $\gamma > 0$ is the ***step-size*** or ***gain*** of the identification algorithm. Since the system is operating in closed loop, $y(k)$ and $u(k)$ depend implicitly on the estimated parameter $\theta(k)$. In the sequel this dependence will be made explicit.

In order to analyze how the model and controller uncertainties influence $y$ and $u$, begin by reconfiguring the system as in Figure 2, according to the definitions of Section 2. Using the linearity, and the time invariance of $G_0$ and $K_0$, the input–output behavior can be written as

$$x(k,\theta) = \mathcal{P}_G r(k) + \mathcal{T}_G\Delta_G x(k,\theta) + \mathcal{T}_K v(k,\theta(k)) \tag{6}$$

where the following abbreviations have been introduced:

$$x(k,\theta) := \begin{bmatrix} y(k,\theta) \\ u(k,\theta) \end{bmatrix}, \qquad \mathcal{P}_G := \begin{bmatrix} G_0S_0V_0^{-1} \\ S_0V_0^{-1} \end{bmatrix}$$

44.

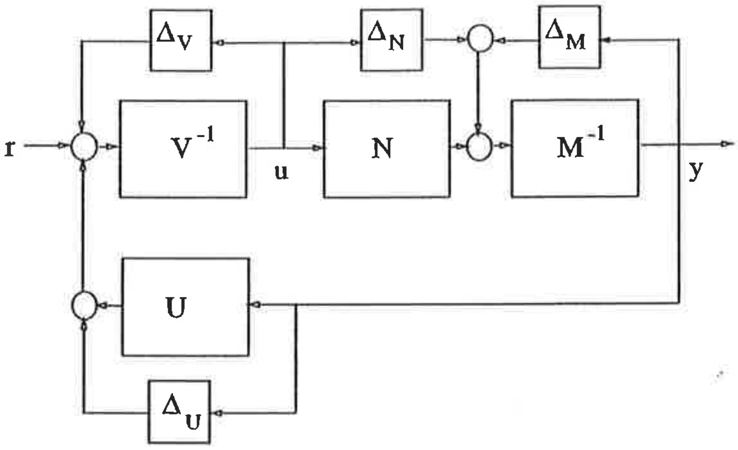



**Figure 2.** Adaptive control system with nominal plant and controller stable factors perturbations.

$$\mathcal{T}_G := \begin{bmatrix} S_o M_o^{-1} \\ K_o S_o M_o^{-1} \end{bmatrix} = \begin{bmatrix} V_0 \\ U_0 \end{bmatrix}, \tag{7}$$

$$\mathcal{T}_K := \begin{bmatrix} G_o S_o V_o^{-1} \\ S_o V_o^{-1} \end{bmatrix} = \begin{bmatrix} N_0 \\ M_0 \end{bmatrix}, \tag{8}$$

$$\Delta_G := [\Delta_M \ \Delta_N],$$

and

$$\begin{aligned} v(k,\theta(k)) &:= \left( [\Delta_{V(\theta)} \ \Delta_{U(\theta)}] \begin{bmatrix} y \\ u \end{bmatrix} \right)(k) \\ &=: \sum_{j=0}^{k} \delta(k,j) x(j,\theta). \end{aligned}$$

Equalities (7) and (8) have been obtained from the Bezout identity (3). Collecting the $x$ terms on the left side, and using equation (4) which guarantees that $I - \mathcal{T}_G\Delta_G$ is invertible, we can divide both sides by $I - \mathcal{T}_G\Delta_G$. Using the matrix identity $(I-X)^{-1} = I + X(I-X)^{-1}$, equation (6) becomes

$$\begin{aligned} x(k,\theta) &= \mathcal{P}_G r(k) + \mathcal{T}_G\Delta_G(I - \mathcal{T}_G\Delta_G)^{-1}\mathcal{P}_G r(k) \\ &\quad + (I - \mathcal{T}_G\Delta_G)^{-1}\mathcal{T}_K v(k,\theta(k)) \\ &=: x_0(k) + \tilde{x}(k) + \bar{x}(k,\theta). \end{aligned} \tag{9}$$

Since $x_0$ depends only on the nominal plant and controller, it can be regarded as the ideal system input and output. In contrast $\tilde{x}(k)$ corresponds to input and output perturbations that are introduced by the existance of unmodelled dynamics. These perturbations are guaranteed to remain bounded by (4). Finally, the vector $\bar{x}(k,\theta)$ represents perturbations that are due to the time–varying, incorrectly identified plant. These are not guaranteed to remain stable from previous assumptions. Note also that any bound on the size of $\bar{x}(k,\theta)$ will be magnified by the presence of unmodelled dynamics. This can be seen, from the appearance of the $(I - \mathcal{T}_G\Delta_G)^{-1}$ term in front of $\mathcal{T}_K v(k,\theta(k))$ in (9). The following lemma gives a sufficient condition under which $\bar{x}(k,\theta)$ will remain stable.

**Lemma 3**

*The system from $r$ to $[y,u]$ will be $l^\infty$ stable provided that the time–varying perturbations*

$\Delta_{K(\theta)} := [\Delta_{V(\theta)}\ \Delta_{U(\theta)}]$ *are stable, and satisfy*

$$\frac{\|\Delta_{K(\theta)}\|_{S1}\|\mathcal{T}_K\|_1}{1-\|\mathcal{T}_G\Delta_G\|_1} \le \mu < 1. \tag{10}$$

*Proof:* From (9):

$$\begin{aligned}\|x(k,\theta)\|_\infty &\le \|x_0+\tilde{x}\|_\infty + \|(I-\mathcal{T}_G\Delta_G)^{-1}\mathcal{T}_K v\|_\infty \\ &\le \|x_0+\tilde{x}\|_\infty + \|(I-\mathcal{T}_G\Delta_G)^{-1}\mathcal{T}_K\|_1\|v\|_\infty \\ &\le \|x_0+\tilde{x}\|_\infty + (1-\|\mathcal{T}_G\Delta_G\|_1)^{-1}\|\mathcal{T}_K\|_1\|\Delta_{K(\theta)}\|_{S1}\|x(k,\theta)\|_\infty.\end{aligned}$$

Collecting the $\|x(k,\theta)\|_\infty$ terms on the left, gives

$$\left(1-\frac{\|\Delta_{K(\theta)}\|_{S1}\|\mathcal{T}_K\|_1}{1-\|\mathcal{T}_G\Delta_G\|_1}\right)\|x(k,\theta)\|_\infty \le \|x_0+\tilde{x}\|_\infty$$

which, if (10) holds, and since $x_0$ and $\tilde{x}$ are bounded implies $\|x(\cdot,\theta)\|_\infty \le (1-\mu)^{-1}\times \|x_0+\tilde{x}\|_\infty < \infty$ as required.

□

Lemma 3 provides a simple, albeit conservative condition for the stability of the overall time–varying system. Consider the identification algorithm (5). Note that the regression vector $\phi(k,\theta)$ is obtained by a linear filtering operation on $x(k,\theta)$, say $\phi = \Gamma x$. Then $\phi(k,\theta)$ can be divided into the sum of three components $\phi_0$, $\tilde{\phi}$ and $\bar{\phi}(k,\theta)$ according to (9). Similarly, the prediction error $e(k,\theta)$ can be separated as follows:

$$\begin{aligned}e(k,\theta) &= y(k,\theta) - \phi(k,\theta)^T\theta(k) \\ &= y_o(k) - \phi(k,\theta)^T\theta(k) + \tilde{y}(k) + \bar{y}(k,\theta) \\ &= -\phi(k,\theta)^T(\theta(k)-\theta_0) + \tilde{y}(k) - \tilde{\phi}(k)^T\theta_0 + \bar{y}(k,\theta) - \bar{\phi}(k,\theta)^T\theta_0 \\ &=: -\phi(k,\theta)^T\tilde{\theta}(k) + \tilde{e}(k) + \bar{e}(k,\theta)\end{aligned} \tag{11}$$

where the third equality was obtained using the fact that $y_o(k) = \phi_o(k)^T\theta_0$. The expressions for $\bar{e}(k,\theta)$ and $\tilde{e}(k)$ can be evaluated explicitly in terms of the unmodelled dynamics $\Delta_G$ and controller perturbations $\Delta_{K(\theta)}$. Using the fact that $[M_0\ -N_0]\mathcal{T}_G = 1$ and $[M_0\ -N_0]\mathcal{T}_K = 0$ gives the following expressions:

$$\begin{aligned}\tilde{e}(k) &= \tilde{y}(k) - \theta_0^T\tilde{\phi}(k) = A_0\tilde{y}(k) - B_0\tilde{u}(k) \\ &= \xi\,[M_0\ -N_0]\tilde{x}(k) \\ &= \xi\,[M_0\ -N_0]\begin{bmatrix} S_oM_o^{-1} \\ K_oS_oM_o^{-1}\end{bmatrix}\Delta_G(I-\mathcal{T}_G\Delta_G)^{-1}x_0(k) \\ &= \xi\,M_0[I\ -G_0]\begin{bmatrix} I \\ K_0\end{bmatrix} S_0M_0^{-1}\Delta_G(I-\mathcal{T}_G\Delta_G)^{-1}x_0(k) \\ &= \xi\,\Delta_G(I-\Delta_G\mathcal{T}_G)^{-1}x_0(k)\end{aligned} \tag{12}$$

and similarly,

$$\begin{aligned}\bar{e}(k,\theta) &= \bar{y}(k,\theta) - \theta_0^T\bar{\phi}(k,\theta) = A_0\bar{y}(k,\theta) - B_0\bar{u}(k,\theta) \\ &= \xi\,[M_0\ -N_0](I-\mathcal{T}_G\Delta_G)^{-1}\mathcal{T}_K(\Delta_{K(\theta)}x)(k) \\ &= \xi\,[M_0\ -N_0]\{I+\mathcal{T}_G\Delta_G(I-\mathcal{T}_G\Delta_G)^{-1}\} \\ &\quad\times \mathcal{T}_K(\Delta_{K(\theta)}x)(k) \\ &= \xi\,\Delta_G(I-\mathcal{T}_G\Delta_G)^{-1}\mathcal{T}_K(\Delta_{K(\theta)}x)(k).\end{aligned} \tag{13}$$

46.

Replacing $e(k,\theta)$ in (5), with equation (11) gives

$$\tilde{\theta}(k+1) = \left(I - \gamma\frac{\phi(k,\theta)\phi(k,\theta)^T}{1+\gamma\phi(k,\theta)^T\phi(k,\theta)}\right)\tilde{\theta}(k) + \frac{\gamma\phi(k,\theta)(\tilde{e}(k)+\bar{e}(k,\theta))}{1+\gamma\phi(k,\theta)^T\phi(k,\theta)}. \tag{14}$$

Equation (14) is a non-linear difference equation. Notice that $\bar{e}(k,\theta) \neq 0$ implies that $\theta = \theta_0$ will no longer be a fixed point of (14). To analyze the stability properties of (14) proceed as follows. Freeze the time variations at some parameter $\bar{\theta}$. This means that the time-varying operator $\Delta_{K(\theta)}$ becomes the time-invariant transfer function $\Delta_{K(\bar{\theta})}$. Replacing this equation in (14) and linearizing about the nominal model $\theta_0$, yields the following equation

$$\tilde{\bar{\theta}}(k+1) = \Lambda(k)\tilde{\bar{\theta}}(k) + f(k)\tilde{\bar{\theta}}(k) + g(k) + O(\|\tilde{\bar{\theta}}\|_\infty^2) \tag{15}$$

where

$$\begin{aligned}
\Lambda(k) &:= I - \gamma\phi(k,\theta_0)\phi(k,\theta_0)^T/d(k),\\
f(k) &:= \gamma\left(\phi(k,\bar{\theta}_0)\frac{\partial\bar{e}(k,\bar{\theta})}{\partial\bar{\theta}}|_{\bar{\theta}=\theta_0} + \tilde{e}(k)\frac{\partial\bar{\phi}(k,\theta)}{\partial\bar{\theta}}|_{\theta=\theta_0}\right)/d(k)\\
&\quad - \gamma^2\left(\tilde{e}(k)\phi(k,\theta_0)\phi(k,\theta_0)^T\frac{\partial\bar{\phi}(k,\theta)}{\partial\bar{\theta}}|_{\bar{\theta}=\theta_0}\right)/d(k)^2,\\
g(k) &:= \gamma\phi(k,\theta_0)\tilde{e}(k)/d(k),\\
d(k) &:= 1+\gamma\phi(k,\theta_o)^T\phi(k,\theta_o).
\end{aligned}$$

The first analysis of this type was carried out for Rohrs' counterexamples, see [2]. Since then considerable work has been done rigorizing the validity of the steps above see [1], and [18]. The value of the analysis in this paper is that the bounds of the magnitudes of $f$ and $g$ can be expressed in terms of the transfer function norms of of $\mathcal{T}_G$, $\mathcal{T}_K$, etc.. For notational simplicity, all $\|\cdot\|$ will refer to the induced $l^\infty$ norm, and the time dependence of all functions will be omitted. Let $\|\Delta_G\mathcal{T}_G\| \leq \|\Delta_G\|\|\mathcal{T}_G\| \leq (\epsilon)\ (1/\epsilon_0)$, and $\|\mathcal{T}_K\| \leq 1/\epsilon_{K_0}$. Then

$$\begin{aligned}
\|f\| &\leq \gamma^{\frac{1}{2}}\left\|\frac{\gamma^{\frac{1}{2}}\phi(\theta_0)}{1+\gamma\phi(\theta_o)^T\phi(\theta_o)}\right\|\left\|\frac{\partial\bar{e}(\bar{\theta})}{\partial\bar{\theta}}|_{\bar{\theta}=\theta_0}\right\| + 2\gamma\|\tilde{e}(k)\|\left\|\frac{\partial\bar{\phi}(\bar{\theta})}{\partial\bar{\theta}}|_{\bar{\theta}=\theta_0}\right\|\\
&\leq \frac{1}{2}\gamma^{\frac{1}{2}}\frac{\epsilon/\epsilon_{K_0}}{(1-\epsilon/\epsilon_o)^2}\|\xi\|\left\|\frac{\partial\Delta_{K(\bar{\theta})}}{\partial\bar{\theta}}|_{\bar{\theta}=\theta_0}\right\|\|x_o\|\\
&\quad + 2\gamma\frac{\epsilon/\epsilon_{K_0}}{(1-\epsilon/\epsilon_o)^3}\|\xi\|\ \|\Gamma\|\left\|\frac{\partial\Delta_{K(\bar{\theta})}}{\partial\bar{\theta}}|_{\bar{\theta}=\theta_0}\right\|\|x_o\|^2\\
&= \frac{1}{2}\gamma^{\frac{1}{2}}\frac{\epsilon/\epsilon_{K_0}{}^2}{(1-\epsilon/\epsilon_o)^2}\|\xi\|\left\|\frac{\partial\Delta_{K(\bar{\theta})}}{\partial\bar{\theta}}|_{\bar{\theta}=\theta_0}\right\|\|r\|\left\{1+\frac{4\gamma^{\frac{1}{2}}}{\epsilon_{K_0}(1-\epsilon/\epsilon_o)}\|\Gamma\|\ \|r\|\right\}
\end{aligned}$$

and

$$\|g\| \leq \frac{1}{2}\gamma^{\frac{1}{2}}\frac{\epsilon/\epsilon_{K_0}}{1-\epsilon/\epsilon_o}\|\xi\|\ \|r\|.$$

These inequalities yield the following local stability theorem for the difference equation (15).

**Theorem 1**

*Consider the difference equation*

$$\tilde{\bar{\theta}}(k+1) = \Lambda(k)\tilde{\bar{\theta}}(k) + f(k)\tilde{\bar{\theta}}(k) + g(k) \tag{16}$$

47.

*Suppose that there exists a ball $B(\theta_0, r) \subset \Theta$ such that: Assumption A1; equation (10); and $\left\|\frac{\partial \Delta_{K(\theta)}}{\partial \theta}\right\| < c$ hold for all $\bar{\theta} \in B(\theta_0, r)$. If $\exists T$, $\alpha$, such that*

$$0 < \alpha I \leq \sum_{i=0}^{T-1} \phi(k+i, \theta_0)\phi(k+i, \theta_0)^T < \infty \tag{17}$$

*for all $k \geq 0$, then $\exists \epsilon^*$, $C$, such that $\forall \epsilon$, $0 < \epsilon < \epsilon^*$ and $\|\tilde{\bar{\theta}}(0)\| < \frac{r}{C}$, the solution to (16) will remain in $B(\theta_0, r)$, $\forall k \geq 0$.*

*Proof:* Equation (17) guarantees that the linear time-varying system

$$\tilde{\bar{\theta}}(k+1) = \Lambda(k)\tilde{\bar{\theta}}(k)$$

is exponentially stable. Thus, there exists constants $0 \leq a < 1$ and $C > 0$ such the state transition matrix $F(k, 0)$ satisfies $\|F(k, 0)\| \leq Ca^k$. Let $\|\tilde{\bar{\theta}}(0)\| < r/C$ and choose $\epsilon^*$ small enough so that constants $\beta_1$ and $\beta_2$ satisfy $C(\beta_1 + \beta_2) + a < 1$ where $\|f\| \leq \beta_1$ and $\|g\| < \beta_2 r$. This choice is possible since $\|f\|$ and $\|g\|$ are both bounded by continuous, increasing functions of $\epsilon$ in $B(\theta_0, r)$, and equal to 0 for $\epsilon = 0$. It is then possible to use the the discrete-time total stability theorem of [1], page 26, to show that $\|\tilde{\bar{\theta}}(k)\| \leq r$, $\forall k \geq 0$. □

As long as assumption A1 is satisfied , the results of Theorem 1 will be valid, independent of the particular regulator design law that is implemented in the adaptive controller. The fact that a persistently exciting regression vector will give an identification scheme with a good measure of robustness has been well documented in the literature, see [1] and the references therein. Nevertheless Theorem 1 attempts to quantify the robustness attained by different regulators in terms of transfer function norms, and so gives guidelines for the design of "optimally robust" adaptive controllers.

In an adaptive control setting, the goal of a robust regulator must be to increase the region of attraction $B(\theta_0, r)$ as much as possible, while still maintaining the desired nominal system performance $x_0$.

A necessary condition arising for the local analysis of the theorem is that all $\tilde{\bar{\theta}} \in B(\theta_0, r)$ must provide a stabilizing frozen regulator parameter regulator $K_{\bar{\theta}}$. A sufficient condition for this is given in Lemma 3. Note that a less conservative condition can be obtained from

$$\left\| [\mathcal{T}_G \; \mathcal{T}_K] \begin{bmatrix} \Delta_G \\ \Delta_{K(\theta)} \end{bmatrix} \right\|_\infty < 1$$

This equation, as with Lemma 2 motivates the design of an $\mathcal{H}^\infty$ controller minimizing $\|[\mathcal{T}_G \; \mathcal{T}_K]\|_\infty$. This controller will make $\|\tilde{e}(k)\|_2$ and $\|\bar{e}(k, \bar{\theta})\|_2$ as small as possible for given $\xi$, $x_0$, $\Delta_G$ and $\Delta_{K(\theta)}$. The design of this regulator is considered in the next section.

## 5. Adaptive Robust Controllers

As explained in the previous section, a controller that minimizes

$$\|[\mathcal{T}_G \; \mathcal{T}_K]\|_\infty = \left\| \begin{bmatrix} S_0 M_0^{-1} & G_0 S_0 V_0^{-1} \\ K_0 S_0 M_0^{-1} & S_0 V_0^{-1} \end{bmatrix} \right\|_\infty$$

will improve the robustness of the adaptive control algorithm. The solution to this problem is simple if the coprime factorization is chosen to be normalized as in [9]. To ease notation the subscript "0" will be dropped in the following theorem.

**Theorem 2**

*Let $G = NM^{-1}$, where $[N, M]$ is a normalized coprime factorization of $G$ and $[U, V]$ be the set of all controller coprime factorizations given by identity (3). Then $K$ stabilizes $G$ and minimizes*

$$\left\| \begin{bmatrix} SM^{-1} \\ KSM^{-1} \end{bmatrix} \right\|_\infty \tag{18}$$

*iff $K$ stabilizes $G$ and minimizes*

$$\left\| \begin{bmatrix} SM^{-1} & GSV^{-1} \\ KSM^{-1} & SV^{-1} \end{bmatrix} \right\|_\infty . \tag{19}$$

*Moreover, the optimal values are $(1+\alpha^2)^{1/2}$, and $\{1+\alpha^2/2+\alpha\sqrt{1+\alpha^2/4}\}^{1/2}$, respectively, where $\alpha \geq 0$. The value of $\alpha$ is obtained in [9].*

*Proof:* Here we repeat the procedure of [9], which derives obtains the minimum of (18) to show that the optimal controllers of (18) and (19) are in fact the same. Begin by characterizing all controllers which achieve internal stability. If $X$ and $Y$ are stable transfer functions such that $MY - NX = 1$, then the set of all stabilizing controllers is given by

$$[U, V] = [X + MQ, Y + QN]$$

where $Q$ is any stable transfer function, furthermore, $MV - NU = 1$. Replacing $K$ in (19) with the set of controllers $[U, V]$ gives

$$\left\| \begin{bmatrix} Y + QN & N \\ X + QM & M \end{bmatrix} \right\|_\infty \tag{20}$$

The problem is then reduced to that of finding amongst all stable transfer functions $Q$, the one which minimizes (20). Since the infinity norm is invariant under multiplication of an inner matrix, premultiply the matrix in (20) by

$$\begin{bmatrix} N^* & M^* \\ M & -N \end{bmatrix}$$

where $M(z)^* := M(z^{-1})$. This gives

$$\left\| \begin{bmatrix} N^*Y + M^*X + Q & 1 \\ 1 & 0 \end{bmatrix} \right\|_\infty^2 = \frac{2+\alpha^2+\sqrt{\alpha^4+4\alpha^2}}{2} \tag{21}$$

where $\alpha = \|N^*Y + M^*X + Q\|_\infty$. Proceeding in the same fashion with equation (18), one gets that

$$\left\| \begin{bmatrix} SM^{-1} \\ KSM^{-1} \end{bmatrix} \right\|_\infty^2 = 1 + \alpha^2 \tag{22}$$

Since the equations in (21) and (22) are both monotonically increasing functions of $\alpha$, the optimal values for (18) and (19) are both obtained by minimizing $\alpha$. Thus the optimal $Q$, and hence the optimal $K$, is the same for both problems. The relation between the optimal values follows from (21) and (22). □

The question arises as to whether the controller obtained from Theorem 2 is more sensitive to parameter variations than other less robust regulators. Unfortunately the

49.

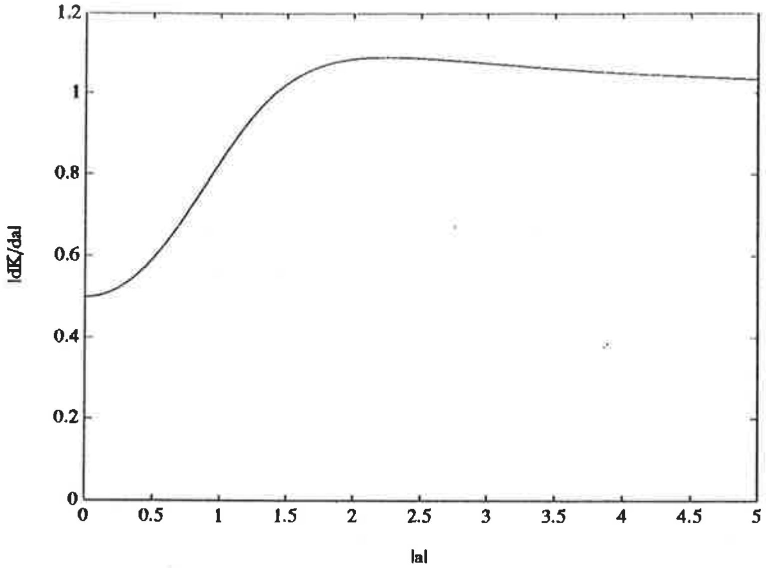



**Figure 3.** Plot of $\left|\frac{dk}{da}\right|$ over $|a|$ for the $\mathcal{H}^\infty$ controller of Example 1.

calculation of this is by no means trivial. The optimal controller of Theorem 2 is obatined from the solution of two Riccati equations. Obtaining values for the sensitivity of these Riccati equatins and hence the controller would give rather cumbersome formulae, see [12]. The following example should serve to illustrate this point.

**Example 1**

Consider a simple first order system with known gain and unknown pole: $G(z) = (z + a_1)^{-1}$. In this case $\theta = a$, and the certainty equivalence controller corresponding to the design law of Theorem 2 is a constant feedback of

$$k(a) = \begin{cases} -a - \frac{a^2+2-\sqrt{a^4+4}}{2a}, & a \neq 0 \\ 0, & a = 0. \end{cases}$$

Differentiating $k$ with respect to $a$ gives

$$\left|\frac{dk}{da}\right| = \begin{cases} \frac{a^4-4+(a^2+2)\sqrt{a^4+4}}{2a^2\sqrt{a^4+4}}, & a \neq 0 \\ 1/2, & a = 0 \end{cases}$$

Figure 3 shows the the graph of $\left|\frac{dk}{da}\right|$ versus $a$. For a pole placement algorithm, where the closed loop pole is placed at a constant location, this would be a constant 1. Thus, at small values of $a$, where the robustness margin is large, the $\mathcal{H}^\infty$ controller is less sensitive to parameter variations. At it's most sensitive point, the increase in sensitivy is less than 10%. Note that to be true to the analysis of Section 4, the factorization $k = u/v$ where $u$ and $v$ satisfy (3) should have been calculated, and then their respective derivatives computed. The equations obtained would be quite complicated and for this reason we have used the simpler formulas for $k$ and $\frac{dk}{da}$. □

50.

## 6. Conclusions

The local stability properties of an adaptive control algorithm for a discrete–time system subject to non–parametric stable factor perturbations have been analyzed. The stability analysis has been done using linearization and averaging techniques. It is shown that the use of specially designed robust regulators will improve the robustness of the overall adaptive controller. Note that the use of robust regulators will also improve the robustness of algorithms, such as those in [11], which ensure that $\theta$ approaches a point of closed loop stability and then turn the adaptation algorithm off. For the set of perturbations $\Delta_G$, the robust controller will provide as large a region as possible where the adaptation can be disconnected. The analysis, since it is based on transfer functions can be easily carried over to continuous time algorithms, as well as to multivariable plants.

### Acknoledgements

The author is indebted to K. Glover for suggesting this problem and for general help and encouragement. The research was partly carried out while the author was visiting the Department of Automatic Control, Lund Institute of Technology, Lund, Sweden. The hospitality of Professor K. J. Åström is gratefully acknowledged, as well as several constructive comments from B. Bernhardsson and K. Gustafsson concerning the manuscript.

51.

## References


[1] Anderson, B. D. O., R. R. Bitmead, C. R. Johnson, Jr., R. L. Kosut, P. V. Kokotović, I. M. Y. Mareels, L. Praly, and B. D. Riedle, (1986). *Stability of Adaptive Systems: Passivity and Averaging Analysis*, MIT Press, Cambridge, MA.

[2] Åström, K. J., (1983). "Analysis of Rohrs' counterexample to adaptive control," *Proc. 22nd IEEE Conf. on Decision and Control, San Antonio, TX*, 982–987.

[3] Boyd, S. P. and J. C. Doyle, (1987). "Comparison of peak and RMS gains for discrete–time systems," *Systems & Control Letters* **9**: 1–6.

[4] Dahleh, M. A. and J. B. Pearson, Jr., (1987). "$l^1$–optimal feedback controller for MIMO discrete–time systems," *IEEE Trans. Automatic Control* **AC–32**: 314–322.

[5] Desoer, C. A. and M. Vidyasagar, (1975). *Feedback Systems: Input–Output Properties*, Academic Press, New York, NY.

[6] Doyle, J. C. and G. Stein, (1981). "Multivariable feedback design: concepts for a classical/modern synthesis," *IEEE Trans. Automatic Control* **AC–26**: 4–16.

[7] Doyle, J. C., K. Glover, P. P. Khargonekar and B. A. Francis, (1989). "State–space solutions to standard $\mathcal{H}^2$ and $\mathcal{H}^\infty$ control problems," *submitted for publication.*

[8] Francis, B. A., (1987). *A course in $\mathcal{H}^\infty$ control theory*, volume 88 in *Lecture notes in control and information sciences*, Springer–Verlag, Berlin, FRG.

[9] Glover, K. and D. McFarlane, (1988). "Robust stabilization of normalized coprime factors: an explicit $\mathcal{H}^\infty$ solution," *Preprints ACC, Atlanta, Ga.*

[10] Glover, K. and D. McFarlane, (1988). "Robust stabilization of normalized coprime factor descriptions with $\mathcal{H}^\infty$–bounded uncertainty," *submitted for publication.*

[11] Hill, D. J., R. H. Middleton and G. C. Goodwin, (1986). "A class of robust adaptive control algorithms," *2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, Sweden*, 25–30.

[12] Kenney, C. and G. Hewer, (1987). "Sensitivity of algebraic Riccati equations," *Proc. IEEE 26th Conf. on Decision and Control, Los Angeles, CA*, 814–815.

[13] Krause, J., P. P. Khargonekar and G. Stein, (1988). "Robust parameter adjustment for model reference adaptive control," *submitted for publication.*

[14] Kreisselmeier, G. (1986). "A robust indirect adaptive control approach", *International Journal of Control,* **43**: 161–175.

[15] Ma, C. C. H. and M. Vidyasagar, (1987). "Parametric conditions for stability of reduced–order linear time–varying control systems," *Automatica,* **23**: 625–634.

[16] Phillips, S. M., R. L. Kosut and G. F. Franklin, (1988). "An averaging analysis of discrete–time indirect adaptive control," *Preprints ACC, Atlanta, Ga.*, 766–771.

[17] Praly, L., (1983). "Robustness of indirect adaptive control based on pole placement design," *1st IFAC Workshop on Adaptive Systems in Control and Signal Processing, San Francisco, CA.*

[18] Riedle, B. D. and P. V. Kokotović, (1986). "Stability bounds for slow adaptation: an integral manifold approach," *2nd IFAC Workshop on Adaptive Systems in Control and Signal Processing, Lund, Sweden*, 155–160.

[19] Rohrs, C., L. S. Valavani, M. Athans and G. Stein, (1985). "Robustness of continuous–time adaptive control algorithms in the presence of unmodeled dynamics," *IEEE Trans. Automatic Control,* **AC–30**: 881–889.

[20] Vidyasagar, M., (1985). *Control System Synthesis: A Factorization Approach*, MIT Press, Cambridge, MA.

# 4. Speciella tillämpningar av adaptiva system

53.

# An Adaptive Autopilot with Feedforward Compensation for Wave Disturbances applied to Ship Steering

Pär Kvist

Hendrik Ruijter

Department of Automatic Control

Lund Institute of Technology

April 1988

54.

# Contents

55.

# 1. Introduction

During design of autopilots for ships it has been noticed that it is not sufficient to use a fixed controller. The dynamics of a ship can vary considerable due to e.g. weather conditions, different load and velocity. In fact, under certain circumstances, not even stability can be guaranteed.

In this report an autopilot based on an adaptive controller is designed. Wave disturbances are suppressed with a feedforward compensation term in the controller. Simulations in the simulation language SIMNON are done. The report is structured as follows. In Chapter 2 the basic equation of motion for a ship is discussed. Chapter 3 is devoted to the design of the autopilot, that is, regulator design and design of the feedforward compensation. Some experiments are done in Chapter 4, where we use parameters from real ships. In Chapter 5 there is a final discussion and conclusions are drawn. Some useful references are given in Chapter 6. Finally, in Appendix the SIMNON program used is listed.

# 2. Ship Steering Dynamics

A very simple mathematical model of ship steering dynamics, due to Nomoto, relating rudder angle, $\delta$, and disturbing torque, $e$, generated by waves, to turning rate, $r$, and to heading angle, $\psi$, can be stated as a system of linear ordinary differential equations (Källström, 1979),

$$\frac{d}{dt}\begin{pmatrix} r(t) \\ \psi(t) \end{pmatrix} = \begin{pmatrix} -\alpha & 0 \\ 1 & 0 \end{pmatrix}\begin{pmatrix} r(t) \\ \psi(t) \end{pmatrix} + \begin{pmatrix} K & C \\ 0 & 0 \end{pmatrix}\begin{pmatrix} \delta(t-\tau) \\ e(t) \end{pmatrix} \tag{2.1}$$

where higher order dynamics are modelled as a time delay, $\tau$, in the system, see Figure (2.1). This model is a linearization of a more complex nonlinear model, and the parameters, $\alpha$, $K$, and $C$, will depend on the working conditions. Eliminating $r$ from (2.1) gives

$$\psi(t) = \frac{Ke^{-p\tau}}{p(p+\alpha)}\delta(t) + \frac{C}{p(p+\alpha)}e(t) \tag{2.2}$$

where $p = d/dt$ is the differential operator.

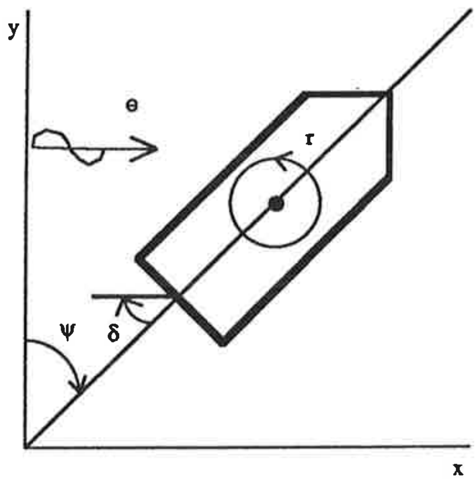



**Figure 2.1** Heading angle, $\psi$, turning rate, $r$, rudder angle, $\delta$, and disturbing torque, $e$, generated by waves

Sampling (2.2) with sampling rate $1/h$ under the assumption that $h > \tau$ yields

$$\psi(k) = \frac{b_1q^2 + b_2q + b_3}{q(q-1)(q-a)}\delta(k) + \frac{c_1q + c_2}{(q-1)(q-a)}e(k) \tag{2.3}$$

where the sampling period is used as time unit and $q$ is the forward shift operator. The discrete time parameters $a$, $b_1$, $b_2$, $b_3$, $c_1$, and $c_2$ can be expressed as functions of $K$, $\alpha$, $\tau$, $C$, and $h$, but for our purposes it is sufficient to treat the ship steering dynamics based on knowledge of the discrete time parameters only.

# 3. Autopilot Structure

The autopilot is designed as an indirect self tuning regulator based on pole placement. The regulator is composed of three parts, a parameter estimator, a design calculation unit, and an implementation of the control law. In the following two sections, the estimator and the regulator structure will be discussed. In Figure (3.1) a block diagram of the ship and the autopilot can be seen.

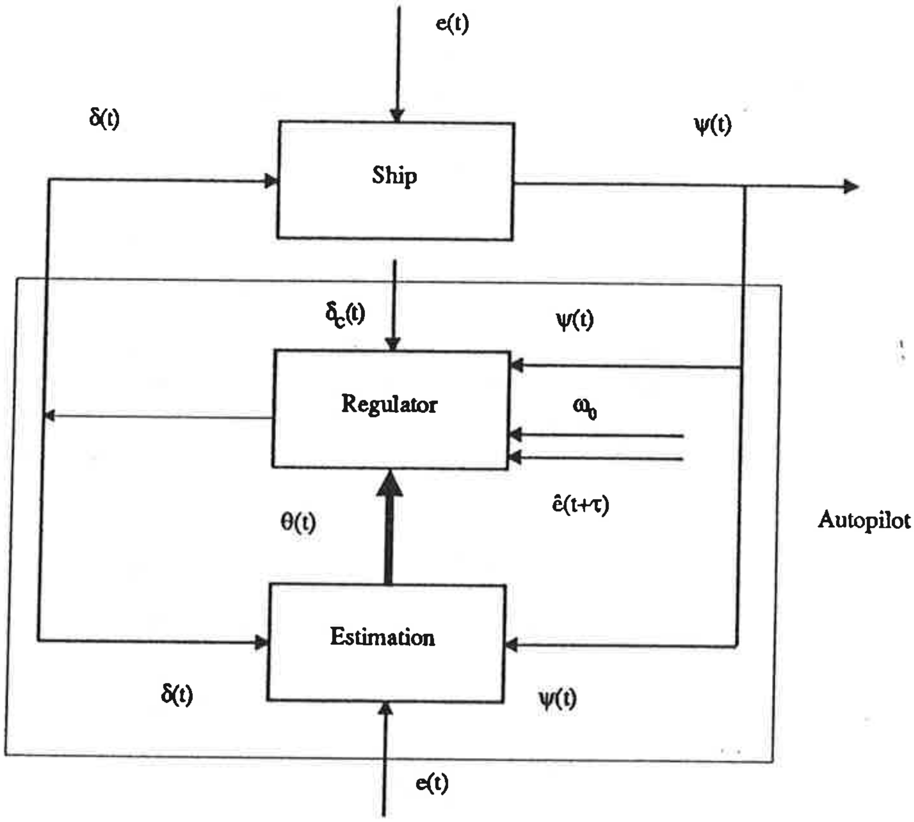



**Figure 3.1** A block diagram of the system involving ship and autopilot

## 3.1 Parameter Estimation

When designing the estimator it is advantaguous to use all the a priori knowledge possible. From Equation (2.3) we have structural knowledge of the system. Equation (2.3) can be written as

$$q(q-1)(q-a)\psi(k) = (b_1q^2 + b_2q + b_3)\,\delta(k) + (c_1q^2 + c_2q)e(k) \qquad (3.1)$$

Rearranging (3.1) and introducing $\Delta\psi(k) = (1 - q^{-1})\psi(k)$ yields

$$\left(1 - aq^{-1}\right)\Delta\psi(k) = \left(b_1 q^{-1} + b_2 q^{-2} + b_3 q^{-3}\right)\delta(k) + \left(c_1 q^{-1} + c_2 q^{-2}\right)e(k) \tag{3.2}$$

Equation (3.2) can be written as

$$\Delta\psi(k) = \varphi^T(k-1)\,\theta \tag{3.3}$$

where we have defined the regression vector, $\varphi$, and the parameter vector, $\theta$, as

$$\begin{aligned} \varphi(k) &= \begin{pmatrix} \Delta\psi(k) & \delta(k) & \delta(k-1) & \delta(k-2) & e(k) & e(k-1) \end{pmatrix}^T \\ \theta &= \begin{pmatrix} a & b_1 & b_2 & b_3 & c_1 & c_2 \end{pmatrix}^T \end{aligned} \tag{3.4}$$

It is now straightforward to apply the recursive least mean square algorithm to (3.3). The recursive least mean square algorithm can be written as

$$\begin{aligned} \widehat{\theta}(k) &= \widehat{\theta}(k-1) + K(k)\left(\Delta\psi(k) - \varphi^T(k)\widehat{\theta}(k-1)\right) \\ K(k) &= P(k-1)\varphi(k)\left(\lambda + \varphi^T(k)P(k-1)\varphi(k)\right)^{-1} \\ P(k) &= \left(I - K(k)\varphi^T(k)\right)P(k-1)/\lambda \end{aligned} \tag{3.5}$$

where $\lambda$ is a forgetting factor. In the basic recursive equations problems are associated with the covariance matrice when there are long periods with no excitation. To make sure that the covariances stay bounded an extended algorithm can be used for estimation of the parameters (Åström and Wittenmark, 1988)

$$\begin{aligned} \widehat{\theta}(k) &= \widehat{\theta}(k-1) + a(k)K(k)\left(\Delta\psi(k) - \varphi^T(k)\widehat{\theta}(k-1)\right) \\ K(k) &= P(k-1)\varphi(k)\left(1 + \varphi^T(k)P(k-1)\varphi(k) + \bar{c}\varphi^T(k)\varphi(k)\right)^{-1} \\ \bar{P}(k) &= \bar{P}(k-1) - a(k)\frac{P(k-1)\varphi(k)\varphi^T(k)P(k-1)}{1 + \varphi^T(k)P(k-1)\varphi(k) + \bar{c}\varphi^T(k)\varphi(k)} \\ P(k) &= c_1\frac{\bar{P}(k)}{\mathrm{tr}\bar{P}(k)} + c_2 I \\ a(k) &= \begin{cases} \bar{a}, & \text{if } |\Delta\psi(k) - \varphi^T(k)\widehat{\theta}(k-1)| > 2\delta \\ 0, & \text{otherwise} \end{cases} \end{aligned} \tag{3.6}$$

In this algorithm the parameters should be choosen such that $c_1 > 0$, $c_2 \geq 0$, $\bar{c} \geq 0$ and $\bar{a} \in [0.1 \ldots 0.5]$. $\delta$ is an estimate of the magnitude of the noise. Compared to the standard implementation of the recursive least squares algorithm with forgetting factor (3.5), the extended method (3.6) leads to slower convergence for the parameter estimates, but the conditional updating gives a security against $P$ matrix explosion, which may occur in the standard algorithm if excitation is poor for a long period. Hence, if we use (3.5) until we get good estimates and then switch algorithm to (3.6), we avoid both slow convergence and $P$ matrix explosion. If we choose $a(k) = 1$, $\bar{c} = 0$, $c_1 = \mathrm{tr}\bar{P}(k)$, and $c_2 = 0$ in (3.6) we get (3.5) with $\lambda = 1$, that is, with no forgetting factor. Thus, it is simple to implement the switching technique.

## 3.2 Regulator Structure

The process to be controlled can be described by the equation

$$A(q)\psi(k) = B(q)\delta(k) + C(q)e(k) \tag{3.7}$$

where the polynomials $A(q)$, $B(q)$, and $C(q)$ are given by (3.1). We postulate a linear control law of the form

$$R(q)\delta(k) = T(q)\delta_c(k) - S(q)\psi(k) + F(q)\widehat{e}(k+\tau) \tag{3.8}$$

where $\widehat{e}(k+\tau)$ is a estimate of $e(k+\tau)$. Eliminating the control signal, $\delta(k)$, in (3.7) and (3.8) yields

$$\psi(k) = \frac{BT}{AR+BS}\delta_c(k) + \frac{1}{AR+BS}(BF\widehat{e}(k+\tau) + RCe(k)) \tag{3.9}$$

where the argument in the polynomials have been suppressed. The problem is now to select the polynomials $R(q)$, $S(q)$, $T(q)$, and $F(q)$. In the first subsection below the polynomials $R(q)$, $S(q)$, $T(q)$ will be chosen according to a pole placement procedure. How to select the feedforward polynomial $F(q)$, is then discussed in the next subsection.

### Regulator Design

Here we only concentrate on the first term on the right hand side of (3.9). The purpose is to select the regulator polynomials such that the closed loop system is

$$\psi(k) = \frac{B_m(q)}{A_m(q)}\delta_c(k) \tag{3.10}$$

where the polynomials $B_m(q)$ and $A_m(q)$ have to be chosen by the designer. Equations (3.9) and (3.10) give the Diophantine equation

$$A(q)R(q) + B(q)S(q) = A_o(q)A_m(q) \tag{3.11}$$

and

$$T(q) = \frac{A_m(1)}{B(1)}A_o(q) \tag{3.12}$$

where the observer polynomial $A_o(q)$ has been introduced, and also has to be chosen by the designer. Demanding a causal regulator and using the fact that the time delay in the closed loop system cannot be less than the time delay in the open loop system results in the polynomials (Åström and Wittenmark, 1984)

$$\begin{aligned}
A_o(q) &= q^3 + a_{o1}q^2 + a_{o2}q + a_{o3} \\
A_m(q) &= q^3 + a_{m1}q^2 + a_{m2}q + a_{m3} \\
B_m(q) &= \frac{A_m(1)}{B(1)}B(q) \\
S(q) &= s_0q^3 + s_1q^2 + s_2q + s_3 \\
R(q) &= (q-1)(q^2 + r_0q + r_1)
\end{aligned} \tag{3.13}$$

where we have avoided cancellation of process zeros and forced an integration action on the regulator. The regulator polynomials are then given as the solution of the Diophantine equation (3.11) with polynomials as in (3.12) and (3.13).

It is advantaguous to make the specifications in continuous time and then transform the polynomials into discrete time. Here we use

$$A_m^c(s) = (s + \alpha_m\omega_m)(s^2 + 2\zeta_m\omega_m s + \omega_m^2)$$

and the relations

$$\begin{aligned} a_{m1} &= p_3 + p_1 \\ a_{m2} &= p_3 p_1 + p_2 \\ a_{m3} &= p_3 p_2 \end{aligned}$$

with

$$\begin{aligned} p_1 &= -2e^{-\zeta_m\omega_m h}\cos(\sqrt{1-\zeta_m^2}\,\omega_m h) \\ p_2 &= e^{-2\zeta_m\omega_m h} \\ p_3 &= -e^{-\alpha_m\omega_m h} \end{aligned}$$

The corresponding relations are also valid for the observer polynomial $A_o(q)$.

The control law (3.8) can be written as

$$R^*(q^{-1})\delta(k) = T^*(q^{-1})\delta_c(k) - S^*(q^{-1})\psi(k) + F^*(q^{-1})\widehat{e}(k+\tau) \qquad (3.14)$$

where we have used the reciprocal polynomials. A practical implementation of the control law (3.14), which compensates for antireset windup, is given by (Åström and Wittenmark, 1988)

$$\begin{aligned} A_o^* v(k) &= T^*\delta_c(k) - S^*\psi(k) + (A_o^* - R^*)\,\delta(k) + F^*\widehat{e}(k+\tau) \\ \delta(k) &= \mathrm{sat}(v(k)) \end{aligned}$$

### Feedforward Compensation

If $\widehat{e}(k+\tau)$ is a good estimate of $e(k+\tau)$ and if the major energy of the waves is centered around the frequency $\omega_0$, the wave disturbance influence on the heading angle can be made small by selecting the polynomial $F(q)$ in a proper way. In the following we will assume that the above holds and that $\omega_0$ and $\widehat{e}(k+\tau)$ are at our disposal. Taking account of the time delay in the control signal and using (3.9), gives the condition

$$B(q)F(q) + R(q)C(q) = 0 \qquad (3.15)$$

Using the fact that the disturbance is almost periodic, with known frequency, gives that (3.15) can be simplified to

$$B(e^{i\omega_0})F(e^{i\omega_0}) + R(e^{i\omega_0})C(e^{i\omega_0}) = 0 \qquad (3.16)$$

that is, we only demand perfect fitting at the frequency $\omega_0$. Setting real- and imaginary parts of (3.16) equal zero, with $F(q) = f_0q^3 + f_1q^2$, yields

$$\begin{aligned} &\mathrm{Re}\{B(e^{i\omega_0})\}\mathrm{Re}\{F(e^{i\omega_0})\} - \mathrm{Im}\{B(e^{i\omega_0})\}\mathrm{Im}\{F(e^{i\omega_0})\} + \\ &\mathrm{Re}\{R(e^{i\omega_0})\}\mathrm{Re}\{C(e^{i\omega_0})\} - \mathrm{Im}\{R(e^{i\omega_0})\}\mathrm{Im}\{C(e^{i\omega_0})\} = 0 \\ &\mathrm{Im}\{B(e^{i\omega_0})\}\mathrm{Re}\{F(e^{i\omega_0})\} + \mathrm{Re}\{B(e^{i\omega_0})\}\mathrm{Im}\{F(e^{i\omega_0})\} + \\ &\mathrm{Im}\{R(e^{i\omega_0})\}\mathrm{Re}\{C(e^{i\omega_0})\} + \mathrm{Re}\{R(e^{i\omega_0})\}\mathrm{Im}\{C(e^{i\omega_0})\} = 0 \end{aligned} \qquad (3.17)$$

Solving (3.17) for $f_0$ and $f_1$ gives the solution. The solution is a rather complicated expression, and it is programmed in Appendix.

61.

### Solution of the Diophantine Equation

The design method used is pole placement. To solve the Diophantine equation

$$A(q)R(q) + B(q)S(q) = A_o(q)A_m(q)$$

is the same as solving a set of linear equations. If there are common factors in the $A$ and $B$ polynomials it is necessary to eliminate them to get a solution. The design of the regulator will be exactly the same but the degree for the $A_o$, $A_m$, $S$, and $R$ polynomials will be reduced with 1 for every common factor. This means tedious calculations for the special cases when $b_3 = 0$, $b_2 = 0$ and $b_3 = 0$, etc. We have only treated the case when the time delay in the continuous process is eliminated, e.g. $b_3 = 0$. Feedforward compensation results in 15 % lower heading deviation for the minesweeper.

# 4. Experiments

In this chapter the autopilot, designed in Chapter 3, is used on different ships, in order to establish the robustness of the design method. In the following experiments are done with a minesweeper and a cargo ship. Both regulation with and without feedforward is treated. The ship velocity is 10 m/s. The values we have used in the experiments are

- Minesweeper $\ell = 55$, $a = -0.14$, and $b = -1.4$.
- Cargo $\ell = 161$, $a = 0.19$, and $b = -1.63$.

The corresponding relationship between these parameters and the parameters used in Chapter 2 are

$$\alpha = \frac{ua}{\ell}$$
$$K = \frac{u^2 b}{\ell^2}$$

where $u$ is the ship velocity. This corresponds to a scaling of the parameters. In these experiments, a high amplitude square wave is exciting the system in open loop in the first 20 time units. This is done to avoid large transients in the heading angle. In a real situation reasonable parameters might be known, and could be used as start values in the estimation. Another possibility is to let the ship be manually steered, with the estimation turned on, until good parameters are available, and then switch on the autopilot.

The results of simulations of a minesweeper are shown in Figures (4.1) and (4.2). Figure (4.1) shows a simulation without feedforward from waves, and Figure (4.2) shows a simulation, with the same wave disturbance acting on the ship, with feedforward compensation. As can be seen, the feedforward gives much better performance on the heading deviation. As can be expected, from the nature of feedforward, the control signal contains a high frequency component, but has a lower amplitude in general.

Better performance is obtained with feedforward than without, doing experiment with a cargo ship. The results are not as good as with the minesweeper, though, see Figure (4.3) and (4.4). Even here a high frequency component is involved in the control signal.

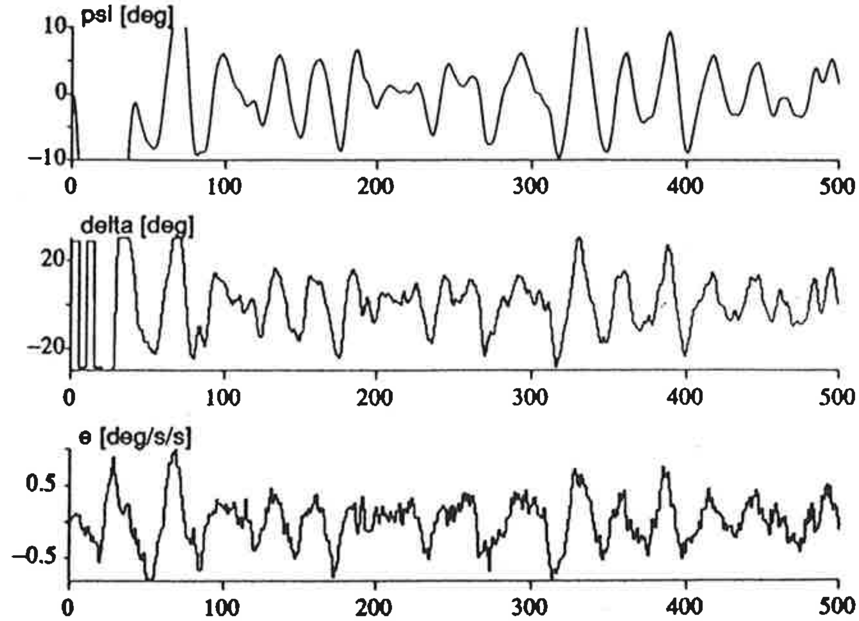



**Figure 4.1** Minesweeper, without feedforward

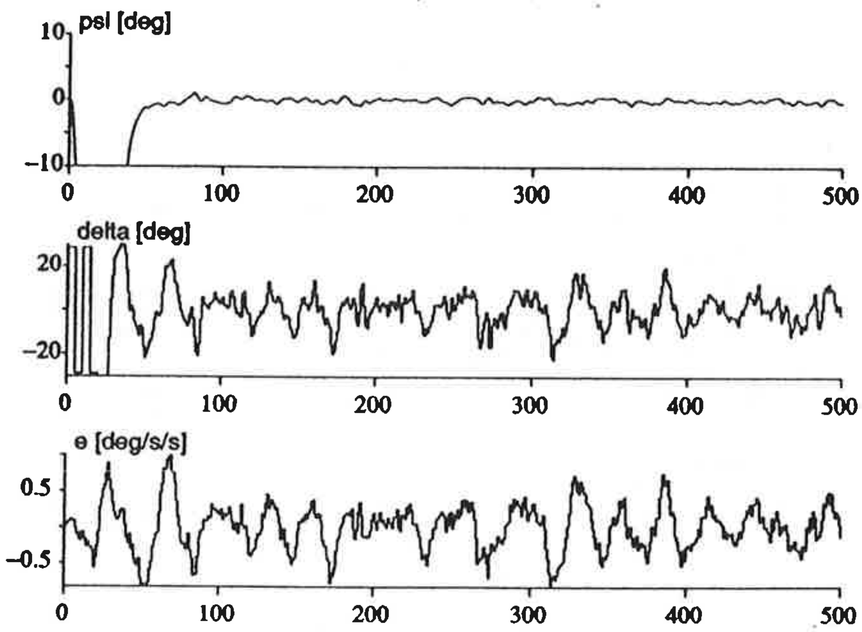



**Figure 4.2** Minesweeper, with feedforward

64.

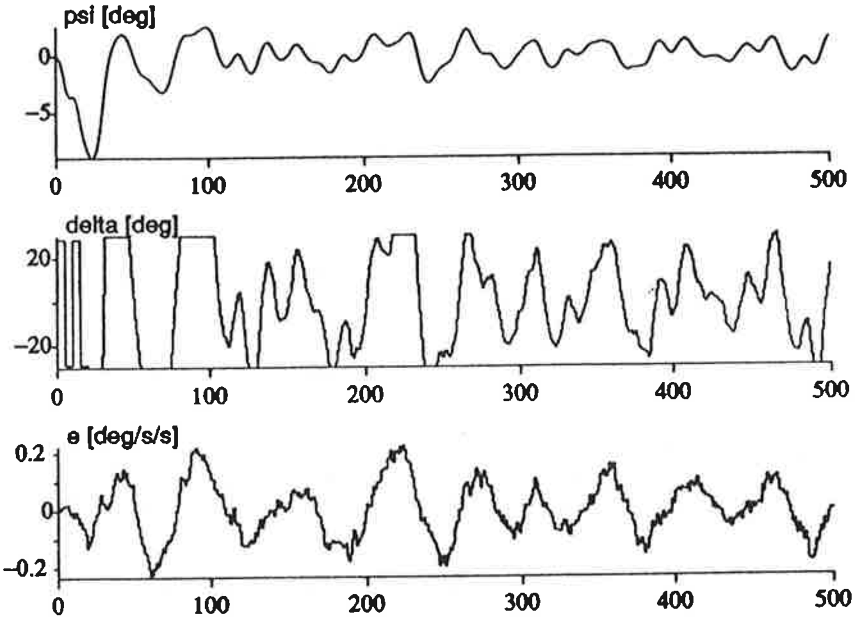



**Figure 4.3** Cargo ship, without feedforward

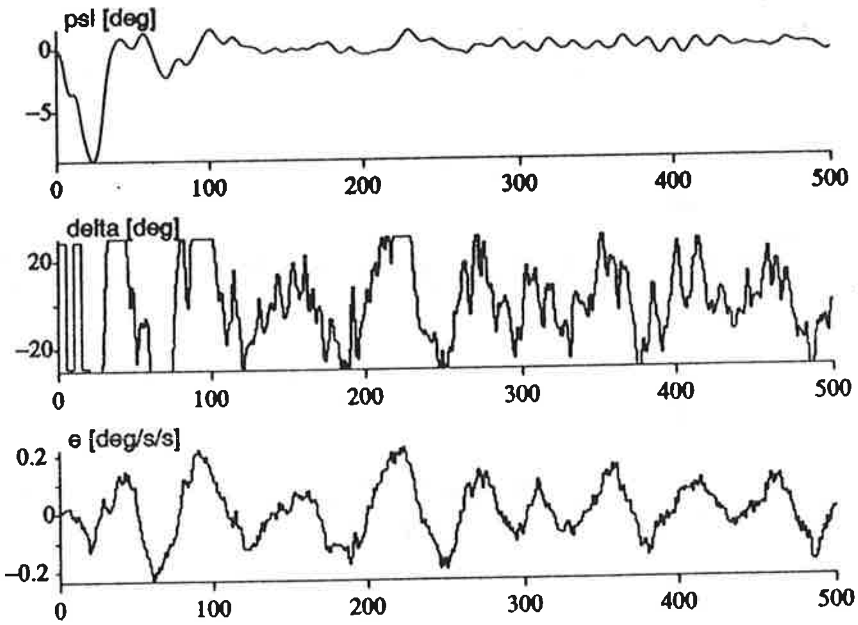



**Figure 4.4** Cargo ship, with feedforward

# 5. Conclusions

In this report the design of an adaptive autopilot for ship steering is discussed. The autopilot is designed as an indirect adaptive regulator based on pole placement with feedforward compensation for wave disturbances. Our approach is different from the usual way of treating this kind of problem. It is customary to use a LQ-controller, since a lossfunction can be derived from physical properties. Severe simulations show that the basic self tuning regulator is very robust and performs well for a wide range of operating conditions. The regulator's ability to compensate for wave disturbances is good. The simulations show that the feedforward increases the performance for a minesweeper and for a cargo ship. Especially the minesweeper's performance is increased with feedforward.

Treatment of the case without time delay in the control signal led to similar results. Experiments with larger vessels gave peculiar results. It might depend on much slower dynamics.

The experience with the simulation language SIMNON has been both positive and negative. The good properties are that differential- and difference equations simply can be written in this language, the ability to connect systems has also been valuable for the structure of our program. We are used to work with high level languages, so we find it remarkable that it is not possible to write a statement on several lines. We also miss a high level programming utility such as the simple if-then-else construction.

# 6. References

KÄLLSTRÖM, CLAES G. (1979): "Identification and Adaptive Control Applied to Ship Steering," PhD Thesis CODEN LUTFD2/(TFRT-1018), Departement of Automatic Control, Lund Institute of Technology, Lund, Sweden.

ÅSTRÖM, K.J, AND B. WITTENMARK (1984): *Computer Controlled Systems*, Prentice Hall.

ÅSTRÖM, K.J, AND B. WITTENMARK (1988): *Adaptive Control*, Addison-Wesley.

# Appendix SIMNON program

```
MACRO MacShip

let n.noise1 = 3
let nodd.noise1 = 13191

let n1.delay = 0
let n2.delay = 2
let space.delay = 7000

SYST delay noise1 Wave Ship STR Con

STORE degpsi[Ship] degdel[Ship] degm[STR]

SIMU 0 500 0.1
SPLIT 3 1
ASHOW degpsi
TEXT 'psi [deg]'
ASHOW degdel
TEXT 'delta [deg]'
ASHOW degm
TEXT 'e [deg/s/s]'

par dt: 1
par stdev1[noise1] : 1 "Wave disturbance
par stdev2[noise1] : 1 "Measurement noise of wave
par stdev3[noise1] : 1 "Measurement noise
END
```

```
CONNECTING SYSTEM Con
TIME t

td1[delay]   = t-tau        " Delay control signal
u1[delay]    = delta[STR]   " delta(t-tau)
delta[Ship] = y1[delay]     "

e[Wave]      = e1[noise1]

mpred[STR] = m[Wave]/wscale+e2[noise1]/nscale

td2[delay]   = t-tau       " Delay wave
u2[delay]    = mpred[STR]  " disturbance
m[STR]       = y2[delay]   "

m[Ship]      = m[STR]

deltac[STR] = 0  " Rudder command signal

psi[STR]     = y[Ship]+e3[noise1]/1000

omega[STR]  = omega " Frequency of
omega[Wave] = omega " wave disturbance

tau   : 0.5
h     : 1
omega : 0.2

wscale : 3000
nscale : 6000
END
```

```
CONTINUOUS SYSTEM Ship
"Model of ship steering dynamics
"
"
"
" d [ r(t) ] [-alfa  0][ r(t) ] [K  C][delta(t-tau)]
"---[       ]=[        ][       ]+[    ][            ]
"dt [psi(t)] [  1    0][psi(t)] [0  0][   m(t)     ]
"
" psi          = heading angle
" r            = turning rate
" delta        = rudder angle
" m            = disturbance torque from waves
" K, alpha, C  = parameters specific for different ships
" tau          = time delay
"
INPUT delta m
OUTPUT y
STATE r psi
DER dr dpsi

dr   = -alpha*r+K*delta+ C*m
dpsi = r

y= psi
degpsi = 180*psi/pi
degdel = 180*delta/pi

alpha = u*a/l
K = u*u*b/(l*l)

pi = 3.1415926536

l : 55      "Ship length [m]
u : 10      "Ship velocity [m/s]

a : -0.14
b : -1.4
c : 1

END
```

```
CONTINUOUS SYSTEM Wave
" Waves modelled as white noise filtered through
"
"
"                        s
" G(s) =  ----------------------------
"           2                         2
"          s + 2*zeta*omega*s + omega
"
"
"
INPUT e omega
OUTPUT m
STATE m1 m2
DER dm1 dm2

dm1 = -2*zeta*omega*m1+m2+e
dm2 = -omega*omega*m1
m=m1

zeta : 0.1

END
```

71.

```
DISCRETE SYSTEM STR
"An indirect self tuning regulator based on pole placement
"with feedforward compensation for wave disturbances.
"A time switch makes it possible to switch between two
"different estimation algorithms, the standard recursive
"least squares method and a method with conditional
"updating.
"
"Input and output parameters
INPUT psi deltac m mpred omega

OUTPUT delta

"State and New Declarations
"Symmetrical P-matrice, covariances for parameters
STATE p11 p12 p13 p14 p15 p16 p22 p23 p24 p25 p26
STATE p33 p34 p35 p36 p44 p45 p46 p55 p56 p66
NEW n11 n12 n13 n14 n15 n16 n22 n23 n24 n25 n26
NEW n33 n34 n35 n36 n44 n45 n46 n55 n56 n66

"Temporary P-matrix
STATE tempp11 tempp12 tempp13 tempp14 tempp15 tempp16
STATE tempp22 tempp23 tempp24 tempp25 tempp26 tempp33
STATE tempp34 tempp35 tempp36 tempp44 tempp45 tempp46
STATE tempp55 tempp56 tempp66
NEW tempn11 tempn12 tempn13 tempn14 tempn15 tempn16
NEW tempn22 tempn23 tempn24 tempn25 tempn26 tempn33
NEW tempn34 tempn35 tempn36 tempn44 tempn45 tempn46
NEW tempn55 tempn56 tempn66

"Phi vector, regression vector
STATE f1 f2 f3 f4 f5 f6
NEW nf1 nf2 nf3 nf4 nf5 nf6

"Theta vector, parameter vector
STATE th1 th2 th3 th4 th5 th6
NEW nth1 nth2 nth3 nth4 nth5 nth6

"Filter variables
STATE psi1 psi2 psi3 delta1 delta2 delta3 m1 m2 m3
NEW npsi1 npsi2 npsi3 ndelta1 ndelta2 ndelta3 nm1 nm2 nm3
STATE v1 v2 v3 deltac1 deltac2 deltac3
NEW nv1 nv2 nv3 ndeltac1 ndeltac2 ndeltac3

"Time and Sample Declarations
TIME t
TSAMP ts

"Start of estimation
```

72

```
"Residual epsilon
e= psi-psi1-th1*f1-th2*f2-th3*f3-th4*f4-th5*f5-th6*f6

"K (Column Vector)= P*Phi
k1= p11*f1+p12*f2+p13*f3+p14*f4+p15*f5+p16*f6
k2= p12*f1+p22*f2+p23*f3+p24*f4+p25*f5+p26*f6
k3= p13*f1+p23*f2+p33*f3+p34*f4+p35*f5+p36*f6
k4= p14*f1+p24*f2+p34*f3+p44*f4+p45*f5+p46*f6
k5= p15*f1+p25*f2+p35*f3+p45*f4+p55*f5+p56*f6
k6= p16*f1+p26*f2+p36*f3+p46*f4+p56*f5+p66*f6

"Denominator (Scalar)= 1+Phi*P*Phi+cbar*Phi*Phi, cbar >= 0
tempff= f1*f1+f2*f2+f3*f3+f4*f4+f5*f5+f6*f6
d= 1+f1*k1+f2*k2+f3*k3+f4*k4+f5*k5+f6*k6+cbar*tempff

"Update Theta values (Row vector)
nth1= th1+xa*k1*e/d
nth2= th2+xa*k2*e/d
nth3= th3+xa*k3*e/d
nth4= th4+xa*k4*e/d
nth5= th5+xa*k5*e/d
nth6= th6+xa*k6*e/d

"Update temporary P-matrix, xa= 0 when residual < ediff
tempn11= (tempp11-xa*k1*k1/d)
tempn12= (tempp12-xa*k1*k2/d)
tempn13= (tempp13-xa*k1*k3/d)
tempn14= (tempp14-xa*k1*k4/d)
tempn15= (tempp15-xa*k1*k5/d)
tempn16= (tempp16-xa*k1*k6/d)
tempn22= (tempp22-xa*k2*k2/d)
tempn23= (tempp23-xa*k2*k3/d)
tempn24= (tempp24-xa*k2*k4/d)
tempn25= (tempp25-xa*k2*k5/d)
tempn26= (tempp26-xa*k2*k6/d)
tempn33= (tempp33-xa*k3*k3/d)
tempn34= (tempp34-xa*k3*k4/d)
tempn35= (tempp35-xa*k3*k5/d)
tempn36= (tempp36-xa*k3*k6/d)
tempn44= (tempp44-xa*k4*k4/d)
tempn45= (tempp45-xa*k4*k5/d)
tempn46= (tempp46-xa*k4*k6/d)
tempn55= (tempp55-xa*k5*k5/d)
tempn56= (tempp56-xa*k5*k6/d)
tempn66= (tempp66-xa*k6*k6/d)

"Update P-matrix, xc1 > 0
trP= tempn11+tempn22+tempn33+tempn44+tempn55+tempn66
n11= if t>chtime then xc1*tempn11/trP+xc2 else tempn11
n12= if t>chtime then xc1*tempn12/trP else tempn12
n13= if t>chtime then xc1*tempn13/trP else tempn13
```

```
n14= if t>chtime then xc1*tempn14/trP else tempn14
n15= if t>chtime then xc1*tempn15/trP else tempn15
n16= if t>chtime then xc1*tempn16/trP else tempn16
n22= if t>chtime then xc1*tempn22/trP+xc2 else tempn22
n23= if t>chtime then xc1*tempn23/trP else tempn23
n24= if t>chtime then xc1*tempn24/trP else tempn24
n25= if t>chtime then xc1*tempn25/trP else tempn25
n26= if t>chtime then xc1*tempn26/trP else tempn26
n33= if t>chtime then xc1*tempn33/trP+xc2 else tempn33
n34= if t>chtime then xc1*tempn34/trP else tempn34
n35= if t>chtime then xc1*tempn35/trP else tempn35
n36= if t>chtime then xc1*tempn36/trP else  tempn36
n44= if t>chtime then xc1*tempn44/trP+xc2 else tempn44
n45= if t>chtime then xc1*tempn45/trP else tempn45
n46= if t>chtime then xc1*tempn46/trP else tempn46
n55= if t>chtime then xc1*tempn55/trP+xc2 else tempn55
n56= if t>chtime then xc1*tempn56/trP else tempn56
n66= if t>chtime then xc1*tempn66/trP+xc2 else tempn66

"Update filter variables
nv1 = v
nv2 = v1
nv3 = v2

npsi1= psi
npsi2= psi1
npsi3= psi2

ndelta1= delta
ndelta2= delta1
ndelta3= delta2

ndeltac1 = deltac
ndeltac2 = deltac1
ndeltac3 = deltac2

nm1= m
nm2= m1
nm3= m2


"Update Phi vector
nf1= psi-psi1
nf2= delta
nf3= f2
nf4= f3
nf5= m
nf6= f5

"Discrete time model polynomial
slaskm = sqrt(1-zetam*zetam)*omegam*h
```

```
slaskm1 = -exp(-alpham*omegam*h)
slaskm2 = -2*exp(-zetam*omegam*h)*cos(slaskm)
slaskm3 = exp(-2*zetam*omegam*h)
am1 = slaskm1+slaskm2
am2 = slaskm1*slaskm2+slaskm3
am3 = slaskm1*slaskm3

"Discrete time observer polynomial
slasko = sqrt(1-zetao*zetao)*omegao*h
slasko1 = -exp(-alphao*omegao*h)
slasko2 = -2*exp(-zetao*omegao*h)*cos(slasko)
slasko3 = exp(-2*zetao*omegao*h)
ao1 = slasko1+slasko2
ao2 = slasko1*slasko2+slasko3
ao3 = slasko1*slasko3

"Process parameters
a  = th1
b1 = th2
b2 = th3
b3 = th4
c1 = th5
c2 = th6

"Solve Diophantine equation
AoEvala = a*a*a+ao1*a*a+ao2*a+ao3
AoEval1 = 1+ao1+ao2+ao3
AmEvala = a*a*a+am1*a*a+am2*a+am3
AmEval1 = 1+am1+am2+am3
BEvala = b1*a*a+b2*a+b3
BEval1 = b1+b2+b3
slask0 = a*b2+a*(2+a)*b1
slask1 = ao2*am3+ao3*am2+a*(2+a)*(2+a)
slask2 = a*(2+a)*(ao1+am1)+(am2+ao1*am1+ao2)*a
slask3 = slask1+slask2-a*(1+2*a)
s3 = ao3*am3/b3
slask4 = slask3-b2*s3-slask0*AmEval1*AoEval1/BEval1
slask5 = slask4+slask0*s3
slask6 = a*a*a*AmEval1*AoEval1/BEval1
slask7 = slask6-a*a*a*s3+s3
slask8 = slask7-AmEvala*AoEvala/BEvala
slask9 = slask5/(a*b1-slask0)-slask8/(a*a*(a-1))
slask10 = (b3-slask0)/(a*b1-slask0)-(a+1)/a
s2 = slask9/slask10
s1 = slask5/(a*b1-slask0)-(b3-slask0)/(a*b1-slask0)*s2
s0 = AmEval1*AoEval1/BEval1-s1-s2-s3
r0 = 2+a+ao1+am1-b1*s0
r1 = (b2*s3+b3*s2-ao2*am3-ao3*am2)/a
slask11 = AmEval1/BEval1
t0 = slask11
t1 = slask11*ao1
```

```
t2 = slask11*ao2
t3 = slask11*ao3

"Calculate Feedforward compensation term
ReC = c1*cos(2*omega)+c2*cos(omega)
ImC = c1*sin(2*omega)+c2*sin(omega)
ReB = b1*cos(2*omega)+b2*cos(omega)+b3
ImB = b1*sin(2*omega)+b2*sin(omega)
ReR = cos(3*omega)+(r0-1)*cos(2*omega)+(r1-r0)*cos(omega)-r1
ImR = sin(3*omega)+(r0-1)*sin(2*omega)+(r1-r0)*sin(omega)
ReHL = -ReR*ReC+ImR*ImC
ImHL = -ReR*ImC-ReC*ImR
FFslask1 = ReB*cos(3*omega)-ImB*sin(3*omega)
FFslask2 = ReB*cos(2*omega)-ImB*sin(2*omega)
FFslask3 = ImB*cos(3*omega)+ReB*sin(3*omega)
FFslask4 = ImB*cos(2*omega)+ReB*sin(2*omega)
FFslask5 = ReHL*FFslask3-FFslask1*ImHL
FFslask6 = FFslask2*FFslask3-FFslask1*FFslask4
ff1 = FFslask5/FFslask6
ff0 = (ReHL-FFslask2*ff1)/FFslask1

"Compute control signal with anti-windup
aor1 = ao1-r0+1
aor2 = ao2-r1+r0
aor3 = ao3+r1
AoRdelta = aor1*delta1+aor2*delta2+aor3*delta3
Tdeltac  = t0*deltac+t1*deltac1+t2*deltac2+t3*deltac3
FFmpred  = ff0*mpred+ff1*(m1+tau*(m-m1)/h)
Spsi     = s0*psi+s1*psi1+s2*psi2+s3*psi3
v = -ao1*v1-ao2*v2-ao3*v3+Tdeltac+FFmpred+AoRdelta-Spsi

d1 = if mod(t,10)<5 then 0.5 else -0.5  " Generate square
y  = if t<20 then d1 else v             " wave

delta=if y<dlow then dlow else if y<dhigh then y else dhigh


"Update sampling time
ts= t+h

"Initial Values and Constants

h: 1        "Sampling time
tau : 0.5   "Time delay in control signal

"Parameters for dead zone
"estimation algorithm
ediff: 0.001
slaskxa = if abs(e) > ediff then abar else 0
xa= if t<chtime then 1 else slaskxa
slaskxc1 = if tempff<0.000001 then 1 else 100/tempff
```

```
xc1= if t<chtime then trP else slaskxc1
xc2=0.001
cbar: 0
abar: 0.1
chtime : 250

degdc = 180*deltac/3.1415926536 "Command and wave in degrees
degm  = 180*m/3.1415926536       "

dlow : -0.5236    " Corresponds to a maximum/minimum
dhigh : 0.5236    " rudder angle of +-30 degrees

zetam  : 1      " Continuous time model
omegam : 0.5    " specifications
alpham : 0.5

zetao  : 1      " Continouos time observer
omegao : 1      " specifications
alphao : 1

th1: 0.5
th2: 0.5
th3: 0.5
th4: 0.5
th5: 0.5
th6: 0.5

tempp11: 100
tempp22: 100
tempp33: 100
tempp44: 100
tempp55: 100
tempp66: 100

END
```

77.

Bengt Ekelund och Michael Johansson

Projekt i Adaptiv reglering

Vi fick i uppdrag att försöka förbättra en regulator till en autopilot på en båt. Regulatorn var av en adaptiv typ där parametrarna bestämdes med hjälp av kunskaper om båtens fart, riktning, roderutslag mm. I algoritmen som användes, antogs båten vara påverkad av vissa störningar. En av dessa var orsakad av vågorna. Störningarna pga havets rörelse dvs, dvs vågorna, sågs i modellen som vitt brus. Vi skulle finna ett sätt att adaptera för en störning som istället för vitt brus, hade sinusform.

Vårt arbete koncentreades till en början på att skatta en given sinussignal. För att skatta en sinusvåg är det viktigt att känna till att den satisfierar ekvationen
$d^2f/dx^2 + w^2f = 0$
detta motsvaras av den samplade ekvationen
$f(t + 2h) + a1f(t + h) + a2f(t) = 0.$
Den allmänna formen på en funktion som uppfyller differnsekvationen ovan är $f(t) = \exp(-zwt)\sin(wt(1 - z^2)^{\frac{1}{2}})$, med $a1 = -2\exp(-zwh)\cos(wh(1 - z^2)^{\frac{1}{2}})$ och $a2 = \exp(-2zwh)$. Parametrarna a1 och a2 ovan kan skattas tex med en minsta-kvadrat algoritm.

Vi är i första hand bara intresserade av en ren sinusvåg. Det exponentiella avtagandet kommer visserligen att vara intressant då vågfrekvensen ändras, och systemet väntas svänga lite, men framförallt vill vi bara veta om det går bra att skatta frekvensen på signalen. Genom att sätta $z = 0$ eliminerar vi det exponentiella avtagandet och vinner dessutom att $a2 = 1$. Vi behöver bara skatta a1. Ett program som skattar a1 enligt minsta kvadrat metoden metoden med glömskefaktorn $\lambda = 0.98$ ges av BMK10, se bilaga.

## Kommentarer till vissa av programmen

### BMK10 och BMK20

Programmen är i stort sett de samma. Båda arbetar efter algoritmen i kapitel 3 sidan 11 i läroboken. Skillnaden ligger i att vi i BMK20 skattar två parametrar och i BMK10 bara en. Detta får till följd att P i BMK20 är en 2x2 matris medan den iBMK10 är en skalär. Feltermerna e och eps till uppdateringen skiljer sig också lite då man i den ena tar hänsyn till två skattade parametrar och i den andra bara en.

### BMK11 och BMK22

Båda programmen arbetar efter algoritmen i kapitel 11 sidan 8 i läroboken. Precis som i programmen ovan får vi olika P och olika feltermer. Valen P = 0,1 i BMK11 och c1 = 0,5 samt c2 = 1E-4 i BMK21 är empiriskt framtaget. Dessa värdena visar sig vid simulering ge en bra kompromiss mellan snabbhet och brusnivå.

### BOAT

En enkel modell för en båt. Båten antas bete sig som en ren tidsfördröjning. Värt att nämna är att vi samplar 10 ggr snabbare än i de övriga programmen.

### BCONN och BCONNX

Connecting system som binder ihop skattaren BMK** med sinusvågen som önskas skattad.

_RIGEL$DUA2:[STUDENTS.F83MJ.REG]BCONN.T;4 24-APR-1988 15:10

```
CONNECTING SYSTEM BCONN
TIME t
y[bmk] = y[bsin] + sigma*e1[noise1]
sigma : 0.1
END
```

```
CONTINUOUS SYSTEM bsin
OUTPUT y omega
STATE y0 y1
DER dy0 dy1
TIME t

INITIAL
y0 = 1

SORT
dy1 = -2*zeta*omega*y1-omega*omega*(y0-u)
dy0  = y1
y = y0

omega = if mod(t,per)<per/2 then om1 else om2

zeta : 0
om1: 10
om2 :20
per :40
u : 0
END
```

82

```
DISCRETE SYSTEM bmk
INPUT y
STATE y1 y2 p a1
NEW ny1 ny2 nyp nya1
TIME t
TSAMP ts

nya1 =a1+k*eps
eps = y+a1*y1+y2
k = -p*y1/(lambda+y1*y1*p)
nyp = (1+k*y1)*p/lambda
ny1 = y
ny2 = y1

arg = min(max(-a1/2,-1),1)
w = arccos(arg)/h

ts = t+h

lambda : 0.98
h :0.1
p : 10
END
```

83

_RIGEL$DUA2:[STUDENTS.F83MJ.REG]BMK11.T;6 24-APR-1988 14:55

```
DISCRETE SYSTEM bmk
INPUT y
STATE y1 y2  a1
NEW ny1 ny2  nya1
TIME t
TSAMP ts
nya1 =a1+k*eps
eps = y+a1*y1+y2
k = -p*y1/(1+y1*y1*p)
ny1 = y
ny2 = y1

arg = min(max(-a1/2,-1),1)
w = arccos(arg)/h

ts = t+h

h :0.1
p : 0.1
END
```

23

_RIGEL$DUA2:[STUDENTS.F83MJ.REG]BMK20.T;2 24-APR-1988 14:57

```
DISCRETE SYSTEM bmk
INPUT y
STATE  f1   f2   t1   t2   p11   p12   p22
NEW   nf1  nf2  nt1  nt2  np11  np12  np22
TIME t
TSAMP ts

e = y-t1*f1-t2*f2

k1  = p11*f1+p12*f2
k2  = p12*f1+p22*f2
d = lam+f1*k1+f2*k2

nt1  = t1+k1*e/d
nt2  = t2+k2*e/d

np11  = (p11-k1*k1/d)/lam
np12  = p12-k1*k2/d
np22  = (p22-k2*k2/d)/lam

nf1  = -y
nf2  = f1

arg = if t2>0 then min(max(-t1/2/sqrt(t2),-1),1) else 1
w = arccos(arg)/h
z = if t2>0 then -ln(t2)/2/h/w else 0

ts = t+h

lam : 0.98
h : 0.1
t1  : 0
t2  : 1
p11  : 10
p12  : 10
p22  : 10
END
```

```
DISCRETE SYSTEM bmk
INPUT y
STATE  f1  f2  t1  t2  p11  p12  p22
NEW   nf1 nf2 nt1 nt2 np11 np12 np22
TIME t
TSAMP ts

e = y-t1*f1-t2*f2

k1  = p11*f1+p12*f2
k2  = p12*f1+p22*f2
d  = 1+f1*k1+f2*k2

nt1  = t1+k1*e/d
nt2  = t2+k2*e/d

np11s  = p11-k1*k1/d
np12s  = p12-k1*k2/d
np22s  = p22-k2*k2/d

np11  = c1*np11s/(np11s+np22s)+c2
np12  = c1*np12s/(np11s+np22s)
np22  = c1*np22s/(np11s+np22s)+c2

nf1  = -y
nf2  = f1

arg = if t2>0 then min(max(-t1/2/sqrt(t2),-1),1) else 1
w = arccos(arg)/h
z = if t2>0 and w>0 then -ln(t2)/2/h/w else 0

ts = t+h

c1  : 0.5
c2  : 1e-4
h  : 0.1
t1  : 0
t2  : 1
p11  : 0.25
p12  : 0.25
p22  : 0.25
END
```

85

_RIGEL$DUA2:[STUDENTS.F83MJ.REG]BCONNX.T;10 24-APR-1988 15:08

```
CONNECTING SYSTEM BCONNX
TIME t
u[boat] = uut[bstyr]
d[boat] = y[bsin] + sigma*e1[noise1]
y[bmk] = y[boat]
yin[bstyr] = y[boat]
uin[bstyr] = uc
w[bstyr] = omega[bsin]
sigma : 0.05
uc : 0
END
```

98

```
DISCRETE SYSTEM boat
INPUT u d
OUTPUT y
STATE x
NEW  nx
TIME t
TSAMP ts

nx = u+d
y = x

ts = t+h
h:0.01
END
```

88

```
DISCRETE SYSTEM bstyr
INPUT yin w uin
OUTPUT uut
STATE  f1  f2  t1  t2  p11  p12  p22
NEW    nf1 nf2 nt1 nt2 np11 np12 np22
TIME t
TSAMP ts

e = yin-uut-t1*f1-t2*f2

k1  = p11*f1+p12*f2
k2  = p12*f1+p22*f2
d  = lam+f1*k1+f2*k2

nt1  = t1+k1*e/d
nt2  = t2+k2*e/d

np11  = (p11-k1*k1/d)/lam
np12  = p12-k1*k2/d
np22  = (p22-k2*k2/d)/lam

nf1  = sin(w*(t+d1))
nf2  = cos(w*(t+d1))

"styrlag
uut  = uin-t1*sin(w*(t+d0))-t2*cos(w*(t+d0))


ts  = t+h

lam  : 0.98
h  : 0.01
p11  : 10
p12  : 10
p22  : 10
pi  : 3.1415926536
d0  : 0
d1  : 0
END
```

89.

# HÖGTALARSERVO

Projekt i Adaptiv signalbehandling
och Adaptiv reglering

av
Johan Magnusson och Johan Waldeck

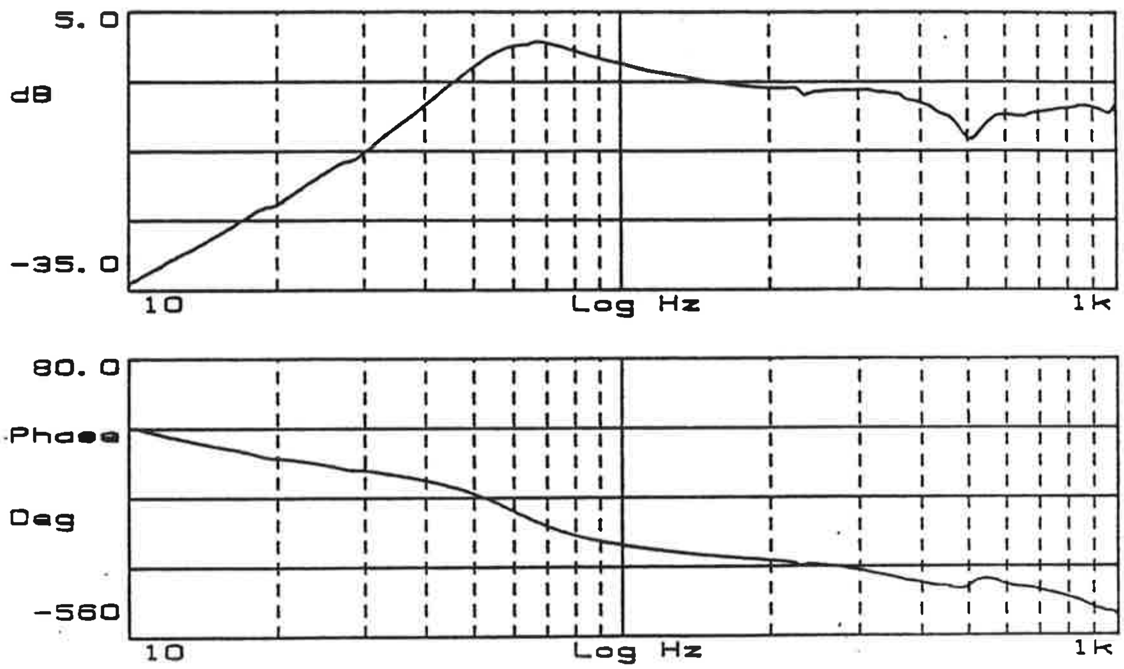
VT 88

90.

# Innehåll

# 1 Inledning

Återkoppling är en vanlig teknik för att komma tillrätta med olinjäriteter i tekniska system. I ett modernt HiFi-system hittar man alltid någon form av återkoppling i förstärkaren. Avsikten är att utsignalen skall likna insignalen i så hög grad som möjligt, oberoende av lyssningsvolym, belastning av förstärkaren o s v. Tyvärr så brukar man inte återkoppla från högtalaren, vilken i sig själv uppvisar kraftiga olinjäriteter. Fig 1.1 visar frekvens och fasgång hos en konvetionell bashögtalare.



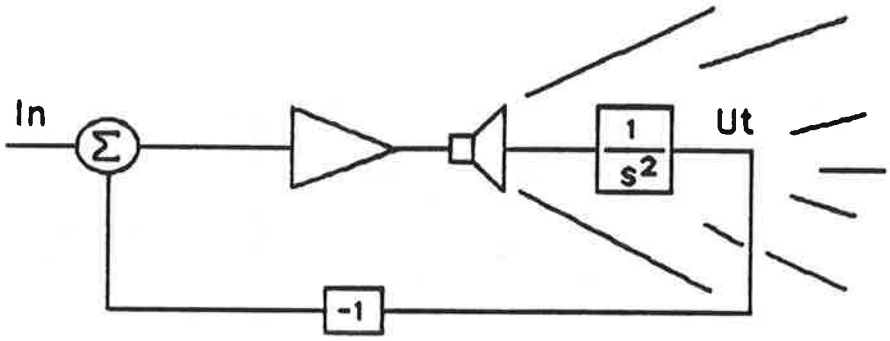

Figur 1.1 Frekvens- och faskurvor för konventionell bashögtalare.

Mätningarna är gjorda i E-husets "ekofria" rum med hjälp av en hp 3562 Dynamic signal analyzer och en Bruel&Kjaer ljudnivåmätare. Med ljudnivåmätaren placerad på ca en meters avstånd ifrån högtalaren sveptes frekvensområdet 10 -1kHz med en sinuston.

Målsättningen med detta projekt är att undersöka om man med hjälp av återkoppling kan förbättra högtalarens egenskaper i basområdet, d v s räta ut amplitud- och fasgång.

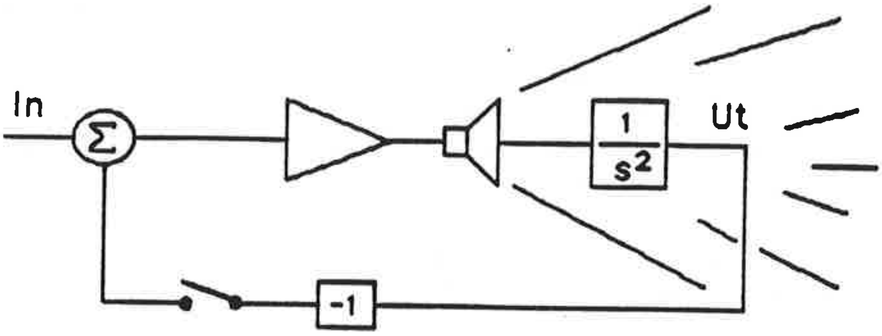
# 2 Metoder och mätningar

Utsignalen från högtalaren kan återkopplas på flera sätt. Det enklaste sättet att få tag i denna utsignal är m h a en mikrofon placerad i rummet. Nackdelen med denna metod är att mikrofonen får gammal information, eftersom det tar en viss tid för ljudet att ytbredas från högtalaren till mikrofonen. Genom att minska avståndet kan tidsfördröjningen minskas. Detta medför att återkopplingssignalen distorderas, eftersom ljudkällan inte längre kan betraktas som punktformig. P g a dessa nackdelar bör signalen återkopplas på annat sätt. Istället för att mäta ljudtrycket får vi försöka mäta en signal som så väl som möjligt överensstämmer med denna. En givare placerad på högtalarkonen som känner av dess läge, rörelse skulle kunna ge samma information. För att kunna minimera felet mellan in- och utsignal behövs en regulator. Det är önskvärt att denna ställer in sig optimalt för alla kombinationer av förstärkare och högtalare. En sådan regulator skulle kunna bestå av en variabel loopförstärkning med någon justeralgoritm. För att praktiskt prova teorierna om återkoppling av signal från högtalare, gjordes mätningar på Philips MFB-högtalare. Denna fanns på marknaden under tidigt 70-tal.




Figur 2.1 Blockschema för Philips MFB-högtalare.

Beskrivning av MFB (Motional FeedBack): Philipshögtalaren är av trevägs tryckkammartyp med inbyggd förstärkare. Ett aktivt delningsfilter ger bassteget en övre gränsfrekvens på 500 Hz. Eftersom baselementet är det mest intressanta i detta fallet, kommer endast detta att behandlas i fortsättningen. Vid låga frekvenser kan det uppstå problem i att kontrollera högtalarkonens långa rörelser. Dessa rörelser kan kontrolleras m h a tvångsstyrning. I detta fall användes en accelerometer, integrerad i högtalarelementet, för att avkänna högtalarens rörelser. Utsignalen från denna givare jämförs med insignalen och differenssignalen används för att styra konen. Återkopplingens storlek, motkopplingen, kan varieras m h a en potentiometer i förstärkaren.




Figur 2.2 Blockschema open / closed loop.

Testhögtalarens amplitud- och fasfunktion uppmättes med och utan återkoppling. Vi fann att återkoppling gav en rakare amplitud- och fasgång jämfört med det fall då ingen signal återkopplades. Se diagram 2.2.

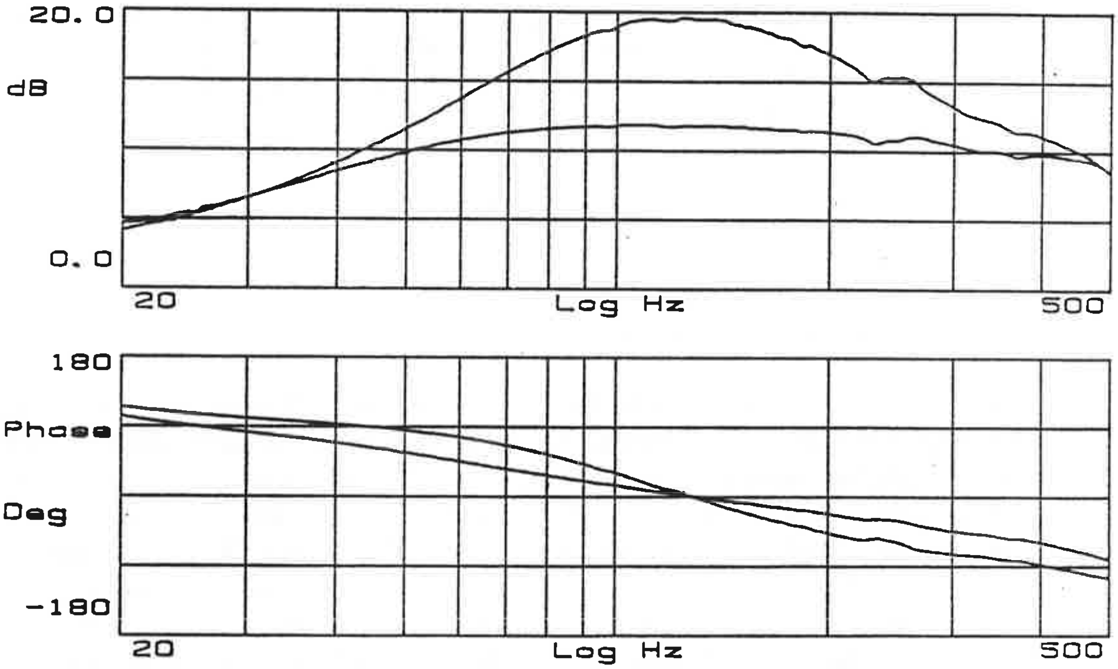
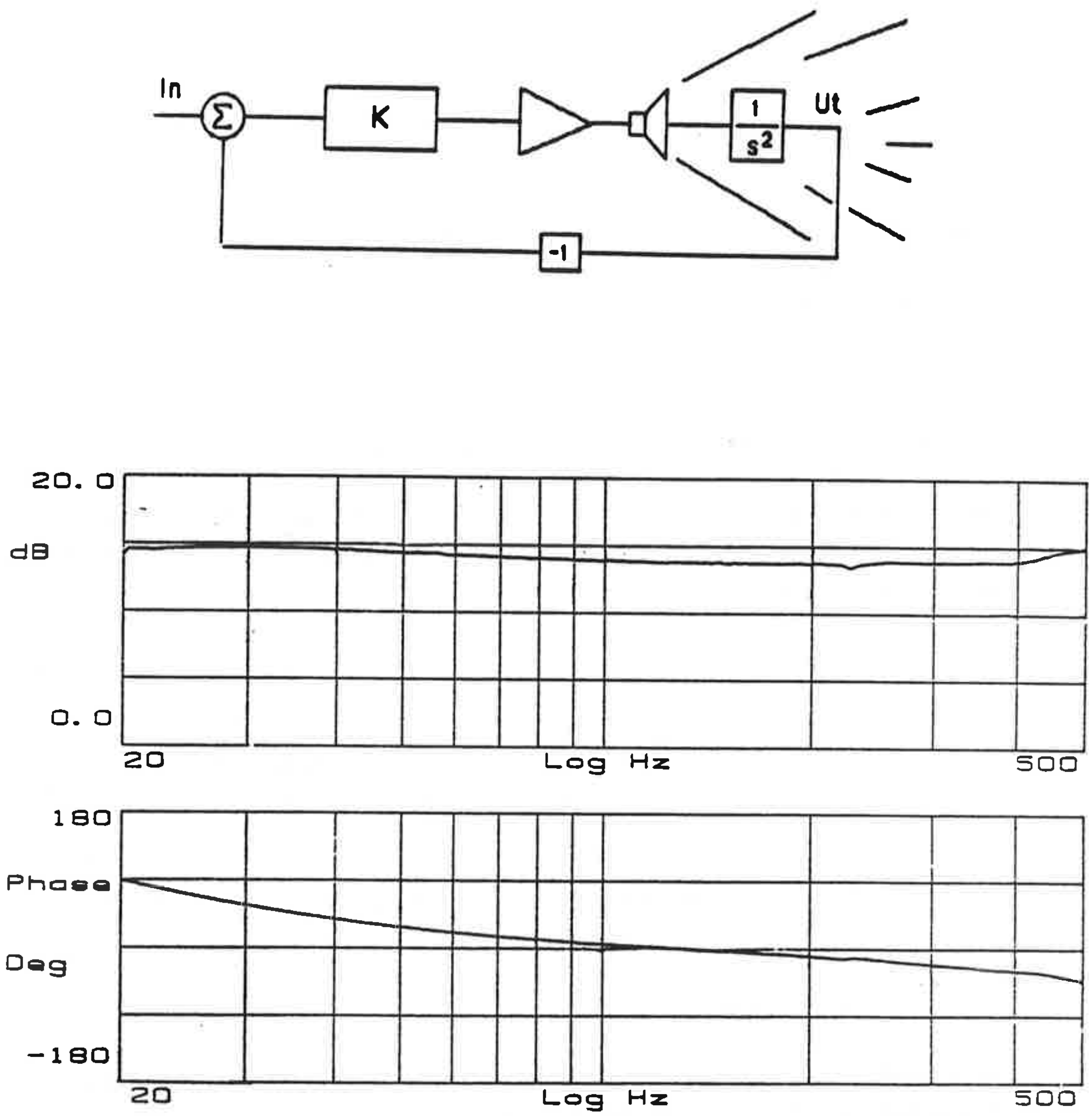



Diagram 2.2 Amplitud- och fasgång open / closed loop.

För att ytterligare förbättra regulatorns egenskaper infördes en förstärkare i övre delen av loopen, samtidigt som potentiometern, motkopplingen, fixerades vid ett "normalläge". Mätningar med olika stor förstärkning gjordes. Vid stor förstärkning fås en näst intill konstant amplitud över hela frekvensområdet. Se figur 2.3.




Figur 2.3 Blockschema samt amplitud- och fasgång vid "optimal" loop förstärkning.

De proportionella regulatorerna som omtalats har det gemensamt att de måste justeras in för varje ny högtalare som skall anslutas. Önskvärt vore en regulator som passar alla typer av högtalarelement, som belastar den. Vad som sökes är en självinställande proportionell återkoppling. Denna skall i uppstartningsskedet adaptera loop-förstärkningen optimalt och sedan hålla den konstant. För att realisera denna regulator har ett Lab90-system använts, bestående av AD-omvandlare, TMS 32010 signalprocessor samt DA-omvandlare.

# 3 Adaptivitet

För att regulatorn skall kunna justeras optimalt för alla kombinationer av förstärkare och högtalare krävs adaptivitet. Genom att använda Lab90-systemet och där implementera lämplig algoritm fås möjlighet till detta. Det adaptiva systemet består av ett FIR-filter realiserad i TMS-kod. Endast en vikt används för att åstadkomma proportionell återkoppling. Som adapteringsalgoritm används LMS:

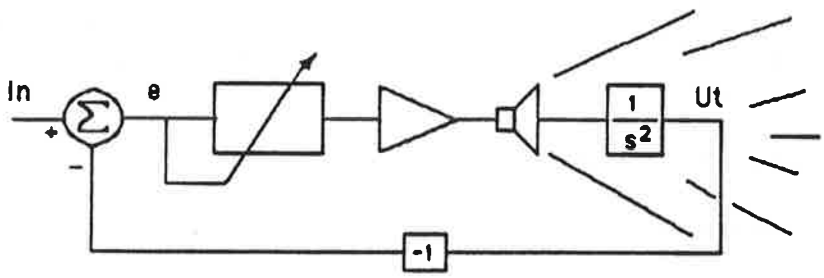
$$W(k+1) = W(k) + 2\mu e(k)X(k)$$

där $W(k+1)$ är den nya vikten och $W(k)$ den gamla. $\mu$ anger hur stort korrigeringssteg som skall tas, $e(k)$ anger felsignal och $X(k)$ insignal.

LMS-algoritmen kan i detta fall liknas vid ett modellreferenssystem med modellen 1, eftersom det är önskvärt att utsignalen är identisk med insignalen. Vid modellreferens används MIT-regeln, vilken ger:

$$d\theta/dt = -\text{gamma} \cdot e \cdot de/d\theta$$

MIT-regeln används för att justera parametrarna, eller i vårt fall vikterna, på ett effektivt sätt. $\theta$ är parametrarna, gamma anger hastigheten hos justeringen och e är felet. Innebörden är att felet minskas om parametrarna ändras i felets negativa gradients riktning. Samma strategi används i LMS-algoritmen, där sista produkttermen anger just detta.




Figur 3.1 Blockschema

Vid testkörning av regulator fungerade adapteringen väl. Förstärkningen och därmed ljudvolymen ökade i inledningsskedet och intog så småningom ett slutläge. En ny mätning gjordes, se figur 3.1. Problem uppstod däremot när förförstärkaren gav en för kraftig signal. Ljudvolymen steg kraftigt och adapteringen upphörde att fungera. Detta berodde på att regulatorn kunde adaptera uppåt men inte nedåt. Studerar man LMS-algoritmen närmare finner man att X(k) och e(k) i vårt fall är samma signal, varför vi får en kvadratisk korrigeringsterm som alltid är positiv. Detta innebär att systemet kan bli instabilt vid stora insignaler, samt att adaption ej kan ske nedåt.

I diagram 3.1 visar hur amplitud- och fasgång ser ut när systemet har adapterat.

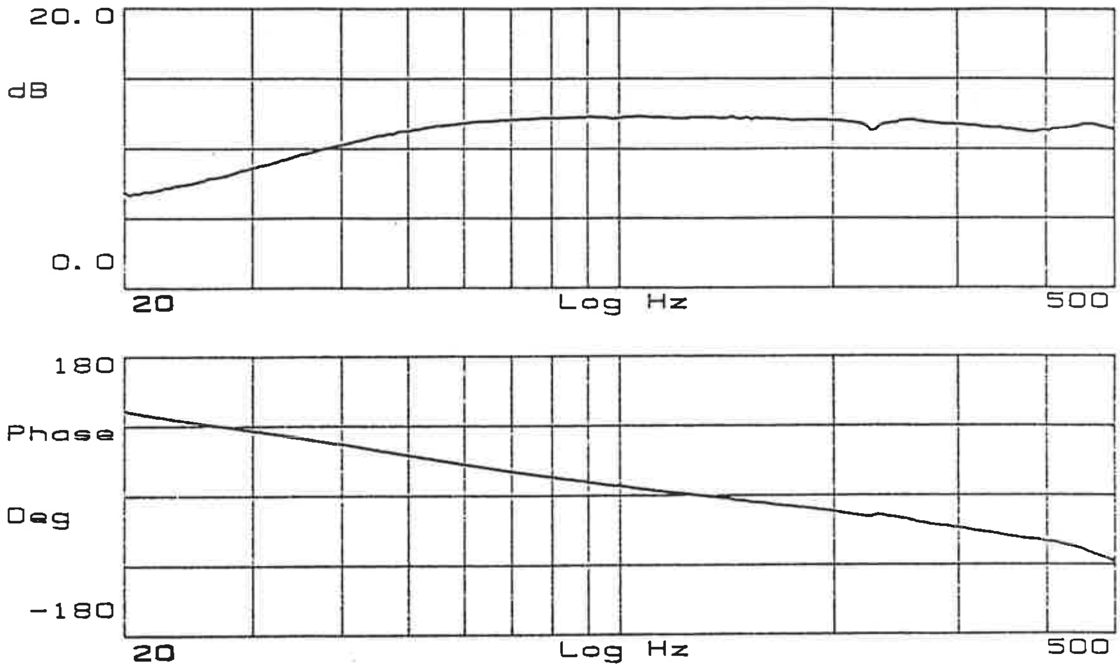



Diagram 3.1

# 4 Sammanfattning

Vi har sett hur man med enkel proportionell återkoppling kan förbättra en högtalares egenskaper. En enkel metod att på lab90 system implementera en adaptiv p-regulator har provats. Mätningar har visat att denna har förbättrat högtalarens egenskaper vid måttliga signalnivåer, men kan bli instabil vid höga signalnivåer då den saknar möjlighet att adaptera nedåt.

# REGLERING AV INSPRUTNING

Fult med understyrning.

Projekt i kursen

ADAPTIV REGLERING 1989

Instutitionen för reglerteknik

Utfört av

**MATS BÖRJESSON**

Kosmetik
1) Understyrn
2) Högermarginalen
3) Ekvationer
4) generera

Rapporten behandlar ett projekt i kursen adaptiv reglering, där jag har provat olika regleringssätt av bränsleinsprutningen i en bilmotor. Proverna som har utförts med hjälp av simulering i Simnon visar på fördelarna med parameterstyrning för att uppnå en snabb och stabil reglering.

# INNEHÅLLSFÖRTECKNING

# INLEDNING

~~Den här~~ R rapporten behandlar olika metoder att reglera luft/bränsle-blandningen hos en bilmotor. De äldre insprutningssystemen på bilar hade ingen reglering av insprutningstiden, vilket medförde att motorn inte alltid arbetade under de mest gynnsamma förhållanden. Med dagens luftföroreningar måste man därför se till att luft/bränsle-blandningen är optimal. I moderna bilar görs detta genom att återkoppla signalen från en lambda-sensor. Eftersom dynamiken i motorn beror på varvtal och belastning vore det lämpligt att ändra regulatorn efter körförhållanden. Med hjälp av datorsimuleringar har jag undersökt prestanda hos olika regleringsmetoder, och jämfört dessa med varandra.

101.

## OLIKA SÄTT ATT REGLERA LUFT/BRÄNSLE-BLANDNINGEN

Nästan alla dagens bilar har insprutningsmotorer, dvs bensinen sprutas in i motorn via insprutare. Genom att variera öppningstiden hos insprutarna, varieras mängden insprutat bränsle, och därigenom även luft/bränsle-förhållandet. Beroende på luft/bränsle-förhållandet fås olika mycket avgaser och även verkningsgraden påverkas. Avgasernas sammansättning varierar med varvtal och belastning hos motorn. Ett typexempel finns i figur 1. Den bästa verkningsgraden och minsta avgaserna fås för A/F (air to fuel ratio) på 19 -22. Vid hög belastning kan detta värde behöva sänkas till stökiometriska punkten 14.64 för att få ut tillräcklig effekt ur motorn
För att bestmma insprutningstiden kan man gå tillväga på 3 tre olika sätt :

* ~~Open-loop~~ ~~Programmerads~~ Öppen styrning
* Reglering med hjälp av lambda-sond
* Reglering med hjälp av bättre givare

Öppen styrning

### OPEN-LOOP

Principen för open-loop framgår av figur 2. Genom att mäta varvtal och luftflöde (som är ett mått på belastningen), kan insprutningstiden räknas ut ~~genom~~ med formeln
Tn=K*alfa*qa/N .
där K är en konstant, alfa är en faktor som beror på vilket A/F-värde man vill ha (bestäms av qa och N),
och qa=luftflöde [kg/s] och N=varvtal [rpm]. Praktiskt genomförs detta genom att i en tabell lägga in insprutningstider för olika driftfall (qa och N). Nackdelarna med open-loop är att man inte har någon återkoppling och följaktligen inte vet om rätt A/F-förhållande uppnås. Speciellt fås en A/F-transient vid gaspådrag pga att det tar en liten stund innan bensinen kommer in i cylindrarna från det att luftspjället har öppnats. 128

## REGLERING MED LAMBDA-SOND

En lambda-sond har en ~~givare~~karakteristik som ~~ger~~ beskrivs av
utsignal~~en~~ :

+1 om A/F < 14.64 (=A/F stökiometrik)

-1 om A/F > 14.64

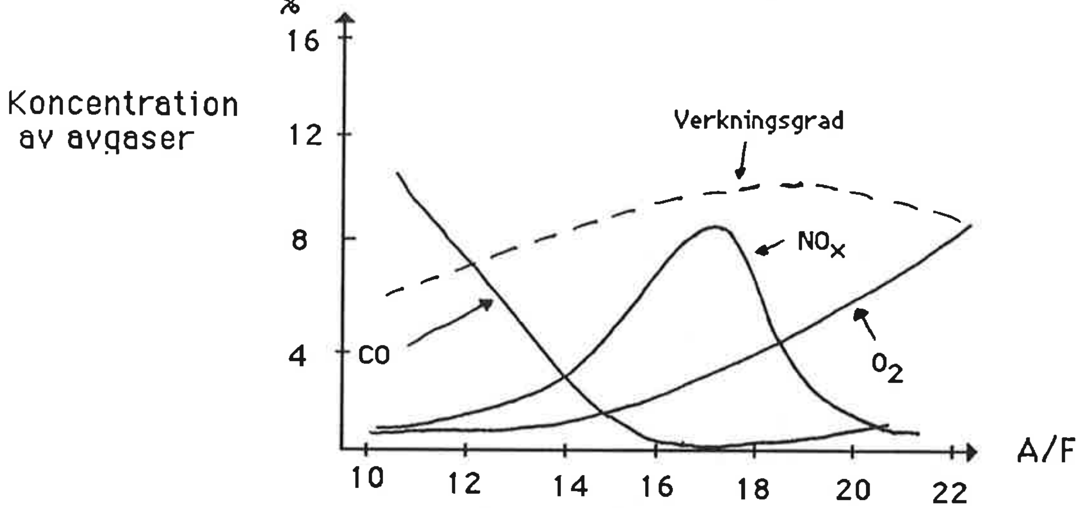
Genom att återkoppla signalen från lambda-sonden till en regulator kan ~~man~~ modulera insprutningstiden så att rätt A/F uppnås (figur 3). Pga att lambda-sonden bara känner av om A/F är större eller mindre än 14.64, går det bara att reglera A/F till 14.64. Reläkarakteristiken hos lambd-sonden ger upphov till en ~~"limit-cycle" oscillation~~ periodisk svängning. Dvs A/F svänger runt 14.64 hela tiden. Regulatorn brukar utgöras av en ren integrator : Ki/s där Ki kommer att bestämma amplitud och frekvens hos oscillationen. Det är alltså denna typ av regulator som sitter i dagens bilar med katalysator. Nackdelen med den är att man inte kan låta A/F vara optimalt (19-22).

## REGLERING MED BÄTTRE GIVARE

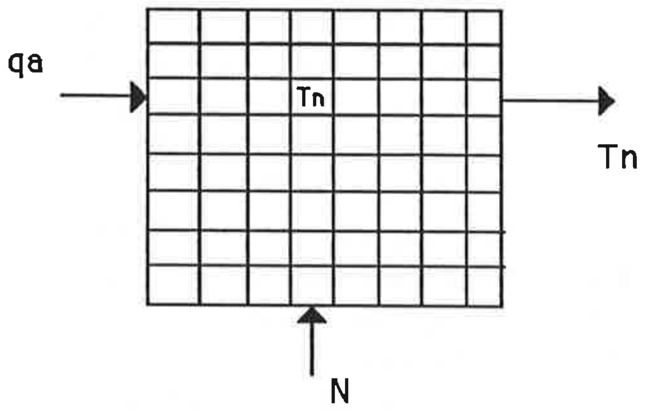
För att kunna reglera insprutningstiden optimalt måste A/F mätas mer noggrant än med lambda-sond. Enligt figur 1 är andelen syrgas nästan linjärt beroende av A/F i området 14.5-22. Detta innebär att med en syresensor kan man få ett exakt mått på A/F. Sättet att reglera insprutningstiden blir densamma som för lambda-sonden, med den skillnaden att man slipper oscillationen (figur 4). Börvärdet A/F ref bestäms av qa och N och lagras i tabellen på samma sätt som Tn.

14-22

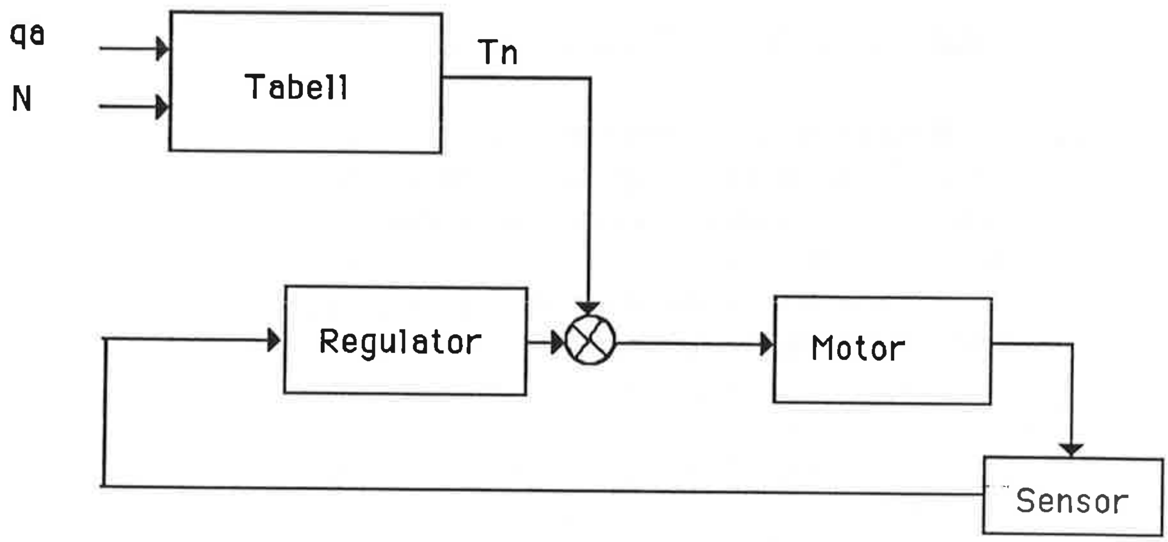
luft-bränsle kvot




Figur 1 Avgasernas beroende av A/F




Figur 2 Tabellens utseende




Figur 3 Reglering med lambda-sond

Index

# MODELL FÖR BILMOTORN

Dynamiken i en bilmotor kan beskrivas av följande modell (fig 5) :

Blandning i insugningsrör : dx1/dt=(u-x1)/Tc
u=A/F in =qa/(Ttot*ß*N)
x1=A/F in i cylindern
ß=en konstant
Tc=ktc/N

Förbränning : dx2/dt=x1(t-Td)
x2=A/F "ut ur motorn"
Td=ktd/N

Parametern Tc är tidskonstanten för blandningen av bränsle och luft i insugningsröret. Fördröjningen Td är tiden mellan blandningen i insugningsröret till avgaserna kommer ut. Typiska värden på ktc och ktd kan vara 175 resp 285.

?? Dessa kraftiga variationer i Tc och Td ger begränsningar i regulatorn som måste anpassas till att fungera för alla varvtal.
En annan orsak till variationer i processen är att insprutningsventilen ~~insprutarna som~~ åldras och inte ger samma mängd bränsle/sekund som från början (ß ändras). Detta är dock en mycket långsam variation jämfört med variationerna hos Tc och Td, och den påverkar ej heller det slutna systmet lika mycket.
Även luftmängdsmätaren kan ge upphov till liknande variationer som de i ß.

Varvtalet varierar från 600 - 6000 RPM vid normal körning. Detta medför att Tc och Td varierar med en faktor 10.

105.

# OLIKA SÄTT ATT KOMPENSERA FÖR VARIATIONER

## PARAMETERSTYRNING

För att få bra reglerprestanda borde man ha en regulator för varje varvtal. Detta kan uppnås genom att styra regulatorparametrarna efter varvtalet. Därigenom kan man anpassa regulatorn så att den blir optimal. Praktiskt genomförs detta genom att i tabellen för Tn, också lägga in värden på regulatorparametrarna. Som regulator kan en PI-regulator användas, men eftersom tidsfördröjningen är ganska lång bör man införa dödtidskompensering. Den Otto-Smith regulator som då fås skall även innehålla en modell av processen. Denna kan erhållas genom mätning av luftflöde och varvtal. Om bandbredden hos det slutna systemet väljs för stor innebär det att regulatorn blir känslig för variationer i ktc och ktd. Eftersom A/F beror på Ttot som 1/Ttot måste en olinjäritet u=1/v läggas in vid utsignalen från regulatorn (figur 6).

## SJÄLVINSTÄLLANDE REGULATOR

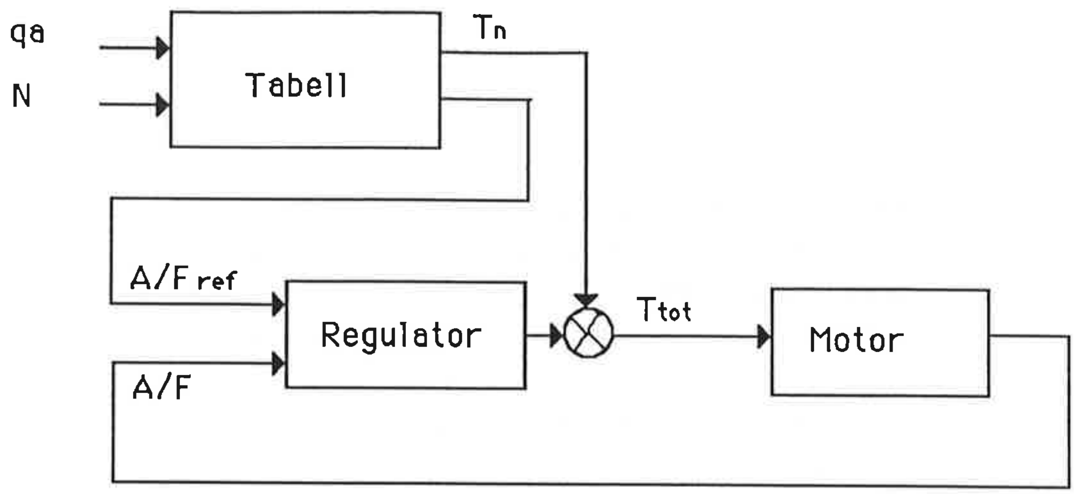
Istället för att ändra regulatorparametrarna genom att känna av varvtalet, kan man låta regulatorn ställa in sig själv genom att skatta Tc, Td samt ß. För att hinna med i processens ändringar måste samplingshastigheten väljas ganska hög. Det ger upphov till stora gradtal i polynomen, eftersom tidsfördröjningen blir många sampel. Det är en nackdel som gör metoden mindre attraktiv.

Ett alternativt sätt att göra en RST-regulator är att låta samplingshastigheten vara lika med tidsfördröjningen i motorn. Därigenom slipper man de höga gradtalen i polynomen. Dessutom innebär det att den tidsdiskreta modellen av processen blir oberoende av hastigheten, vilket innebär att skattningen av Tc och Td behöver göras (förutsatt att ktc och ktd är kända). Tyvärr visar simuleringarna att denna metod inte fungerar så bra, pga att samplingstiden blir för lång i förhållande till tidskonstanten Tc.

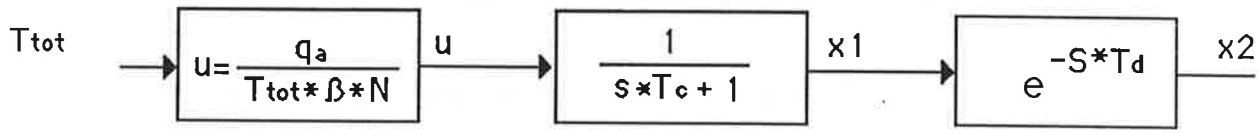
## UPPDATERING AV TABELLEN

För att kompensera för förändringar i luftmätaren och i insprutarna kan man uppdatera tabellen för Tn. Detta kan ske genom att mäta olika signaler (tex Tn, qa, N, Ttot) i stationäritet, under acceleration och vid bromsning. Genom att jämföra de olika signalernas värden vid de olika fallen, kan man uppdatera tabellen.

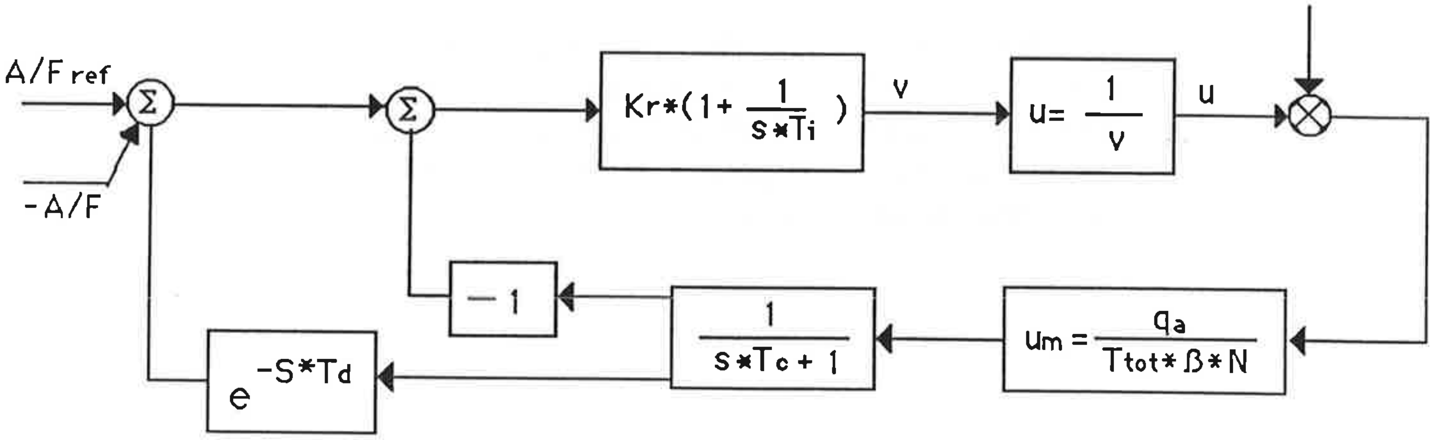
Tex vid stationäritet skall Ttot/Tn vara ett visst värde. Om ß ändras medför det att Ttot/Tn ändras och värdet på Tn för den körsituationen kan uppdateras. Försök enligt referens 2 visar att detta minskar avgaserna märkbart.




Figur 4  Reglering med syresensor




Figur 5   Motormodell




Figur 6   Regulator

# SIMULERING

Jag har gjort simuleringar i Simnon av:

* Open-loop beräkning av insprutningstiden
* Konstant PI-reglering med dödtidskompensering
* Parameterstyrd PI-reglering med dödtidskomp.
* RST-regulator med samplingstid=fördröjningen

Jag har använt den tidigare beskrivna modellen med ktc=175 och ktd=285.

Den konstanta PI-regulatorn är optimal vid 1500 rpm men har gjorts tämligen långsam för att fungera vid alla varvtal, även med variationer i ktc och ktd.

Parametrarna i den parameterstyrda PI-regulatorn har delats in i 4 olika klasser med gränser vid 800, 1500 och 3000 rpm. Kr och Ti har valts så att slutna systemet är ungefär lika snabbt som processen. Orsaken till att det inte är snabbare är att regulatorn ska klara av ändringar i ktc och ktd, samt att styrsignalen inte ska ha för snabba stora ändringar.

RST-regulatorns polynom har valts så att den är ungefär 5 gånger snabbare än processen.

Simnonblocken heter mixing, reg, tabell och conproj.

Som insignal har använts 2 olika varvtal, 600 och 6000 rpm. Efter en viss tid (4 alt 1.5 sekunder) kommer en stegändring i luftflödet som ger upphov till en stegändring i referensvärdet. Något senare kommer en negativ bränsletransient (medför positiv A/F transient) som ska symbolisera en tämligen snabb acceleration.

## SIMULERINGSRESULTAT OCH SLUTSATSER

Simuleringarna visar tydligt fördelarna med att reglera insprutningstiden.

Eftersom den parameterstyrde regulatorn är avpassad för varje varvtal är den snabbare och tar bort transienten bättre än den konstanta regulatorn. Även variationer i ktc och ktd klaras bättre av med parameterstyrd regulator.

RST-regulatorn fungerar skapligt vid höga varvtal, beroende på att samplingsintervallet är mindre då. Vid låga varvtal blir uppträdandet sämre eftersom det tar lång tid för den att reagera.

Slutsatsen man kan dra av simuleringarna är att man kan förbättra insprutningssystemet genom att använda en syregivare och reglera med parameterstyrning. Eventuellt kan det kompletteras med uppdatering av tabellen.

1 = A/F open-loop

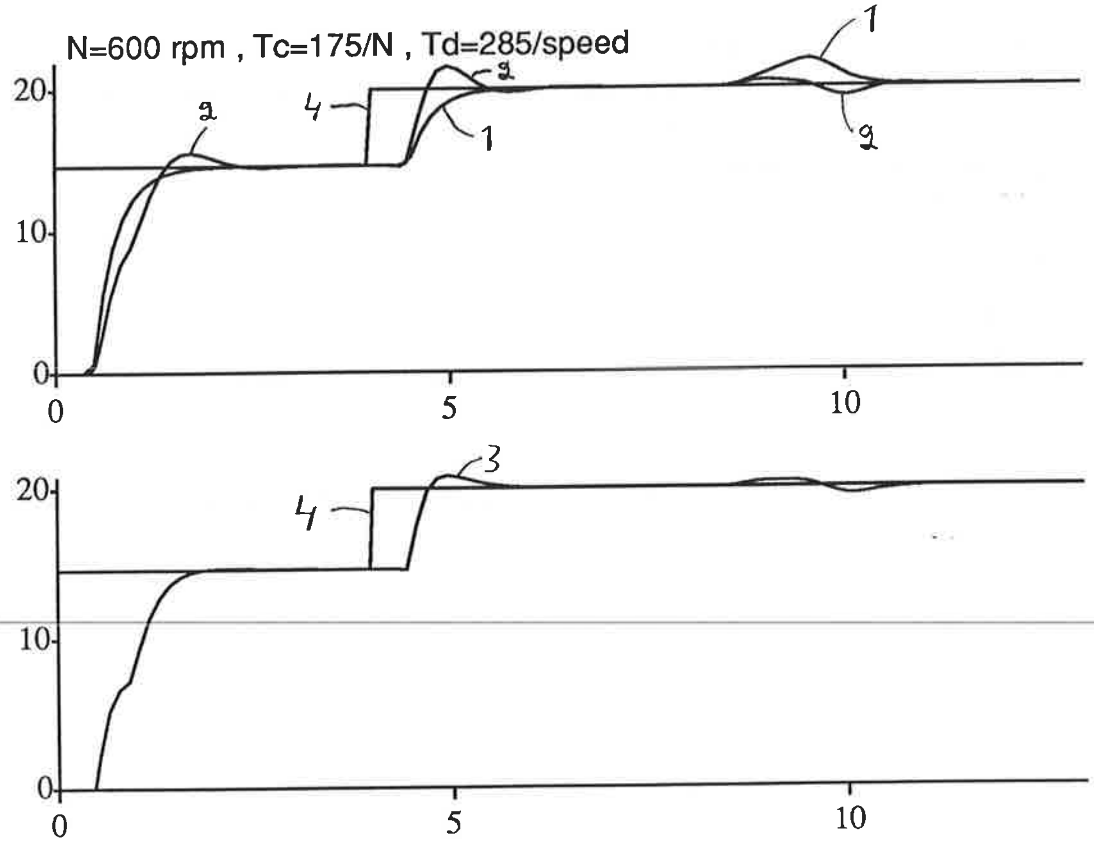
2 = A/F konstant PI-regulator

3 = A/F parameterstyrd PI-regulator

4 = A/F ref

89.03.02 - 11:05:39 nr: 1
hcopy meta/pro

N=600 rpm , Tc=175/N , Td=285/speed



1 = A/F open-loop

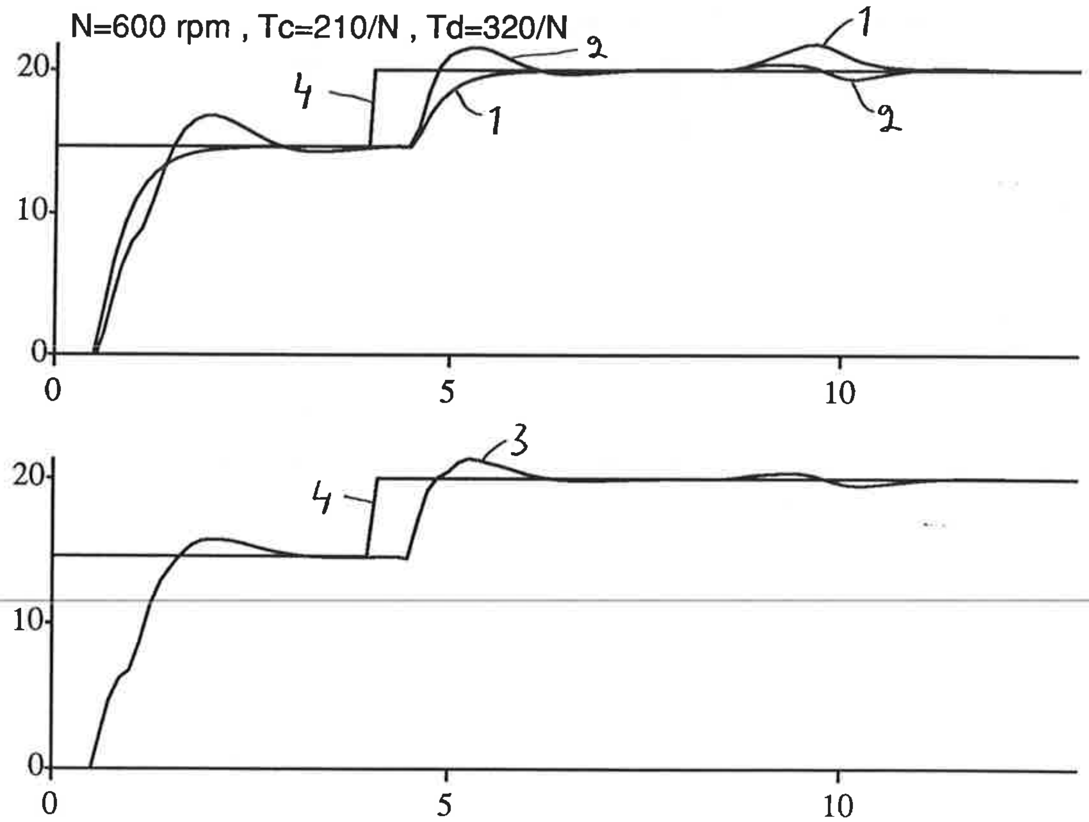
2 = A/F konstant PI

3 = A/F parameterstyrd PI

4 = A/F ref

89.03.02 - 11:33:31 nr: 5
hcopy meta/pro1

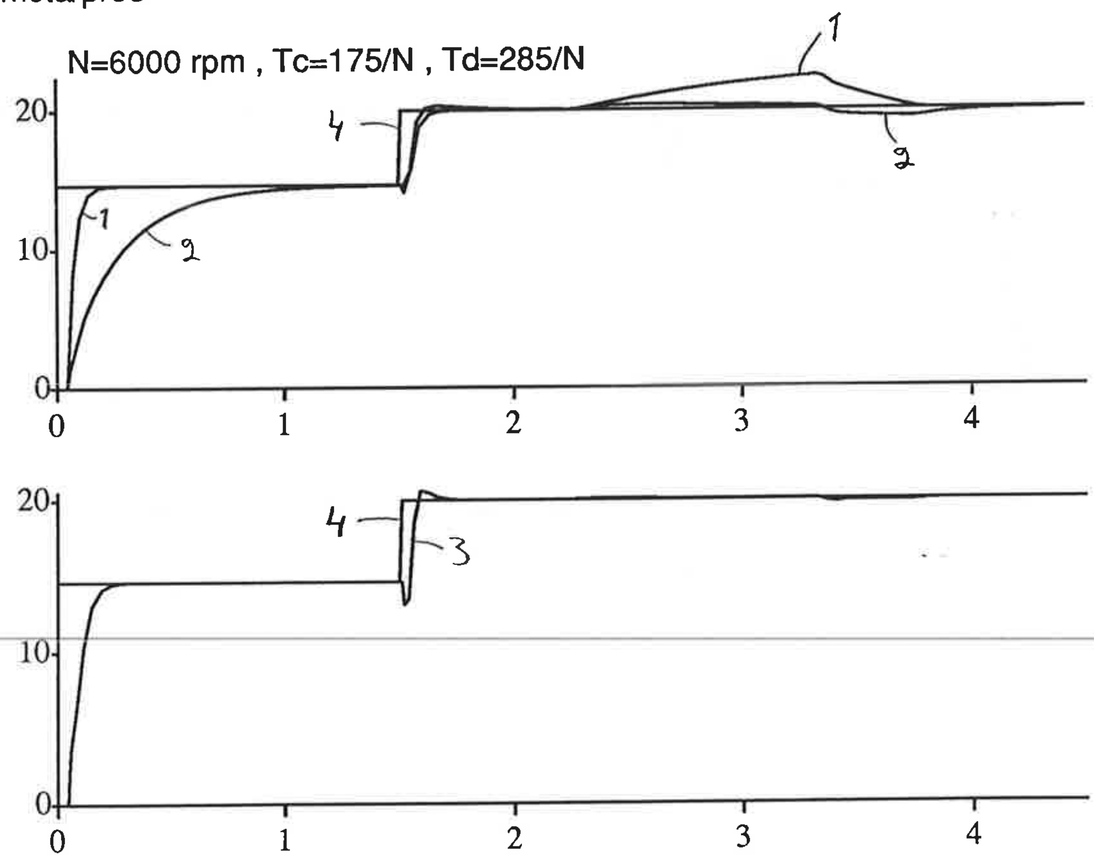
N=600 rpm , Tc=210/N , Td=320/N



1 = A/F open-loop
2 = A/F konstant PI
3 = A/F parameterstyrd PI
4 = A/F ref

89.03.02 - 11:29:07 nr: 4
hcopy meta/pro3

N=6000 rpm , Tc=175/N , Td=285/N



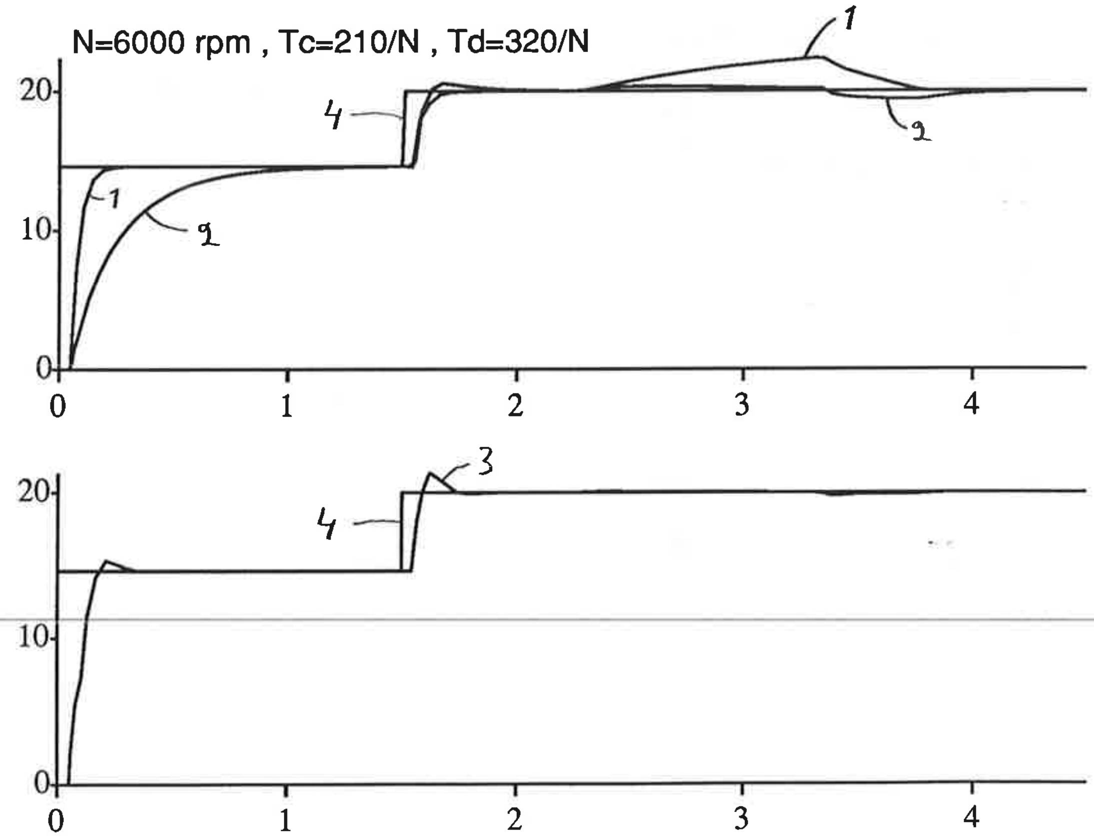
1 = A/F open-loop

2 = A/F konstant PI

3 = A/F parameterstyrd PI

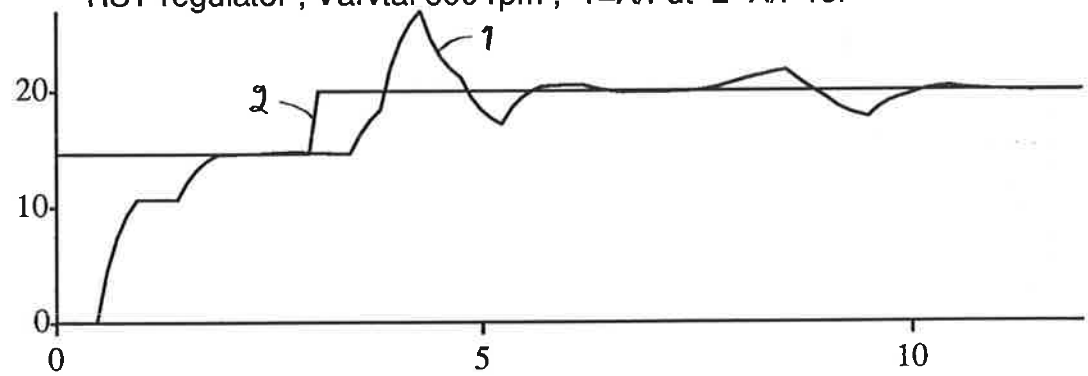
4 = A/F ref

89.03.02 - 11:27:01 nr: 3
hcopy meta/pro2

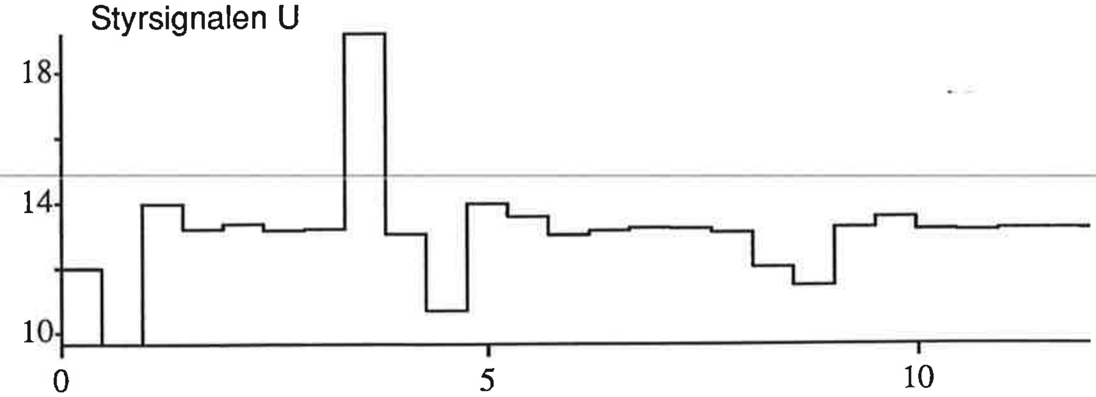


89.03.01 - 16:05:49 nr: 6
hcopy meta/pro

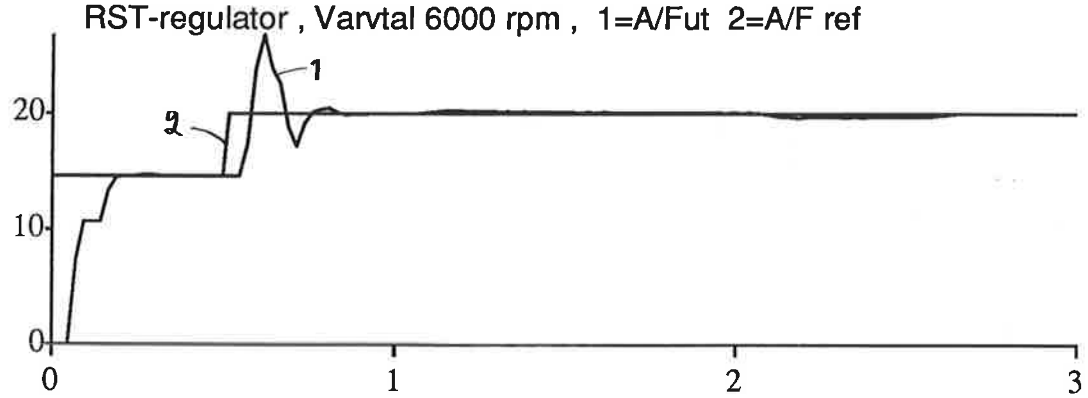
RST-regulator , Varvtal 600 rpm , 1=A/Fut 2=A/F ref

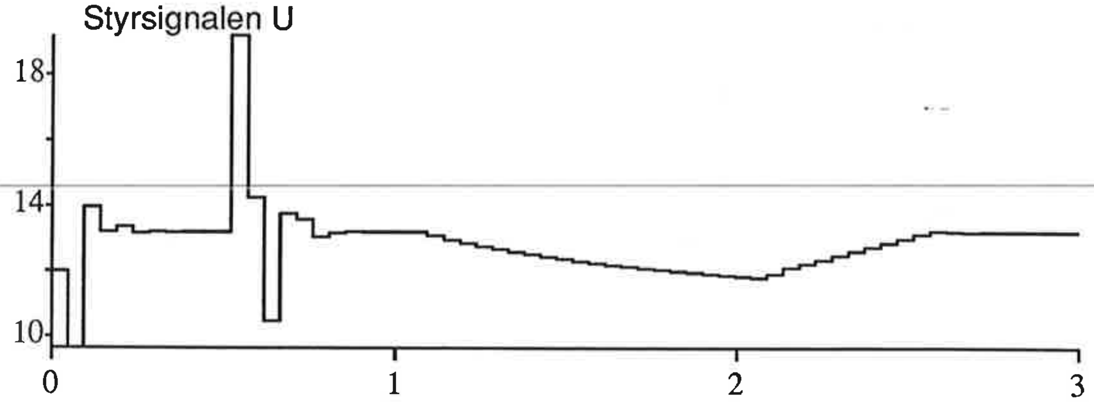



Styrsignalen U



89.03.01 - 16:01:59 nr: 5
hcopy meta/pro1



```
continuous system tabell
input speed qa d
output tn
kc=if qa/speed>0.04 then 1.366 else 1
tn=qa*kc/speed+d
end
continuous system mixing
input u speed qa tn
output y
state x
der dx
tc=tkc/speed
v=u*qa/(tn*speed*beta)
dx=(v-x)/tc
y=x
beta:0.66
tkc:175
end
```

PI-regulator med dödtidskompensering

```
continuous system reg
input ref y speed tn xdel qa
output u xd
state i x
der di dx
time t
kr=if konst<0.5 then 0.8 else kk
tih1=if speed<3000 then 0.067 else 0.03
tih2=if speed<800 then 0.23 else if speed<1500 then 0.13 else tih1
ti=if konst<0.5 then tih2 else tik
dx=(u*qa/(speed*beta*tn)-x)*speed/175
xd=x
di=ref-y-x+xdel
u=if open<0.5 then kr*(ref-y-x+xdel+i/ti) else 13.2
open:0          "open=1 ger openloop
konst:0         "konst=1 ger konstant regulator
kk:0.26
tik:0.06
beta:0.66
end
```

Connecting system för PI-regulator

```
connecting system conproj
time t
speed=n
qlow=speed*0.02
qtr=speed*0.018*(t-ttr)+qlow
qm=speed*0.038
qhigh=speed*0.068
qh=if t<ttr1 then qtr else qm
airflow=if t<tc then qhigh else if t<ttr then qlow else qh
td=tkd/speed
```

```
u1ÄdelayÅ=yÄmixingÅ
yÄregÅ=if t<td then yÄmixingÅ else y1ÄdelayÅ
speedÄregÅ=speed
uÄmixingÅ=uÄregÅ
speedÄmixingÅ=speed
qaÄmixingÅ=airflow
tnÄmixingÅ=tnÄtabellÅ
speedÄtabellÅ=speed
qaÄtabellÅ=airflow
tdr=285/speed
td2ÄdelayÅ=t-tdr
u2ÄdelayÅ=xdÄregÅ
xdelÄregÅ=y2ÄdelayÅ
qaÄregÅ=airflow
tnÄregÅ=tnÄtabellÅ
trans=ttopp*(t-ttr)
trans2=ttopp*(ttr+1-t)*2+ttopp
transh=if t<ttr2 then trans2 else 0
dÄtabellÅ=if t<ttr then 0 else if t<ttr1 then trans else transh
ttopp:0
tkd:285
n:600         "=varvtalet
tc:4          "=tiden för steget
ttr:8         "=tiden för transienten
ttr1:9
ttr2:9.5
end
```

connecting system for STR

```
connecting system conproj
time t
speed=n
qlow=speed*0.02
qtr=speed*0.018*(t-ttr)+qlow
qm=speed*0.038
qhigh=speed*0.068
qh=if t<ttr1 then qtr else qm
airflow=if t<tc then qhigh else if t<ttr then qlow else qh
td=285/speed
td1ÄdelayÅ=t-td
ucÄregÅ=if airflow/speed < 0.04 then 20 else 14.64
u1ÄdelayÅ=yÄmixingÅ
yÄregÅ=if t<td then yÄmixingÅ else y1ÄdelayÅ
speedÄregÅ=speed
uÄmixingÅ=uÄregÅ
speedÄmixingÅ=speed
qaÄmixingÅ=airflow
tnÄmixingÅ=tnÄtabellÅ
speedÄtabellÅ=speed
qaÄtabellÅ=airflow
trans=ttopp*(t-ttr)
trans2=ttopp*(ttr+1-t)*2+ttopp
transh=if t<ttr2 then trans2 else 0
dÄtabellÅ=if t<ttr then 0 else if t<ttr1 then trans else transh
ttopp:0
n:600
tc:4
ttr:8
ttr1:9
ttr2:9.5
end
```

119.

STR

```
discrete [illegible]
input y uc s[illegible]
output u
state y1 ydf uc1 u[illegible] f2 th1 th2 p11 p12 p22
new ny1 nydf nuc1 nu1 nu2 nf1 nf2 nth1 nth2 n11 n12 n22
time t
tsamp ts
nydf=af*ydf+y-y1
e=nydf-f1*th1-f2*th2
k1=p11*f1+p12*f2
k2=p12*f1+p22*f2
den=lambda+f1*k1+f2*k2
nth1=th1+k1*e/den
nth2=th2+k2*e/den
n11=(p11-k1*k1/den)/lambda
n12=(p12-k1*k2/den)/lambda
n22=(p22-k2*k2/den)/lambda
ny1=y
nuc1=uc
nu1=u
nu2=u1
nf1=af*f1+y-y1
nf2=af*f2+u1-u2
am=exp(-alfam*h)
bm=(1-am)*betam/alfam
a=th1
b=th2
r1=1+a-am
s0=(1+a-a0-am)/b
s1=(a0*am-a)/b
t0=bm/b
v=t0*uc-s0*y-s1*y1-(r1-1)*u1+r1*u2
u=if v<ulow then ulow else if v<uhigh then v else uhigh
ts=t+h
alfam=w*speed/175
betam=w*speed/175
a0:0
h=285/speed
af:0.2
lambda:0.98
uhigh:100
ulow:-100
th1:0.196
th2:1.22
p11:10
p22:10
w:2.5
end
```

## REFERENSER

[1] Powell, Wu, Aquino : Stoichiometric air-fuel ratio control analysis

[2] Sasamaya m.fl. : An experimental investigation of an electronic control system using new control algorithms

/21.

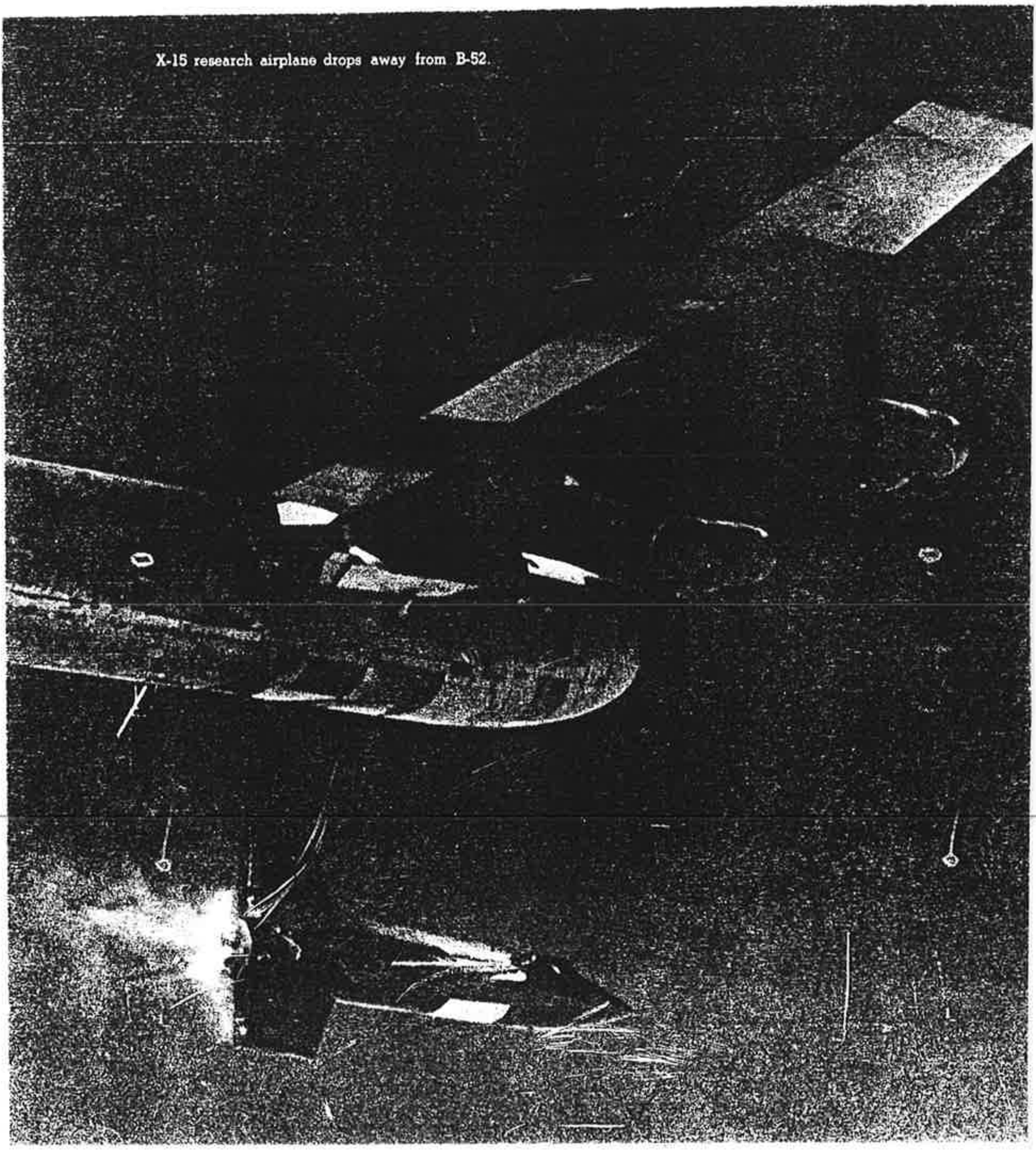
# Adaptiv Reglering
# X-15
och andra glassiga saker


av

Allan Gustavsson

och

Anders GM Dahlberg

# Inledning

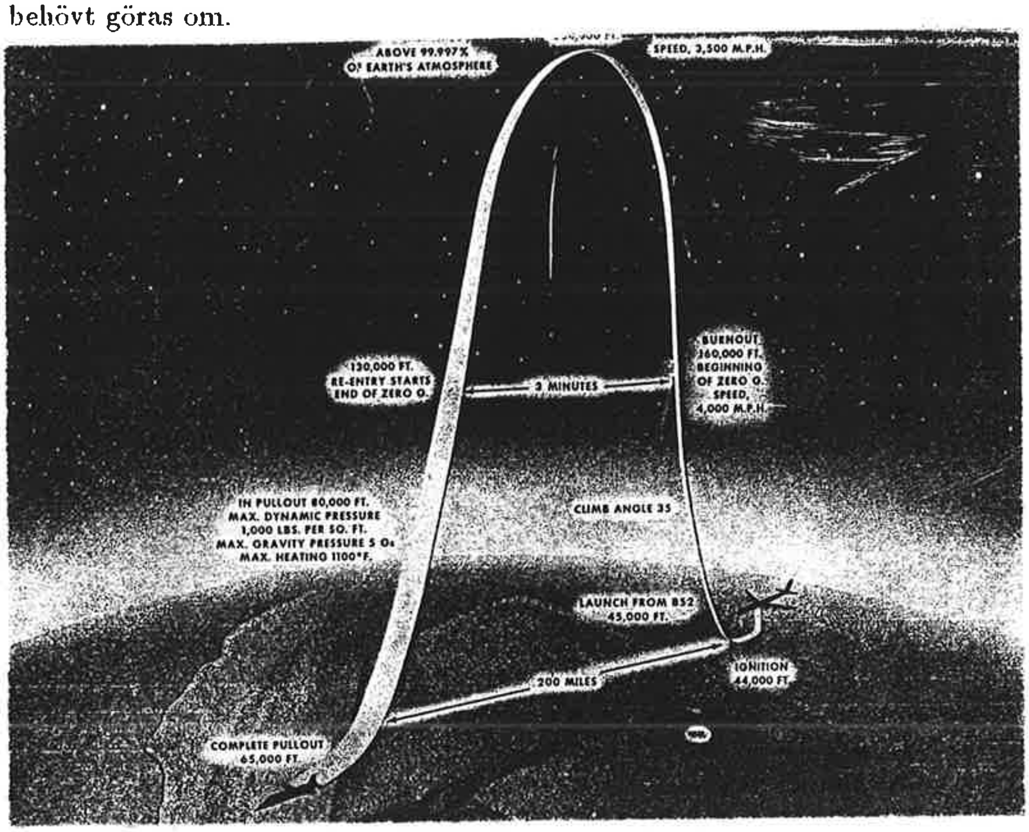
X-15 var ett raketdrivet experimentflygplan som användes vid höghöjdsflygningar för systemtester som förberedelse för rymdprogrammet. Ett av de system som testades var ett adaptivt styrsystem. Det är vad denna rapport kommer att handla om. X-15 programmet inleddes 1957 och avslutades 1967 då den 65:e flygningen slutade i en katastrof.

Flygplanet, som var 15 meter långt och hade en spännvidd på 7 meter, drevs av en raketmotor med 250kN dragkraft. X-15 var konstruerat för en topphastighet på sex gånger ljudhastigheten och en maximal höjd av 90.000 meter.

En normal flygning tillgick så att X-15 släpptes på 13.000m från en B-52, steg till 76.000m, återinträdde i atmosfären och landade 32 mil från utgångsplatsen, allt under 12 minuter.

Varför ville man ha ett adaptivt styrsystem? Ett flygplans reaktioner på roderutslag beror kraftigt på höjd och hastighet. Detta löses normalt sett med parameterstyrning, dvs att förstärkningen i styrsystemet ändras efter luftdata enligt ett förutbestämt schema.

För ett så extremt flygplan som X-15 var naturligtvis dessa variationer ovanligt stora, på toppen av banan där endast ca 0,003 % av atmosfären återstod hade rodren ingen verkan alls utan styrningen fick ske med styrraketer. Man ville därför undvika gain scheduling som hade lett till att en mycket stor mängd experiment hade måst göras. Dessutom skulle ett väl fungerande adaptivt system kunna användas även till andra flygplan utan att experimenten behövt göras om.




**Figur 1.** Exempel på en X-15 flygning.

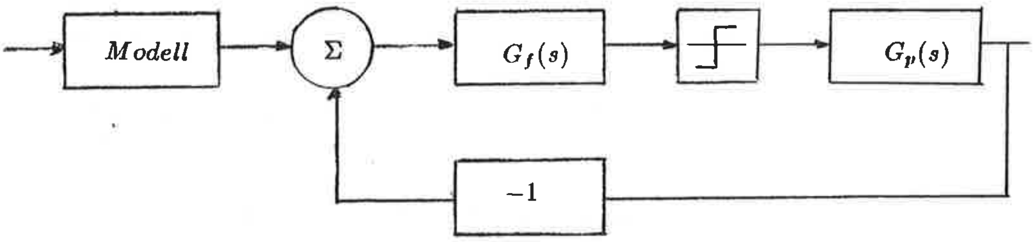
Målet för systemet var alltså att, ur pilotens synvinkel, planet alltid skulle uppföra sig likadant, såväl på 76.000 som 76 meters höjd.

# SOAS

Det adaptiva system som användes inom X-15 projektet var ett så kallat Self Oscillating Adaptive System, SOAS. Grundtanken i detta är att återkoppling med en hög förstärkning ger ett slutet system, okänsligt för parametervariationer i processen. SOAS bygger på två huvudidéer, modellföljning och automatisk generering av testsignaler genom den höga förstärkningen. I X-15 projektet användes i huvudsak tre varianter av SOAS.

### Reläsvängningsmetoden

Denna metod, som provades i F-94 flygplanen, bygger på att en gränssvängning induceras genom ett relä.




**Figur 2.** Grundprincip för reläsvängningsmetoden.

Först väljs reläamplituden $d$ med hänsyn till de prestandakrav man ställer på systemet. Sedan bestämmer man sig för den amplitud $e_0$ på svängningen som kan tolereras. Detta ger svängningens vinkelfrekvens $\omega_l$. Dock måste man se till att $\omega_l$ ligger inom systemets bandbredd, men ändå tillräckligt högt.

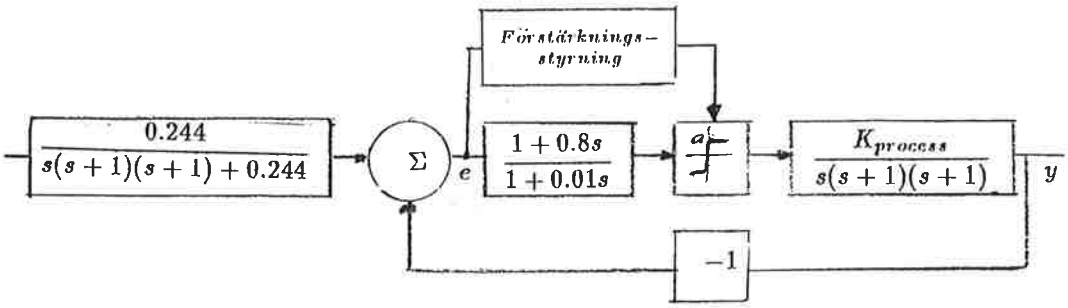
$$d|G_p(i\omega_l)| = e_o$$

För att få en svängning vid denna frekvens väljs ett fashöjningsfilter $G_f$ så att det slutna systemets fasvridning blir $-\pi$.

$$argG_f(i\omega_l) + argG_p(i\omega_l) = -\pi$$

Dock krävs att det slutna systemets bandbredd är större än processens (flygplanets) naturliga frekvens. Detta för att processens fasbidrag vid reläsvängningsfrekvensen skall vara praktiskt taget konstant trots parametervariationer i processdynamiken.

För att begränsa svängningarna kan det vara lämpligt att kunna variera reläamplituden. Detta användes i F-94C flygplanen och vi har simulerat det i ett förenklat system. Detta system har konstruerats av Per Hagander och illustrerar hur reläamplituden varieras i olika situationer. Programkoden redovisas i appendix A. Förstärkningen i processen varierar, vilket kan representera de skiftande aerodynamiska förhållandena vid olika höjder och hastigheter. Under

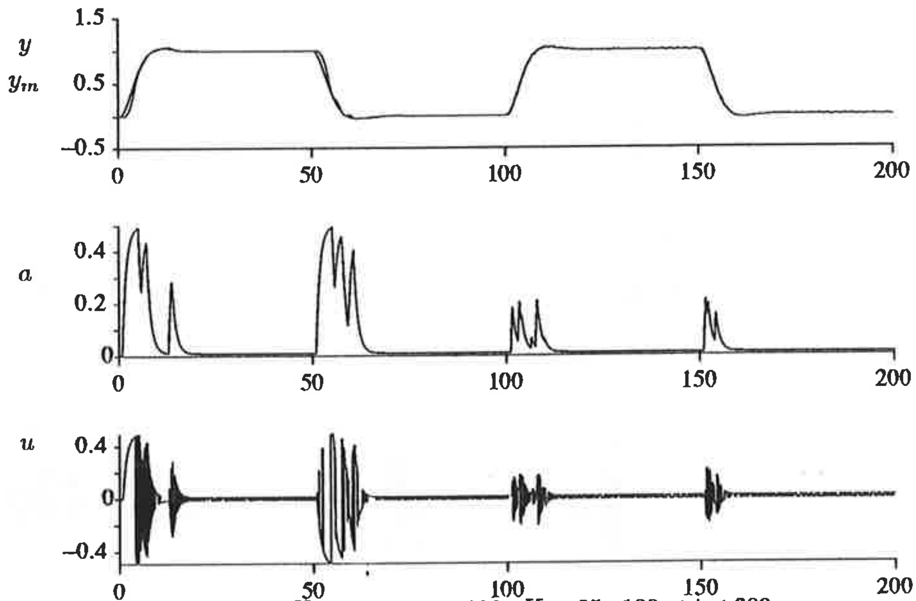



**Figur 3.** Förenklad modell med variabel reläamplitud.

en flygning kan denna förstärkning variera med en faktor 20. Reläamplituden $a$ ändras enligt

$$\frac{da}{dt} = \begin{cases} -a + a_{max} & |e| > e_0 \\ -a + a_{min} & |e| < e_0 \end{cases}$$

Som insignal har till samtliga simuleringar använts en fyrkantvåg. I figur 4 och 5 har förstärkningen plötsligt ändrats vid tiden $t = 100$ med en faktor 25 respektive 50. Trots detta följer processen modellen mycket väl.

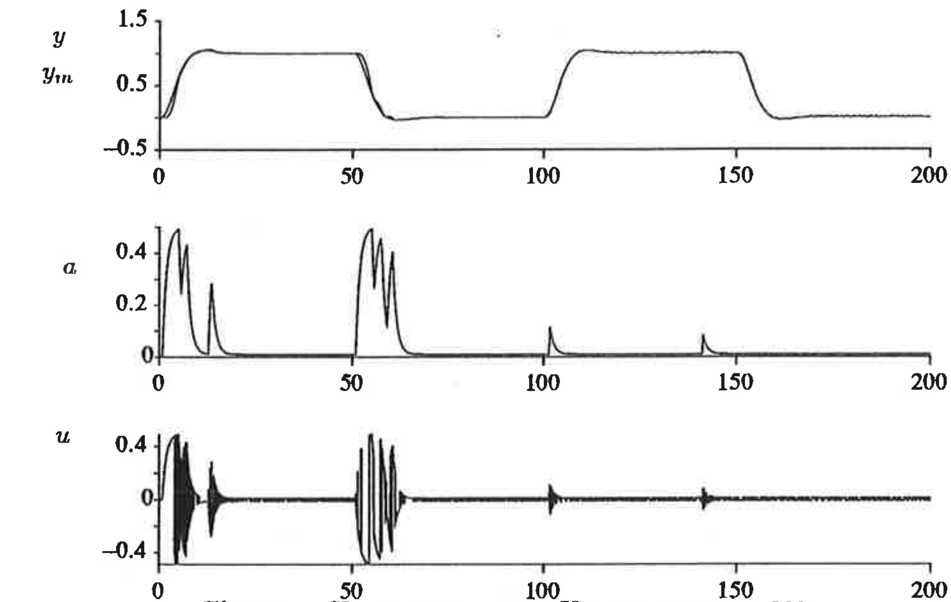



**Figur 4.** $K = 1$; $0 < t < 100$, $K = 25$; $100 < t < 200$

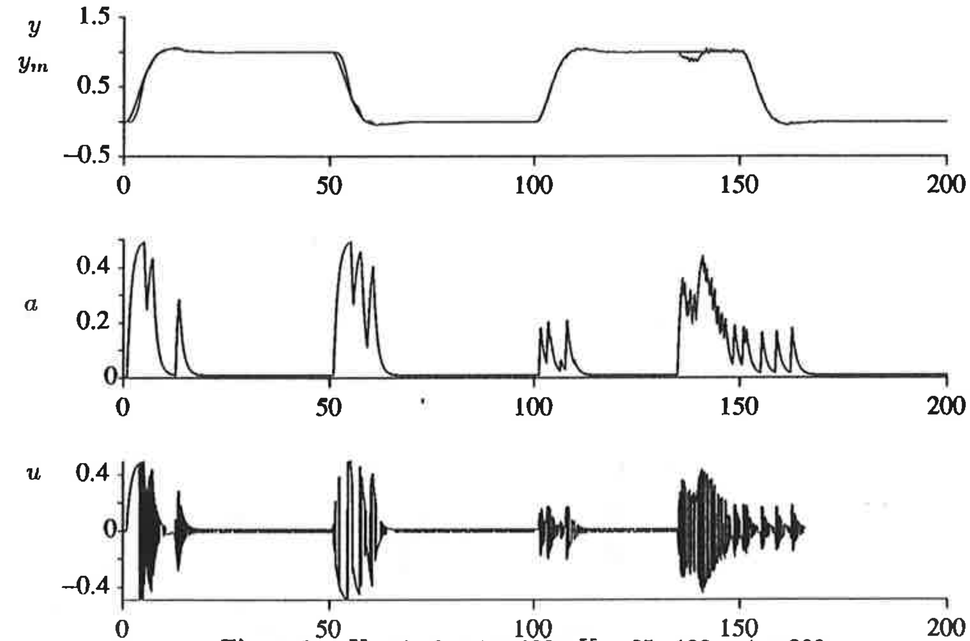
I figur 6 har processen störts mellan $t = 135$ och $t = 140$ (luftgrop !). Detta fungerar bra, även om en liten störning uppstår på grund av den plötsligt höjda reläamplituden.

Figur 7 visar vad som händer då systemet störs av en högfrekvent sinussignal. Trots att störningens amplitud är så stor som 10% av insignalen klarar systemet störningen väl åtminstone vid den högre förstärkningen, efter tiden $t = 50$.

För att riktigt plåga systemet störde vi det, i figur 8, med en fyrkantvåg med amplitud 0.1 och period $5s$. Även detta klarar systemet förvånansvärt väl.




**Figur 5.** $K = 1$; $0 < t < 100$, $K = 50$; $100 < t < 200$

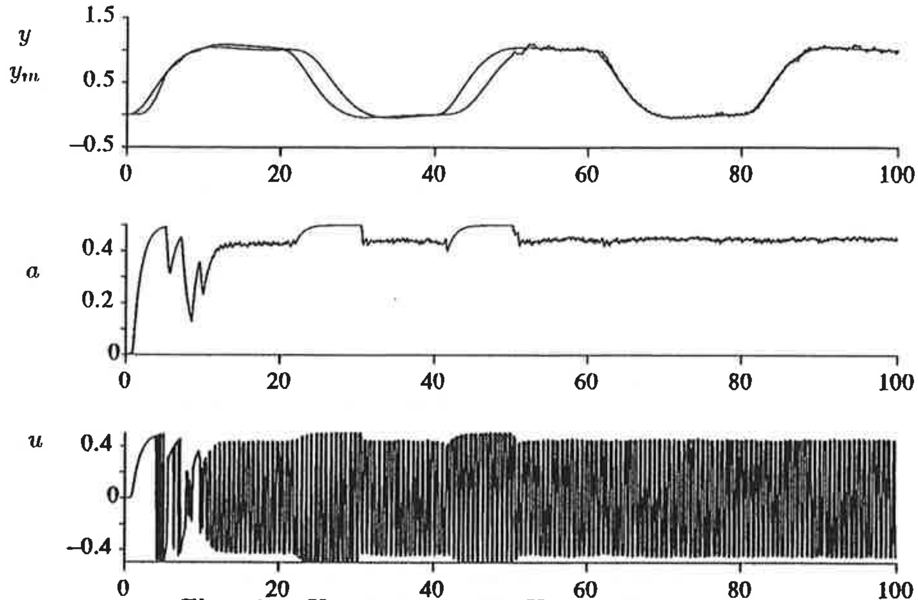



**Figur 6.** $K = 1$; $0 < t < 100$, $K = 25$; $100 < t < 200$

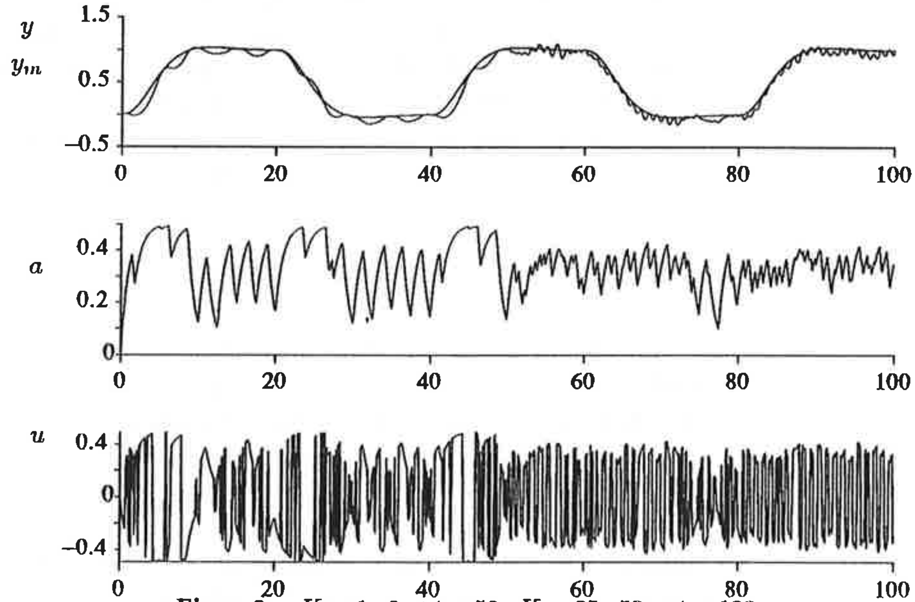
## Frekvensstyrd förstärkningsfaktor

Detta system konstruerades av General Electric och användes vid några tidiga simuleringar men flög aldrig. Systemet använder inget relä utan bygger endast på en hög variabel förstärkning. Förstärkningen hålls hela tiden vid gränsen till instabilitet genom att den uppkomna svängningens frekvens jämförs med en referensfrekvens, så att om frekvensen är högre än referensen minskas förstärkningen och tvärtom. En detalj som roar oss barn av dataåldern är att den variabla förstärkningen implementerades med en elmotor och en potentiometer !

126.



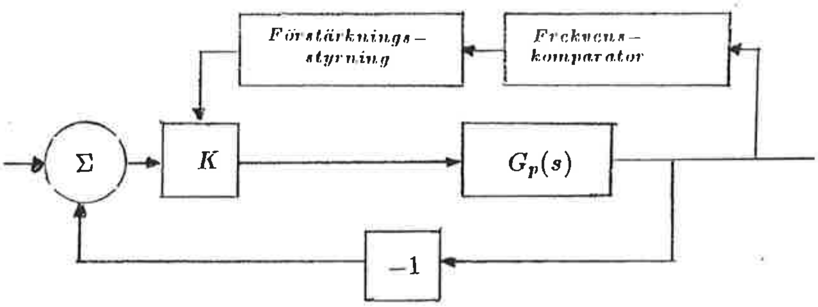

**Figur 7.** $K = 1;\ 0 < t < 50,\ K = 25;\ 50 < t < 100$




**Figur 8.** $K = 1;\ 0 < t < 50,\ K = 25;\ 50 < t < 100$

## Amplitudstyrd förstärkningsfaktor

Det system som kom att användas i X-15 byggdes av Honeywell och baserade sig på samma tanke som General Electrics, men kontrollerade amplituden på svängningen istället för frekvensen. Styrsignalen till rodren filtrerades genom ett bandpassfilter, varpå den likriktades och jämfördes med en referensnivå. Skillnaden integrerades och styrde den variabla förstärkningen.




**Figur 9.** General Electrics system.

## Problem

Några av de problem som uppstod vid flygningarna kan vara intressanta att notera som exempel på de svårigheter som uppstår när man praktiskt skall tillämpa sina teorier.

Störsignaler såsom elektriskt och aerodynamiskt brus eller pilotens styrkommandon (!) har ofta någon frekvenskomponent som går igenom bandpassfiltret och ökar amplituden hos den överlagrade svängningen varvid systemet svarar med att sänka förstärkningen vilket försämrar systemets prestanda.

Stora lågfrekventa signaler kan mätta kretsarna så att den överlagrade gränssvängningen försvinner, vilket systemet tolkar som att förstärkningen är för låg och höjer den över det kritiska värdet varpå systemet blir instabilt.

Ett annat "problem" var att man av säkerhetsskäl behöll det traditionella mekaniska eller hydrauliska styrsystemet jämte det elektriska. Detta innebar att piloten alltid kunde rent mekaniskt påverka planet. Av styrsystemet kom detta att tolkas som störningar. Lösningen som framfördes var ett rent "fly-by-wire" system som idag används bland annat på F-16 och JAS.

## Resultat

Sammanfattningsvis konstaterades att systemet fungerat bra, dock inte så mycket bättre än ett system med fast förstärkning, som man kunnat vänta sig, utom under vissa förhållanden till exempel att systemet användes för att automatiskt koppla in styrraketerna på höga höjder vilket gjorde att piloten hela tiden styrde med en och samma spak och med oförändrat resultat.

Det visade sig att många problem man väntat sig aldrig uppstod, livslängden hos de mekaniska komponenterna minskade inte märkbart trots de ständiga vibrationerna, och tillförlitligheten hos det komplicerade systemet var också mycket god. Den katastrof som avslutade programmet berodde på en kombination av tekniska fel och pilotens reaktioner som aldrig kunnat förutses.

# Epilog

X-15 programmet, och därmed mycket av forskningen på adaptiva system, avbröts efter en tragisk olycka under den 65:e flygningen. Ungefär en minut efter start uppträdde en elektrisk störning, troligen härrörande ifrån ett experiment i en vingkapsel, som kom att vid flera tillfällen kraftigt sänka förstärkningen. Detta hade troligen inte fått några svårare konsekvenser vid en vanlig atmosfärisk flygning eftersom planet var kontrollerbart även med minimal förstärkning. Nu ledde dock förstärkningsminskningen till att styrraketerna kopplades ur. Detta märkte till att börja med inte piloten. Strax efter att topphöjden hade nåtts uppfattade piloten problemet och övergick till manuell styrning. Dock måste han ha misstolkat instrumenten och vände flygplanet 90 grader i förhållande till flygriktningen vilket efter ett tag ledde till att flygplanet gick i spinn. Piloten försökte häva spinnen med både roder och styrraketer, och den automatiska girdämparen kopplade också in styrraketerna för att häva spinnen. Dock kopplades dessa ur igen efter ca 10 sekunder då den höga rotationshastigheten tolkades som ett gyrofel. På 36.000m gick planet ur spinn genom en kombination av styrning och flygplanets aerodynamiska stabilitet. Trots det kraftigt ökade dynamiska trycket sänktes inte förstärkningen som den borde vilket ledde till instabilitet i höjdroderservot, varvid höjdrodren inledde en sågtandssvängning med 20 graders amplitud. Då samma roder användes som både höjd- och skevroder förlorades också kontrollen i rolled genom mättningen av höjdroderstyrningen. De svängningar som då uppstod ledde slutligen till att flygplanet slets sönder på 18.000 meters höjd. Simuleringar visade senare att svängningarna kunnat hävas om automatiken kopplats ur. Anledningen till att piloten inte gjorde detta var troligen att ingen indikation gavs på de fel som uppträdde.

# Appendix A

```
CONTINUOUS SYSTEM MODELL
STATE x1 x2 x3
DER dx1 dx2 dx3
INPUT ur
dx1 = -x1+kp*(ur-x3)
dx2 = -x2 +x1
dx3 = x2
kp : 0.244
END

CONTINUOUS SYSTEM LEADLAG
INPUT u
OUTPUT y
STATE x
DER dx
y = (x+a*u)/b
dx = (-x+(b-a)*u)/b
a : 0.8
b : 0.01
END

CONTINUOUS SYSTEM RELAY
INPUT e u
STATE a
DER da
OUTPUT y
y = a*sign(u)
ain=IF ABS(e)>eo THEN a1 ELSE a2
t=IF ABS(e)>eo THEN t1 ELSE t2
da =(-a+ain)/t
t1 : 1
t2 : 1
a1 : 0.5
a2 : 0.01
eo : 0.1
END
```

```
CONTINUOUS SYSTEM GS
STATE x1 x2 x3
DER dx1 dx2 dx3
INPUT u
dx1=-x1+u
dx2=-x2+x1
dx3=x2
END


CONNECTING SYSTEM SOAS
TIME t
ur[modell]=amp*(IF MOD(t,tr)>tr/2 THEN 0 ELSE 1)
pi : 3.1415926535
tr : 40
amp : 1
e[relay] = x3[modell]-x3[gs]-noise
u[leadlag] = e[relay]
u[relay] = y[leadlag]
k=IF t<50 THEN k0 ELSE k1
k0 : 1
k1 : 25
tck : 100
u[gs] = y[relay]*k
namp : 0.1
noise = namp*(IF MOD(t,5)>2.5 THEN 0 ELSE 1)
END
```

131.

# Adaptiv reglering

# Adaptiv reglering av flexibelt servo

Lars Carlsson E-85
Anders Isberg D-85
Thomas Letterkrantz E-85
Lund maj 1989

/32.

# Innehållsförteckning

# 1 Inledning

I kursen adaptiv reglering ingår som en del av kursen att genomföra ett mindre projekt. Detta projektet går ut på att konstruera och implementera en adaptiv regulator i simuleringsmiljö (programpaketet SIMNON). Den process som ska adapteras och regleras är en liksrtrömsmotor som är kopplad med en vek axel till en stor massa. I rapporten har antagits att motorns dynamik är försumbar i förhållande till det mekaniska systemet. Vidare har antagits att inga förluster finns i den mekaniska delen av systemet, d v s att alla dämptermer där är satta till noll. Genom att göra detta antagande erhålls en mycket enkel modell för processen. Syftet med denna rapport är bl a att utreda om en sådan förenklad modell kan användas för den verkliga processen för att designa en indirekt självinställande tidsdiskret regulator.

Kapitel 2 tar upp modeller för den process som beskrivits ovan. Modellerna ligger sedan till grund för design av den adaptiva regulatorn.

# 2 Modeller

Avsnitt 2.1 beskriver en relativt fullständig modell för processen. Denna modell förenklas sedan i avsnitt 2.2. I avsnitt 2.3 behandlas den samplade modellen.

## 2.1 Lineär modell

Låt variabler relaterade till motorn indexeras med m och lasten med l. Rotorn har tröghetsmomentet $J_m$ och lasten $J_l$. Massorna är sammankopplade med en vek axel som beskrivs med en fjäderkonstant k och dämptermen d. Dämpning i motor och last noters med $d_m$ respektive $d_l$. Likströmsmotorn är separatmagnetiserad och således genereras ett konstant magnetiseringsflöde. I rotorkretsen finns en ledningsresistans $R_a$ och en lindningsinduktans $L_a$. $K_m$ är en motorkonstant, se figur 2.1. För en separatmagnetiserad likströmsmotor är den inducerade emk:n propotionell mot motorvarvtalet

$$E = k_E \cdot \omega \tag{2.1}$$

Motorns vridande moment ges av sambandet

$$M_m = K_m \cdot I_a \tag{2.2}$$

och dess dynamik beskrivs av

$$U = E + R_a \cdot I_a + L_a \cdot \frac{dI_a}{dt} \tag{2.3}$$

där U är matningspänningen och E den inducerade emk:n. Ekvation (2.1) i (2.3) ger sambandet

$$\frac{dI_a}{dt} = \frac{R_a}{L_a}\cdot I_a - \frac{k_E}{L_a}\cdot\omega + \frac{1}{L_a}\cdot U \tag{2.4}$$

Det mekaniska systemet beskrivs av differentialekvationerna

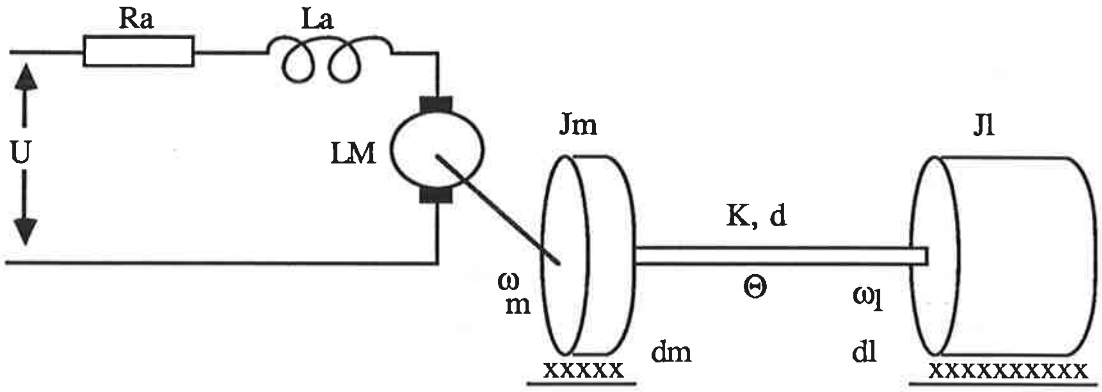
$$\frac{d\omega_m}{dt} = -\frac{k}{J_m}\cdot\Theta - \frac{d}{J_m}\cdot(\omega_m - \omega_l) - \frac{d_m}{J_m}\cdot\omega_m + \frac{K_m}{J_m}\cdot I_a \tag{2.5}$$

$$\frac{d\omega_m}{dt} = \frac{k}{J_l}\cdot\Theta + \frac{k}{J_l}\cdot(\omega_m - \omega_l) + \frac{d_l}{J_l}\cdot\omega_l - \frac{1}{J_l}\cdot M_l \tag{2.6}$$

och ur definitionen av läge och hastighet erhålls sambandet

$$\frac{d\Theta}{dt} = \omega_m - \omega_l \tag{2.7}$$

där Θ är torsionsvinkeln för axeln.




**Figur 2.1** Beskrivning av en relativt noggrann modell för en likströmsmotor som är sammankopplad till en stor massa via en vek axel.

## 2.2 Förenklad lineär modell

Om det mekaniska systemet är mycket vekt och likströmsmotorn är "stark", kan motorns dynamik försummas i förhållande till det mekaniska systemet. Om man dessutom antar att alla dämptermer är små kan dessa också försummas. Det kan verka att vara grova antaganden, men i många fall är inskränkningen som man gör liten. Se t ex examensarbetet [Carlsson,Isberg (1988)] eller projektarbetet [Carlsson,Isberg (1988)]. Modellen som då erhålls innehåller få parametrar, på tillståndsform kan den t ex beskrivas av

$$\begin{bmatrix} \dot{\omega}_m(t) \\ \dot{\omega}_l(t) \\ \dot{\Theta}(t) \end{bmatrix} = \begin{bmatrix} 0 & 0 & -k/J_m \\ 0 & 0 & k/J_l \\ 1 & -1 & 0 \end{bmatrix} \cdot \begin{bmatrix} \omega_m(t) \\ \omega_l(t) \\ \Theta(t) \end{bmatrix} + \begin{bmatrix} K_m/J_m \\ 0 \\ 0 \end{bmatrix} \cdot I_a(t) + \begin{bmatrix} 0 \\ -1/J_l \\ 0 \end{bmatrix} \cdot M_l(t) \tag{2.8}$$

och överföringsfunktionen blir för det förenklade fallet

$$G(s) = \frac{K_m}{J_m} \cdot \frac{s^2 + \frac{k}{J_l}}{s \cdot (s^2 + k \cdot \frac{J_m + J_l}{J_m \cdot J_l})} \tag{2.9}$$

d v s den innehåller endast tre parametrar att skatta [Wallenborg (1987)]. Om man hade försummat motorns dynamik, men tagit hänsyn till dämptermerna hade det blivit sex parametrar att skatta i stället, vilket hade medfört att den adaptiva regulatorn för det tidskontinuerliga fallet hade blivit betydligt mer komplicerad.

## 2.3 Tidsdiskret lineär modell

När man samplar en lineär tidskontinuerlig modell är det ofta svårt att utnyttja a priori kunskaper. Men om samplingsteoremet är uppfyllt för detta specialfall, kommer processens poler i det tidsdiskreta planet att projiceras på enhetscirkeln (De ligger på imaginära axeln i det tidskontinuerliga planet). Hur nollställen däremot hamnar är svårare att bestämma, men det visar sig att om samplingsfrekvensen är hög (väl över resonansfrekvensen) så hamnar de också på enhetscirkeln. (Även de ligger på imaginära axeln i det tidskontinuerliga planet). Den samplade processen får därför lyckligtvis också en enkel struktur

$$H(q) = K \cdot \frac{q^2 + b \cdot q + 1}{(q - 1) \cdot (q^2 + a \cdot q + 1)} \tag{2.10}$$

d v s även här finns det endast tre parametrar som behövs skattas. Det är en stor fördel om denna förenklade modell kan användas för att adaptera den beskrivna processen, eftersom realtidskraven ofta är höga och att det många gånger är önskvärt att processen samplas snabbt.

# 3 Skattare och regulator

I kapitel 3.1 beskrivs skattningen av parametrarna i processens överföringsfunktion, och i kapitel 3.2 används dessa för att beräkna en regulator enligt polplaceringsalgoritmen.

136.

### 3.1 Skattaren

Vi har den diskreta modellen

$$y(k)= \frac{b_1q^2+ b_2q+ b_3}{(q^2+ a_1q+ a_2)(q-1)} \tag{3.1}$$

Under antagandet att dämpningen i servot är noll så blir faktorn a2 = 1 i formel (1) .Dessutom kommer , under förutsättning om rimligt små sampeltider, nollställena att hamna på enhetscirkeln. Detta verifierades via Matlab, och innebär att parametrarna $b_1$ och $b_3$ blir lika. Totalt skulle vi då nu kunna skriva (3.1) som

$$(y(k+1) - y(k))(q^2+ a_1q+ 1) = b_1(u(k+2)+ u(k)) + b_2u(k+1) \tag{3.2}$$

eller

$$\partial y(k+2) + \partial y(k) = -a_1\partial y(k+1) + b_1(u(k+2) + u(k)) + b_2u(k+1) \tag{3.3}$$

där $\partial y(k) = y(k+1)- y(k)$.

Systemet är nu ordnat enligt modellen

$$y(t) = \varphi^T(t)*\theta \tag{3.4}$$

med samtliga helt kända eller mätbara signaler i ekvation (3.2) samlade i "y(t)". Skattningen av parametrarna sker nu enligt metoden med rekursiv minsta kvadrat (RLS) och exponentiell glömskefaktor, enligt [ Åström-Wittenmark (1989)].

För att i det verkliga fallet förhindra inverkan av bias eller högfrekvent brus filtreras alla signaler med ett bandpassfilter. Skattaren implementerades som ett separat block i Simnon , varur de skattade parametervärdena fås som utsignaler.

## 3.2 Regulator

De skattade parametrarna användes till design av en RST-regulator med hjälp av polplacering enligt [Åström-Wittenmark (1989)]. Vi har att lösa ekvationen

$$AR + BS = A_oA_mB^+ \tag{3.5}$$

Eftersom inga nollställen kan förkortas blir $B^- = B$ och $B^+ = 1$. Polerna läggs kring w=5 och dämpningen z= 0.7. Detta ger ett karakteristiskt polynom enligt

$$(q^2+ a_{m1}q+ a_{m2})(z+a_{m3}) \quad (3.6)$$

med

$a_{m1}= -2e^{-zwh}$

$a_{m2}= e^{-2zwh}$

$a_{m3}= 2e^{-wh}$

$B_m$ sätts till $B'_mB^-$, där $B'_m$ är en konstant, och eftersom vi vill ha integratorverkan i regulatorn så blir observeraren av tredje ordningen.

$$R(q)= (q-1)(q^2 + r_1q + r_2) \quad (3.7)$$

$$S(q)= s_0q^3 + s_1q^2 + s_2q + s_3 \quad (3.8)$$

Detta sättes in ekvation (3.4) , koefficienterna sätts lika i höger- och vänsterleden, och renderar till slut i ett sjätte ordningens ekvationssystem med de obekanta r1, r2, s0, s1, s2 och s3. En summarisk lösning på detta finns i Appendix.

Vid bestämning av T(q) utnyttjar vi att stationära förstärkningen ska vara lika med ett. Uttrycket för stationära förstärkningen blir

$$\frac{B(1)T(1)}{(A(1)R(1) + B(1)S(1))} = 1 \quad (3.9)$$

Eftersom R(q) innehåller en integratordel så blir R(1)= 0, och uttrycket (3.9) reduceras därmed till villkoret att

$$T(1)/S(1)= B'_m*Ao(1)/S(1)= 1 \quad (3.10)$$

det vill säga

$$B'_m= \Sigma s_i/\Sigma ao_i \quad (3.11)$$

Därmed erhålles det slutliga uttrycket för respektive koefficient i T(q)

$$ti= aoi*\Sigma si/\Sigma aoi \quad (3.12)$$

Parametrarna för R(q),S(q), respektive T(q) ges sedan som utsignaler från "ekvationslösningsblocket" till det block där sedan själva regulatorn finns implementerad.

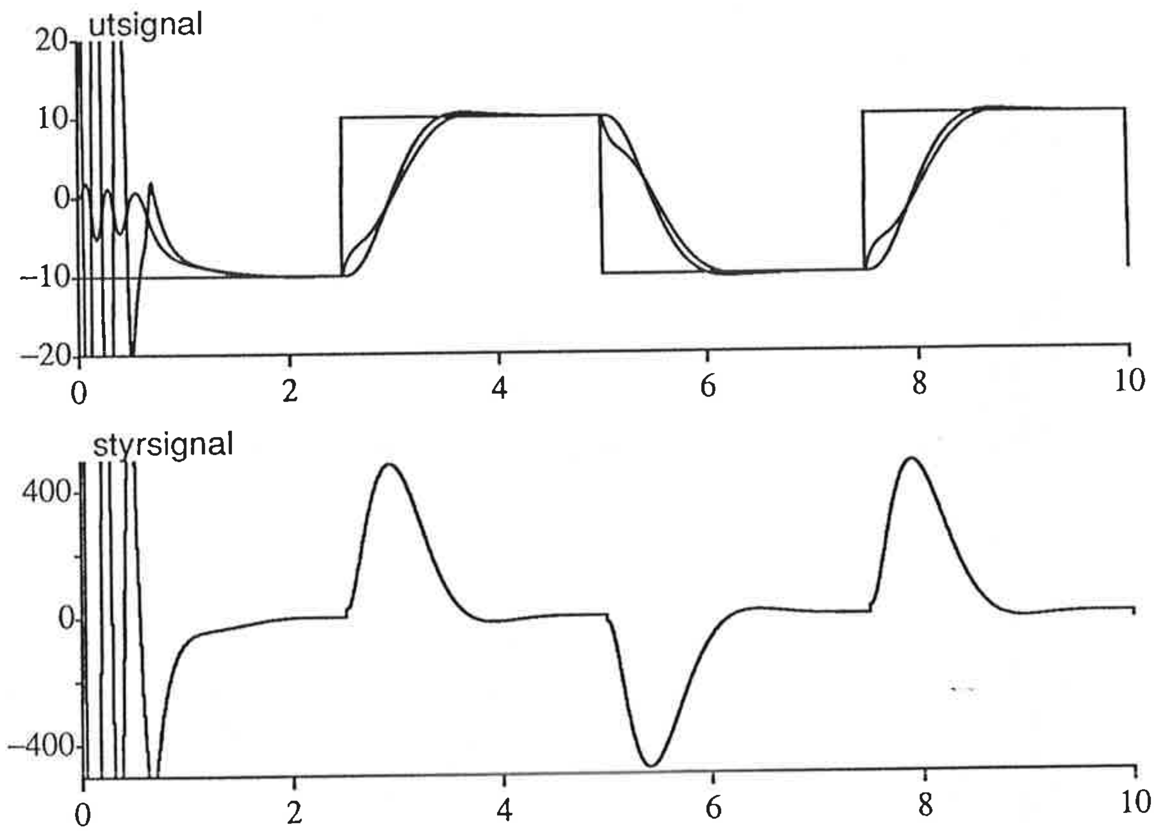
# 4 Simulering

Simuleringarna i detta kapitlet är gjorda med programpaket SIMNON [Elmqvist (1987)]. Den adaptiva regulatorn coh processen som reglerats är de som har blivit beskrivna i de föregående kapitlen. Processen, som är hämtad som ett typfall från ABB Drives på ett typiskt vekt system, skall föreställa en likströmsmotor som ska reglera en pappersmaskin kopplad med en vek axel. Mer om processen finns i [Carlsson, Isberg (1988)]. I samtliga simuleringar har regulator- och observerarpolerna valts till

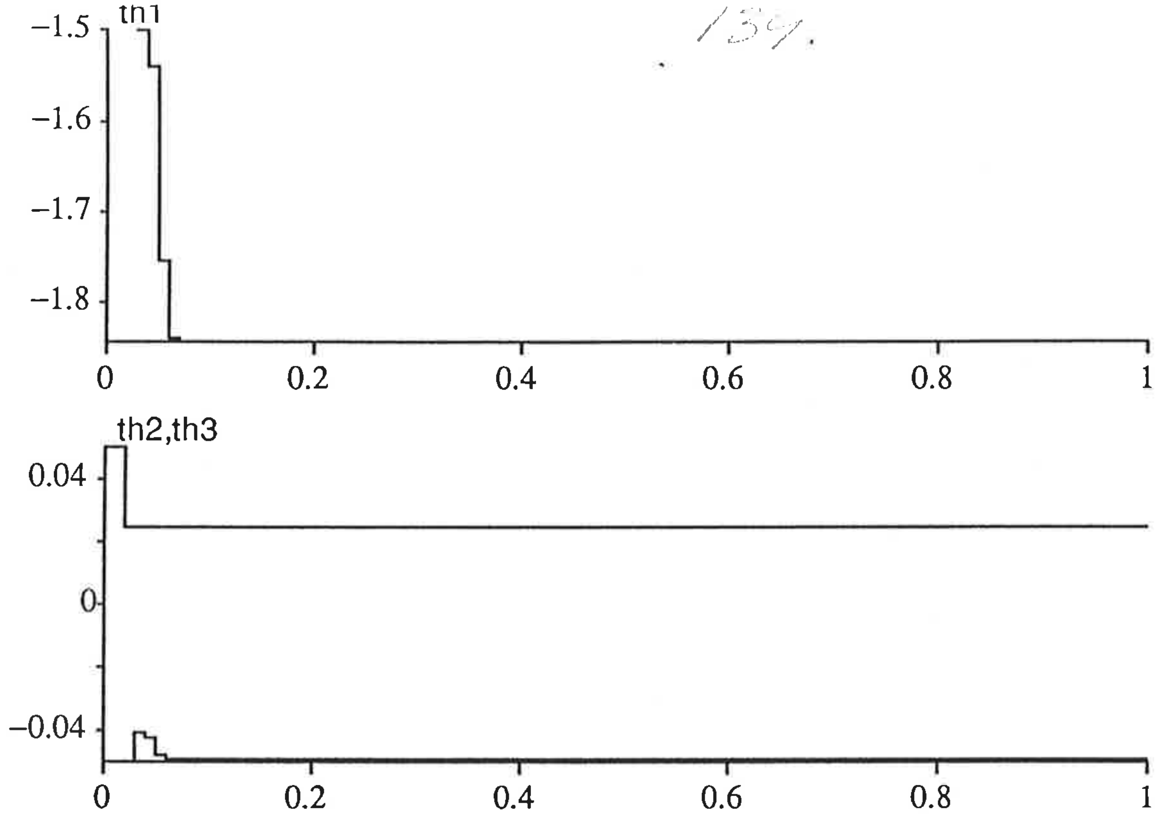
$$\omega = 5 \text{ rad/s} \qquad \zeta = 0.707 \qquad \alpha = 5 \text{ rad/s}$$

$$\omega_o = 35 \text{ rad/s} \qquad \zeta_o = 0.6 \qquad \alpha_o = 35 \text{ rad/s}$$

Först antags processen att vara ideal, dvs att ingen naturlig dämpning finns i processen. I figur 4.1 och 4.2 går det att se att parameterskattaren svänger in sig snabbt och utsignalen följer referensvärdet mycket bra.

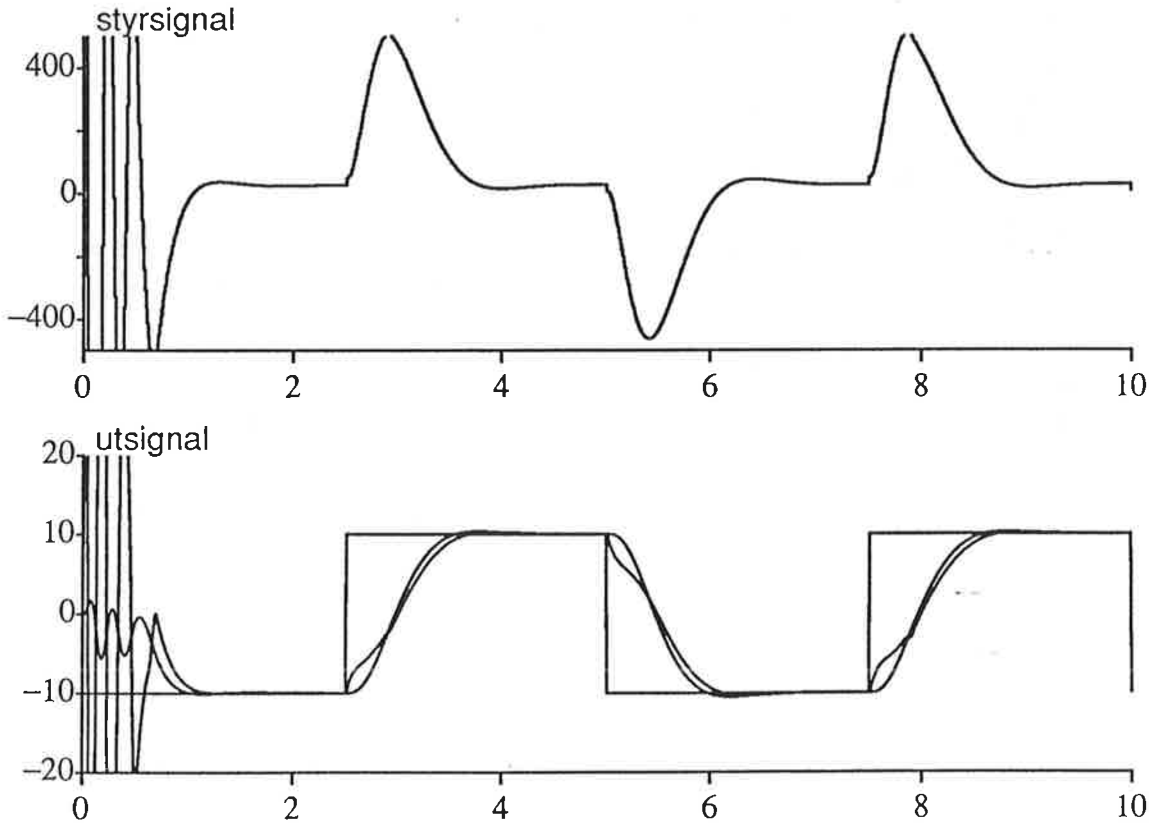



**Figur 4.1** Simulering där parameterskattaren skattar tre parametrar och dämpningen i processen är noll.

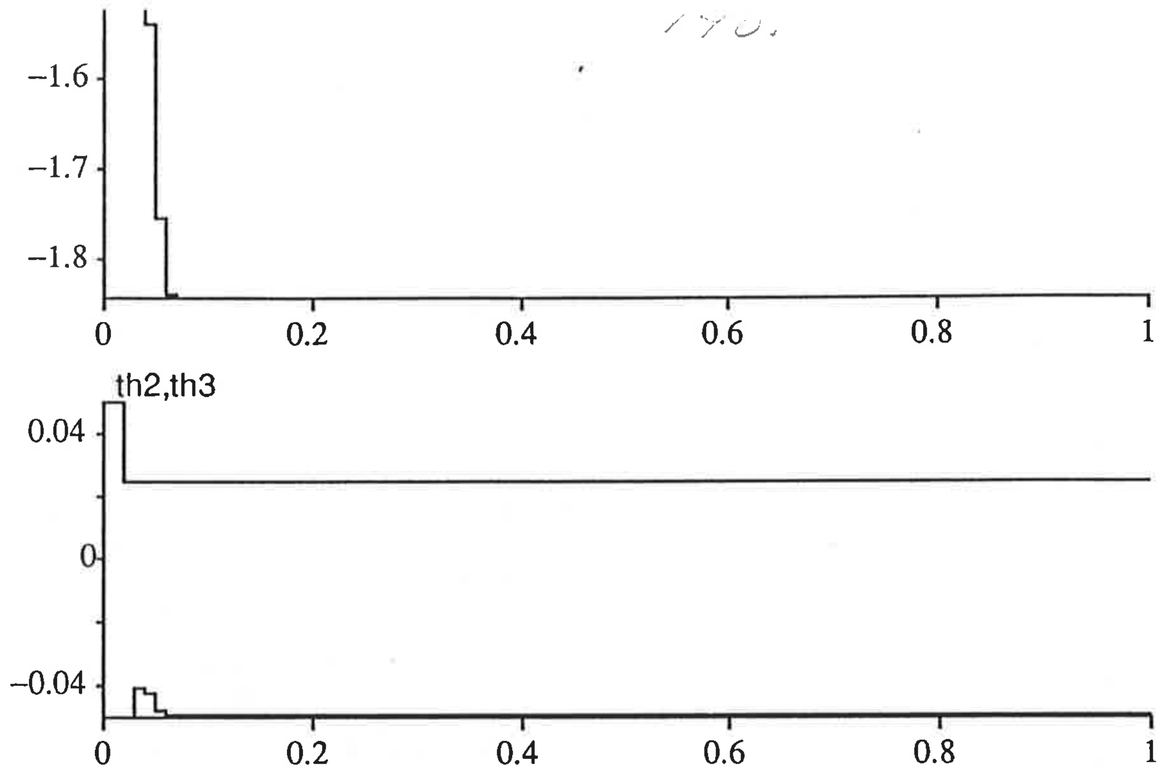



**Figur 4.2** Parameterskattaren svänger in sig snabbt och utsignalen följer referensvärdet mycket bra.

Refererande till figur 4.1 och 4.2 så verkar regulatorn fungera bra för det ideala fallet. Figur 4.3 och 4.4 visar en simulering där ett motmoment har lagts på lasten för att ge en laststörning. Regulatorn klarar fortfarande av att reglera processen bra och parameterskattaren skattar parametrarna rätt. Anledningen till att parameterskattaren fortfarande skattar rätt är att insignalerna till parameterskattaren bandpassfiltreras varvid ingen bias kommer att synas när parametrarna skattas.




**Figur 4.3** Simulering där ett motmoment har lagts på lasten. Regleringen blir utmärkt.




**Figur 4.4** Parameterskattaren skattar fortfarande rätt.

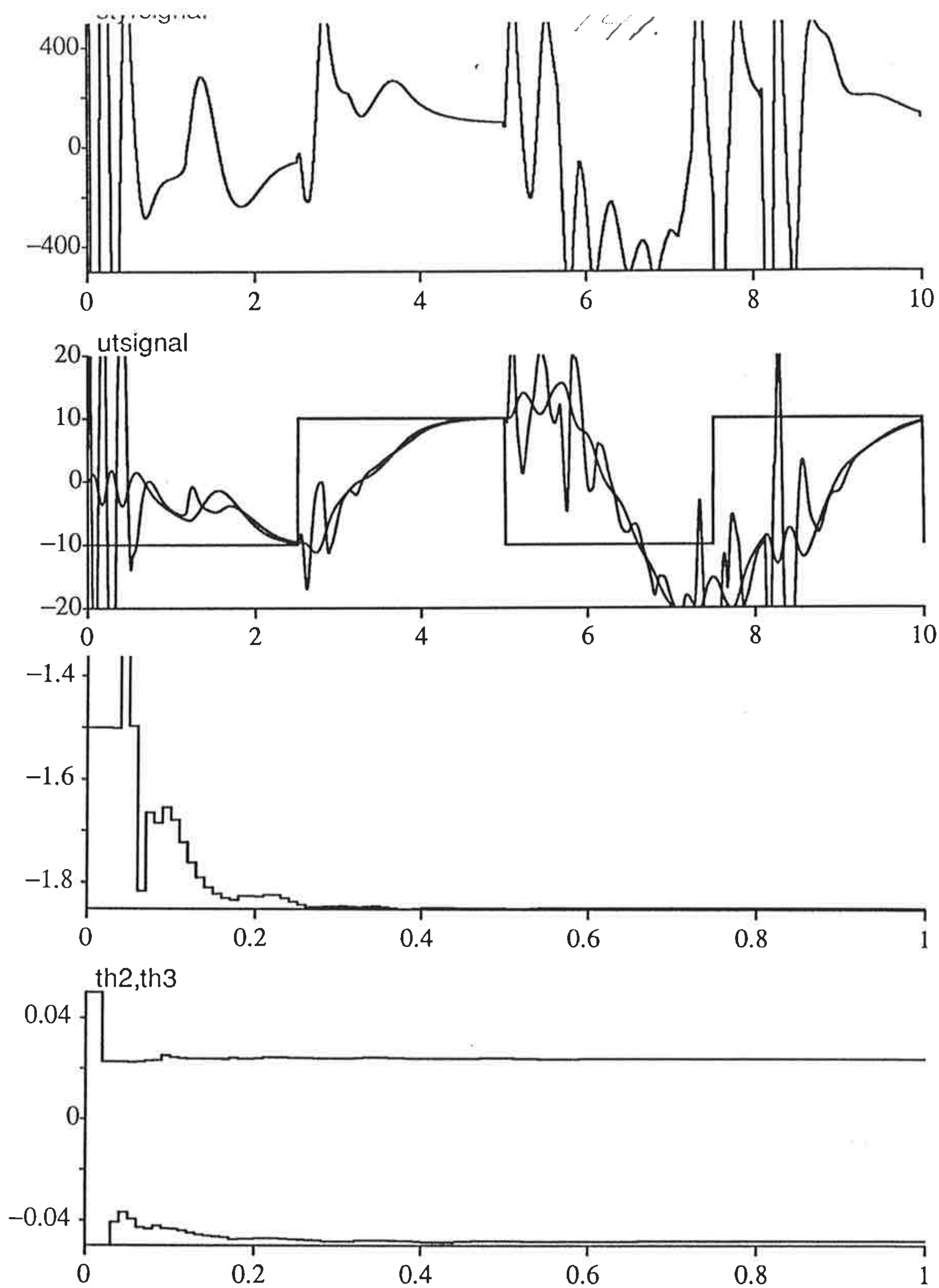
Som sista fall testades hur regulatorn uppträder när dämpning har lagts in i processen. I figur 4.5 visas hur systemet uppträder då dämptermerna är satta till

$$d_m = 10 \text{ Nm·s/rad}$$

$$d_l = 10 \text{ Nm·s/rad}$$

$$d = 10 \text{ Nm·s/rad}$$

Regulatorn klarar längre inte av att reglera systemet väl, men det bör påpekas att dämpningen är relativt stor. För att klara av att reglera processen nu måste fler parametrar skattas.




**Figur 4.8** Simulering där dämpning har lagts in i processen. Regleringen blir inte längre speciellt bra.

# 5 Slutsatser

Om dämpningen är väldigt liten i systemet finns det förutsättningar för att en indirekt adapativ regulator som skattar endast tre parametrar kommer att fungera. Dessutom kommer regulatorn att vara helt okänslig för laststörningar som kommer att filtreras bort i bandpassfiltret i parameterskattaren. Innehåller processen en icke försumbar dämpning, räcker det inte att skatta tre parametrar utan upp till sex parametrar måste skattas.

# 6 Referenser

[1] Carlsson, L., Isberg, A., *Varvtalsreglering av likströmsmotor med vek axel*, Inst för Industriell Elektroteknik och Automation, Lunds tekniska högskola, LUTEDX/(TEIE-5039)/1-69/(1988).

[2] Carlsson, L., Isberg, A., *Identifiering av flexibelt servo*, Inst för reglerteknik, Lunds tekniska högskola.

[3] Elmqvist, H., Åström, K.J. och Schöntal, T.S. (1987) *SIMNON - User's Guide for MS-DOS Computers*, Inst för reglerteknik, Lunds tekniska högskola.

[4] Wallenborg, A. (1987) *Control of Flexible Servo Systems*, Inst för reglerteknik, Lunds tekniska högskola, LUFTD2/(TFRT-3188)/1-104.

[5] Åström, K.J. och Wittenmark, B. (1989) Adaptive control, Addison-Wesley.

# Appendix

Ekvationssystem med lösning, för att lösa ut parametrarna i R(q) och S(q).

$$r_1 + b_1 s_0 = am_1 + am_3 + ao_1 - a_1 + 2$$

$$(a_1-2)r_1 + r_2 + b_2 s_0 + b_1 s_1 = am_1 am_3 + am_2 + ao_1(am_1+am_3) + ao_2 + 2a_1 - 2$$

$$(2-2a_1)r_1 + (a_1-2)r_2 + b_1 s_0 + b_2 s_1 + b_1 s_2 = am_2 am_3 + ao_1(am_1 am_3+am_2) + ao_2(am_1+am_3) + ao_3 - a_1 + 2$$

$$(a_1-2)r_1 + (2-2a_1)r_2 + b_1 s_1 + b_2 s_2 + b_1 s_3 = ao_1 am_2 am_3 + ao_2(am_1 am_3+am_2) + ao_3(am_1+am_3) - 1$$

$$r_1 + (a_1-2)r_2 + b_1 s_2 + b_2 s_3 = ao_2 am_2 am_3 + ao_3(am_1 am_3+am_2)$$

$$r_2 + b_1 s_3 = ao_3 am_2 am_3$$

Nu kallar vi högerleden i ovanstående ekvationer för h1..h6, och inför följande hjälptillstånd:

$$x1 = -1/b_1$$
$$x2 = x1(a_1 - 2 + b_2 x1)$$
$$x3 = x1(2 - 2a_1 + b_2 x2 - 1)$$
$$x4 = x1(a_1 - 2 + b_2 x3 + b_1 x2)$$
$$x5 = x1 b_2 h6 + h5$$
$$x6 = x1(x1 x5 b_2 + h6 - h4)$$
$$x7 = -x1(h3 - b_2 x6 - x5)$$
$$x8 = -(a_1 - 2 + b_2 x3 + b_1 x2)/(1 + b_2 x4 + b_1 x3)$$
$$x9 = (h2 - b_2 x7 - b_1 x6)/(1 + b_2 x4 + b_1 x3)$$

Lösningen till ekvationssystemet blir nu

$$r_1 = (h1 - b_1 x7 - b_1 x4 x9)/(1 + b_1 x3 + b_1 x4 x8)$$
$$r_2 = x8 r_1 + x9$$
$$s_0 = x3 r_1 + x4 r_2 + x7$$
$$s_1 = x2 r_1 + x3 r_2 + x6$$
$$s_2 = x1 r_1 + x2 r_2 - x1 x5$$
$$s_3 = x1(r_2 - h6)$$

I Simnon-implementeringen ser lösningen vid en första anblick lite annorlunda ut, då vi har försökt att kringgå vissa numeriskt känsliga avsnitt, men detta är grundlösningen vilken är lite mer överskådlig.

```
CONTINUOUS SYSTEM Process
"En enkel modell av tv} massor sammankopplade med en vek axel med glapp i
"v{xell}dan. Motorn {r approximerad som en konstant.

input u                 "u=regulatorns styrsignal
output y                "y=motorns varvtal/[rad/s]
state x1 x2 x3          "x1=motorvarvtalet/[rad/s],x2=lastvarvtalet[rad/s]
der dx1 dx2 dx3         "x3=torsionsvinkeln/[rad]
time t

glapp = if x3>fi/2 then fi/2 else if x3<-fi/2 then -fi/2 else x3 "Glappet
Mm    = 78/28*u                                                  "Motormomentet
Ml    = if t< ts then Mb else Ms                                 "Laststeg

dx1 = -k*(x3-glapp)/Jm - (d + dm)/Jm*x1 + d/Jm*x2 + Mm/Jm
dx2 =  k*(x3-glapp)/Jl + d/Jl*x1 - (d+dl)/Jl*x2 - Ml/Jl
dx3 =  x1 - x2

y   =  x1

nmotor = x1*30/3.14159          "Varvtalet i varv per minut
nlast  = x2*30/3.14159          "Varvtalet i varv per minut
tv     = x3*180/3.14159         "Vridningen i axeln med glapp i grader

k:1700                          "Fj{der konstanten f|r axeln
Jm:1.1                          "Motorns tr|ghetsmoment
Jl:38.5                         "Lastens tr|ghetsmoment
dm:10                           "d{mpningen vid motorn
dl:10                           "d{mpningen vid lasten
d:10                            "d{mpningen i axeln
fi : 0".06981                   "Totala glappet/[rad]
Mb : 78                         "Motverkande moment
Ms : 78                         "Laststeg
x1 :0" 41.8879                  "Begynnelsev{rde
x2 :0" 41.8879
x3 :0" 0.08079
ts :0" 1
end
```

```
Macro start
"Starts a simultion of problem projekt

let n.noise1=1
let nodd.noise1=12345
syst parsk process reg tl noise1 conn
par dt[noise1]:0.1
par stdev1[noise1]:10
"store x4 x5 x6 x7 x10 x11
store x2[process] u[process] y[process] uc[reg] th1 th2 th3
"store r1[reg] r3[reg] s0[reg] s1[reg] s2[reg] s3[reg] -add
simu 0 10/a
"split 3 2
"ashow x5/a
"ashow x6 x7/a
"ashow x10/a
"ashow x11/a
"ashow x4/a
"ashow r1 r3
"ashow s0 s1 s2 s3
"suspend
split 2 1
ashow u
text 'styrsignal'
axes h 0 10 v -20 20
show y x2 uc
text 'utsignal'

end
```

```
connecting system conn

time t

"b=e1[noise1]
b=if mod(t,tp)>tp/2 then 10 else -10
"b=10*sin(1.2*t)
uc[reg]=b
inu[parsk]=u[reg]
u[process]=u[reg]
iny[parsk]=y[process]
y[reg]=y[process]
a[tl]=th1[parsk]
b1[tl]=th2[parsk]
b2[tl]=th3[parsk]
s0[reg]=s0[tl]
s1[reg]=s1[tl]
s2[reg]=s2[tl]
s3[reg]=s3[tl]
t0[reg]=t0[tl]
t1[reg]=t1[tl]
t2[reg]=t2[tl]
t3[reg]=t3[tl]
r0[reg]=r0[tl]
r1[reg]=r1[tl]
r2[reg]=r2[tl]
r3[reg]=r3[tl]
ao0[reg]=ao0[tl]
ao1[reg]=ao1[tl]
ao2[reg]=ao2[tl]
ao3[reg]=ao3[tl]

tp:5
end
```

```
DISCRETE SYSTEM reg

INPUT y uc s0 s1 s2 s3 r0 r1 r2 r3

INPUT t0 t1 t2 t3 ao0 ao1 ao2 ao3

OUTPUT u

STATE xt1 xt2 xt3 xs1 xs2 xs3 xr1 xr2 xr3 v1 v2 v3

NEW nxt1 nxt2 nxt3 nxs1 nxs2 nxs3 nxr1 nxr2 nxr3 nv1 nv2 nv3

TIME t

TSAMP ts

nxt1=uc
nxt2=xt1
nxt3=xt2
u1 = t0*uc + t1*xt1 + t2*xt2 + t3*xt3

nxs1=y
nxs2=xs1
nxs3=xs2
u2 = - s0*y - s1*xs1 - s2*xs2 - s3*xs3

nxr1 = u
nxr2 = xr1
nxr3 = xr2
u3 = (ao1 - r1)*xr1 + (ao2 - r2)*xr2 + (ao3 - r3)*xr3

p = u1 + u2 + u3

nv1 = -ao1*v1 - ao2*v2 - ao3*v3 + p
nv2 = v1
nv3 = v2
v = -ao1*v1 - ao2*v2 - ao3*v3 + p

u = if v>umax then umax else if v<umin then umin else v

ts=t+h

umax:500
umin:-500
h:0.01

END
```

```
Discrete system parsk
"Regulator based on indirect self tuning adaption

input iny inu
state y1 y2 u1 u2 u3 inu1 inu2 iny1 iny2
state f1 f2 f3 th1 th2 th3 uf1 uf2 yf1 yf2
state p11 p12 p13 p22 p23 p33
new ny1 ny2 nu1 nu2 nu3 ninu1 ninu2 niny1 niny2
new nf1 nf2 nf3 nt1 nt2 nt3 nuf1 nuf2 nyf1 nyf2
new n11 n12 n13 n22 n23 n33
time t
tsamp ts

"Bandpass filtrering
u =  -af1*uf1 - af2*uf2 + bf0*inu + bf1*inu1 + bf2*inu2
nuf1 = u
nuf2 = uf1
ninu1 = inu
ninu2 = inu1

yf = -af1*yf1 - af2*yf2 + bf0*iny + bf1*iny1 + bf2*iny2
nyf1 = yf
nyf2 = yf1
niny1 = iny
niny2 = iny1

y = yf - yf1

"The residual
e=y+y2-th1*f1-th2*f2-th3*f3
ymp=y+y2
ymskatt=th1*f1+th2*f2+th3*f3
"Estimation gain and den=fi*P*fi
k1=p11*f1+p12*f2+p13*f3
k2=p12*f1+p22*f2+p23*f3
k3=p13*f1+p23*f2+p33*f3

den=lam+f1*k1+f2*k2+f3*k3

"Updating the estimates
nt1=th1+k1*e/den
nt2=th2+k2*e/den
nt3=th3+k3*e/den

"Compute covariances
n11=(p11-k1*k1/den)/lam
n12=(p12-k1*k2/den)
n13=(p13-k1*k3/den)
n22=(p22-k2*k2/den)/lam
n23=(p23-k2*k3/den)
n33=(p33-k3*k3/den)/lam

"Update y1=y(t-1)....
ny1=y
ny2=y1
nu1=u
nu2=u1
nu3=u2

"Update regression vector fi
nf1=-y
nf2= u+u2
nf3= u1

ts=t+h
```

```
pi:3.1415926
h:0.01
lam:0.98
af1:-0.854081"-1.718945
af2:4.7411e-17"0.753671
bf0:0.517789
bf1:0
bf2:-0.517789


th1:-1.5
th2:0.05
th3:-0.05
p11:100
p22:100
p33:100
END
```

```
DISCRETE SYSTEM TL

INPUT a b1 b2

OUTPUT s0 s1 s2 s3 r0 r1 r2 r3 t0 t1 t2 t3

OUTPUT ao0 ao1 ao2 ao3

TIME t

TSAMP ts

ao1p= -2*exp(-zo*wo*h)*cos(wo*sqrt(1-zo*zo)*h)
ao2p= exp(-2*zo*wo*h)
ao3p= -exp(-beta*h)
ao1 = ao1p + ao3p
ao2 = ao1p*ao3p + ao2p
ao3 = ao2p*ao3p

a1= -2*exp(-z*w*h)*cos(w*sqrt(1-z*z)*h)
a2= exp(-2*z*w*h)
a3= -exp(-alpha*h)

am1= a1+ a3+ ao1
am2= a1*a3+ a2+ ao1*(a1+ a3)+ ao2
am3= a2*a3+ ao1*(a1*a3+ a2)+ ao2*(a1+ a3)+ ao3
am4= ao1*a2*a3+ ao2*(a1*a3+ a2)+ ao3*(a1+ a3)
am5= ao2*a2*a3+ ao3*(a1*a3+ a2)
am6= ao3*a2*a3

g= 2-2*a
k= a-2
x1= -1/b1
x2= x1*(k-b2/b1)
x3= x1*(g-1+b2*x2)
x4= x1*(k+b2*x3+ b1*x2)
x5= x1*(k-am3+b2*x11+x10)
x6= -(k+ b2*x3+b1*x2)/(1+b2*x4+b1*x3)
x7= (am2-g-b2*x5-b1*x11)/(1+b2*x4+b1*x3)
x10= b2*x1*am6+am5
x11= x1*(1- b2*x1*x10- b1*x1*am6- am4)
r1p= (am1-k-b1*x5-b1*x4*x7)/(1+b1*x3+b1*x4*x6)
r2p= x6*r1p+x7
s0= x3*r1p+ x4*r2p+ x5
s1= x2*r1p+ x3*r2p+ x11
s2= x1*r1p+ x2*r2p- x1*x10
s3= x1*(r2p- am6)
r0 = 1
r1 = r1p - 1
r2 = r2p - r1p
r3 = -r2p
t0 = (s0 + s1 + s2 + s3)/(1+ ao1+ ao2+ ao3)
t1 = ao1*t0
t2 = ao2*t0
t3 = ao3*t0
ao0= 1

ts=t+h

z:0.707
w:5
alpha:5
zo:0.6
wo:35
beta:35
h:0.01
```

102.

```
"a: -1.84313355274486
"b1: 0.0246775380345534
"b2: -0.0492447262956053
END
```

153

ADAPTIV REGLERING

Projekt:

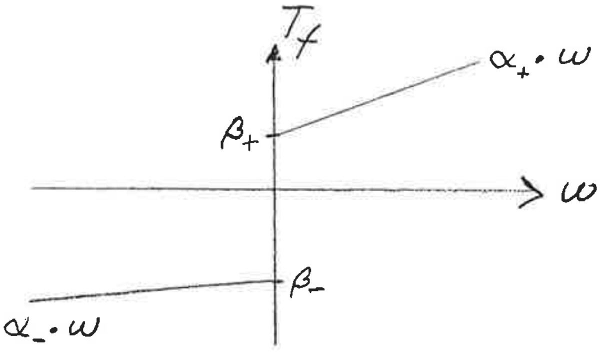
Frictionskompensering av DC-servo

Åsa Göransson
Håkan Hedlund

## INLEDNING

Varför adapterande friktionskompensering?

Ett lagers friktion kan mätas och kan beskrivas enligt modell.




Ofta har lagret olika friktion i olika riktning. Varför identifierar man då inte friktionen och kompenserar med fixa värden i styrlagen.

Friktionen kan beroende på lagertyp ändra sig olika mycket med tiden. Ett lager har ofta större friktion när det är nytt än när det blivit väl inkört, för att sedan få större friktion igen när det blir utslitet. Val av smörjmedel påverkar även friktionen. Mängden smörjmedel är även friktionspåverkande. Slutligen påverkar även belastning och temperatur.

## SLUTSATS

Det finns anledning att adaptera friktionskompenseringen för motor som arbetar under olika driftsbetingelser, när noggrann reglering krävs.

## DC-SERVO

Modell av hastighetsloopen

$$\frac{dw}{dt} = -a_{vc}*w+b_{vc1}*u-alfa_i*w-beta_i$$ (se bil 8)

För att få över modellen på diskret form gör vi en derivataapproximation

$$\frac{w(t)-w(t-1)}{h} = -a_{vc}*w(t-1)+b_{vc1}*u(t-1)-alfa_i*w(t-1)-beta_i$$

$$w(t) = (1-a_{vc}*h)w(t-1)+b_{vc1}*h*u(t-1)-alfa_i*h*w(t-1)-h*beta_i$$

Vi skattar olika alfa och beta beroende på rotationsriktning.

Med MK-metoden blir beteckningarna

| | |
|---|---|
| th1= alfa+ | f1= -w(t-1)*h |
| th2= beta+ | f2= -h |
| th3= alfa- | f3= -w(t-1)*h |
| th4= beta- | f4= -h |

Felet blir då

$$e_+ = w(t) - w(t) =$$

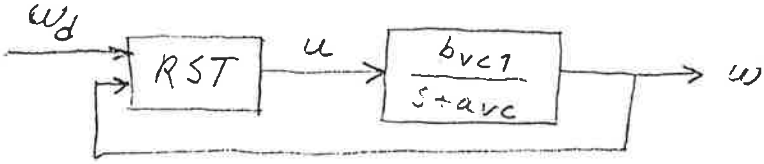
$$= w(t) - \left[(1-h*a_{vc})w(t-1)+h*b_{vc1}u(t-1)+th1*f1+th2*f2\right]$$

$e_-$ analogt.

Som grund för projektet har använts artikeln "Adaptive Friction Compensation in DC-Motor Drives" av Kanudas, Åström och Braun. IEEE nr 687. Dock har konceptet modifierats åtskilligt.

De skattade parametrarna kontrolleras så att de har rätt värde. Friktionen kompenseras genom att man i styrlagen lägger till $\hat{g} = (\widehat{alfa}_i \cdot w + \widehat{beta}_i)/b_{vc1}$ (Bil 1)

$R*u = T*w_d - S*w + R*\hat{g}$ (Bil 6-7)




Regulatorn är en integrerande RST-regulator i digital form, specifiserad enligt kontinuerliga parametrar.

$b_{vc1} = 2.45$

$a_{vc} = 0,22$

$$H_m = \frac{w_{mv}^2}{s^2 + 2*w_{mv}*z_{mv}\cdot s + w_{mv}^2}$$

$w_{mv} = 2$

$z_{mv} = 0,7$

$h = 0,05$

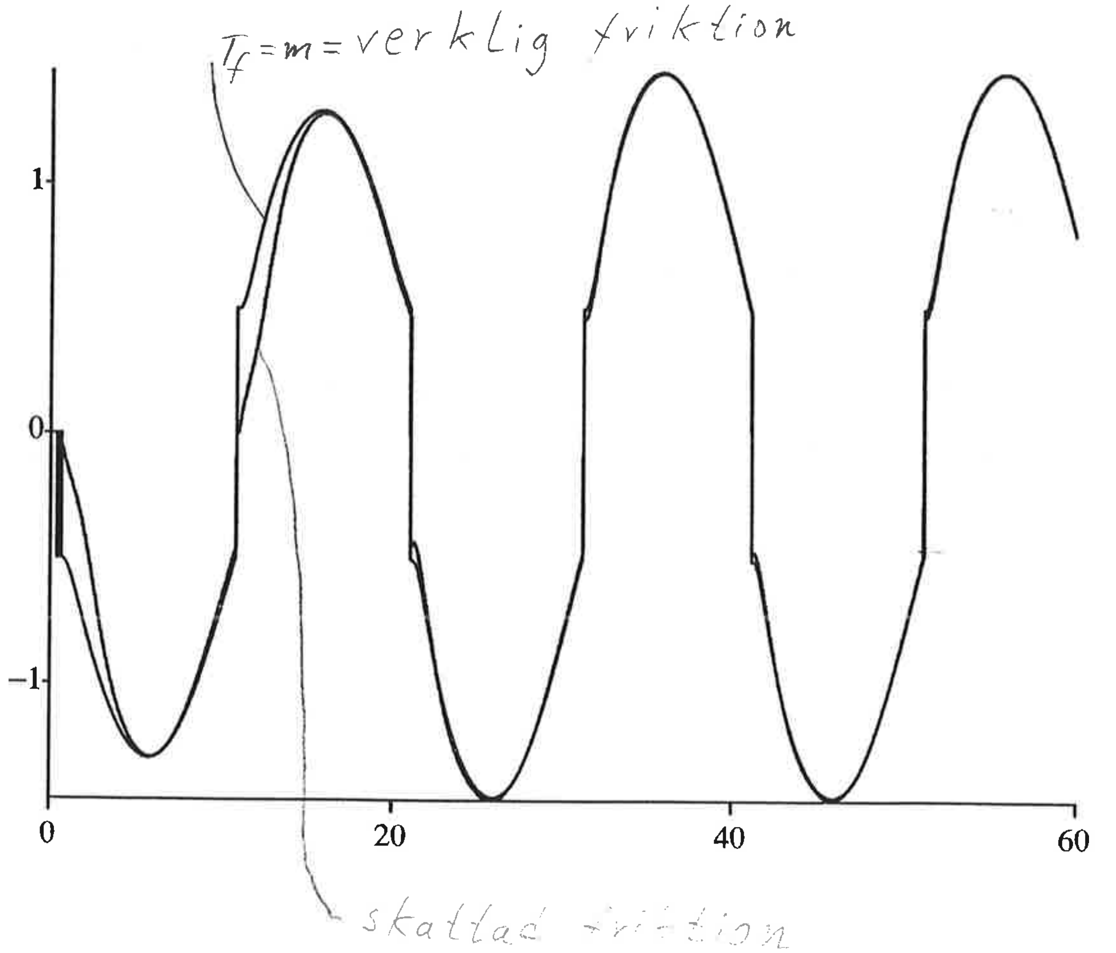
Resultat se bilaga 2-4.
Vi har provat med fyrkant-, sågtand- och sinussignal med gott resultat. Har även provat med olika friktion med likartat resultat, varför vi ej redovisar de diagrammen.

I ett projekt i kursen "Datorimplementering av reglersystem" ingår denna hastighetsregulator i innerloopen till en positionsregulator. Positionsregulatorn är en PD-regulator.

157.

bilaga 1

89.05.31 - 00:17:05 nr: 4
hcopy meta/dia



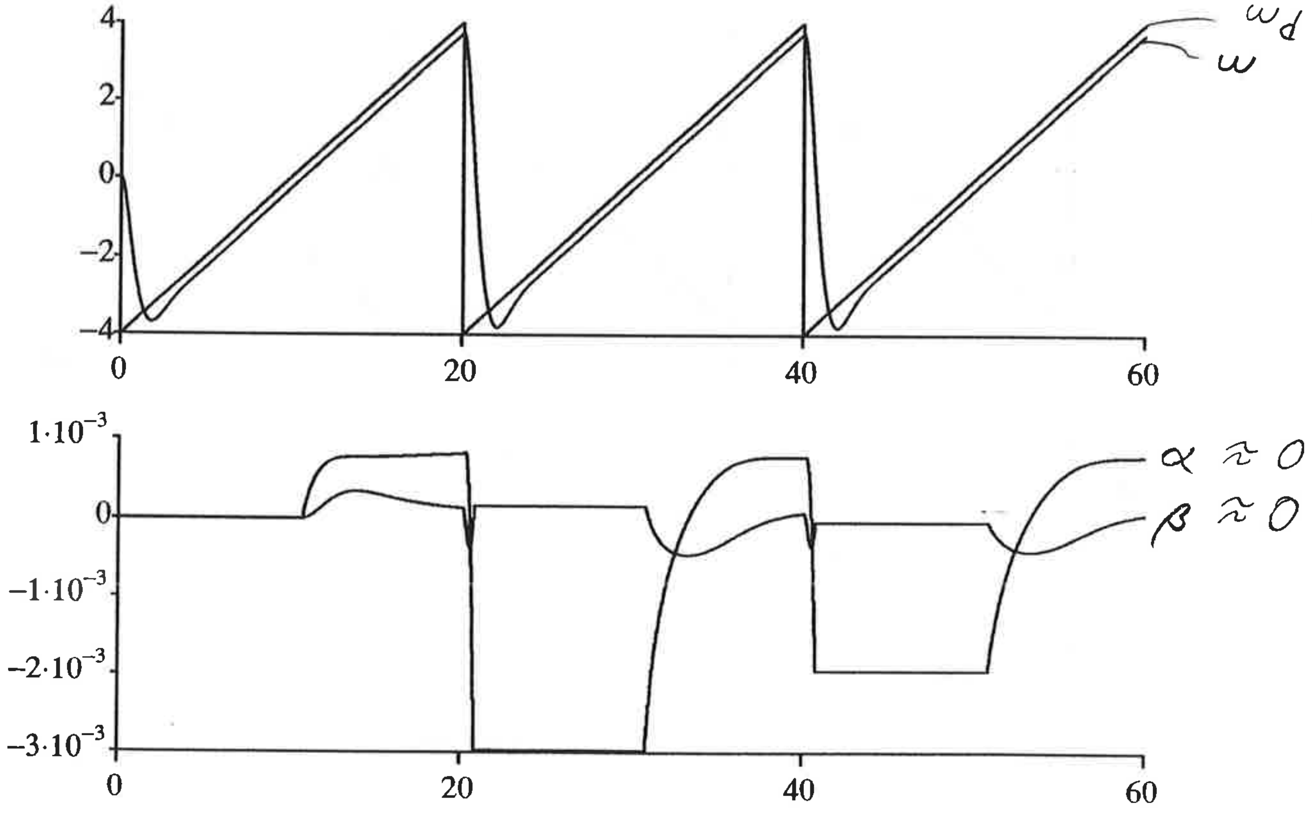

4

bilaga 2a

sågtandvåg

utan friktion

89.05.26 - 00:25:10 nr: 7
hcopy meta/dia

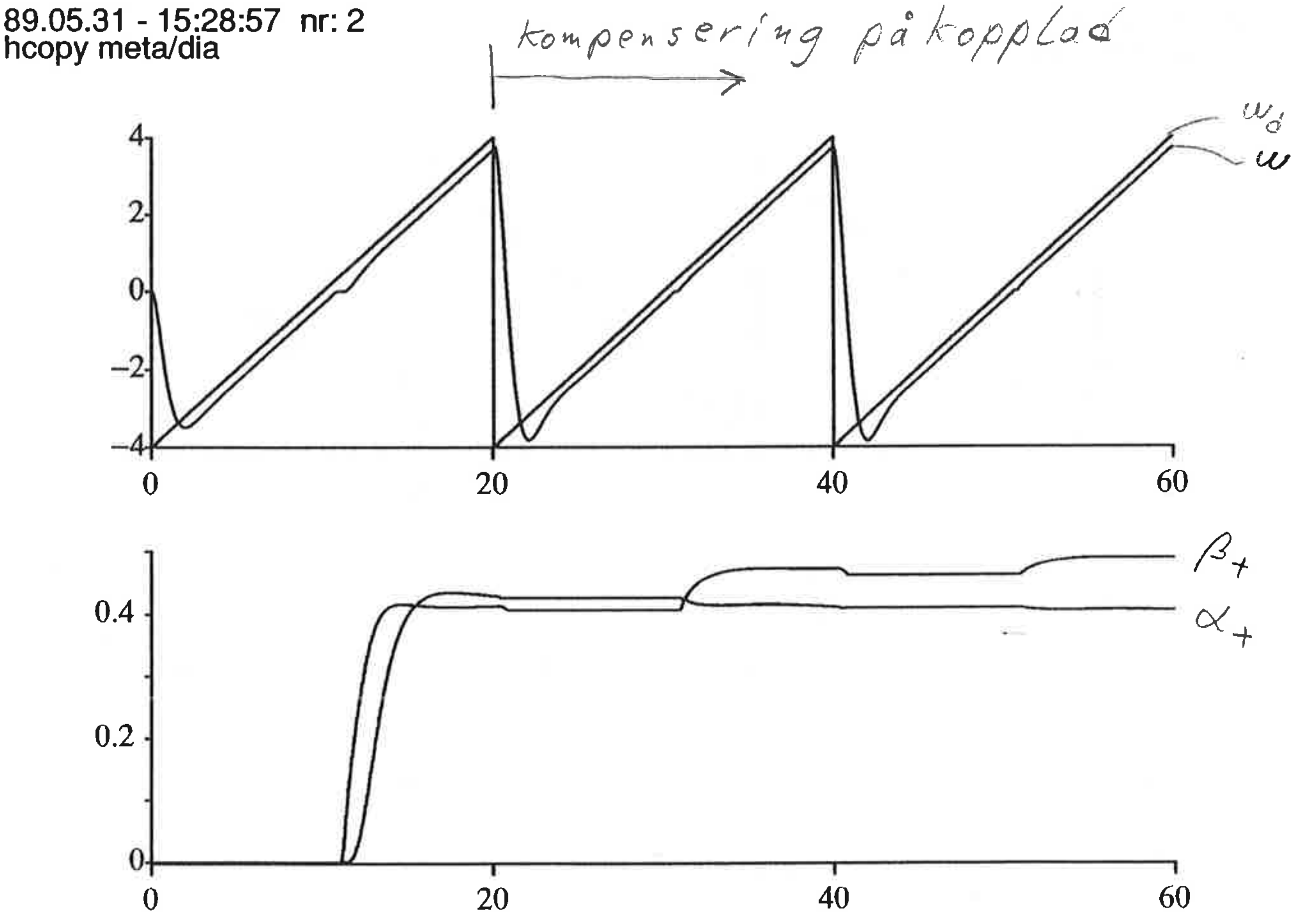


bilaga 2b

sågtandvåg
med friktion

$\alpha_+ = \alpha_- = 0,4$

$\beta_+ = \beta_- = 0,5$



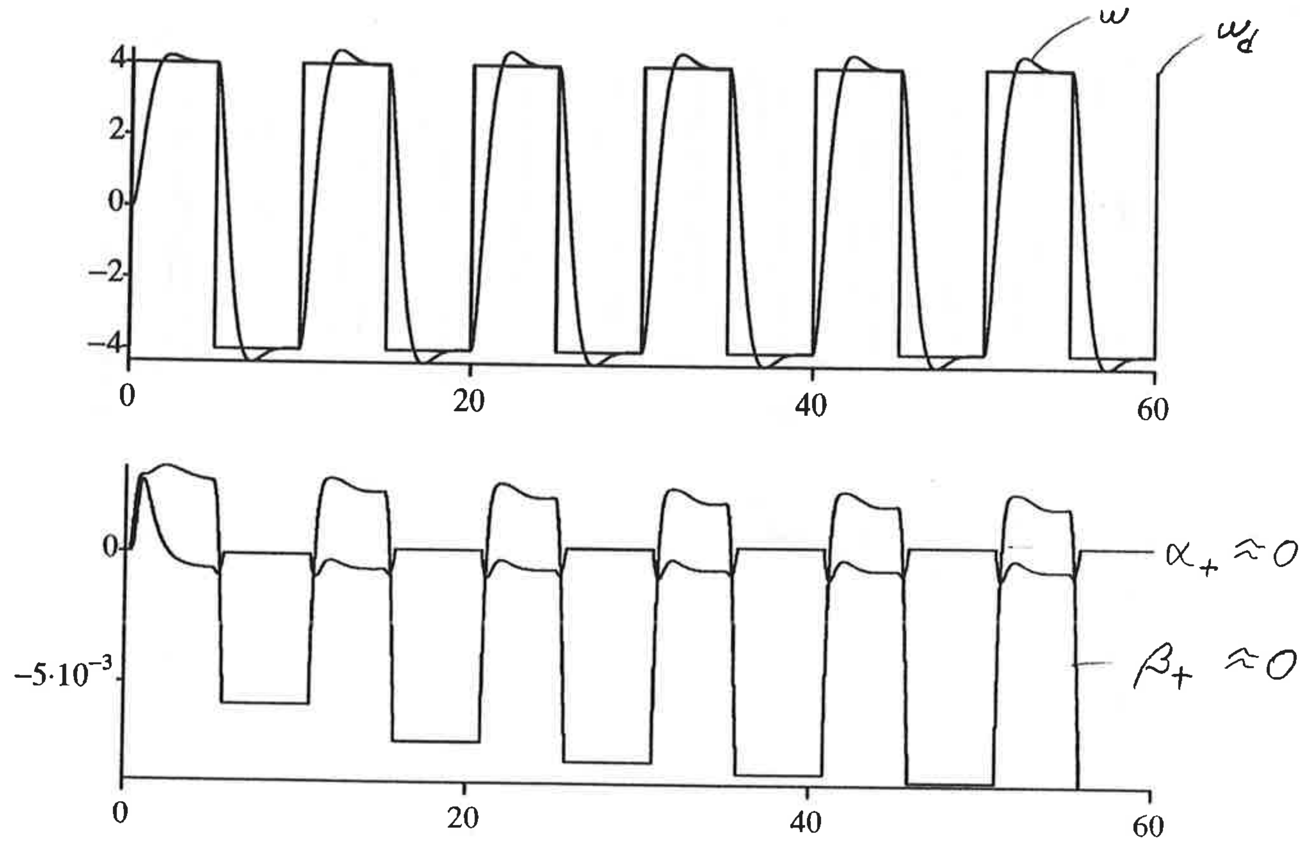
160.

bilaga 3c

fyrkantvåg

utan friktion

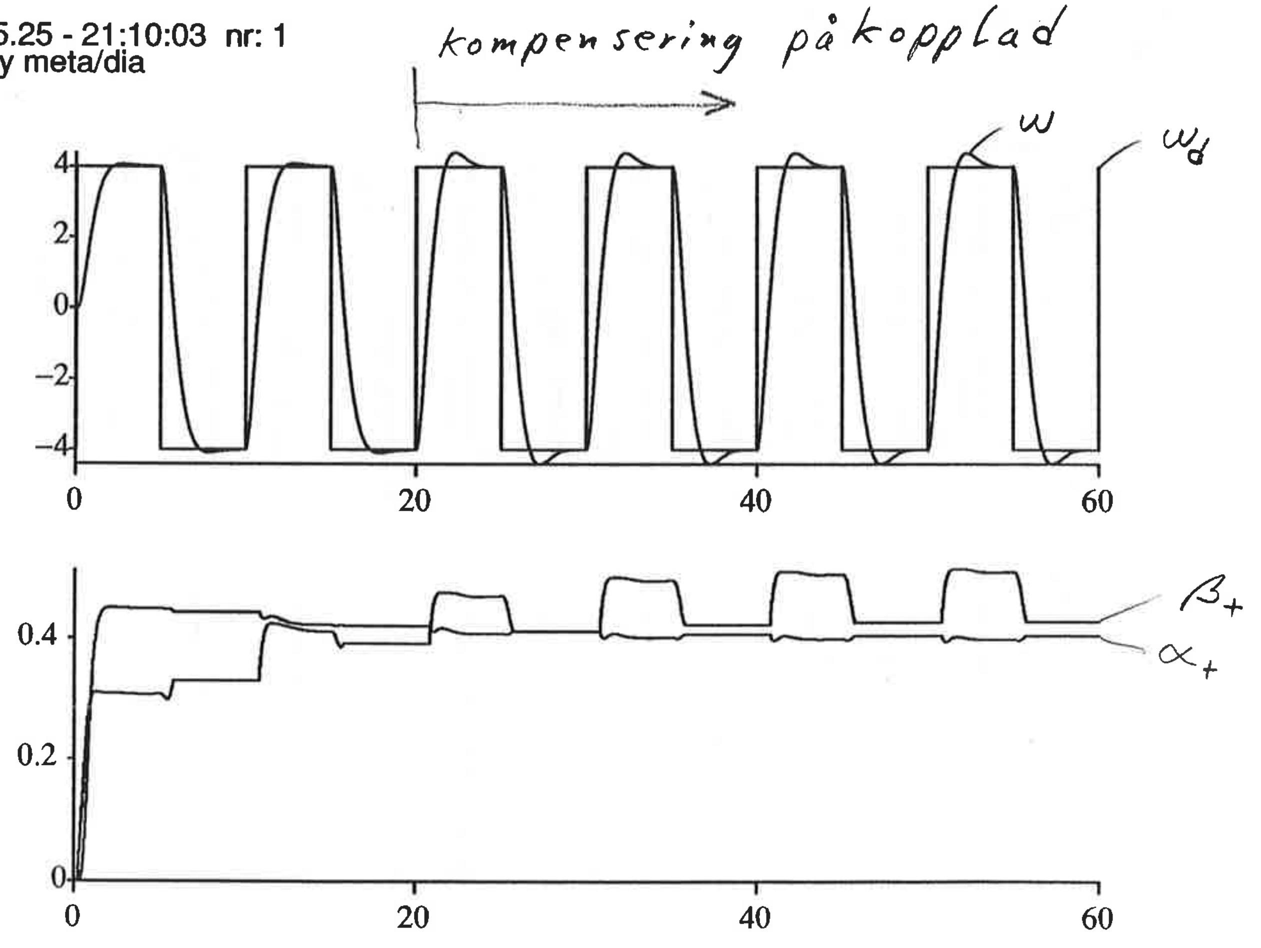
89.05.25 - 20:14:46 nr: 4
hcopy meta/dia.p



biLaga 3k

fyrkantvåg
med friktion
$\alpha_+ = \alpha_- = 0,4$
$\beta_+ = \beta_- = 0,5$

89.05.25 - 21:10:03 nr: 1
hcopy meta/dia

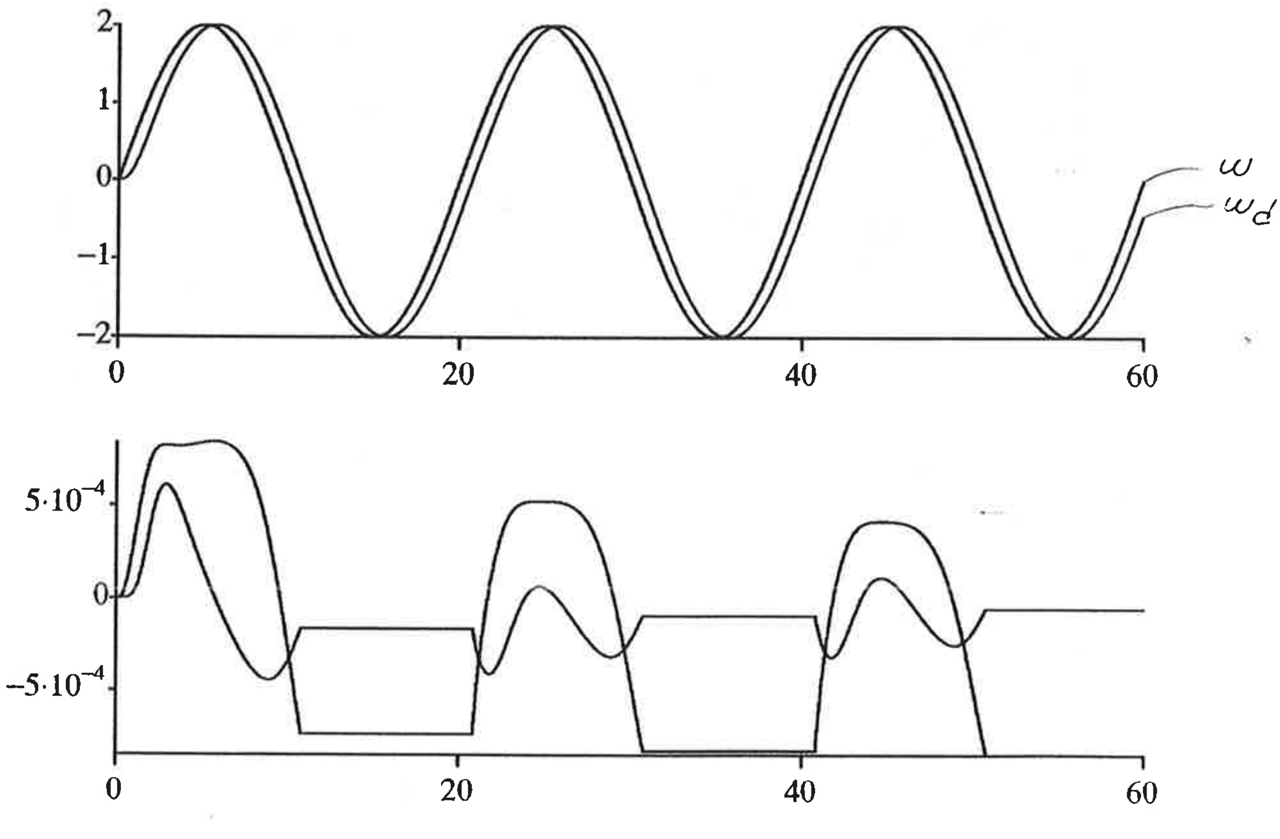


bilaga 4a

sinussignal

utan friktion

89.05.31 - 17:25:46 nr: 1
hcopy meta/dia



bilaga 4b

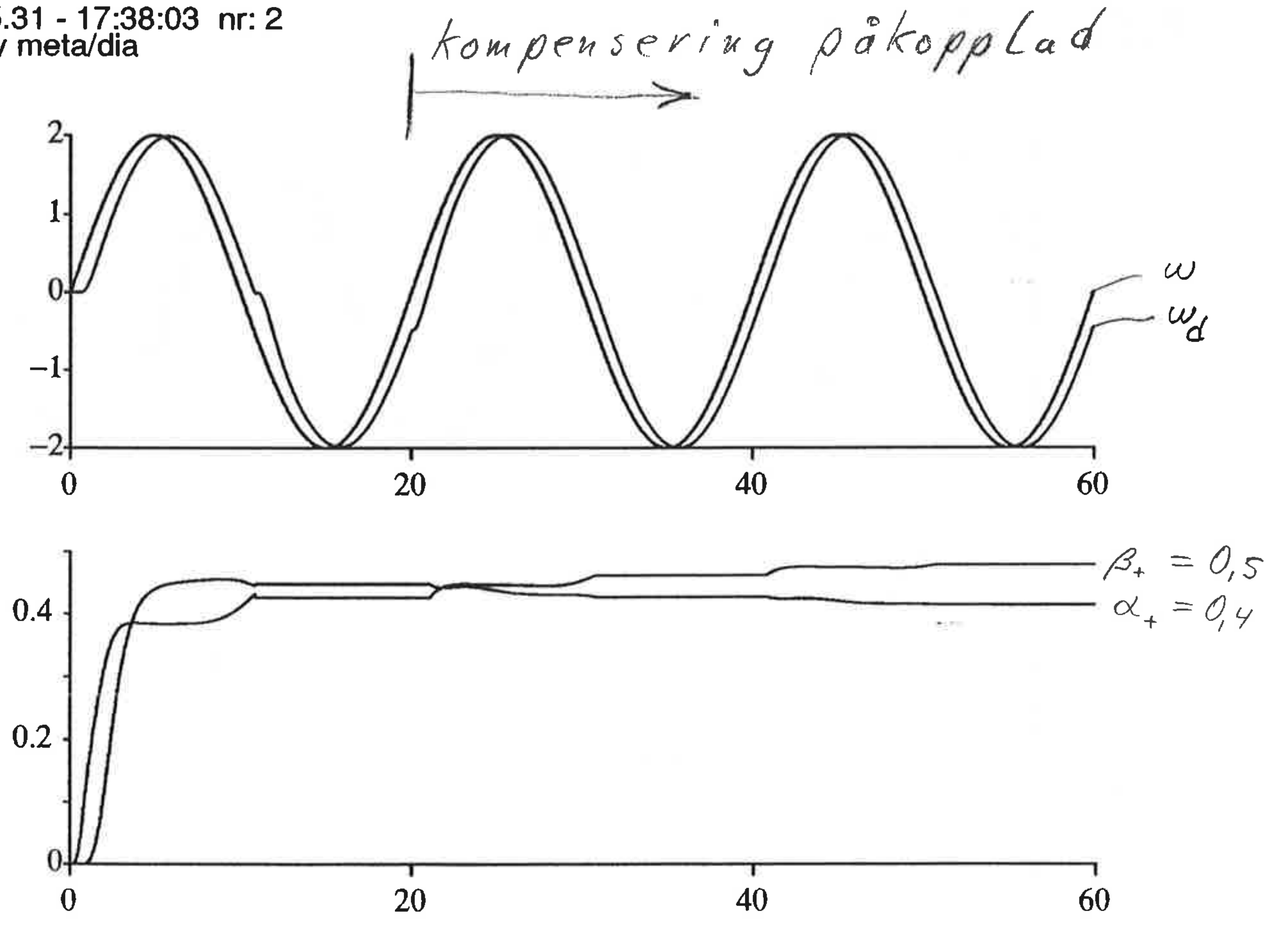
Sinussignal

med friktion

$\alpha_+ = \alpha_- = 0,4$

$\beta_+ = \beta_- = 0,5$

89.05.31 - 17:38:03 nr: 2
hcopy meta/dia



```
DISCRETE SYSTEM reg
INPUT  w
OUTPUT u
STATE u1  w1  wd1 g1
STATE th1 th2 th3 th4
STATE p11 p12 p22 p33 p34 p44
NEW nu1 nw1 nwd1 ng1
NEW nth1 nth2 nth3 nth4
NEW np11 np12 np22 np33 np34 np44
TIME t
TSAMP ts

notzero  = IF w * w1 > 0 THEN 1 ELSE 0
plus   = notzero * (IF w > 0 THEN 1 ELSE 0)
minus = notzero * (IF w < 0 THEN 1 ELSE 0)

" FRICTIONESTIMATION
" Compute the residual

eplus  = w - ((1 - avc * h) * w1 + bvc1 * h * u1 + th1 * f1 + th2 * f2)

eminus = w - ((1 - avc * h) * w1 + bvc1 * h * u1 + th3 * f3 + th4 * f4)

e = plus * eplus + minus * eminus

derapp = (w - w1)/h

" Compute estimator gain k=P*fi and den=lam+fi*P*fi
k1 =  p11 * f1 + p12 * f2
k2 =  p12 * f1 + p22 * f2
k3 =                          p33 * f3 + p34 * f4
k4 =                          p34 * f3 + p44 * f4

ddplus  = lam + f1 * k1 + f2 * k2
ddminus = lam + f3 * k3 + f4 * k4

" Update estimates
nth1 = th1 + plus   * k1 * e / ddplus
nth2 = th2 + plus   * k2 * e / ddplus
nth3 = th3 + minus  * k3 * e / ddminus
nth4 = th4 + minus  * k4 * e / ddminus

" Update covariances
np11= IF plus  THEN (p11 - k1 * k1 / ddplus ) / lam ELSE p11
np12= IF plus  THEN  p12 - k1 * k2 / ddplus         ELSE p12
np22= IF plus  THEN (p22 - k2 * k2 / ddplus ) / lam ELSE p22
np33= IF minus THEN (p33 - k3 * k3 / ddminus) / lam ELSE p33
np34= IF minus THEN  p34 - k3 * k4 / ddminus        ELSE p34
np44= IF minus THEN (p44 - k4 * k4 / ddminus) / lam ELSE p44

alfaplus = th1
betaplus = th2
alfaminu = th3
betaminu = th4

" Update w1=w(t-1)...wd1=wd(t-1)...wwd1=wwd(t-1)...u1=u(t-1)
nw1  = w
nwd1 = wd
nu1  = u
ng1  = g
```

```
" Update regression vector fi
f1 =  - w1 * h
f2 =       - h
f3 =  - w1 * h
f4 =       - h

gpos=IF w > 0 THEN  (  w*alfaplus + betaplus) ELSE 0
gneg=IF w < 0 THEN  (  w*alfaminu + betaminu) ELSE 0

gconneg = -gpos - gneg
g = IF adapt THEN (gpos + gneg)/bvc1  ELSE 0  "Frictionest

adapt = IF t < 20 THEN 0 ELSE 1

" REGULATOR DESIGN
" Computation of discrete polynomials Bv,Av,Bmv,Amv

bv=(1-exp(-avc*h))*bvc1/avc                    "Bv -poly

av=-exp(-avc*h)                                "Av -poly

ola=zmv*wmv*sin(sqv*h)/sqv                     "Bmv-poly
bmv1=1-exp(-zmv*wmv*h)*(cos(sqv*h)+ola)
bmv2=exp(-2*zmv*wmv*h)+exp(-zmv*wmv*h)*(ola-cos(h*sqv))

sqv=wmv*sqrt(1-zmv*zmv)                        "Amv-poly
amv1=-2*exp(-zmv*wmv*h)*cos(h*sqv)
amv2=exp(-2*zmv*wmv*h)

"Calculation of the  regulators' parameters
r0= 1
r1=-1
s0=(amv1-av+1)/bv
s1=(amv2+av)/bv
t0=bmv1/bv
t1=bmv2/bv

" Calculation of control signals

wd = trerefw * trekant + fyrrefw * fyrkant + sinrefw * sinus

u  = t0 * wd + t1 * wd1 - s0 * w - s1 * w1 - r1 * u1 + r0 * g + r1 * g1


trekant = biasw3 + 2*rampw3*(mod(t,perw3)-perw3/2)/perw3
fyrkant = biasw4 + rampw4*(IF sin(3.142*t/perw4)>0 THEN 1 ELSE -1)
sinus   = biasw5 + rampw5*sin(3.142*t/perw5)

"Update sampling time
ts=t+h

"Specification parameters

h:0.05
bvc1:  2.45        "Bv -poly in continuous form
avc :  0.12        "Av -poly in continuous form
zmv :  0.7         "Amv-poly in continuous form
wmv :  2


lam:0.995        "Forgetting factor
```

164.

Bilaga 6

```
"Initial values
p11:20
p22:20
p33:20
p44:20
hyst:0.00
perw3:20
perw4:5
perw5:10
perp:4
refwin:0
fyrrefw:0
trerefw:1
sinrefw:0
rampw3:4
rampw4:4
rampw5:2
biasw3:0.0
biasw4:0.0
biasw5:0.0

END
```

165.

Bilaga 7

```
CONTINUOUS SYSTEM servula
INPUT u
OUTPUT w
STATE x1 x2
DER dx1 dx2
TIME t

mwp = -alfap*x2 - betap   "w>0
mwm = -alfam*x2 - betam   "w<0
mwz : 0

albert = IF x2<-eps THEN mwm ELSE IF x2>eps then mwp ELSE mwz
m =IF fric THEN albert ELSE 0

dx1= bvc2 * x2

dx2= -avc * x2 + bvc1 * u + m

pos=x1

w = x2

avc:  0.12
bvc1: 2.45
bvc2:10.10
eps:  0.0001
fric :  1
alfap:  0.4
betap:  0.5
alfam:  0.4
betam: -0.5
END
```

106

Bilaga 8

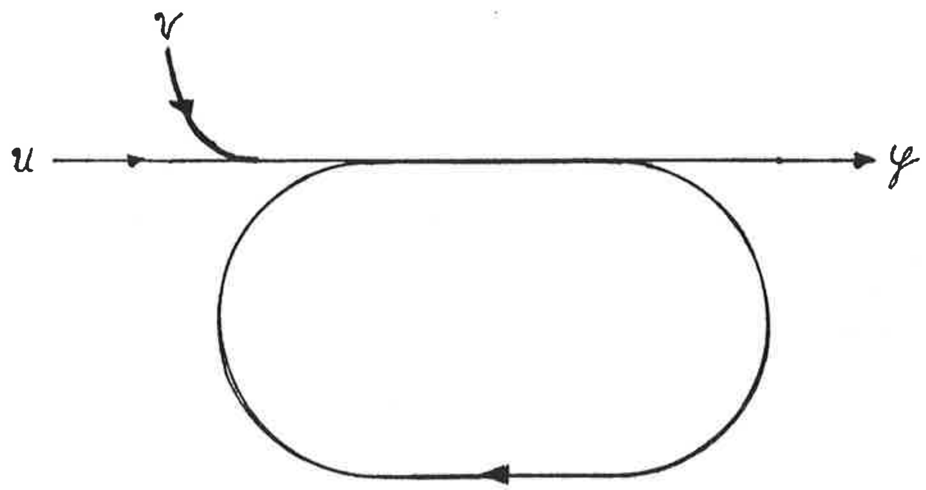
# 5. Experiment med kommersiella adaptiva system

108.

Projekt i adaptiv reglering

# K O M E R S I E L L   R E G U L A T O R

FIRST LOOP




Lund   April 1988

Kenneth Carlsson

Ola Söderström

## INLEDNING

Detta är ett projekt utfört inom kursen Adaptiv Reglering ,våren 1988.Avsikten med projektet är att närmare undersöka en kommersiell regulator av typen First Loop från First Control Systems.

## PRAKTISK UPPKOPPLING

När man bygger upp ett system med en F.C. kopplar man ihop ett antal färdiga moduler.Detta görs via ett tämligen användar-fientligt snitt.Förutom in,ut och själva reglermodulen finns ett stort antal moduler som t.ex. möjliggör filtrering av insignalerna..En typisk enkel uppkoppling kan se ut enligt nedan:

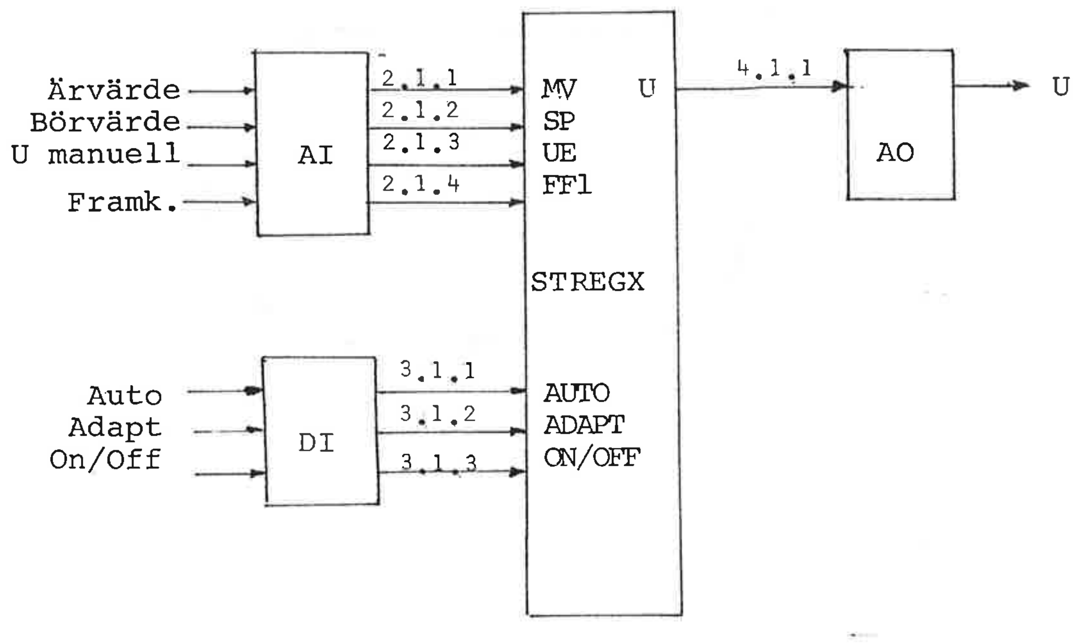



Fig. 1 Typisk enkel uppkoppling av controller

## HUVUDPRINCIP

Firstlines regulator arbetar med rekursiv parameterskattning och indirekt polplacering.I princip ser den skattade modellen ut som (förenklat):

$$AY=Bu+Cv$$

Det önskade beteendet på systemet karaktäriseras som ett första ordningens system med fördröjning.Man har även möjlighet att placera polen själv.

## VIKTIGA PARAMETRAR

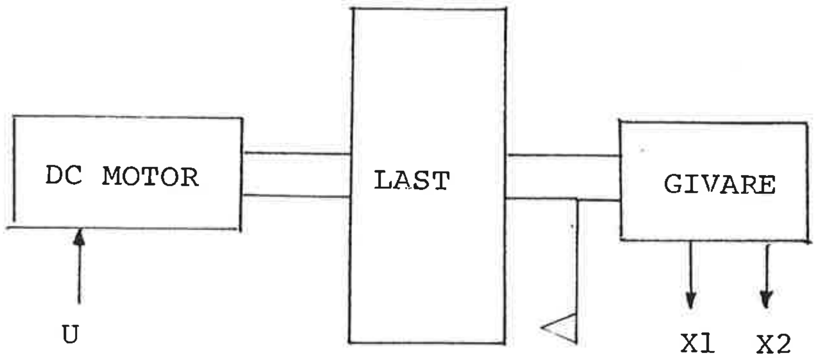
NA : A-polynomets gradtal (1-5).
NB : B-polynomets gradtal (1-5).
NC : C-polynomets gradtal (1-5).
MD : Processens fördröjning (1-5).
SAMP: Sampeltid.
POLE : Det slutna systemets önskade pol.

## VIKTIGA SIGNALER

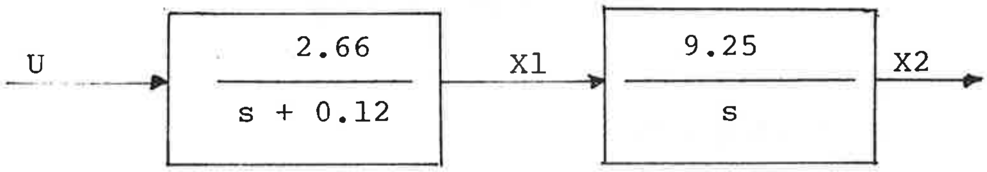
MV : y,ärvärdet.
SP : $u_r$,börvärdet.
UE : Manuell styrsignal.
FF1,FF2 : Framkopplingssignaler.
u : Regulatorns styrsignal till processen.

# REGLERING AV SERVO

Dc-servo, lägesstyrning:




Enligt laborationen i Digital reglering kan processen beskrivas enligt:




Där $x_1$=läge,$x_2$=hastighet,u=insignal och v=störning.
Detta gér att överföringsfunktionen ges av:

y=(2.66*9.25)/(s(s+0.12))u , då detta samplas fås

$y=(b_1q+b_2)/(q^2+a_1q+a_2)u$ , d.v.s. det borde vara lämpligt med NA=2 och NB=1.

Praktiska prov visar att med NA=3 och NB=2 ställer regulatorn in sig ganska bra.En pol i 0.99 ger med SAMP=5 ett mjukt stegsvar medan en pol i 0.95 ger en liten översläng.

Några betraktelser av intresse är:

-Valet av gradtal verkar inte vara särskilt kritiskt.Det gäller bara att inte välja för små gradtal.Dock verkar systemet vara instabilare under insvängningsförloppet om gradtalen är alldeles för höga.

-Insvängningen i början är ett klart problem.Systemet blir mycket gärna instabilt,d.v.s. det svänger utanför givarens område.Att köra i manuell mod och sedan slå över till auto hjälper inte.

-Valet av pol,fördröjning/sampeltid är kritiskt.För snabb pol orsakar instabilitet,liksom felaktigt val av delay och sampeltid.

-Systemet svänger in sig på olika sätt från gång till gång även om parametrarna är de samma.Med t.ex. B=3,A=2,pol=0.95 och samp=3 kan man få systemet att ställa in sig på både stegsvar med liten översläng och helt bra stegsvar.

## REGLERING AV BOM-PROCESS

Vi försökte även reglera en process av bom-typ. P.g.a. häftiga insvängniningsförlopp fick vi ge upp försöken med kula,och ägna oss åt vinkelstyrning av själva bomen.Med något högre gradtal på polynomen än ovan gick det skapligt att reglera processen redan från början.Dock var stegsvaret lite väl långsamt.Därför minskade vi samplingstiden och fick ett bra resultat.Precis som tidigare hade vi problem med insvängningsförloppet i början. Detta medförde att bomen ibland kunde svänga utanför sitt område och koppla ur processen.Med följande parametrar fick vi bäst resultat:

NA : 4.<br>
NB : 6.<br>
POL : 0.97.<br>
SAMP: 2.

173.

## SAMMANFATTNING

Med en usel manual och ett direkt avskräckande användarsnitt kan man effektivt gömma många av en regulatortyps fördelar.

Lund

1988 - 04 - 27

# Automatinställning av Novatune

## Projektarbete i Adaptiv Reglering VT88

Fredrik Owman F84
Patrik Hylta F85
Niklas Sjövall F85

# Innehållsförteckning

## Uppgiften

Denna var att automatinställa novatune reglerenhet.Med automatinställa menar vi bestämning av samplingstid och i samband därmed prediktionshorisonten.

## Systembeskrivning

Novatune reglerenhet är ett generellt system för styrning och reglering av processer.Systemet innehåller ett modulbibliotek med olika funktionsmoduler.Den viktiga dessa är en självinställande adaptiv regulator.

Reglerenheten konfigureras direkt från ett underlag i form av ett funktionsschema som definierar det reglersystem som skall installeras.Konfigureringen innebär att man definierar kopplingar mellan olika funktionsenheter och mot yttervärlden,parametervärden och hur varje funtionsblock ska exekveras i tiden.

Ett funktionsblock består av ett antal ihopkopplade funktionsmoduler som definierar dels den interna funktionen och dels signalutbytet med andra block och omvärlden.Modulerna knytes samman med signaler vilka är av två olika typer,reella signaler och heltalssignaler.Vid konfigureringen ska förutom den strukturmässiga ihopkopplingen,även parametervärden till modulerna anges (Kan ej ändras under körning).

Varje funktionsblock ges en prioritetsnivå,triggningstyp och period.Triggningen kan ske på tre olika sätt:Klocka (Time),pulsräknare (PC) och internavbrott (SC).Perioden anger hur många avbrottspulser av blockets typ det ska gå för varje triggning av blocket.Avbrottspulserna kan vara av typ TIME,PC eller SC.

# Blocken

## Block1, Relay

Blocket är automatinställarens relä och det är realiserat som en trepunktsregulator med referensvärdet 0. Utsignalen från regulatorn (0 eller 1) förs till en omkopplare (SW5) så att blockets utsignal vid TPT10 är -1 eller +1. Processens utsignal tages in via TPT1. Regulatorn har en viss hysteres som kan inställas som parameter vid programmeringen.

## Block2, Star2

I detta block sitter den egentliga adaptiva regulatorn av typen star2. Referensvärdet erhålles från en a/d omvandlare AI1 och återkopplingen från processen kommer via TPT1. Styrsignalen finns på TPT10. Regulatorns parametrar måste ställas in vid programmeringen.

## Block3, Process

Processen realiseras i detta block som ett tredje ordningens system med hjälp av tre filter (FILT2,3,4) med samma brytfrekvens (inställes vid programmeringen). Fördröjningen i processen åstadkommes i fördröjningsmodulerna MDEL5,6,7,8,9. Insignalen tages in via TPT10 och utsignalen fås vid TPT1.

## Block4, Measure

Den egentliga mätningen av periodtid, fördröjning och amplitud sker i detta block.

Tid:

Insignalen föres från TPT1 till en komparator (COMP1) där den jämföres med 0. Under den tid som insignalen är positiv är signalen I1 1 och räknarna COUNT3, 4 räknar antalet triggpulser. Dessa heltalssignaler omvandlas till reella, decimala tal och finns sedan tillgängliga på TPT2. Nollställning av räknarna sker en triggpuls senare än den triggpuls som erhålles när insignalen blir negativ (och som ger en programstyrd avbrottspuls på kanal 1, CH1)

Amplitud:

Denna mätes genom att successivt jämföra signalen med densamma, en triggpuls tidigare. Om den senare är större så lagras den i en hållkrets, SW12. Amplituden erhålles vid TPT3 när avbrottspulsen ovan kommer. Registreringen nollställes samtidigt som tidmätaren med ZERO11. ZERO8 håller komparatorns ingång på noll under insignalens negativa del.

Fördröjning:

När insignalen har nått sitt maximum lagras det aktuella tidsvärdet och kan hämtas vid TPT4.

## Block5, Check

I detta block kontrolleras den relativa skillnaden mellan två på varandra följande periodtider. Om avvikelsen är mindre än 10% så erhålles ett programstyrt avbrott på kanal 2, CH2. Urvalet sker i en alarmmodul, ALARM5. Medelvärdet av de godkända tiderna finns på TPT5. Medelvärdena för de aktuella fördröjningstiderna finns på TPT7

## Block6, Parameters

Se "Val av samplingstid och prediktionshorisont".

## Val av samplingstid och prediktionshorisont

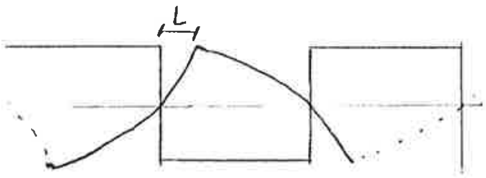
Generellt gäller att samplingstid och prediktionshorisont inte får väljas för korta med hänsyn till dynamik och tidsfördröjning hos processen. Bäst men känsligast reglering ger en kortare samplingstid och längre prediktionshorisont.

**Samplingstid:**

Praktiska erfarenheter säger att man bör sampla 3-6 ggr per halv-period (motsvarande wh=0.5-1). Vi har grovt indelat utfallsmöjligheterna i två fall A och B.

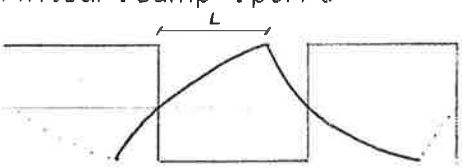
I fall A har vi en försumbar tidsfördröjning (<0.5 **T**samp) varvid vi samplar 5 ggr per halvperiod.

Alltså: **T**samp=**T**per/10




I fall B måste vi ta hänsyn till en tidsfördröjning varvid vi samplar mer sällan, 3 ggr per halvperiod.

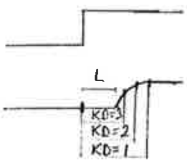
Alltså: **T**samp=**T**per/6




**Prediktionshorisont:**

Ett nödvändigt krav är att **ph** väljs längre än processens tidsfördröjning, annars kan inte **ph** användas för att bestämma styrvärde. **Ph** bör vara så lång tid som det tar för den ostyrda processens stegsvar att uppnå en avsevärd del (20-50%) av slutvärdet. I Novatune väljer användaren parametern **KD** (heltal).

Prediktionshorisonten=**KD***Tsamp




För processer utan tidsfördröjning väljs **KD**=1. Då blir **ph**=Tsamp vilket verkar vettigt.

När tidsfördröjning finns tas hänsyn till denna. Tidsfördröjningen adderas sålunda i prediktionshorisonten enligt följande:

**Ph**=Tsamp+Tförd=**KD***Tsamp    **KD**=1+Tförd/Tsamp

Ovanstående val av samplingstid och prediktionshorisont är mha logiska moduler enkelt realiserat i block 6.

## Experiment

Vid simulerade följande processer: $1/(s+0.2)^3$ och $e^{-sL}/(s+0.2)^3$ med tidsfördröjning L=2. Nedanstående värden på samplingstid och **KD** (prediktionshorisont=**KD***Tsamp) användes:

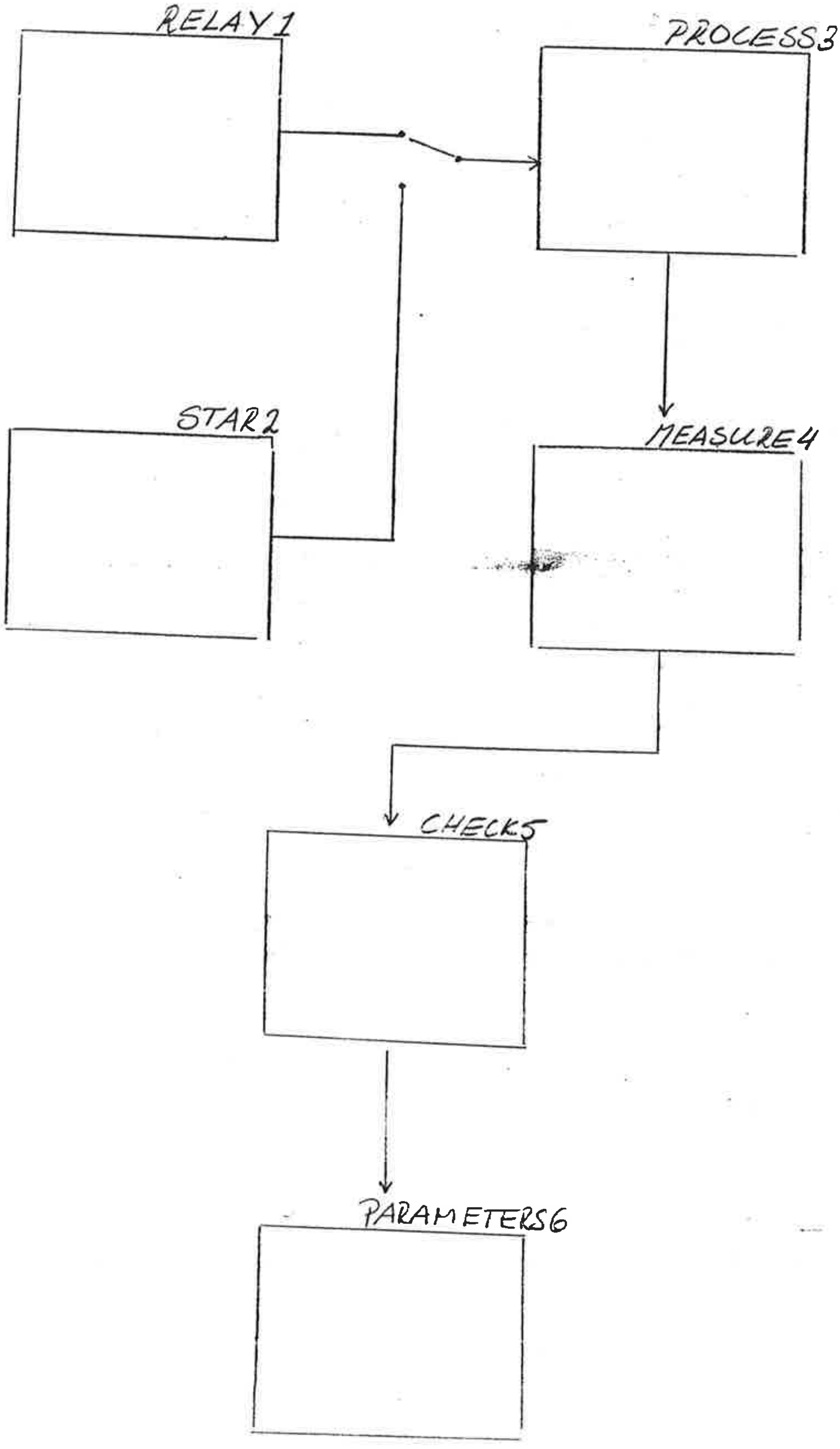
| | Tsamp(sec) | **KD** | Plottnr |
|---|---|---|---|
| **L=0:** | 1 | 2 | 1 |
| | 2 | 2 | 2 (automatinställt) |
| | 8 | 2 | 3 |
| **L=2:** | 2 | 4 | 4 |
| | 3 | 4 | 5 |
| | 4 | 4 | 6 (automatinställt) |
| | 6 | 4 | 7 |
| | 3 | 2 | 8 |
| | 6 | 5 | 9 |

## Sammanfattning

Vid studie av de bifogade plottningarna upptäcker vi till vår stora glädje att våra automatinställda värden på samplingstid och prediktionshorisont är de bästa. Vid automatinställningen utnyttjas inte signalens amplitud som innehåller viss information och som dessutom kan användas för reglering av reläamplituden. Vid våra mätningar tas inte hänsyn till störningar vilket är en (stor) nackdel vid praktisk användning.
Amplitudmätningen fungerar inte för mycket långa periodtider (storleksordning 2000 ggr samplingstiden) och små amplituder eftersom stegen i a/d omvandlaren då blir för små.
Tyvärr finns inte någon möjlighet på regulatorn att automatiskt föra in de uträknade parametrarna, utan det hela måste stoppas och programmeras om.

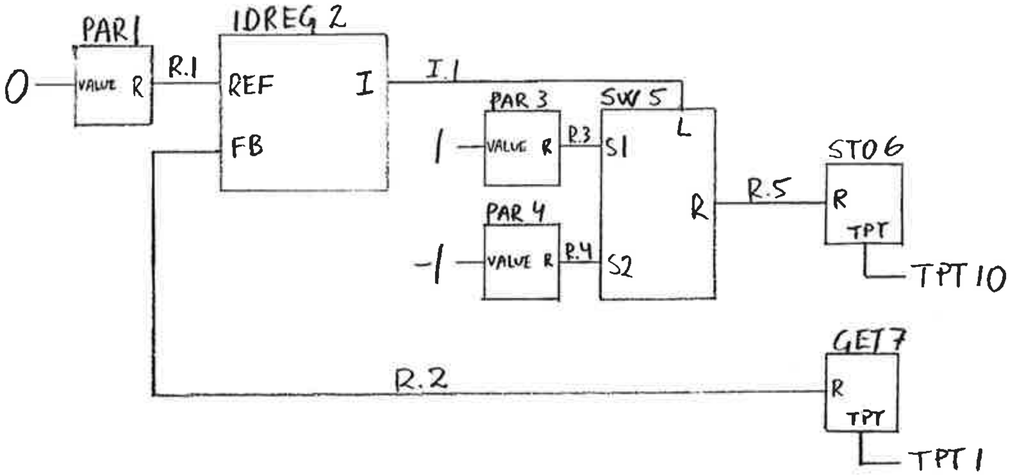
A

Blocköversikt

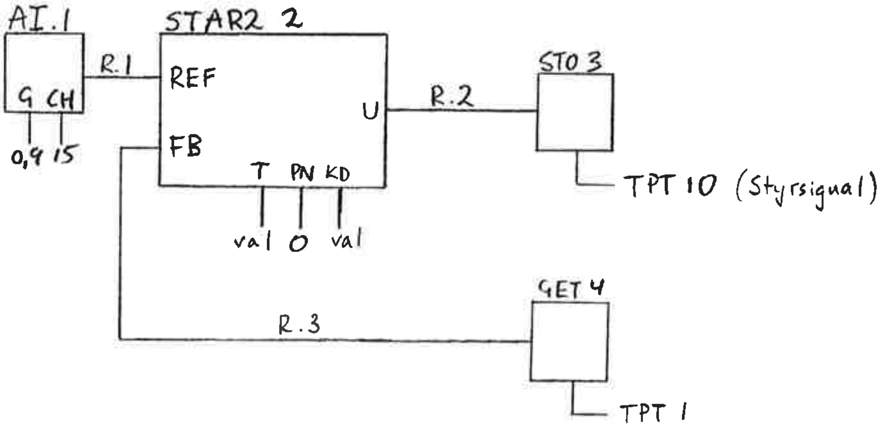


B

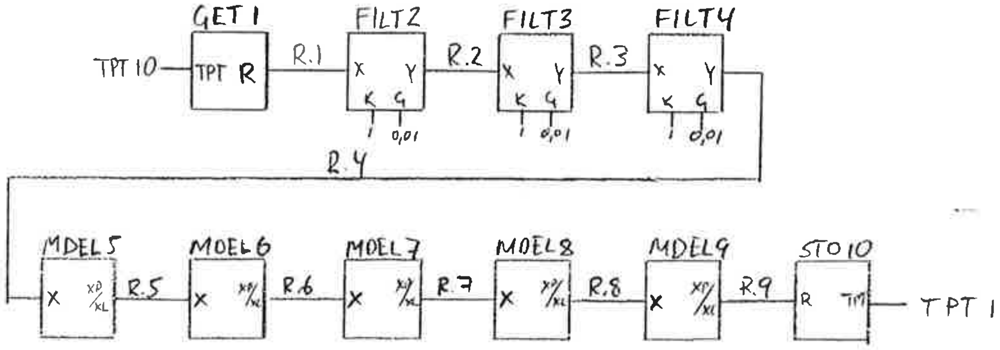
Block 1, Relay: Type time, period 1, prior 2



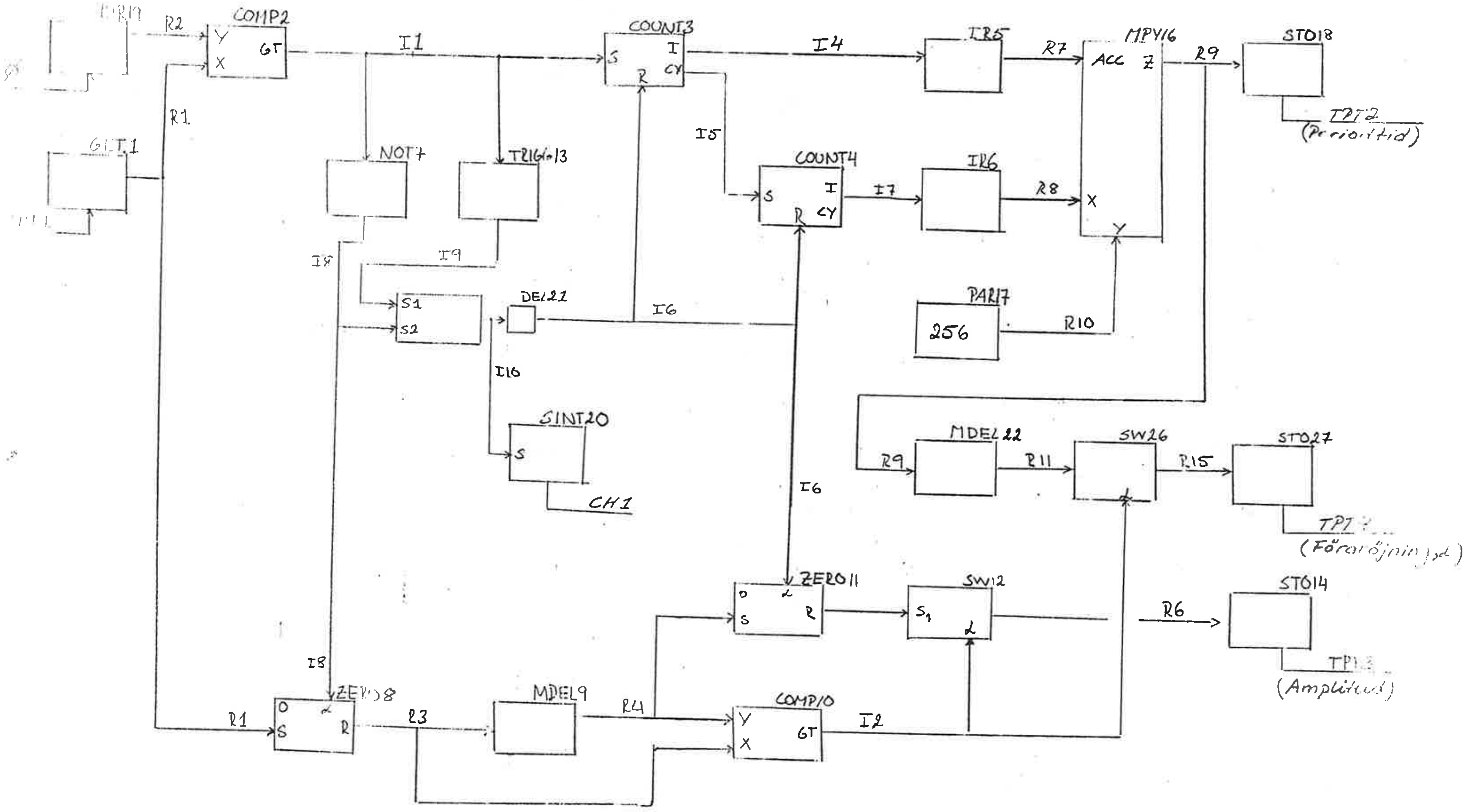

Block 2, Star2: Type time, period 10, prior 3




Block 3, Process: Type time, period 1, prior 2

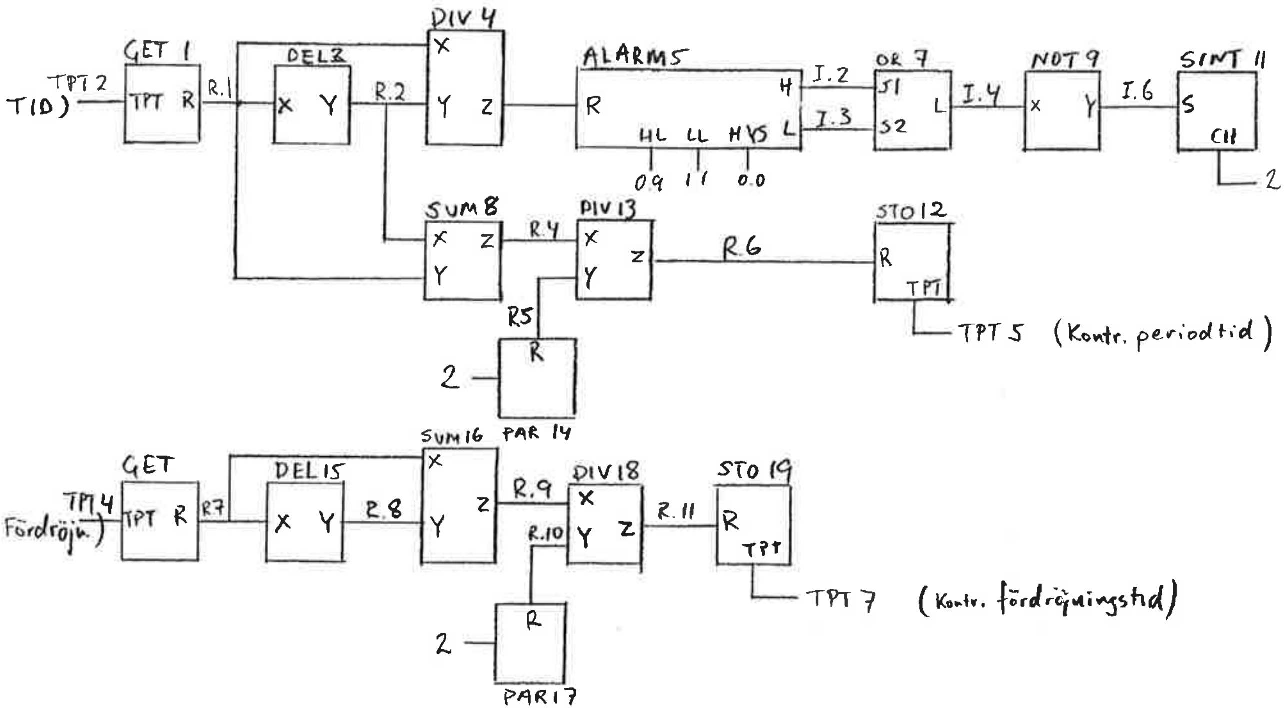


Block 4: Measure : Type time, period 1, prior 2

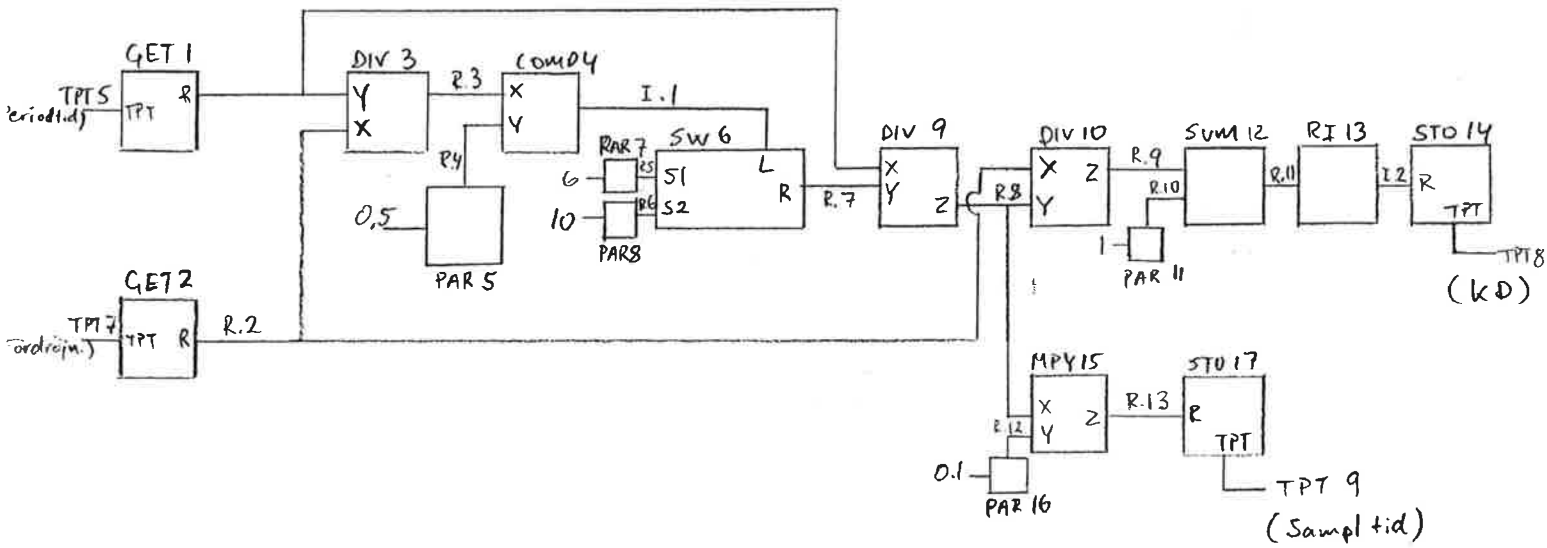


D

## Block 5, Check: Type SC1, period 1, prior 1

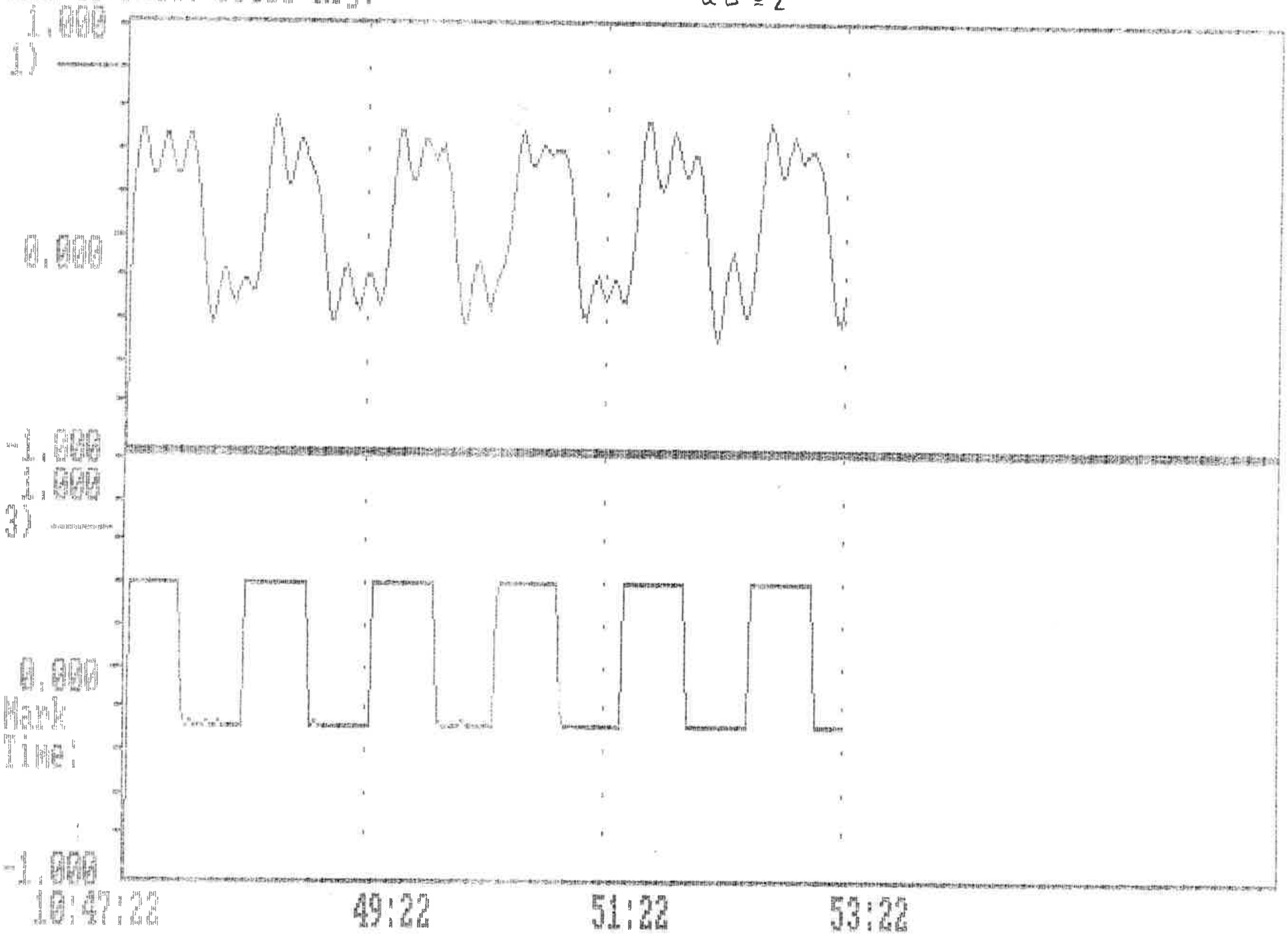



## Block 6, Parameters: Type SC2, period 1, prior 1



End of file. Press key!

$uD = 2$ $(S+1)^3$



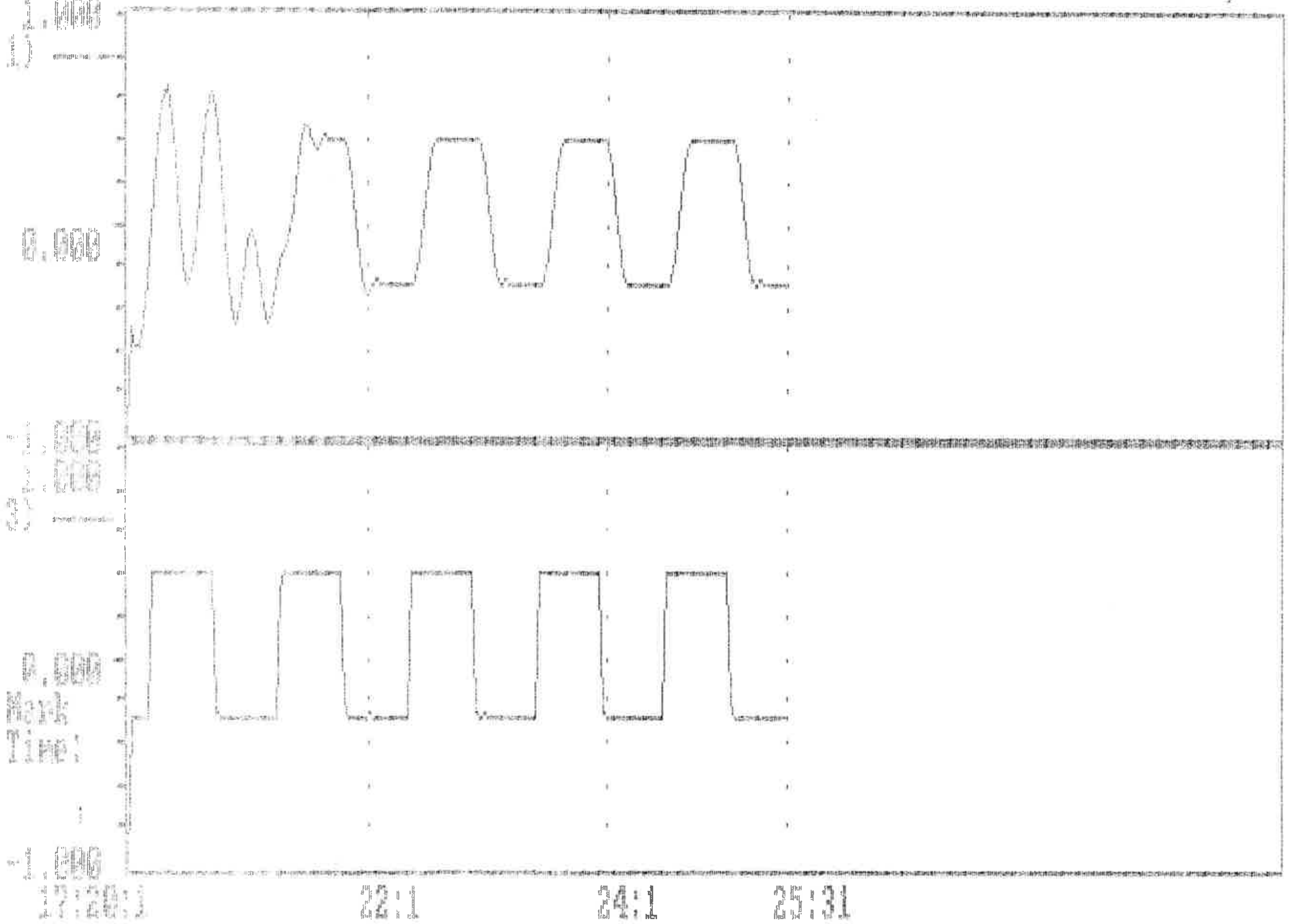

(1)

End of file. Press key!

STAL 2.H

$T_S = 4$
$K_D = 2$

$\frac{1}{(s+1)^3}$

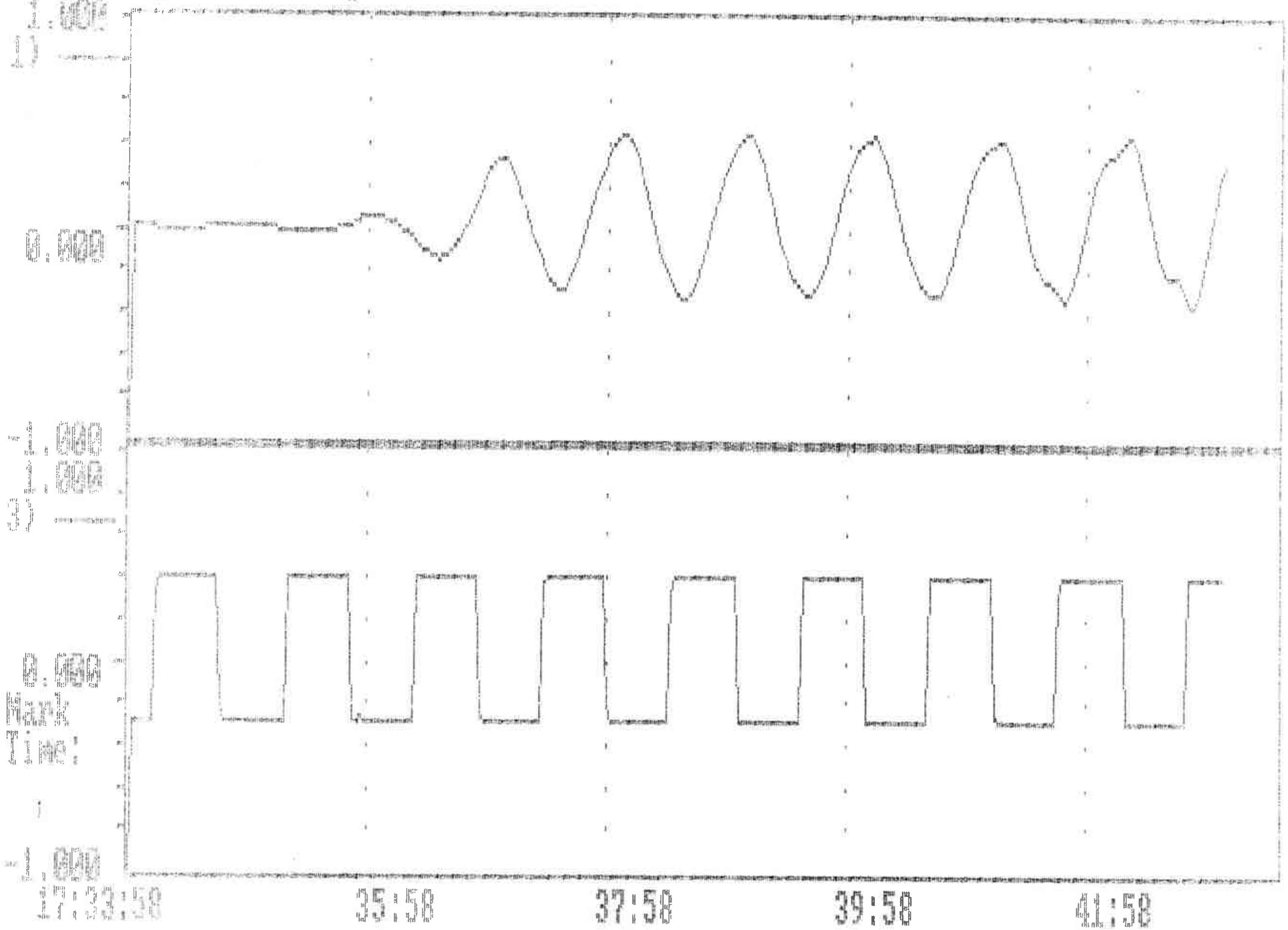



(2)

STAK 2.16

$\frac{1}{(s+1)^3}$ $T_i = 16$ KD = 2

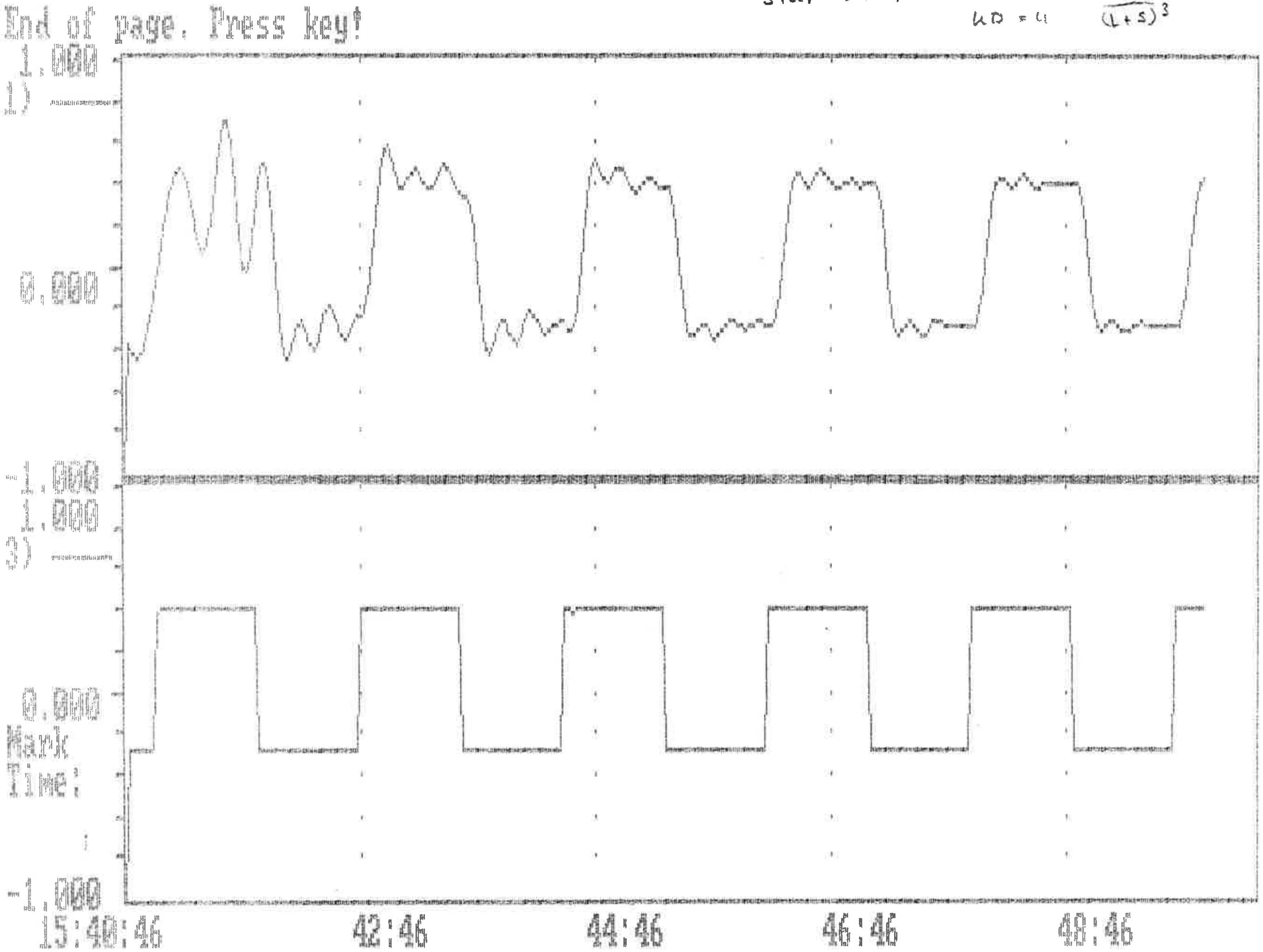
End of page. Press key!




100.

(3)

Star 2.417 KD = 4 $(1+s)^3$

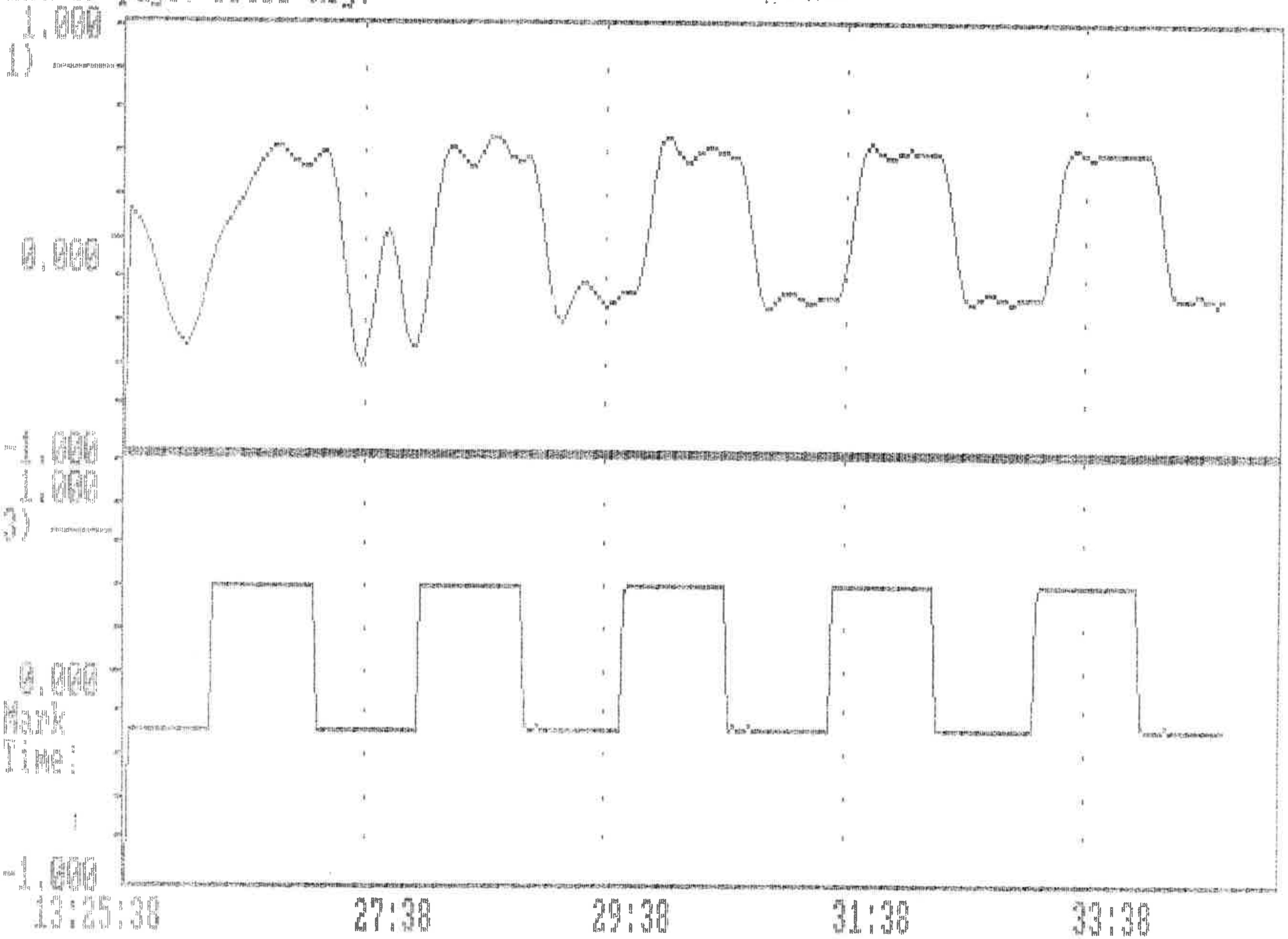



4

End of page. Press key!

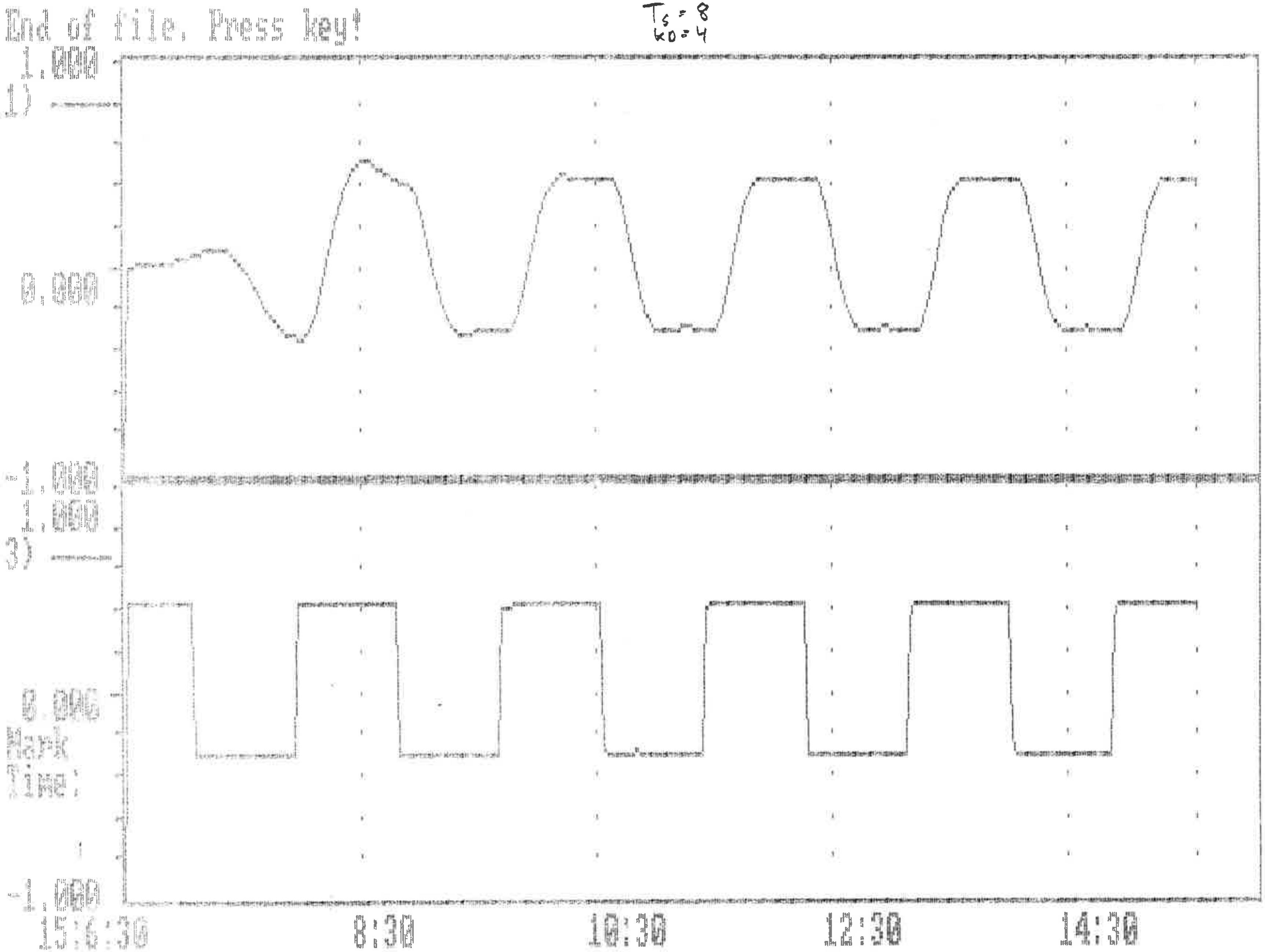
$T_s = 6$ Leb = 4

1: a rider



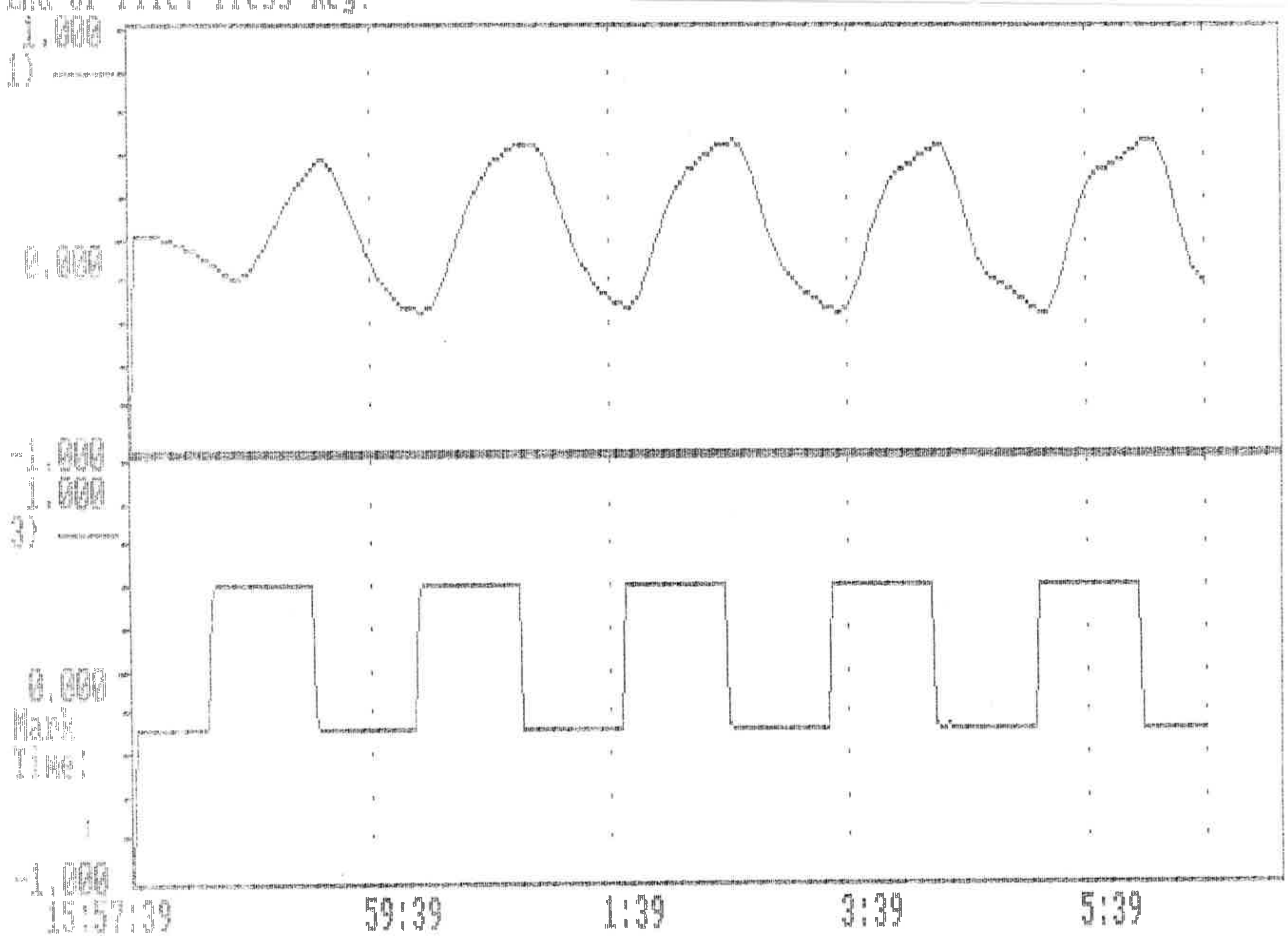

(5)

Ts = 8
k0 = 4

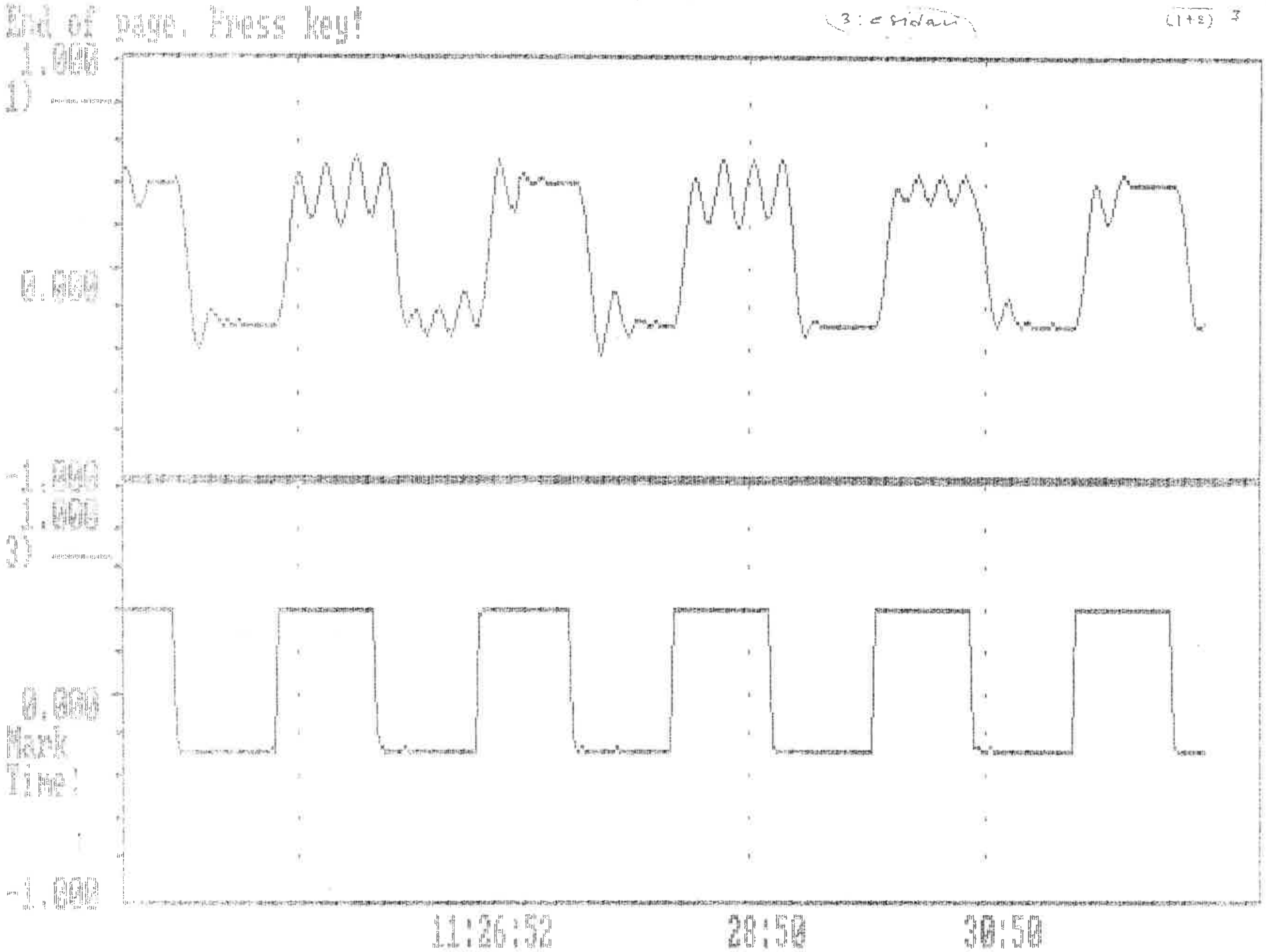

End of file. Press key!
1.000
1)
0.000
-1.000
1.000
3)
0.000
Mark
Time:
-1.000
15:6:30
8:30
10:30
12:30
14:30

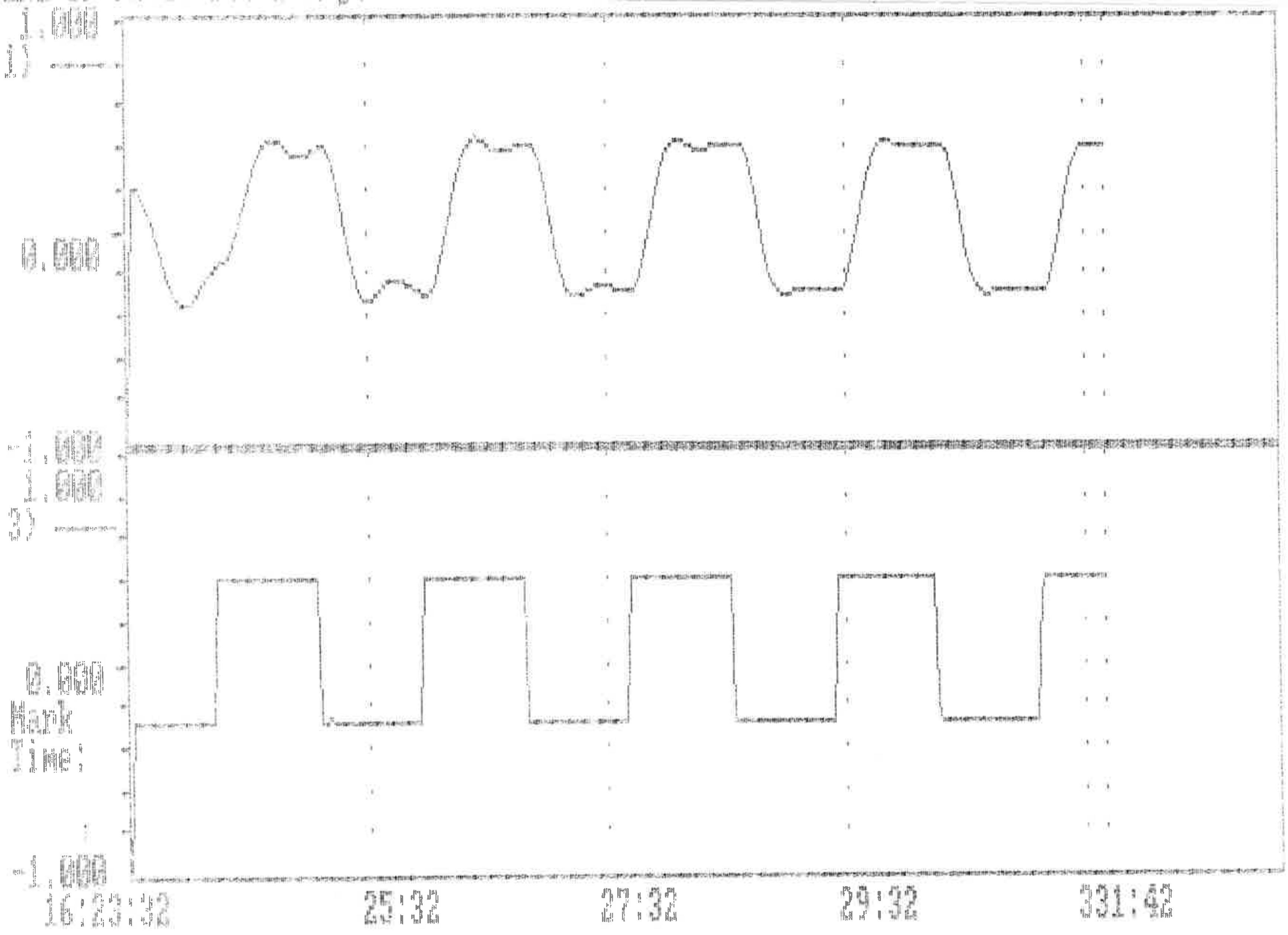
190

9


1.000
2.000
1.000
1.000
3)
0.000
Mark
Time!
-1.000
15:57:39
59:39
1:39
3:39
5:39

7


End of page. Press key!
1.000
1)
0.000
-1.000
1.000
3)
0.000
Mark
Time!
-1.000
11:26:52
28:50
30:50


1.000
0.000
1.000
0.000
0.000
Mark
Time:
-1.000
16:23:32
25:32
27:32
29:32
331:42

9

# 6. Experiment med automatinställning

# Automatinställning med parameterstyrning

Ulf Andersson
Ola Persson
Mårten Åkesson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
April 1989

# Inledning

Som ett led i kurserna Adaptiv Reglering och Datorimplementering av Reglersystem vid Institutionen för Reglerteknik, Lunds Tekniska Högskola, har ett kombinerat projekt genomförts. Uppgiften gick ut på att implementera en automatinställare med parameterstyrning. Stor vikt lades på användargränssnittet med mushantering och diagramritning som viktiga delar. Programmet skrevs i Modula 2. En realtidskärna och ett grafiksystem stod till förfogande.

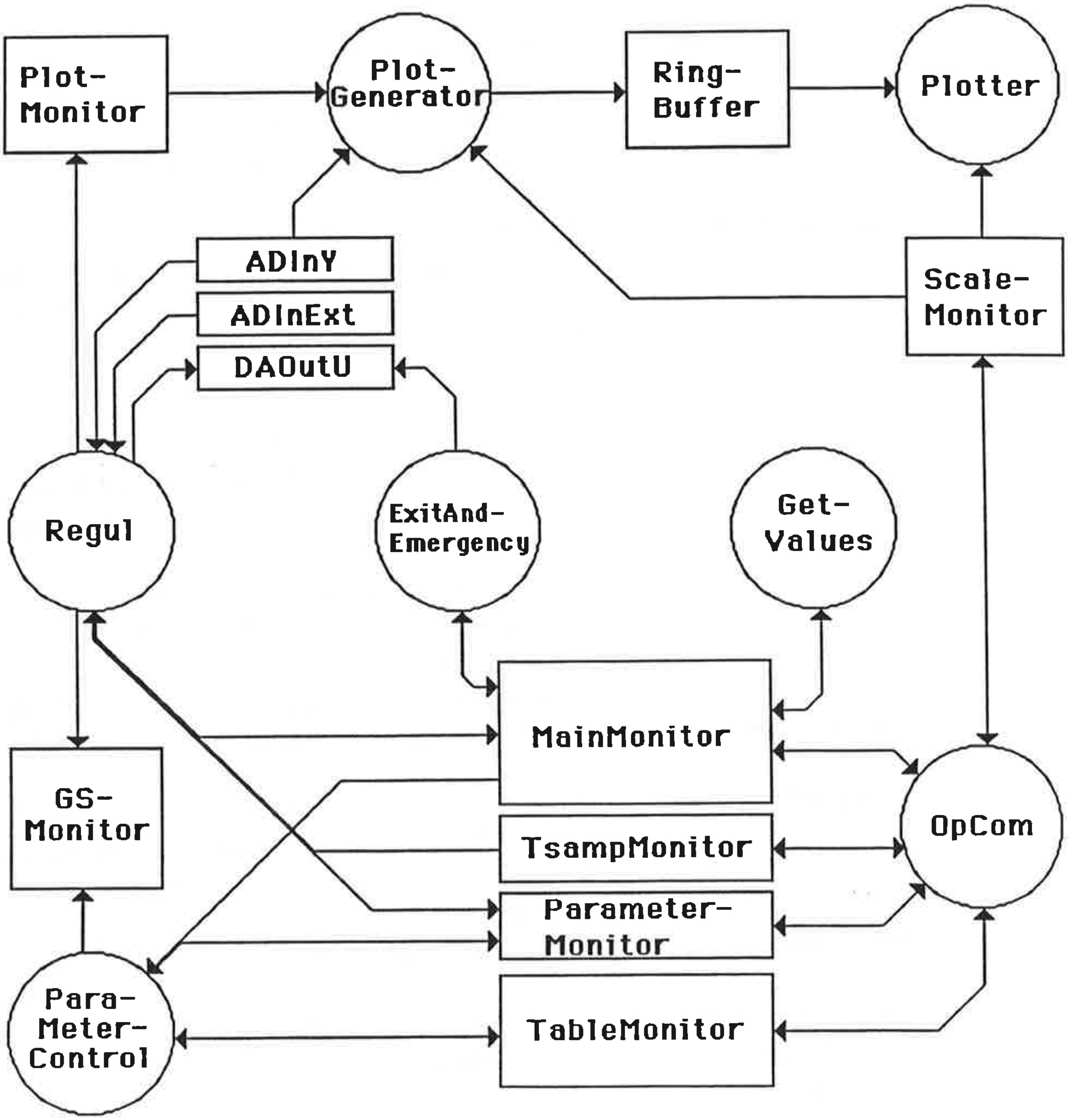
Automatinställaren realiserades med relämetoden och algoritmen för beräkningen av PID-parametrarna utgjordes av Ziegler-Nichols självsvängningsmetod. Parameterstyrningen implementerades med hjälp av en länkad lista. Det gör det möjligt att efterhand stoppa in nya element med aktuella parametervärden. Interpolering mellan punkterna kan göras, men det är även möjligt att ange intervall där parametrarna är konstanta. Flera listor kan ligga parallellt om intresse finns att köra parameterstyrning med avseende på olika variabler.

Programmet provkördes på en labprocess där vattennivån i en tank reglerades med en PID-regulator.

146.

## Programbeskrivning

Programmet består av 7 processer och 11 monitorer. Dessa kommunicerar enligt processgrafen nedan.




Figur 1. Processgraf.

Monitorerna som sköter synkroniseringen är följande:

**MainMonitor:** Denna monitor har hand om vilken mod (Man,PID,Tune) som regulatorn befinner sig i. Dessutom finns aktuella signaler (Uc,U,Y,Ext) lagrade här. Ext är en extern signal som kan vara intressant.

**ParameterMonitor:** De aktuella regulatorparametrarna finns lagrade i denna monitor.

**TsampMonitor:** Denna monitor lagrar samplingstiden.

**GSMonitor:** Variabeln som väljer regulatorparametrar handhas i denna monitor.

**TableMonitor:** Tabellen till parameterstyrningen finns realiserad i denna monitor med dubbellänkade listor. Man kan ha en lista för varje variabel till parameterstyrningen. Ett element i tabellen har följande datatyp:

```
TypeOfData=RECORD
        Low,High:REAL;
        P       :ParType;
        Hard    :BOOLEAN;
        TuneInfo :TuneParType;
    END;
```

Variabeln Hard indikerar om interpolering är tillåten. TuneInfo är en outnyttjad variabel i vilken man skulle kunna lägga information om eventuell automatinställning, tex reläamplituden.

**Plotmonitor:** Denna monitor lagrar aktuella U- och Uc-signaler som ska plottas i diagram.

**Ringbuffer:** Plottern är en process med låg prioritet som ska kunna hämta signalerna U, Uc och Y vid tillfälle. Därför läggs dessa i en ringbuffert i väntan på plottning.

**ScaleMonitor:** Tidsskalan för plottningen lagras i denna monitor.

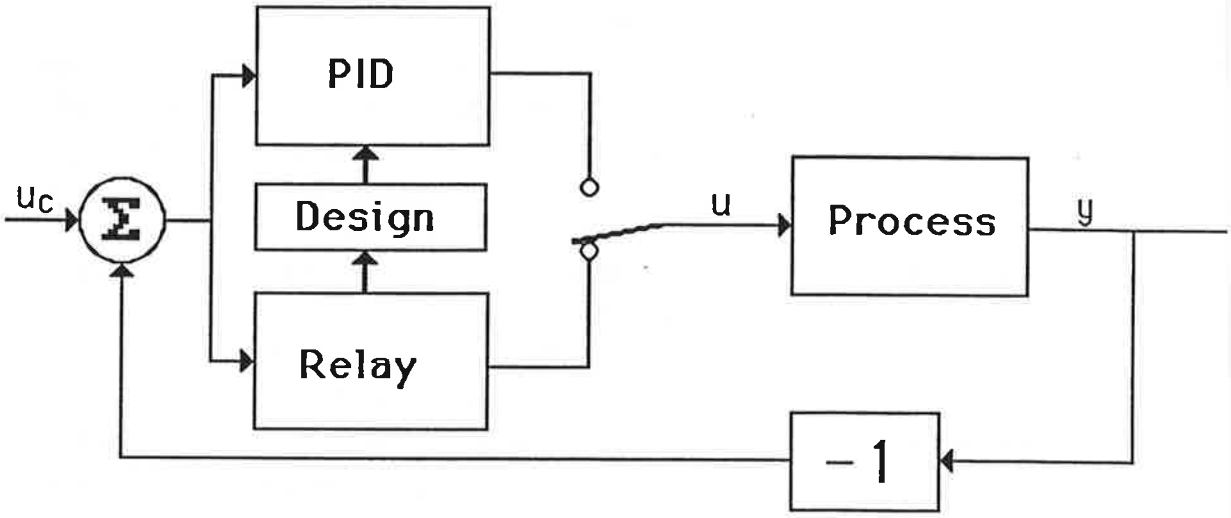
Det finns ytterligare tre monitorer. Dessa skyddar kommunikationen med AD och DA-omvandlare vid inläsning av mätsignaler och utställning av styrsignaler.

Programmet använder sig av nedanstående processer:

| | |
|---|---|
| **OpCom:** | Denna process handhar all kommunikation med användaren. Den har låg prioritet. |
| **Regul:** | Här kontrolleras vilken mod som programmet befinner sig i och beroende av detta anropas olika rutiner för beräknande av utsignal. Nya regulator-parametrar hämtas alltid in för vidarebefordran till regulatorrutinen. Efter avslutad tuning sparas regulatorparametrarna i parametermonitorn. |
| **ParameterControl:** | Under tiden gainscheduling är påkopplad ser denna process till att det alltid finns korrekta regulator-parametrar i parametermonitorn. Efter avslutad tuning sparas parametrarna i tabellen. |
| **ExitAndEmergency:** | Nödstopp och programavslutning tas omhand av denna process. |
| **GetValues:** | Här sköts inläsningen av styrsignal och börvärde via musklick i plotgraferna. |
| **PlotGenerator:** | Denna process genererar plotdata i en takt definierad av tidsskalan i plotten och antal plotpunkter på skärmen. |
| **Plotter:** | En lågprioriterad process som plottar på skärmen. |

## Automatinställning

Nedanstående text beskriver ett automatiskt sätt att bestämma parametrarna i en PID-regulator. Med hjälp av till exempel relämetoden skaffar man sig information om processen. Denna information användes sedan för att utforma en PID-regulator.




Figur 2. Blockdiagram för automatinställare med relä.

Den enkla iden bakom relämetoden är att många processer självsvänger med begränsad amplitud när de återkopplas med ett relä. Återkopplingen bibehålls tills en stationär svängning har uppnåtts. För processer som är störda av brus är det lämpligt att använda ett relä med hysteres i omslaget. Metoden finns noggrant beskriven i *Automatic Tuning of PID Regulators** sid 45-49. Som resultat av metoden erhålles en punkt på Nyquistkurvan.

Med hjälp av en modifierad Ziegler-Nichols metod kan man nu flytta den erhållna punkten bort från det dåligt dämpade området. För en noggrannare beskrivning av metoden hänvisas till *Automatic Tuning of PID Regulators** sid 63-66.

* Åström, K.J. och T. Hägglund: Automatic Tuning of PID Regulators.

.ν00.

# Parameterstyrning

Parameterstyrning är ett sätt att anpassa regulatorn till rådande förhållanden. Först bestämmer man signal för parameterstyrningen. Beroende på denna signals värde hämtar man regulatorparametrar ur en tabell. Tabellen kan antingen byggas upp med hjälp av automatinställning, beskriven i förra kapitlet, eller genom egen inmatning.

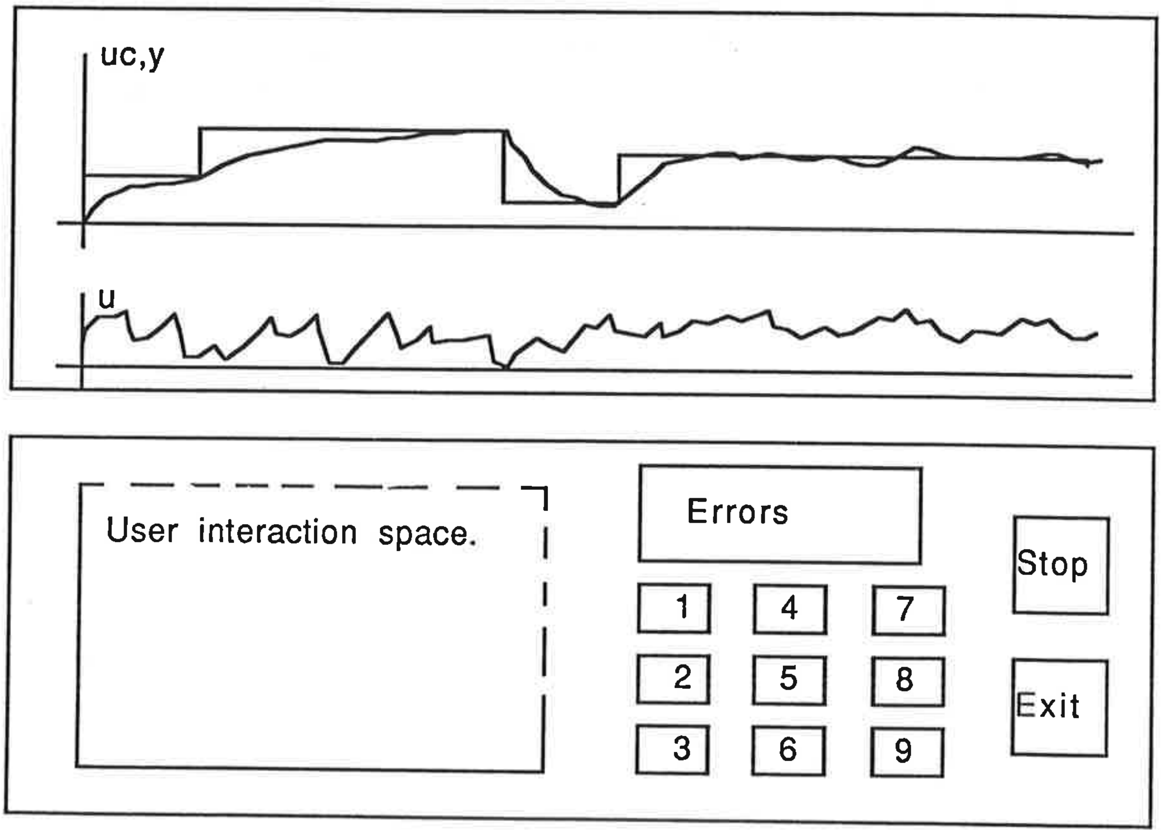
Elementen i tabellen innehåller förutom regulatorparametrar även en logisk variabel som anger om interpolering är tillåten eller inte. Om interpolering är tillåten finns också information om för vilket värde parametrarna gäller, annars finns undre och övre intervallgräns sparad.

Mellan punkter används linjär interpolering, dock ej vid ändpunkterna där man i stället använder den yttersta punktens parametrar.

## Manual

### Översikt

Nedanstående figur beskriver skärmens utseende vid uppstart. På den övre halvan av skärmen visas mätsignalen y, börvärdet uc och styrsignalen u. Nedtill visas ett antal muskänsliga knappar vars funktion beskrives nedan.




1:Finns ej vid uppstart.
3:Ed. GS.
5:Ch. Par.
7:Man.
9:Tune.

2:GS Off.
4:GS Var.
6:Scale/Tsamp.
8:PID.

Programmet startas genom att skriva **autoreg**. Man kan köra programmet i 3 olika moder, Man(uell), PID (reglering med PID-regulator) eller Tune (automatisk inställning av PID-parametrar). Byte av mod sker genom klick i någon av rutorna 7, 8 eller 9. I manuell mod ställer man sedan styrsignalen u direkt genom att klicka i u:s diagram. I PID-mod ställer man börvärdet genom att klicka i uc:s diagram. Man kan också byta regulatorparametrar genom att klicka i ruta 5 (Change Parameters). Ruta 5 och ruta 6 (Tidsskala/Samplingstid) använder båda utrymmet märkt "User interaction space". En meny kommer upp och du får ange vilken parameter du vill ändra genom att mata in en siffra följt av **<CR>**. Sedan matar du in det nya värdet på parametern du ville ändra följt av **<CR>**. Efter avslutad editering slår du e(xit) för att spara de inamtatade värdena eller q(uit) om du inte vill spara. Under tiden editering sker kan du inte klicka i rutorna 1-9. Under alla faser av programmet kan man alltid använda de till höger placerade rutorna märkta Stop och Exit. Stop bromsar processen genom att koppla över till manuell mod och ställa ut styrsignalen 0. Exit stannar programmet och för dig åter till operativsystemet.

### Automatinställning

Programmets tredje mod heter Tune och används av operatören för att automatiskt beräkna parametrarna i PID-regulatorn. Arbetsgången bör vara följande:Kör manuellt eller med en egen regulator upp processen till ett stationärt värde på mätsignalen y. Vid manuell körning måste du klicka i det översta diagrammet så att uc och y överensstämmer. Klicka sedan i ruta 9 (Tune). Du får besked i rutan markerad "Error" om resultatet av inställningen.

### Parameterstyrning

Parameterstyrningen kan användas på flera olika sätt, gemensamt för alla är dock att du måste bestämma dig för vilken variabel du vill använda som styrsignal till tabellen. Detta gör du genom att klicka i rutan 4 (GS Var.). Här används liksom tidigare "User interaction space" och aktuell variabel är markerad med en *. Ett klick i ruta 2 (GS Off) sätter igång parameterstyrning vilket syns genom att rutans text ändras till GS On. Det lättaste är sedan att göra automatinställning vid ett antal olika punkter, dessa lagras då automatiskt i tabellen.

Det kan nu finnas ett behov av att titta och editera i tabellen. Detta göres genom att klicka i ruta 3 (Ed(it) GS(tabell)). Man får då upp en ny uppsättning av muskänsliga knappar. Enligt tidigare numrering har de då följande betydelse.

| | |
|---|---|
| 1:Plot/Text | 2:Get Act. |
| 3:Exit | 4:Ed. Tmp. |
| 5:Forward | 6:Backward |
| 7:Change | 8:Insert |
| 9:Delete | |

Tillsammans med de nya knapparna sker också visning i "User interaction space", där visas två uppsättningar tabellelement. Ett temporärt element till vänster som du kan jobba med och det första elementet i listan till höger. Om du vill skapa dig en översikt över tabellelementen kan du göra det genom att klicka i ruta 1 (Plot/Text). Parametrarna plottas då i relativ skala. För att åter se tabellelementen i skriven form trycker du en gång till i ruta 1. Den regulator som för tillfället används kan hämtas till det temporära elementet genom att klicka i ruta 2 (Get Actual). Ett klick i ruta 3 gör att du lämnar editeringsmenyn och återvänder till huvudmenyn. För att ändra i det temporära elementet klickar du i ruta 4 och du kan då liksom tidigare ange vilken parameter du vill ändra och dess nya värde. Med ett e(xit) lämnar du "User interaction space" och du kan åter trycka i rutorna. Med klick i rutorna 5 respektive 6 går du framåt respektive bakåt i tabellen. Om du vill ändra ett element i listan stegar du dig fram till det elementet, därefter editerar du i det temporära elementet tills du är nöjd och slutligen trycker du på 7 (Change). Om du istället vill sätta in det temporära elementet som ett nytt element i listan trycker du på 8 (Insert). För att ta bort ett element ur listan trycker du 9 (Delete).

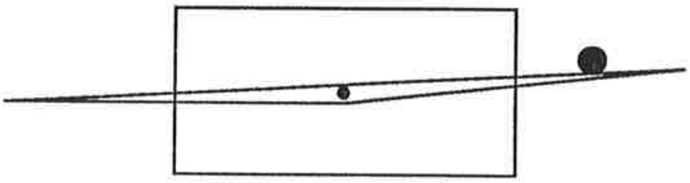
## Avslutning

Den resulterande automatinställaren fungerar inte helt tillfredsställande. Till exempel ger den ibland ganska stora skillnader i parametrarna mellan olika inställningar kring samma punkt. Det verkar som om stationariteten av utsignalen vid början av en automatinställning är av avgörande betydelse för resultatet. Dålig stationaritet ser också ut att medföra asymmetriska reläsvängningar.

Ziegler-Nichols designmetod har visat sig ge en regulator med hög förstärkning, dvs orolig styrsignal och översläng vid börvärdesändringar, och då kretsförstärkningen och periodtiden som beräknats av automatinställaren är lite osäkra kan resultatet bli lite varierande.

Operatörssnittet fungerar i stort sett tillfredsställande, dock borde operationerna på tabellen för parameterstyrningen ha valts annorlunda eftersom nuvarande operationer gör tabellediteringen tungarbetad.

# *Auto-tuners för bomprocess*



**Projekt i Adaptiv reglering och Datorimplementering av reglersystem vid Institutionen för reglerteknik på Lunds Tekniska Högskola**


av: Rickard Nilsson
Olof Samuelsson
Mikael Wilroth

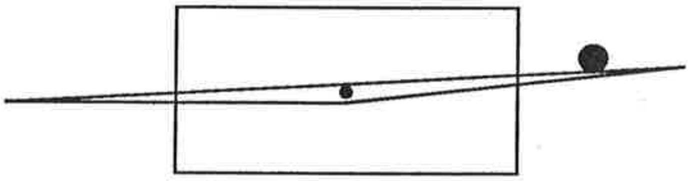
# Innehåll

# Inledning

Föreliggande rapport beskriver ett kombinerat projektarbete i kurserna Adaptiv reglering och Datorimplementering av reglersystem vid Institutionen för reglerteknik på Lunds Tekniska Högskola. Målet med projektet har varit att implementera självinställande PID-regulatorer i Modula-2 på en IBM-AT och använda dessa för att reglera en av institutionens så kallade bomprocesser.

# 1 Hårdvarubeskrivning

Processen som skall regleras består av en metallkula på en kulränna (så kallad bom) vars mittpunkt är fäst på en vridbar horisontell axel. Vridningen styrs av en servomotor och insignal till processen är motorns spänning. Utsignaler är bommens vinkel och kulans position på bommen. Bilaga 3 beskriver sammankopplingen av regulatorer och process.



Figur 1 Bom med kula

# 2 Mjukvarubeskrivning

Programmet är indelat i åtta implementationsmoduler -- sex processer och två monitorer. Processgrafen kan studeras i bilaga 4. Som programspråk användes Modula-2.

## 2.1 Referensvärdesgenerator, RefGen

Processen RefGens uppgift är att i realtid lägga sinus-, fyrkant- eller triangelvågformade börvärdessekvenser i variabeln SetPoint i RegMonit. När RefGen slås på (Waveform < > Off) kontrolleras senast givna börvärde (= Level). Om Level ligger utanför den valda svängningens område (-Amplitude till +Amplitude) görs först en rampning till områdets ytterkant varefter generatorn startas. Om i stället Level ligger i svängningens område beräknas först den tid vid vilken svängningens värde = Level. Därefter startas generatorn med den beräknade tiden. Samtliga svängningar görs symmetriskt kring motorspänning/vinkel/kulläge = noll.

## 2.2 Referensvärdesgeneratorns monitor, RefMon

Referensvärdesgeneratorn RefGen styrs via monitorn RefMon. I denna finns aktuella värden på vågform (sinus-, fyrkant-, triangelvåg och off), amplitud och periodtid (i sekunder) hos svängningen samt operationerna PutRefVal och GetRefVal som lägger in respektive hämtar värdena på nämnda storheter. När OpCom gör PutRefVal sätts dessutom variabeln Level i RefMon till aktuellt börvärde. Ömsesidig uteslutning ordnas med en semafor.

## 2.3 Plotprocess, Plot

Plotprocessen plottar bör- och ärvärden till den yttre och inre loopen (bommens vinkel och kulans position), även motorns spänning skrivs ut. Värdena hämtas från regulatorprocessen via en brevlåda.

## 2.4 Regulatorns monitor, RegMonit

RegMonit förmedlar värden mellan i huvudsak OpCom och Regul, men även mellan RefGen och Regul och mellan Quickstop och Regul. De data som finns i monitorn är regulatorernas parameteruppsättningar, auto-tuningparametrar, regulatormoder, samplingstid, hur ofta sampel ska plottas samt ett börvärde. I monitorn ligger också procedurer för operationer på skyddade data. Vidare sköter RegMonit uppdateringen av Man-, Auto- och Tune-knapparna på bildskärmen.

## 2.5 Regulatorprocess, Regul

Regulatorprocessen reglerar bommen med hjälp av två kaskadkopplade PID-regulatorer, se även bilaga 3. Regulatorns olika moder styrs från RegMonit som bestämmer om den yttre och den inre loopen befinner sig i moderna Man, Auto eller Tune. Regulatorn ändrar själv mode när tex kulan fallit av vid Auto eller Tune.

Processen samplas med samplingstiden Ts. Samplingstiden används även när nya tidsdiskreta parametrar beräknas från K, Ti och Td.

Bommen tillsammans med motorn har en resonanstopp i överföringsfunktionen vid 7 Hz. Denna omöjliggjorde auto-tuning på innerloopen. Lösningen på problemet blev att sända vinkelregulatorns utsignal till motorn genom ett femte ordningens lågpassfilter av FIR-typ.

Här nedan följer ett Modula-2-program med huvuddragen i regulatorprocessen.

```
BEGIN
   Init;
   LOOP;
      WaitForTick;
      GetRegPar;           (* Hämtar mode, PID par  och SetPoint*)
      CASE Mode2 OF                 (* Yttre loopen *)
         Man : Uc1:=SetPoint                         |
         Auto  : Uc1:=CalculatePIDOutput         |
         Tune:  Tune2;
      IF NOT Mode2=Tune2 THEN
         CASE Mode1 OF
            Man: MotorVoltage:=SetPoint          |
            Auto: MotorVoltage:=CalculatePIDOutput      |
            Tune:Tune1;
      END;
   END;
END.
```

## 2.6 Operatörskommunikation, OpCom

Processen OpComs uppgift är att sköta kommunikationen med användaren. OpCom håller reda på musposition och musklick. Vid klick i muskänsligt område utförs de funktioner som hör till området. Varje muskänslig ruta innehåller status, färger, mode och olika funktioner som tex möjlighet att lagra värden. Genom att klicka på Man1, Man2, Auto1, Auto2, Tune1 och Tune2 kan regulatorernas mode i RegMonit ändras. Parameterfönstret ritas upp vid klick i 'Show Para'-rutan. Är parameterfönstret öppnat kan parametrarnas värden och referensgeneratorns funktion ändras. De nya värdena lagras i knapparna tills man klickar i Activate-rutan. Då läggs de in i monitorena. Ändrade regulatorparametrar, tuningparametrar, samplingsintervall och plottäthet läggs i RegMonit och i RefMon läggs ändrad funktion, amplitud, period och samplingintervall. Genom att klicka eller hålla musknappen nere i rutorna med pilar kan man ändra på SetPoint i RegMonit. När regulator 2 är i Auto kan positionsbörvärde väljas genom att klicka direkt på bommen.

### 2.7 Inläsning av nya värden, ReadProcess

Denna process startas av OpCom när något nytt värde skall läsas in. Processen väntar på inmatning, känner av vad som skrivs på tangentbordet och kollar så att det är ett tal inom rätt intervall. Om inmatat värde är felaktigt raderas det inskrivna och ett nytt tal måste skrivas in. När värdet är tillåtet eller parameterfönstret stängs lägger processen sig åter att vänta på en ny inläsning.

### 2.8 Nödstoppsprocess, QuickStop

Processen QuickStop har högst prioritet och väntar hela tiden på musklick i skärmens Stop-ruta. När detta händer stängs RefGen av, börvärde till Regul sätts till 0 och båda regulatorerna ställs i manuell mode. Texten "Regulators stopped" skrivs ut på skärmen.

## 3 Bruksanvisning

När programmet startas upp befinner sig båda regulatorerna i manuell mode och plotfönstret visas.

Följande händer om man klickar i någon av rutorna på understa raden:

**Stop:** Nödstopp -- börvärdesgeneratorn stängs av och båda regulatorerna går över till manuell mode. Vinkeln på bommen fryses. Under bommen på skärmen visas meddelandet 'Regulators stopped'.

**Freeze/Plot:** Plottningen stoppas alternativt återstartas. Ej utritade mätvärden sparas ej.

**Man/Auto/Tune:** Val av regulator-mode. Aktuell mode markeras med grön färg. Knappar som får användas är röda. Icke tillåtet val av mode anges med grå knapp.

**Show Para/ Show Plot:** Plotfönstret alternativt parameterfönstret visas. I parameterfönstret kan regulatorernas parametrar, samplingstiden, plottningsintervallet och parametrar till auto-tuningen och referensgeneratorn ändras. Vidare kan referensgeneratorn sättas på och av.

**Exit:** Exekveringen av programmet stoppas. Utsignalen från DA-omvandlaren ligger kvar.

**Parameterfönstret:**
I parameterfönstret visas de olika parametrarnas värden. De parametrar som det är tillåtet att ändra värde på visas i röda rutor, övriga i gråa rutor. Klick i en ruta ger möjlighet att ändra aktuell storhet. Om värdet är tillåtet att ändra blir rutan gul och en inmatningsruta fås. Om det nya värdet ligger inom rätt intervall hamnar det efter tryckning på <return> inom parentes bredvid det gamla värdet och rutan får blå färg. Så fort någon parameter har ändrats blir activate-knappen grön. Genom att klicka på activate-knappen lagras de nya värdena och rutorna får tillbaka sin röda färg. Detta ger

möjlighet att ändra flera parametrar samtidigt. Rutorna i parameterfönstret har följande innebörd med tillåtna intervall inom parentes:

**Regl/Reg2:**

| | |
|---|---|
| **K:** | Regulatorns förstärkning. (0 - 99.99) |
| **Ti:** | Regulatorns integraldel. (-10-99.99) Negativt tal stänger av integraldelen. |
| **Td:** | Regulatorns derivatadel. ( -10 - 99.99) Negativt tal stänger av derivatadelen. |
| **To:** | Anti-windup-funktionens tidskonstant. Sätts vid auto-tuning till Ti. |
| **N:** | Derivatadelens övre gränsfrekvens. (0 - 99.99) |

**Tuning parameters:**

| | |
|---|---|
| **e:** | Relähysteres. (0 - 0.99) |
| **d:** | Reläamlitud. (0 - 0.99) |

**Reference Generator:** Akuellt läge markeras med grönfärgad ruta. Funktionen byts först vid klick i activate-rutan.

| | |
|---|---|
| **Sine:** | Sinusformad börvärdessignal. |
| **Square:** | Fyrkantsformad börvärdessignal. |
| **Triangle:** | Triangelformad börvärdessignal. |
| **Off:** | Referensgeneratorn avstängd. |
| **A:** | Referensgeneratorns amplitud. (0 - 0.99) |
| **P:** | Referensgeneratorns period i sekunder. (0 - 99.99) |

**Sample Interval:**
**Ts:** Samplingsintervallet i sekunder. (0 - 0.99)

**Plot Interval:**
**PI:** Plottningsintervallet i sekunder. (0 - 99.99)

**Inläsning av nya värden:**
Vid klick i parameterruta fås en inmatningsruta med texten 'Enter New Value:'. Det nya värdet skrivs in från tangentbordet följt av < return >. Om talet inte är inom det tillåtna intervallet accepteras det ej. Upprepad ändring före aktivering är möjlig.

**Börvärdesändring:**
Om regulator 2 inte är i Auto kan börvärde för regulator 1 ändras genom att klicka i rutorna med pilar under det vänstra plotfönstret. Hålls musknappen nere ökas börvärdet i accelererande takt. Är regulator 2 i Auto kan dess börvärde ändras antingen på samma sätt som för regulator 1 eller genom att klicka direkt vid önskad position på bommen på skärmen.

# 4 Auto-tuning

Auto-tuning kan utföras på både inner- och ytterloopen. Auto-tunern består av ett relä som kopplas in i stället för PID-regulatorn. Systemet kommer då att svänga med en viss periodtid och amplitud. Dessa två parametrar avläses när variationerna har avtagit tillräckligt mycket. Genom att vikta periodtiden och amplituden fås de nya PID-parametrarna K,Ti och Td. Metoden kan studeras i referens 1.

## 4.1 Procedur

När auto-tuning av innerloopen startas går bommen först till horisontellt läge varefter svängningen sätts igång. Grundinställningen på reläets hysteres (epsilon) är vald lite större än glappet i bommen och reläamplituden (d) har valts till 0.2 vilket ger en bra svängning.

För att kulan skall stanna kvar på bommen vid tuningen av ytterloopen måste svängningen ske symmetriskt kring horisontalläget. Detta åstadkoms genom att bommen vid start av programmet är fulkomligt horisontell (och startvinkeln läses in som horisontalläge vid uppstart). Tuningen startar när kulan är nära mitten och har en låg hastighet. Reläamplitud och -hysteres väljs mycket små för måttlig storlek på svängningen.

När tre svängningar i följd har tillräckligt lika periodtid anses svängningen stabil och tuningen stoppas. Därefter beräknas de nya PID-parametrarna och regulatorn går över i Auto. Om svängningen inte har blivit stabil innom 50 perioder eller kulan har fallit av avbryts tuningen, regulatorn sätts i Man och en felutskrift görs.

## 4.2 Parameterberäkningar

På innerloopen används en modifierad variant av Ziegler-Nichols metod för beräkning av K, Ti och Td från de uppmätta storheterna Ku och Tu. Ziegler-Nichols ger ett dåligt dämpat system vilket i innerloopen dock var önskvärt eftersom svängigheten minskar problemen som friktionen mellan kulan och bommen ger. I ytterloopen var det nödvändigt med hög dämpning och Ziegler-Nichols metod gick inte att använda. Vi valde därför egna viktningskonstanter enligt nedan som gav ett gott beteende. För att undvika "ryckig" styrsignal elimineras derivataverkan för K större än 2.

| Ziegler-Nichols | Modifierad Z-N | Egna |
|---|---|---|
| K=0.6Ku | K=0.35Ku | K=0.9Ku |
| Ti=0.5Tu | Ti=0.77Tu | Ti=5.0Tu |
| Td=0.12Tu | Td=0.2Tu | Td=0.5Tu |

# 5 Resultat

Beteendet som parametrarna från auto-tuningen ger skall inte jämföras med det som fås med korrekta handvalda parametrar. För god vinkelreglering används tex P-reglering medan auto-tuning ger ganska kraftig I-verkan. För just bomprocessen är emellertid den svängighet som I-delen ger positiv -- överslängen i vinkelregleringen ger en knyck som kraftigt bromsar/accelererar kulan vid lägesreglering. Vidare är auto-tuning på ytterloopen inte ens möjlig utan slängig innerloop pga dubbelintegralen mellan bommens vinkel och kulans läge.

Exempel på erhållna parametrar vid auto-tuning:

Innerloop
K=3.02
Ti=0.1
Td=0

| Ytterloop | Liten kula | Mellanstor kula | Stor kula |
|---|---|---|---|
| | K=0.17 | K=0.12 | K=0.13 |
| | Ti=16.1 | Ti=17.7 | Ti=17.2 |
| | Td=1.61 | Td=1.78 | Td=1.72 |

Resultaten varierar med ca ±5% mellan försöken.Egenskaperna hos lägesregleringen för den stora och den mellanstora kulan är klart acceptabla medan systemet med liten kula inte blir mer än precis stabilt.

# 6 Validering

Resultatet från tuningen av innerloopen beror på bommens och motorns egenskaper samt vilken omräkningsmetod som använts. Det är svårt att uttala sig om parametrarnas riktighet men utseendet på ett stegsvar är karaktäristiskt för ett system som är designat med Ziegler Nichols metod (tex fås så kallad quarter-amplitude-damping).

Ytterloopens parametrar beror förutom på innerloopen även på kulan. Vi hade möjlighet att prova olika storlekar på kula och fick då ovan nämda värden.

Enligt bilaga 1 fås större acceleration för stor kula än för liten. Förväntat tuningresultat är då att stor kula ger motsvarande mindre K så att hela systemets snabbhet ej beror av kulans storlek. Mätresultaten ovan med auto-tunad innerloop bekräftar ej denna teori. Det olikartade uppförandet för lägesreglering med olika kulor tyder på att även innerloopen (tex dess dämpning) beror av kulans storlek. För att minska detta beroende minskades innerloopens integralverkan för hand. Med K=2.5 och Ti=0.2 erhölls följande parameteruppsättningar för ytterloopen:

| Liten kula | Mellanstor kula | Stor kula |
|---|---|---|
| K=0.3 | K=0.22 | K=0.18 |
| Ti=5.95 | Ti=6.55 | Ti=8.5 |
| Td=0.6 | Td=0.66 | Td=0.85 |

Dessa värden på K överensstämmer i storleksordning väl med de förväntade enligt bilaga 1.

# 7 Slutsats

Auto-tuning av innerloopen gav som väntat gott resultat. Valet av reläets utamplitud och hysteres var inte helt trivialt, men med rätt storleksordning på värdena kan dessa ändras inom ett visst område utan inverkan på tuning-resultatet.Ytterloopen var betydligt känsligare för valet av viktningskonstanter och arbete fick läggas ner på att finna användbara värden. Överföringsfunktionen mellan bommens vinkel och kulans läge är en dubbelintegral på vilken det teoretiskt sett är omöjligt att utföra auto-tuning. Men tack vare att regleringen av innerloopen inte är ideal (man vill normalt sett ha en etta som överföringsfunktion mellan bör- och ärvinkel) fås ett komplexare system på vilket auto-tuning med reläåterkoppling går bra.

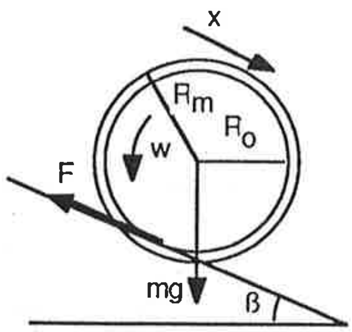
# Referenser

Referens 1 Åström K. J. och Wittenmark B., 1989. Adaptive Control. Addison-Wesley.

**Bilaga 1 Massiv kula i lutande ränna**

(Förstärkning från lutningsvinkel ß till acceleration för olika kulradier.)

För en kula i en lutande ränna gäller följande:




$$m\frac{d^2x}{dt^2} = mg\sin\beta - F$$

$$\frac{dx}{dt} = -wR_o$$

$$J\frac{dw}{dt} = -FR_o$$

där Ro är avståndet mellan kulans mittpunkt och det plan som rännans ovankanter ligger i och J är kulans tröghetsmoment.

Friktionskraften F elimineras:

$$m\frac{d^2x}{dt^2} = mg\sin\beta - \frac{J}{R_o}\frac{dw}{dt}$$

Rullvillkoret insättes:

$$m\frac{d^2x}{dt^2} = mg\sin\beta + \frac{J}{R_o^2}\frac{d^2x}{dt^2}$$

$$\frac{d^2x}{dt^2} = \frac{mgR_o^2\sin\beta}{mR_o^2+J}$$

För en massiv kula gäller

$$J = \frac{2}{5}mR_m^2$$

där $R_m$ anger kulans massradie.
Då ges kulans acceleration längs planet av

$$\frac{d^2x}{dt^2} = \frac{g\sin\beta}{1+\frac{2}{5}(\frac{R_m}{R_o})^2}$$

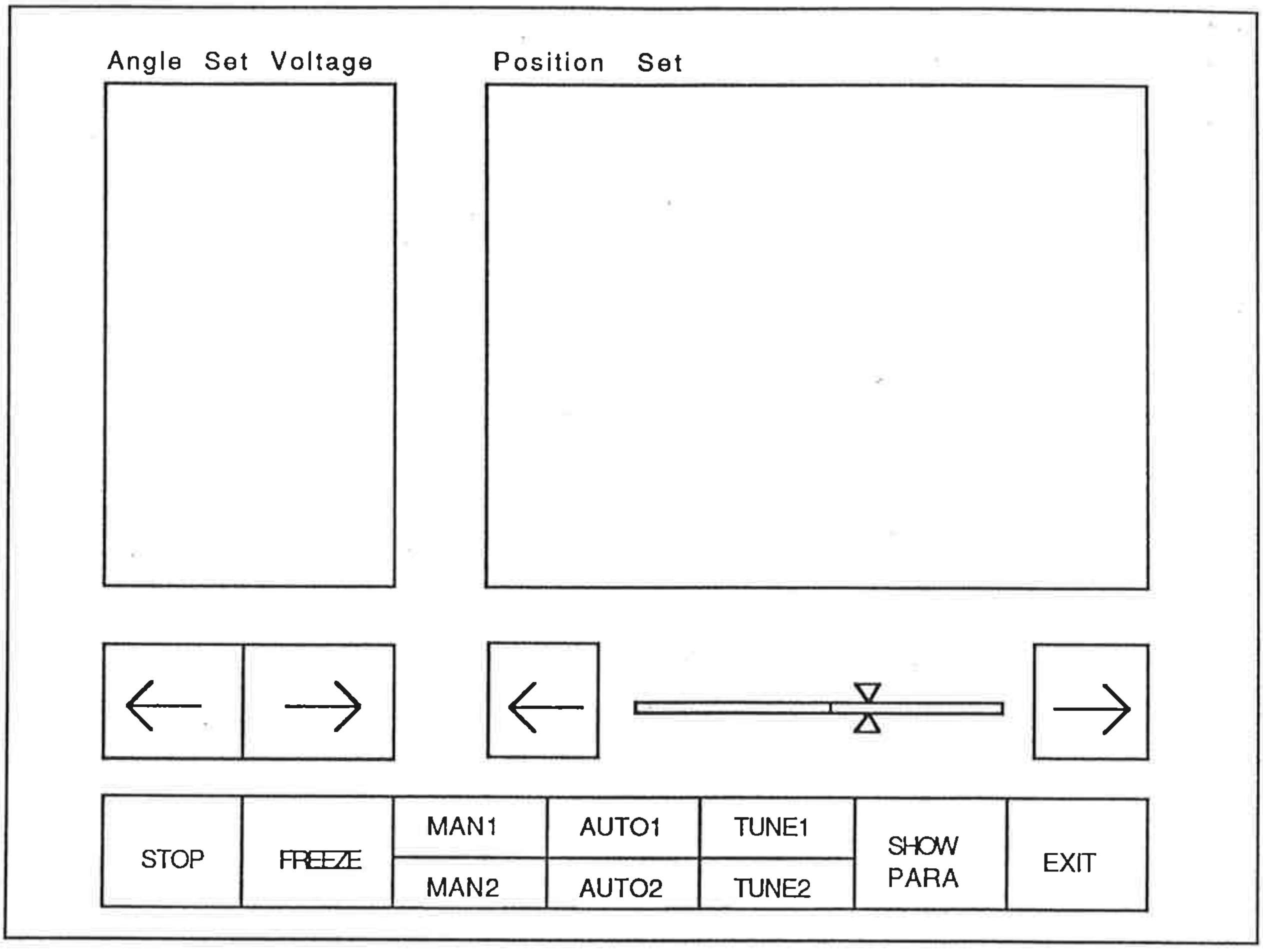
För de använda kulorna gäller följande:

| | Liten kula | Mellanstor kula | Stor kula |
|---|---|---|---|
| $\frac{R_m}{R_o}$ | 1.177 | 1.054 | 1.031 |
| $\frac{d^2x}{dt^2}$ | 0.643gsinß | 0.692gsinß | 0.702gsinß |

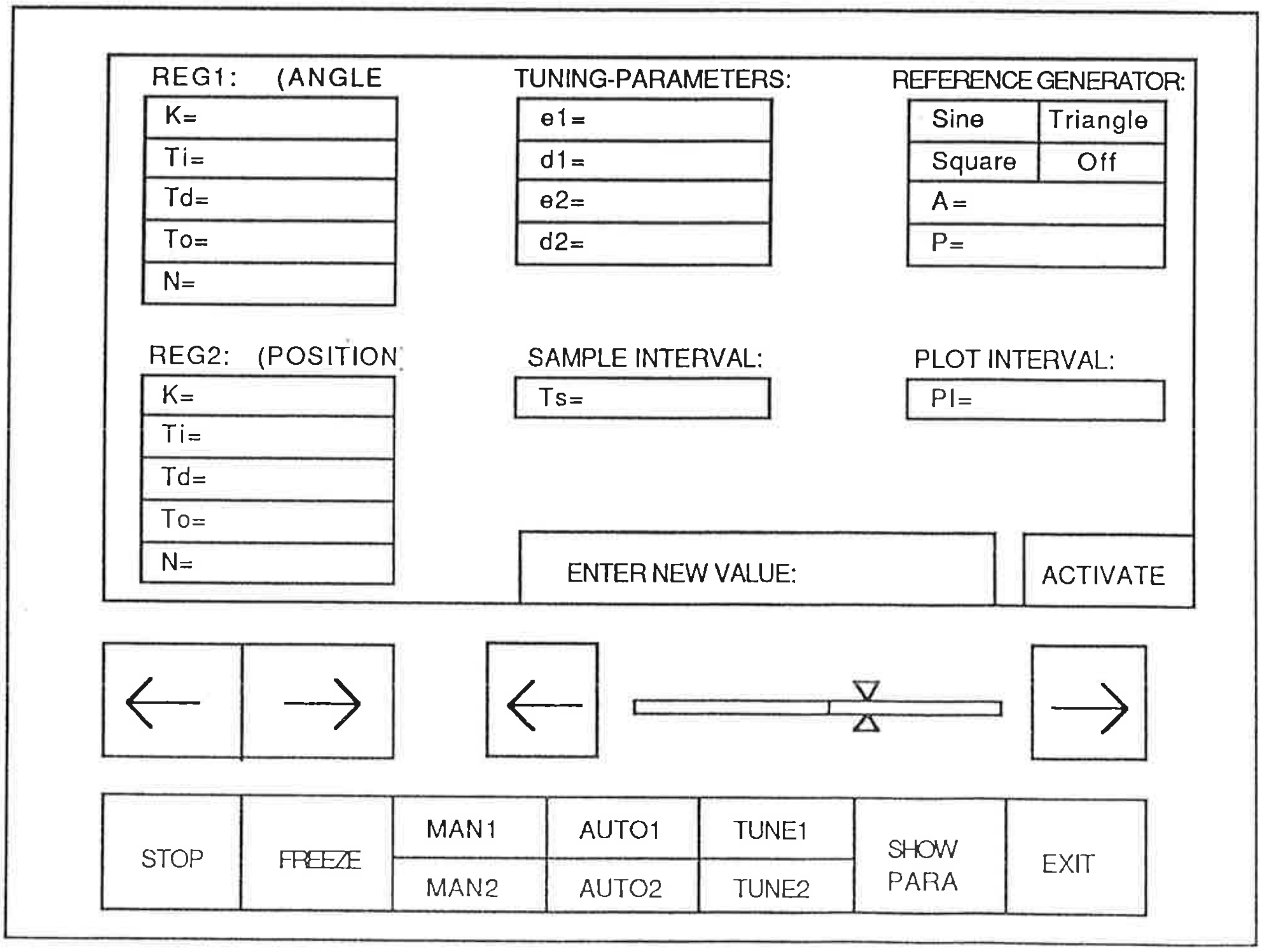
215.

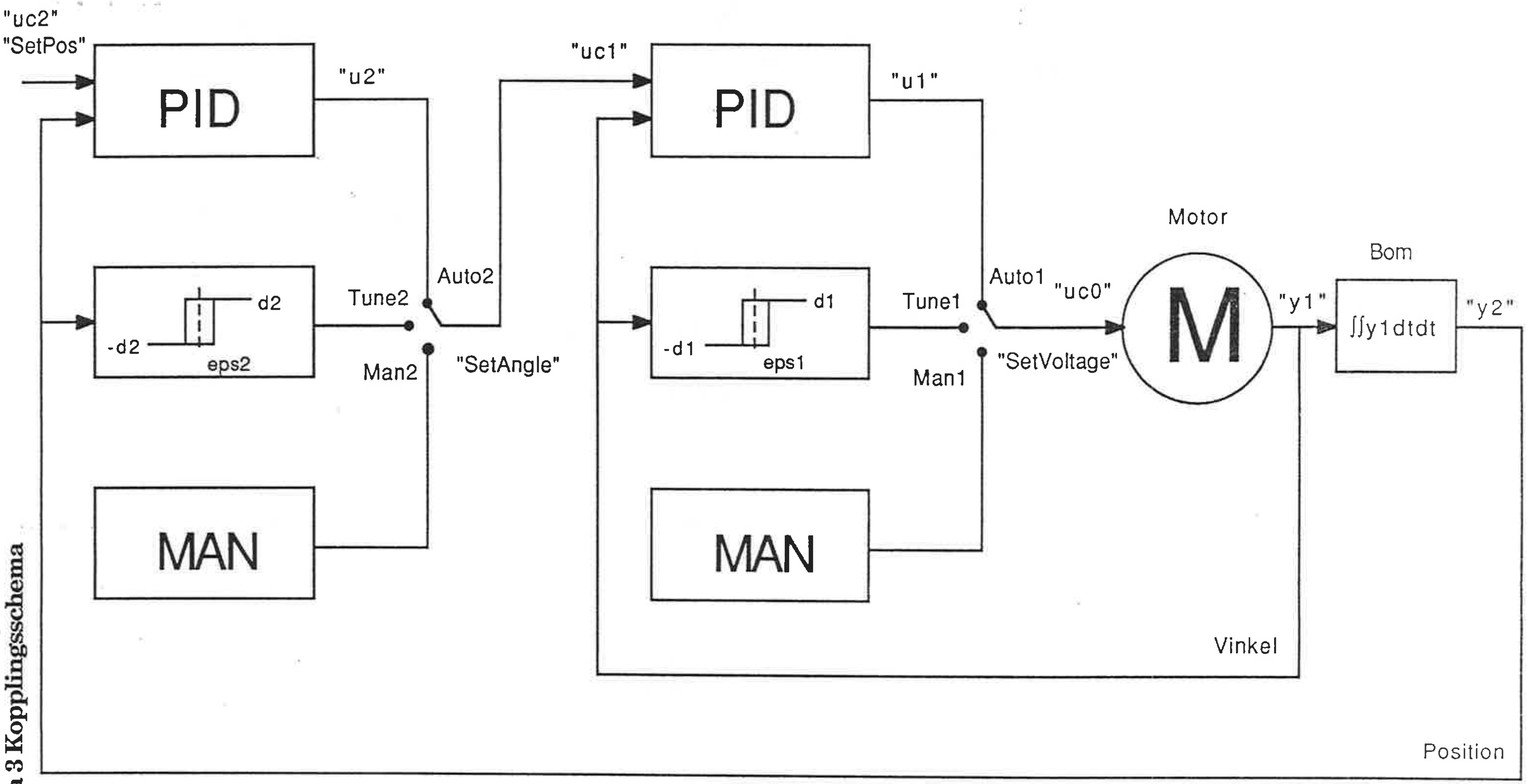
## Bilaga 2 Skärmlayout

Plot-fönster



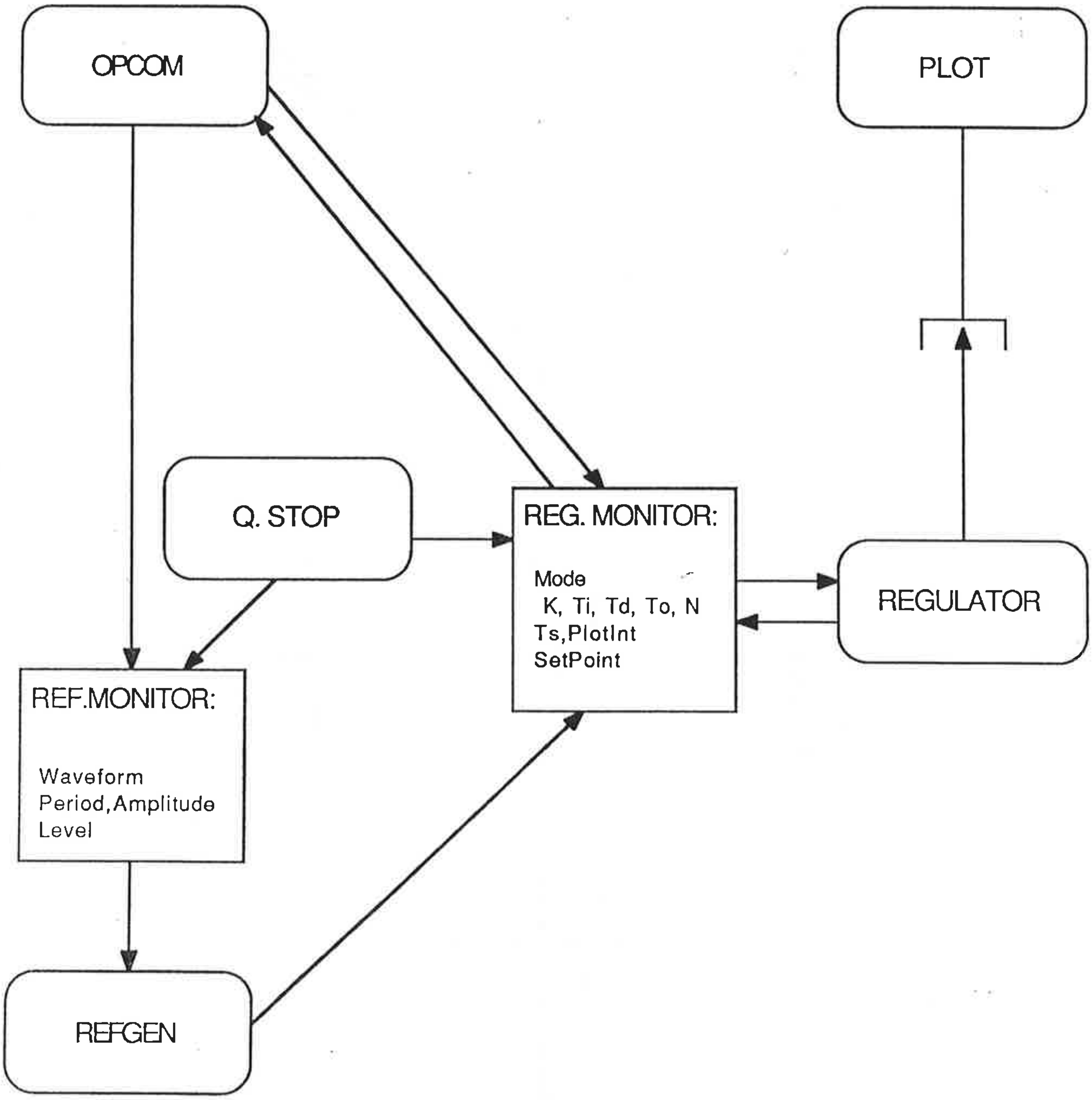

Parameterfönster






**Bilaga 3 Kopplingsschema**

**Bilaga 4 Processgraf**



218.

# PID_EXPERT

An expert system for PID regulator design.

Using NEXPERT OBJECT an expert system shell.

Anders Kjellberg
Gunnar Elmér

Department of Automatic Control
Lund Institute of Technology
May 1989

219.

**Contents**

# PID_EXPERT

An expertsystem for PID regulator design.

Using NEXPERT OBJECT an expertsystem shell.

Anders Kjellberg, Gunnar Elmér


## Abstract.

The expertsystem, PID_EXPERT, has been developed for PID regulator design. The system uses modified Ziegler-Nichols tuning and setpoint weighting, based on either normalised dead time or normalised process gain to characterise the open loop process dynamics in time- or frequency domain resp. The expert also recommends when to use feedforward and deadtime compensation.

221.

## 1. Introduction



Although PID regulators are common and well known, they are often poorly tuned. The derivate action is often not even used because it is difficult to tune. The PID_EXPERT is an expert tool for control- and process engineers to design and tune a PID regulator using a simple step- or frequency response.
The PID_EXPERT classifies a process to be one of four classes, described in section 2.2. According to rules based on process class, specification and environment the system will select an appropriate type of regulator. The regulator parameters are calculated using refined Ziegler-Nichols tuning formula (Hang et. al. 1987). If the process can not be classified PID_EXPERT uses standard Ziegler-Nichols tuning and the user must decide the type of the regulator .
The NEXPERT OBJECT expertsystem shell was used to develop the PID_EXPERT. It is a general tool for expertsystem development in a traditional Macintosh environment.

222.



## 2 Knowledge of the PID_EXPERT

This section describes the knowledge (rules and formulas) of the PID_EXPERT expertsystem.

### 2.1 Process characteristics

The process can be characterised either in time- or frequency domain depending on how process dynamics has been tested (step- or frequency response).

#### 2.1.1 Time domain characterisation

The step response has the general characteristics shown in figure 2.1. The number kp is the static process gain, the number L is the apparent dead time and the number T is the apparent time constant. The parameters T and L are obtained by the graphical construction indicated in figure 2.1 where the tangent is drawn in the inflexion point of the step response.

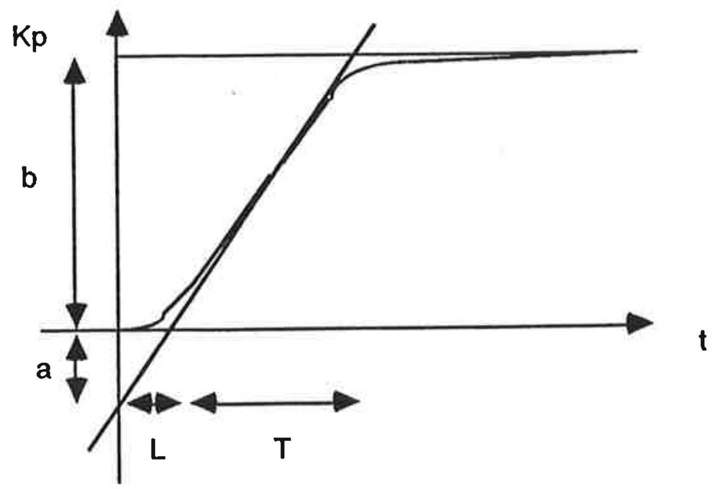



figure 2.1 Step response

### 2.1.2 Frequency domain characterisation



It is assumed that the Nyquist curve has the shape indicated in figure 2.2. The first intersection with the negative real axis defines the ultimate frequency, wu, and the ultimate gain, ku.

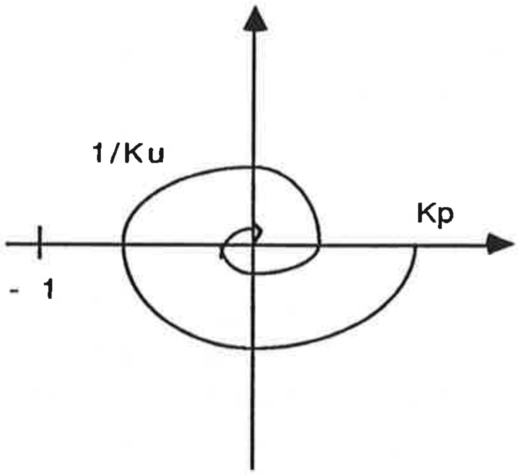



figure 2.2 Nyquist curve for frequency response.

### 2.1.3 Normalised dead time and normalised process gain

Two dimension free parameters are used in PID_EXPERT to select and tune a PID regulator (Åström et. al. 1988).

Normalised dead time, NDT, is used if the process is characterisability in time domain. It is defined as the ratio of the apparent dead time and the apparent time constant.

NDT = L/T = a/Kp = a/b

See figure 2.1.

Normalised process gain, NPG, is used in frequency domain. NPG is defined as the process gain multiplied by the ultimate gain.

NPG = Kp*Ku

See figure 2.2.

## 2.2 Process classification



The process is classified by PID_EXPERT into one of four classes depending on NDT or NPG (Åström et. al. 1988).

**Class I** NDT < 0.15 or NPG > 20 : A proportional regulator could be chosen if a static error around 10% is tolerable. If smaller static error are required it is necessary to use integral action.

**Class II** 0.15 < NDT < 0.6 or 2.0 < NPG < 20 : This is the prime application area for PID controllers with Ziegler-Nichols tuning. Derivative action is often very helpful.

**Class III** 0.6 < NDT < 1.0 or 1.5 < NPG < 2.0 : When NDT approaches 1 Ziegler-Nichols tuning becomes less useful. PID_EXPERT uses modified Ziegler-Nichols tuning with setpoint weighting.

**Class IV** NDT > 1.0 or NPG < 2.0 : PID control based on Ziegler-Nichols tuning is not recommended when NDT is larger than 1.0. PID_EXPERT uses modified Ziegler-Nichols tuning with setpoint weighting.

## 2.3 Regulator selection

PID_EXPERT selects regulator type according to the table 1 in figure 2.3 (Åström et. al. 1988).
If the process can not be classified by PID_EXPERT the user has to select regulator type.

| Class | Tight control is not required | Tight control is required | | |
|---|---|---|---|---|
| | | High measurement noise | Low saturation limit | Low measurement noise and high saturation limit |
| I | P | PI | PID | PI |
| II | PI | PI | PID | PID |
| III | PI | PI + A | PI + A | PID + A + C |
| IV | PI | PI + B + C | PI + B+ C | PI + B + D |

figure 2.3 table 1. A: feedforward compensation recommended, B: feedforward compensation essential, C: dead time compensation recommended, D: dead time compensation essential.

## 2.4 Regulator parameter calculations



It is assumed in the following text that PID_EXPERT has calculated NDT or NPG as described in section 2.2. Otherwise the regulator parameters are calculated with standard Ziegler-Nichols tuning rules.
Heuristics for regulator tuning of processes with monotone step- or frequency response shows that NDT*NPG ≈ 1.3 (Åström 1989, Åström et. al. 1988). The parameter NDT can therefore be used instead of NPG.

For processes of class I or II PID_EXPERT uses standard Ziegler-Nichols tuning and setpoint weighting. Regulator parameters for class III and IV processes are calculated using modified Ziegler-Nichols tuning and setpoint weighting.

### 2.4.1 Setpoint weighting factor

The setpoint weighting factor (b) is calculated with respect to acceptable overshoot (x) and NDT (Hang et. al. 1987).

NDT < 0.3 : b = 2*(x - 0.1) + 5/3*NDT

0.3 < NDT < 0.6 : b = 2*x + NDT

0.6 < NDT < 0.8: b = 1.6 - NDT

NDT > 0.8: b = 0.8

### 2.4.2 Modified Ziegler-Nichols tuning formula

The integral action of the standard tuning formula ($Ti_{Z\text{-}N}$) is modified (Hang et. al. 1987).

$$Ti = \beta * Ti_{Z\text{-}N}$$

NDT < 0.6: ß = 1.0

0.6 < NDT < 1.0 ß = 1.5 - 0.83*NDT

NDT > 1.0 ß = 0.67

## 2.5 Peak load error

The maximum error , e, due to a load disturbance of amplitude 1 is calculated as (Åström et. al. 1988)

e = Kp*lambda,

where lambda*NPG ≈ 1.3.

# 3 NEXPERT OBJECT

NEXPERT OBJECT is a hybrid system that support both reasoning and object-oriented representation.

## 3.1 The reasoning dimension

To represent reasoning , NEXPERT OBJECT uses rules. Rules are symbolic structures which express deductive or evocative progressions in a reasoning path. Deductive means that a rule can be used to verify conditions in another rule. Evocative means that a rule can trigger the activation or the evaluation of other rules.
A rule is a chunk of knowledge representing a situation and its immediate consequences. The format of a rule is a symbolic structure of the type:

if ... then ... and do ...

where *if* is followed by a set of conditions, *then* by a hypothesis or goal which becomes true when the conditions are met, and *do* by a set of actions to be performed as result of a positive evaluation of the rule.

## 3.2 The representation dimension

The representation of the problem is made in terms of class, object and property.
An object is an elementary unit of description. Everything is an object.
A property is a characteristic of an object.
As an example, consider a condition of a NEXPERT rule:

Is object1.property1 "white"

A translation of this syntax is:

Is the value of the property "property1" of the object "object1" "white"?

A class is a collection of objects that usually share properties.
Consider the following condition:

Is <class1>.property1 "white"

which translates into:

Is there any object in the class "class1" whose property "property1" has the value "white"?

The NEXPERT inference engine can approach a problem in three different ways: via hypotheses to be tested, through evidence alone (data), or through a mixed approach of the previous methods.

The hypothesis driven method (backward chaining) focuses on working from a hypothesis and proceeding back to the evidence. During this goal driven process, NEXPERT may find that one, or several other hypotheses are relevant to a problem. Therefore, after exploring the initial hypothesis, NEXPERT will automatically investigate other potential solutions.

The data driven method (forward chaining) focuses on symptoms, changes of values, or other incidental information about a problem. When forward chaining is used, the system begins with evidence and finds the relevant hypotheses or solutions to a problem.

## 4 Implementation of the PID_EXPERT

The PID_EXPERT is implemented in the NEXPERT environment using objects and rules. The regulator design problem is solved with backward chaining.

### 4.1 Objects

Type of properties: (F) = Float, (S) = String, (B) = Boolean.

```
PROCESS
   Subobjects:
      TIME_DOMAIN
        Properties:
           a = (F)
           b = (F)
           L = (F)
           T = (F)
      FREQUENCY_DOMAIN
         Properties:
           kp = (F)
           ku = (F)
           wu = (F)
   Properties:
      MEASUREMENT_NOISE = (S)
      NORM_DEAD_TIME = (F)
      NORM_PROCESS_GAIN = (F)
      SATURATION_LIMIT = (S)

REGULATOR
   Subobjects:
      PID_PARAMETERS
         Properties:
           b = (F)
           K = (F)
           Ti = (F)
           Td = (F)
   Properties:
      TYPE = (S)

CLOSEDLOOP_SPEC
   Properties:
      OVER_SHOOT = (F)
      PEEK_LOAD_ERROR = (F)
      RISE_TIME = (F)
      TIGHT_CONTROL_REQUIRED = (B)
```

CLOSEDLOOP_PERFORMANCE
    Properties:
        OVER_SHOOT = (F)
        PEEK_LOAD_ERROR = (F)
        RISE_TIME = (F)

### 4.2 Rules

There are about fifty rules divided into four types: domain, class, regulator and parameter rules.
Domain rules concerns process characterisation, see section 2.1.
Class rules classifies the process, see section 2.2.
Regulator rules decides what type of regulator to use, see section 2.3.
Parameter rules calculates the regulator parameters, see section 2.4.

230.

# 5 How to run the PID_EXPERT



1 Start the NEXPERT OBJECT.
2 Select Load Knowledge Base from the Expert menu.
3 Load the knowledge base entitled PID_EXPERT.kb.
4 Select the function  Knowcess from the Expert menu.
5 When the Session Control window displays the message: End of Session select Case Status from the Report menu to display the result of the session.

231

Åström, K.J. , C.C. Hang, and P.Persson (1988): "Heuristics for Assessment of PID control with Ziegler-Nichols Tuning," Internal report CODEN: LUTFD2/TFRT-7404, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Hang, C.C. and K.J. Åström (1988): "Refinements of the Ziegler-Nichols Tuning Formula for PID Auto-tuners," Proc. ISA Annual Conf. Houston, USA.

Åström, K.J. (1989): "Assessment of Achievable Performance of Simple Feedback Loops,"Internal report CODEN: LUTFD2/(TFRT-7411)/1-18/(1989), Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.

Åström, K.J. and T. Hägglund (1988): "Automatic Tuning of PID Regulators," ISA, Research Triangle Park, NC, USA.

232.

## 7. Undersökning av design utmaning från American Control

233.

# Special 4

## Projekt i adaptiv reglering.

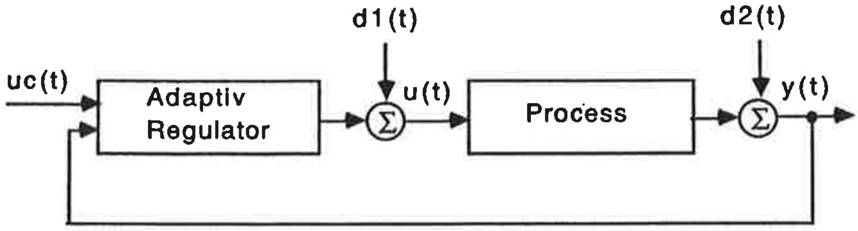
Jerker Persson
Magnus Hansson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Maj 1988

234

# Innehållsförteckning

# 1. Förutsättningar




**Figur 1.1** Blockschema för processen och den adaptiva regulatorn.

Förutsättningarna för projektet var att med en adaptiv regulator reglera en process , där processparametrarna varierade, så att den uppförde sig som en förutbestämd modell (fig 1.1) . Processen beskrivs av överföringsfunktionen

$$G(s) = \frac{b}{s^2 + a_1 s + a_2 s}$$

där parametrarna varierar mellan gränserna:

$$0.5 \leq b \leq 3.0$$
$$-0.6 \leq a_1 \leq 3.4$$
$$-2.0 \leq a_2 \leq 4.0$$

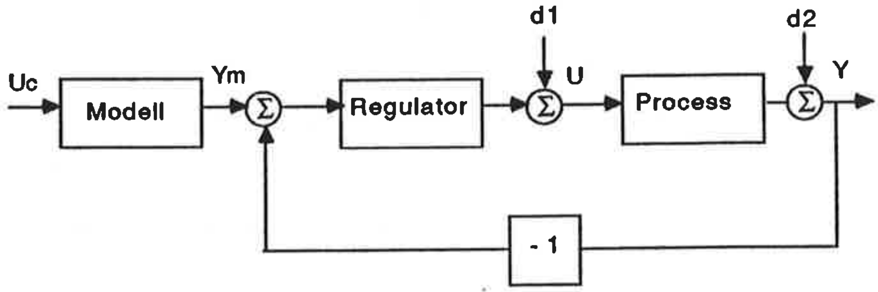
Modellen som skall följas beskrivs av överföringsfunktionen

$$G_m(s) = \frac{1}{s^2 + 1.4s + 1}$$

Styrsignalen till processen $u(t)$ störs av laststörningen $d_1(t)$, dessutom störs utsignalen från processen $y(t)$ av mätbruset $d_2(t)$. De olika regulatorer som simuleras i detta projekt är: återkoppling med hög förstärkning, modell referenssystem enligt MIT, modell referenssystem enligt Lyapunov, indirekt självinställare samt direkt självinställare.

# 2. High-Gain reglering

Den High-Gain regulator vi använt består av två delar.En modell $G_m(s)$, som beskriver hur vi önskar att den reglerade processen skall uppföra sig, och själva regulatorn $G_r(s)$. I fig. 2.1 visas ett blockschema för process och regulator.




**Figur 2.1** Blockschema för processen och High-Gain regulatorn.

Regulatorn kan väljas på många sätt men karakteriseras av en hög förstärkning.

Med den tidigare specificerade processen $G_p(s)$ fås utsignalen:

$$Y(s) = G_m \cdot \frac{G_r G_p}{1 + G_r G_p} U_c(s)$$

där

$$G_p(s) = \frac{b}{s^2 + a_1 s + a_2}$$
$$G_m(s) = \frac{b_m}{s^2 + a_{m1} s + a_{m2}}$$

Den enklaste varianten av $G_r(s)$ är en proportionell reg. $G_r(s) = k$. Detta ger utsignalen:

$$Y(s) = G_m \cdot \frac{k \cdot b}{s^2 + a_1 s + a_2 + k \cdot b} \cdot U_c(s)$$

För stora $k$ och måttliga frekvenser kan detta approximeras med:

$$Y(s) = G_m(s) \cdot U_c(s)$$

Dvs precis vad som önskades.Tyvär fungerar denna enkla regulator inte då $a_1 < 0$ vilket gör att $G_p(s)$ blir instabil.

Försöker man istället med en PI-regulator:

$$G_r(s) = k \left(1 + \frac{1}{T_i s}\right)$$

337.

fås förutom de stabila polerna hos $G_m(s)$ polpolynomet

$$A(s) = s^3 + a_1 \cdot s^2 + (a_2 + bk) \cdot s + \frac{bk}{T_i}$$

För stabilitet krävs att:

$$a_1 > 0$$
$$\frac{bk}{T_i} > 0$$
$$a_1 \cdot (a_2 + bk) > \frac{bk}{T_i}$$

Även i detta fall kan alltså systemet bli instabilt ex. då $a_1 < 0$. Nästa steg blir att prova en PID-regulator.

$$G_r(s) = k\left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + \frac{T_d s}{N}}\right)$$

Vi får nu förutom de stabila polerna hos $G_m(s)$ även polpolynomet

$$A(s) = s^4 + z_1 \cdot s^3 + z_2 \cdot s^2 + z_3 \cdot s + z_4$$
$$z_1 = \left(\frac{N}{T_d} + a_1\right)$$
$$z_2 = \left(kb(1 + N) + \frac{a_1 N}{T_d + a_2}\right)$$
$$z_3 = \left(kb\left(\frac{N}{T_d} + \frac{1}{T_i}\right) + \frac{a_2 N}{T_d}\right)$$
$$z_4 = \frac{kbN}{T_i T_d}$$

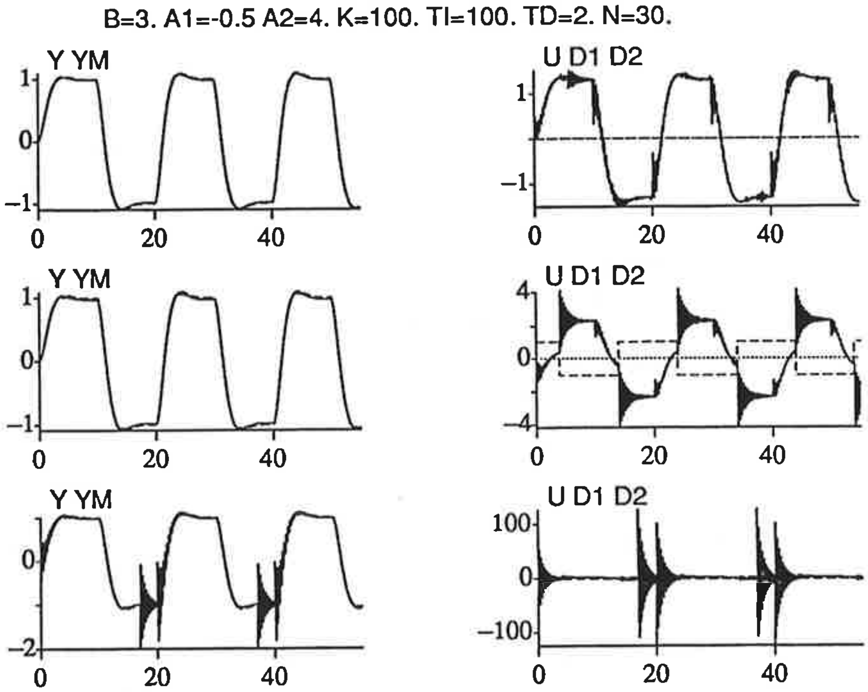
För stabilitet krävs nu att följande är uppfyllt:

$$z_1 > 0$$
$$z_4 > 0$$
$$z_1 \cdot z_2 > z_0 \cdot z_3$$
$$z_3 \cdot (z_1 z_2 - z_0 z_3) > z_1^2 z_4$$

Med lämpligt valda PID-parametrar går dessa krav att uppfylla. Regulatorn kommer alltså att kunna fungera för den givna processen. Ett minus är att om processens parametrar ändras mycket, kan nya inställningar av regulatorn behövas. Har man väl gjort ett antal olika inställningar och dessutom har tillgång till någon mätsignal med vilken man kan påvisa ändringarna i processen, kan sk. Gain-Scheduling användas.

Tittar man lite på systemets störkänslighet visar det sig att laststörningar dämpas mycket bra. Mätbrus däremot resulterar i orimligt stora styrsignaler som en följd av regulatorns höga förstärkning. Fig 2.2 visar tre olika simuleringar med processen

$$G_p(s) = \frac{3}{s^2 - 0.5s + 4} \cdot U(s)$$

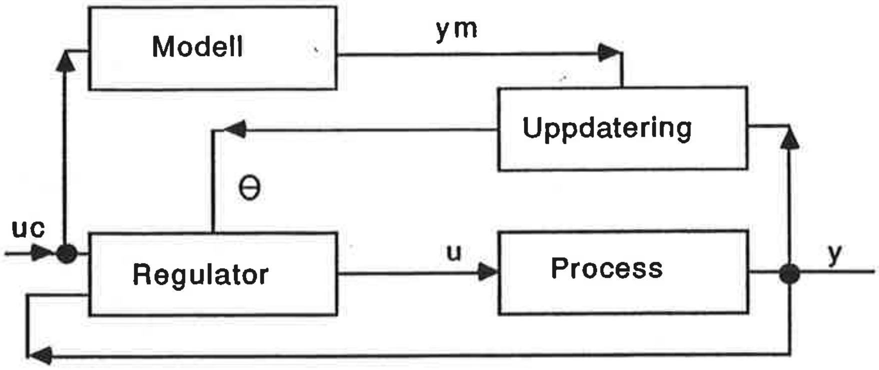



**Figur 2.2** Simuleringar med High-Gain reglering.Överst utan störningar, i mitten med laststörning och nederst med både last- och mätstörningar.

I den första simuleringen finns ingen störning pålagd. I den andra har en laststörning med amplitud 1.0 adderats till styrsignalen. I den tredje har dessutom en mätstörning med amplitud 0.5 adderats till utsignalen.

Sammanfattningsvis kan om High-Gain regulatorn sägas att den fungerar utmärkt om man har en okänd process, där processparametrarna inte ändras så mycket och ofta att det blir opraktiskt och påfrestande att hålla på och ställa in regulatorn. Vidare krävs att eventuellt mätbrus är mycket litet.

# 3. MRAS enligt MIT




**Figur 3.1** Blockschema för ett modellreferenssystem med uppdatering enligt MIT-regeln.

Modell referens systemet är uppbyggd av två loopar en inre loop, återkoppling från processen till regulatorn, som varierar fort samt en yttre loop, uppdatering av parametrarna, som varierar långsamt (fig 3.1). Det önskade uppförandet av processen specificeras i form av en modell. Regulatorparametrarnas uppdatering baseras på skillnaden mellan utsignalen från processen och utsignalen från modellen. Det är önskvart att justera parametrarna i regulatorn så att detta fel går mot noll. Med MIT regeln justeras parametrarna enligt följande. Metoden använder förlustfunktionen

$$J(\theta) = \frac{1}{2}e^2$$

där $\theta$ är parametrarna och e är felet mellan process och modell. För att minimera förlustfunktionen justeras parametrarna i en riktning som bestäms av förlustfunktionens negativa gradient.

$$\frac{d\theta}{dt} = -\gamma \frac{dJ}{d\theta} = -\gamma e \frac{\partial e}{\partial \theta}$$

Eftersom parametrarna varierar mycket långsammare än variablerna i systemet kan $\frac{\partial e}{\partial \theta}$ beräknas under antagande av att $\theta$ är konstant. Felet mellan process och modell kan skrivas

$$e = y - y_m = \left( \frac{BT}{AR + BS} - \frac{B_m}{A_m} \right) u_c$$

Känslighetsderivatorna kan då skrivas

$$\frac{\partial e}{\partial r_1} = -\frac{ABT}{(AR+BS)^2}u_c$$
$$\frac{\partial e}{\partial s_0} = -\frac{pBTB}{(AR+BS)^2}u_c$$
$$\frac{\partial e}{\partial s_1} = -\frac{BTB}{(AR+BS)^2}u_c$$
$$\frac{\partial e}{\partial t_0} = -\frac{pB}{AR+BS}u_c$$
$$\frac{\partial e}{\partial t_1} = -\frac{B}{AR+BS}u_c$$

där p=$\frac{d}{dt}$. Eftersom koefficienterna i A och B inte är kända approximeras AR+BS med $A_0A_mB^+$, approximationen kommer att vara exakt när regulator parametrarna har konvergerat. Uppdatering av parametrarna sker enligt följande. Parametern b ingår i $\gamma$.

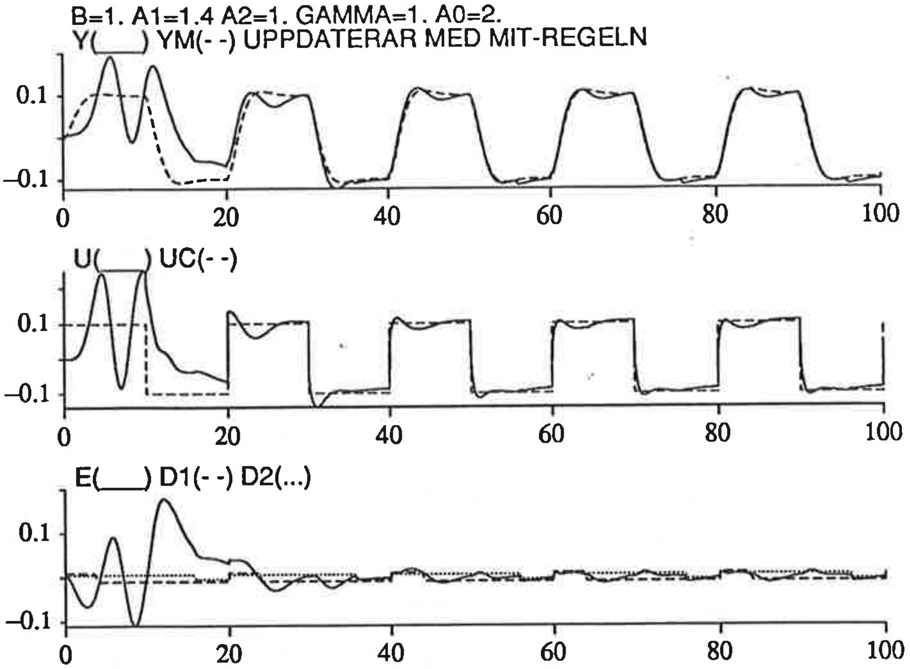
$$\frac{dr_1}{dt} = -\gamma e\frac{\partial e}{\partial r_1} \quad = \gamma e\frac{u}{A_0A_m}$$
$$\frac{ds_0}{dt} = -\gamma e\frac{\partial e}{\partial s_0} \quad = \gamma e\frac{y}{A_0A_m}$$
$$\frac{ds_1}{dt} = -\gamma e\frac{\partial e}{\partial s_1} \quad = \gamma e\frac{y}{A_0A_m}$$
$$\frac{dt_0}{dt} = -\gamma e\frac{\partial e}{\partial t_0} \quad = \gamma e\frac{u_c}{A_0A_m}$$
$$\frac{dt_1}{dt} = -\gamma e\frac{\partial e}{\partial t_1} \quad = \gamma e\frac{u_c}{A_0A_m}$$

För att göra uppdateringen av regulatorparametrarna oberoende av amplituden på insignalen används följande modifierade uppdateringsregel.

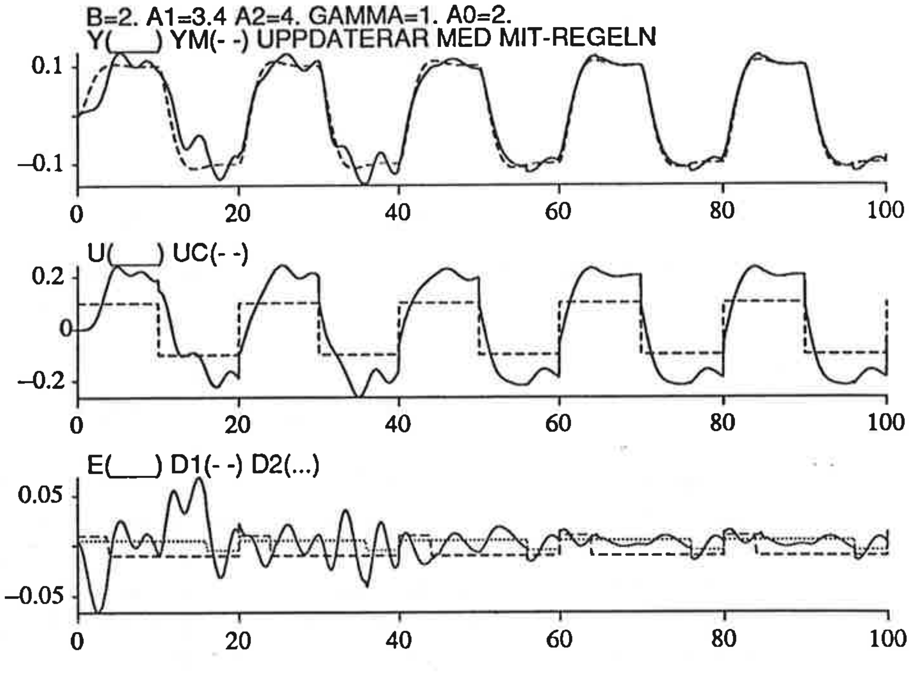
$$\frac{d\theta}{dt} = -\gamma\frac{e\frac{\partial e}{\partial \theta}}{\alpha + \left(\frac{\partial e}{\partial \theta}\right)^T\left(\frac{\partial e}{\partial \theta}\right)}$$

En listning av simnon-koden för regulatorn finns i appendix. Simuleringarna visar att systemet klarar av att reglera utsignalen från processen till önskat värde för alla olika parametervariationer hos processen, se figur 3.2 till figur 3.4 . De enda parametervärden som ger regulatorn problem är när processen har bägge sina polerna i högra halvplanet, se figur 3.5 och figur 3.6. Regulatorn klarar av att lösa även detta problemet efter en viss tid. Variationerna i $\gamma$ visar att ett större värde ger snabbare konvergens men får också till följd att regulatorn styr ut processen mycket kraftigare, se figur 3.2 och figur 3.7.

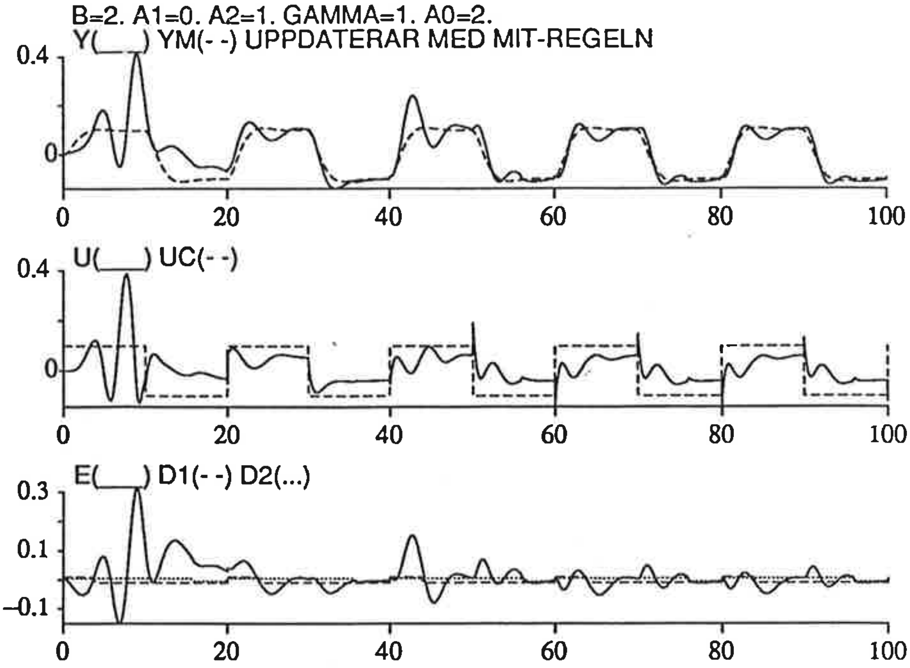
241.



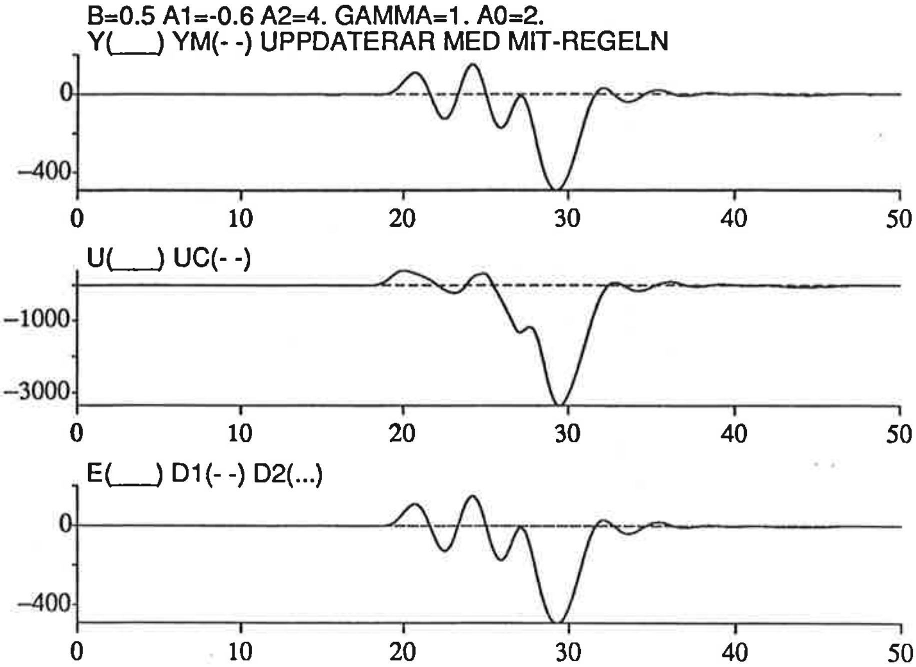

**Figur 3.2** MIT-regulator processen är lika med modellen



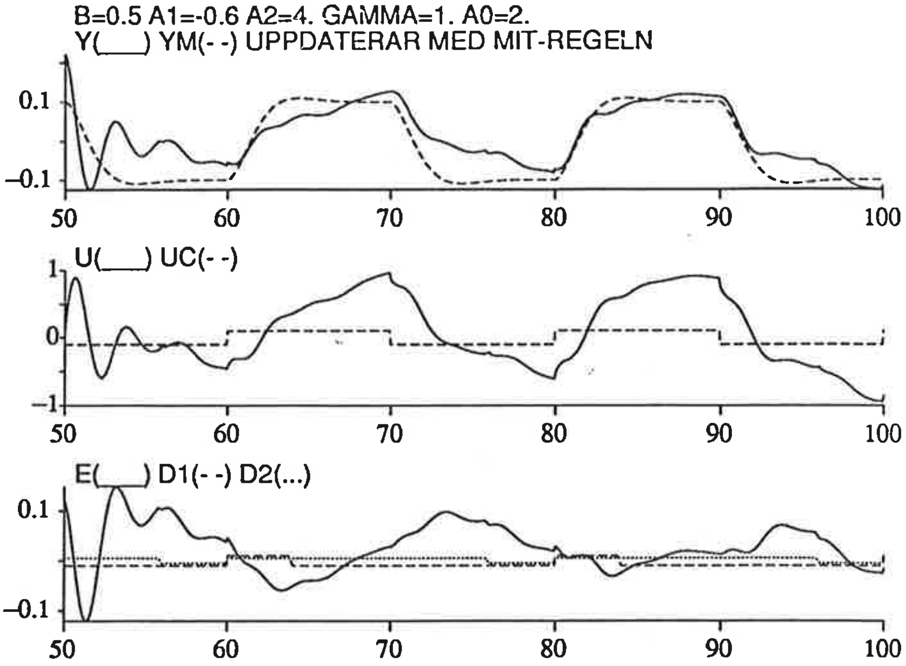

**Figur 3.3** MIT-regulator polerna i VHP



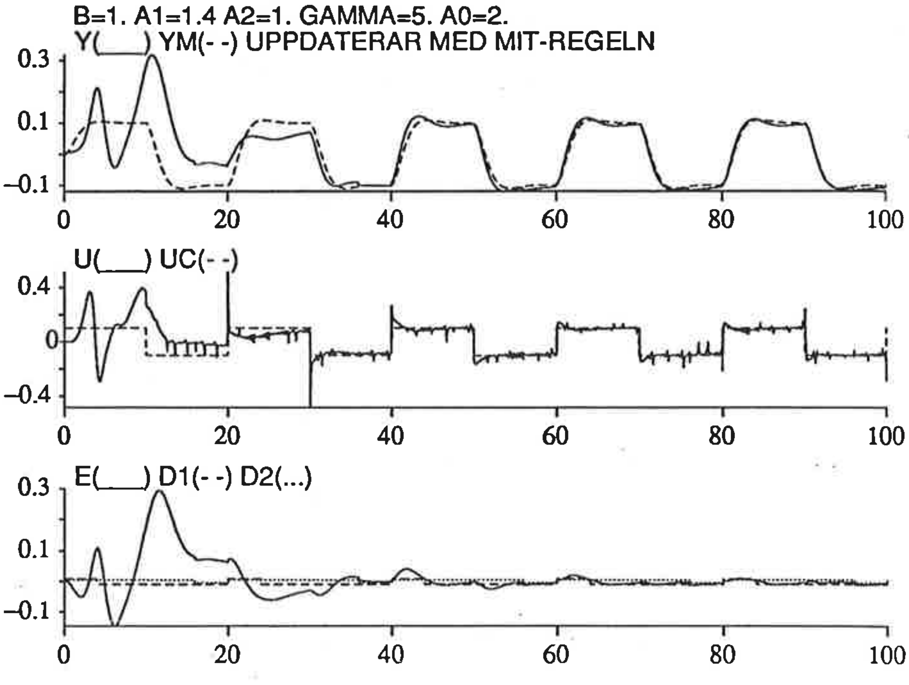

**Figur 3.4** MIT-regulator polerna på imaginära axeln




**Figur 3.5** MIT-regulator polerna i HHP




**Figur 3.6** MIT-regulator polerna i HHP

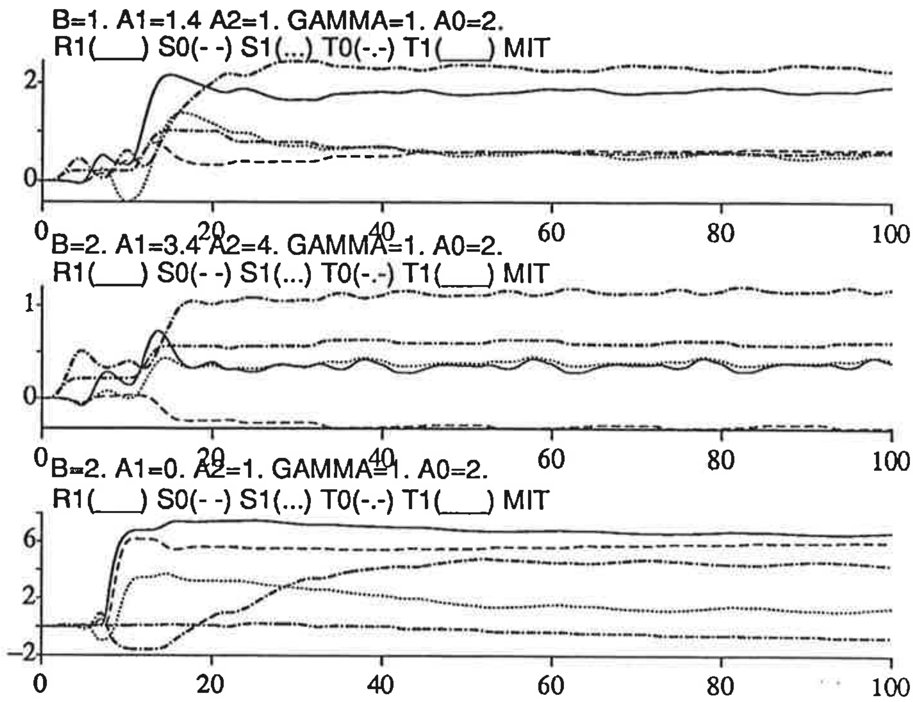



**Figur 3.7** MIT-regulator gammas påverkan på systemet

Med polynomsyntes kan de rätta värdena på regulatorparametrarna beräknas.

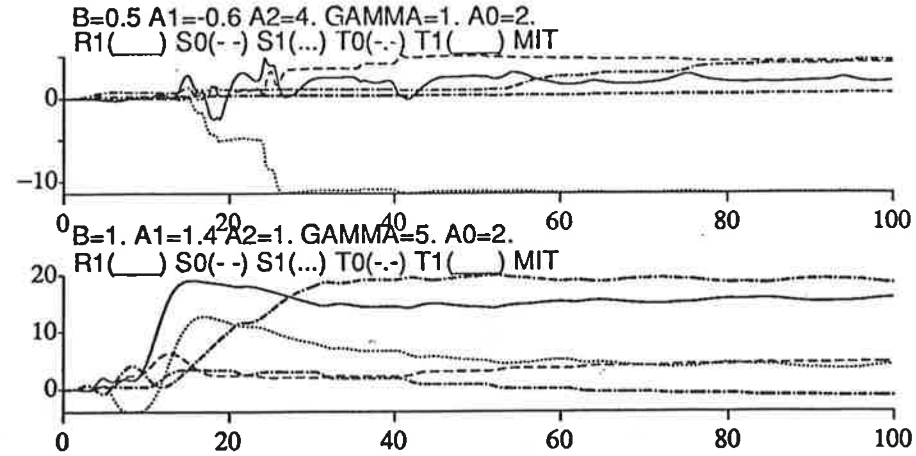
$$
\begin{aligned}
r_1 &= a_0 + a_{m1} - a_1 \\
s_0 &= \frac{1}{b}[(a_{m2} - a_2) - a_1(a_{m1} - a_1) - a_0(a_{m1} - a_1)] \\
s_1 &= \frac{1}{b}[a_0(a_{m2} - a_2) - a_2(a_{m1} - a_1)] \\
t_0 &= \frac{b_m}{b} \\
t_1 &= \frac{b_m a_0}{b}
\end{aligned}
$$

En direkt jämförelse med dessa formler visar att parametrarna i simuleringen antar värden som är skilda från de exakta värdena, beräknade med formlerna, se figur 3.8 och figur 3.9. Att detta händer, trots att felet går mot noll, är karakteristiskt för MIT-systemet.




**Figur 3.8** Parametranas uppförande från figur 1 till figur 3

2 45.

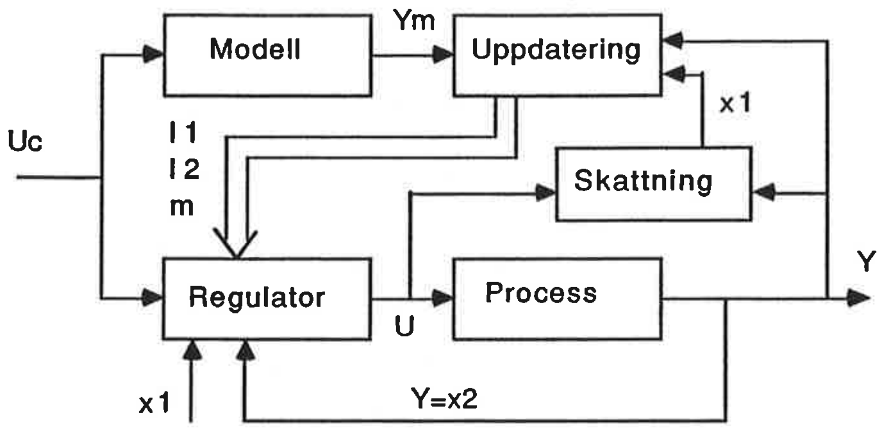



**Figur 3.9** **Parametranas uppförande från figur 4 till figur 6**

3 46.

# 4. MRAS enligt Lyapunov

Denna regulator baseras på en tillståndsåterkoppling, där återkopplingsparametrarna uppdateras enl. Lyapunov. Ett blockschema över process och regulator visas i fig 4.1.




**Figur 4.1** Blockschema för ett modellreferenssystem med uppdatering enl. Lyapunov där endast utsignalen är mätbar.

Vår process kan på tillståndsform skrivas som:

$$\begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$

Styrbar kanonisk form ger $A$, $B$ och $C$ matriserna följande utseende:

$$A = \begin{pmatrix} -a_1 & -a_2 \\ 1 & 0 \end{pmatrix} \quad B = \begin{pmatrix} b \\ 0 \end{pmatrix}$$
$$C = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

Med styrlagen:

$$U = -Lx + mu_c$$

fås

$$\begin{cases} \dot{x} = A_s x + B_s u_c \\ y = Cx \end{cases}$$

där

$$A_s = A - BL = \begin{pmatrix} -a_1 - l_1 & -a_2 - l_2 \\ 1 & 0 \end{pmatrix}$$

$$B_s = B \cdot m = \begin{pmatrix} bm \\ 0 \end{pmatrix}$$

Systemets önskade egenskaper beskrivs av modellens tillståndsmatriser:

$$A_m = \begin{pmatrix} -a_{m1} & -a_{m2} \\ 1 & 0 \end{pmatrix} \quad B_m = \begin{pmatrix} b_m \\ 0 \end{pmatrix}$$

$$C_m = C = \begin{pmatrix} 0 & 1 \end{pmatrix}$$

Felen i $A_s$, $B_s$ och deras derivater blir då:

$$\widetilde{A} = A_s - A_m = \begin{pmatrix} -a_1 - l_1 - a_{m1} & -a_2 - l_2 - a_{m2} \\ 0 & 0 \end{pmatrix}$$

$$\dot{\widetilde{A}} = \dot{A}_s = \begin{pmatrix} -\dot{l}_1 & -\dot{l}_2 \\ 0 & 0 \end{pmatrix}$$

$$\widetilde{B} = B_s - B_m = \begin{pmatrix} bm - b_m \\ 0 \end{pmatrix}$$

$$\dot{\widetilde{B}} = \dot{B}_s = \begin{pmatrix} b\dot{m} \\ 0 \end{pmatrix}$$

Felet i tillstånden och dess derivata definieras som:

$$\begin{aligned} e &= x - x_m \\ \dot{e} &= \dot{x} - \dot{x}_m \\ &= A_s x + B_s u_c - A_m x_m - B_m u_c \\ &= A_m e + \widetilde{A}x + \widetilde{B}u_c \end{aligned}$$

Eftersom vi var så lyckligt lottade att vi räknat på ett liknande exempel under kursen, uppg. 4.4 närmare bestämt, har vi utan extra besvär en färdig Lyapunov-funktion. Framtagandet av en sådan torde annars kunna vålla vissa bryderier. Lyapunov-funktionen $V$ ges av:

$$V = tr\ e^T P e + tr\ \widetilde{A}^T Q_A \widetilde{A} + tr\ \widetilde{B}^T Q_B \widetilde{B}$$

Där $P$, $Q_A$ och $Q_B$ är pos. def. matriser.

$$\begin{aligned} \frac{dV}{dt} =& tr\left(\dot{e}^T P e + e^T P \dot{e}\right) + tr\left(\dot{\widetilde{A}}^T Q_A \widetilde{A} + \widetilde{A}^T Q_A \dot{\widetilde{A}}\right) \\ &+ tr\left(\dot{\widetilde{B}}^T Q_B \widetilde{B} + \widetilde{B}^T Q_B \dot{\widetilde{B}}\right) \end{aligned}$$

248

Med lite tålamod och en våldsam rotation på matriserna fås:

$$\frac{dV}{dt} = tr\ e^T \left(PA_m + A_m^T P\right) e + tr\ 2\tilde{A}^T \left(Q_A \dot{\tilde{A}} + Pex^T\right)$$
$$+ tr\ \tilde{B}^T \left(Peu_c^T + Q_B \dot{\tilde{B}}\right)$$

Om nu följande är uppfyllt:

$$A_m^T P + PA_m = -H$$

Där $-H$ är pos. def. och

$$tr\ 2\tilde{A}^T \left(Q_A \dot{\tilde{A}} + Pex^T\right) = 0$$
$$tr\ 2\tilde{B}^T \left(Peu_c^T + Q_B \dot{\tilde{B}}\right) = 0$$

så blir

$$\frac{dV}{dt} = -tr\ e^T He$$

Välj $Q_A = Q_B = I$. Sedan innan vet vi att $\dot{\tilde{A}} = \dot{A}_s$ och att $\dot{\tilde{B}} = \dot{B}_s$. Det räcker då att följande gäller:

$$tr\ \tilde{A}^T \left(\dot{A}_s + Pex^T\right) = 0$$
$$tr\ \tilde{B}^T \left(\dot{B}_s + Peu_c^T\right) = 0$$

Skriv $\tilde{A}$, $P$ och $e$ som:

$$\tilde{A} = \begin{pmatrix} \tilde{a}_1 & \tilde{a}_2 \\ 0 & 0 \end{pmatrix}$$

$$P = \begin{pmatrix} p_1 & p_2 \\ p_2 & p_3 \end{pmatrix}$$

$$e = \begin{pmatrix} e_1 \\ e_2 \end{pmatrix}$$

Man får då följande två ekvationer som skall uppfyllas

$$\tilde{a}_1 \left((p_1 e_1 + p_2 e_2)x_1 - \dot{l}_1\right) + \tilde{a}_2 \left((p_1 e_1 + p_2 e_2)x_2 - \dot{l}_2\right) = 0$$

$$\tilde{b}\left((p_1 e_1 + p_2 e_2)u_c + b\dot{m}\right) = 0$$

Ur dessa ekvationer kan sedan uppdateringslagen för $l_1$, $l_2$ och $m$ fås som:

$$\begin{cases} \dot{l}_1 = (p_1 e_1 + p_2 e_2)x_1 \\ \dot{l}_2 = (p_1 e_1 + p_2 e_2)x_2 \\ \dot{m} = -(p_1 e_1 + p_2 e_2)u_c \end{cases}$$

Eftersom tecknet på $b$ är känt kan, $b$ "bakas in" i $P$.

Då det endast är möjligt att mäta utsignalen $y = x_2$ i detta problem, måste $x_1 = \dot{x}_2 = \dot{y}$ på något sätt bestämmas ur $u(t)$ och $y(t)$. Detta kan göras t.ex. med en Luenberger observerare där $x1$ rekonstrueras enligt.

$$\begin{cases} \hat{x}_1 = \dfrac{1}{t_1}\hat{z} - \dfrac{t_2}{t_1}y \\ \dfrac{d\hat{z}}{dt} = -\alpha_1\hat{z} + t_1 bu + ky \end{cases}$$

Där $\alpha_1$ och $k$ är positiva konstanter. $T = (t_1\ t_2)$ löses ur ekvationen:

$$TA + \alpha_1 T = kC$$

$$\begin{cases} t_1 = \dfrac{k}{\alpha_1(a_1 - \alpha_1) - a_2} \\ \\ t_2 = \dfrac{k(a_1 - \alpha_1)}{\alpha_1(a_1 - \alpha_1) - a_2} \end{cases}$$

Men då $A$ och $B$ inte är kända kommer felet i rekonstruktionen inte att konvergera på vanligt sätt. Man får följande uttryck:

$$z = Tx$$
$$\hat{z} = T\hat{x}$$
$$\tilde{z} = z - \hat{z}$$

$$\frac{d\tilde{z}}{dt} = -\alpha_1\tilde{z} + T\widetilde{B}u + T\widetilde{A}x$$

$$\widetilde{B} = B - B_m \; \widetilde{A} = A - Am$$

Detta innebär att då $\widetilde{A}$ och $\widetilde{B}$ är skilda från noll kommer felet i skattningen $\tilde{z}$ att exciteras av både $u$ och $x$. En simulering med processen:

$$G_p(s) = \frac{2}{s^2 + 1.4s + 1}$$

visas i fig. 4.2 . Man ser att $\hat{x}_1$ (XS1) inte konvergerar mot $x_1$ vilket också syns på parametern $\hat{z}$ (EZ),detta leder i sin tur till att regulatorparametrarna $l_1$, $l_2$ och $m$ inte heller konvergerar. Problemet är att vi med Luenberger observeraren måste basera rekonstruktionen på okända parametrar i $A$ och $B$. En annan metod där man undviker detta är att återskapa $x_1$ genom en lågpassfiltrerad derivering av $y$.

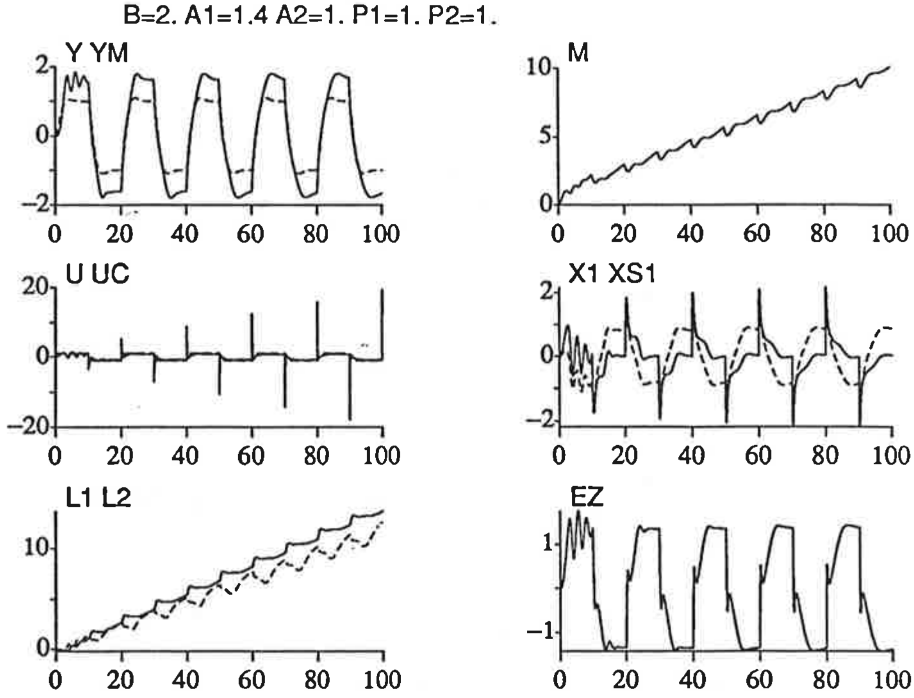
$$\widehat{X}_1(s) = \frac{s}{1 + s/\omega_f} \cdot Y(s)$$

Fig. 4.3 visar en simulering för denna rekonstruktion av $x_1$ med processen

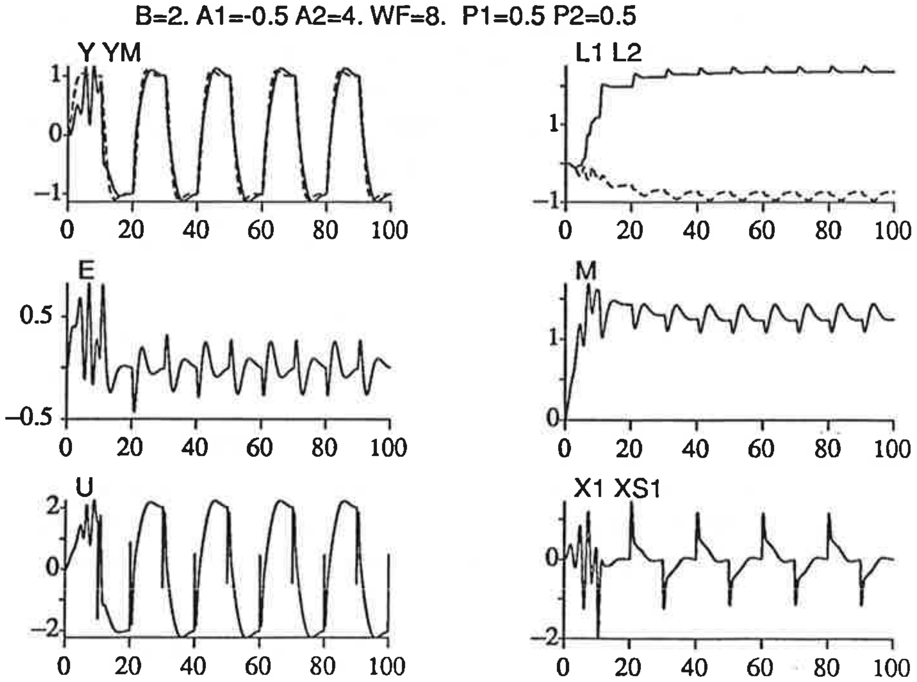
$$G_p(s) = \frac{2}{s^2 - 0.5s + 1}$$

Tittar man endast på simuleringen $t = 0 - 100s$ ser allt bra ut. I fig. 4.4 visas återkopplingsparametrarna $l_1, l_2$ och $m$ från en längre simulering $t = 0 - 500s$.

250.

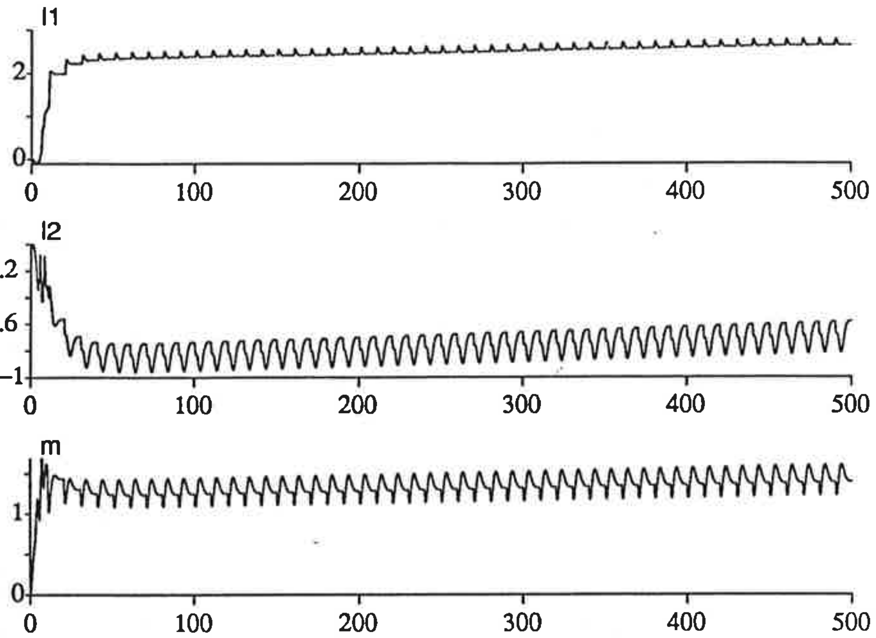



**Figur 4.2** Simulering med ett modellreferenssystem med uppdatering enl. Lyapunov. $x_1$ rekonstrueras med en Luenberger observerare.




**Figur 4.3** Simulering med ett modellreferenssystem med uppdatering enl. Lyapunov. $x_1$ rekonstrueras med en lågpassfiltrerad derivering av y.

Här visar det sig att parametrarna långsamt fortsätter att driva även om god överensstämmelse uppnåtts mellan $y$ och $y_m$. Detta skulle kunna åtgärdas genom att införa en sk. "Dead-Zone" på uppdateringen av regulatorparametrarna. I och med att regulatorn är realiserad på kontinuerlig form skulle man kunna använda derivatan istället för differensen mellan två på varandra föl-




**Figur 4.4** Simulering med ett modellreferenssystem med uppdatering enl. Lyapunov. $x_1$ rekonstrueras med en lågpassfiltrerad derivering av y.

jande värden som i det diskreta fallet. $e_x = (p_1 e_1 + p_2 e_2)$ är proportionell mot derivatan för parametrarna och samtidigt ger den en uppfattning om hur stort felet är i de bägge tillstånden. Använder man denna signal för att bestämma om uppdatering skall ske eller ej, upptäcker man snart att den är full av spikar som gör att värdet tillfälligt överstiger död-zonen, vilket i sin tur leder till att man får "oönskade uppdateringar". Vi provade då att filtrera $e_x$ med ett tredje ordningens lågpassfilter $G_f(s)$.

$$G_f(s) = \frac{\omega_e^3}{(s + \omega_e)^3}$$

Algoritmen för uppdateringen av t.ex $l_1$ kan uttryckas som:

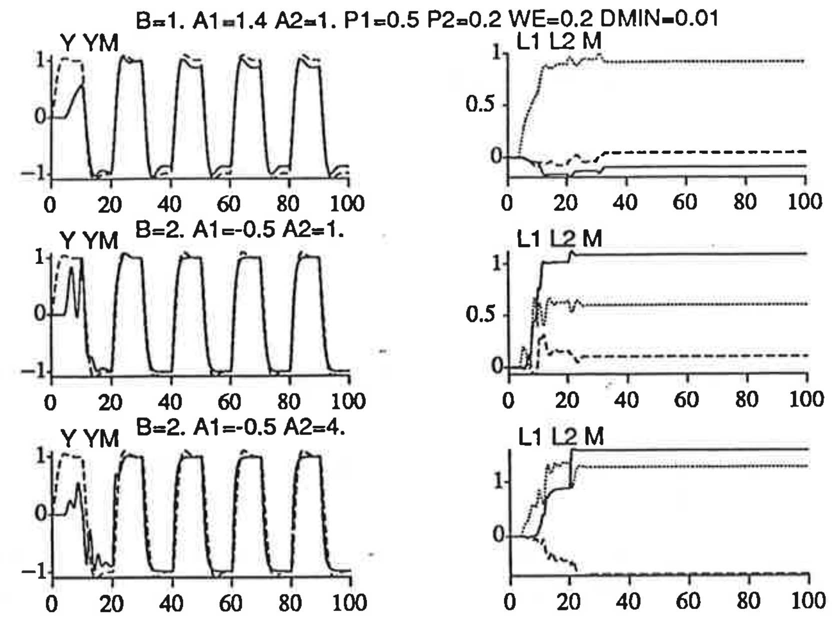
$$\frac{dl_1}{dt} = \begin{cases} e_x \cdot x_1 & \text{om } \mid G_f(s) \cdot e_x \mid > d_{min} \\ 0 & \text{annars} \end{cases}$$

Simnon koden för regulatorn finns listad i appendix . Fig 4.5 visar 3 simuleringar med olika processer. Regulatorn fungerar ungefär lika bra i alla tre fallen. I fig 4.6 är regulatorn simulerad med processen:

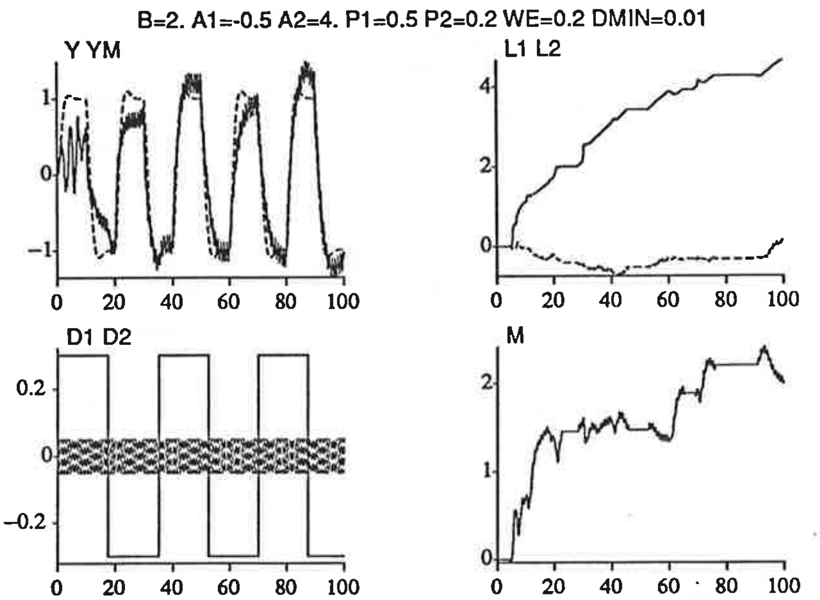
$$G_p(s) = \frac{2}{s^2 - 0.5s + 4}$$

Dvs samma som för den tredje simuleringen i fig 4.5. Här är dessutom en laststörning med amplitud 0.3 samt mätbrus med amplituden 0.05 pålagt. Den fungerar alltså inte alldeles felfritt och man ser att en justering av $d_{min}$ hade behövts för att få parametrarna att stabilisera sig. idealet hade varit en "intelligent övervakare" som slog ifrån uppdateringen då felet i utsignalen blivit tillräckligt litet.

Sammanfattningsvis kan om denna typ av regulator sägas att man spar mycken möda om alla tillstånden är mätbara, då den fungerar alldeles utmärkt.




**Figur 4.5** Simulering med ett modellreferenssystem med uppdatering enl. Lyapunov. På uppdateringen finns en "död-zon" $d_{min}$ .



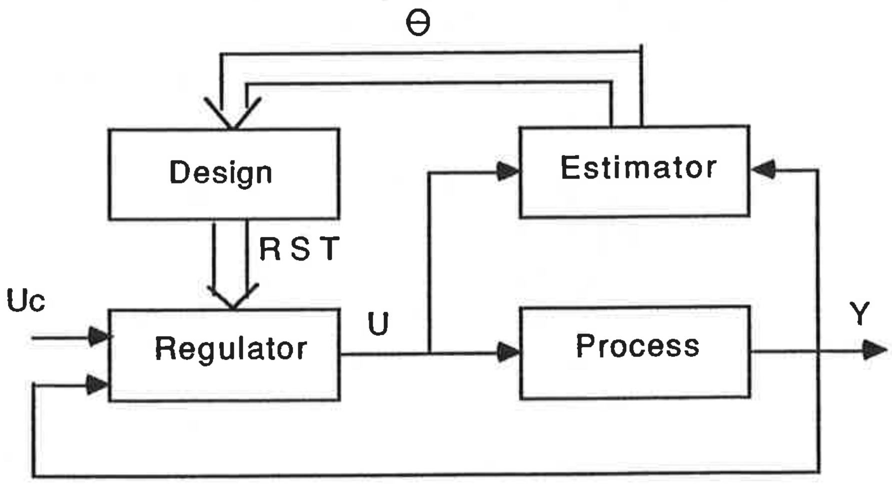

**Figur 4.6** Samma simulering som nederst i fig 4.5 ovan men med en laststörning $d_1$ och mätbrus $d_2$.

# 5. Indirekt självinställare

Denna självinställare består av tre delar:

1. En minstakvadrat skattare som skattar en modell av processen.
2. En design del där regulatorparametrarna med utgångspunkt från den skattade modellen beräknas.
3. En regulator i form av en polplacerare, baserad på parametrar från design delen.

Till skillnad från de tidigare regulatorerna har denna implementerats på diskret form. Ett blockschema över process och regulator visas i fig 5.1 .




**Figur 5.1** Blockschema för en indirekt självinställare med polplacering som underliggande reglerprincip.

Processens överföringsfunktion kommer på diskret form att bli:

$$y(t) = \frac{B(q)}{A(q)} \cdot u(t) = \frac{b_0 q + b_1}{q^2 + a_1 q + a_2} \cdot u(t)$$

Vilket kan skrivas om som:

$$\begin{aligned} y(t) &= b_0 u(t-1) + b_1 u(t-2) - a_1 y(t-1) - a_2 y(t-2) \\ &= \theta^T \phi(t-1) \end{aligned}$$

$$\theta^T = \begin{pmatrix} b_0 & b_1 & a_1 & a_2 \end{pmatrix}$$

$$\phi^T(t-1) = \begin{pmatrix} u(t-1) & u(t-2) & -y(t-1) & -y(t-2) \end{pmatrix}$$

Algoritmen för den rekursiva minstakvadrat skattaren är (enl. ekv. 5.2-5.4 Adaptive Control).

$$
\begin{cases}
\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\varepsilon(t) \\
\varepsilon(t) = y(t) - \phi^T(t-1)\hat{\theta}(t-1) \\
K(t) = \dfrac{P(t-1)\phi(t-1)}{(\lambda + \phi^T(t-1)P(t-1)\phi(t-1))} \\
P(t) = \left(I - K(t)\phi^T(t-1)\right)\dfrac{P(t-1)}{\lambda}
\end{cases}
$$

Signalerna som används vid skattningen differensbildas för att undvika att parametrarna driver iväg vid stationaritet. Sampeltiden $h$ anges i antal sampel per period ($np$) av självsvängningsfrekvensen $\omega = \omega_0\sqrt{1-\zeta^2}$, vilket ger:

$$h = \frac{2\pi}{np \cdot \omega}$$

En listning av simnon koden för regulatorn finns i appendix . I fig 5.2 har regulatorn simulerats med processen:

$$G_p(s) = \frac{3}{s^2 - 0.5s + 4}$$

och modellen:

$$G_m(s) = \frac{1}{s^2 + 1.4s + 1}$$

På diskret form blir processens överföringsfunktion:

$$H_p(q) = \frac{1.029q + 1.211}{q^2 + 0.4399q + 1.553}$$

Den överföringsfunktion som regulatorn identifierat är följande:

$$H(q) = \frac{1.024q + 1.203}{q^2 + 0.4237q + 1.545}$$

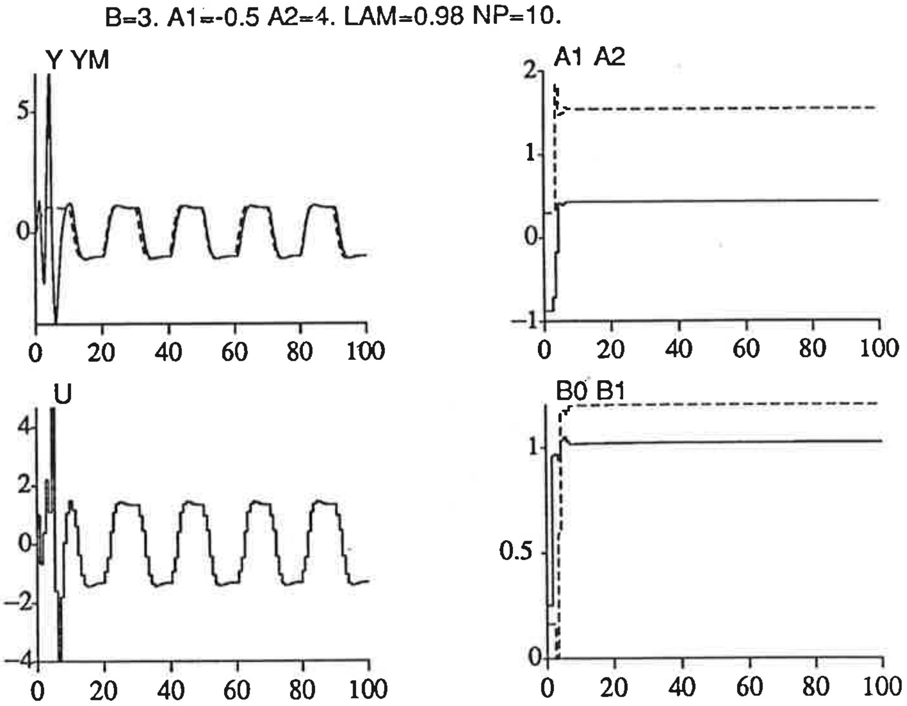
Som synes råder god överenstämmelse med den verkliga processen.

Självinställaren fungerar alltså bra även då processen har helt andra egenskaper än modellen, så länge dom är av samma ordning. Anledningen är att denna regulator så att säga bildar sig en egen uppfattning om hur processen ser ut och sedan baserar regleringen på detta.

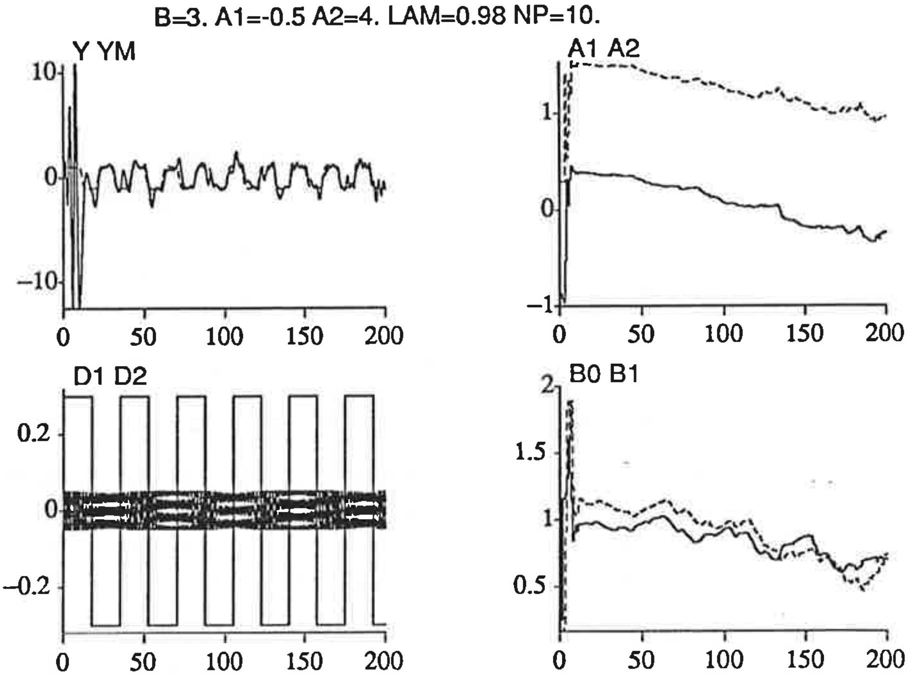
En simulering med en laststörning med amplitud 0.3 och mätbrus med amplituden 0.05 pålagt visas i fig 5.3. Man ser att modellparametrarna exciteras av störningarna och driver förbi sina rätta värden. Fortsätter man simuleringen ser man att parametrarna pendlar kring ett någorlunda konstant värde, fig 5.4. Mätbruset skulle man kanske kunna klara av med hjälp av filtrering av de signaler som används för identifieringen. När det gäller laststörningen kunde en framkoppling troligen förbättra uppförandet.

Nackdelen med denna typ av regulator är att om inte in- och utsignalerna är tillräckligt exciterande kommer skattningarna och därmed regleringen att fungera dåligt. Man kan då behöva lägga på extra störningar eller hålla på och ändra på börvärdet.

255

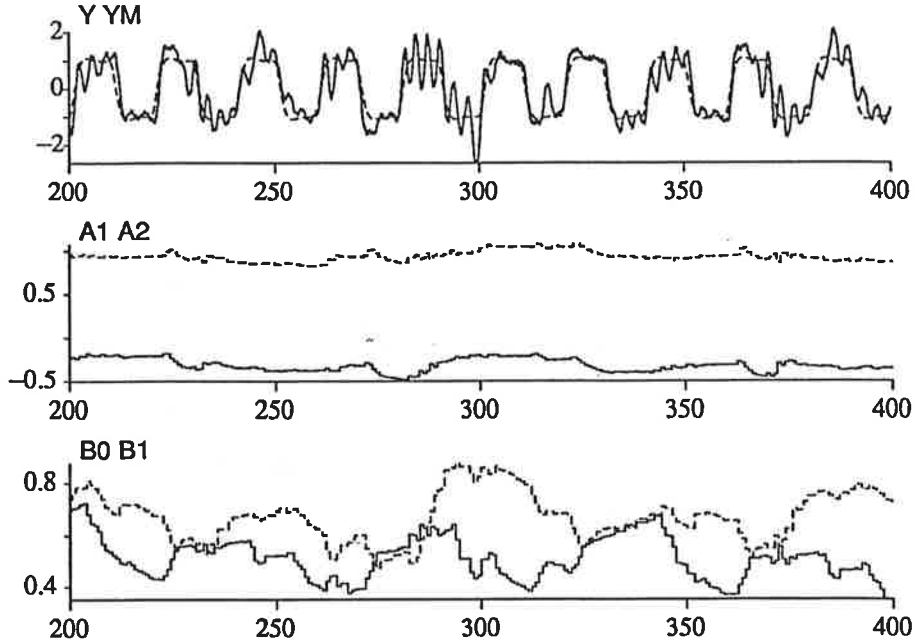



**Figur 5.2** Simulering med en självinställande regulator.



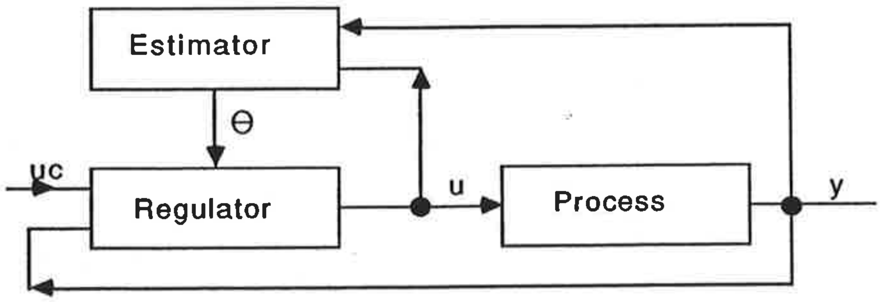

**Figur 5.3** Simulering med en självinställande regulator, där en laststörning och mätbrus lagts på.

256.




**Figur 5.4 Fortsättning på simuleringen i fig 5.3**

# 6. Direkt självinställare




**Figur 6.1** Blockschema för en direkt självinställande regulator.

I en direkt självinställande regulator skrivs processmodellen om så att systemet direkt skattar regulator parametrarna, utifrån signalerna från process och modell. Därmed kan man minska på antalet beräkningssteg jämfört med en indirekt självinställande regulator.

Estimeringen av regulator parametrarna sker i diskret form vilket medför att beräkningarna måste ske med följande samplade överföringsfunktioner.

$$H(q) = \frac{b_1 q + b_2}{q^2 + a_1 q + a_2}$$
$$H_m(q) = \frac{b_{m1} q + b_{m2}}{q^2 + a_{m1} q + a_{m2}}$$

Den diophantiska ekvationen opererande på y(t) samt ekvationen för den önskade överföringsekvationen, ger ekvationerna

$$\begin{aligned} A_0 A_m y(t) &= R_1 A y(t) + B^- S y(t) \\ &= R_1 B u(t) + B^- S y(t) \\ &= B^-(R u(t) + S y(t)) \\ A_m y_m(t) &= B_m u_c(t) \\ A_0 A_m y_m(t) &= A_0 A_m u_c(t) \end{aligned}$$

felet mellan process och modell kan då skrivas.

$$\begin{aligned} e(t) &= y(t) - y_m(t) \\ &= \frac{B^-}{A_0 A_m}(R u(t) + S y(t) - A_0 B'_m u_c(t)) \\ &= \frac{B^-}{A_0 A_m}(R u(t) + S y(t) - T u_c(t)) \end{aligned}$$

Förkortning av processnollstället samt integralverkan ger följande gradtal för polynomen: $A_0$ ett, S två och $R'_1$ noll.

258.

De här valen på gradtalen skulle ge integralverkan i en icke-adaptiv regulator, där parametrarna beräknas en gång och $R(z) = (z-1)R_1'B^+$. I den adaptiva regulatorn skattas hela $R$-polynomet och därför är det inget som garanterar att $R$-polynomet innehåller nollstället $z = 1$. För att erhålla integralverkan måste ekvationerna modifieras.

Diophantiska ekvationen skrivs

$$A_0(z)A_m(z) = R_1'(z)\cdot(z-1)A(z) + B^-(z)S(z)$$

varur fås att

$$A_0(1)A_m(1) = B^-(1)S(1)$$

$S$-polynomet kan då skrivas

$$\begin{aligned} S(z) &= S(1) + S'(z-1) \\ &= \frac{A_0(1)A_m(1)}{B^-(1)} + S'(z-1) \end{aligned}$$

inför

$$\begin{aligned} \Delta u(t) &= (z-1)u(t) \\ \Delta y(t) &= (z-1)y(t) \end{aligned}$$

Den diophantiska ekvationen opererande på y(t) ger ekvationen.

$$\begin{aligned} A_0(z)A_m(z)y(t) &= R_1'(z)\cdot(z-1)A(z)y(t) + B^-(z)[S(1) + S'(z-1)]y(t) \\ &= R_1'(z)B(z)\Delta u(t) + \frac{B^-(z)A_0(1)A_m(1)}{B^-(1)}y(t) + B^-(z)S'(z)\Delta y(t) \end{aligned}$$

Vilket kan skrivas som:

$$\begin{aligned} A_0(z)A_m(z)y(t) - \frac{B^-(z)A_0(1)A_m(1)}{B^-(1)}y(t) = B^-(z)[&R_1'(z)B^+(z)\Delta u(t) \\ &+ S'(z)\Delta y(t)] \end{aligned}$$

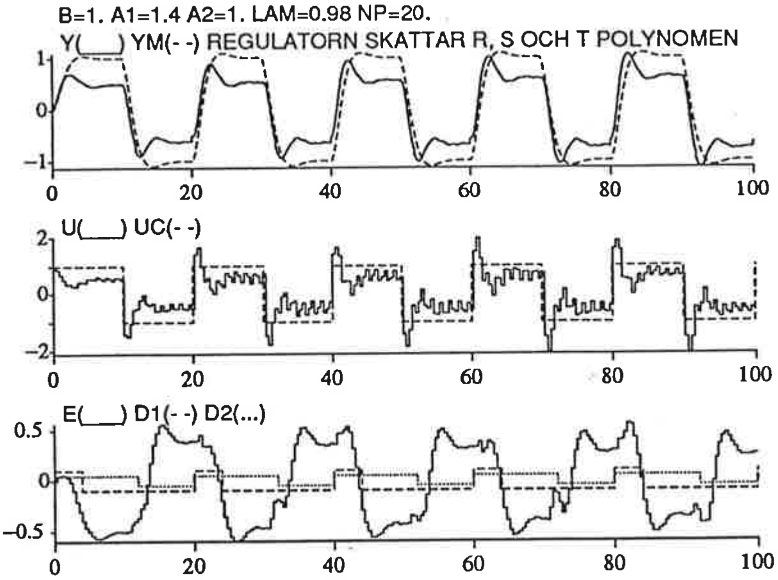
Används denna ekvationen garanteras integralverkan i den adaptiva regulatorn. Regulatorn som simuleras i detta projekt använder sig av den första versionen, det vill säga en regulator utan integralverkan.

Parametrarna uppdateras med en minsta kvadrat skattare med glömskafaktor.

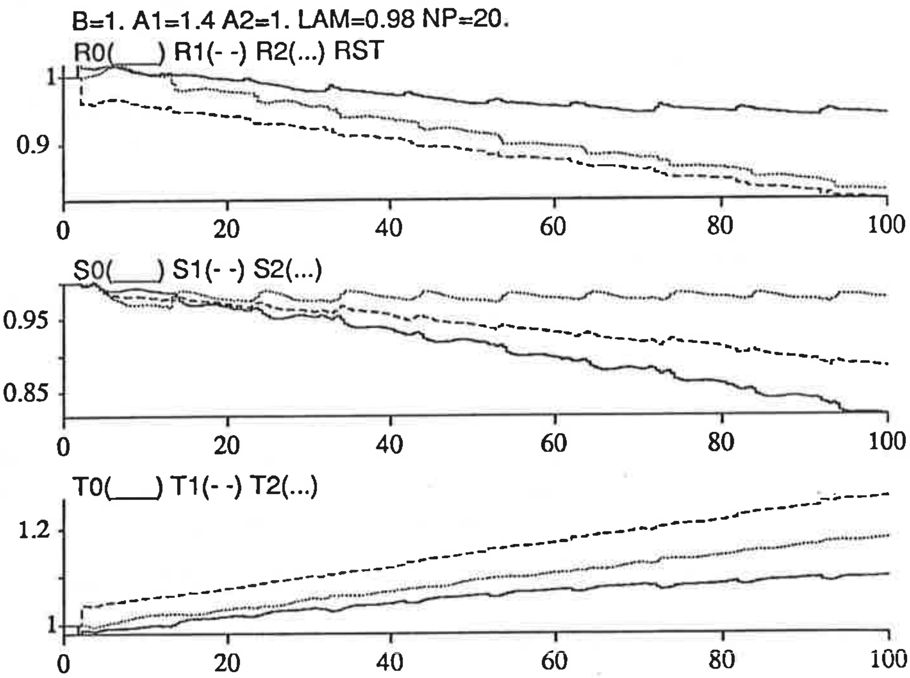
$$\begin{aligned} \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\epsilon(t) \\ \epsilon(t) &= e(t) - \phi^T(t-1)\hat{\theta}(t-1) \\ K(t) &= P(t-1)\phi(t-1)(\lambda + \phi^T(t-1)P(t-1)\phi(t-1))^{-1} \\ P(t) &= \frac{I - K(t)\phi^T(t-1)P(t-1)}{\lambda} \end{aligned}$$

Där $\theta$ vektorn skrivs $\theta^T = [r_0 r_1 r_2 s_0 s_1 s_2 t_0 t_1 t_2]$ Glömskefaktorn visar sig vara kritisk, för att få konvergens av parametrarna och därmed önskat uppförandet på systemet se figur 6.2 och figur 6.3 jämför med med figur 6.4 och figur 6.5 där glömskefaktorn är ett.

359.

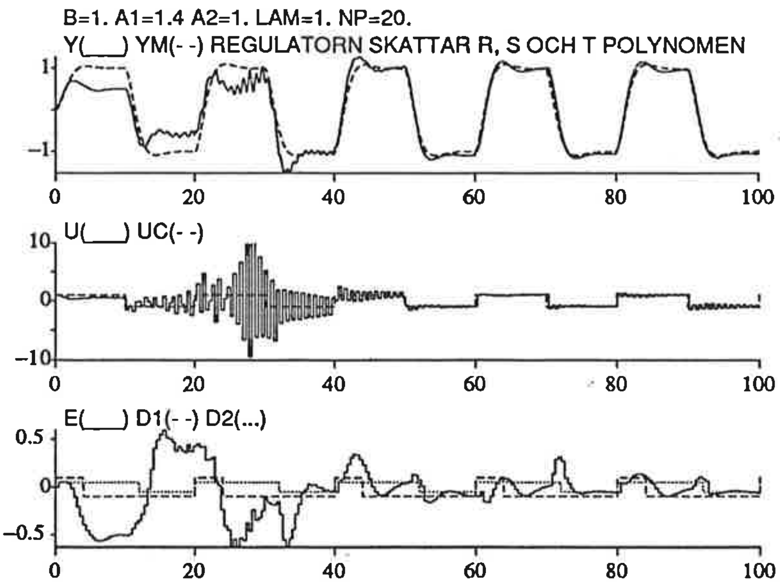



**Figur 6.2** Direkt STR

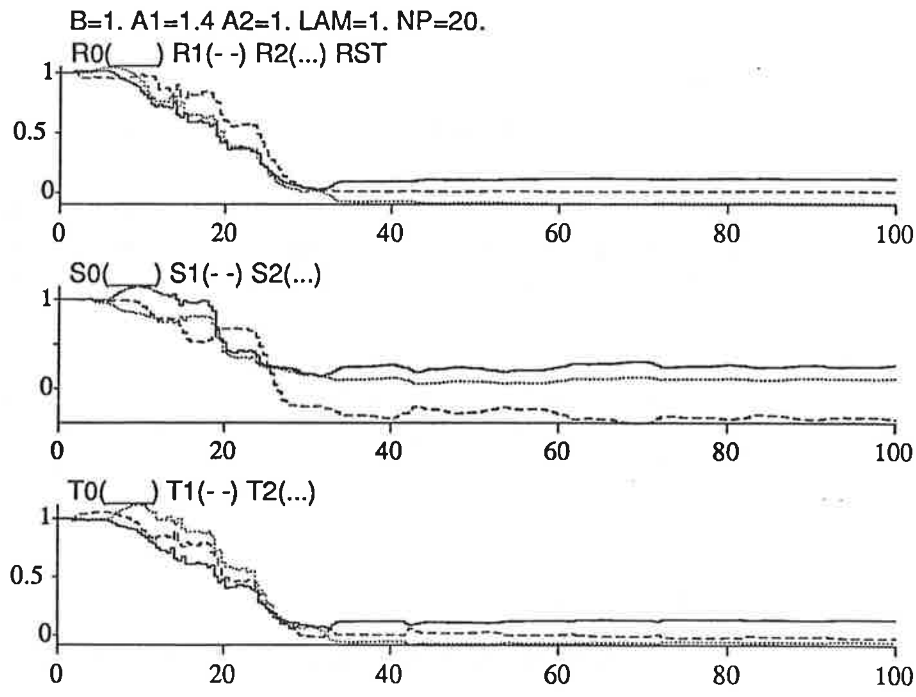



**Figur 6.3** Direkt STR ,parametrarna

260-

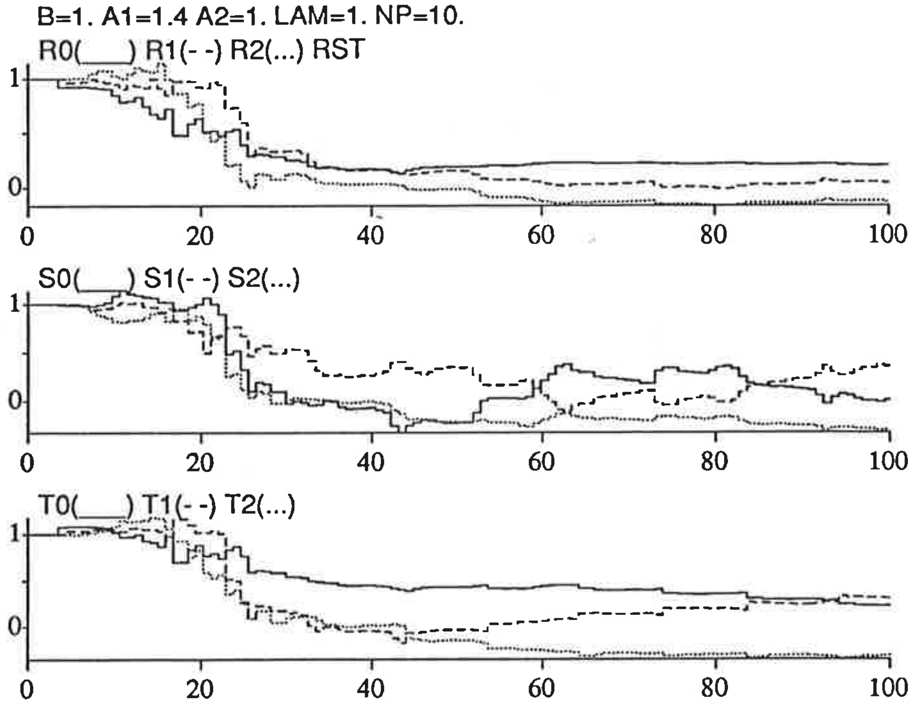



**Figur 6.4** Direkt STR glömskefaktorn är ett



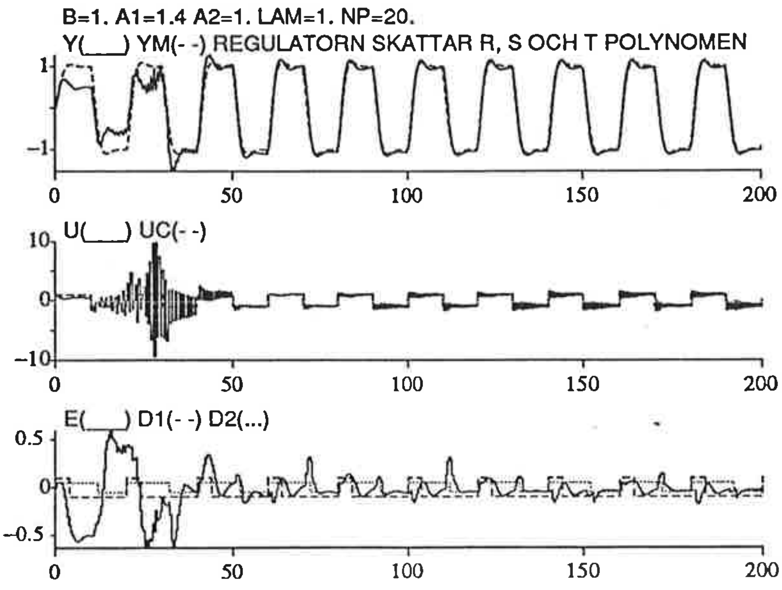

**Figur 6.5** Direkt STR glömskefaktorn är ett

Långsammare sampling medför att nollstället närmar sig det väldämpade området i enhetscirkeln, men det medför också långsammare parameterkonvergens se figur 6.6 .




**Figur 6.6** Direkt STR långsammare sampling

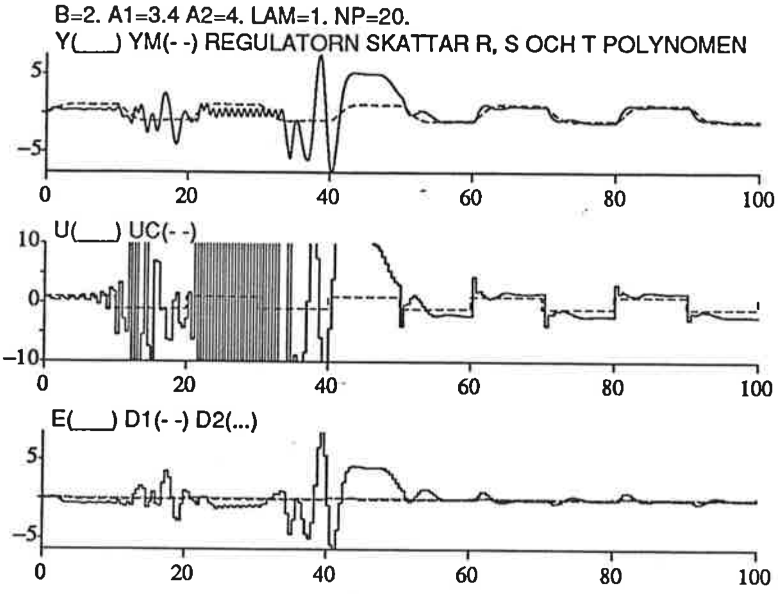
Simuleringarna visar att det krävs snabb sampling för att felet skall gå mot noll, vilket ger upphov till ett dåligt dämpat processnollställe. Förkortningen av det dåligt dämpade processnollstället ger ringningar i styrsignal se figur 6.7 .



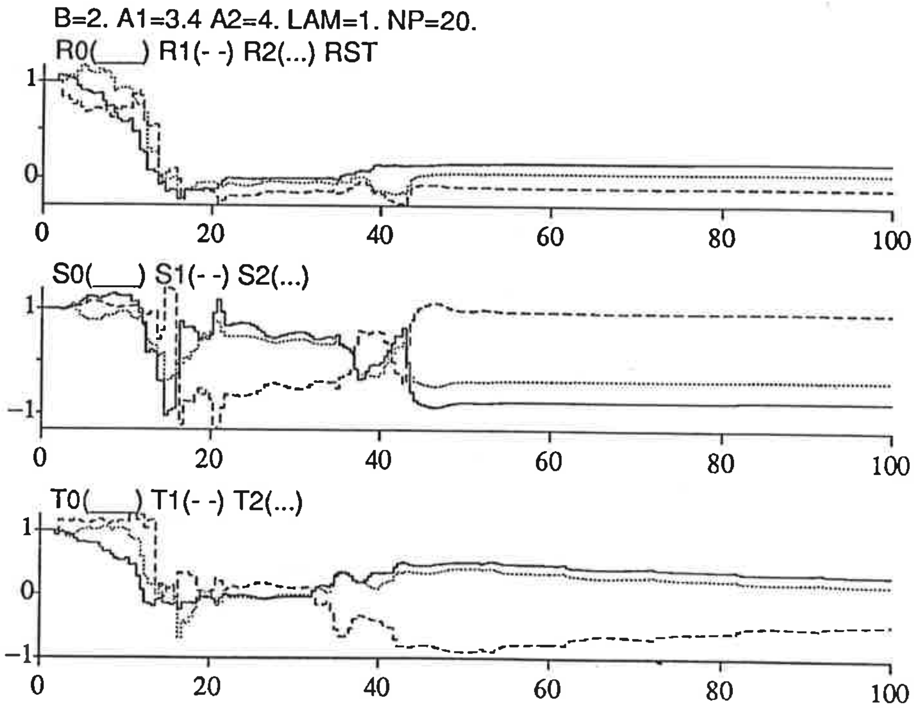

**Figur 6.7** Direkt STR ringningar i styrsignalen

262.

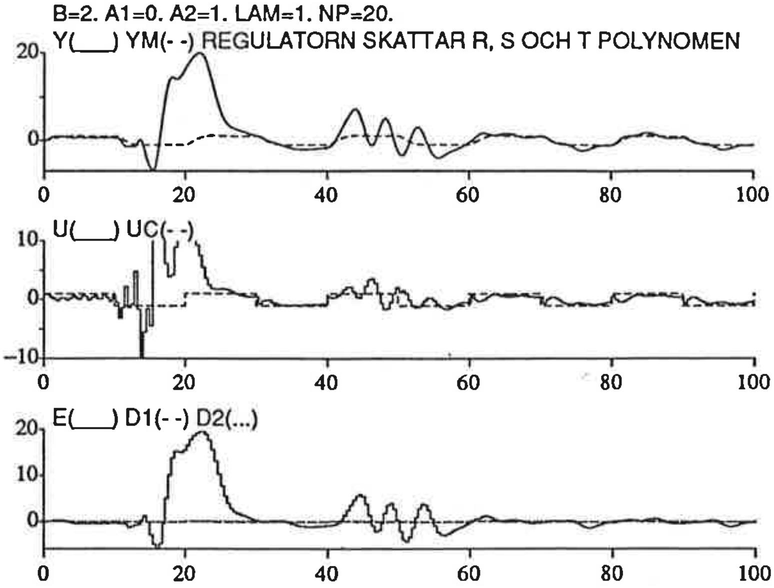
Förutsatt snabb sampling och glömskefaktorn lika med ett klarar systemet av att reglera processen för många olika värden på processparametrarna dock långt ifrån alla. Några exempel på simuleringar visas i figurerna 6.8 till 6.13 . Simnon-koden för regulatorn finns i appendix.



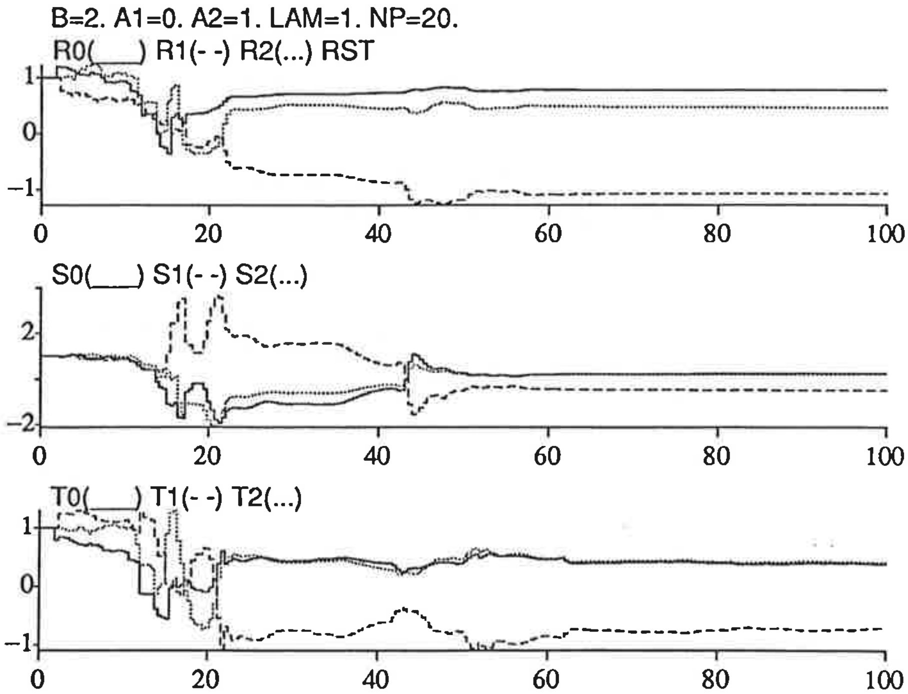

**Figur 6.8** Direkt STR polerna i VHP




**Figur 6.9** Direkt STR polerna i VHP, parametrarna

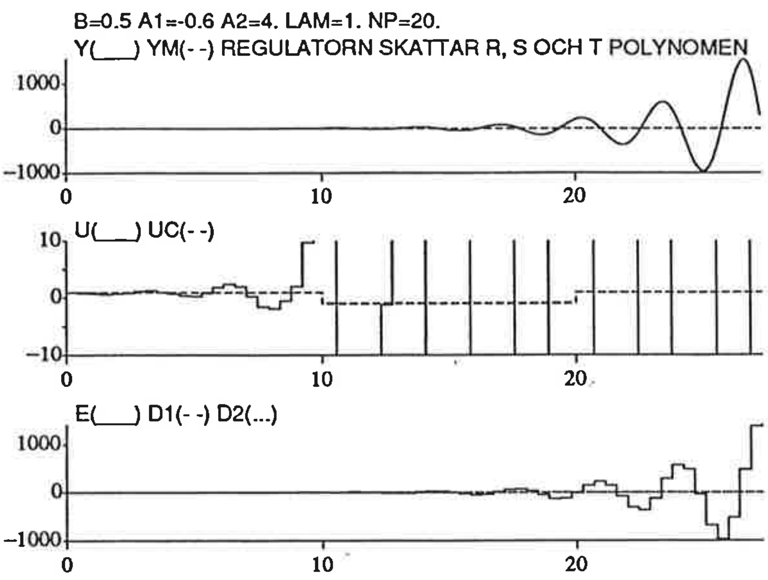



**Figur 6.10** Direkt STR polerna på imaginära axeln

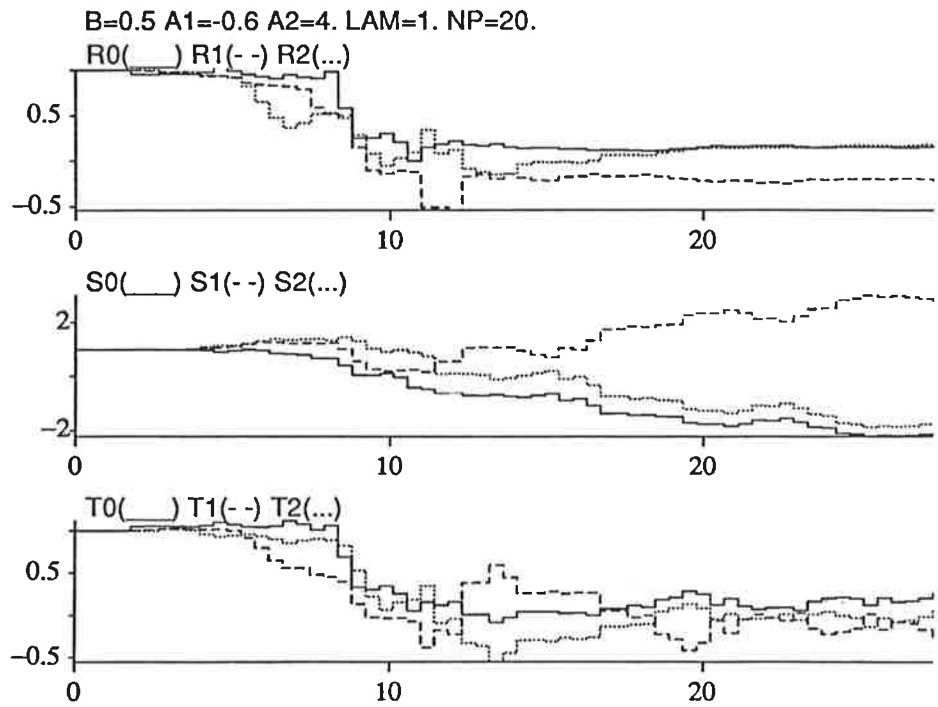



**Figur 6.11** Direkt STR polerna på imaginära axeln, parametrarna

264.




**Figur 6.12** Direkt STR polerna i HHP




**Figur 6.13** Direkt STR polerna i HHP, parametrarna

265.

# Appendix

## Simnon listningar

266.

```
CONTINUOUS SYSTEM REG
"MRAS REALISERAD AV FORMLERNA PA SIDAN 11 I AVSNITT 4.3 .
"REGULATORN SKATTAR (UPPDATERAR) PARAMETRARNA
"R1, S0, S1, T0 OCH T1 SAMT NORMERAR
"                 gamma*e*de/dfi
"       dfi/dt=- ----------------------
"                 alfa+(de/df)T*(de/df)
"
"
"                        B
"       G(s)=    ------------------------
"                     s^2+A1*s+A2
"
"
"                        BM
"       GM(s)=    -------------------------
"                     s^2+AM1*s+AM2
"
"
"


INPUT UC Y
OUTPUT U
STATE XR11  XR12  XR13  XS01  XS02  XS03  XS11  XS12 XS13
STATE XT01  XT02  XT03  XT11  XT12  XT13  XM1   XM2
STATE XT    XS    R1    S0    S1    T0    T1
DER   DXR11 DXR12 DXR13 DXS01 DXS02 DXS03 DXS11 DXS12 DXS13
DER   DXT01 DXT02 DXT03 DXT11 DXT12 DXT13 DXM1  DXM2
DER   DXT   DXS   DR1   DS0   DS1   DT0   DT1
TIME T

BM:1            "MODELLENS PARAMETRAR
AM1:1.4
AM2:1

ALFA:0.01       "UNDVIKER DIV MED NOLL, ATERFINNS I DEN
                "DEN AR NAMNARE I UPPDATERINGSREGLEN

A0:2            "OBSERVERARPOLYNOMETS KOEFFICIENT

GAMMA:1         "REGULATOR UPPDATERINGSFARSTARKNING

"A0(s)*AM(s)=s^3+AA1*s^2+AA2*s+AA3

AA1=A0+AM1
AA2=AM2+A0*AM1
AA3=A0*AM2
```

Lista 1. MIT-regulator

357.

```
"                 -B INGÅR I GAMMA
"
"            de            -B
"       XR1=---- = ---------------- * U
"            dR1      A0(s)*AM(s)
"
"               ! -AA1 -AA2 -AA3 !          ! 1 !
"       dXRi    !                !          !   !
"       ----- = !   1    0    0  ! * XR1i + ! 0 ! * U
"        dt     !                !          !   !
"               !   0    1    0  !          ! 0 !
"
"       XR1  =  !   0    0    1  ! * XR1i
"
" STYRBARFORM  Xprick=A(matris)*X+B(matris)*U
"                     Y=C(matris)*X
"
"       C1   =  !   0    0    1  !
"
"       C2   "  !   0    1    0  !

DXR11=U-AA1*XR11-AA2*XR12-AA3*XR13
DXR12=XR11
DXR13=XR12
XR1=XR13

"
" XS0iprick=A(matris)*XS0i+B(matris)*Y
"
" XS0=C2(matris)*XS0
"

DXS01=Y-AA1*XS01-AA2*XS02-AA3*XS03
DXS02=XS01
DXS03=XS02
XS0=XS02


"
" XS1iprick=A(matris)*XS1i+B(matris)*Y
"
" XS1=C1(matris)*XS1
"

DXS11=Y-AA1*XS11-AA2*XS12-AA3*XS13
DXS12=XS11
DXS13=XS12
XS1=XS13
```

Lista 2. MIT-regulator

268.

```
DXM1=UC-AM1*XM1-AM2*XM2
DXM2=XM1
YM=BM*XM2          "MODELLEN GM(S)=BM/(S^2+AM1*S+AM2

E=Y-YM


"
"        XT0iprick=A(matris)*XT0i+B(matris)*UC
"
"        XT0=C2(matris)*XT0
"

DXT01=UC-AA1*XT01-AA2*XT02-AA3*XT03
DXT02=XT01
DXT03=XT02
XT0=XT02


"
"        XT1iprick=A(matris)*XT1i+B(matris)*UC
"
"        XT1=C1(matris)*XT1
"

DXT11=UC-AA1*XT11-AA2*XT12-AA3*XT13
DXT12=XT11
DXT13=XT12
XT1=XT13


"
"    DEN=ALFA+(de/dfi)T(de/dfi)
"

DEN=ALFA+XR1*XR1+XS0*XS0+XS1*XS1+XT0*XT0+XT1*XT1

"UPPDATERINGEN AV R1 S0 S1 T0 T1

DR1=GAMMA*E*XR1/DEN
DS0=GAMMA*E*XS0/DEN
DS1=GAMMA*E*XS1/DEN
DT0=-GAMMA*E*XT0/DEN
DT1=-GAMMA*E*XT1/DEN

"STYRLAGEN R(S)*U(T)=T(S)*UC(T)-S(S)*Y(T)
"R(S)=S+R1  T(S)=T0*S+T1  S(S)=S0*S+S1
"XT=UC/(S+R1)  XS+Y/(S+R1)
```

**Lista 3.** MIT-regulator

369.

```
DXT=UC-R1*XT
DXS=Y-R1*XS

U=T0*UC+(T1-T0*R1)*XT-S0*Y-(S1-S0*R1)*XS

END
```

**Lista 4.** MIT-regulator

```
CONTINUOUS SYSTEM REG

" MRAS-REGULATOR ENLIGT LYAPUNOV.
" DESIGN MED MODELLFOLJNING ENLIGT UPPG 4.8
" PARAMETRAR: P1 P2
"
"                                           BM
" ANDRA ORDNINGENS MODELL Gm(S) = -----------------
"                                  S^2 + AM1*S + AM2
"
" STANDARD: BM = 1 , AM1 = 1.4 , AM2 = 1

INPUT UC Y
OUTPUT U
TIME T

STATE L1 L2 M
STATE XM1 XM2 XF1 XF2 XF3 Z
DER DL1 DL2 DM
DER DXM1 DXM2 DXF1 DXF2 DXF3 DZ

" MODELLEN

DXM1 = BM*UC - AM1*XM1 - AM2*XM2
DXM2 = XM1
YM = XM2

" APPROXIMERAD DERIVERING

DZ = -WF*Z-WF*WF*Y
XS1 = Z + WF*Y

" REGULATORN

E1 = XS1 - XM1
E2 = Y - YM                    " Y=X2
```

**Lista 5.** MRAS regulator enligt Lyapunov

270.

```
"LÅGPASSFILTRERING AV UPPDATERINGSHASTIGHETEN PE

PE=(P1*E1+P2*E2)
DXF1= -3*WE*XF1-3*WE*WE*XF2-WE*WE*WE*XF3+PE*WE*WE*WE
DXF2= XF1
DXF3= XF2
FPE=XF3
UPPD= IF ABS(FPE)>DMIN THEN PE ELSE 0

"UPPDATERING AV REGLERPARAMETRARNA

DL1 = XS1 * UPPD
DL2 = Y * UPPD
DM = -UC * UPPD
U = -L1*XS1 - L2*Y + M*UC      "XS1 SKATTNING AV X1, Y = X2

" PARAMETRAR

 P1:0.5
 P2:0.2
BM:  1.0
AM1: 1.4
AM2: 1.0
DMIN:0.01   "DÖD ZON

" FILTER PARAMETRAR

WE:0.2
WF : 8.0
END
```

**Lista 6.** MRAS regulator enligt Lyapunov

371.

```
DISCRETE SYSTEM REG

" INDIREKT STR-REGULATOR
" MODELLENS POLPOLYNOM ANGES PÅ FORMEN: S^2 + 2*Z*W*S + W^2
" DEFAULTVÄRDEN ÄR : Z = 0.7
"                    W = 1.4
" OBSERVERARPOLERNA ANGES PÅ FORMEN: S^2 + 2*Z0*W0*S + W0^2
" DEFAULTVÄRDEN ÄR : Z0 = 0.7
"                    W0 = 2.0
"
INPUT Y UC
OUTPUT U

STATE Y1  Y2  YF  UC1 UC2 U1  U2  V1  V2
STATE F1  F2  F3  F4  TH1 TH2 TH3 TH4
STATE P11 P12 P13 P14 P22 P23 P24 P33 P34 P44
NEW NY1 NY2 NYF NUC1 NUC2 NU1 NU2 NV1 NV2
NEW NF1 NF2 NF3 NF4 NTH1 NTH2 NTH3 NTH4
NEW NP11 NP12 NP13 NP14 NP22 NP23 NP24 NP33 NP34 NP44
TIME T
TSAMP TS

" STEG 1  PARAMETER SKATTNING

" 1.1 DIFFERENSBIKDNING AV Y => YF
"     E = YF - FI*TH
NYF = Y-Y1
E = NYF-F1*TH1-F2*TH2-F3*TH3-F4*TH4

" 1.2  K=P*FI  DEN=LAM+FI*P*FI
K1 = P11*F1+P12*F2+P13*F3+P14*F4
K2 = P12*F1+P22*F2+P23*F3+P24*F4
K3 = P13*F1+P23*F2+P33*F3+P34*F4
K4 = P14*F1+P24*F2+P34*F3+P44*F4
DEN = LAM+K1*F1+K2*F2+K3*F3+K4*F4

" 1.3  UPPDATERING AV SKATTNINGARNA
"       FI(T)=FI(T-1)+K*E
NTH1 = TH1+K1*E/DEN
NTH2 = TH2+K2*E/DEN
NTH3 = TH3+K3*E/DEN
NTH4 = TH4+K4*E/DEN

" 1.4 UPPDATERING AV KOVARIANSER
NP11 = (P11-K1*K1/DEN)/LAM
NP12 = (P12-K1*K2/DEN)
NP13 = (P13-K1*K3/DEN)
NP14 = (P14-K1*K4/DEN)
```

**Lista 7.** Indirekt STR-regulator

```
NP22 = (P22-K2*K2/DEN)/LAM
NP23 = (P23-K2*K3/DEN)
NP24 = (P24-K2*K4/DEN)
NP33 = (P33-K3*K3/DEN)/LAM
NP34 = (P34-K3*K4/DEN)
NP44 = (P44-K4*K4/DEN)/LAM
" 1.5  UPPDATERING AV Y1=Y(T-1).. UC1=UC(T-1)...
NY1  = Y
NY2  = Y1
NUC1 = UC
NUC2 = UC1
NU1  = U
NU2  = U1
NV1  = V
NV2  = V1

" 1.6  UPPDATERING OCH DIFFERENSBILDNING AV REGRESSIONSVEKTORN FI
NF1 = U-U1
NF2 = F1
NF3 = Y+Y1
NF4 = F3

" STEG 2  REGULATOR
" 2.1  BERÄKNING AV AM OCH AO
SQ = SQRT(1-Z*Z)
H = 2*PI/(NP*W*SQ)
AF = EXP(-NF/NP)
AM1 = -2*EXP(-Z*W*H)*COS(W*H*SQ)
AM2 = EXP(-2*Z*W*H)
SQ0 = SQRT(1-Z0*Z0)
A01 = -2*EXP(-Z0*W0*H)*COS(W0*H*SQ0)
A02 = EXP(-2*Z0*W0*H)

" 2.2  BER[KNING AV REGULATORPARAMETRAR
B0 = TH1
B1 = TH2
A1 = TH3
A2 = TH4
```

**Lista 8.** Indirekt STR-regulator

273-

```
P = -B1/B0
SLASK = (P-1)*(P*P+A1*P+A2)
R = (P*P+A01*P+A02)*(P*P+AM1*P+AM2)/SLASK-P
S0 = (AM1+A01-A1+1-R)/B0
S2 = (A02*AM2+A2*R)/B1
S1 = (A01*AM2+A02*AM1+A2-R*(A2-A1)-B0*S2)/B1
R1 = R-1
R2 = -R
T0 = (1+AM1+AM2)/(B0+B1)
T1 = T0*A01
T2 = T0*A02


" 2.3  BERÄKNING AV UTSIGNAL MED ANTI-WINDUP
V = -R1*U1-R2*U2+T0*UC+T1*UC1+T2*UC2-S0*Y-S1*Y1-S2*Y2
U = MIN(MAX(V,ULOW),UHIGH)

TS = T+H
"PARAMETRAR
PI: 3.1415926

"SPECIFIKATION AV REGULATORPARAMETRAR
W: 1
Z: 0.7
W0: 2
Z0: 0.7
NP: 10              "ANTAL SAMPLINGAR PER PERIOD
LAM: 0.98
UHIGH: 5000
ULOW: -5000

" INITIALVÄRDEN
TH1: 0.2518
TH2: 0.166
TH3: -0.874
TH4: 0.2918
P11: 10
P22: 10
P33: 10
P44: 10

END
```

**Lista 9.** Indirekt STR-regulator

374

```
DISCRETE SYSTEM REG
"DIREKT SELF TUNING REGULATOR
"SKATTAR R, S OCH T SKATTNINGARNA
"STYRLAGEN R(q)*U(t)=T(q)*Uc(t)-S(q)*Y(t)


INPUT Y YM UC
OUTPUT U
STATE Y1   Y2   YF   YF1  YF2  U1   U2   TH7  TH8  TH9
STATE UF   UF1  UF2  UC1  UC2  E    F7   F8   F9
STATE UCF  UCF1 UCF2 F1   F2   F3   F4   F5   F6   TH1
STATE TH2  TH3  TH4  TH5  TH6  P11  P12  P13  P14  P15
STATE P16  P17  P18  P19  P22  P23  P24  P25  P26  P27
STATE P28  P29  P33  P34  P35  P36  P37  P38  P39  P44
STATE P45  P46  P47  P48  P49  P55  P56  P57  P58  P59
STATE P66  P67  P68  P69  P77  P78  P79  P88  P89  P99
NEW NY1   NY2   NYF   NYF1  NYF2  NU1   NU2   NTH7  NTH8  NTH9
NEW NUF   NUF1  NUF2  NUC1  NUC2  NE    NF7   NF8   NF9
NEW NUCF  NUCF1 NUCF2 NF1   NF2   NF3   NF4   NF5   NF6   NTH1
NEW NTH2  NTH3  NTH4  NTH5  NTH6  NP11  NP12  NP13  NP14  NP15
NEW NP16  NP17  NP18  NP19  NP22  NP23  NP24  NP25  NP26  NP27
NEW NP28  NP29  NP33  NP34  NP35  NP36  NP37  NP38  NP39  NP44
NEW NP45  NP46  NP47  NP48  NP49  NP55  NP56  NP57  NP58  NP59
NEW NP66  NP67  NP68  NP69  NP77  NP78  NP79  NP88  NP89  NP99

TIME T
TSAMP TS

"FILTRERING
"Uf(t)=U(t)/(A0(q)*AM(q))
"Uc(t)=Uc(t)/(A0(q)*AM(q))
"Yf(t)=Y(t)/(A0(q)*AM(q))


NUF=U2-UF*(AM1+A01)-UF1*(AM2+A01*AM1)-UF2*(A01*AM2)

NUCF=UC2-UCF*(AM1+A01)-UCF1*(AM2+A01*AM1)-UCF2*(A01*AM2)

NYF=Y2-YF*(AM1+A01)-YF1*(AM2+A01*AM1)-YF2*(A01*AM2)


"PARAMETER ESTIMATION

"1.1 RESIDUALEN
EPS=NE-F1*TH1-F2*TH2-F3*TH3-F4*TH4-F5*TH5-F6*TH[illegible]TH8-F9*TH9
```

Lista 10. Direkt STR

375

```
"1.2 BER[KNAR K VEKTORN, K=P*Fi OCH DEN=LAM+Fi*P*Fi
K1=P11*F1+P12*F2+P13*F3+P14*F4+P15*F5+P16*F6+P17*F7+P18*F8+P19*F9
K2=P12*F1+P22*F2+P23*F3+P24*F4+P25*F5+P26*F6+P27*F7+P28*F8+P29*F9
K3=P13*F1+P23*F2+P33*F3+P34*F4+P35*F5+P36*F6+P37*F7+P38*F8+P39*F9
K4=P14*F1+P24*F2+P34*F3+P44*F4+P45*F5+P46*F6+P47*F7+P48*F8+P49*F9
K5=P15*F1+P25*F2+P35*F3+P45*F4+P55*F5+P56*F6+P57*F7+P58*F8+P59*F9
K6=P16*F1+P26*F2+P36*F3+P46*F4+P56*F5+P66*F6+P67*F7+P68*F8+P69*F9
K7=P17*F1+P27*F2+P37*F3+P47*F4+P57*F5+P67*F6+P77*F7+P78*F8+P79*F9
K8=P18*F1+P28*F2+P38*F3+P48*F4+P58*F5+P68*F6+P78*F7+P88*F8+P89*F9
K9=P19*F1+P29*F2+P39*F3+P49*F4+P59*F5+P69*F6+P79*F7+P89*F8+P99*F9

D=LAM+F1*K1+F2*K2+F3*K3+F4*K4+F5*K5+F6*K6+F7*K7+F8*K8+F9*K9

"1.3 UPPDATERAR SKATTNINGARNA
NTH1=TH1+K1*EPS/D
NTH2=TH2+K2*EPS/D
NTH3=TH3+K3*EPS/D
NTH4=TH4+K4*EPS/D
NTH5=TH5+K5*EPS/D
NTH6=TH6+K6*EPS/D
NTH7=TH7+K7*EPS/D
NTH8=TH8+K8*EPS/D
NTH9=TH9+K9*EPS/D

"1.4 UPPDATERAR KOVARIANSEN
NP11=(P11-K1*K1/D)/LAM
NP12= P12-K1*K2/D
NP13= P13-K1*K3/D
NP14= P14-K1*K4/D
NP15= P15-K1*K5/D
NP16= P16-K1*K6/D
NP17= P17-K1*K7/D
NP18= P18-K1*K8/D
NP19= P19-K1*K9/D
NP22=(P22-K2*K2/D)/LAM
NP23= P23-K2*K3/D
NP24= P24-K2*K4/D
NP25= P25-K2*K5/D
NP26= P26-K2*K6/D
NP27= P27-K2*K7/D
NP28= P28-K2*K8/D
NP29= P29-K2*K9/D
NP33=(P33-K3*K3/D)/LAM
NP34= P34-K3*K4/D
NP35= P35-K3*K5/D
NP36= P36-K3*K6/D
```

**Lista 11.** Direkt STR

n76.

```
NP37= P37-K3*K7/D
NP38= P38-K3*K8/D
NP39= P39-K3*K9/D
NP44=(P44-K4*K4/D)/LAM
NP45= P45-K4*K5/D
NP46= P46-K4*K6/D
NP47= P47-K4*K7/D
NP48= P48-K4*K8/D
NP49= P49-K4*K9/D
NP55=(P55-K5*K5/D)/LAM
NP56= P56-K5*K6/D
NP57= P57-K5*K7/D
NP58= P58-K5*K8/D
NP59= P59-K5*K9/D
NP66=(P66-K6*K6/D)/LAM
NP67= P67-K6*K7/D
NP68= P68-K6*K8/D
NP69= P69-K6*K9/D
NP77=(P77-K7*K7/D)/LAM
NP78= P78-K7*K8/D
NP79= P79-K7*K9/D
NP88=(P88-K8*K8/D)/LAM
NP89= P89-K8*K9/D
NP99=(P99-K9*K9/D)/LAM

"1.5 UPPDATERA Y1=Y(t-1)....
NY1=Y
NY2=Y1
NYF1=YF
NYF2=YF1
NU1=U
NU2=U1
NUF1=UF
NUF2=UF1
NUCF1=UCF
NUCF2=UCF1
NUC1=UC
NUC2=UC1
NE=Y-YM

"1.6 UPPDATERA REGRESSIONSVEKTORN Fi
NF1=NUF
NF2=F1
NF3=F2
NF4=NYF
```

**Lista 12.** Direkt STR

```
NF5=F4
NF6=F5
NF7=-NUCF
NF8=F7
NF9=F8

"REGULATOR

"2.1 BER[KNING AV AM(q), BM(q), AO(q) OCH BO(q)
W = WO*SQRT(1-Z*Z)
H = 2*PI/(NP*W)
ALFA = EXP(-Z*WO*H)
BETA = COS(W*H)
GAMMA = SIN(W*H)
AM1 = -2*ALFA*BETA
AM2 = ALFA*ALFA
BM1 = 1-ALFA*(BETA+(Z*WO*GAMMA/W))
BM2 = (1+ALFA*((Z*WO*GAMMA/W)-BETA))*ALFA
AO1 = -EXP(-AO*H)
BO1 = (1-EXP(-AO*H))/AO

"2.2 BER[KNAR PARAMETRARNA I R, S OCH T POLYMOMEN

RO=TH1
R1=TH2
R2=TH3

SO=TH4
S1=TH5
S2=TH6

TO=TH7
T1=TH8
T2=TH9

"2.3 BER[KNAR STYRSIGNALEN
"R(q)*U(t)=T(q)*UC(t)-S(q)*Y(t)

UVERKLIG=(TO*UC+T1*UC1+T2*UC2-SO*Y-S1*Y1-S2*Y2-R1*U1-R2*U2)/RO
U=IF UVERKLIG<ULOW THEN ULOW ELSE IF UVERKLIG<UHIGH THEN UVERKLIG
ELSE UHIGH

"UPPDATERA SAMPLINGSTIDEN
TS=T+H
```

Lista 13. Direkt STR

```
"PARAMETRAR
PI=3.1415926

"SPECIFIKATION AV REULATORPARAMETRAR
W0:1            "\NSKAD BANDBREDD
Z:0.7           "\NSKAD D[MPNING
A0:2            "BANDBREDD P] OBSERVERARE
LAM:0.98        "GL\MSKEFAKTOR
NP:20           "ANTAL SAMPEL PER PERIOD
ULOW:-10        "BEGR[NSNING P] UTSIGNALEN
UHIGH:10        "BEGR[NSNING P] UTSIGNALEN


"INITIALV[RDEN
TH1:1
TH2:1
TH3:1
TH4:1
TH5:1
TH6:1
TH7:1
TH8:1
TH9:1
P11:10
P22:10
P33:10
P44:10
P55:10
P66:10
P77:10
P88:10
P99:10

END
```

**Lista 14.** Direkt STR

277.

# Projekt Adaptiv Reglering

Utförd av Claudio Diaz E-84

Analys av

a) Återkoppling med hög förstärkning

b) MRAS

c) STR

på ett andra ordningens system.

Lund, May 1988

Projektet har bestått i att prova med

a) återkoppling med hög förstärkning

b) MRAS

c) STR baserad på polplacering

på det andra ordningens system beskriven av:

$$G(s) = \frac{k}{s^2 + a_2 s + a_1}$$

där

$$0.5 \leq k \leq 3.0$$

$$-0.6 \leq a_2 \leq 3.4$$

$$-2.0 \leq a_1 \leq 4.0$$

Dessa parametrar får variera mellan olika körningar men är konstanta under en körning.

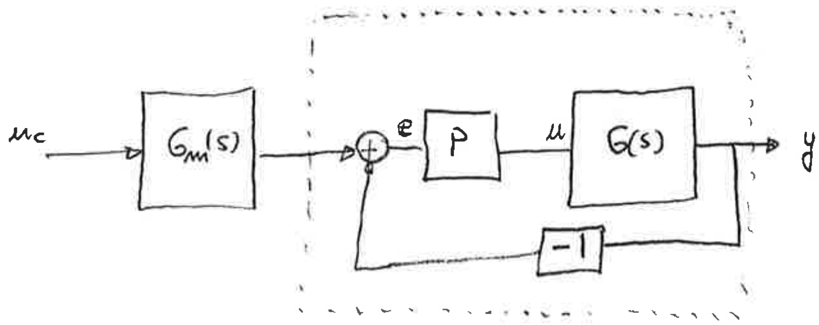
Det önskade beteende är

$$G_m(s) = \frac{1}{s^2 + 1.4 s + 1}$$

Simulering av de olika lösningarna har utförts med hjälp av Simnon.

a) Återkoppling med hög förstärkning

Idén är att med återkoppling med hög förstärkning få en överföringsfunktion som är ett.




$$G_{closed\ loop} = \frac{PG(s)}{1 + PG(s)} \simeq 1 \quad \text{om } P \text{ stort.}$$

Överföringsfunktionen från $u_c$ til $y$ blir

$$y = \frac{PG(s)}{1 + PG(s)} \cdot G_m(s)\, u_c \approx G_m(s)\, u_c \quad \text{för stora } P.$$

För att kunna realisera denna lösning krävs att

$$G_c(s) = \frac{PG(s)}{1 + PG(s)} \quad \text{är stabilt.}$$

Systemets karakteristiska ekvationen är

$$1 + PG(s) = 0 \iff s^2 + a_2 s + a_1 + Pk = 0$$

Om $a_2 \leq 0$ är systemet instabilt för alla värden på $P$.

Om $a_2 > 0$ är systemet stabilt när $a_1 + Pk > 0$, dvs $P > -a_1/k$

I vårt fall är $k > 0$, så vi har inga begränsningar på hur stort $P$ kan göras (i verkligheten skulle mätbruset sätta en övre gräns för förstärkningen).

Lösningen med återkoppling med hög förstärkning har den stora fördelen att vara enkel, men löser inte fallet med $a_2 \leq 0$.

Figur 1 visar resultaten för några olika val av parametrar, referenssignalen är en fyrkantsvåg, med perioden 40 sekunder, som varierar mellan $-1$ och $1$, denna referenssignal används för alla simuleringar.

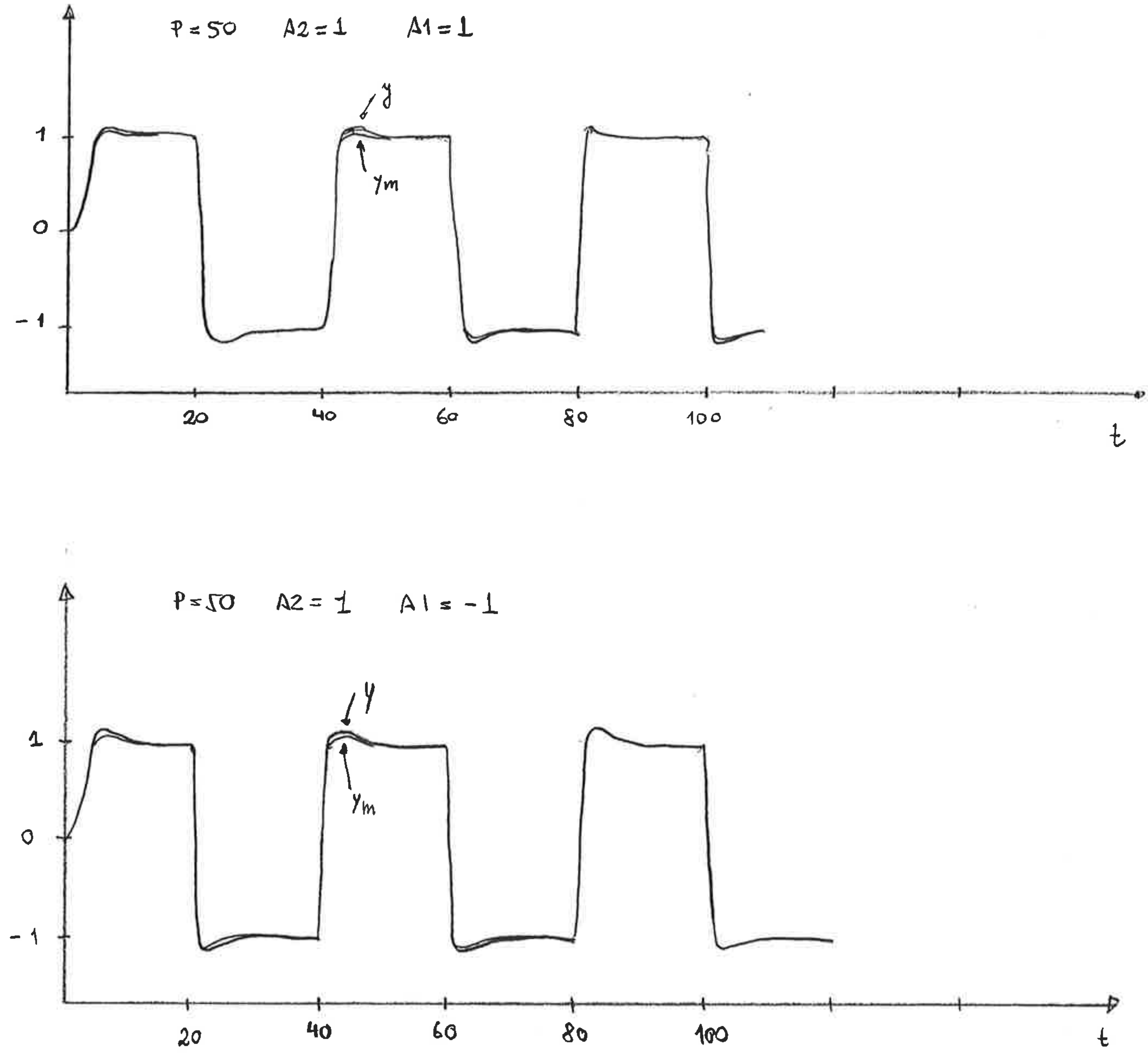



Fig 1 Återkoppling med hög förstärkning.

b) MRAS

Metoden som användes var den "normaliserad" versionen av MIT-regeln, enligt följande

Regulator

$$Ru = Tu_c - Sy$$

$$R(s) = s + r_1$$
$$T(s) = t_0 s + t_1$$
$$S(s) = s\phi s + s1$$

Model

$$A_m y_m = B_m u_c$$

$$A_m(s) = s^2 + 1.4s + 1$$
$$B_m(s) = 1$$

$$A_o(s) = s + \omega_1$$

Process

$$Ay = Bu$$

$$A(s) = s^2 + a_2 s + a_1$$

$$B(s) = K$$

Uppdateringsregeln

$$\theta = [r_1 \quad s\phi \quad s1 \quad t\phi \quad t1]^T$$

$$\frac{de}{d\theta} = \varphi = \left[ \frac{u}{A_oA_m} \quad \frac{sy}{A_oA_m} \quad \frac{y}{A_oA_m} \quad -\frac{s u_c}{A_oA_m} \quad -\frac{u_c}{A_oA_m} \right]^T$$

$$e = (y - y_m)$$

$$\frac{d\theta}{dt} = -\gamma e \frac{de/d\theta}{\alpha + \left(\frac{de}{d\theta}\right)^T\left(\frac{de}{d\theta}\right)} = -\gamma e \frac{\varphi}{\alpha + \varphi^T\varphi}$$

Ekvationen

$$\frac{BT}{AR + BS} = \frac{A_o B_m}{A_o A_m}$$

har lösningen

$$r_1 = \omega_1 + 1.4 - a_2$$
$$s\phi = (1.4\omega_1 + 1 - a_1 - a_2 r_1)/k$$
$$s1 = (\omega_1 - a_1 r_1)/k$$
$$t\phi = 1/k$$
$$t1 = \omega_1/k$$

Figur 2 visar simuleringar gjorda för olika val av parametrar, man ser att uppstarten är den kritiska perioden, med stora överslängar i utsignalen, speciellt när själva processen är instabilt.
Värdena som regulatorns parametrar konvergerat till, beror av värdet på $\gamma$, de skiljer sig från resultaten som ekvationen ger.

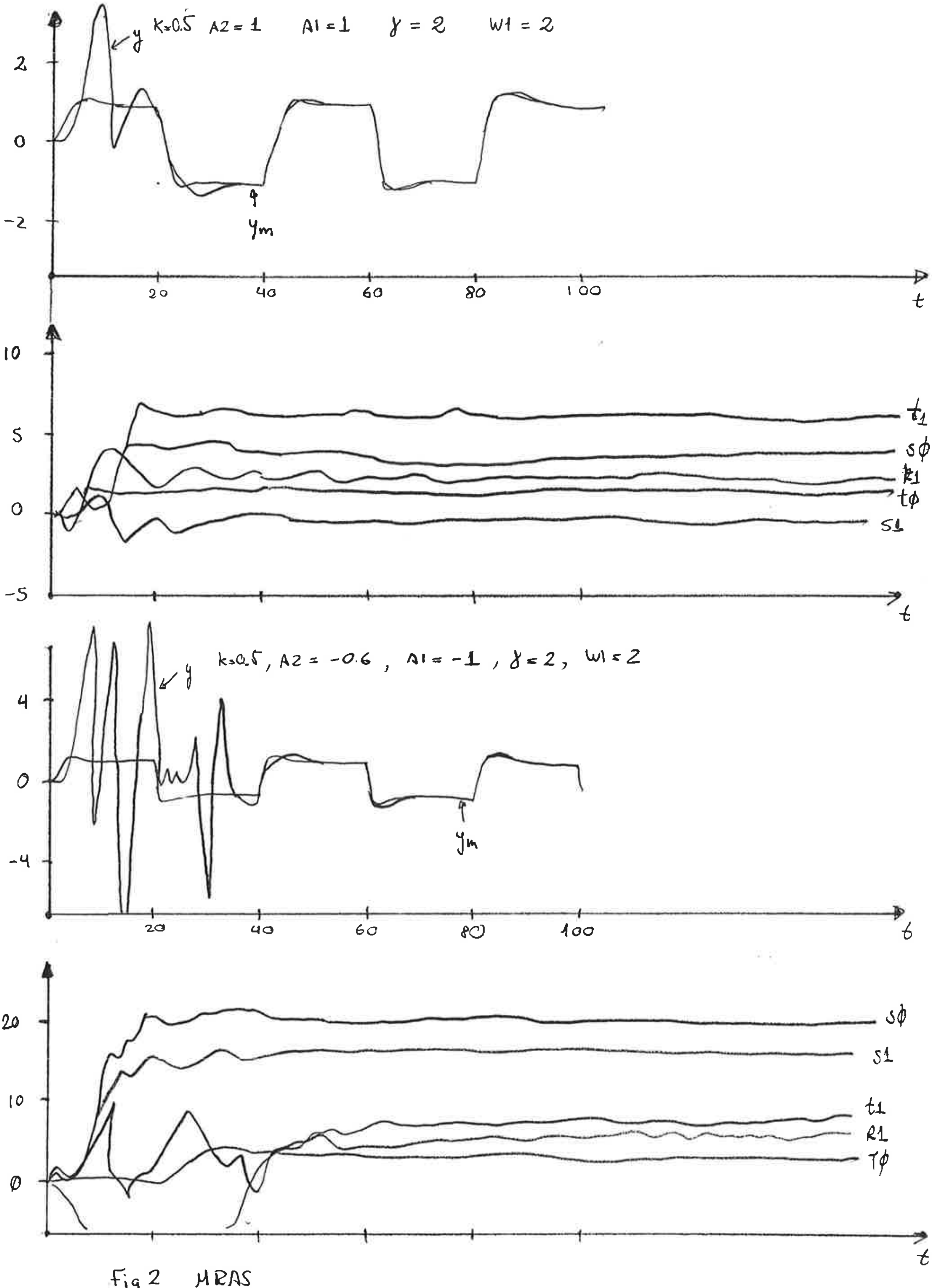

y
K=0.5 A2 = 1 A1 = 1 γ = 2 W1 = 2
ym
2
0
-2
20
40
60
80
100
t
10
5
0
-5
t1
sφ
R1
tφ
S1
y
k=0.5, A2 = -0.6 , A1 = -1 , γ = 2, W1 = 2
ym
4
0
-4
20
40
60
80
100
t
20
10
0
sφ
S1
t1
R1
Tφ
t

Fig 2 MRAS

c) STR           av 66.

Lösningen som har valts har varit indirekt självinställande regulator, med parameter uppskattning baserad på den rekursiva minsta kvadratmetoden allt implementerad i kontinuerlig tid.

"Design" är baserad på polplacering, om processen är känd så beräknas regulatorns parametrar enligt följande:

Regulator

$$Ru = Tu_c - Sy$$

$$R(s) = s + r_1$$
$$T(s) = t_0 s + t_1$$
$$S(s) = s_0 s + s_1$$

Model

$$A_m y_m = B_m u_c$$

$$A_m(s) = s^2 + 1.4s + 1.$$
$$B_m(s) = 1$$
$$A_o(s) = s + \omega_1$$

Process

$$Ay = Bu$$

$$A(s) = s^2 + \hat{a}_2 s + \hat{a}_1$$
$$B(s) = k$$

har lösningen (se MRAS)

$$r_1 = \omega_1 + 1.4 - \hat{a}_2$$
$$s_0 = (1.4\omega_1 + 1 - \hat{a}_1 - \hat{a}_2 r_1)/\hat{k}$$
$$s_1 = (\omega_1 - \hat{a}_1 r_1)/\hat{k}$$
$$t_0 = 1/\hat{k}$$
$$t_1 = \omega_1/\hat{k}$$

För att undvika numeriska fel då $\hat{k}$ är liten ersätts alla divisioner med $\hat{k}$ med multiplikationer med

$$\frac{\hat{k}}{\hat{k}^2 + \xi}, \text{ där } \xi = 0.01.$$

De här ekvationer bestämmer regulatorns parametrar då $a_2$, $a_1$ och $k$ är kända. I vårt fall används estimeringar på $a_2, a_1$ och $k$ som fås med den rekursiva minsta kvadrat metoden.

Estimering

$$\Theta = [a_2 \quad a_1 \quad k]^T$$

$$\varphi = [-s F(s) y \quad -F(s) y \quad -F(p) u]^T$$

$$y_f = s^2 F(s)$$

$$\frac{d\hat{\theta}}{dt} = P(t) \varphi(t) e(t)$$

$$e(t) = y_f(t) - \varphi^T(t) \hat{\theta}(t)$$

$$\frac{dP(t)}{dt} = \alpha P(t) - P(t) \varphi(t) \varphi^T(t) P(t)$$

med

$$F(s) = \frac{\omega_f^2}{s^2 + 2\xi\omega_f s + \omega_f^2} \cdot \frac{\omega_2}{s + \omega_2}$$

( under simuleringen är $\omega_f = 5$, $\xi = 0.7$, $\omega_2 = 10$, $\alpha = 0.01$, $P(0) = I$ )

Figur 3 visar resultaten av körning för olika val av parametrar. För att starta estimeringen används en exciterings signal till processen (insignal $u$) under 10 sekunder, amplitud 0.2, medelvärde 0 och perioden 2.

Denna insignal behövs eftersom i den idealiserad miljön som simulering pågår, kommer alla signaler att förbli noll. (loopen är öppen när alla parametrar är lika med noll)

Uppförandet hos systemet är mycket bättre än fallet MRAS, det finns inga stora översvängar i uppstarten. Parametrarna konvergerar snabbt till de rätta värdena.

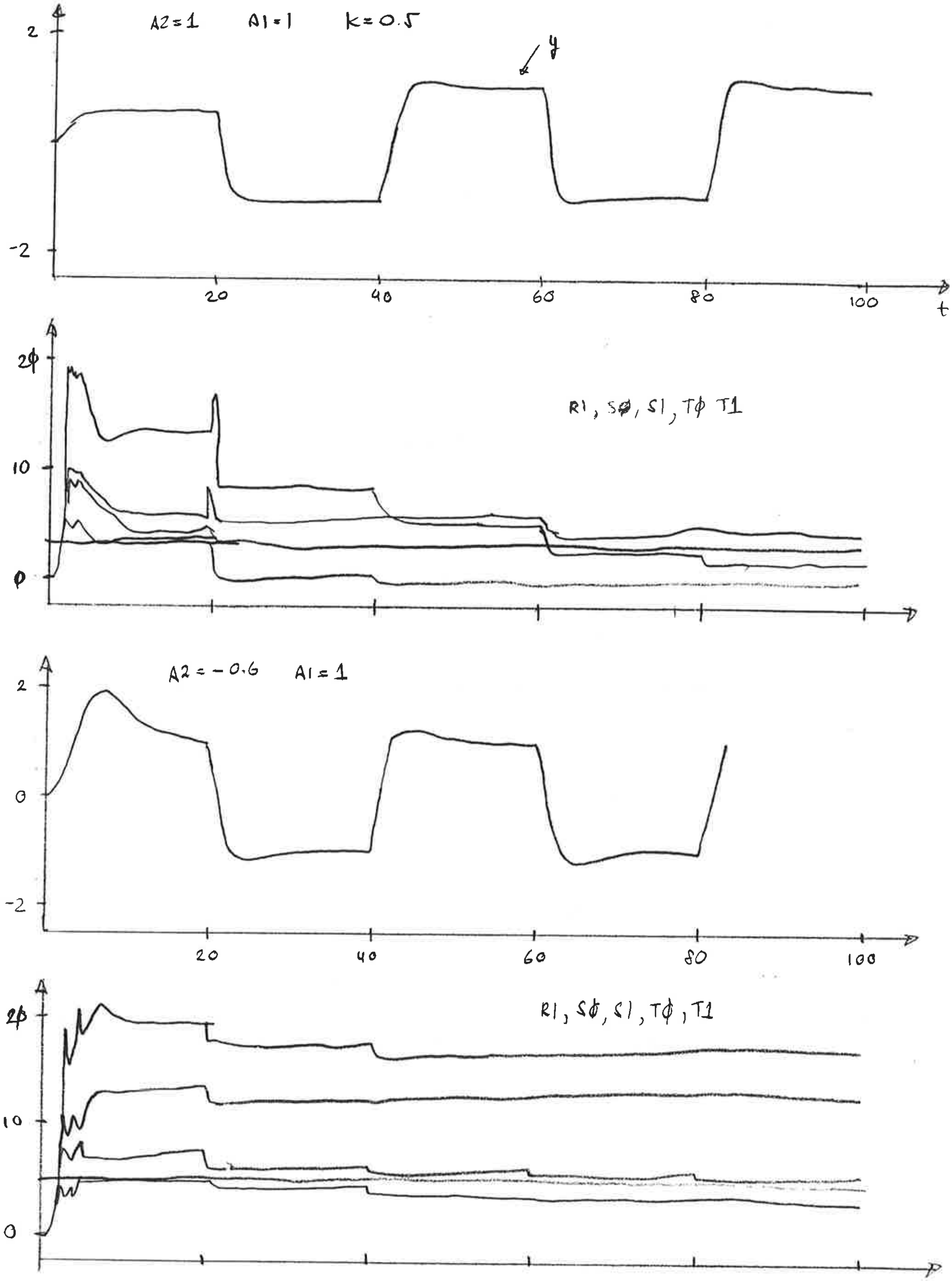
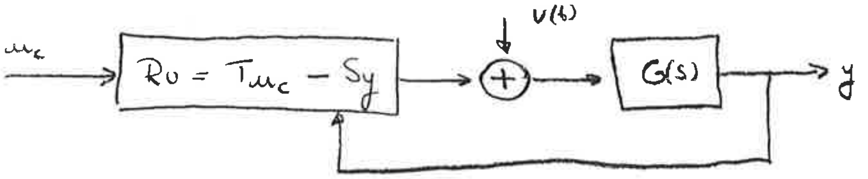

A2=1 A1=1 k=0.5
y
2
-2
20
40
60
80
100
t
R1, S∅, S1, T∅ T1
2∅
10
∅
A2 = −0.6 A1 = 1
2
0
-2
20
40
60
80
100
R1, S∅, S1, T∅, T1
2∅
10
0

Fig 3 STR

## Störningar

Om störningar kommer in till processen enligt följande




$$G(s) = \frac{B(s)}{A(s)}$$

så är överföringsfunktionen från referenssignal till utsignal

$$y = \frac{BT}{AR+BS} u_c + \frac{BR}{AR+BS} v$$

Man ser att påverkan av störningar beror på valet av regulatorn, känner man naturen av störningen kan valet av regulatorn göras så att påverkan blir "så liten som möjligt".

När regulatorn är adaptiv, och störningen kan ej mätas, är huvudfrågan om konvergens mot den önskade regulatorn är möjligt. På grund av att systemet är olinjärt, för det adaptiva fallet, är en analys av konvergensen mycket svårt.

En analys baserad på simulering har utförts för de olika fall

a) Återkoppling med hög förstärkning

I detta fall är $R = 1$
$T = S = P$

och

$$y = \frac{BP}{A+BP} u_c + \frac{B}{A+BP} v$$

Effekten av störningar kan minskas genom att öka P.

"Macron FIG2" visar utsignalen för en störning. Störningen har "producerats" genom att skicka ett diskret tid vitt brus (tid mellan ändringar $dt = 0.05$) genom ett filter med formen

$$H_{noise}(s) = \frac{\omega_n^2}{s^2 + 2\xi_n\omega_n s + \omega_n^2}$$

Variansen hos det vita bruset är 1, genom att välja $\xi_n$ och $\omega_n$ kan man få brus med olika egenskaper (man kan till ex. få en nästan periodisk utseende genom att välja $\xi_n$ litet). Variansen av det filtrerade bruset får man uppskatta genom att sampla utsignalen och beräkna variansen hos denna. Man kan också addera en konstant för att få ett medelvärde skilt från noll.

b) MRAS och STR

Analys av effekten av störningar på MRAS-systemet blir mycket svårt att genomföra eftersom störningen kommer in på ett mycket komplicerat sätt, det samma gäller för STR.

Med hjälp av simulering kan man undersöka hur systemet beter sig för olika slags störningar. Tyvärr har författaren begränsade möjligheter att rita resultaten för hand, men på filen RIGEL:: USER1:<EF4CD> finns Simnon macros för olika fall som tillgängliga utifrån. (MACRO FIG1 ... FIG12).

I stora drag kan man säga följande:

1) Både MRS och STR är relativ okänsliga för bredbandigt brus (medelv. = 0)

2) Konvergensen försämras kraftigt för både konstanta laststörningar eller störningar som är nästan periodiska.

Resultat.



Analys och implementering av tre olika metoder för att reglera ett andra ordningens system med varierande egenskaper har utförts, metoderna är

a) Återkoppling med hög förstärkning

b) MRAS ( modifierad MIT-regeln)

c) STR ( indirekt självinställande regulator)

Resultatet visar att a) är att föredra, om den är möjligt att implementera (stabilt slutet system), fördelarna är enkelheten och garanterad stabilitet, b) och c) däremot är mer komplexa, stabiliteten kan ej garanteras men har fördelen att vara mer flexibla om processen är okänd.

Författaren själv föredrar lösning c), b) har nackdelen att vara mycket känslig i starten och kan ge upphov till mycket stora översvängar i utsignalen (icke önskvärt).

Simnon's filer använda för simuleringen.

```
MACRO MNOISE
LET N.NOISE1=1
LET NODD.NOISE1=12347
SYST NMRAS  NGEN  NOISE1 MNCON
PAR DT:0.05
END
```

```
CONTINUOUS SYSTEM MRAS
STATE X1 X2 XM1 XM2 XU1 XU2 XU3 XY1 XY2 XY3 XUC1 XUC2 XUC3 XR1 XR2
STATE R1 S0 S1 T0 T1
DER DX1 DX2 DXM1 DXM2 DXU1 DXU2 DXU3 DXY1 DXY2 DXY3 DXUC1 DXUC2 DXUC3
DER DXR1 DXR2 DR1 DS0 DS1 DT0 DT1
TIME T

"PROCESS A(S)Y=B(S)U
"A(S)=S^2+A2*S+A1
"B(S)=K
DX1=-A2*X1+X2
DX2=-A1*X1+K*U
Y=X1
K:0.5
A1:1
A2:1

"MODEL Am(s)Ym=Bm(s)Uc
"Am(s)=s^2+1.4s+1
"Bm(s)=1
DXM1=-1.4*XM1+XM2
DXM2=-XM1+UC
YM=XM1

"REF Uc
UC=IF MOD(T,PER)<PER/2 THEN HIGH ELSE LOW
PER:40
LOW:-1
HIGH:1

"Ao(s)=s+w
"AoAm(s)=s^3+(w+1.4)*s^2+(1.4w+1)*s+w
"Phi=1/AoAm*Ä -u -s*y -y s*uc uc Å

DXU1=-(W+1.4)*XU1-(1.4*W+1)*XU2-W*XU3+U
DXU2=XU1
DXU3=XU2
PHI1=-XU3


DXY1=-(W+1.4)*XY1-(1.4*W+1)*XY2-W*XY3+Y
DXY2=XY1
DXY3=XY2
PHI2=-XY2
PHI3=-XY3
```

```
DXUC1=-(W+1.4)*XUC1-(1.4*W+1)*XUC2-W*XUC3+UC
DXUC2=XUC1
DXUC3=XUC2
PHI4=XUC2
PHI5=XUC3

W:2

"UPPDATE

E=Y-YM
DEN=PHI1*PHI1+PHI2*PHI2+PHI3*PHI3+PHI4*PHI4+PHI5*PHI5+ALPHA

DR1=-GAMMA*E*PHI1/DEN
DS0=-GAMMA*E*PHI2/DEN
DS1=-GAMMA*E*PHI3/DEN
DT0=-GAMMA*E*PHI4/DEN
DT1=-GAMMA*E*PHI5/DEN


ALPHA:0.1
GAMMA:1

"REGULATOR
"R(s)=s+r1
"S(s)=s0*s+s1
"T(s)=t0*s+t1

DXR1=-R1*XR1+(T1-T0*R1)*UC
DXR2=-R1*XR2+(S1-S0*R1)*Y
U=T0*UC+XR1-(S0*Y+XR2)

END
```

2/2

```
CONNECTING SYSTEM NCON
UNÄNGENÅ=E1ÄNOISE1Å
V=CNÄNGENÅ
END
```

```
CONTINUOUS SYSTEM NGEN
STATE NX1 NX2
DER DNX1 DNX2
INPUT UN
OUTPUT CN

Z1=2*EN*WN
Z2=WN*WN

DNX1=-Z1*NX1+NX2
DNX2=-Z2*NX1+Z2*UN
CN=SCALE*NX1+OFFSET

EN:0.7
WN:10
SCALE:1
OFFSET:0

END
```

```
CONTINUOUS SYSTEM NMRAS
INPUT V
STATE X1 X2 XM1 XM2 XU1 XU2 XU3 XY1 XY2 XY3 XUC1 XUC2 XUC3 XR1 XR2
STATE R1 S0 S1 T0 T1
DER DX1 DX2 DXM1 DXM2 DXU1 DXU2 DXU3 DXY1 DXY2 DXY3 DXUC1 DXUC2 DXUC3
DER DXR1 DXR2 DR1 DS0 DS1 DT0 DT1
TIME T

"PROCESS A(S)Y=B(S)U
"A(S)=S^2+A2*S+A1
"B(S)=K
DX1=-A2*X1+X2
DX2=-A1*X1+K*U+V
Y=X1
K:0.5
A1:1
A2:1

"MODEL Am(s)Ym=Bm(s)Uc
"Am(s)=s^2+1.4s+1
"Bm(s)=1
DXM1=-1.4*XM1+XM2
DXM2=-XM1+UC
YM=XM1

"REF Uc
UC=IF MOD(T,PER)<PER/2 THEN HIGH ELSE LOW
PER:40
LOW:-1
HIGH:1

"Ao(s)=s+w
"AoAm(s)=s^3+(w+1.4)*s^2+(1.4w+1)*s+w
"Phi=1/AoAm*Ä -u -s*y -y s*uc uc Å

DXU1=-(W+1.4)*XU1-(1.4*W+1)*XU2-W*XU3+U
DXU2=XU1
DXU3=XU2
PHI1=-XU3


DXY1=-(W+1.4)*XY1-(1.4*W+1)*XY2-W*XY3+Y
DXY2=XY1
DXY3=XY2
PHI2=-XY2
```

```
PHI3=-XY3

DXUC1=-(W+1.4)*XUC1-(1.4*W+1)*XUC2-W*XUC3+UC
DXUC2=XUC1
DXUC3=XUC2
PHI4=XUC2
PHI5=XUC3

W:2

"UPPDATE

E=Y-YM
DEN=PHI1*PHI1+PHI2*PHI2+PHI3*PHI3+PHI4*PHI4+PHI5*PHI5+ALPHA

DR1=-GAMMA*E*PHI1/DEN
DSU=-GAMMA*E*PHI2/DEN
DS1=-GAMMA*E*PHI3/DEN
DIO=-GAMMA*E*PHI4/DEN
DT1=-GAMMA*E*PHI5/DEN


ALPHA:0.1
GAMMA:1

"REGULATOR
"R(s)=s+r1
"S(s)=s0*s+s1
"T(s)=t0*s+t1

DXR1=-R1*XR1+(T1-T0*R1)*UC
DXR2=-R1*XR2+(S1-S0*R1)*Y
U=T0*UC+XR1-(S0*Y+XR2)

END
```

```
MACRO NOISE
LET N.NOISE1=1
LET NODD.NOISE1=12347
SYST NGEN  NOISE1 NCON
PAR DT:0.05
END
```

```
CONTINUOUS SYSTEM NPREG
INPUT V
STATE X1 X2 XM1 XM2
DER DX1 DX2 DXM1 DXM2
TIME T

"Process A(s)y=B(s)u
" A(s)=s^2+a2*s+a1
" B(s)=K

DX1=-A2*X1+X2
DX2=-A1*X1+K*U+V
Y=X1

"Model Am(s)ym=Bm(s)ur
" Am(s)=s^2+1.4*s+1
" Bm(s)=1

DXM1=-1.4*XM1+XM2
DXM2=-XM1+UR
YM=XM1

"Regulator
" u=p(ym-y)

U=P*(YM-Y)


P:1
A1:1
A2:1
K:0.5

"REF
UR=IF MOD(T,PER)>PER/2 THEN LOW ELSE HIGH

PER:40
LOW:-1
HIGH:1


END
```

```
CONTINUOUS SYSTEM NSTR

INPUT V

STATE X1 X2 XR1 XR2 FU1 FU2 FU3 FY1 FY2 FY3 TH1 TH2 TH3
STATE P11 P12 P13 P21 P22 P23 P31 P32 P33

DER DX1 DX2 DXR1 DXR2 DFU1 DFU2 DFU3 DFY1 DFY2 DFY3 DTH1 DTH2 DTH3
DER DP11 DP12 DP13 DP21 DP22 DP23 DP31 DP32 DP33

TIME T

"PROCESS A(s)y=B(s)u
" A(s)=s^2 + a2*s + a1
" B(s)=k

DX1=-A2*X1+X2
DX2=-A1*X1+K*U+V
Y=X1

K:0.5
A2:1
A1:1

"ESTIMATOR
" TH=ÄTH1 TH2 TH3Ä^T
" PHI=ÄPHI1 PHI2 PHI3Ä^T = Ä -S*F(S)Y -F(S)*Y F(S)*U Ä^T
" P = Ä P11 P12 P13 ; P21 P22 P23 ; P31 P32 P33 Ä
" F(S)=WF*WF/ ( S^2 + 2*E*WF *S + WF*WF )*W2/S+W2
" YF =F(S)*Y*s^2
" UF =F(S)*U
" EF=YF- PHI^T*TH
" D(TH)/DT= P*TH*EF
" D(P)/DT=ALPHA*P - P*PHI*PHI^T*P

P1=2*E*WF+W2
P2=WF*WF+2*E*WF*W2
P3=WF*WF*W2

DFY1=-P1*FY1-P2*FY2-P3*FY3+Y
DFY2=FY1
DFY3=FY2
PHI1=-FY2*P3
PHI2=-FY3*P3
YF=FY1*P3
```

```
DFU1=-P1*FU1-P2*FU2-P3*FU3+U
DFU2=FU1
DFU3=FU2
PHI3=FU3*P3

E:0.7
WF:5
W2:10

EF=YF- (PHI1*TH1 + PHI2*TH2 +PHI3*TH3)

DTH1=(P11*PHI1+P12*PHI2+P13*PHI3)*EF
DTH2=(P21*PHI1+P22*PHI2+P23*PHI3)*EF
DTH3=(P31*PHI1+P32*PHI2+P33*PHI3)*EF

Z11=P11*PHI1*PHI1+P12*PHI2*PHI1+P13*PHI3*PHI1
Z12=P11*PHI1*PHI2+P12*PHI2*PHI2+P13*PHI3*PHI2
Z13=P11*PHI1*PHI3+P12*PHI2*PHI3+P13*PHI3*PHI3

Z21=P21*PHI1*PHI1+P22*PHI2*PHI1+P23*PHI3*PHI1
Z22=P21*PHI1*PHI2+P22*PHI2*PHI2+P23*PHI3*PHI2
Z23=P21*PHI1*PHI3+P22*PHI2*PHI3+P23*PHI3*PHI3

Z31=P31*PHI1*PHI1+P32*PHI2*PHI1+P33*PHI3*PHI1
Z32=P31*PHI1*PHI2+P32*PHI2*PHI2+P33*PHI3*PHI2
Z33=P31*PHI1*PHI3+P32*PHI2*PHI3+P33*PHI3*PHI3

DP11=ALPHA*P11-(Z11*P11+Z12*P21+Z13*P31)
DP12=ALPHA*P12-(Z11*P12+Z12*P22+Z13*P32)
DP13=ALPHA*P13-(Z11*P13+Z12*P23+Z13*P33)

DP21=ALPHA*P21-(Z21*P11+Z22*P21+Z23*P31)
DP22=ALPHA*P22-(Z21*P12+Z22*P22+Z23*P32)
DP23=ALPHA*P23-(Z21*P13+Z22*P23+Z23*P33)

DP31=ALPHA*P31-(Z31*P11+Z32*P21+Z33*P31)
DP32=ALPHA*P32-(Z31*P12+Z32*P22+Z33*P32)
DP33=ALPHA*P33-(Z31*P13+Z32*P23+Z33*P33)
```

```
P11:1
P12:0
P13:0
P21:0
P22:1
P23:0
P31:0
P32:0
P33:1
ALPHA:0.01

"REGULATOR
" R(s)u=T(s)uc-S(s)y
" Ao(s)=s+w1
" Am(s)=s^2+1.4*s+1
" Bm(s)=1
" R(s)= s + r1
" S(s)= s0*s+s1
" T(s)= t0*s+t1

"R1=W1+1.4-A2
"S0=(1.4*W1+1-A1-A2*R1)/K
"S1=(W1-A1*R1)/K
"T0=1/K
"T1=W1/K

K1=TH3/(TH3*TH3+EPS)
R1= W1+1.4-TH1
S0= (1.4*W1+1-TH2-TH1*R1)*K1
S1= (W1-TH2*R1)*K1
T0= K1
T1= K1*W1


W1:2
EPS:0.01

DXR1=-R1*XR1+(T1-T0*R1)*UC
DXR2=-R1*XR2+(S1-S0*R1)*Y

U=T0*UC+XR1-(S0*Y+XR2)+PE

"REFERENCE
UC=IF MOD(T,PER)>PER/2 THEN LOW ELSE HIGH
UC1=IF MOD(T,PER1)>PER1/2 THEN LOW1 ELSE HIGH1
```

```
PE=IF T<TRY THEN  UC1 ELSE 0

PER:40
LOW:-1
HIGH:1

PER1:2
LOW1:0.1
HIGH1:0.1
TRY:10

END
```

```
CONNECTING SYSTEM PNCON
UNÄNGENÅ=E1ÄNOISE1Å
VÄNPREGÅ=CNÄNGENÅ
END
```

306.

```
MACRO PNOISE
LET N.NOISE1=1
LET NODD.NOISE1=12347
SYST NPREG NGEN  NOISE1 PNCON
PAR DT:0.05
END
```

```
CONTINUOUS SYSTEM PREG
STATE X1 X2 XM1 XM2
DER DX1 DX2 DXM1 DXM2
TIME T

"Process A(s)y=B(s)u
" A(s)=s^2+a2*s+a1
" B(s)=K

DX1=-A2*X1+X2
DX2=-A1*X1+K*U
Y=X1

"Model Am(s)ym=Bm(s)ur
" Am(s)=s^2+1.4*s+1
" Bm(s)=1

DXM1=-1.4*XM1+XM2
DXM2=-XM1+UR
YM=XM1

"Regulator
" u=p(ym-y)

U=P*(YM-Y)


P:1
A1:1
A2:1
K:0.5

"REF

UR=IF MOD(T,PER)>PER/2 THEN LOW ELSE HIGH

PER:40
LOW:-1
HIGH:1

END
```

```
MACRO RESET
LET NODD.NOISE1=12347
END
```

```
CONNECTING SYSTEM SNCON
UNÄNGENA=E1ÄNOISE1A
VÄNSTRA=CNÄNGENA
END
```

```
MACRO SNOISE
LET N.NOISE1=1
LET NODD.NOISE1=12347
SYST NSTR  NGEN  NOISE1 SNCON
PAR DT:0.05
END
```

```
CONTINUOUS SYSTEM STR

STATE X1 X2 XR1 XR2 FU1 FU2 FU3 FY1 FY2 FY3 TH1 TH2 TH3
STATE P11 P12 P13 P21 P22 P23 P31 P32 P33

DER DX1 DX2 DXR1 DXR2 DFU1 DFU2 DFU3 DFY1 DFY2 DFY3 DTH1 DTH2 DTH3
DER DP11 DP12 DP13 DP21 DP22 DP23 DP31 DP32 DP33

TIME T

"PROCESS A(s)y=B(s)u
" A(s)=s^2 + a2*s + a1
" B(s)=k

DX1=-A2*X1+X2
DX2=-A1*X1+K*U
Y=X1

K:0.5
A2:1
A1:1

"ESTIMATOR
" TH=ÄTH1 TH2 TH3Å^T
" PHI=ÄPHI1 PHI2 PHI3Å^T = Ä -S*F(S)Y -F(S)*Y F(S)*U Å^T
" P = Ä P11 P12 P13 ; P21 P22 P23 ; P31 P32 P33 Å
" F(S)=WF*WF/ ( S^2 + 2*E*WF *S + WF*WF )*W2/S+W2
" YF =F(S)*Y*S^2
" UF =F(S)*U
" EF=YF- PHI^T*TH
" D(TH)/DT= P*TH*EF
" D(P)/DT=ALPHA*P - P*PHI*PHI^T*P

P1=2*E*WF+W2
P2=WF*WF+2*E*WF*W2
P3=WF*WF*W2

DFY1=-P1*FY1-P2*FY2-P3*FY3+Y
DFY2=FY1
DFY3=FY2
PHI1=-FY2*P3
PHI2=-FY3*P3
YF=FY1*P3

DFU1=-P1*FU1-P2*FU2-P3*FU3+U
```

```
DFU2=FU1
DFU3=FU2
PHI3=FU3*P3

E:0.7
WF:5
W2:10

EF=YF- (PHI1*TH1 + PHI2*TH2 +PHI3*TH3)

DTH1=(P11*PHI1+P12*PHI2+P13*PHI3)*EF
DTH2=(P21*PHI1+P22*PHI2+P23*PHI3)*EF
DTH3=(P31*PHI1+P32*PHI2+P33*PHI3)*EF

Z11=P11*PHI1*PHI1+P12*PHI2*PHI1+P13*PHI3*PHI1
Z12=P11*PHI1*PHI2+P12*PHI2*PHI2+P13*PHI3*PHI2
Z13=P11*PHI1*PHI3+P12*PHI2*PHI3+P13*PHI3*PHI3

Z21=P21*PHI1*PHI1+P22*PHI2*PHI1+P23*PHI3*PHI1
Z22=P21*PHI1*PHI2+P22*PHI2*PHI2+P23*PHI3*PHI2
Z23=P21*PHI1*PHI3+P22*PHI2*PHI3+P23*PHI3*PHI3

Z31=P31*PHI1*PHI1+P32*PHI2*PHI1+P33*PHI3*PHI1
Z32=P31*PHI1*PHI2+P32*PHI2*PHI2+P33*PHI3*PHI2
Z33=P31*PHI1*PHI3+P32*PHI2*PHI3+P33*PHI3*PHI3

DP11=ALPHA*P11-(Z11*P11+Z12*P21+Z13*P31)
DP12=ALPHA*P12-(Z11*P12+Z12*P22+Z13*P32)
DP13=ALPHA*P13-(Z11*P13+Z12*P23+Z13*P33)

DP21=ALPHA*P21-(Z21*P11+Z22*P21+Z23*P31)
DP22=ALPHA*P22-(Z21*P12+Z22*P22+Z23*P32)
DP23=ALPHA*P23-(Z21*P13+Z22*P23+Z23*P33)

DP31=ALPHA*P31-(Z31*P11+Z32*P21+Z33*P31)
DP32=ALPHA*P32-(Z31*P12+Z32*P22+Z33*P32)
DP33=ALPHA*P33-(Z31*P13+Z32*P23+Z33*P33)

P11:1
P12:0
```

```
P13:0
P21:0
P22:1
P23:0
P31:0
P32:0
P33:1
ALPHA:0.01

"REGULATOR
"  R(s)u=T(s)uc-S(s)y
"  Ao(s)=s+w1
"  Am(s)=s^2+1.4*s+1
"  Bm(s)=1
"  R(s)=  s  + r1
"  S(s)=  s0*s+s1
"  T(s)=  t0*s+t1

"R1=W1+1.4-A2
"S0=(1.4*W1+1-A1-A2*R1)/K
"S1=(W1-A1*R1)/K
"T0=1/K
"T1=W1/K

K1=TH3/(TH3*TH3+EPS)
R1=  W1+1.4-TH1
S0=  (1.4*W1+1-TH2-TH1*R1)*K1
S1=  (W1-TH2*R1)*K1
T0=  K1
T1=  K1*W1


W1:2
EPS:0.01

DXR1=-R1*XR1+(T1-T0*R1)*UC
DXR2=-R1*XR2+(S1-S0*R1)*Y

U=T0*UC+XR1-(S0*Y+XR2)+PE

"REFERENCE
UC=IF MOD(T,PER)>PER/2 THEN LOW ELSE HIGH
UC1=IF MOD(T,PER1)>PER1/2 THEN LOW1 ELSE HIGH1
PE=IF T<TRY THEN  UC1 ELSE 0
```

```
PER:40
LOW:-1
HIGH:1

PER1:2
LOW1:0.1
HIGH1:0.1
TRY:10

END
```