

SIMULERING AV EN HYBRID SJÄLVINSTÄLLANDE REGULATOR

MATS JÖNSSON

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA
SEPTEMBER 1985

LUND INSTITUTE OF TECHNOLOGY DEPARTMENT OF AUTOMATIC CONTROL Box 118 S 221 00 Lund Sweden		Document name Master thesis	
		Date of issue September 1985	
		Document number Coden:LUTFD2/(TFRT-5334)/1-051/(1985)	
Author(s) Mats Jönsson		Supervisor Björn Wittenmark	
		Sponsoring organization	
Title and subtitle Simulering av en hybrid självinställande regulator. (Simulation of a hybrid self-tuning control.)			
Abstract Hybrid self-tuning controllers are introduced which combine two sample rates to give both discrete-time and continuous-time features. The limiting properties of such algorithms, as the sample rates increases, are shown to correspond to a continuous-time control law. Undesirable intersample behavior associated with the discrete-time control of continuous systems with high pole-zero excess is thus avoided. The hybrid self-tuning controller can very well handle non-minimumphase systems. Several process have been examined, which illustrated the features of the hybrid self-tuning controllers.			
Key words			
Classification system and/or index terms (if any)			
Supplementary bibliographical information			
ISSN and key title			ISBN
Language Swedish	Number of pages 51	Recipient's notes	
Security classification			

DOKUMENTDATABLAD RT 3/81

Distribution: 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.

ABSTRACT

Hybrid self-tuning controllers are introduced which combine two sample rates to give both discrete-time and continuous-time features. The limiting properties of such algorithms, as the sample rates increases, are shown to correspond to a continuous-time control law. Undesirable intersample behavior associated with the discrete-time control of continuous systems with high pole-zero excess is thus avoided.

The hybrid self-tuning controller can very well handle non-minimumphase systems.

Several process have been examine, which illustrated the features of the hybrid self-tuning controllers.

INNEHALLSFÖRTECKNING

	Förord	1
	Lista över förekommande principiella symboler	2
1.	Självinställande hybridregulator	3
	Introduktion av hybridregulatorn	4
	Hybridprediktor	5
	Blockschema	6
	Sammanfattning	6
2.	Beskrivning av hybridregulatorn	7
2.1.	Matematisk härledning	7
	Allmänna gradtal	8
2.2.	Blockbeskrivning	9
2.2.1.	Filter	9
2.2.2.	Regulator	10
	Designmetod 1, algoritm 1	11
	Designmetod 2, algoritm 2	12
2.2.3.	Estimator	13
	Rekursiv minsta kvadrat algoritm	13
3.	Resultat av simuleringar	14
3.1.	Process s^{-3} med designmetod 2	15
	Val av filtrets tidskonstanter	16
	Val av Q	24
	Initialvärde, begränsningar	25
3.2.	Process s^{-3} med designmetod 1	26
3.3.	Process $(1-s)/(s(s+1))$	29
	Hur kan h väljas	30
3.4.	Diskret version av regulatorn	33
	Slutsatser	36
	Appendix A	37
	Appendix B	41
	Appendix C	44
	Appendix D	47
	Litteraturförteckning	50

FÖRORD

Examensarbetet är utfört och gjort för Institutionen för Reglerteknik i Lund under sommarmånaderna 1985. Examensarbetet har gått ut på att studera och urvärdera en adaptiv hybridregulator. Dess egenskaper är redovisade i denna rapport.

Jag vill slutligen framföra ett tack till Björn Wittenmark som har varit min handledare och gjort detta examensarbetet möjligt.

LISTA ÖVER FÖREKOMMANDE PRINCIPIELLA SYMBOLER

$A(z^{-1}) B(z^{-1}) C(z^{-1})$:polynom som anges till utsignal, styrsignal och brussignal.
$E(z^{-1}) F(z^{-1}) G(z^{-1})$:polynom som anges till designmodellen.
$u(t) y(t) w(t)$:systemets styrsignal, utsignal och referenssignal.
$\Psi(t)$:systemets brussignal.
z^{-1}	:bakåtoperatoren.
λ	:glömskefaktor.
θ	:parametervektor.
ϕ	:datavektor.
Δ, h	:samplingsintervall.
grad	:grad på polynom.
ϕ	:pseudoprocess till utsignal.
ϕ^*	:skattat värde till ϕ .
$(i+1 i)$:argumentet utläses: prediktionens värde vid tidpunkten $(i+1)$ som baseras på den information vid tidpunkten (i) .

1. SJÄLVINSTÄLLANDE HYBRIDREGULATOR

Detta kapitlet ger en introduktion av hybrid regulatorns egenskaper och möjligheter.

Anledningen till intresset av adaptiva (självinställande) regulatorer beror på att dels är ofta systemets dynamik okänd och dels är systemets dynamik inte alltid konstant i tiden. För att kunna lösa detta gör man med hjälp av olika identifieringsmetoder en identifiering av systemet och anpassar därefter parametrar till regulatorn. Det finns en mängd olika typer av adaptiva regulatorer men de bygger dock på samma princip.

Aström och Wittenmark publicerade 1973 en artikel om självinställande regulatorer. Där visades hur en enkel identifiering och regulatordesign kan kombineras till en självinställande regulator som har goda transienta och asymptotiska egenskaper. Deras självinställande regulator fungerade endast om det reglerade systemet (i diskret tid) var minimumfas. Olika försök har gjorts för att utvidga klassen av regulatorer. I denna rapport studeras en hybridregulator föreslagen av Gawthrop¹.

Introduktion av hybridregulatorn

Traditionella adaptiva regulatorer har varit helt baserade på antingen kontinuerlig form eller diskret form. I denna rapport behandlas en hybrid regulator som består dels av en kontinuerlig del och dels av en diskret del. Den anses kunna kombinera fördelarna från respektive form. Nackdelen med den tids-kontinuerliga algoritmen är att den implementeras med analoga kretsar vilket gör att den är dåligt anpassad till den digitala teknologin. Den beskriver dock verkligheten bättre, eftersom de flesta fenomenen inom naturvetenskapen identifieras just med tids-kontinuerliga system.

En fördel med tids-diskret metod, är att den passar dagens mikroprocesser utmärkt. Men eftersom deras design skiljer sig från den tids-kontinuerliga ger detta upphov till icke önskade fenomen beroende på samplingen. Detta diskuteras mer ingående längre fram.

Hybriden kan beskrivas av

- a) filter (kontinuerligt)
- b) estimator (diskret)
- c) regulator med design (diskret)

Typiskt för

- a) är att den består av en linjäröverföring
- b) är att den innehåller relativt komplicerade beräkningar
- c) är att den innehåller en tidsvarierande överföringsfunktion.

Hybrid prediktor

Det är viktigt att inte förlora insikten över det underliggande kontinuerliga systemet, då man använder tids-diskreta algoritmer som implementeras digitalt. Vid övergång från tids-kontinuerlig form till tids-diskret form är det viktigt att veta hur nollställe och poler förhåller sig efter samplingen. I denna rapport har samplingen skett med "zero order hold".

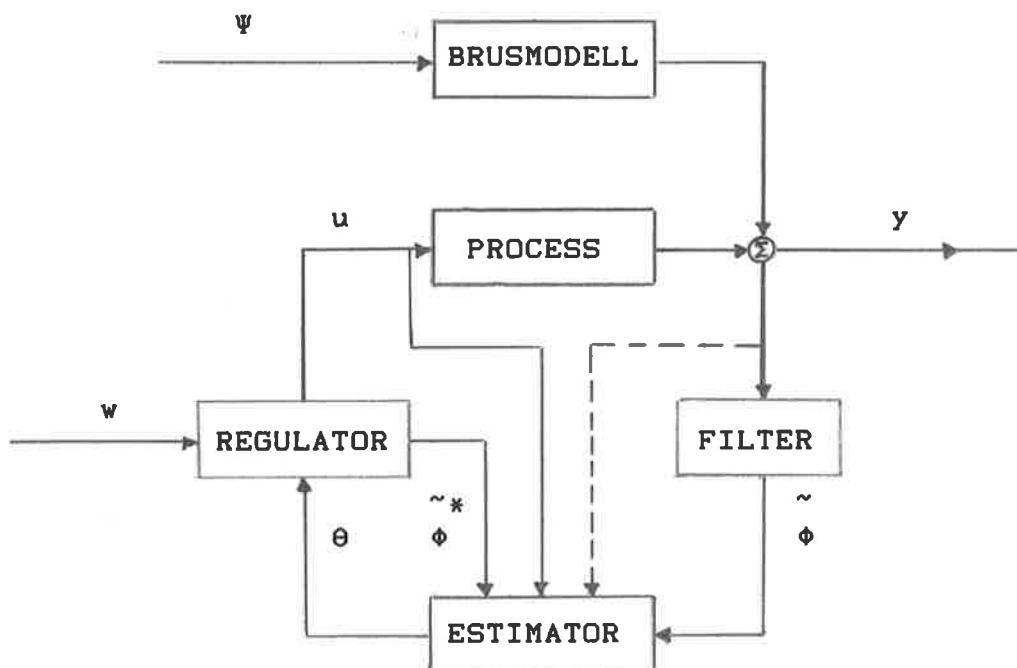
Om ett system med gradtalet n samplas med "zero order hold" på insignalen, flyttas de kontinuerliga nollställena mot de diskreta nollställena och förbättras ju mer samplingsintervallet avtar. Detta motsvaras av en samplad version av systemet $1/s^n$. För $n > 2$ ger detta nollställe på eller utanför enhetscirkeln. Därför är det möjligt att styrlagen kan ge upphov till olämpliga och otillfredsställande styrsignaler eftersom denna bygger på diskret form. Om det samplade systemet har nollställe innanför enhetscirkeln, har det visat sig att man med väl valda diskret-tids modeller kan undvika denna olämpliga signal. Den är dock inte godtagbar om det kontinuerliga systemet har en alltför stor skillnad mellan gradtalen på poler och nollställe.

Hybrid regulatorn påstås ha de önskvärda egenskaperna när det gäller att få en diskret-tids prediktors stabilitet för det slutna systemet att tendera till det kontinuerliga, då samplingsintervallet avtar.

Det finns ett problem med den kontinuerliga prediktorn som måste övervinnas om en motsvarande självinställande metod skall vara möjlig. Problemmet är att ϕ inte alltid är realiserbar utan ren derivering. Detta övervinns i hybriden genom att en naturlig tidsfördröjning kommer in i och med användandet av diskret tid i prediktions kontrollen. Tidsfördröjningen motsvaras av ett samplingsintervall.

Blockschema

Hybridregulatorn kan beskrivas enligt nedanstående blockschema.



Blockscema för hybridregulatorn där designmetod 1 representeras med heldragen linje och designmetod 2 med heldragen + streckad linje.

Sammanfattning

Hybridregulatorn är en kombination av tids-diskret form och tids-kontinuerlig form. Den anses kunna kombinera fördelarna från dessa två olika tidssystem. Den främsta förmåga är dess möjlighet att reglera icke-minimumfas system. Hybridregulatorn påstås ha de önskvärda egenskaperna när det gäller att få en diskret-tids prediktors stabilitet för det slutna systemet att tendera till det kontinuerliga, då samplingsintervallet avtar.

2. BESKRIVNING AV HYBRIDREGULATORN

2.1. Matematisk härledning

För att modulera stokastiska processer använder man ARMA-processen som beskrivs enligt

$$A(s)y(t) = B(s)u(t) + C(s)\Psi(t) \quad (1)$$

I denna rapport har man bortsett från det vita bruset och bruset överhuvudtaget. Men för härledningens fullständighet, har bruspolynomen ändå medtagits.

Beräkna pseudoprocessen ϕ

$$\phi(t) = Py(t) = \frac{PB}{A} u(t) + \frac{PC}{A} \Psi(t) \quad (2)$$

processen ϕ existerar inte i någon praktisk mening om inte CP/A är proper, vilket är analogt med det som diskuterades under rubriken hybridprediktor. Lös därför identiteten

$$\frac{CP}{A} = E + \frac{F}{A} \quad (3) ; \text{ där } G = EB \quad (4)$$

vilket överensstämmer med den identiteten som bl a har diskuterats av Aström och Wittenmark².

Finn en process ϕ^* med hjälp av ekvationerna (1), (2) och (4) insatta, således blir

$$\phi^*(t) = \frac{F}{A} \Psi(t) + \frac{PB}{A} u(t) = \frac{F}{C} y(t) + \frac{G}{C} u(t) \quad (5)$$

införandet av ett polynom D som beskrivs senare ger en ny process

$$\tilde{\phi}(t) = \frac{1}{D} \phi(t) = \frac{P}{D} y(t) \quad (6)$$

Felet mellan det verkliga och det predikterade värdet representeras av

$$e(t) = \phi(t) - \phi^*(t) = E\Psi(t) \quad (7)$$

Felet beror alltså ej av $u(t)$ eller $y(t)$.

Styrlagen på diskret form som Gawthrop har beskrivit den, blir

$$u'(i) = \frac{1}{Q'(s')} [\phi^*(i+1|i) - w(i)] \quad (8)$$

där s' ges av Eulers theorem

$$s' = \frac{1-q^{-1}}{\Delta} \quad (9)$$

$Q(s)$ är en skalär och beskrivs framöver. Den kan också vara $(1/Q)=PI$ -reg.

Slutligen fås

$$y(t) = \frac{EB + QC}{PB + QA} \psi(t) + \frac{B}{PB + QA} w(t) \quad (10)$$

eller

$$u(t) = \frac{-F}{PB + QA} \psi(t) + \frac{A}{PB + QA} w(t) \quad (11)$$

Med de ovanstående formlerna ser man att man har möjlighet att påverka polplaceringen genom olika val på polynomen P och Q .

Allmänna gradtal på polynomen

$$\text{grad}(P) \leq \text{grad}(A) - \text{grad}(B)$$

$$\text{grad}(D) \geq \text{grad}(P)$$

$$\text{grad}(C) = \text{grad}(A) - 1$$

$$\text{grad}(F) = \text{grad}(A) - 1$$

$$\text{grad}(E) = \text{grad}(C) + \text{grad}(P) - \text{grad}(A)$$

$$\text{grad}(G) \leq \text{grad}(C)$$

2.2. Blockbeskrivning

I de närmaste underrubrikerna ges en beskrivning av de ingående delarna i hybridregulatorn som delas upp i filter, regulator och estimator.

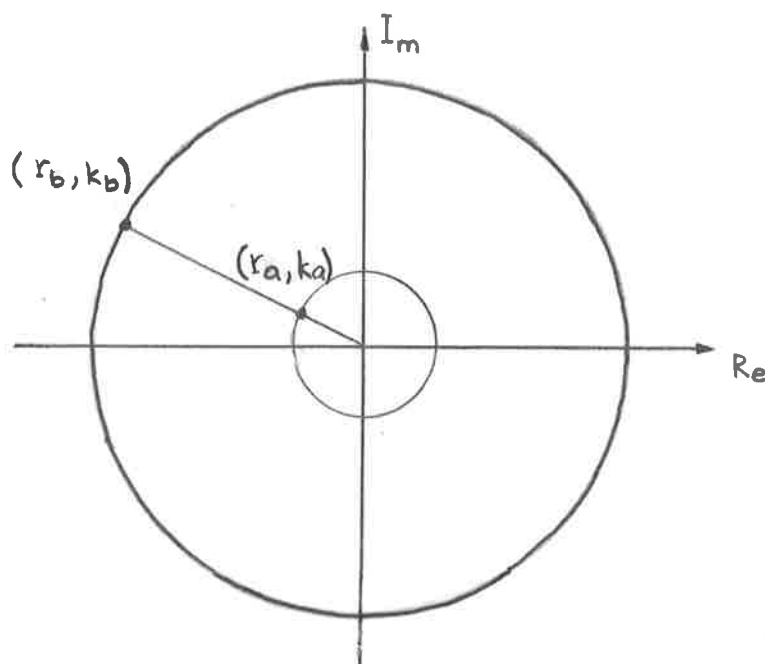
2.2.1. Filter

Det digitala filtret består av ett P-polynom och ett D-polynom. Anledningen med att använda ett fasavancerande element P, är att den ger en dämpning utan en allt för överdriven högfrekvent brusförstärkning.

Med hjälp av filtret undertrycker man också oönskade frekvenser av $\Phi(E\Psi)$. Vilka frekvenser som filtreras bort bestäms av D-polynomet som är filtrets karakteristiska ekvation.

Med P-och D-polynomen har man möjligheter att ändra polplaceringen enligt ekvationerna (10) och (11) och därmed stabiliteten.

Vid simuleringarna har ett visst förhållande preciserats enligt nedanstående figur.



$$\text{Förhållandet väljs } (r_a + ik_a) = \epsilon^2 (r_b + ik_b)$$

Rötterna som är komplexa förekommer i komplexkonjugerade par, eftersom differentialekvationerna endast innehåller reella rötter.

2.2.2. Regulator

Detta block innehåller dels styrlagen och dels designen. Gawthrop lägger fram två typer av designmetoder i sin artikel, som påminner ganska mycket om varandra. På följande sidor ges en beskrivning av designmetoderna (2 st) och de tillhörande algoritmerna.

Styrlagen fås genom att sätta in någon av designmetoderna i

$$u(i) = \frac{1}{Q} [\phi^*(i+1|i) - w(i)] \quad (12)$$

Styrsignalen var begränsad för att kunna "fånga" in skattningsparametrarna. Det visade sig senare att begränsningarna inte hade någon större påverkan då man hade initialvärde på parametrarna.

Fördröjning på skattningsparametrarna infördes för att undvika algebraiska loopar.

Design metod 1

Enligt formeln får man

$$\tilde{\phi}(t) = \frac{PB}{DA} u'(t) + \frac{PC}{DA} \psi(t) \quad (13)$$

$u'(t)$ är den samplade och approximerade till $u(t)$

Den samplade versionen av ovanstående blir

$$\tilde{\phi}'(i) = \frac{P'B'}{D'A'} u'(i-1) + \frac{P'C'}{D'A'} \psi(i) \quad (14)$$

Observera tidsfördröjningen i $u'(i-1)$ som beror på samplingen.

Prediktionen ges nu av

$$\tilde{\phi}^*(i+1|i) = \frac{G_1}{C_1} u'(i) + \frac{F_1}{C_1} \tilde{\phi}'(i) \quad (15)$$

Algoritm 1

$$\tilde{\phi}^*(i+1|i) = \varphi^T(i)\theta(i) \quad (16)$$

där datavektorn representeras av

$$\varphi^T(i) = (u(i), u(i-1)\dots; \tilde{\phi}(i), \tilde{\phi}(i-1)\dots; \phi^*(i|i-1)\dots;) \quad (17)$$

och parametervektorn av

$$\theta(i) = (g_0, g_1\dots; f_0, f_1\dots; -c_1, -c_2\dots;) \quad (18)$$

Design metod 2

Denna ges på samma sätt som designmetod 1 men man använder processens utsignal istället för filtrets utsignal.

Prediktionen ges av

$$\phi^*(i+1|i) = \frac{F_2}{C_2} y'(i) + \frac{G_2}{C_2} u'(i) \quad (19)$$

eller

$$\phi^*(i+1|i) = Fy'(i) + Gu'(i) - (C-1)\phi^*(i|i-1) \quad (20)$$

Men felet mellan det skattade värdet av ϕ och det verkliga värdet av ϕ (här kan man sätta $\phi = \tilde{\phi}$ då D-polynomet har en liten tidskonstant), ges då av

$$\tilde{\phi}(i) = \phi^*(i|i-1) + e(i) \quad (21)$$

Med ekvation (21) insatt i ekvation (20) ger detta datainformation och lägg märke till att även här kommer det in en tidsfördröjning som beror på samplingen.

Alltså

$$\tilde{\phi}(i) = Fy'(i-1) + Gu'(i-1) - (C-1)\phi^*(i|i-1) \quad (22)$$

Algoritm 2

$$\phi^*(i+1|i) = \phi^T(i)\theta(i) \quad (23)$$

där datavektorn representeras av

$$\phi^T(i) = (u(i), u(i-1) \dots; y(i), y(i-1) \dots; \phi^*(i|i-1) \dots;) \quad (24)$$

och parametervektorn av

$$\theta(i) = (g_0, g_1 \dots; f_0, f_1 \dots; -c_1, -c_2 \dots;) \quad (25)$$

2.2.3. Estimator

För beräkning av parametervektorn θ har ett underprogram (self-tuning regulators, STUREG) använts. I detta subsystem har man tillgång till rekursiv identifikation som tex minsta-kvadrat metod eller maximum-likelihood metod.

I denna rapport har simuleringarna skett med rekursiv minsta-kvadrat algoritm.

Rekursiv minsta-kvadrat algoritm

Man är intresserad av att beräkna en modell av systemet rekursivt efterhand som nya mätningar blir tillgängliga, utan att man skall behöva minnas gamla mätningar. Man vill alltså beräkna en ny skattning θ vid tidpunkten (i) baserat på det gamla θ vid tiden (i-1) och ny mätdatainformation.

Således har man tillgång till någon av designmetoderna och därmed en parametervektor och en datavektor.

Den rekursiva minsta-kvadrat algoritmen kan då skrivas som

$$\theta(i) = \theta(i-1) + K(i) [\tilde{\phi}(i) - \phi^T(i)\theta(i-1)] \quad (26)$$

estimatorns förstärkning ges av

$$K(i) = \frac{P(i-1)\phi(i)}{\lambda + \phi^T(i)P(i-1)\phi(i)} \quad (27)$$

estimatorns felvarians ges av

$$P(i) = P(i-1) - \frac{P(i-1)\phi(i)\phi^T(i)P(i-1)}{1 + \phi^T(i)P(i-1)\phi(i)} \quad (28)$$

λ är en glömskefaktor och är ett mått på hur snabbt data "glöms bort".

3. RESULTAT AV SIMULERINGAR

Flera olika processer har berörts och simulerats. Detta har varit nödvändigt för att kunna utvärdera hybridregulatorns egenskaper. Trippelintegratorn har använts som en referensmodell för att kunna jämföra och se om de påståenden var riktiga som Gawthrop har lagt fram i sin artikel. Det är allmänt känt att en trippelintegrator är svårreglerad och det visades sig också genom att den inte "fångade" in skattningsparametrarna. Därför fick polerna läggas ut till -0.1 som är ett stabilare system. Dessa skattningsparametrar användes sedan som initialvärde till trippelintegratorn.

För att studera dess möjligheter att reglera icke-minfassystem har en diskret version jämförts med en hybrid version av processen $(1-s)/((s(s+1)))$.

I rapporten redovisas simuleringar och resultat av följande processer

1) Hybrid med design 2 av processen

$$\frac{1}{s^3}$$

2) Hybrid med design 1 av samma process som ovanstående.

3) Hybrid med design 2 av processen

$$\frac{1-s}{s(s+1)}$$

4) Diskret version av regulatorn enligt Gawthrop¹

3.1. Process s^{-3} med design 2

Programmen för processen, filtret och regulatorn med dess design 2 återfinns i appendix A.

Det är av stor vikt att veta hur nollställena förhåller sig efter samplingen, ur icke-minimumfas synpunkt. Därför behövs följande beräkningar göras.

Den diskreta överföringsfunktionen är

$$H(z) = C(zI - \Phi)^{-1} \Gamma$$

där

$$\Phi = e^{Ah}$$

och

$$\Gamma = \int_0^h e^{As} ds B$$

A är processens systemmatris.

Det tids-diskreta systemet har nollställe enligt

$$z = -2 + (9 - 6h^3)^{-1/2}$$

Detta ligger utanför enhetscirkeln oavsett vilket samplingsintervall man än väljer. Lägg märke till att då samplingsintervallet är större än 1.14 får man komplexa nollställe i det diskreta tidsplanet.

Val av filtrets tidskonstanter

Graden på polynomen P och D erhålles genom att lösa identiteten enligt ekvation (4), samt m h a de allmänna gradtalen. Detta gav att

$\text{grad}(P)=3$ och $\text{grad}(D)=3$

För att undvika numeriska fel, delades filtret upp i tre element. Ett element kunde då beskrivas som

$$k \frac{s+a}{s+b} \quad ; \quad k = \frac{b}{a} \quad ; \quad a = \epsilon b$$

Vid låga frekvenser vill man ha förstärkningen 1, därav konstanten k.

Genom en mängd simuleringar erhöles vissa rekommenderbara värden på tidskonstanterna som är de inverterade värden till a och b. Simuleringarna för olika tidskonstanter skedde tills dess att signalerna inte längre hade de acceptabla transienta uppföranden.

Rekommenderbara värden uttryckt i b, då samplingsintervallet är 1, är

$$5 < b < 10 \quad ; \quad 0.075 < \epsilon < 0.15$$

I figur 1 visas det exemplet som Gawthrop illustrerade i sin artikel, och i figur 2 visas insvängningsförloppet för parametrarna.

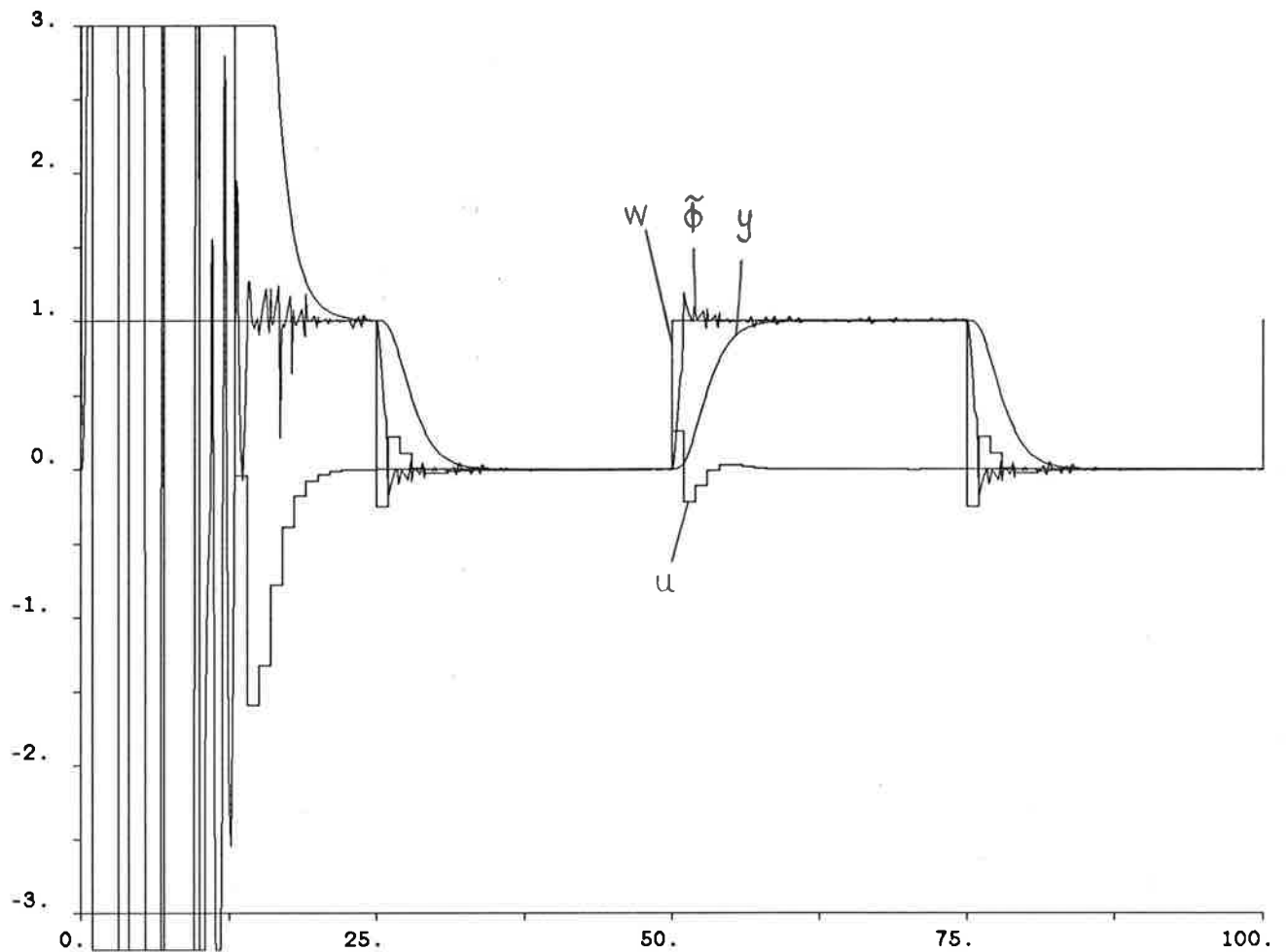


Fig.1. $b=10, \epsilon=0.1$ och $h=1$. Figuren illustrerar hur snabbt ϕ -signalen svänger in till referenssignalen. Styrsignalen reglerar snabbt och koncist. Signalerna kommer att ha svängt in sig efter omkring 25 intervall. Nämnas bör också att insvängningsförloppet blir annorlunda då initialvärden till designpolynomen ändras.

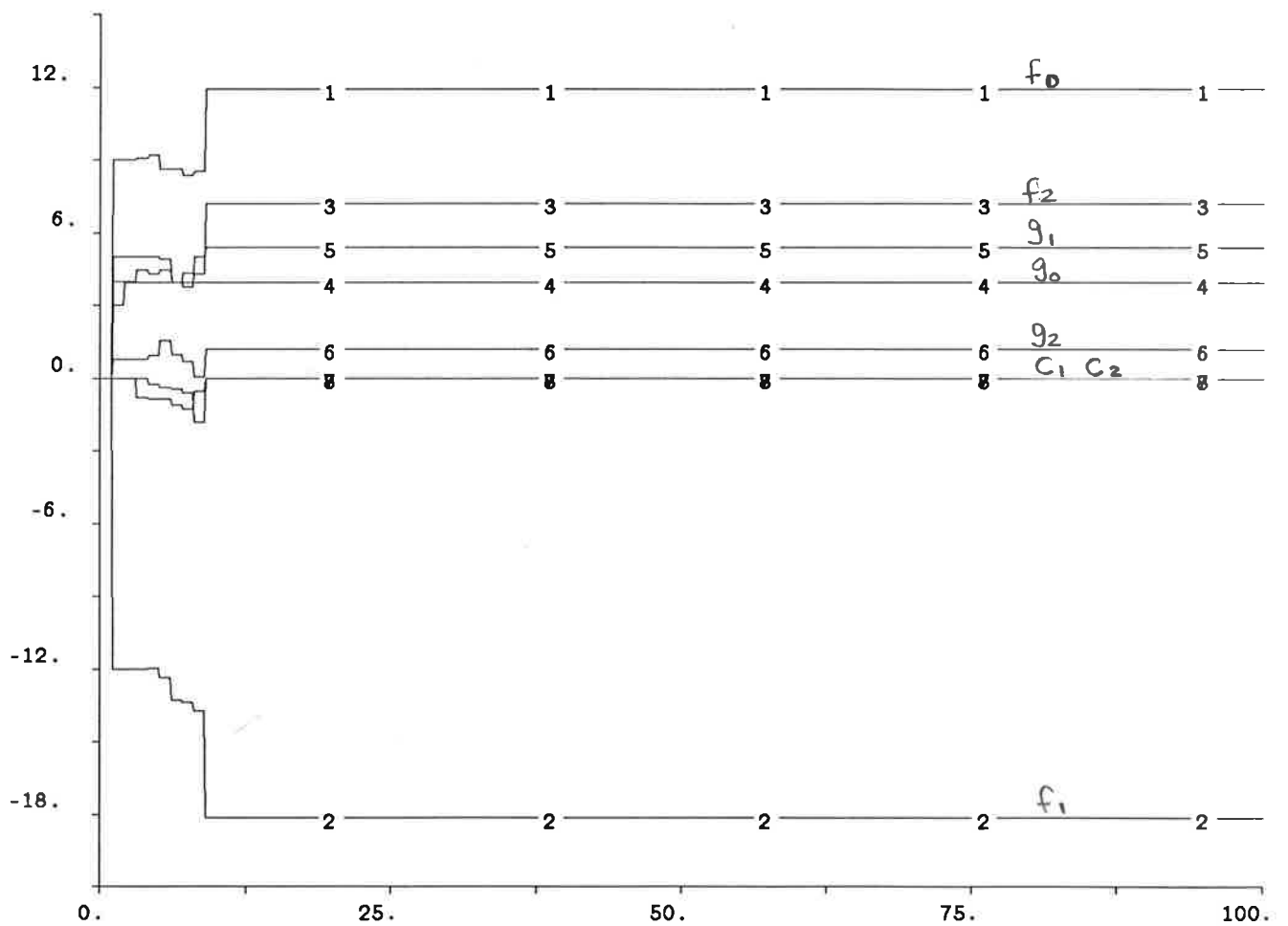
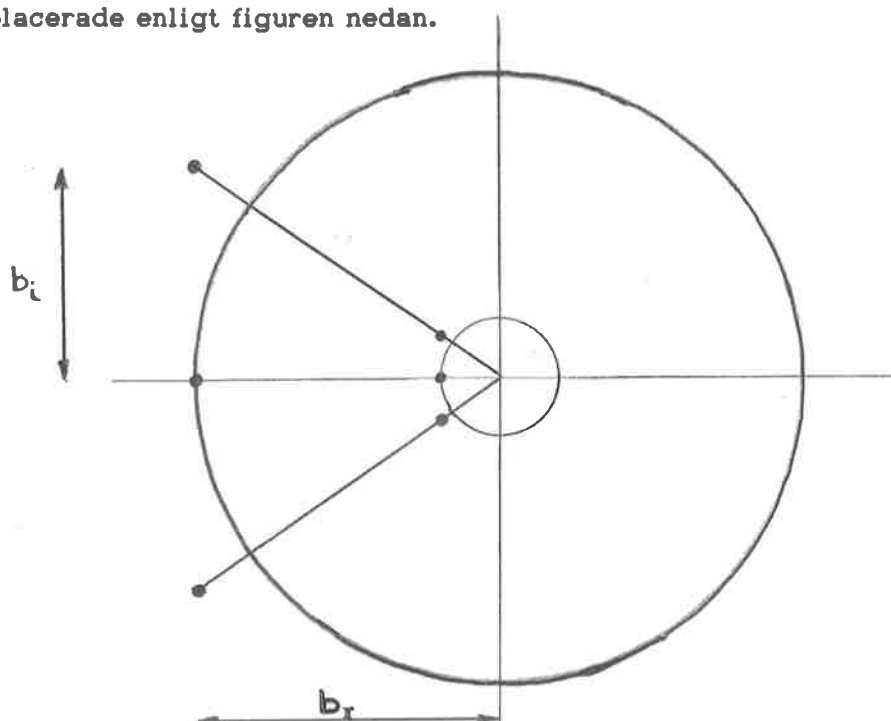


Fig.2. Figuren åskådliggör insvängningsförloppet för skattningsparametrarna i designmodellen. Dessa svänger in snabbt och ligger konstant efter 10-15 intervall.

I det föregående fallet hade filtret endast reella rötter. Simuleringar genomfördes också då filtret hade komplexa rötter. Filtrets poler är placerade enligt figuren nedan.



Vid komplexkonjungerande trippelfilter erhålls

$$P = (s^2 + 2\epsilon b_r s + \epsilon^2 (b_r^2 + b_i^2))(s + \epsilon b_r)$$

och

$$D = (s^2 + 2b_r s + (b_r^2 + b_i^2))(s + b_r)$$

På nästa sida görs en jämförelse mellan ett filter med reella rötter och ett med komplexa rötter. Figurerna 3 och 4 visar att det slutna systemet blir något snabbare med den ovanstående polplaceringen för filtret, och att ju mer den imaginära delen ökar desto mer kommer ϕ -signalen att "spika". Det bör påpekas att den imaginära delen ej bör vara större än den reella delen.

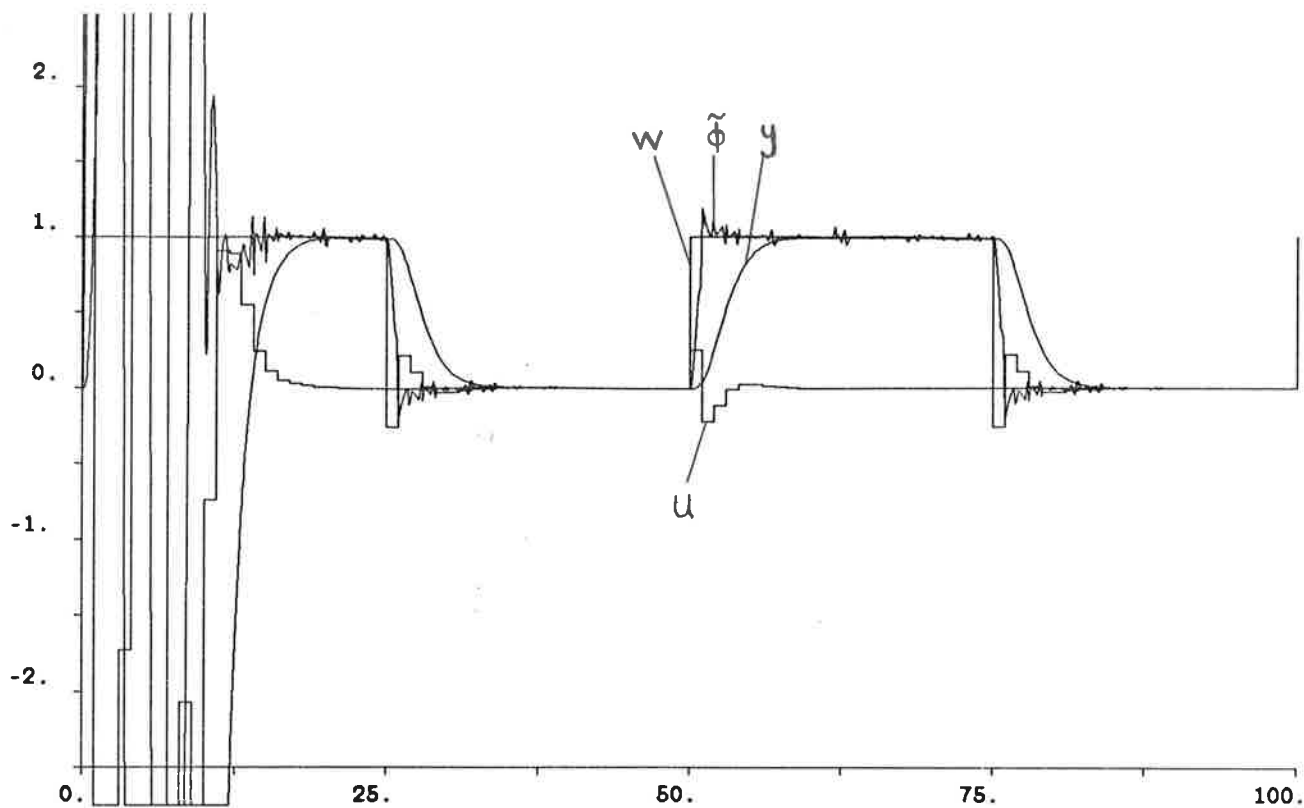


Fig.3. $b_r=10$, $\epsilon=0.1$ och $h=1$. Regleringsförloppet då filtret har reella rötter.

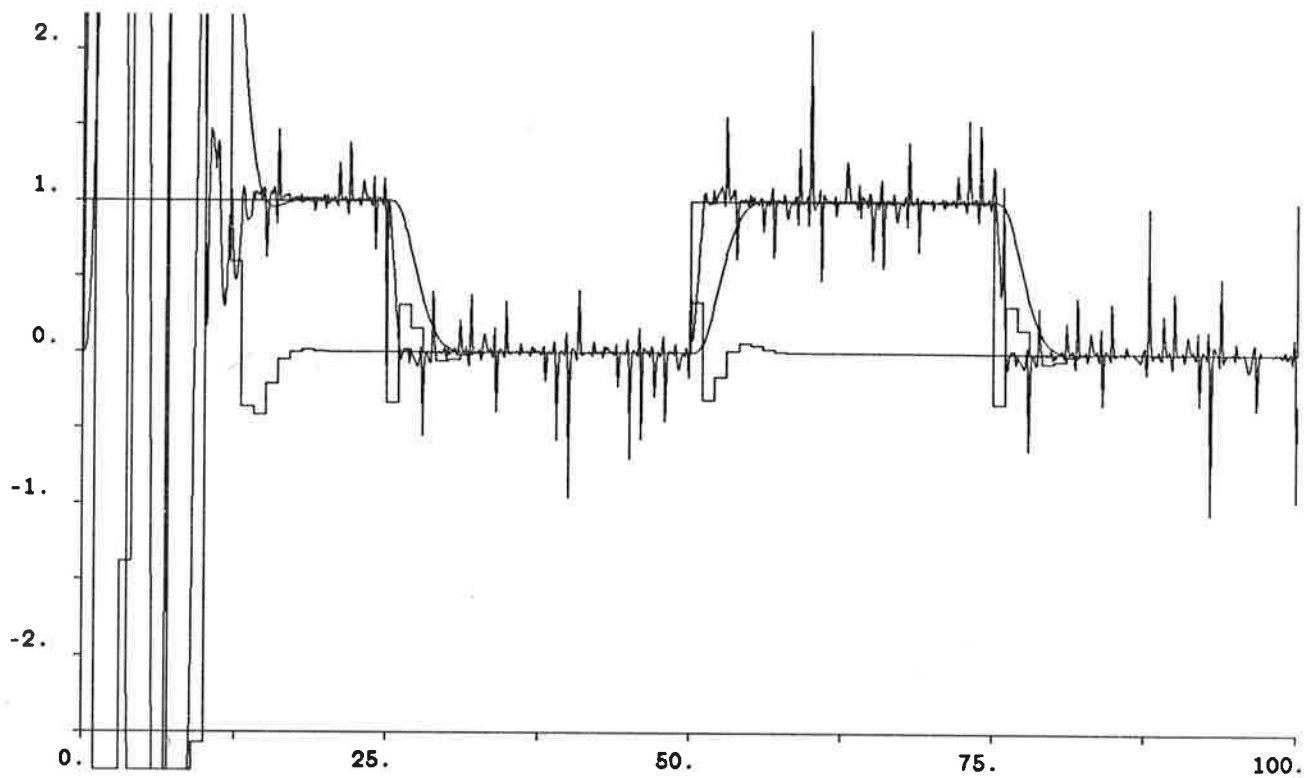


Fig.4. $b_r=10$, $b_i=7$, $\epsilon=0.1$ och $h=1$. Regleringsförloppet då filtret har komplexa rötter.

Ännu ett sätt att snabba upp systemet visade sig vara beroende på hur man beskrev filtret. Egentligen är det inte skillnaden mellan snabbheten för det slutna systemet som är avgörande, utan det är de numeriska beräkningarna som gör att ϕ "spikar" mindre vid de aktuella tidskonstanterna för fall 2. Kontentan blir således att det slutna systemet kan göras snabbare för fall 2 utan att ϕ uppför sig dåligt.

I föregående simuleringar har tillståndsbeskrivningen för ett element erhållits ur

$$\begin{aligned} \frac{dx}{dt} &= -bx + u && \text{fall 1} \\ y &= k(a-b)x + ku \end{aligned}$$

jämfört med den diskuterade tillståndsbeskrivningen

$$\begin{aligned} \frac{dx}{dt} &= -bx + ku && \text{fall 2} \\ y &= (a-b)x + ku \end{aligned}$$

Simuleringar med fall 2 gav

$$8 < b < 16, \quad 0.6 < a < 1.4$$

då samplingsintervallet var 1.

De två fallen har alltså samma överföringsfunktion men olika tillståndsbeskrivning. Att det är de numeriska beräkningarna som är orsak till skillnaden mellan de två fallen kan t ex studeras genom att simulera filtret separat med en insignal som är 1 fram till tiden t . Ovanstående resonemang kan studeras i figur 5,6 och 7, där b -värdet var 16 och ϵ -värdet var 0.1. ($a = \epsilon b$).

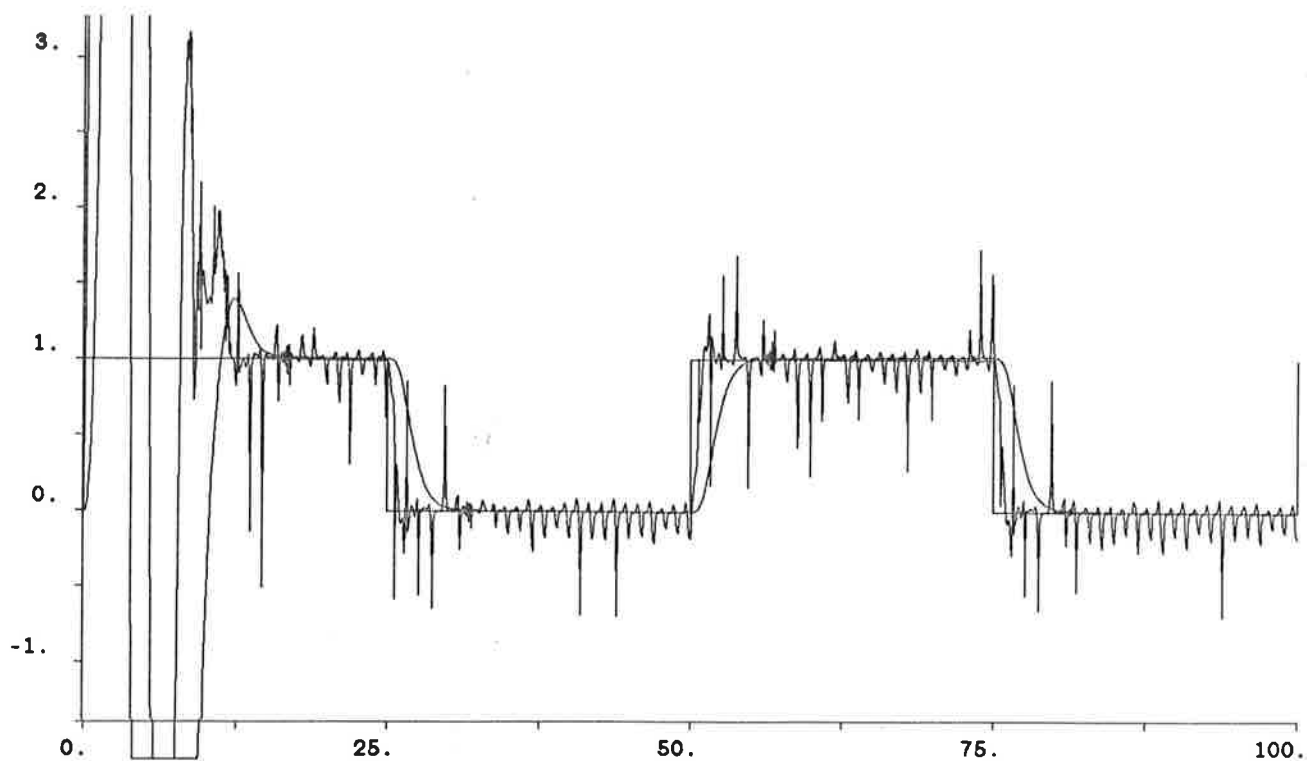


Fig.5. $b=16, \epsilon=0.1$ och $h=1$. Med filterbeskrivningen enligt fall 1, kommer ϕ -signalen att spika betydligt vid de aktuella tidskonstanterna. Förövrigt uppför de andra signalerna sig identiskt med fall 2.

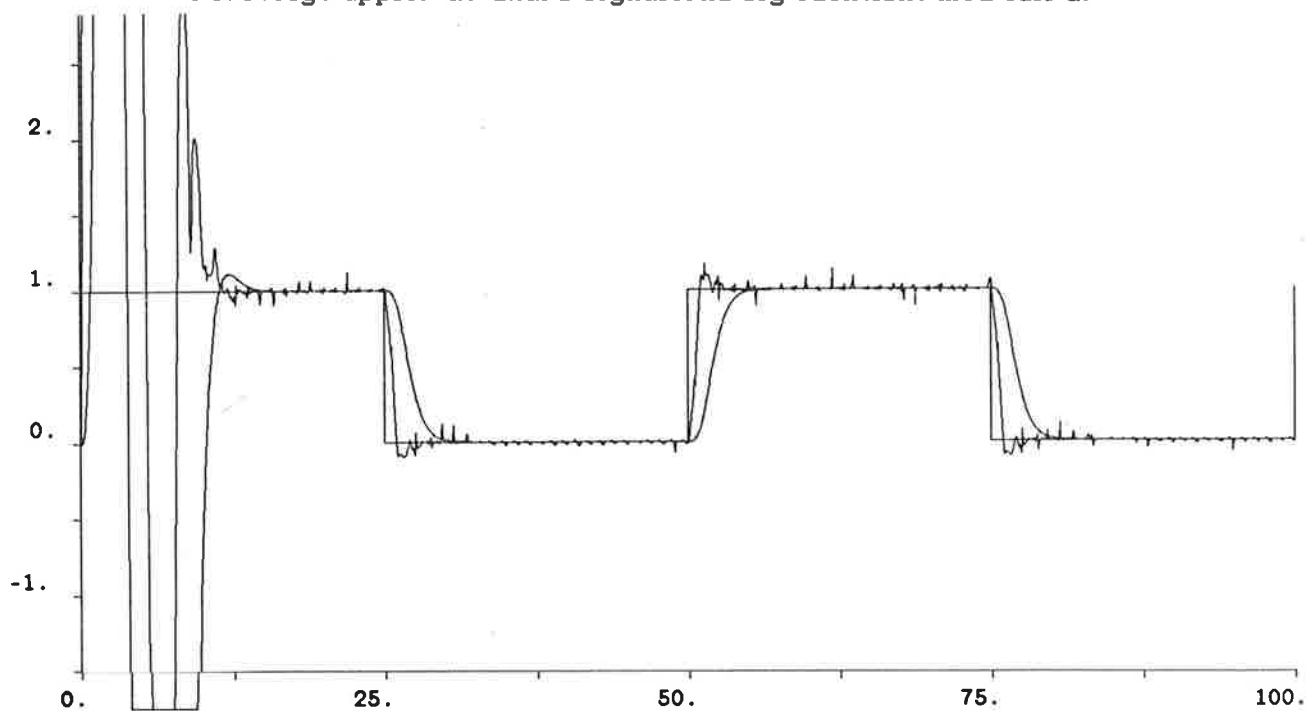


Fig.6. $b=16, \epsilon=0.1$ och $h=1$. Med filterbeskrivningen enligt fall 2, uppför sig ϕ -signalen betydligt bättre även då det slutna systemet görs snabbare.

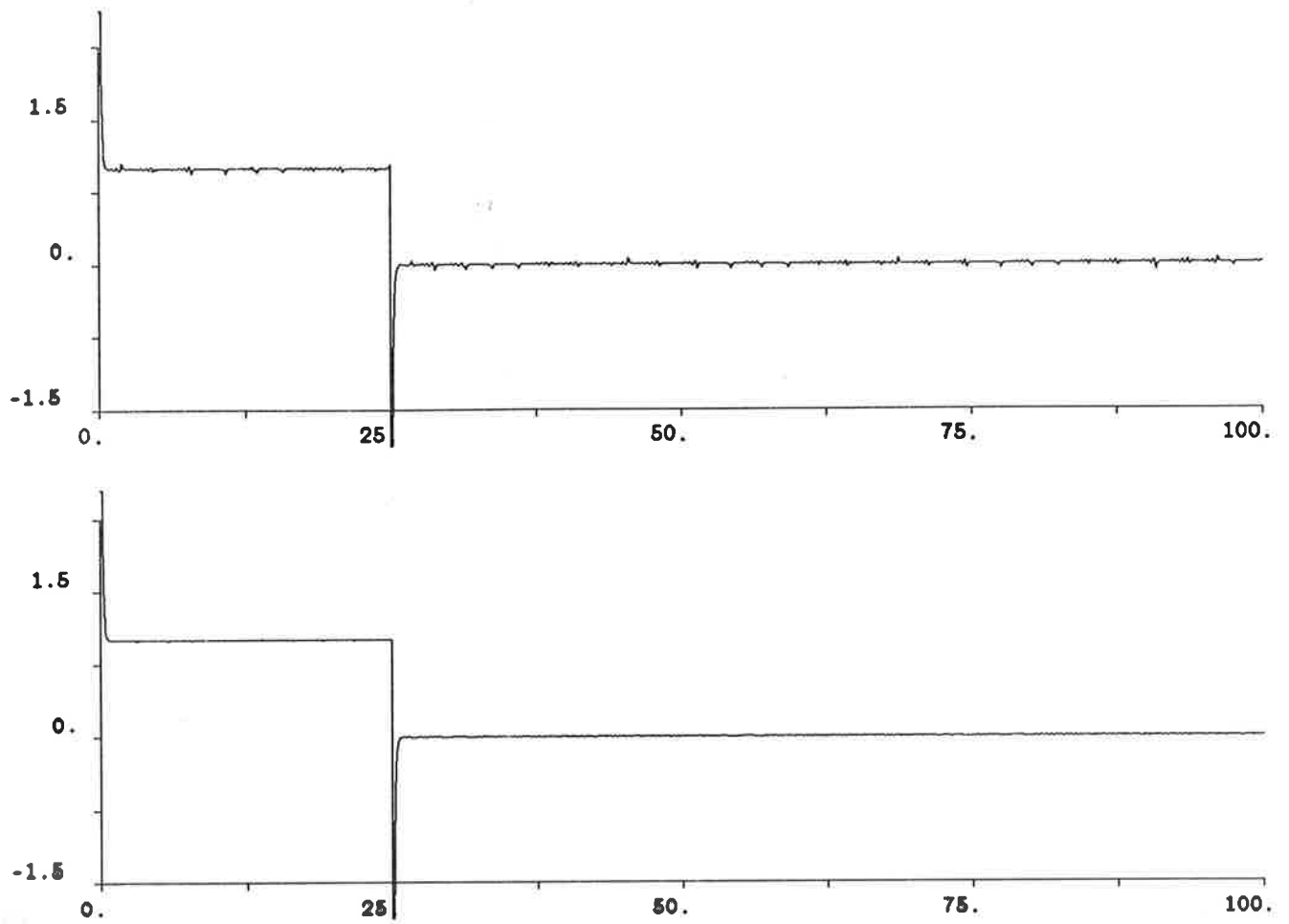


Fig.7. Separata simuleringar av filtret. Övre figuren anger fall 1 och den undre figuren fall 2.

Val av Q

Styrlagen är beroende på vilken av designmetoderna man använder. Graden på designpolynomen G, F och C bestäms av identiteten (4). Alltså får man designmetod 2 enligt

$$\phi^*(i+1|i) = (g_0 + g_1 q^{-1} + g_2 q^{-2})u(i) + (f_0 + f_1 q^{-1} + f_2 q^{-2})y(i) - (c_1 q^{-1} + c_2 q^{-2})\phi^*(i|i-1)$$

som sedan sätts in i

$$u'(i) = \frac{1}{Q} [\phi^*(i+1|i) - w(i)]$$

I special fallet är $Q = 0$, vilket innebär att

$$\phi^*(i+1|i) = w(i)$$

Styrsignalen löses då direkt ur designmetoden.

Q behöver inte nödvändigtvis vara en P-regulator, utan kan ersättas med en PI-regulator. Styrsignalens överföringsfunktion blir då istället

$$Q' = \frac{1}{g_0 - Q}$$

Med endast proportionell delen inkopplad visade det sig att ju mer a-värdet ökar desto mer begränsat blev Q. För små a-värden kunde Q (k_p) väljas upp till ett par enheter, beroende på b-värdet, medan det för större a-värden begränsades till några tiondelar. Här varierades a och b fritt, utan att sambandet $a = eb$ gällde. Genom införandet av integratordelen (k_I) kunde man justera bort det stationära felet, vilket är anledningen till att använda en PI-regulator. ϕ påverkas också positivt i den meningen att signalen inte "spikar" så mycket. Q kan därmed variera mer fritt med hänsyn till väl valda konstanter. Den blir dock känslig för ett alltför stort avvikande på konstanterna.

Initialvärde, begränsningar

Även initialvärde beroendet på skattningsparametrarna G , F och C undersöktes och gav

* några initialvärde till F -polynomen måste vara kända.

* alla initialvärde till G -polynomen kan inte vara okända, men i vissa fall beroende på filterkonstanterna, behövs inga initialvärde.

* speciellt parametern som står i nämnaren, g_0 , behöver man inte ha någon vetskap om.

Nämnas bör också att skattningsparametrarna till C -polynomen var så små, att dessa skulle kunna approximeras till 0.

Det visade sig också att glömskefaktorn inte hade någon märkbar påverkan på styrsignalen vid längre simuleringar.

3.2. Process s⁻³ med design 1

Programmen för processen, filtret och regulatorn med dess design 1 återfinns i appendix B.

Här visas hybridregulatorns egenskaper applicerad med designmetod 1. Förövrigt gäller samma antaganden som gjordes i föregående kapitel. Fall 1 har använts för att beskriva filtret. Skillnaden mellan designmetoderna ligger i att man använde filtrets utsignal istället för processens utsignal. Med samma förutsättningar som föregående kapitel, får man designmodellen enligt

$$\begin{aligned} \phi^*(i+1|i) = & (g_0 + g_1 q^{-1} + g_2 q^{-2})u(i) + (f_0 + f_1 q^{-1} + f_2 q^{-2})\tilde{\phi}(i) \\ & - (c_1 q^{-1} + c_2 q^{-2})\phi^*(i|i-1) \end{aligned}$$

som sedan sätts in i

$$u'(i) = \frac{1}{Q} [\phi^*(i+1|i) - w(i)]$$

Figurerna 8,9 och 10 på nästa sida visar att denna designmetod har helt klart sämre egenskaper än designmetod 2. Samma värden på b, ϵ och h har valts för att illustrera skillnaden i regleringen, alltså $b=10, \epsilon=0.1$ och $h=1$. I figur 11 åskådliggörs insvängningsförloppet för skattningsparametrarna. En jämförelse mellan insvängningsförloppen för de båda designmetoderna visar att insvängningen är betydligt "trögare" och även instabilare för designmetod 1.

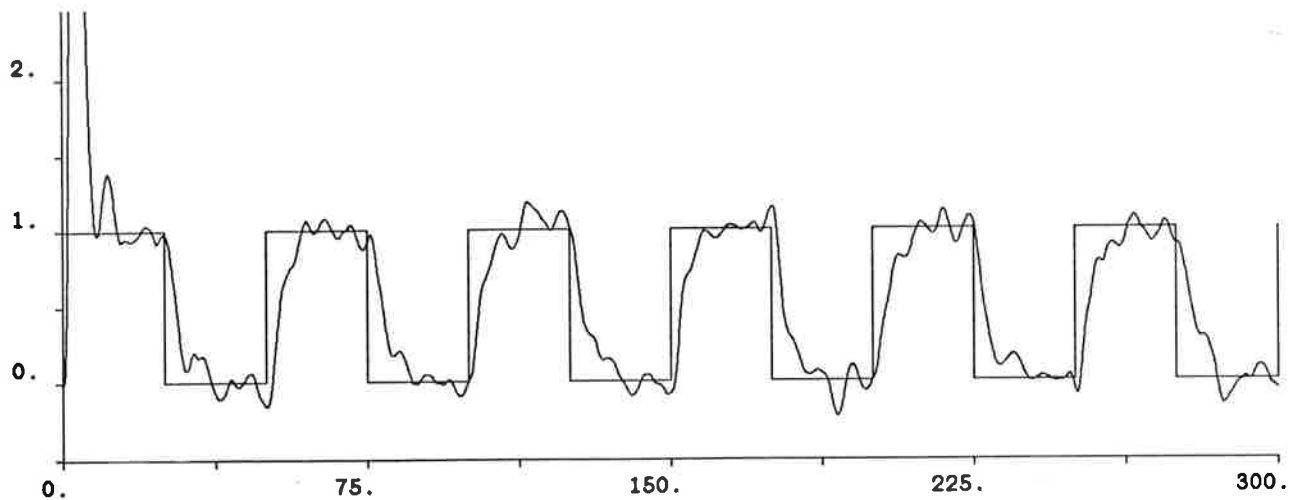


Fig.8. Utsignalen svänger in väldigt snabbt men följer referenssignalen dåligt.

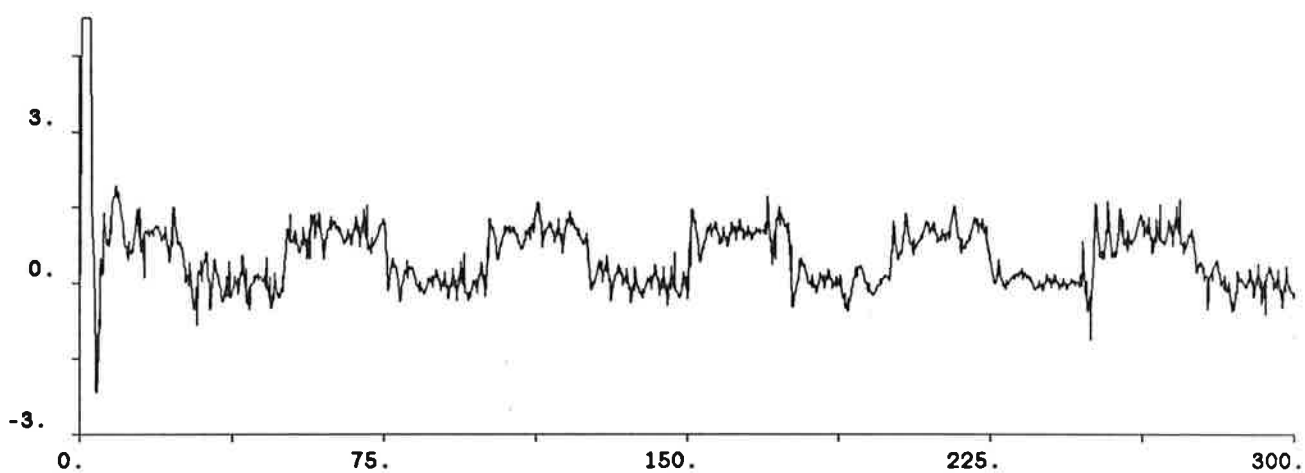


Fig.9. ϕ -signalens uppförande visar att denna är väldigt instabil.

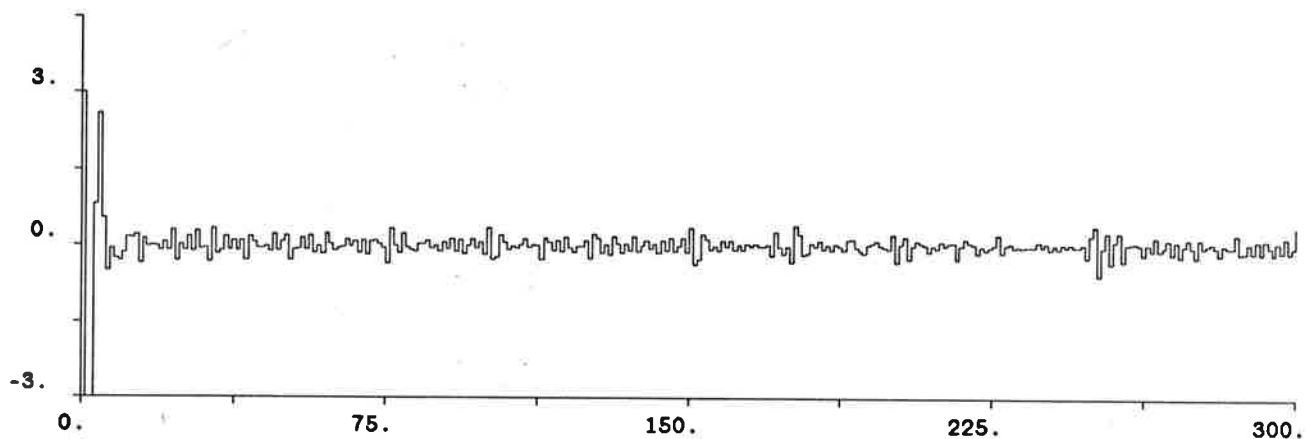


Fig.10. Styrsignalens uppförande visar att denna är instabil.

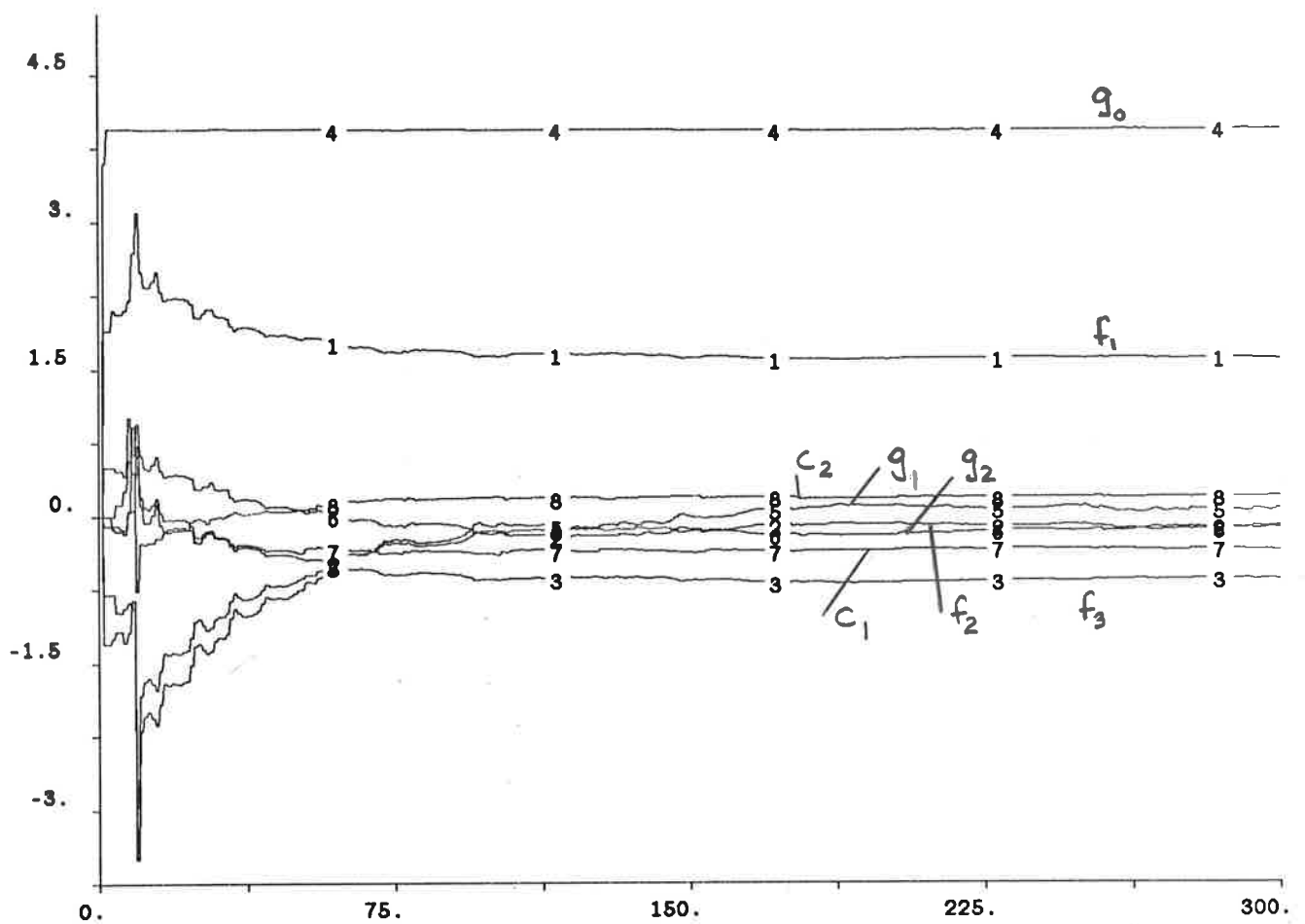


Fig.11. Insvängningsförloppet för skattningsparametrarna med designmetod 1 visar att insvängningen är trögare än med designmetod 2. Parametrarna är också instabila.

3.3. Process $(1-s)(s(s+1))^{-1}$

Programmen för processen, filtret och regulatorn med dess design återfinns i appendix C.

Denna process representerar ett icke-minfassystem i det kontinuerliga rummet. Orsaken till att icke-minfassystem är svårreglerade beror på att fasmarginalen minskar och ger därigenom lättare upphov till instabilitet. För att klargöra hur nollställena förhåller sig efter samplingen görs följande beräkningar

$$\Phi = e^{Ah} = \begin{bmatrix} e^{-h} & 0 \\ 1 - e^{-h} & 1 \end{bmatrix}$$

och

$$\Gamma = \int e^{As} ds B = \begin{bmatrix} 1 - e^{-h} \\ h - 1 + e^{-h} \end{bmatrix}$$

A och B hämtas från tillståndsformen vilket insatt i överföringsfunktionen ger

$$H(z) = \frac{z(2 - 2e^{-h} - h) + 2 - he^{-h} - 2e^{-h}}{(z-1)(z-e^{-h})}$$

Nollställena ges av

$$z = \frac{(2 - he^{-h} - 2e^{-h})}{(2 - 2e^{-h} - h)}$$

alltså ger denna minfas då $h > 3.8$

P-och D-polynomens gradtal bestämdes ur de allmänna gradtalen, vilket gav

$$\text{grad}(P)=1; \text{grad}(D)=1$$

Designmetod 2 användes även i detta fallet och graden på skattningsparametrarna erhöles m h a sambandet

$$y(t) = \frac{A}{B} u(t) + \frac{C}{A} \psi(t) = \frac{-s+1}{s(s+1)} u(t) + \frac{s+1}{s(s+1)} \psi(t)$$

och identiteten

$$\frac{CP}{A} = E + \frac{F}{A} = 1 + \frac{s+1}{s(s+1)}$$

Simuleringarna visade att regleringen ej fungerade tillfredsställande för de beräknade gradtalen på filtret. Istället sattes gradtalet på P och D till 2.

Hur kan h väljas ?

e har under simuleringarna varit fixerad till 0.1 och b-värdet till 6. Det har redan nämnts att det är samplingstiden som är avgörande för regleringen. Speciellt intressant är det kring punkten h=3.8.

I figur 12 och 13 ges exempel på hur regleringen fungerar för minfas respektive icke-minfas system. Resultatet visar att hybriden utmärkt klarar av att reglera kontinuerliga icke-minfas system då samplingstiden är mindre än 3.8.

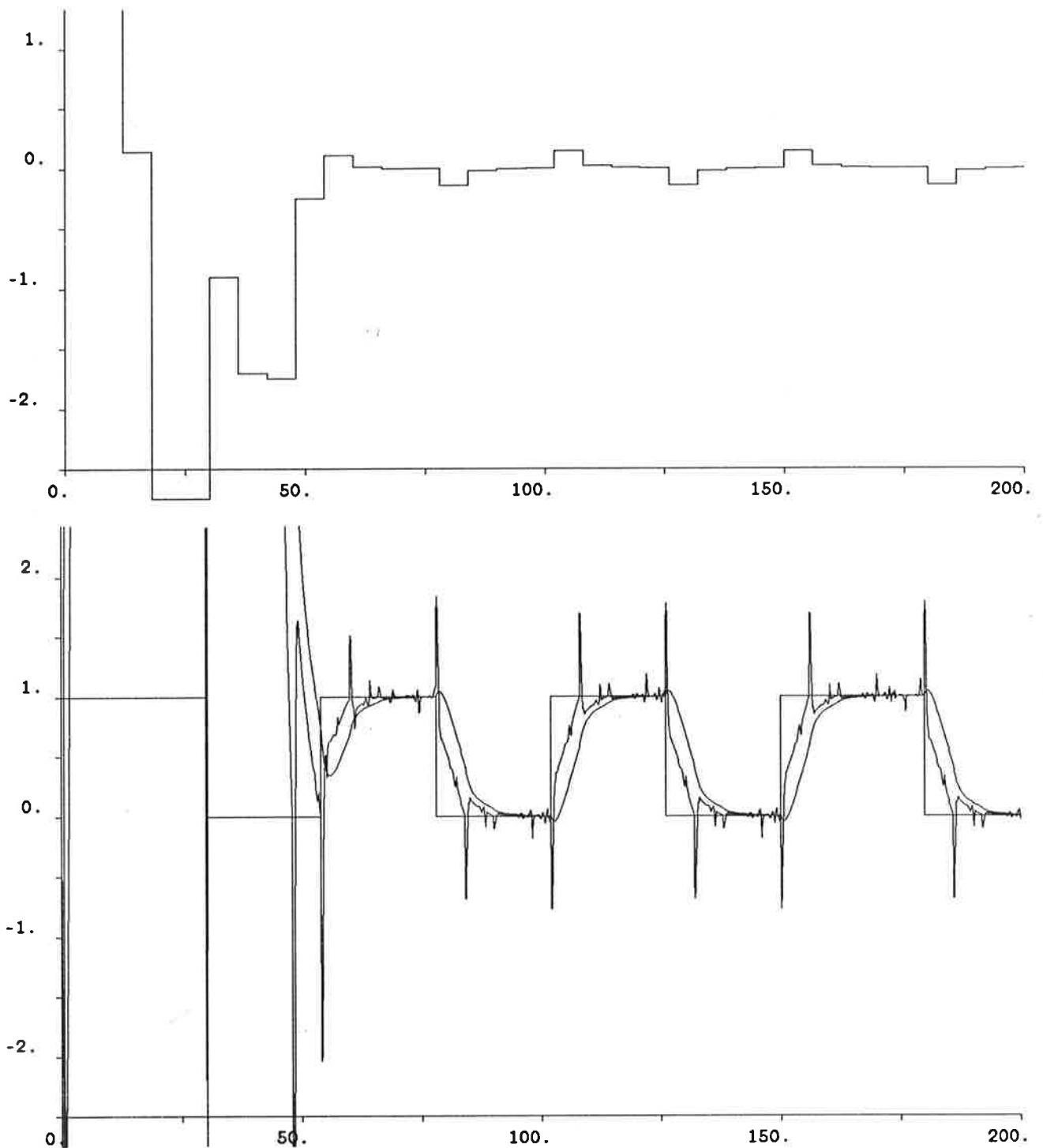


Fig.12. $b=6, \epsilon=0.1$ och $h=6$. Överst visas styrsignalen som har ett stabilt uppförande. I den undre visas uppförandet för y -signalen resp. ϕ -signalen i förhållande till referenssignalen. Signalerna svänger in efter 50 intervaller. Tydligt är också att ϕ -signalen mer följer y -signalens uppförande istället för att snabbt svänga in mot referenssignalen.

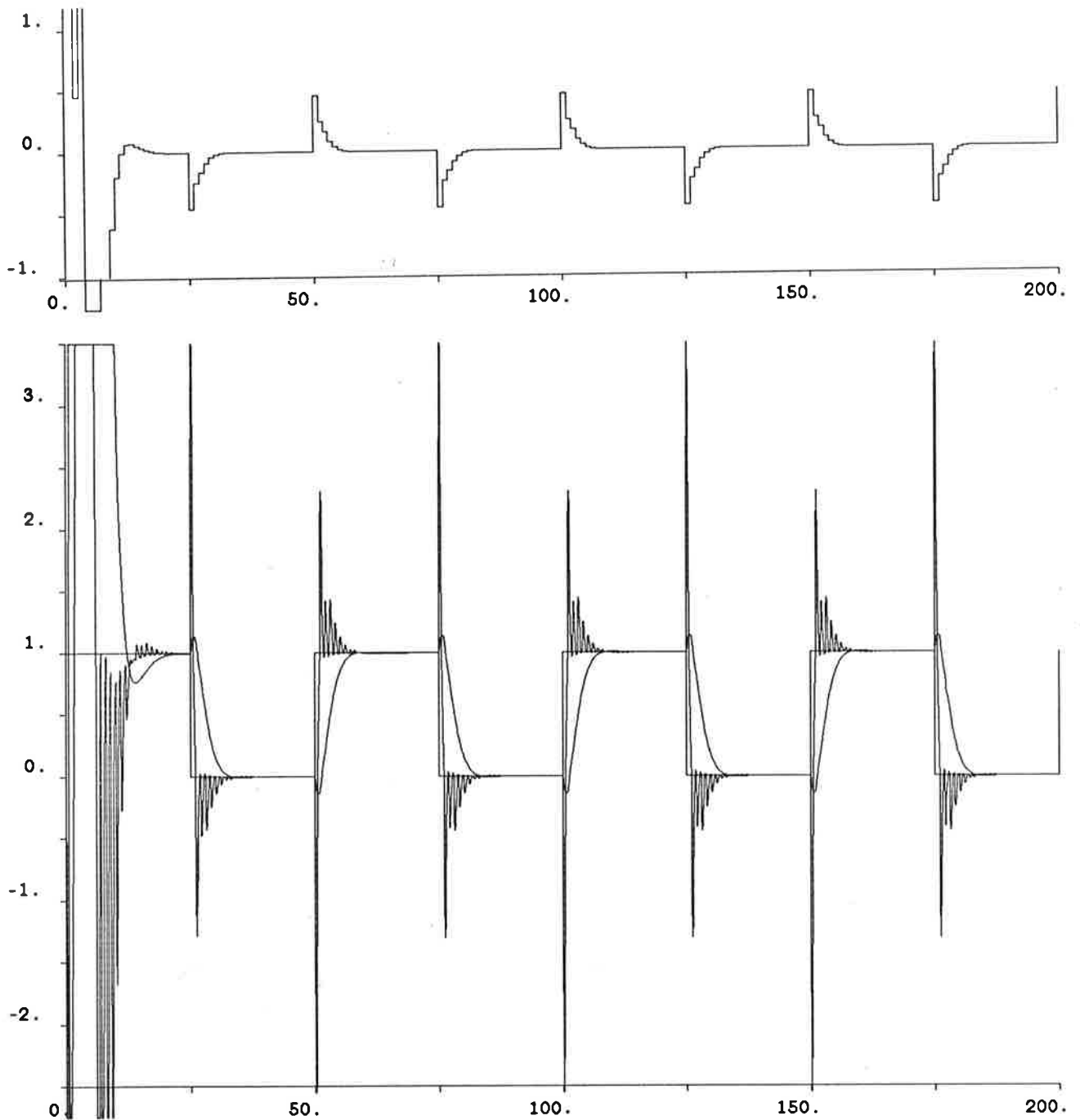


Fig.13. $b=6, \epsilon=0.1$ och $h=1$. Signalerna svänger in snabbt och stabiliseras efter 20 intervaller. Styrsignalen svänger lite mer i förhållande till minfassetmet men däremot är de övriga signalernas uppförande utmärkt vilket visar att regulatorn även klarar av att reglera icke minfassetmet.

3.4. Diskret version av regulatorn

Programmen för processen, filtret och regulatorn med dess design återfinns i appendix D.

Gawthrop påstår att man får ett bättre resultat om man samplar PB/A istället för att sampla B/A och P separat. För att kunna sampla dessa var för sig måste filtret på kontinuerlig form ersättas med ett filter på diskret form.

Samma antaganden vad beträffar designpolynomens gradtal och filtrets gradtal gäller från föregående kapitel. Det finns dock inget D-polynom.

Den kontinuerliga formen på filtret

$$\frac{a}{s+a}$$

motsvaras av

$$\frac{1-e^{-ah}}{q-e^{-ah}}$$

på diskret form.

Förenklat kan man uttrycka ϕ med

$$\phi(i) = (1 - e^{-ah} q^{-1}) y(i)$$

Simuleringarna gav att då a-värdet minskas kommer styrsignalen och utsignalen att svänga mer. Det gav också ett stationärt fel. Man måste vara observant på att samplings tiden inte kan göras hur liten som helst oavsett a-värdet. Regleringen fungerar inte då samplings tiden är mindre än 4. Därav ser man skillnaden mellan hybriden och den diskreta versionen som har antytts innan.

I figur 14 åskådliggörs ovanstående resonemang och i figur 15 visas insvängningsförloppet för skattningsparametrarna.

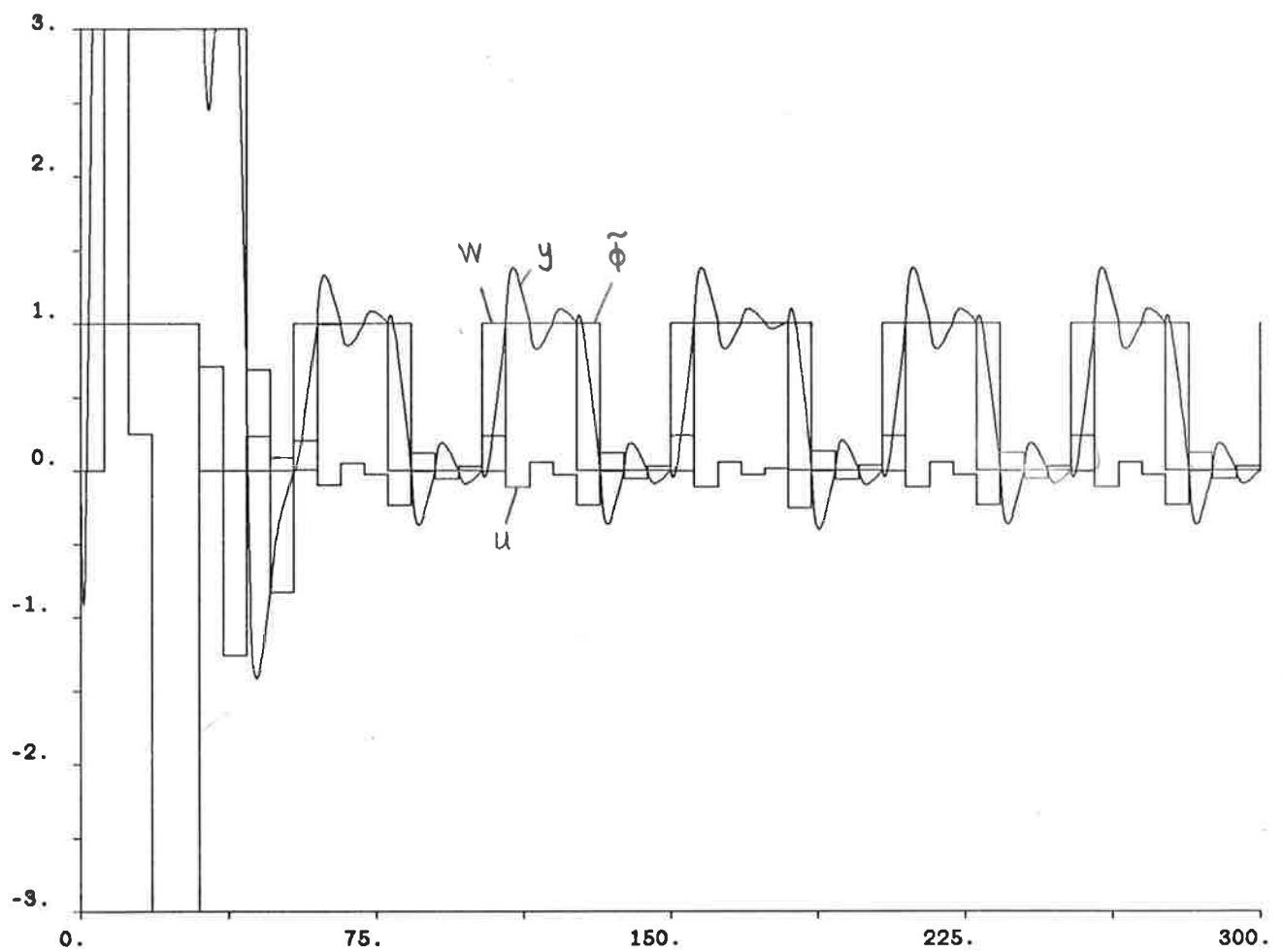


Fig.14. $a=1$ och $h=6$. Figuren visar signalernas uppförande för den diskreta versionen. Jämför regleringsförmågan med fig.12, d v s hybrid versionen. Regleringen fungerar ej för icke-minfassystem.

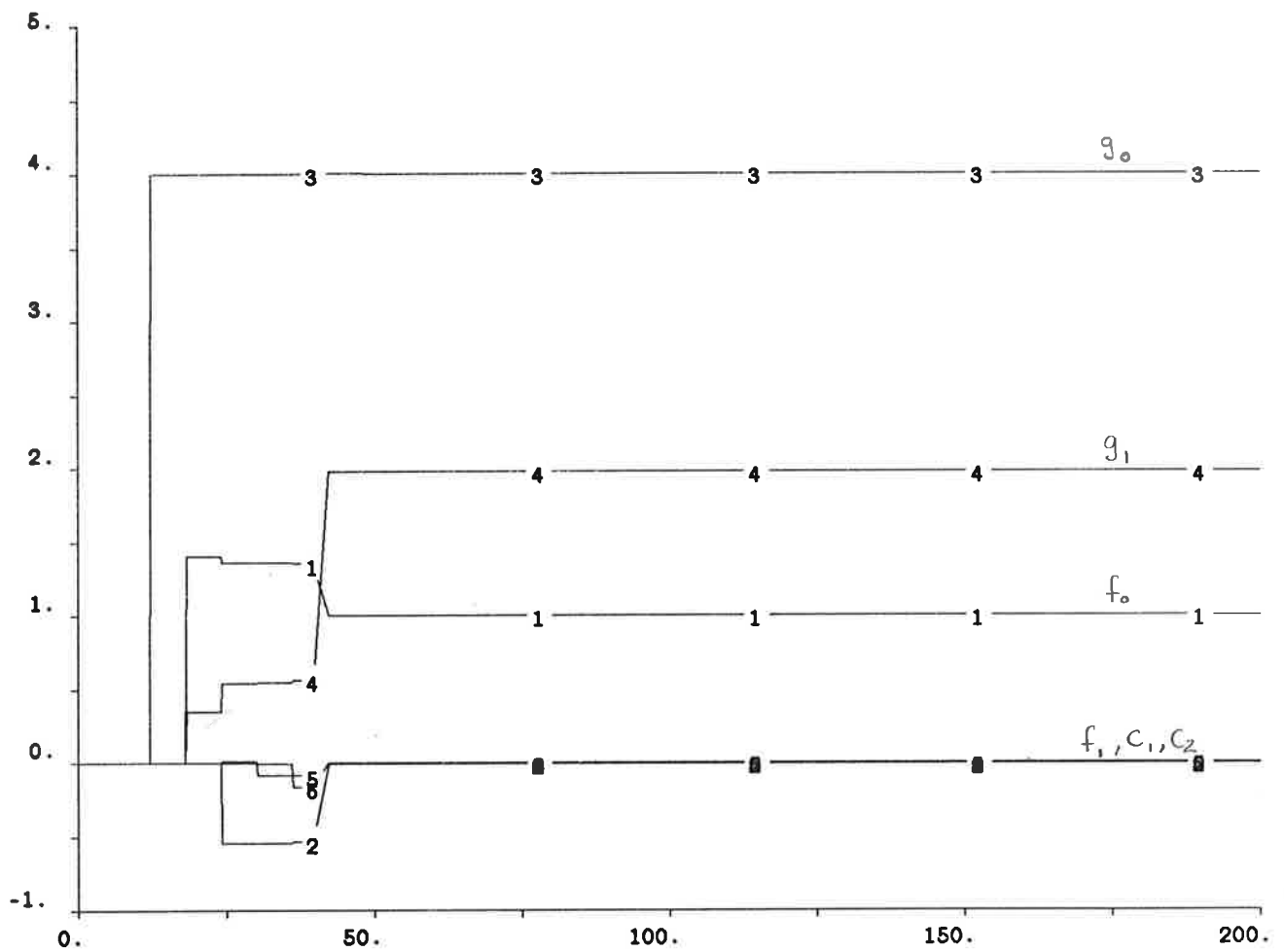


Fig.15. Parametrarna är stabila och svänger in sig efter 30-40 samplingsintervaller.

SLUTSATSER

Resultaten från simuleringar har visat att hybridregulatorn utmärkt klarar av tids-kontinuerliga system med stor grad skillnad mellan poler och nollställe. För trippelintegratorn som bl a har behandlats i denna rapport har den diskreta ekvivalensen alltid nollställe utanför enhetscirkeln oavsett samplingsintervallet. Simuleringarna har också visat att designmetod 2 är att föredra. Filtrets beskrivning på tillståndsformen har diskuterats m a p det slutna systemets snabbhet.

Slutligen undersöktes egenskaperna beroende på om samplingen skedde före eller efter filtret, enligt vad Gawthrop har föreskrivet. Detta gav att hybridregulatorn (sampling efter filtret) fungerade bra för tids-kontinuerliga icke minfssystem. Men däremot fungerade den inte om samplingen skedde före filtret, vilket påvisar det förutsagda. Även för tids-diskreta minfssystem fungerar denna regulator sämre.

CONTINUOUS SYSTEM MOTT

```

input u
output y
state s1 s2 s3
der ds1 ds2 ds3
ds1=u
ds2=s1
ds3=s2
y=s3
end

```

MACRO START

```

let ivs.=4
let isa.=4
let ivr.=8
let isb.=8
syst mott exfi stureg exstyr styr con
par n1:0
par n2:3
par n3:3
par n4:2
par th01:9
par th02:-12
par th03:5
par th04:3
par th05:4
par th06:0.8
axes v -3 3 h 0 100
plot y[mott] u[mott] w fi[exfi]
store u[styr] y[exstyr] w
store p1 p2 p3 p4 fi-add
store p5 p6 p7 p8-add
end

```

CONTINUOUS SYSTEM EXF1

```

input y
output fi
state x1 x2 x3
der dx1 dx2 dx3
dx1=-2*br*x1-(bi*bi+br*br)*x2+y
dx2=x1
yg=2*g3*br*g1*x1+((g2*bi*bi+g2*br*br)*g1)*x2+g1*y
dx3=-br*x3+yg
fi=g3*br*(1/e)*x3+(1/e)*yg

```

```

g1=(1/e)*(1/e)
g2=e*e-1
g3=e-1

```

```

e:0.1
br:10
bi:0
end

```

CONTINUOUS SYSTEM FILTER

```

input y
output fi

state r s t
der dr ds dt
dr=(-b)*r+c*y
ksi1=k*(a-b)*r+k*c*y

ds=(-b)*s+c*ksi1
ksi2=k*(a-b)*s+k*c*ksi1

dt=(-b)*t+c*ksi2
fi=k*(a-b)*t+k*c*ksi2

a=e*b
b:10
k:10
c:1
e:0.1
end

```

DISCRETE SYSTEM EXSTYR

```

input th1 th2 th3 th4 th5 th6 th7 th8 y u
output ut fisk p
state y1 y2 y3 u1 u2 u3 p1 p2 p3 p4 p5 p6 p7 p8 w1 w2 fs1 fs2 fs3
new ny1 ny2 ny3 nu1 nu2 nu3 np1 np2 np3 np4 np5 np6 np7 np8 nw1 nw2
new nfs1 nfs2 nfs3
time t

tsamp ts
p=p4
ny1=y
ny2=y1
ny3=y2
nu1=u
nu2=u1
nu3=u2
np1=th1
np2=th2
np3=th3
np4=th4
np5=th5
np6=th6
np7=th7
np8=th8
nfs1=fisk
nfs2=fs1
nfs3=fs2
fisk=y*p1+y1*p2+y2*p3+u*p4+u1*p5+u2*p6-fs1*p7-fs2*p8
w=if mod(t,per)<per/2 then step else -steg
nw1=w
nw2=w1
ut=(w+p7*w1+p8*w2-p1*y-p2*y1-p3*y2-p5*u1-p6*u2)
ts=t+h
h:1
step:1
steg:0
per:50
end

```

CONTINUOUS SYSTEM STYR

```

input p ut
output u
state x
der dx

dx=-f*x+ut
us=1*(k0-f)*x+l*ut
u=if us<-lim then -lim else if us>lim then lim else us

k0=ki/kp
f=p*k0/rest
l=1/rest
rest=if abs(p-kp)<eps then eps else (p-kp)
eps=0.01
kp=1.0E-30
ki=0
lim=10

end

```

CONNECTING SYSTEM CON

```

time t
u[mott]=u[styr]
ut[styr]=ut[exstyr]
p[styr]=p[exstyr]
u[exstyr]=u[mott]
y[exfi]=y[mott]
y[exstyr]=y[mott]
u1[stureg]=fi[exfi]
u2[stureg]=y[mott]
u3[stureg]=u[exstyr]
u4[stureg]=-fisk[exstyr]
th1[exstyr]=th1[stureg]
th2[exstyr]=th2[stureg]
th3[exstyr]=th3[stureg]
th4[exstyr]=th4[stureg]
th5[exstyr]=th5[stureg]
th6[exstyr]=th6[stureg]
th7[exstyr]=th7[stureg]
th8[exstyr]=th8[stureg]
end

```

MACRO BYTE H1

```

par h:h1
par dt:h1
axes
simu
end

```

CONTINUOUS SYSTEM MOTT

```
input u
output y
state s1 s2 s3
der ds1 ds2 ds3
ds1=u
ds2=s1
ds3=s2
y=s3
end
```

CONTINUOUS SYSTEM FILTER

```
input y
output fi

state r s t
der dr ds dt
dr=(-b)*r+y
ksi1=i*k*(a-b)*r+i*k*y

ds=(-b)*s+ksi1
ksi2=i*k*(a-b)*s+i*k*ksi1

dt=(-b)*t+ksi2
fi=i*k*(a-b)*t+i*k*ksi2

a:1
b:10
k=b/a
i:1
end
```

MACRO START

```

let ivs.=4
let isa.=4
let ivr.=8
let isb.=8
syst mott filter stureg exstyr con
par n1:0
par n2:3
par n3:3
par n4:2
par th01:1.9
par th02:-0.8
par th03:-0.1
par th04:3.6
par th05:-1.3
par th06:0.5
axes v -3 3 h 0 100
plot y[mott] u[mott] w
store u[exstyr] fi[filter] w
store p1 p2 p3 p4-add
store p5 p6 p7 p8-add
end

```

CONNECTING SYSTEM CON

```

time t
u[mott]=u[exstyr]
y[filter]=y[mott]
fi[exstyr]=fi[filter]
u1[stureg]=fi[filter]
u2[stureg]=fi[filter]
u3[stureg]=u[exstyr]
u4[stureg]=-fisk[exstyr]
th1[exstyr]=th1[stureg]
th2[exstyr]=th2[stureg]
th3[exstyr]=th3[stureg]
th4[exstyr]=th4[stureg]
th5[exstyr]=th5[stureg]
th6[exstyr]=th6[stureg]
th7[exstyr]=th7[stureg]
th8[exstyr]=th8[stureg]
end

```

DISCRETE SYSTEM EXSTYR

```

input th1 th2 th3 th4 th5 th6 th7 th8 fi
output u fisk
state fi1 fi2 fi3 u1 u2 u3 p1 p2 p3 p4 p5 p6 p7 p8 w1 w2 fs1 fs2 fs3
new nfi1 nfi2 nfi3 nu1 nu2 nu3 np1 np2 np3 np4 np5 np6 np7 np8 nw1 nw2
new nfs1 nfs2 nfs3
time t
tsamp ts
nfi1=fi
nfi2=fi1
nfi3=fi2
nu1=u
nu2=u1
nu3=u2
np1=th1
np2=th2
np3=th3
np4=th4
np5=th5
np6=th6
np7=th7
np8=th8
nfs1=fisk
nfs2=fs1
nfs3=fs2
fisk=fi*p1+fi1*p2+fi2*p3+u*p4+u1*p5+u2*p6-fs1*p7-fs2*p8
w=if mod(t,per)<per/2 then step else -steg
nw1=w
nw2=w1
ut=(w+p7*w1+p8*w2-p1*fi-p2*fi1-p3*fi2-p5*u1-p6*u2)/korr
korr=if abs(p4-q)<eps then eps else (p4-q)
u=if ut<-lim then -lim else if ut>lim then lim else ut
ts=t+h
eps:0.01
lim:3
h:1
step:1
steg:0
per:50
q:0
end

```

CONTINUOUS SYSTEM MOTT

```

input u
output y
state s1 s2
der ds1 ds2
ds1=-s1+u
ds2=s1
y=-s1+s2
end

```

CONTINUOUS SYSTEM FILTER

```

input y
output fi

state r s
der dr ds
dr=(-b)*r+y
yi=k*(a-b)*r+k*y
ds=(-b)*s+yi
fi=k*(a-b)*s+k*yi
a=e*b
b:5
e:0.1
k=b/a
end

```

MACRO HYB t

```

par h:t
par dt:t
end

```

MACRO WER aa bb hh

```

par a:aa
par b:bb
par h:hh
par dt:hh
axes v -3 3 h 0 100
plbt w u[mott] y[mott] fi[filter]
simu 0 100
end

```


DISCRETE SYSTEM EXSTYR

```

input th1 th2 th3 th4 th5 th6 y
output u fisk
state y1 y2 y3 u1 u2 u3 p1 p2 p3 p4 p5 p6 w1 w2 fs1 fs2 fs3
new ny1 ny2 ny3 nu1 nu2 nu3 np1 np2 np3 np4 np5 np6 nw1 nw2
new nfs1 nfs2 nfs3.,
time t
tsamp ts

ny1=y
ny2=y1
ny3=y2
nu1=u
nu2=u1
nu3=u2
np1=th1
np2=th2
np3=th3
np4=th4
np5=th5
np6=th6
nfs1=fisk
nfs2=fs1
nfs3=fs2
fisk=y*p1+y1*p2+u*p3+u1*p4-fs1*p5-fs2*p6
w=if mod(t,per)<per/2 then step else -steg
nw1=w
nw2=w1
ut=(w+p5*w1+p6*w2-p1*y-p2*y1-p4*u1)/korr
korr=if abs(p3-q)<eps then eps else (p3-q)
u=if ut<-lim then -lim else if ut>lim then lim else ut
ts=t+h
eps:0.01
lim:10
h:1
step:1
steg:0
per:50
q:0
end

```

CONNECTING SYSTEM CON

```

time t
u[mott]=u[exstyr]
y[filter]=y[mott]
y[exstyr]=y[mott]
u1[stureg]=fi[filter]
u2[stureg]=y[mott]
u3[stureg]=u[exstyr]
u4[stureg]=-fisk[exstyr]
th1[exstyr]=th1[stureg]
th2[exstyr]=th2[stureg]
th3[exstyr]=th3[stureg]
th4[exstyr]=th4[stureg]
th5[exstyr]=th5[stureg]
th6[exstyr]=th6[stureg]
end

```

MACRO START

```

let ivs.=4
let isa.=4
let ivr.=6
let isb.=8
syst mott filter stureg exstyr con
par n1:0
par n2:2
par n3:2
par n4:2
par th01:0
par th02:0
par th03:0
par th04:0
par th05:0
par th06:0
axes v -3 3 h 0 100
plot y[mott] u[mott] w fi[filter]
store u[exstyr] y[exstyr] w
store p1 p2 p3 fi-add
store p4 p5 p6-add
store fi[filter]-add
end

```

CONTINUOUS SYSTEM MOTT

```
input u
output y
state s1 s2
der ds1 ds2
```

```
ds1=-s1+u
ds2=s1
y=-s1+s2
end
```

DISCRETE SYSTEM FILTER

```
input y
output fi
time t
tsamp ts
state y1 y2
new ny1 ny2
```

```
ny1=y
ny2=y1
```

```
fi=y-c*y1
```

```
a=0.5
c=exp(-h*a)
ts=t+h
h=6
end
```

```

DISCRETE SYSTEM EXSTYR
input th1 th2 th3 th4 th5 th6 y
output u fisk
state y1 y2 y3 u1 u2 u3 p1 p2 p3 p4 p5 p6 w1 w2 fs1 fs2 fs3
new ny1 ny2 ny3 nu1 nu2 nu3 np1 np2 np3 np4 np5 np6 nw1 nw2
new nfs1 nfs2 nfs3
time t
tsamp ts
ny1=y
ny2=y1
ny3=y2
nu1=u
nu2=u1
nu3=u2
np1=th1
np2=th2
np3=th3
np4=th4
np5=th5
np6=th6
nfs1=fisk
nfs2=fs1
nfs3=fs2
fisk=y*p1+y1*p2+u*p3+u1*p4-fs1*p5-fs2*p6
nw1=w
nw2=w1
ut=(w+p5*w1+p6*w2-p1*y-p2*y1-p4*u1)/korr
korr=if abs(p3-q)<eps then eps else (p3-q)
u=if ut<-lim then -lim else if ut>lim then lim else ut
ts=t+h
eps:0.01
lim:10
h:1
step:1
steg:0
per:50
q:0
end

```

CONNECTING SYSTEM CON

```

time t
u[mott]=u[exstyr]
y[filter]=y[mott]
y[exstyr]=y[mott]
u1[stureg]=fi[filter]
u2[stureg]=y[mott]
u3[stureg]=u[exstyr]
u4[stureg]=-fisk[exstyr]
th1[exstyr]=th1[stureg]
th2[exstyr]=th2[stureg]
th3[exstyr]=th3[stureg]
th4[exstyr]=th4[stureg]
th5[exstyr]=th5[stureg]
th6[exstyr]=th6[stureg]

```

```

end

```

MACRO START

```

let ivs.=4
let isa.=4
let ivr.=8
let isb.=8
syst mott filter stureg exstyr con
par n1:0
par n2:2
par n3:2
par n4:2
par th01:0
par th02:0
par th03:0
par th04:0
par th05:0
par th06:0
axes v -3 3 h 0 100
plot y[mott] u[mott] w fi[filter]
store u[exstyr] y[exstyr] w
store p1 p2 p3 p4 fi-add
store p5 p6-add
store fi[filter]-add
end

```

Litteraturförteckning

- (1) P.J. Gawthrop : Hybrid self-tuning control. IEE PROC. vol. 127 , pp 229-236 (1980).
- (2) K.J. Åström, B. Wittenmark : Computer controlled system. Prentice Hall Inc. (1984).
- (3) K.J. Åström, B. Wittenmark : Practical issues in the implementation of self-tuning control. Automatica. vol 20, No. 5, pp 595-605 (1984).
- (4) K.J. Åström : Theory and applications of adaptive control-A survey. Automatica, vol. 19, No. 5, pp 471-486 (1983).
- (5) I. Gustavsson : User's guide for a program package for simulations of self-tuning regulators. (1978).
- (6) P.J. Gawthrop : Optimal control applications and methods. Vol. 3 , pp 399-414 (1982).
- (7) D.W. Clarke, A.J.F. Hodgson, P.S. Tuffs : The offset problem and k-incremental predictions in self-tuning control. Report No. 1472/83.
- (8) K.J.Åström : A simnon tutorial. Rapport TFRT-3168. Institutionen för reglerteknik, LTH. (1982).