

CODEN: LUTFD2/(TFRT-5225)/1-066/(1979)

UNDERSÖKNING AV TVÅ SJÄLVINSTÄLLANDE
REGULATORER FÖR SERVOPROBLEMET

SVEN GUNNAR TILJANDER

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA
SEPTEMBER 1979

UNDERSÖKNING AV TVÅ SJÄLVINSTÄLLANDE REGULATORER
FÖR SERVOPROBLEMET:

Svengunnar Tiljander

BT 2000-0000-0000-0000

Dokumentutgivare
Lund Institute of Technology
Handläggare Dept of Automatic Control
Björn Wittenmark
Författare
Sven Gunnar Tiljander

Dokumentnamn
BERFÖRT LUTED2/(TFRT-5225)/1-066/(1979)
Utgivningsdatum
September 1979
Dokumentbeteckning
06T6
Ärendebeteckning
06T6

10T4

Dokumenttitel och undertitel

Undersökning av två självinställande regulatorer för servoproblemet
(Investigation of two self-tuning regulators for the servoproblem)

Referat (sammandrag)

Two self-tuning algorithms to take care of the problem with reference signals have been compared in this work. The first algorithm is given by B Wittenmark (1975) and the second by Clarke-Gawthrop (1975). Both regulators are based on a recursive least-squares estimator of the parameter of a feedback control law combined with feedforward signals. The estimated parameters are then used in the control law as if they are the true ones. Some precaution must then be taken to handle steady-state-errors. In this work this has been done by the following two possibilities: i) cascade the regulator with an integrator ii) and increase the order of the polynomials in the model. The properties of the algorithms are investigated by using simulations.

Referat skrivet av

author

Förslag till ytterligare nyckelord

44T0

Klassifikationssystem och -klass(er)

50T0

Indextermer (ange källa)

52T0

Omfång

66 pages

Övriga bibliografiska uppgifter

56T2

Språk

Swedish

Sekretessuppgifter

60T0

ISSN

60T4

ISBN

60T6

Dokumentet kan erhållas från

Department of Automatic Control
Lund Institute of Technology
P O Box 725, S-220 07 LUND 7, Sweden

Mottagarens uppgifter

62T4

Pris

66T0

DOKUMENTATABLAD enligt SIS 62 10 12

Blankett LU 11:25 1976-07

1. INLEDNING.

Den grundläggande teorin för självinställande regulatorer har hämtats i Åström-Wittenmark (1973), de bägge modifierade algoritmerna för servoproblemet har hämtats i Wittenmark (1975) och Clarke-Gawthorpe (1975), vid simuleringen av exemplen har ett programpaket framtaget av Gustavsson (1978) använts.

Följande blockschema ligger till grund för servoproblemet (fig 1.1).

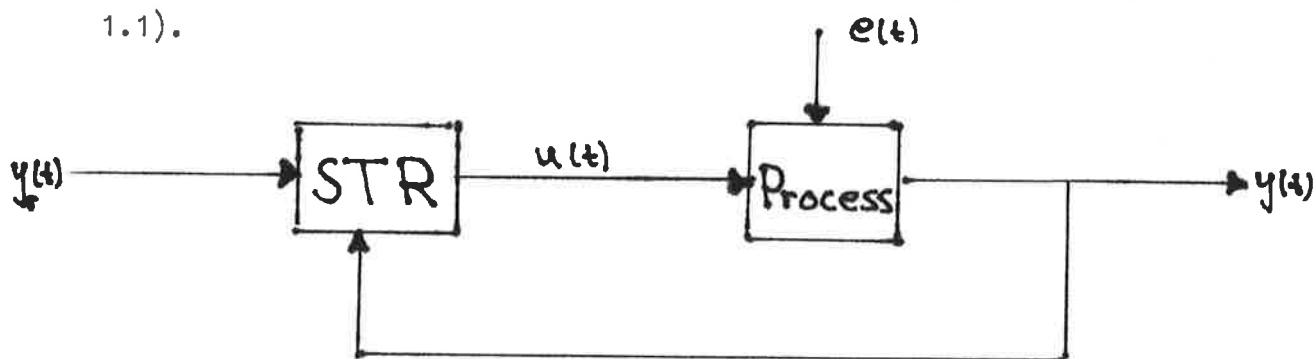


Fig 1.1.

Där processen ovan störs av stationärt vitt brus, samtidigt som denna skall följa referensvärdet Y_r .

STR= self-tuning-regulator, detta är en särskild typ av adaptiv regulator som används då processens koefficienter är konstanta. Tidigare har försök med regulatorer som endast har tillgång till felet $e(t) = y(t) - y_r(t)$ använts för att lösa servoproblemet se Wittenmark (1973), men denna struktur har visat sig innehålla en del brister se Åström-Gustavsson (1978).

För att undvika dessa brister har regulatorer som utnyttjar både reglerfelet $y(t) - y_r(t)$ till återkoppling och referenssignalen $y_r(t)$ till framkoppling framtagits, se Wittenmark (1975) och Clarke-Gawthorpe (1975).

INNEHÅLLSFÖRTECKNING

I. Inledning med problembeskrivning	
2. Beskrivning av algoritmerna	3
3. Simulerade exempel	8
1:a ordningens process, utan integrator	9
3:e ordningens process, utan integrator	21
Känslighet för felaktig β_0 -parameter	32
Konstant störnings inverkan på regleringen	38
Kaskadkoppling med en integrator	44
4. Sammanfattning	59
5. Referenser	63

För att parametrarna som är associerade med framkopplingssignalen skall kunna estimeras effektivt, har en fyrkantvåg med relativt hög amplitud och långsam variation använts. (fig 1.2).

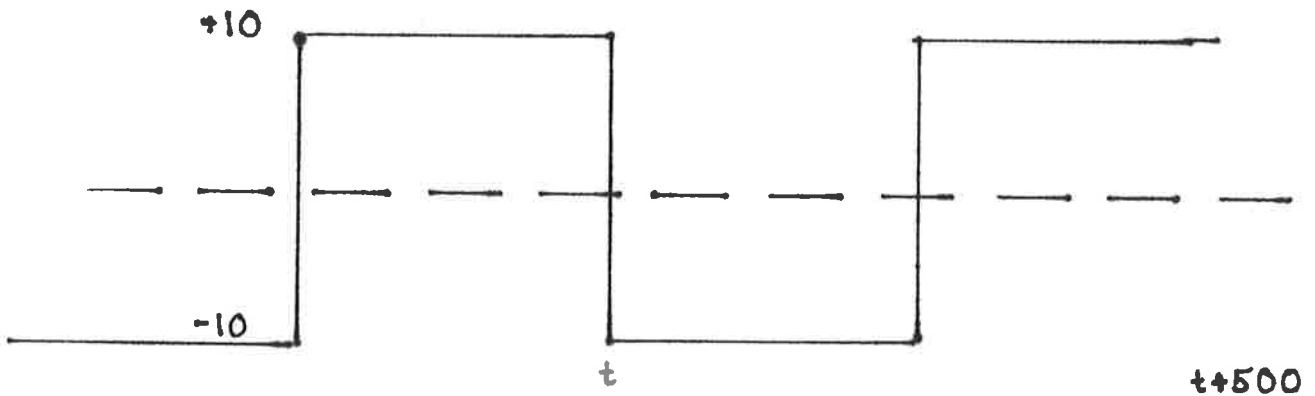


Fig 1.2 Referenssignalens utseende.

Den huvudsakliga uppgiften med detta examensarbete är att undersöka och jämföra de bägge metoderna, under en rad olika förutsättningar.

För att undvika stationära fel, föreslår Wittenmark att processens bör innehålla en integrator, Clarke-Gawthrop på andra sidan löser detta genom att estimerar en eventuell bias-term. I detta arbete kommer dessa förutsättningar att undersökas genom att jämföra hur processen beter sig, då vi ej har integrator och då vi kaskadkopplar med en integrator.

Vidare kommer att undersökas hur en parameter β_0 (som i princip påverkar styrsignalens storlek), inverkar på regleringen då β_0 varierar.

För att se hur utsignalens beter sig, då vi låter en konstant yttre störning påverka styrsignalen, kommer storleken på estimeringsmodellerna att varieras.

Arbetet är upplagt så, att i kapitel 2 presenteras algoritmerna som används och simuleringarna med utförligare diskussioner om de problem som uppstår, presenteras i kapitel 3. Slutligen ges en sammanfattning i kapitel 4.

Till sist vill jag rikta ett tack till min handledare Björn Wittenmark för uppläggning och diskussion av arbetets utförande.

2. BESKRIVNING AV ALGORITMerna.

Algoritmerna som används är avsedda att styra system av typen

$$\hat{A}(q^{-1})y(t) = q^k \hat{B}(q^{-1})u(t) + \hat{C}(q^{-1})e(t)$$

där $e(t)$ är en sekvens av oberoende $N(q^{-1})$ -fördelade stokastiska variabler och

$$\hat{A}(q^{-1}) = 1 + a_1 q^{-1} + \dots + a_n q^{-n}$$

$$\hat{B}(q^{-1}) = b_1 q^{-1} + \dots + b_n q^{-n}$$

$$\hat{C}(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_n q^{-n}$$

Vidare förutsetts att \hat{A} -, \hat{B} - och \hat{C} -polynomen är okända men konstanta Dessutom skall systemet vara av minimumfas-typ.

För att styra dessa system har en speciell klass av regulatorer framtagits, nämligen de s.k. själv-inställande regulatorerna, se Åström-Wittenmark (1973), dessa har sedan vidareutvecklats av Wittenmark (1975) och Clarke-Gawthrop (1975), för att kunna följa referensvärde och dess förändringar i det s.k. servoproblemet. I fortsättningen kommer jag att kalla dessa bägge algoritmerna för algoritm 1 resp. algoritm 2.

ALGORITM 1 :

Denna algoritm kan i blockschema representeras av följande figur 2.1.

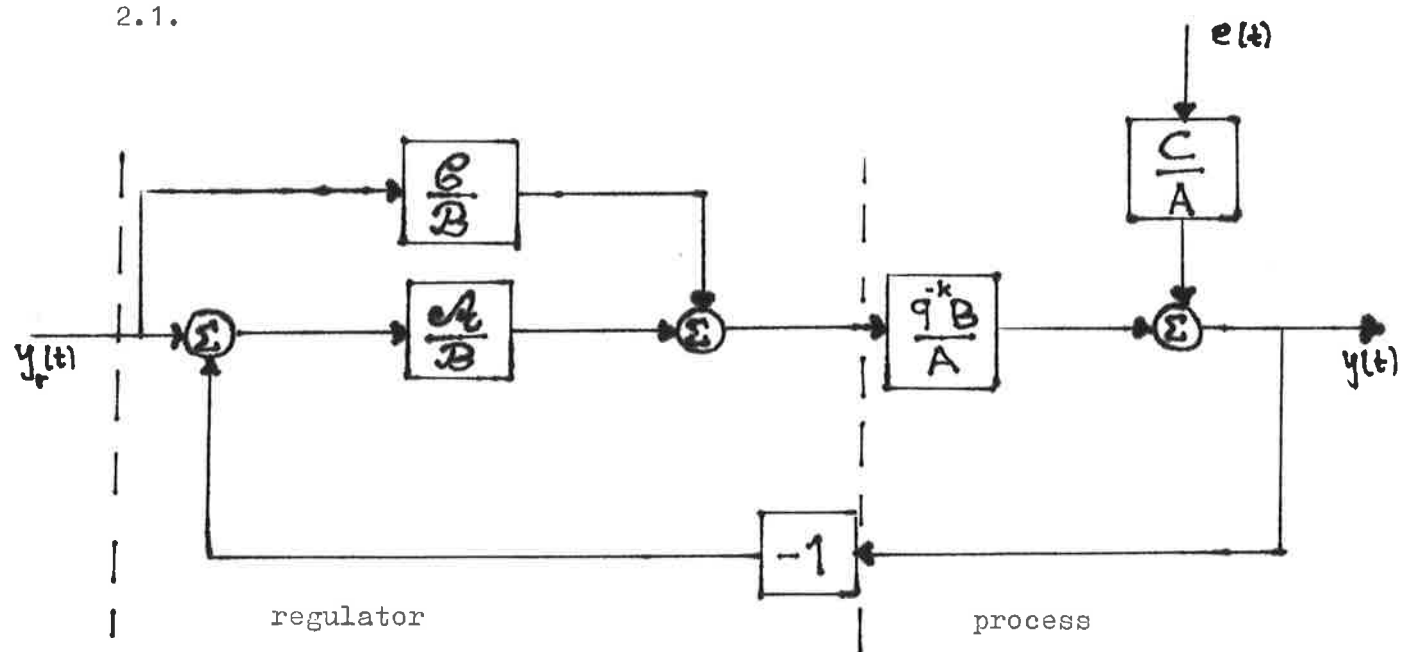


Fig 2.1. Blockschema av Wittenmarks algoritm.

Den självinställande algoritmen kan beskrivas av

Steg 1:

I varje samplingsintervall t estimeras parametrarna i modellen:

$$y(t) - y_r(t) = -\alpha_1 [y(t-k-1) - y_r(t-k-1)] - \dots - \alpha_m [y(t-k-m) - y_r(t-k-m)] + \beta_0 [u(t-k-1) + \beta_1 u(t-k-2) + \dots + \beta_l u(t-k-l-1)] - \gamma_0 y_r(t) - \gamma_1 y_r(t-k-1) - \dots - \gamma_p y_r(t-p) + \varepsilon(t)$$

med användande av LS (minsta kvadrat)-metoden d.v.s. minimera:

$$\sum_{i=0}^t \varepsilon(t_i)^2$$

polynomens storlek bestäms av $m=n$, $l=n+k-1$ och $p=n+k$.

Steg 2:

Sedan beräknas styrsignalen också i varje samplingsintervall, med användande av de estimerade parametrarna i steg 1 till:

$$u(t) = \frac{1}{\beta_0} \sum_{i=1}^m \alpha_i [y(t-i+1) - y_r(t-i+1)] - \sum_{i=1}^l \beta_i u(t-i) + \frac{1}{\beta_0} \sum_{i=0}^p \gamma_i y_r(t+k-i+1)$$

Om parametrarna konvergerar till den optimala regulatorn skall dessa vara:

$$\mathcal{A} = -G, \quad \mathcal{B} = BF \quad \text{och} \quad \mathcal{C} = AF$$

där G - och F -polynomen erhålles ur identiteten:

$$C = AF_k + q^{-(k+1)} G_{n-1}$$

$$F_k = 1 + f_1 q^{-1} + \dots + f_k q^{-k}$$

$$G_{n-1} = g_0 + g_1 q^{-1} + \dots + g_{n-1} q^{-(n-1)}$$

Algoritm 2.

Denna metod representeras av blockschemat i figur 2.2.

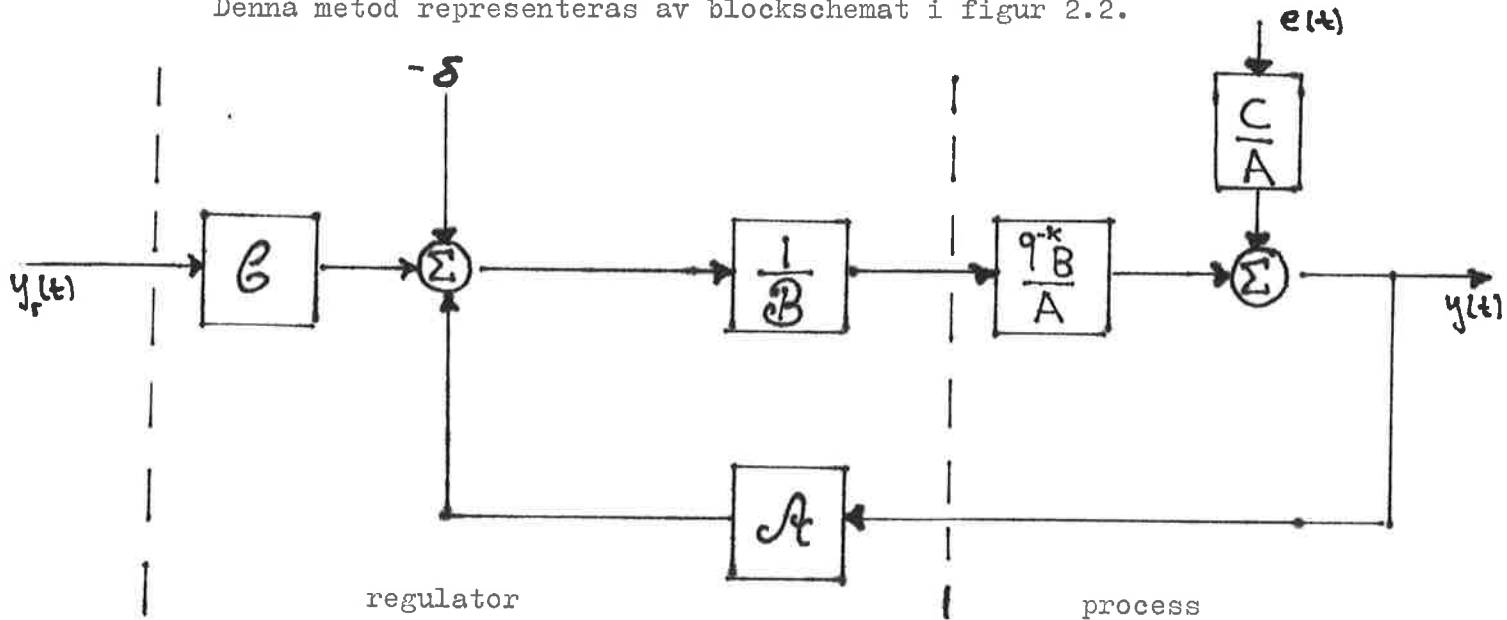


Fig. 2.2. blockshema över Clarke-Gawthrops algoritm.

Självinställande algoritmen består även denna av två steg.

Steg 1.

I varje samplingsintervall t , estimeras parametrarna ur modellen

$$y(t) - y_r(t) = -\alpha_1 y(t-k-1) - \dots - \alpha_m y(t-k-m) + \\ + \beta_0 [u(t-k-1) + \beta_1 u(t-k-2) + \dots + \beta_l u(t-k-l)] - \\ - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) - \dots - \gamma_p y_r(t-p) + \delta + \varepsilon(t)$$

med användande av LS-metoden.

Steg 2.

Sedan beräknas m.h.a. de estimerade parametrarna ur steg 1, styrsignalen till:

$$u(t) = \frac{1}{\beta_0} \sum_1^m \alpha_i y(t-i+1) - \sum_1^l \beta_i u(t-i) + \\ + \frac{1}{\beta_0} \sum_0^p \gamma_i y_r(t+k-i+1) - \frac{\delta}{\beta_0}$$

$m=n$, $l=n+k-1$ och $p=n_c$

Om parametrarna konvergerar mot den optimala regulatorn skall dessa vara:

$$A = -G, \quad B = BF \text{ och } C = C$$

där F-och G-polynomen erhålles som tidigare.

δ =nivå som estimeras fram.

Om vi tittar närmare på de bägge metoderna så ser vi att, de kombinerar återkoppling av reglerfelet med framkoppling av referensvärdet. Detta syns om vi skriver styrsignalen på formen:

$$u(t) = -\frac{S}{R} y(t) + \frac{T}{R} y_r(t+k+1)$$

där

$$S=G, \quad R=BF \text{ och } T=C$$

I algoritm 2, tillkommer en term $\frac{1}{8}\delta$ p.g.a. nivån som estimeras.

Den stora skillnaden algoritmerna emellan är förutom nivån som estimeras, att algoritm 2 utnyttjar ett θ -polynom av lägre ordning än algoritm 1, för att undvika stationära fel förutsätter denna algoritm en integrator i processen, medan nivån i algoritm 2 har i uppgift att eliminera ev. stationära fel.

Det återkopplade systemet blir i bägge fallen:

$$y(t) = F_k e(t) + y_r(t) \quad ; \quad e(t) \in N(0, \sigma)$$

med ett reglerfel utav

$$E(y(t) - y_r(t))^2 = \sigma^2 (1 + f_1^2 + \dots + f_k^2)$$

Vidare förutsetts att polynomet $q^*(z)$ har samtliga nollställen innanför enhetscirkeln.

För att kunna använda algoritmerna on-line, används en rekursiv form av LS-estimeringen, beskriven av ekvationerna;

$$\theta(t+1) = \theta(t) + P(t+1)z(t+1)\varepsilon(t+1)$$

$$P(t+1) = \left[P(t) - \frac{P(t)z(t+1)\varphi(t+1)^T P(t)}{\lambda(t+1) + \varphi(t+1)^T P(t)z(t+1)} \right] / \lambda(t+1)$$

$$\lambda(t+1) = \lambda(t)\lambda_0(t) + (1 - \lambda_0)$$

där

$\theta(t)$ är parametervektor, $P(t)$ kovariansmatris, $\varphi(t)$ och $z(t)$ vektorer av gamla värden utav $u(t)$, $y(t)$, $\varphi_r(t)$ och $\varepsilon(t)$, $\varepsilon(t)$ definierade av tidigare modeller och $\lambda(t)$ är en glömskefaktor som medför att vi minimerar

$$\sum_{n=0}^t \lambda^{t-n} \varepsilon^2(n) \quad \text{istället för} \quad \sum_{n=0}^t \varepsilon^2(n)$$

Detta har fördelen att vi glömmar eventuella vilda regleringar i initialskedet, vikten blir 0.1 efter ungefär $\frac{2.3}{\lambda-1}$ steg, då regleringen blivit bättre, kan man genom att införa en glömskeprofil λ_0 få $\lambda \rightarrow 1$.

3. SIMULERADE EXEMPEL.

Då vi använder algoritmerna för servoproblemet så kommer som redan nämnts, problemet med stationära fel in. Wittenmark föreslår i algoritm 1, att en integrator kan införas genom att reglera med $\nabla u(k) = u(k) - u(k-1)$ istället för $u(k)$.

Clarke-Gawthrop löser detta genom att estimerar fram en bias-term δ .

Simuleringarna skall då jämföras med om vi ej har en integrator i processen och om vi inför en genom att kaskadkoppla med en integrator.

Vidare skall vi se hur algoritmerna fungerar då vi inför en konstant biasterm=1 i styrsignalen och reglerar genom att utöka algoritmernas gradtal, för att på så sätt ge dessa möjlighet att estimerar fram en integrator.

Detta försök har betydelse då man ej vet om en yttre konstant störning påverkar systemet eller om vi har ett medelvärdesfel i styrsignalen, det senare kommer att framgå då vi försöker reglera ett komplext system utan integrator i processen.

I algoritmerna förekommer två parametrar β_0 och δ_0 , som anses kända på förhand, man kan då välja mellan att estimerar en av dessa och låta den andra vara konstant, vanligtvis väljer man $\delta_0 = 1$ och estimerar β_0 . Om man känner β_0 något så när, kan man kanske använda ett approximativt värde och försöka klara sig utan att estimerar denna. I detta fall kan konvergenssvårigheter uppstå.

Jag skall undersöka hur regleringen påverkas då man varierar parametern β_0 och se hur pass känslig regleringen är för detta och jämföra detta med då β_0 estimeras.

OBS.

Tyvärr hade ett programmeringsfel insmugit sig i Gustavssons programpaket och detta upptäcktes ej förrän i ex. arbetets slutskede, då det var för sent att göra om körningarna. Det är dock endast δ -polynomet som har påverkats av detta, det återkopplade systemets dynamik och utsignal har lyckligtvis varit opåverkade. Det är ett skalningsfel, förorsakad av parametern β_0 i algoritm I, då denna är $\neq 1$.

EXEMPEL 1. (1:a ordningens process, utan integrator).

Studera systemet:

$$y(t) + a_1 y(t-1) = b_1 u(t-2) + e(t) + c_1 e(t-1)$$

där

$$a_1 = 0.9$$

$$b_1 = 0.5$$

$$c_1 = -0.5$$

detta system är alltså minimumfas och innehåller ej en integrator. Minimalvariansstyrningen ges utav uttrycket: (algoritm 1).

$$u(t) = \frac{-0.36}{0.5(1+0.4q^{-1})} [y(t) - y_r(t)] + \frac{1 - 0.5q^{-1} - 0.36q^{-2}}{0.5(1+0.4q^{-1})} y_r(t+2)$$

En regulator med denna struktur kan erhållas genom att identifiera parametrarna i modellen:

$$y(t) - y_r(t) = -\alpha_1 [y(t-2) - y_r(t-2)] + \beta_0 [u(t-2) + \beta_1 u(t-3)] - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) - \gamma_2 y_r(t-2) + \varepsilon(t)$$

Vid simuleringen har följande värden använts.

Glömskefaktor $\lambda = 0.98$ och $\lambda_0 = 1$, $P = 100 \times I$, $\alpha_i(0) = \beta_i(0) = \gamma_i(0) = 0$
 $\gamma_0 = 1$ och $\beta_0 = 0.5$.

Den optimala regulatorns varians väntas bli:

$$V = \frac{1}{N} \sum_{t=1}^N (y(t) - y_r(t))^2 = 1.16.$$

Ur kurvorna över varians och utsignal (fig. 3.1 och 3.2), följer att regleringen fungerar utmärkt. Variansen blir 1.15 per steg beräknad över intervallet 10-250 steg, utsignalen håller sig väl inom standardavvikelsen $\sigma = 1.1$.

Som synes av kurvorna över parameterestimeringarnas tidsförlopp (fi 3.3), (där jag använt en glömskeprofil på $\lambda_0 = 0.99$ och $\lambda = 0.99$), så stämmer α och β - väl överens med den optimala regulatorns, medan däremot γ -polynomet ej stämmer. Detta beror på det tidigare nämnda programmeringsfelet, för att övertyga läsaren om detta har en simulering gjorts med $\beta_0 = 1$ och $\lambda_0 = 0.999$ samt $\lambda = 0.99$.

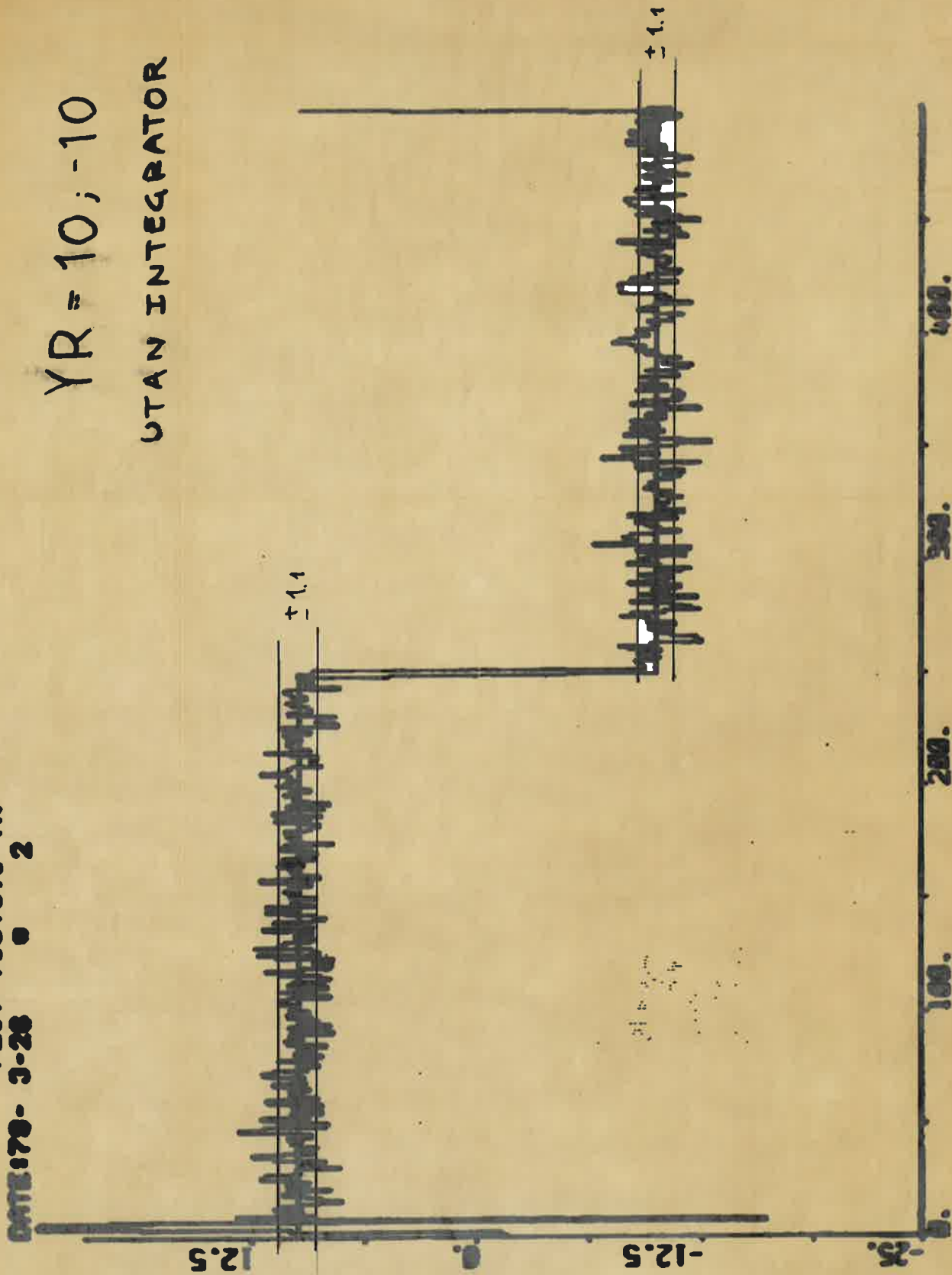
Estimeringen startades i rätt punkt. I figur 3,4 har dessutom överföringsfunktionen från y till y_r plotats. Som synes har parametrarna konvergerat mot den optimala efter ca:5000 steg, medan y har erhållit rätt stationärt värde efter redan ett par steg i initialskedet. (dc-parametern).

B.W. - ALGORITHM

YR = 10; -10

UTAN INTEGRATOR

DATE: 179-3-28
PLOT YESYS13 YR
2



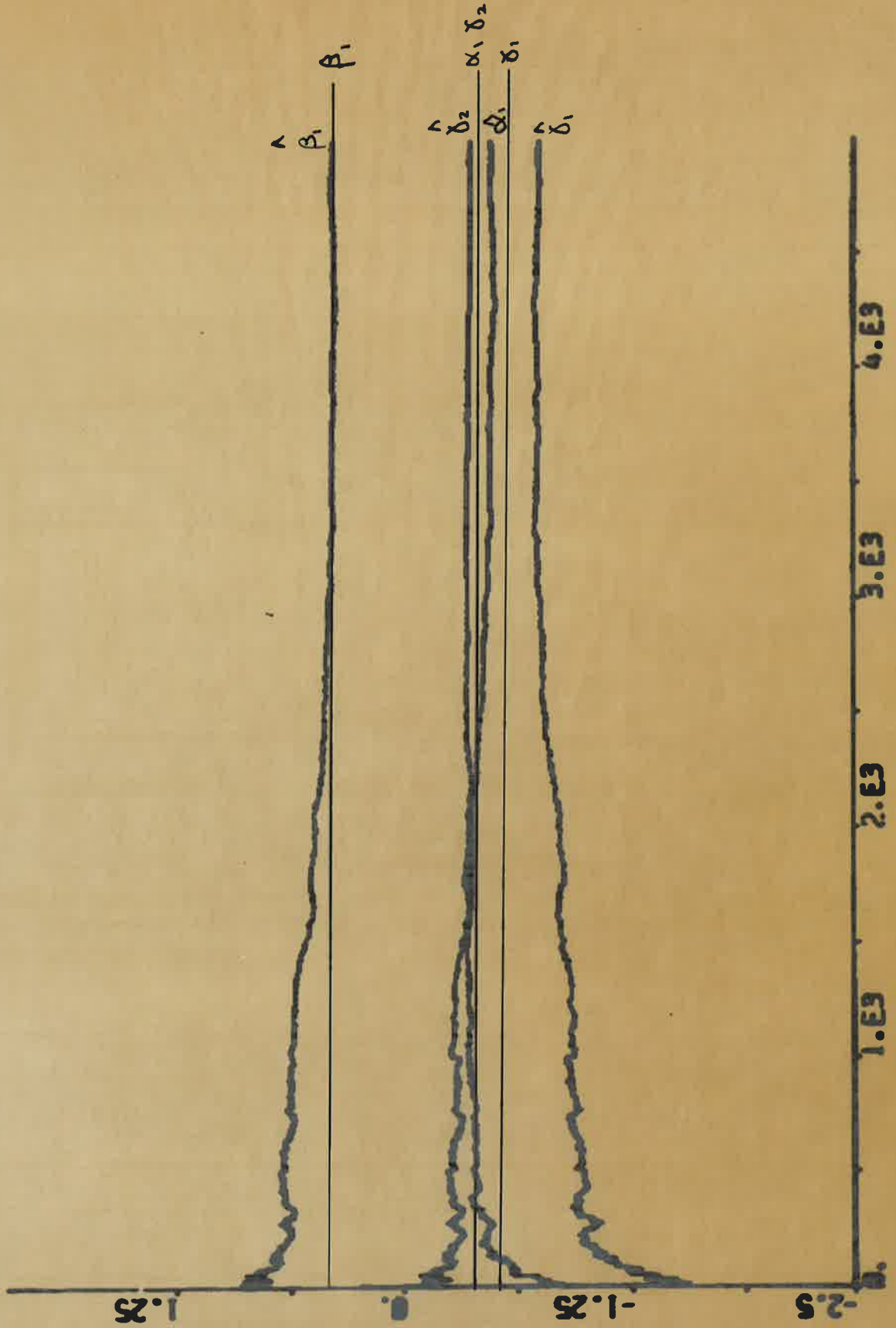
Figur 3.4. Ut-signal då vi ej har integrator i processen. Destreckade linjerna

PLOT V



Figur 3.2. Varians för utsignalen i fig. 3.1. Denna blir omkring 1.14 per steg

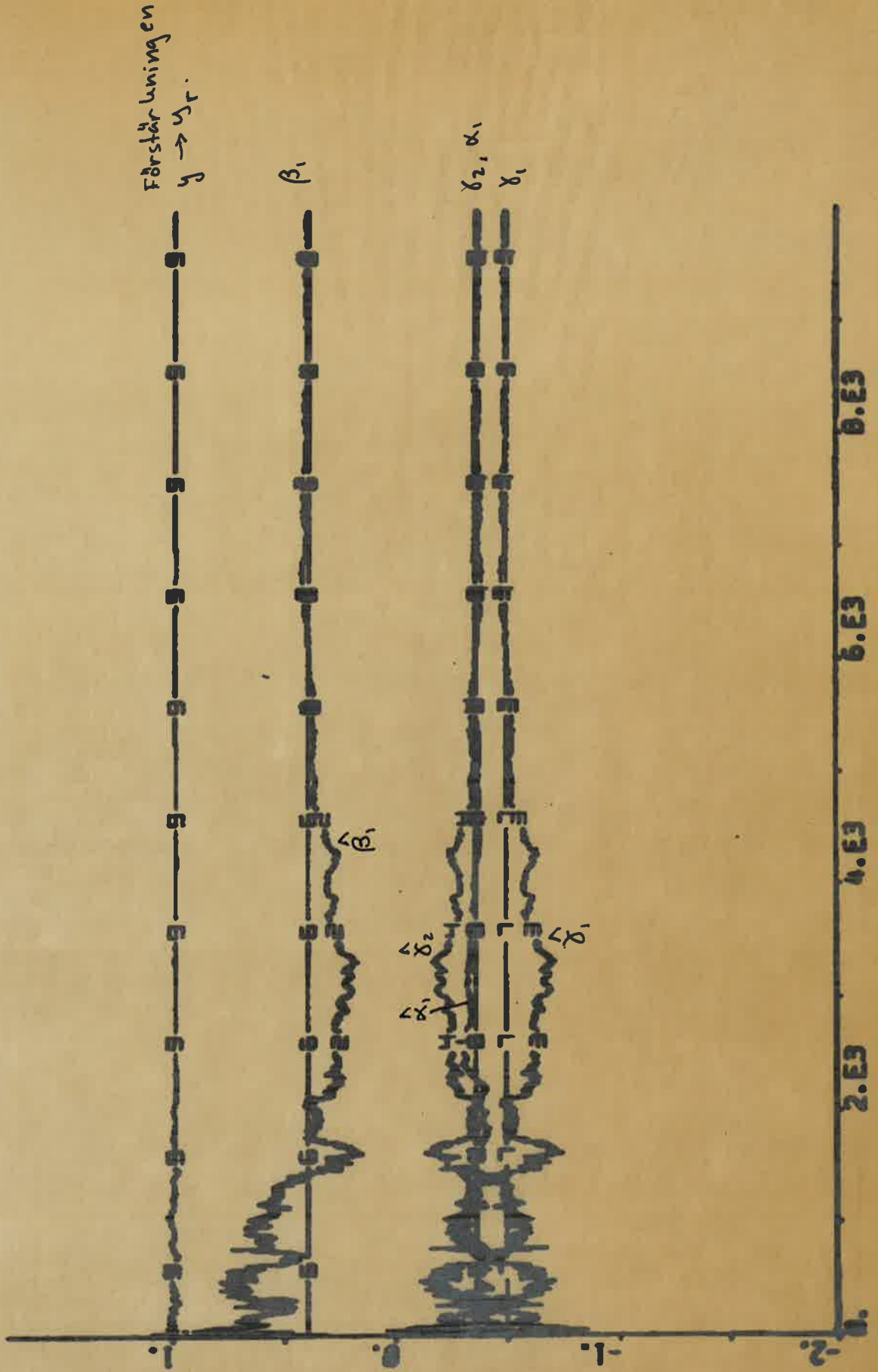
PLOT TH1 TH2 TH3 TH4



Figur 3.3. Estimeringens tidsförlopp, observera att β -polynomets ej stämmer detta $\lambda_0 = \lambda = 0.99$

PLOT TH1 TH2 TH3 TH4 TH51 TH52 TH53 TH54 DC

DATE: 0-0-0 10



Figur 3.4. Samma system som innan, men med $\beta_0 = 1 (= b_1)$. Uppstart i rätt punkt

1.2.

Samma system som tidigare (ex.1.1). I algoritm 2 ändras modellen till:

$$y(t) - y_r(t) = -\alpha_1 y(t-2) + \beta_0 [u(t-2) + \beta_1 u(t-3)] - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) + \delta + \varepsilon(t)$$

Den optimala regulatorn blir då,

$$u(t) = \frac{-0.36}{0.5(1+0.4q^{-1})} y(t) + \frac{q^2 - 0.5q}{0.5(1+0.4q^{-1})} y_r(t) - \frac{\delta}{0.5(1+0.4q^{-1})}$$

denna metod förutsätter ej någon integrator i processen, ty bias-termen δ estimeras.

Vid simuleringen användes:

$$\lambda = 0.98, \lambda_0 = 1, \alpha_1(0) = \beta_1(0) = \gamma_1(0) = 0 \text{ och } P(0) = 100 \times I.$$

Den optimala regulatorns varians väntas bli:

$$V = \frac{1}{N} \sum_{i=0}^T (y(t_i) - y_r(t_i))^2 = 1.16$$

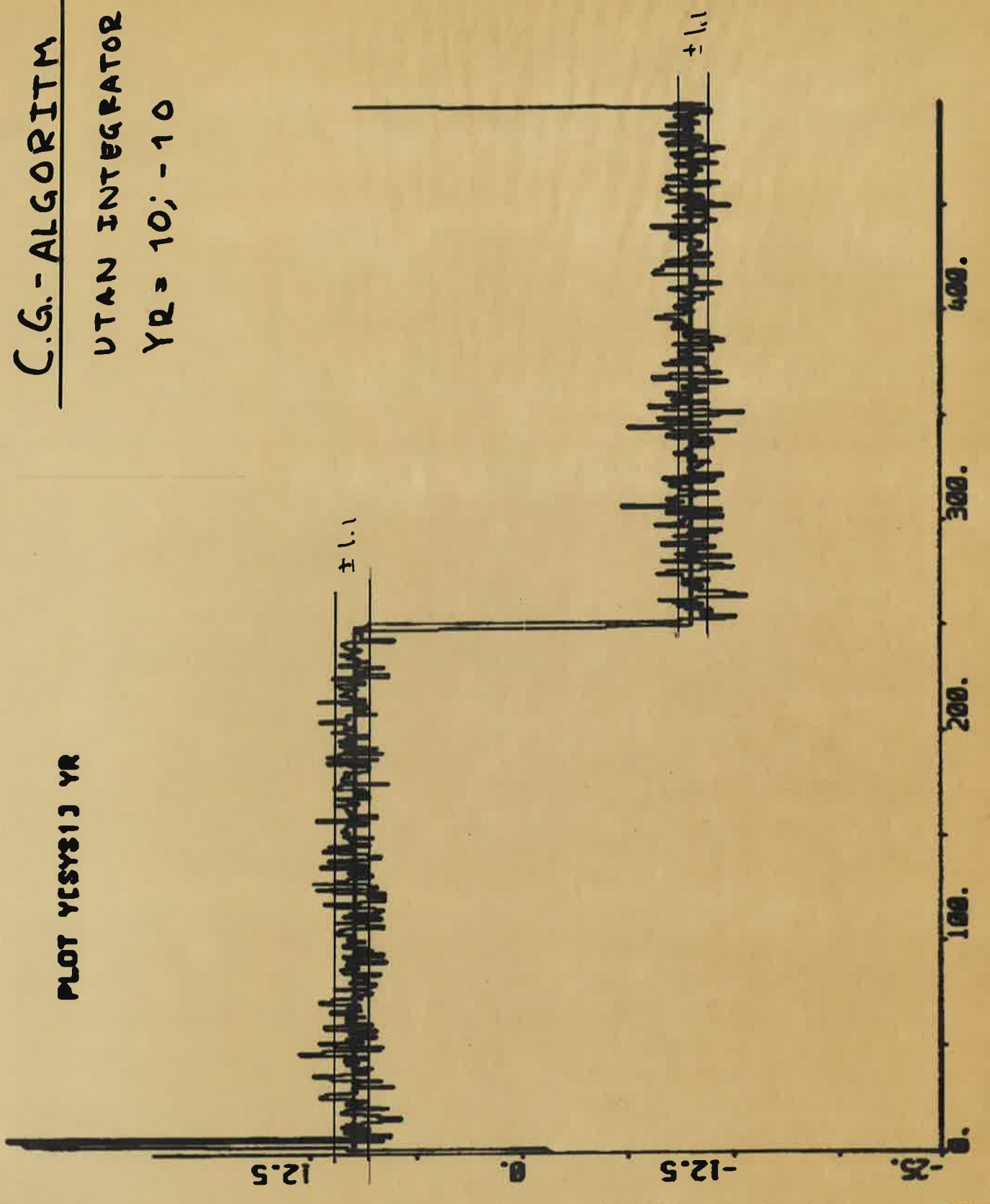
Simuleringen gav goda resultat, utsignalen håller sig väl inom de teoretiska gränserna, variansen över intervallet 10-250 steg blev 1.15 per steg (fig.3.7).

Parameterestimeringen (fig.3.8) konvergerade in efter ca; 2000 steg d.v.s. långsamt insvängningsförlopp. Vid upptagningen användes glömskeprofilen $\lambda = \lambda_0 = 0.99$.

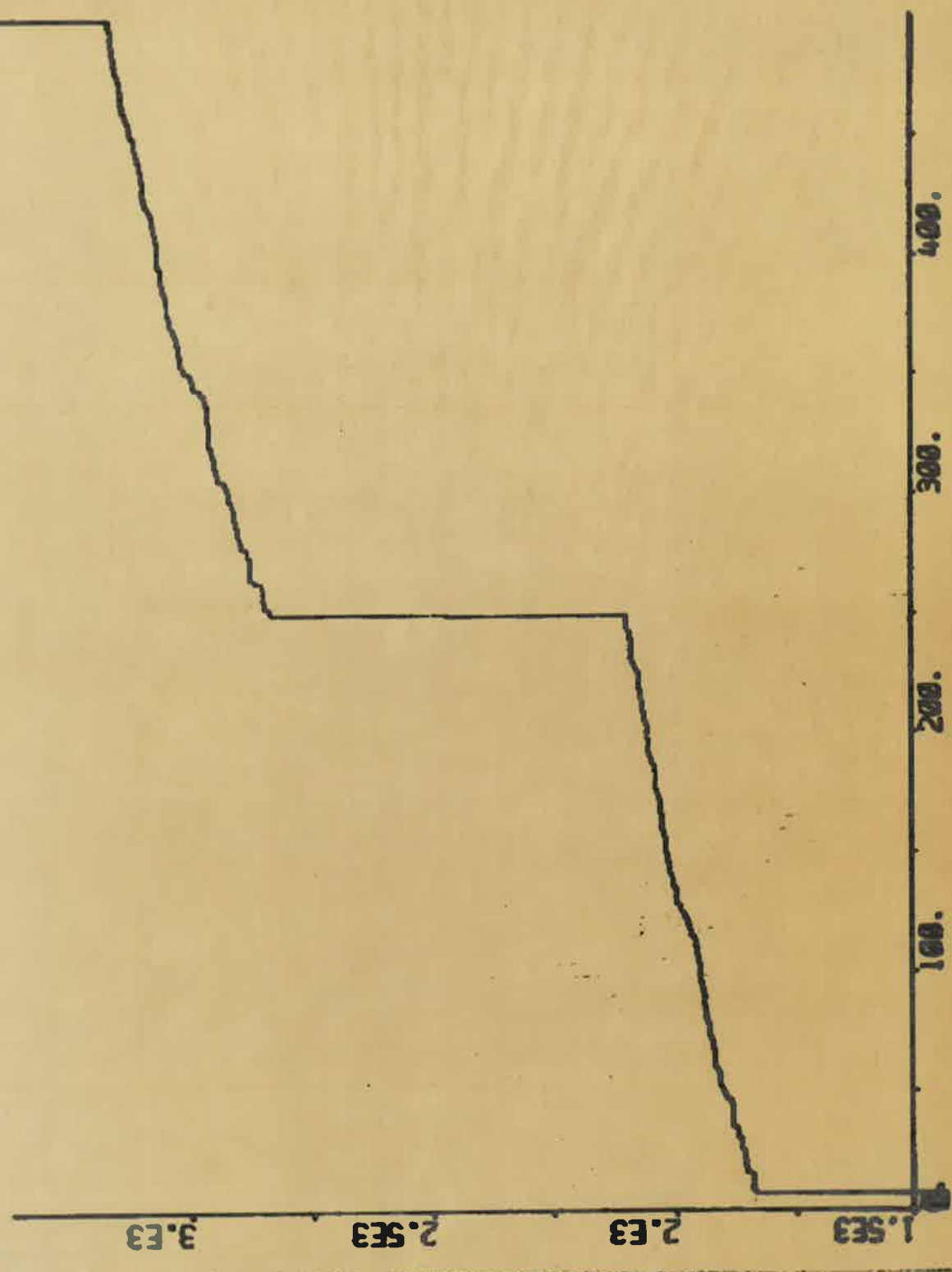
03.22
19.01.1974

PLOT YESYS1J YR

C.G.-ALGORITM
UTAN INTEGRATOR
YR = 10; -10



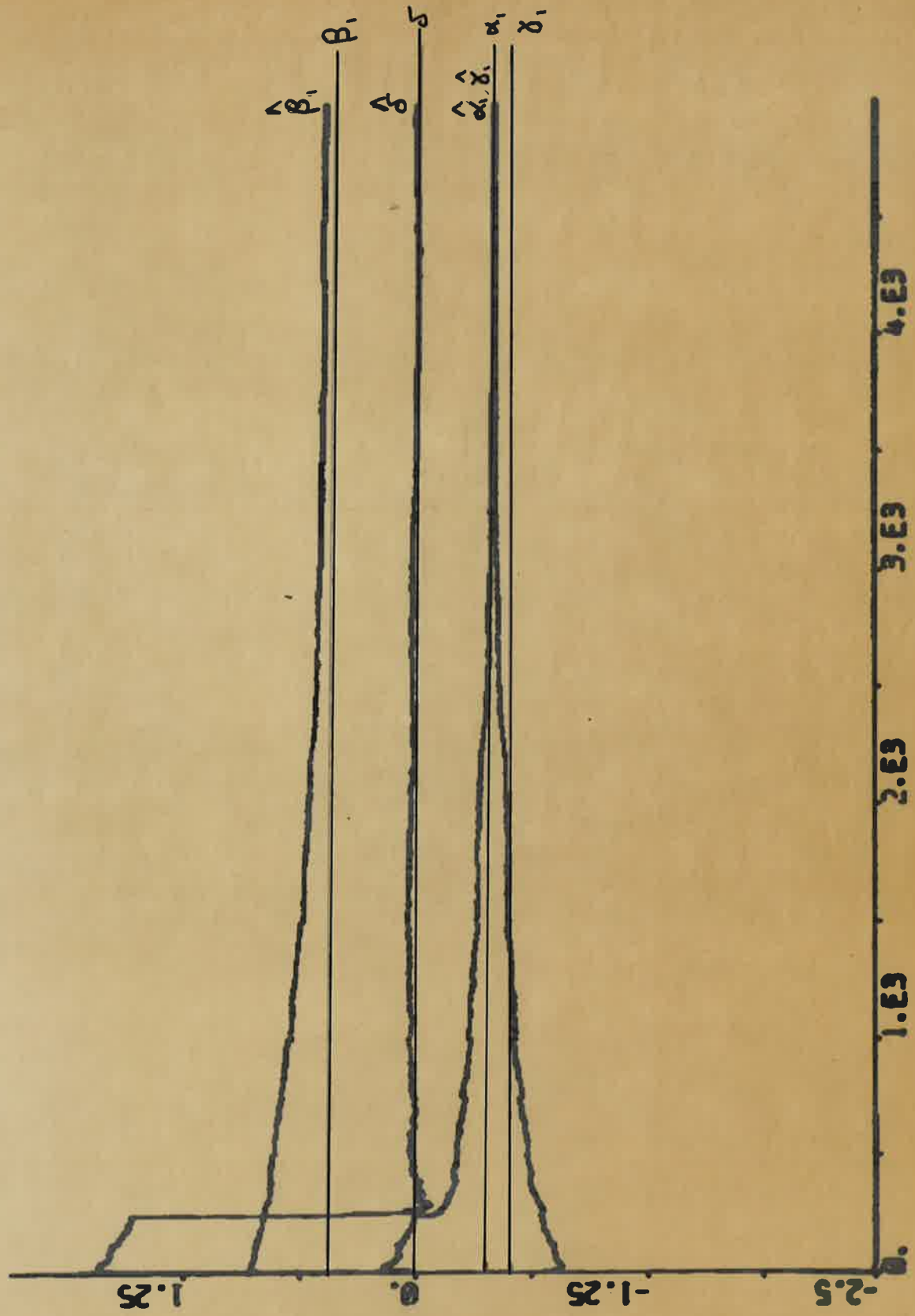
Figur 3.6. Utsignal från samma system som i figur 3.1. men nu styrd av algoritm 2. 16



PLOT V

Figur 3.7. Varians för utsignalen i fig. 3.6. Variansen blir omkring 1.15 per steg. 77

PLOT TH1 TH2 TH3 TH4



Figur 3.8. Estimeringens tidsförlopp. Programmeringsdellet inverterar α och δ denna algoritim Gjömske profil $\lambda_0 = \lambda = 0.99$

I.3. (2:a ordningens system.)

Betrakta systemet

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = u(t-2) + b_1 u(t-3) + e(t) + C_1 e(t-1) + C_2 e(t-2)$$

där

$$a_1 = -1.5 \quad b_1 = 0.5 \quad C_1 = -1$$

$$a_2 = 0.7 \quad C_2 = 0.1$$

Detta system skall regleras med algoritmen¹, där minimalvariansregulatorn ges utav uttrycket:

$$u(t) = - \frac{0.15 - 0.35q^{-1}}{1 + q^{-1} + 0.25q^{-2}} (y(t) - y_r(t)) + \frac{1 - q^{-1} - 0.05q^{-2} + 0.35q^{-3}}{1 + q^{-1} + 0.25q^{-2}} y_r(t+2)$$

Den optimala regulatorns varians blir 1.25 per steg.

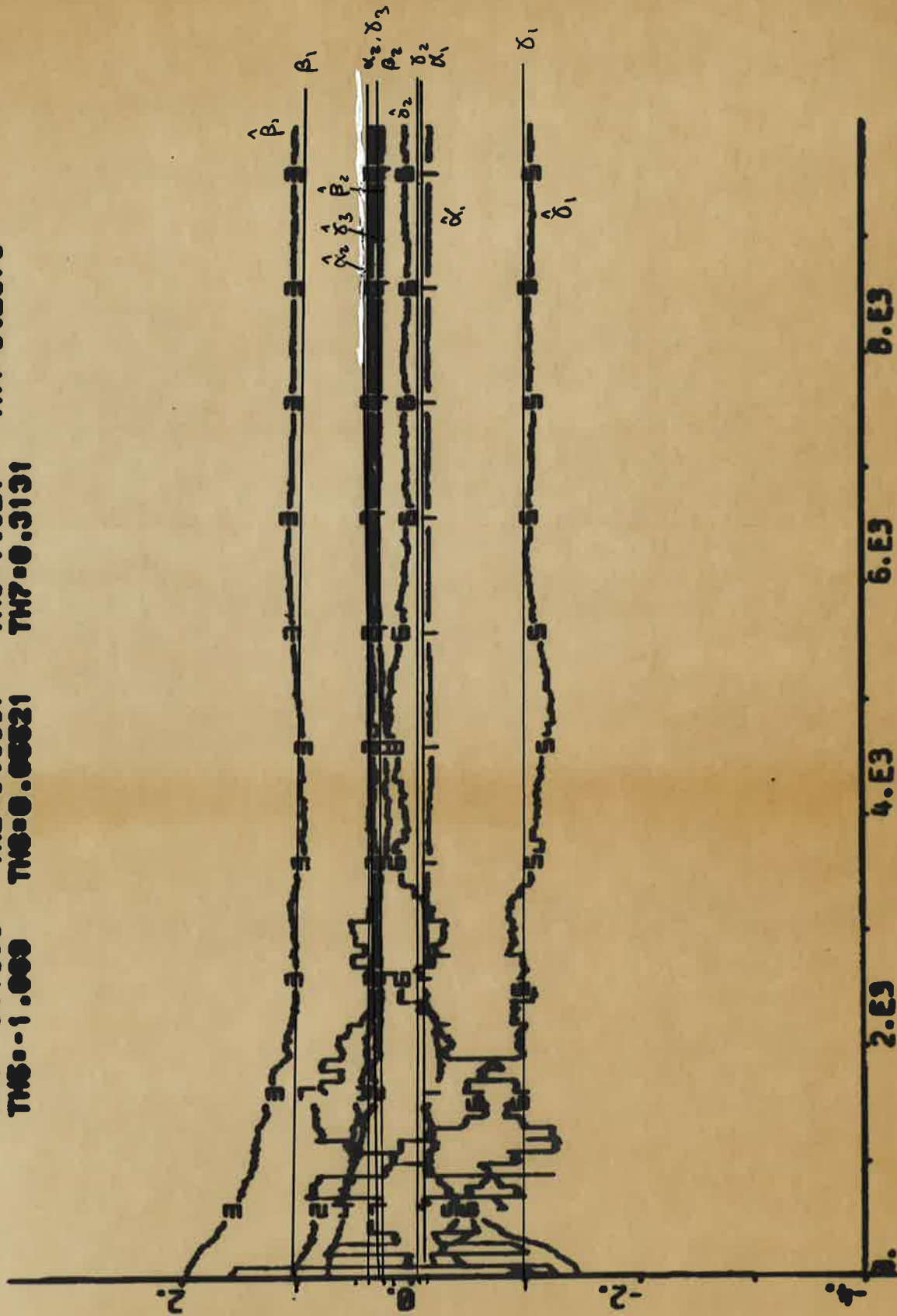
En regulator med denna struktur kan erhållas med modellen:

$$\begin{aligned} y(t) - y_r(t) = & -\alpha_1 [y(t-2) - y_r(t-2)] - \alpha_2 [y(t-3) - y_r(t-3)] + \\ & + \beta_0 [u(t-2) + \beta_1 u(t-3) + \beta_2 u(t-4)] - \\ & - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) - \gamma_2 y_r(t-2) - \gamma_3 y_r(t-3) + \varepsilon(t) \end{aligned}$$

Figur 3.8² visar parameterkonvergensen, vid simuleringen användes glömskeprofilen $\lambda_0 = \lambda = 0.99$.

Detta exempel vill visa att algoritmen konvergerar då vi kringgår programmeringsfelet genom att välja $\beta_0 = 1$.

DATE: 0-0-0 12
 PLOT TH1 TH2 TH3 TH4 TH5 TH6 TH7
 TH1:-0.1005 TH2:0.3007 TH3:1.021 TH4:0.2573
 TH5:-1.003 TH6:0.00321 TH7:0.3131



Figur 3.8.² 2:a ordningens process, styrd med algoritim 1. Vi vill visa att algoritmen konvergerar då $\beta_0 = 1$, ty då inverkar ej programmeringsfelet.

EXEMPEL 2 (tredje ordningens process, utan integrator)

Betrakta systemet

$$y(t) + a_1 y(t-1) + a_2 y(t-2) + a_3 y(t-3) = b_1 u(t-2) + b_2 u(t-3) + e(t) + c_1 e(t-1) + c_2 e(t-2)$$

där

$$\begin{aligned} a_1 &= -2.4 & b_1 &= 0.5 & c_1 &= -1 \\ a_2 &= 2.05 & b_2 &= 0.25 & c_2 &= 0.2 \\ a_3 &= -0.63 \end{aligned}$$

Systemet är minimumfas och innehåller en tidsfördröjning $k=1$, liksom i exempel 1 skall simulera både med algoritm 1 och 2 och nu närmare studera hur konvergensen utvecklar sig med tiden.

2.1. (algoritm 1)

Minimalvarians-regulatorn ges utav uttrycket

$$u(t) = \frac{-1.51 + 2.24q^{-1} - 0.882q^{-2}}{0.5(1 + 1.9q^{-1} + 0.7q^{-2})} (y(t) - y_r(t)) + \frac{1 - q^{-1} - 1.31q^{-2} + 2.24q^{-3} - 0.882q^{-4}}{0.5(1 + 1.9q^{-1} + 0.7q^{-2})} y_r(t-4)$$

där den optimala variansen väntas bli

$$V = \frac{1}{N} \sum_{k=0}^t (y(t_k) - y_r(t_k))^2 = 2.96$$

En regulator med denna struktur kan erhållas med modellen:

$$\begin{aligned} y(t) - y_r(t) &= -\alpha_1 [y(t-2) - y_r(t-2)] - \alpha_2 [y(t-3) - y_r(t-3)] - \alpha_3 [y(t-4) - y_r(t-4)] \\ &+ \beta_0 [u(t-2) + \beta_1 u(t-3) + \beta_2 u(t-4) + \beta_3 u(t-5)] - \\ &- \delta_0 y_r(t) - \delta_1 y_r(t-1) - \delta_2 y_r(t-2) - \delta_3 y_r(t-3) - \delta_4 y_r(t-4) + \\ &+ \varepsilon(t) \end{aligned}$$

Vid simuleringen användes följande parametervärden:

$$\lambda = \lambda_0 = 0.98 \text{ (d.v.s. då går } \lambda \rightarrow 1), P(0) = 100XI \text{ och } \alpha_i(0) = \beta_i(0) = \delta_i(0) = 0.$$

Simuleringen visar (fig.3.9.) att utsignalen följer visserligen referensvärdet, men med en för stor spridning. Tittar man på variansens utveckling med tiden i olika tidsintervall (fig.3.10.)

$$V = \frac{1}{241} \sum_{i=10}^{250} (y(t_i) - y_r(t_i))^2 = 6 \quad V = \frac{1}{246} \sum_{i=10^4}^{10245} (y(t_i) - y_r(t_i))^2 = 4.5$$

$$V = \frac{1}{246} \sum_{i=5000}^{5245} (y(t_i) - y_r(t_i))^2 = 5.3$$

så verkar det troligt att vi har en mycket långsam konvergens mot det korrekta värdet, detta syns också ur parameterestimaten tidsbeteende (fig.3.11.), där \mathcal{A} - och \mathcal{B} -polynomet har konvergerat till de rätta värdena efter ca: 7500 steg.

För att närmare studera den långsamma konvergens skall jag undersöka om $e(t) = y(t) - y_r(t)$ blir ett glidande medelvärde d.v.s. $y(t) - y_r(t) = F e(t)$, där $e(t)$ är vitt brus, för att detta skall gälla måste det slutna systemet uppfylla (se Åström-Wittenmark 1973) följande olikheter

$$\Gamma_e(\tau) = E e(t+\tau) e(t) = 0 \quad \tau \geq k+1$$

$$\Gamma_{eu}(\tau) = E e(t+\tau) u(t) = 0 \quad \tau \geq k+1$$

För att kontrollera detta har jag fortlöpande beräknat summorna

$\sum e(t) e(t-\tau)$ och $\sum e(t) u(t-\tau)$ dessa ger en uppskattning av $\Gamma_e(\tau)$ och $\Gamma_{eu}(\tau)$, om $\Gamma_e(\tau)$ ligger inom gränserna $\pm \frac{1.96}{\sqrt{N}}$ så har vi 5% konfidensnivå för att testet är tillförlitligt, ty vi kan ej vänta att $\Gamma_e(\tau)$ skall bli exakt noll. ($y_r(t)$ är konstant under körningen). I figur 3.12. är $\Gamma_e(\tau)$ plottat och vi ser att denna håller sig väl inom gränserna, t.o.m. $\Gamma_e(1)$ är inom gränserna, detta skulle i så fall tyda på att $y(t) - y_r(t)$ är ett glidande medelvärde av ordning noll.

Däremot om vi tittar på $r_{eu}(\tau)$ (fig. 3.13.) så ser vi att denna ligger ej i närheten utav noll, utan för $k \gg 3$ så har vi $r_{eu}(\tau) \approx -0.5$
För att analysera detta ser vi på uttrycket:

$$\begin{aligned} r_{eu}(\tau) &= E\{ (e(t+\tau) - m_e)(u(t) - m_u) \} = \\ &= \frac{1}{N} \{ \sum e(t+\tau)u(t) - m_e \sum u(t) - m_u \sum e(t) + m_e m_u \} \end{aligned}$$

för att uppskatta m_e , studeras

$$\sum (e(t) + m_e)^2 = \sum e^2(t) + 2m_e \sum e(t) + m_e^2 \cdot N \Rightarrow$$

$$\hat{r}_e(\tau) = \frac{1}{N} \sum e^2(t) + \frac{2m_e}{N} \sum e(t) + m_e^2$$

om $\sum e(t) \approx 0$ kan jag enligt figur 3.13. ta $m_e \approx 0.02$, ty

$$\hat{r}_e(\tau) \approx \frac{1}{N} \sum e^2(t) + m_e^2$$

Detta innebär att

$$\hat{r}_{eu}(\tau) = \frac{1}{N} \{ \sum e(t+\tau)u(t) - m_e \sum u(t) \}$$

statistiska förstärkningen för $u(t)$ blir $\frac{0.75}{0.2 \cdot 0.1} \approx 38$

$$\Rightarrow m_e \sum u(t) \approx 2 \cdot 10^{-2} \cdot N \cdot 38$$

$$\Rightarrow \hat{r}_{eu}(\tau) \approx \frac{1}{N} \sum e(t+\tau)u(t) - 0.76$$

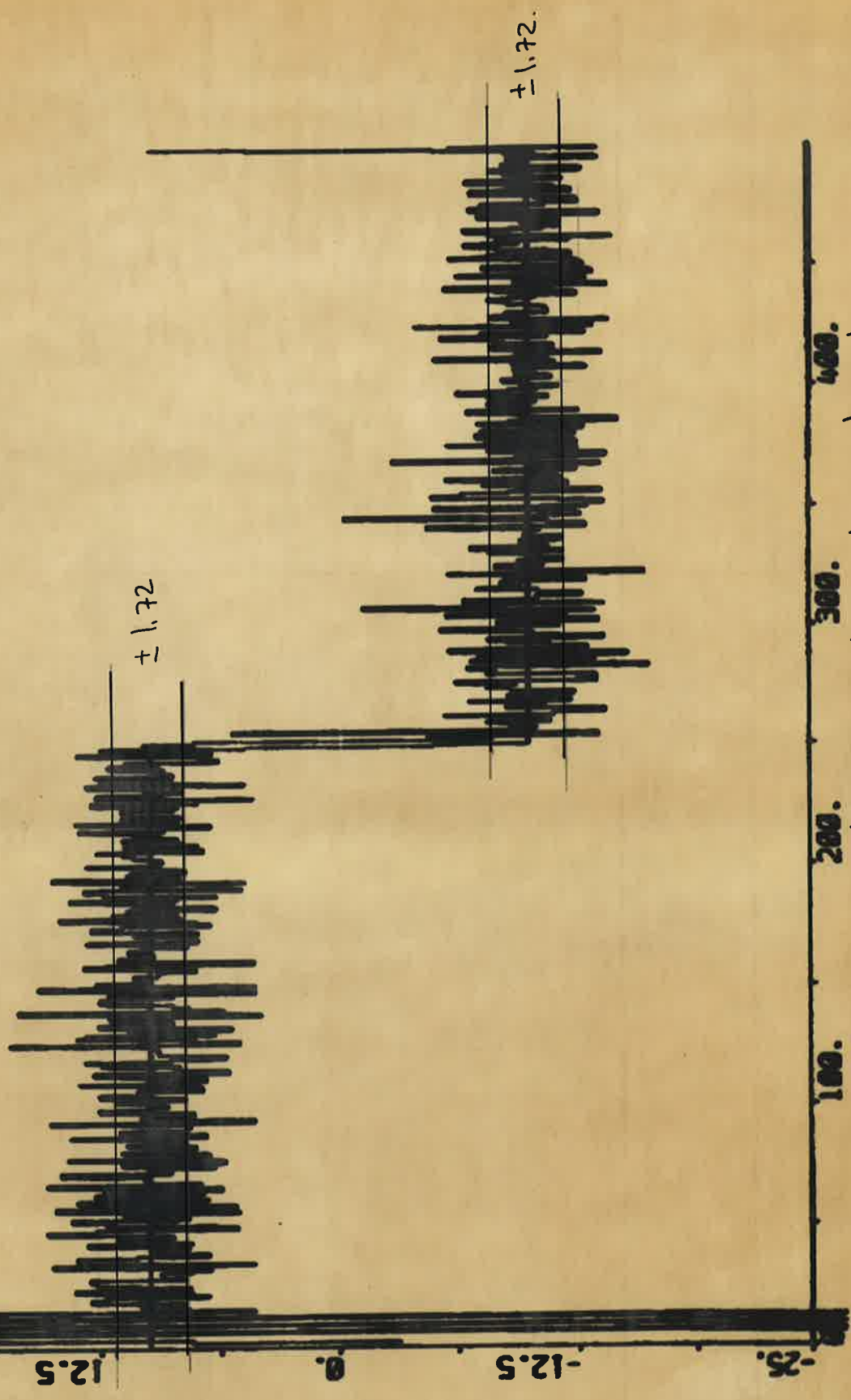
Analysen ovan förklarar varför $r_{eu}(\tau)$ avviker ifrån 0, enligt analysen bör alltså $r_{eu}(\tau)$ antaga värden omkring -0.5, vilket bekräftas av fig. 3.13.

Detta motiverar en undersökning av hur algoritmerna beter sig då systemet påverkas av en konstant störning, detta har gjorts i avsnitt 4.

B.M.S
218
8 p. 2:14 704

B.W.-ALGORITHM

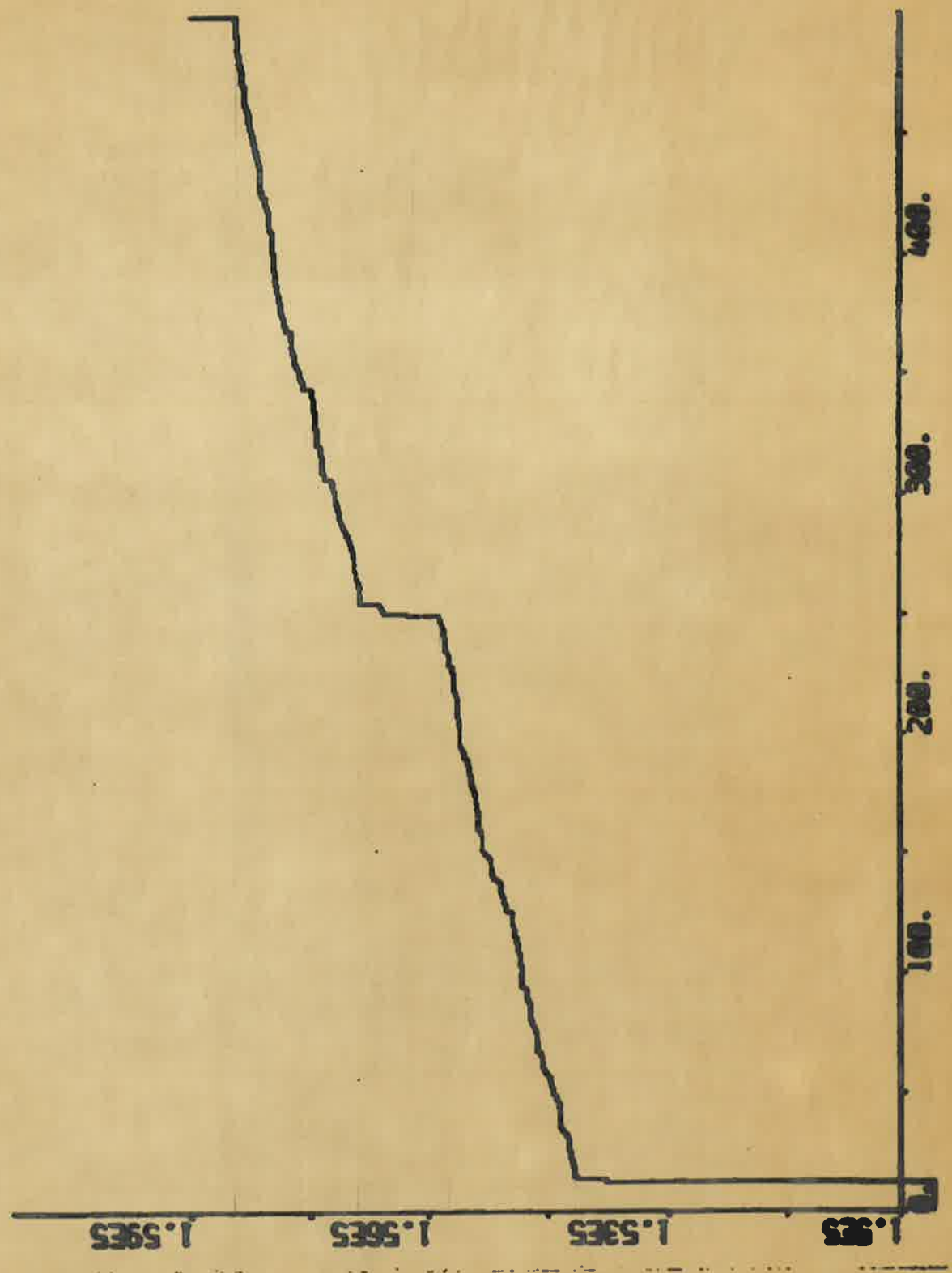
PLOT YESYS1] YR



Figur 3.9. Utsignal från tredje ordningens system, utan integrator.

09.11.8
11.0:14.14

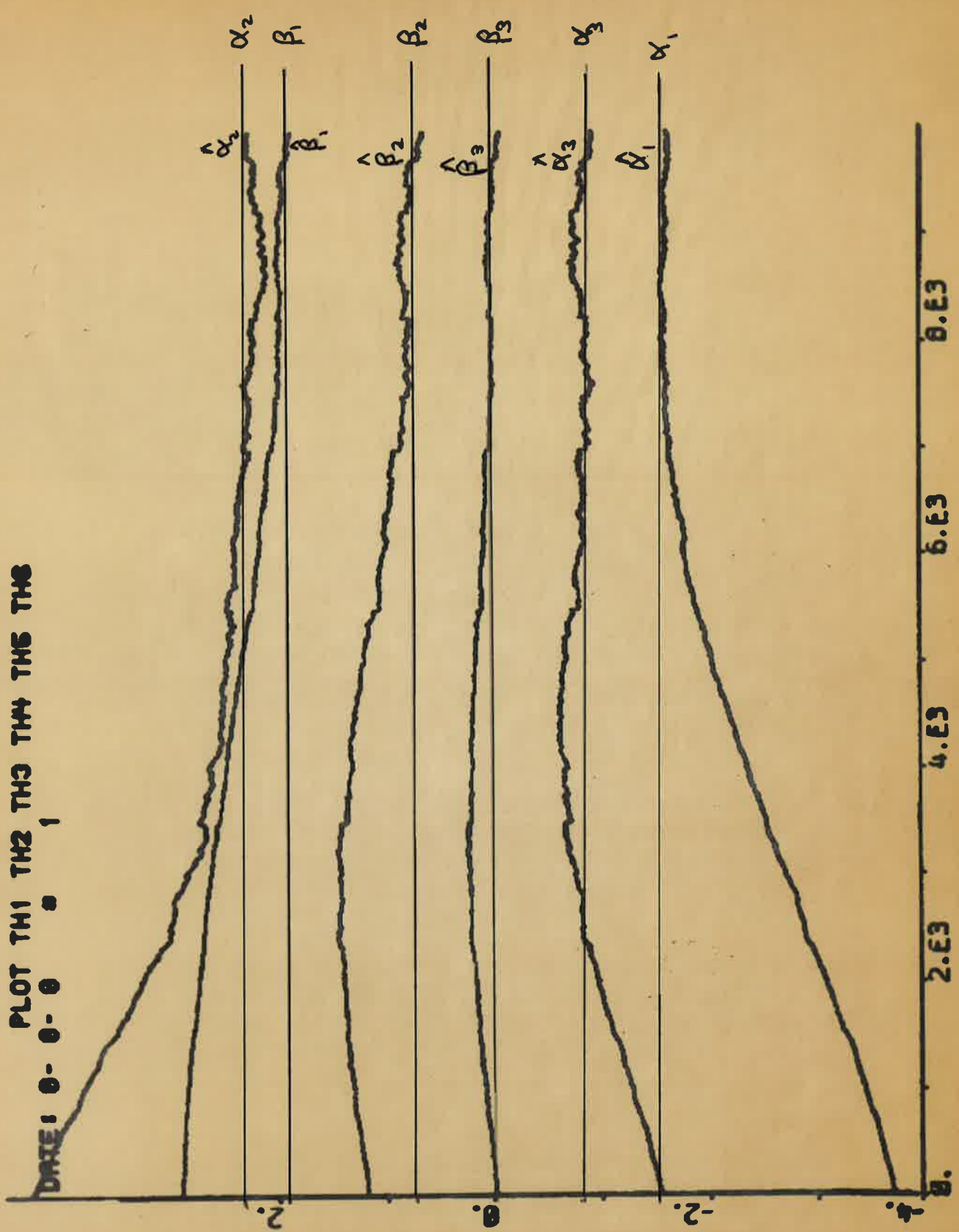
PLOT TH1 TH2 TH3 TH4 TH5 TH6



Figur 3.10. Varfansen för utsignalen i figur 3.9. Under de första stegen 10-250

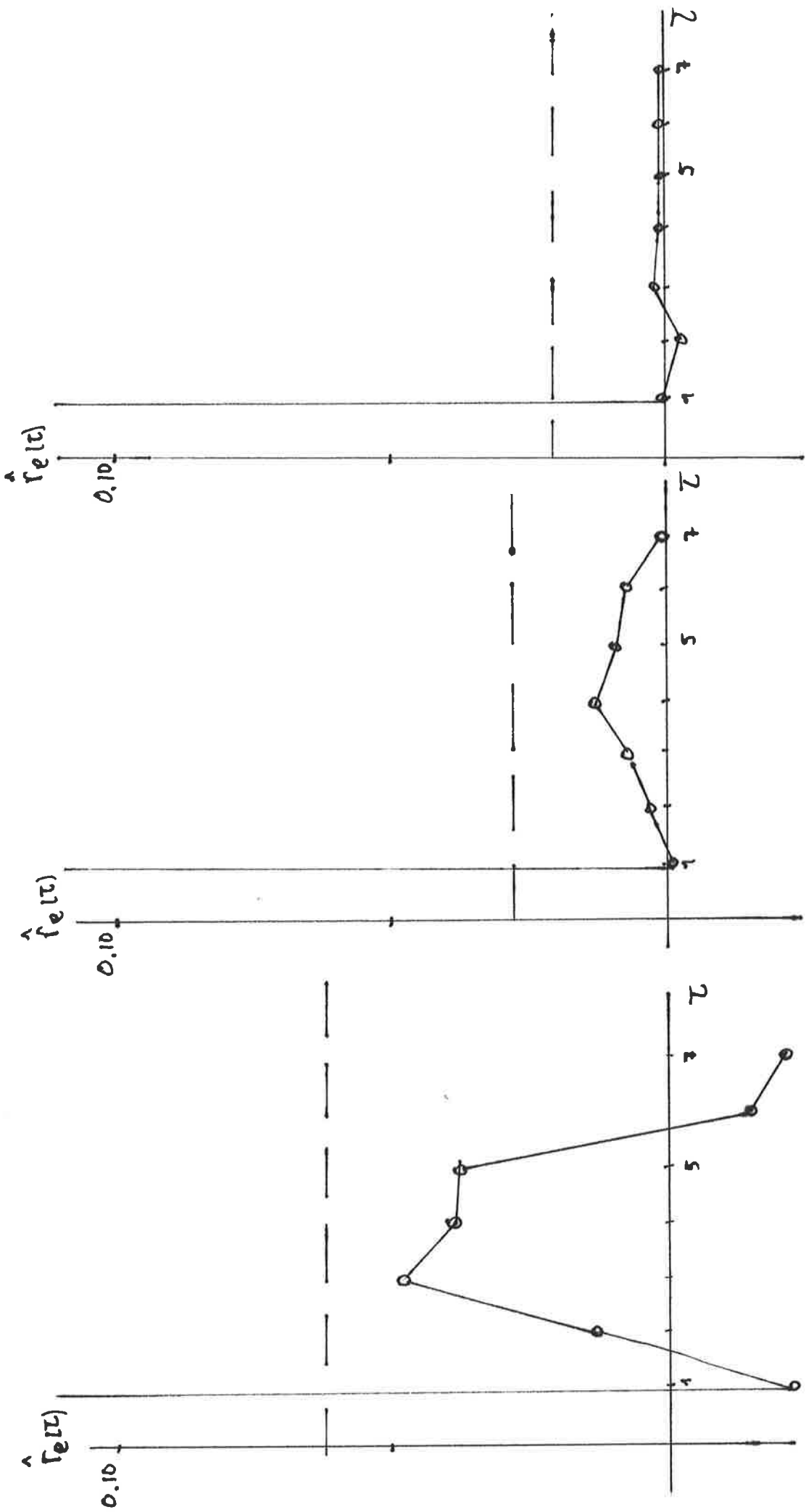
PLOT TH1 TH2 TH3 TH4 TH5 TH6

DATE: 0-0-0 1



Figur 3.11. Estimeringens utseende för α - och β -polynomerna.

PP.0.1170

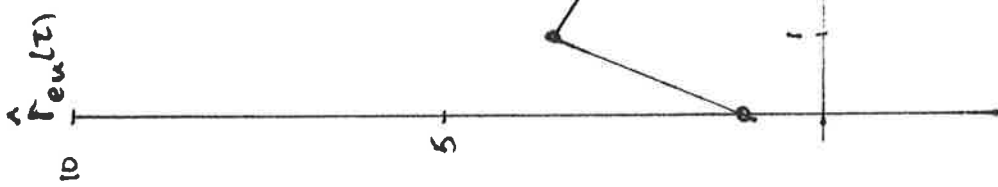


Figur 3.12.

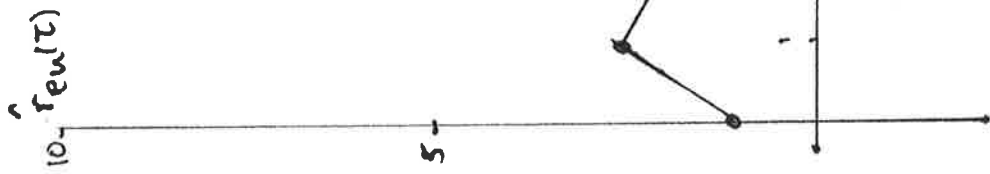
$f_e(t)$ beräknad över 1000-steg
de streckade linjerna är 5%-kon-
fidensnivå (dock endast den övre inritad)

$f_e(t)$ beräknad över 5000-steg

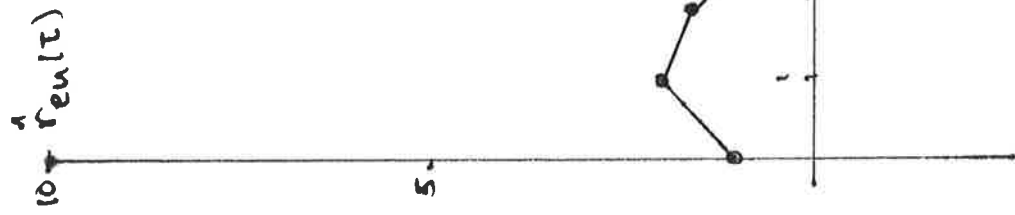
$f_e(t)$ beräknad över 10000-steg



$f_{eu}(t)$ beräknad över 1000-steg



$f_{eu}(t)$ beräknad över 5000-steg



$f_{eu}(t)$ beräknad över 10000-steg

Figur 3.13.

2.2. (Algoritm 2).

Samma system som i 2.1. skall styras, men nu med algoritm 2
Då blir modellen:

$$\begin{aligned}
 y(t) - y_r(t) = & -\alpha_1 y(t-2) - \alpha_2 y(t-3) - \alpha_3 y(t-4) + \\
 & + \beta_0 [u(t-2) + \beta_1 u(t-3) + \beta_2 u(t-4) + \beta_3 u(t-5)] - \\
 & - \delta_0 y_r(t) - \delta_1 y_r(t-1) - \delta_2 y_r(t-2) + \delta + \varepsilon(t)
 \end{aligned}$$

om algoritmen konvergerar mot den optimala regulatorn skall parametrarna bli:

$$\begin{array}{llll}
 \alpha_1 = -1.51 & \beta_1 = 1.9 & \delta_1 = 1 & \delta_0 = 1 \\
 \alpha_2 = 2.24 & \beta_2 = 0.7 & \delta_2 = -0.2 & \beta_0 = 0.5 \\
 \alpha_3 = -0.882 & \beta_3 = 0 & &
 \end{array}$$

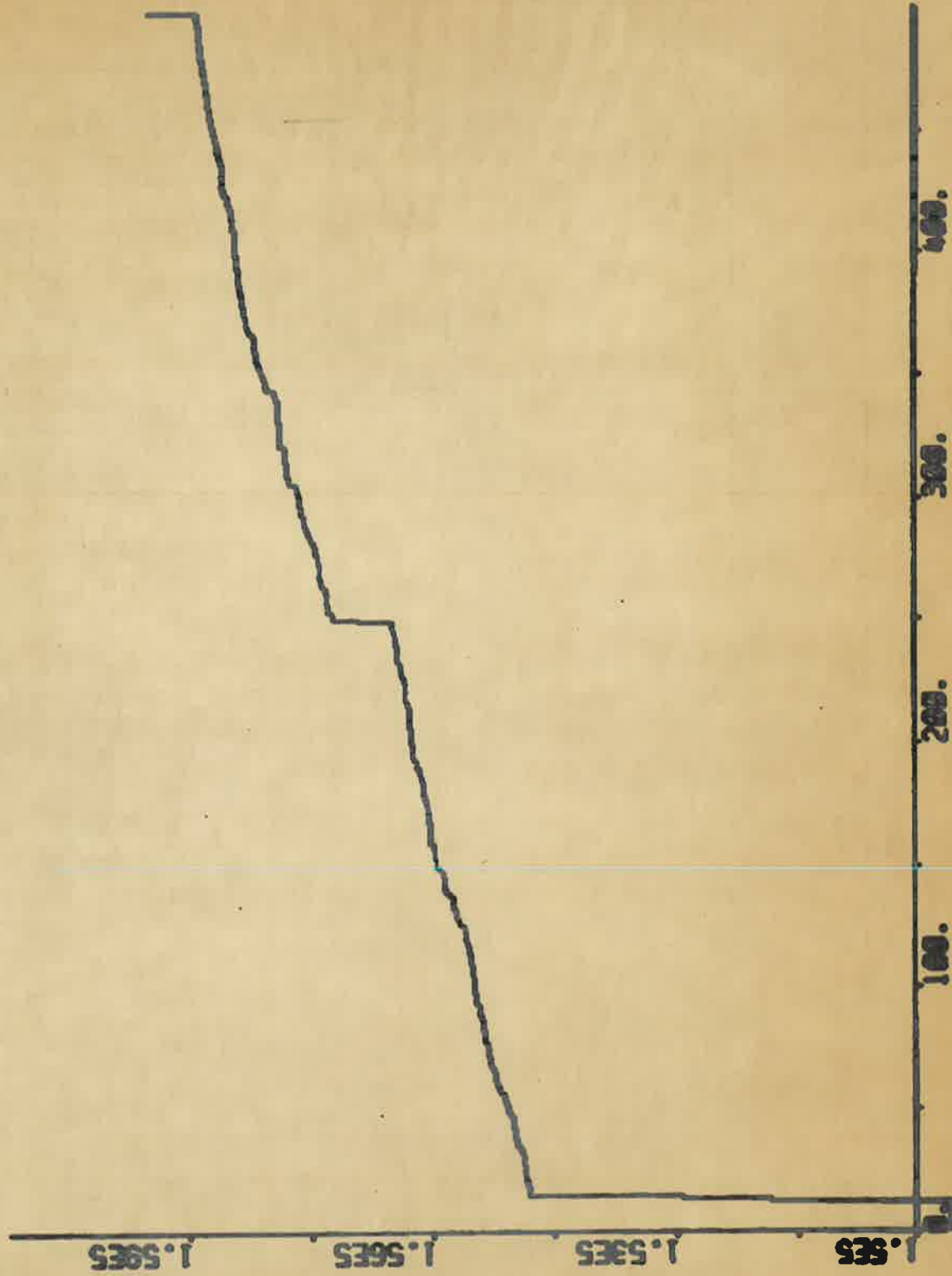
Av utsignalens utseende (fig.3.14.), syns att även här har vi för högt varians.

Variansen blir $E(y(t) - y_r(t))^2 \approx 6$ i inledningsskedet, detta att jämföra med den teoretiska $V=2.96$.

Det troliga är att även algoritm 2 konvergerar långsamt.

09.20

PLOT V



Figur 3.15. Varians för utsignalen i fig. 3.14. Denna blir omkring 6 persteg.
 $AV(250) \approx 750$ d.v.s något lägre än för algoritm 1.

EXEMPEL 3 (känslighet för felaktig β_0 -parameter.)

Processen

$$y(t) + a_1 y(t-1) = b_1 u(t-2) + e(t) + c_1 e(t-1)$$

där

$$a_1 = -0.9$$

$$b_1 = 0.5$$

$$c_1 = -0.5$$

styrdes med bägge algoritmerna för att undersöka hur regleringen påverkades då vi gav felaktigt värde på β_0 . Tidigare har korrekt $\beta_0 = 0.5$ använts, om felaktigt värde används kan stabilitetsproblem uppstå om kvoten $\frac{b_1}{\beta_0}$ ej uppfyller olikheten (se Åström-Wittenmark):

$$0 < \frac{b_1}{\beta_0} < 2$$

I tabell 1 har körningarna sammanställts

Tabell 1

Algorithm 1		Algorithm 2	
V		V	
10^{-6}	--	10^{-6}	--
0.25	1.3	0.25	1.3
1.00	1.4	1.00	1.3
5.00	1.4	5.00	1.3
10.00	--	10.00	--
estimerad	1.3	estimerad	1.3

I tabellen har med-- markerats de körningar som hade för risig utsignal för att få en jämförbar varians.

I fallen $\beta_0 = 10^{-6}$, 0.25 och då β_0 estimerades begränsades styrsignalen $(u(t))$ till 5,.

Man kunde iaktaga att då vi gav ett för högt värde på β_0 , så lyckades regulatorn efter en längre tids körning trycka ner utsignalen till rätt nivå, se figur 3.17 och 3.19. däremot ett för lågt värde (10^{-6}) gjorde att utsignalen ej dämpades ner till acceptabel varians, se figur 3.16 och 3.18.

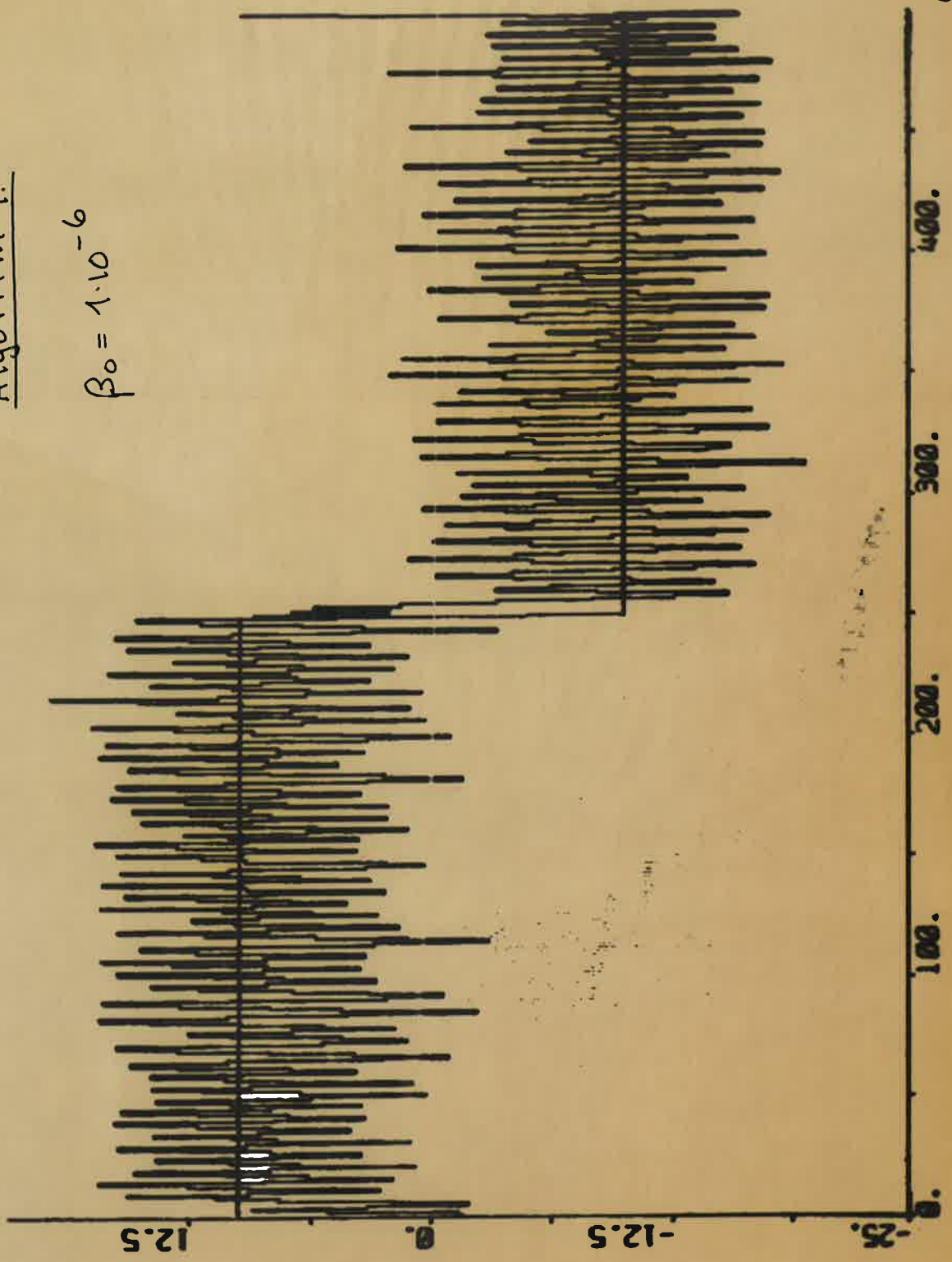
Ur tabellen framgår att inom stabilitetsgränserna så påverkades ej variansen märkbart utan denna låg omkring 1.4 för bägge algoritmerna.

Bas. 1.10⁻⁶

PLOT YLSYS1] YR

Algorithm 1.

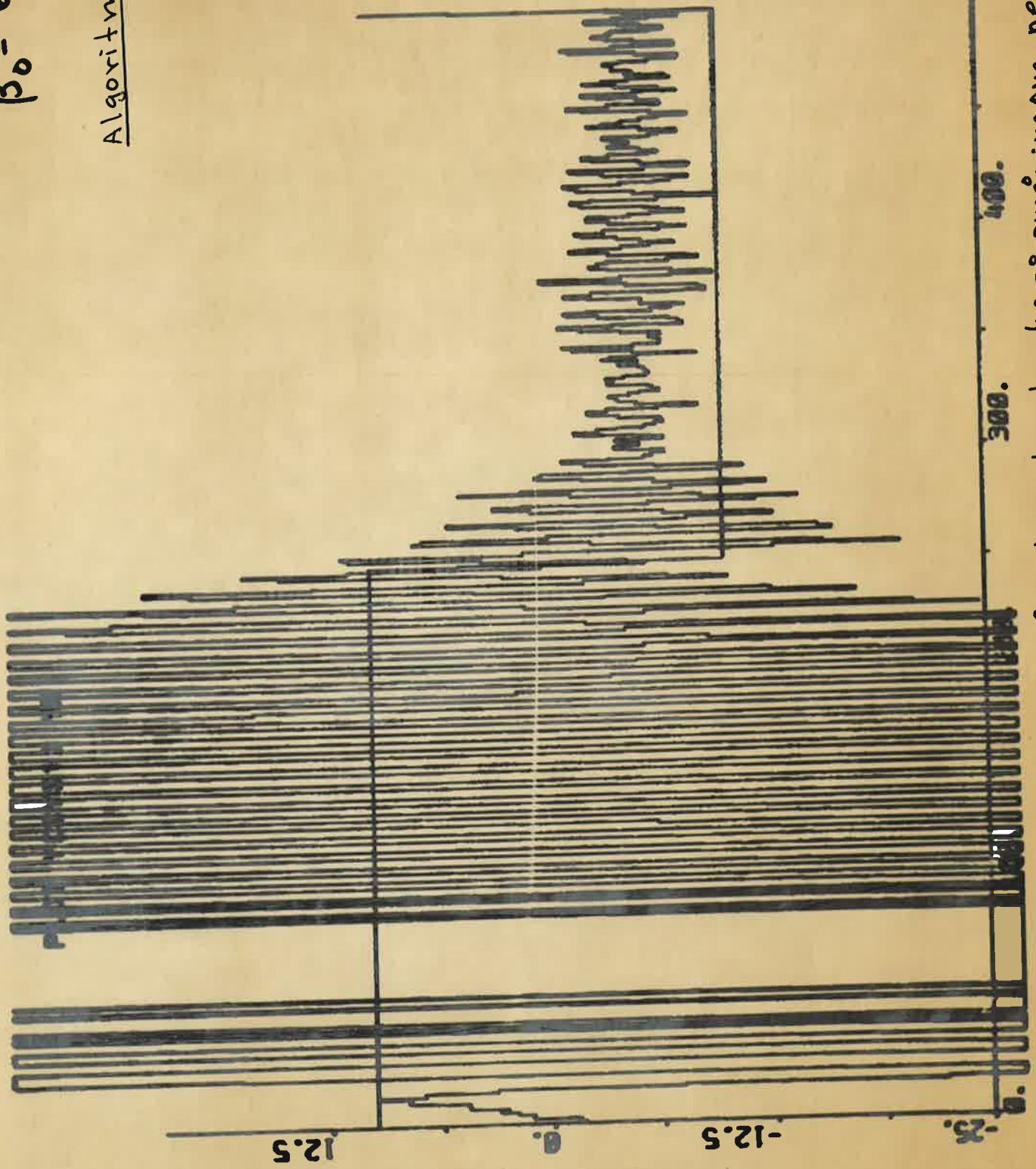
$$\beta_0 = 1 \cdot 10^{-6}$$



Figur 3.16. Utsignalens känslighet för variation av β_0 . Observera att utsignalen är dämpad ner, och att vi ligger långt under den teoretiska

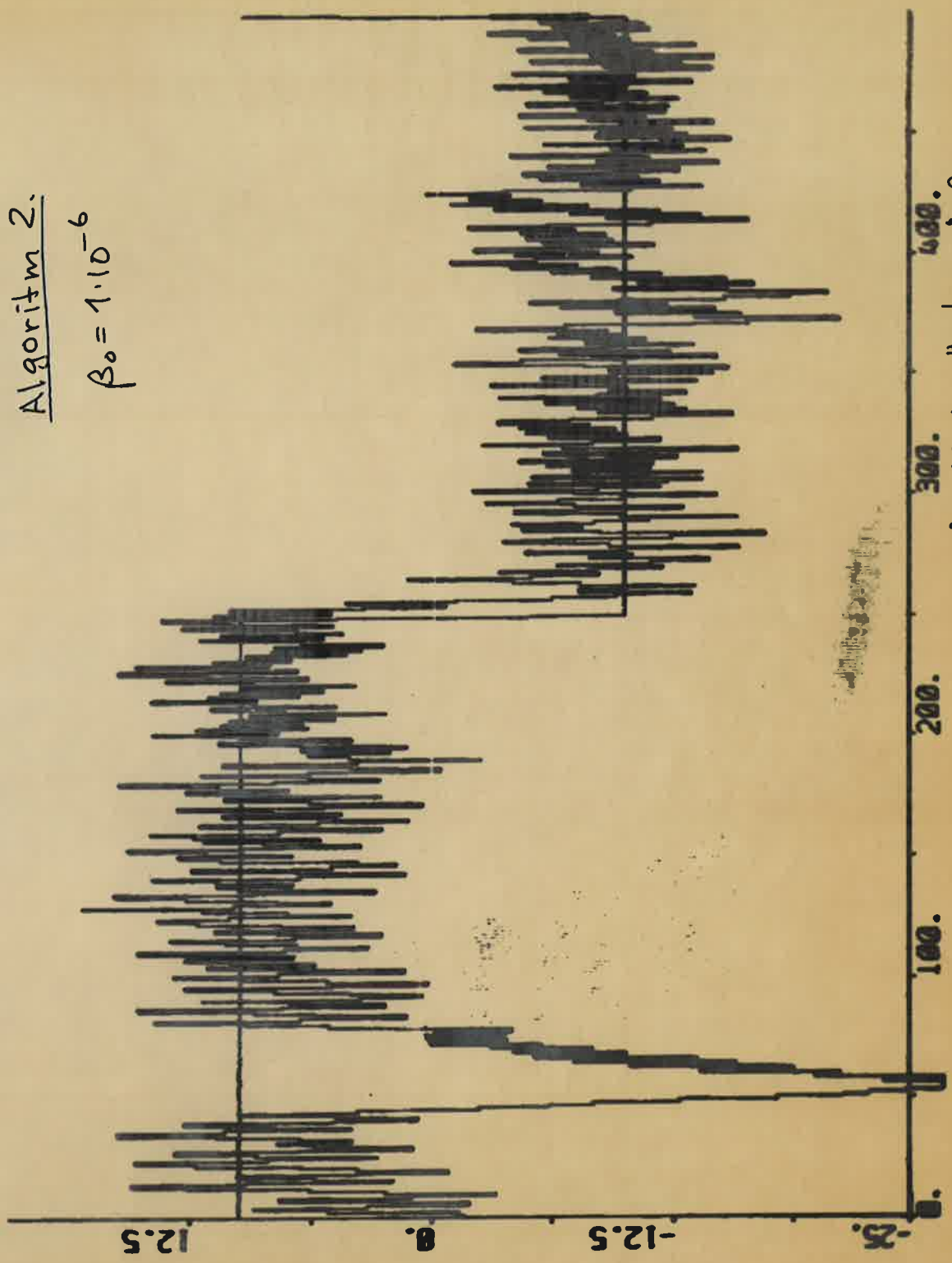
$$\beta_0 = 6.3$$

Algorithm 1.



Figur 3.17. Ett för högt värde på β_0 , utsignalen trycks så småningom ner

PLOT YSYS1] YR

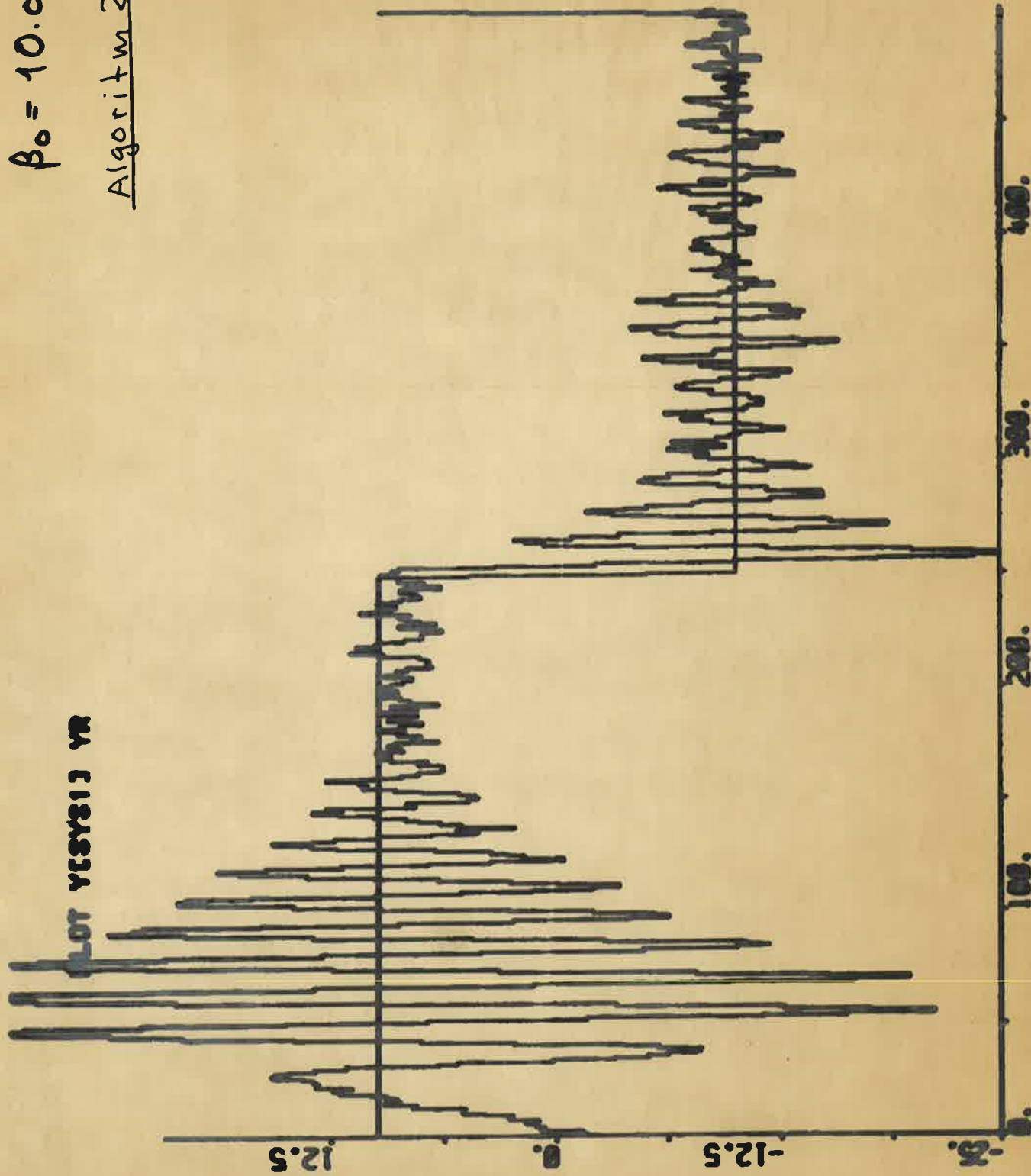


Algorithm 2.
 $\beta_0 = 1 \cdot 10^{-6}$

Figur 3.18. Utsignal då vi använder ett för lågt värde på β_0 .

$$\beta_0 = 10.0$$

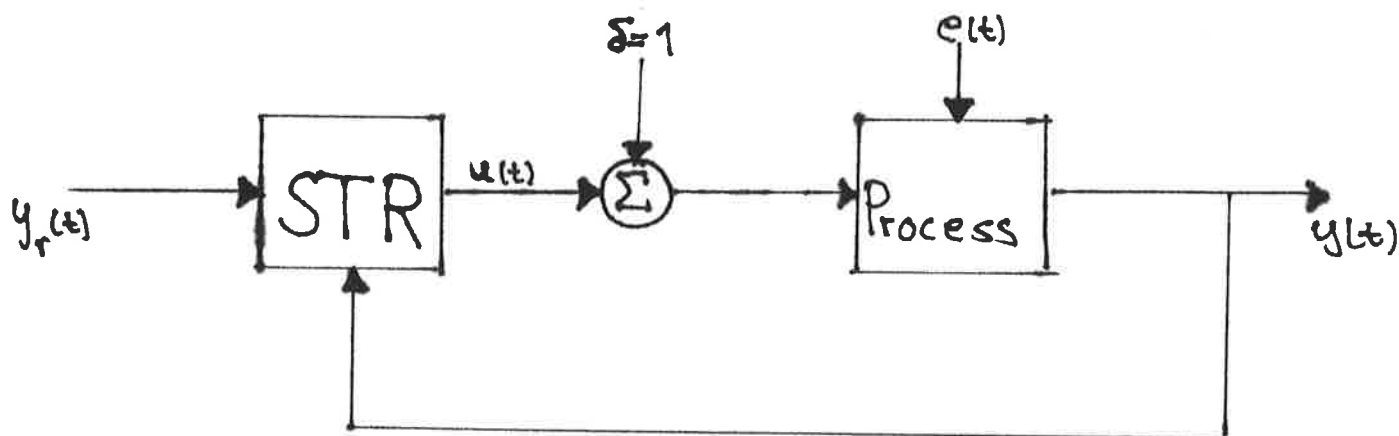
Algorithm 2.



Figur 3.19 Utsignal då vi använder ett β_0 som är 20 ggr. för stort. Denna algoritmen klarade högre värden på β_0 bättre än algoritmen 1. (jmf. 3.13.)

EXEMPEL 4 (Konstant störnings inverkan på regleringen).

I exempel 2 såg vi att styrsignalen, då vi ej har en integrator i processen, mycket väl kan inverka så att $\bar{e}_u(t) \neq 0$ och på så sätt försvåra regleringen, för att undersöka hur en konstant störning påverkar regleringen införes en konstant bias-term enligt fig.4.1.



Figur 4.1 införande av konstant bias-term.

För att få ett enkelt system att studera valdes 1:a ordningens system enligt:

$$y(t) + a_1 y(t-1) = b_1 u(t-2) + e(t) + c_1 e(t-1)$$

där

$$a_1 = -0.9$$

$$b_1 = 0.5$$

$$c_1 = -0.5$$

d.v.s. minimumfas, men utan integrator. För att ge algoritmerna större frihet och möjlighet att estimerar fram en integrator under simuleringen användes en modell med utökat ordningstal.

Detta innebär att den tidigare använda modellen för processen ovan nu ändras till.

Figur 4.1

Algoritm 1.

$$\begin{aligned}
 y(t) - y_r(t) = & -\alpha_1 [y(t-2) - y_r(t-2)] - \alpha_2 [y(t-3) - y_r(t-3)] + \\
 & + \beta_0 [u(t-2) + \beta_1 u(t-3) + \beta_2 u(t-4)] - \\
 & - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) - \gamma_2 y_r(t-2) - \gamma_3 y_r(t-3) + \varepsilon(t)
 \end{aligned}$$

Algoritm 2.

$$\begin{aligned}
 y(t) - y_r(t) = & -\alpha_1 y(t-2) - \alpha_2 y(t-3) + \beta_0 [u(t-2) + \beta_1 u(t-3) + \\
 & + \beta_2 u(t-4)] - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) - \gamma_2 y_r(t-2) + \\
 & + \delta + \varepsilon(t)
 \end{aligned}$$

Vid simuleringarna användes $P(0) = 100 \times I$, glömskeprofil $\lambda = \lambda_0 = 0.98$ d.v.s. under initialskedet tillåtes estimeringsalgoritmen att ta relativt stora steg för att sedan, då styrlagen konvergerar minska steglängden.

4.1. (utan integrator, utökad modell).

I figur 3.20. ser vi att utsignalen skiljer sig mycket lite jämfört med tidigare. Enda skillnaden uppkommer då referenssignalen ändrar nivå, då kan en viss svårighet för regleringen att trycka ner utsignalen till rätt nivå uppkomma.

Detta syns även ur variansens förlopp, där jag räknat ut denna i olika intervall (fig. 3.21.).

$$\begin{aligned}
 \frac{1}{240} \sum_{10}^{249} (y(t) - y_r(t))^2 &= 1.09 \\
 \frac{1}{250} \sum_{251}^{499} (y(t) - y_r(t))^2 &= 1.78 \\
 \frac{1}{150} \sum_{350}^{499} (y(t) - y_r(t))^2 &= 1.10
 \end{aligned}$$

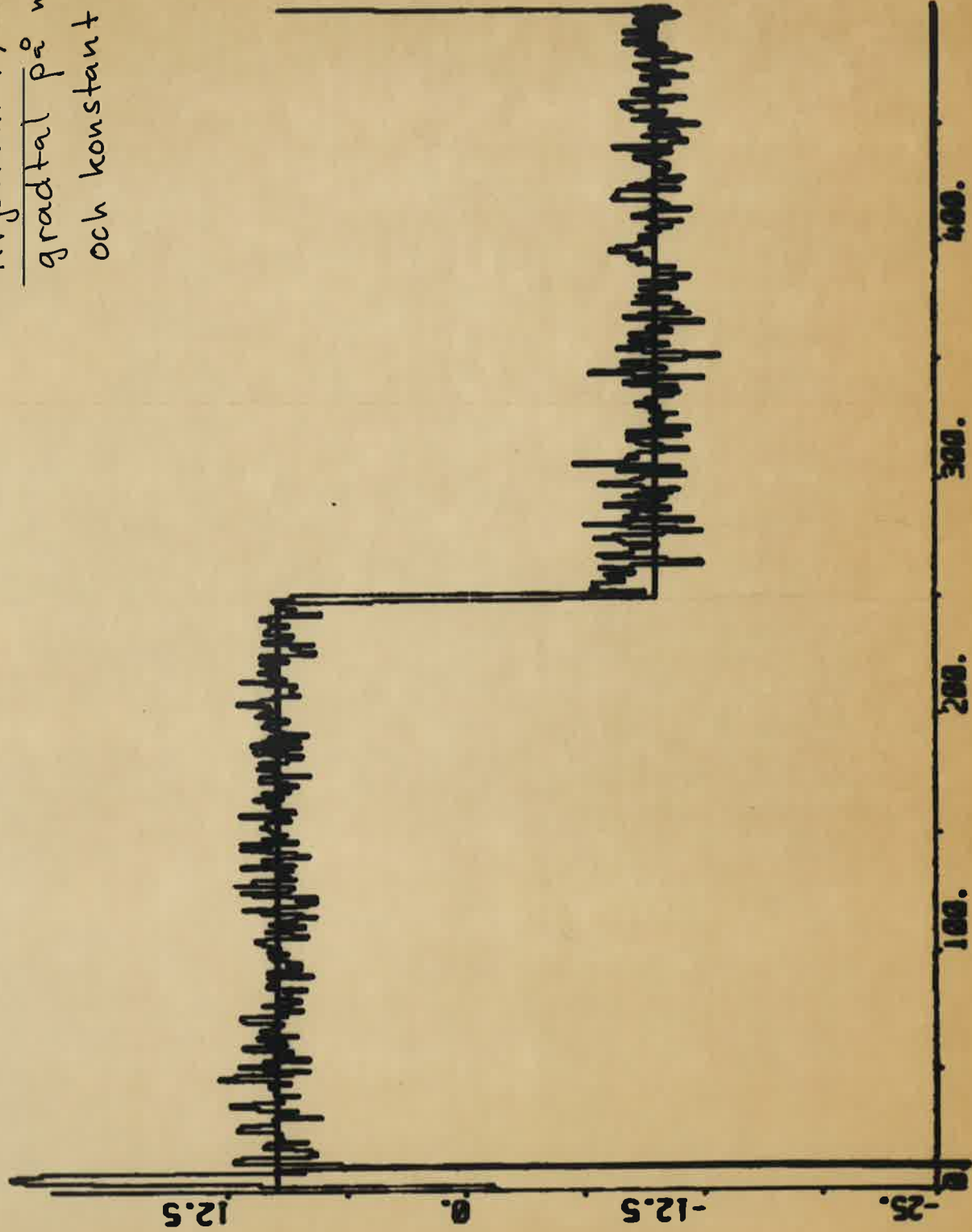
I den andra summan syns inverkan av referenssignalens ändring.
Regulatorn blir efter 2000-steg

$$u(t) = \frac{-0.816 - 0.092q^{-1}}{0.5(1 - 0.817q^{-1} - 0.097q^{-2})} (y(t) - y_r(t)) + \\ + \frac{1 - 0.127q^{-1} - 1.627q^{-2} - 0.013q^{-3}}{0.5(1 - 0.817q^{-1} - 0.097q^{-2})} y_r(t+2)$$

Om man tittar på $B(1)$ så blir detta ungefär 0.086, det kan då tyckas som om den utökade algoritmen försöker att estimera fram en integrator i B -pol. d.v.s. reglera med $\nabla u(t) = u(t) - u(t-1)$, ty då elimineras biastermens inverkan.

PLOT YR YISYS1J

Algoritm 1, med utökad
gradtal på modellen.
och konstant brasterm=1.



Figur 3.20. Nästan identisk utsignal med då vi ej hade någon brasterm som störde stursignalen. Vid ref.dnöring är det dock svårare att trycka

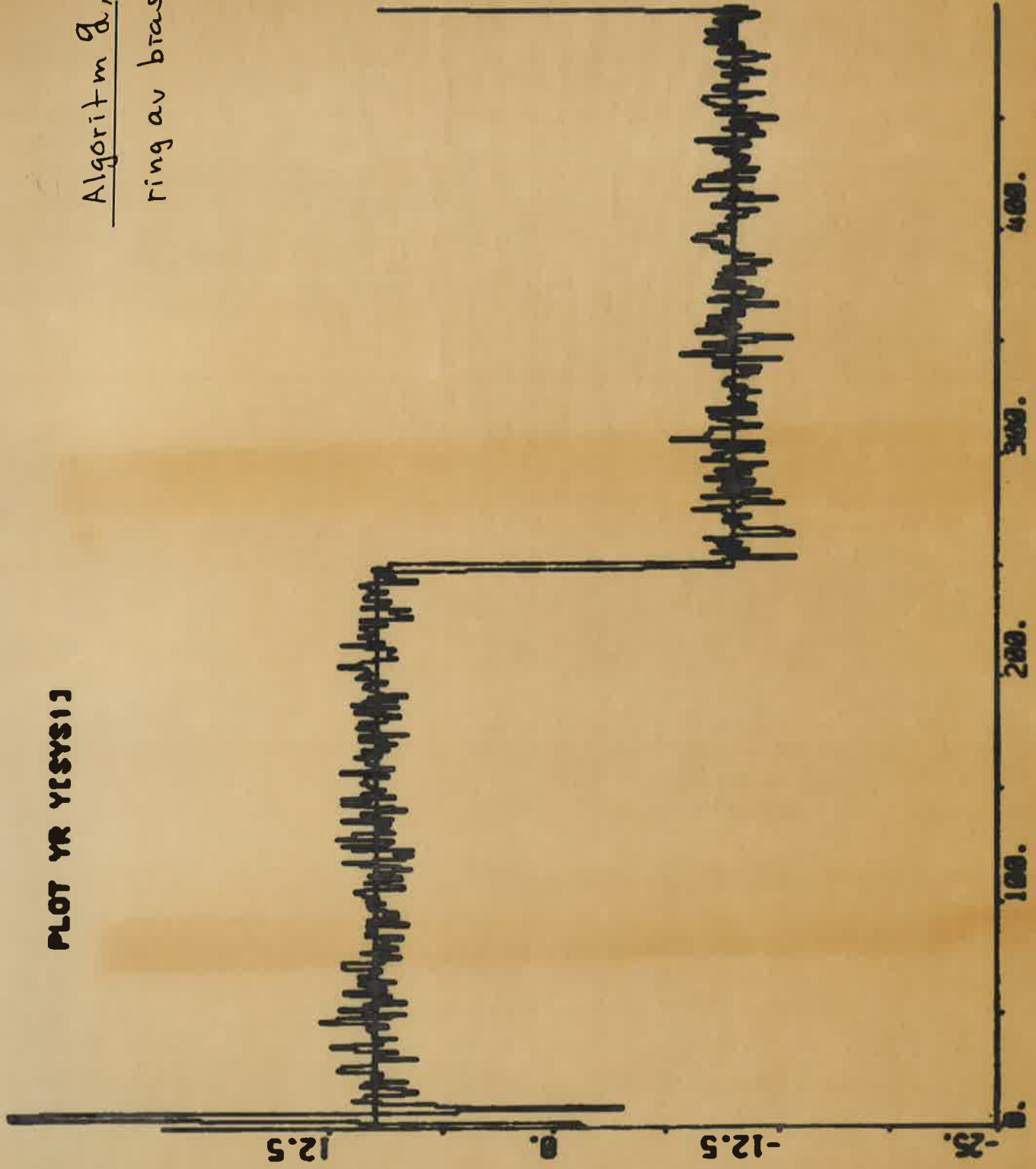
PLOT V



Figur 3.21.

Variansen för utsignalen i figur 3.20. $\Delta V(250) = 750$, precis som tidigare, man ser ur variansens uppförande vid 250, svårigheten att

PLOT YR YLSYS11



Algorithm 9, med estimate-
ring av brastermen.

Figur 3.21. Denna algorithm ger ett exakt uppförande på utsignalen som om vi ej hade några brastermen. (se f. 3.6.)

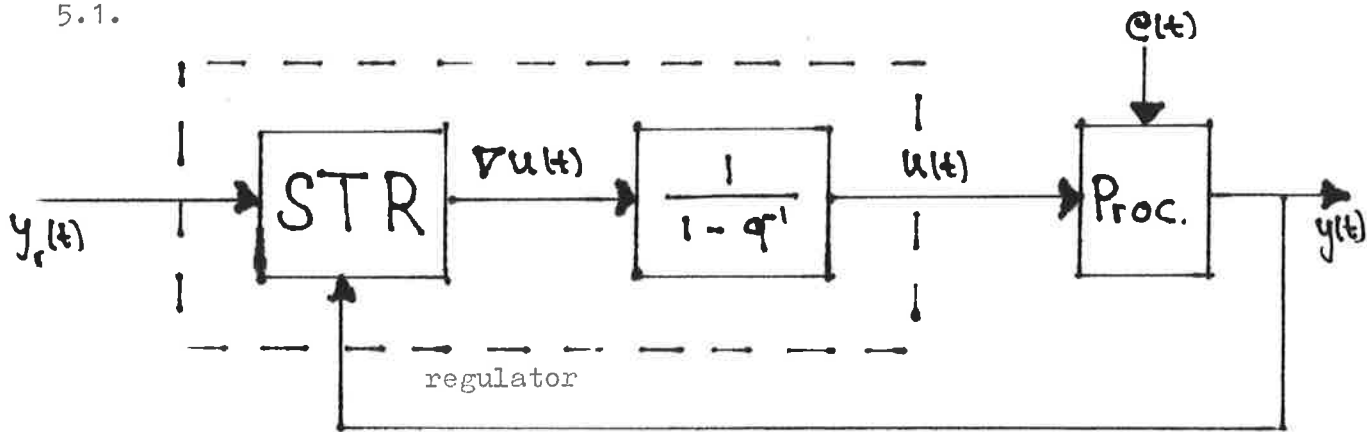
EXEMPEL 5. (kaskadkoppling med integrator).

Försök gjordes att styra processen

$$A^*(q^{-1})y(t) = B^*(q^{-1})u(t-k) + \lambda C^*(q^{-1})e(t)$$

genom att kaskadkoppla regulatorn med en integrator enligt fig.

5.1.



Figur 5.1. Kaskadkoppling med integrator.

P.g.a. att den självinställande regulatorn då känner av process+ integrator som den process som skall styras, ändras systemet till

$$A^*(q^{-1})(1-q^{-1})y(t) = B^*(q^{-1})\nabla u(t-k) + \lambda C^*(q^{-1})(1-q^{-1})e(t)$$

Eftersom den självinställande regulatorn förutsätter att C-polynomet har sina rötter innanför enhetscirkeln kan vi ej med säkerhet vänta att vi skall erhålla konvergens mot den optimala regulatorn, men om vi använder formlerna rakt fram skall denna bli:

$$\nabla u(t) = -\frac{(1-q^{-1})G}{BF} (y(t) - y_r(t)) + \frac{(1-q^{-1})AF}{BF} y_r(t+k+1)$$

Algoritm 2 blir;

$$\nabla u(t) = -\frac{(1-q^{-1})G}{BF} y(t) + \frac{(1-q^{-1})C}{BF} y_r(t+k+1) - \frac{S}{BF}$$

där F - och G -polynomen som tidigare d.v.s. vi skall få samma varians som tidigare.

Ideen med att koppla på detta vis är att vi hoppas få en snabbare konvergens och dessutom undviker vi stationära fel som annars kan uppkomma.

Betrakta systemet

$$y(t) + a_1 y(t-1) = b_1 u(t-2) + e(t) + c_1 e(t-1)$$

där

$$a_1 = -0.9$$

$$b_1 = 0.5$$

$$c_1 = -0.5$$

detta kaskadkopplas nu med integrator och då blir systemet

$$y(t) + a_1 y(t-1) + a_2 y(t-2) = b_1 u(t-2) + e(t) + c_1 e(t-1) + c_2 e(t-2)$$

där

$$a_1 = -1.9 \quad b_1 = 0.5 \quad c_1 = -1.5$$

$$a_2 = 0.9 \quad c_2 = 0.5$$

detta skall nu styras med algoritm 1 och algoritm 2. Väntad varians blir 1.16 per steg.

5.1. (algoritm 1).

Modellen för estimeringen blir:

$$\begin{aligned} y(t) - y_r(t) = & -\alpha_1 [y(t-2) - y_r(t-2)] - \alpha_2 [y(t-3) - y_r(t-3)] + \\ & + \beta_0 [u(t-2) + \beta_1 u(t-3) + \beta_2 u(t-4)] - \\ & - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) - \gamma_2 y_r(t-2) - \gamma_3 y_r(t-3) + \varepsilon(t) \end{aligned}$$

om parametrarna konvergerar mot den optimala skall dessa bli:

$$\alpha_1 = -0.36 \quad \beta_1 = 0.4 \quad \gamma_1 = -1.5$$

$$\alpha_2 = 0.36 \quad \beta_2 = 0 \quad \gamma_2 = 0.14$$

$$\gamma_3 = 0.36$$

Av utsignalens utseende att döma (fig. 3.23) så verkar det som om vi har för hög spridning på utsignalen, variansens utveckling i intervallet 10-250 steg studeras,

$$V = \frac{1}{131} \sum_{120}^{250} (y(t) - y_r(t))^2 = 5.2 ; \frac{1}{91} \sum_{160}^{250} (y(t) - y_r(t))^2 = 3.6$$

det tycks då som om vi skulle få en lugnare simulering efter hand, för att studera om utsignalen följer ett glidande medelvärde, beräknas som tidigare summan; ($y_r(t)$ konstant)

$$\sum (y(t-\tau) - y_r(t-\tau))(y(t) - y_r(t))$$

detta ger en uppskattning utav $r_e(\tau)$, ur figur 3.25 där även 5%-konfidensnivå markerats, framgår att vi verkligen har reglerfelet att följa ett glidande medelvärde av ordning 1.

För att studera parameterkonvergensen har jag valt glömskeprofilen $\lambda = 0.99$ och $\lambda_b = 0.999$ samt startat i rätt punkt, P-matrisens diagonalelement har valts till 0.1.

Ur figur 3.26 erhålles följande regulator efter 10000-steg:

$$u(t) = \frac{-0.44 + 0.4q^{-1}}{1 + 0.53q^{-1} + 0.01q^{-2}} (y(t) - y_r(t)) + \frac{1 - 1.37q^{-1} - 0.09q^{-2} + 0.46q^{-3}}{1 + 0.53q^{-1} + 0.01q^{-2}} y_r(t+2)$$

Denna regulator stämmer nästan med dem optimala, där viskulle erhållit en rot $z=1$ för \mathcal{A} - och \mathcal{B} -polynomen, vi har erhållit en rot $z=0.91$ för \mathcal{A} -polynomet.

För att se om vi erhåller rätt stationära värde för $y(t)$ studerar jag följande system:

$$Ay(t) = Bu(t) + Ce(t) \quad \text{vilket styrs med}$$

$$u(t) = -\frac{G}{BF} (y(t) - y_r(t)) + \frac{A}{B} y_r(t+k+1) \equiv \frac{\mathcal{A}}{\mathcal{B}} (y(t) - y_r(t)) + \frac{\mathcal{C}}{\mathcal{B}} y_r(t+k+1)$$

det återkopplade systemet blir

$$y(t) = \frac{B(\mathcal{B} - \mathcal{A}q^{-k-1})}{A\mathcal{B} - B\mathcal{A}q^{-k-1}} y_r(t) + \frac{\mathcal{B}C}{A\mathcal{B} - B\mathcal{A}q^{-k-1}} e(t)$$

då erhåller vi rätt stationära värde om

$$\frac{B(1)(\mathcal{B}(1) - \mathcal{A}(1))}{A(1)\mathcal{B}(1) - B(1)\mathcal{A}(1)} = 1 \Rightarrow$$

$$\underline{\underline{\frac{\mathcal{B}(1)}{\mathcal{B}(1)} = \frac{A(1)}{B(1)}}}}$$

jag erhåller vänstra ledet ovan till -0.00085, teoretiskt borde det blivit =0, men ganska god överensstämmelse ändå.

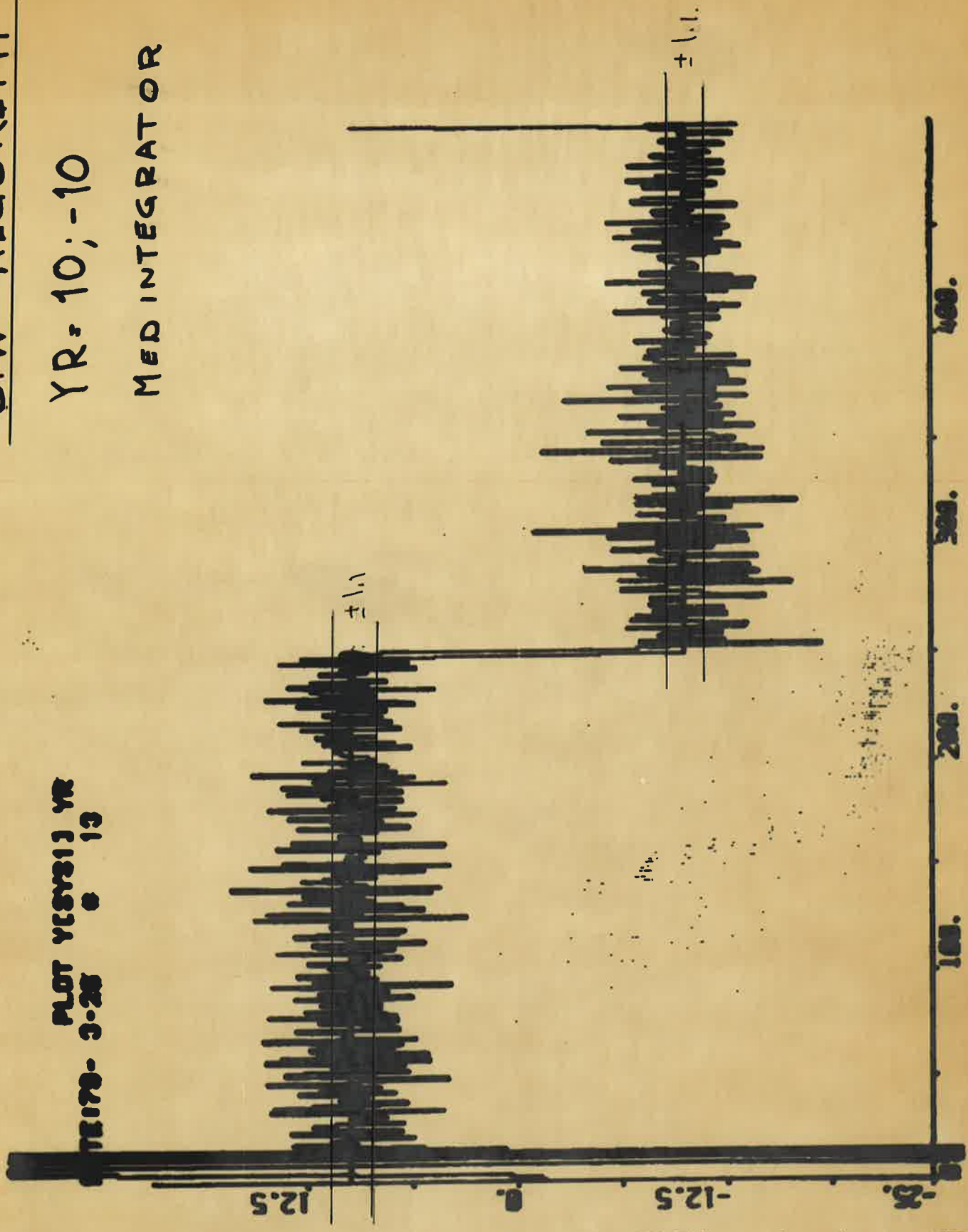
För att se hur regleringen blev med den erhållna regulatorn efter 10000-steg, lät jag systemet följa en fyrkantvåg med perioden 100, i figur 3.27 har utsignalen plottats, som synes är regleringen mycket god.

B.W. - ALGORITM

YR = 10; -10

MED INTEGRATOR

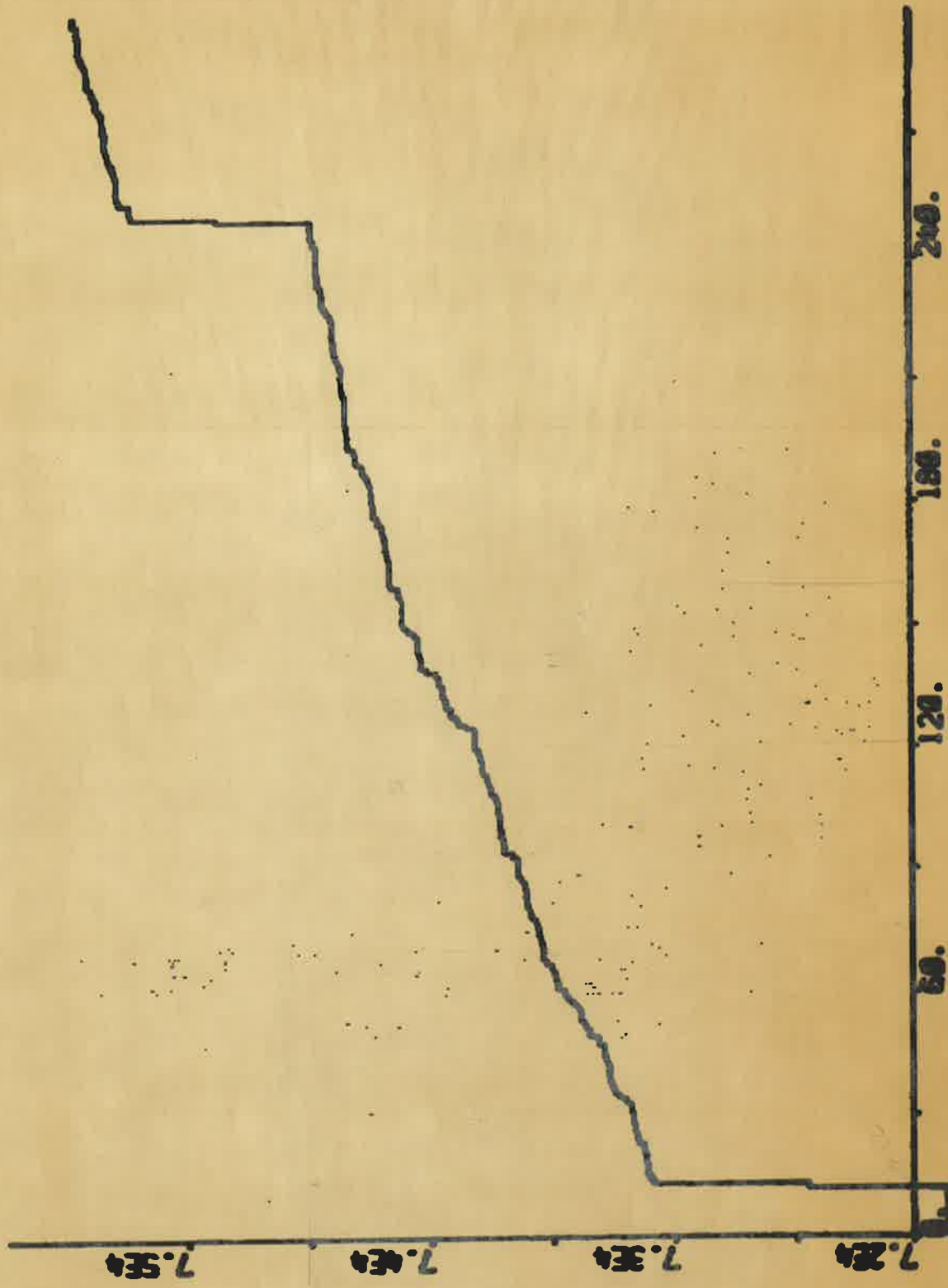
PLÖT YRSY813 YR
3-25 13



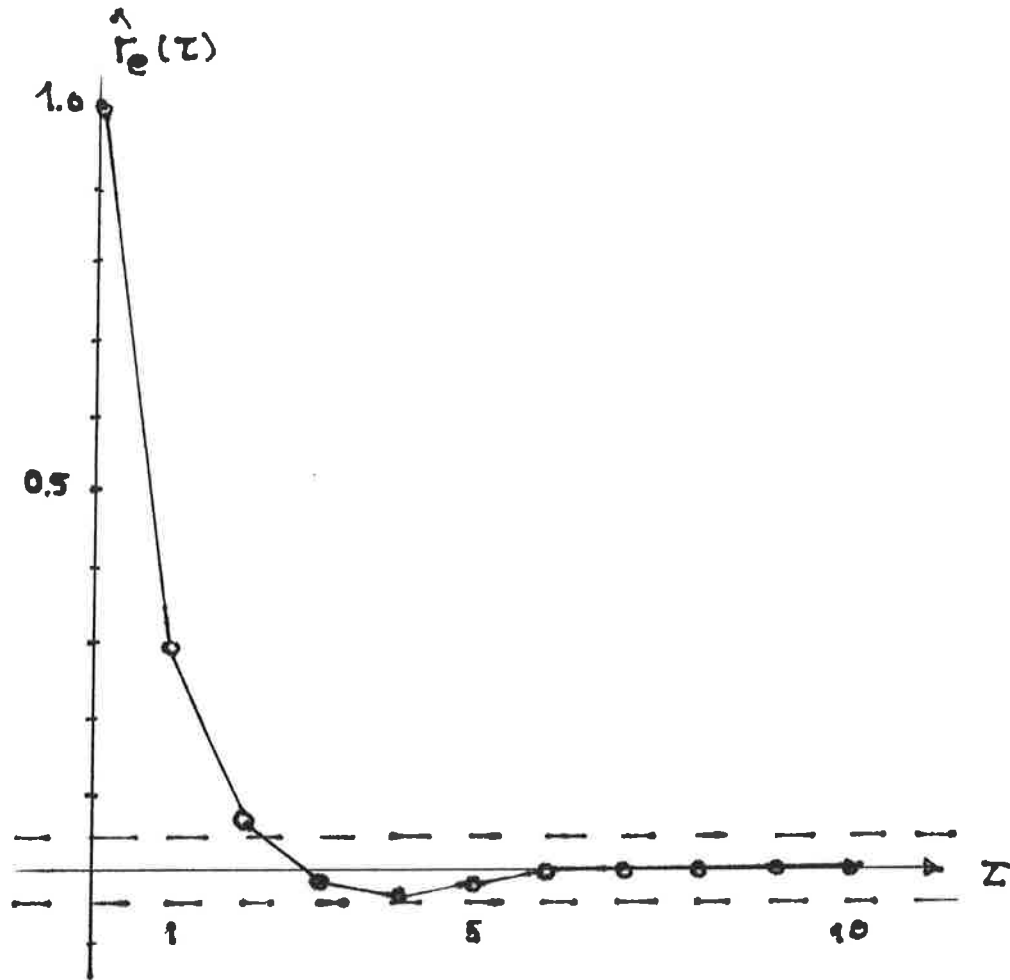
Figur 3.23

Utsignal i:a ordningens process, då vi kaskadkopplar självinställaren
och alldeles för stor spridning.

PLOT V

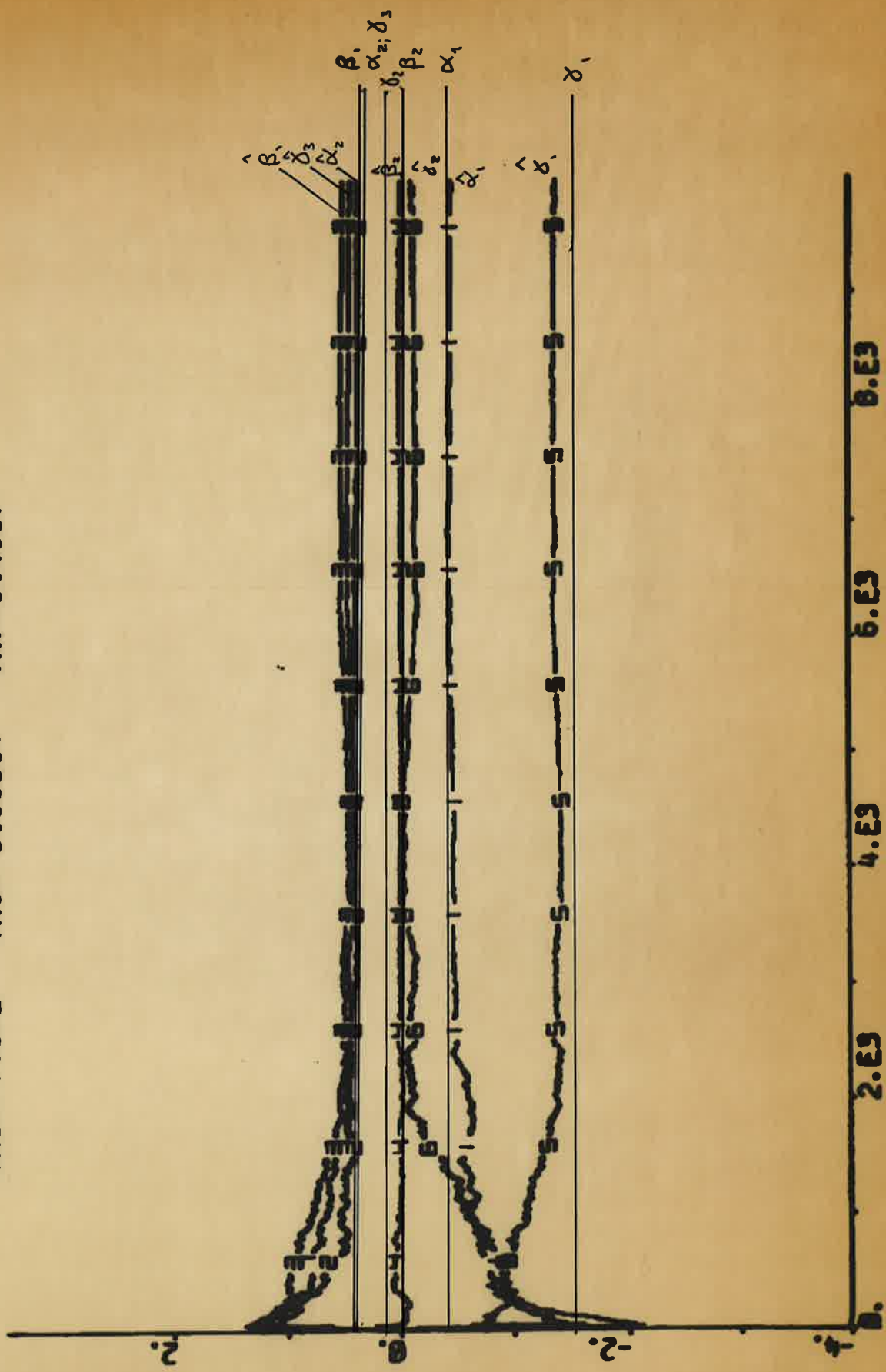


Figur 3.24. Varians för utsignalen i fig. 3.23. $\Delta V(250) \approx 750$



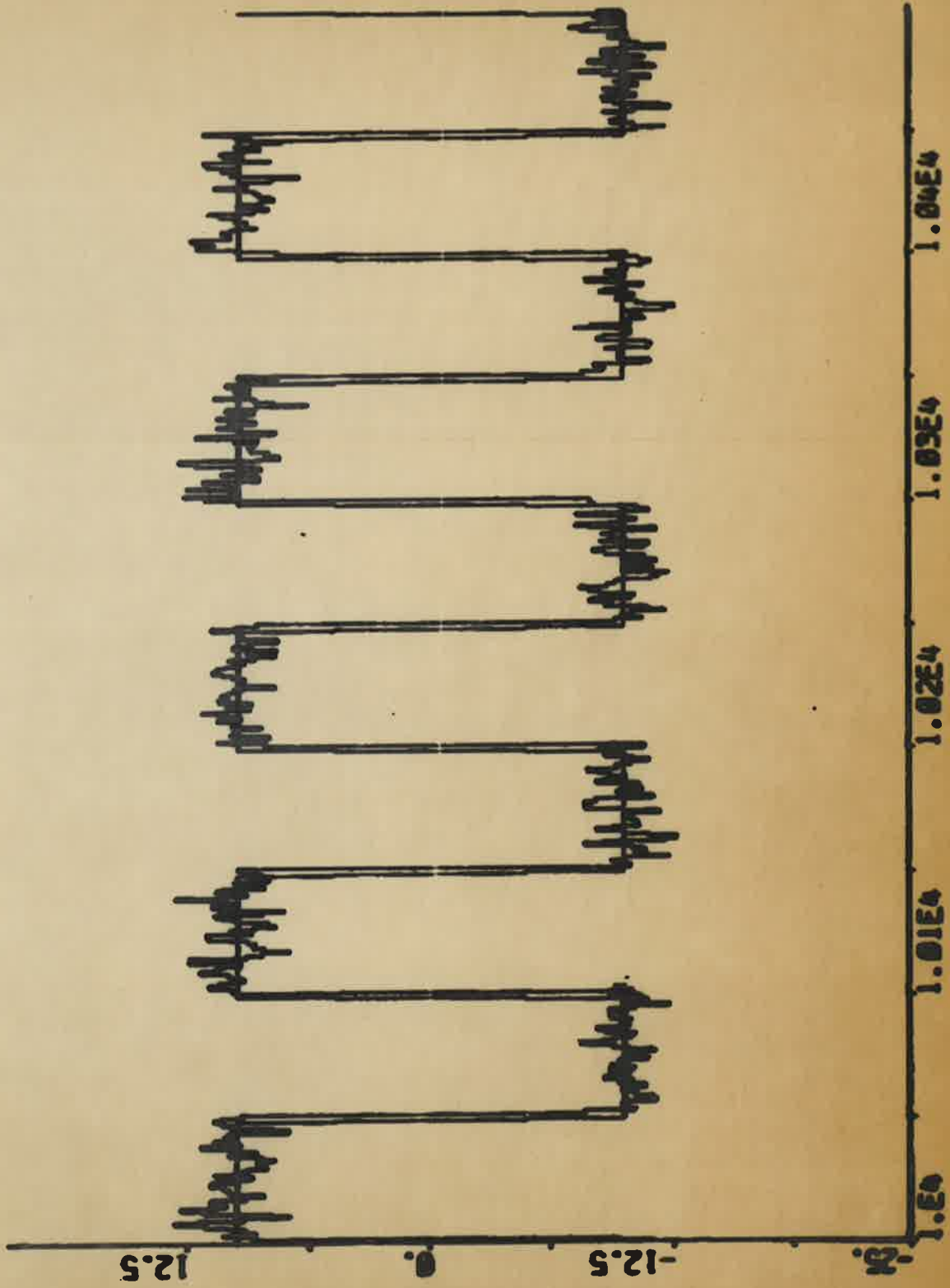
Figur 3.25. $\hat{\Gamma}_e(\tau)$ beräknad över 2000-steg.

PLOT TH1 TH2 TH3 TH4 TH5 TH6 TH7
 TH1--0.4300 TH2--0.3004 TH3--0.5206 TH4--0.01004
 TH5--1.372 TH6--0.00001 TH7--0.4007



Figur 3,26. Estimeringens förlopp; glömskeprofilen $\lambda = 0.99$ och $\lambda = 0.999$ har valts. 51
 ... parametererna kan förhållas. Men vi erhåller rätt station-

PLOT VLSYS13 YR



Figur 3.27. Utsignalen från den ur fig. 3.26. erhållna regulatorn, vi ser nu att variansen är liten och regleringen fungerar bra.

5.2. (algoritm 2).

Modellen för algoritmen blir

$$y(t) - y_r(t) = -\alpha_1 y(t-2) - \alpha_2 y(t-3) + \beta_0 [u(t-2) + \beta_1 u(t-3) + \beta_2 u(t-4)] - \gamma_0 y_r(t) - \gamma_1 y_r(t-1) - \gamma_2 y_r(t-2) + \mathcal{D} + \varepsilon(t)$$

där parametrarna om de konvergerar mot den optimala regulatorn skall bli:

$$\begin{aligned} \alpha_1 &= -0.36 & \beta_1 &= 0.4 & \gamma_1 &= -1.5 \\ \alpha_2 &= 0.36 & \beta_2 &= 0 & \gamma_2 &= 0.5 \end{aligned}$$

Om vi ser på utsignalens beteende (figur 3.28), så liknar den mycket utsignalens beteende i algoritm 1, d.v.s. stor spridning i initialskedet, variansen blir (figur 3.29) omkring 7 per steg. För att kontrollera regleringen beräknades även här summan

$$\sum (y(t-2) - y_r(t-2))(y(t) - y_r(t))$$

resultatet har plottats i figur 3.30, ur denna framgår att vi ej har kravet för glidande medelvärde uppfyllt, för reglerfelet $y(t) - y_r(t)$, det är närmast $r(3)$ som avviker.

För att se hur parametrarna konvergerar använde jag en fyrkantvåg med perioden 100 och glömskeprofilen $\lambda = 0.99$ samt $\lambda_0 = 0.999$, uppstart gjordes i rätt punkt och $P(0) = 0.1 \times I$.

Regulatorn blev efter 10000-steg: (figur 3.31.)

$$u(t) = \frac{-0.47 + 0.44q^{-1}}{1 + 0.5q^{-1} + 0.01q^{-2}} y(t) + \frac{1 - 1.40q^{-1} + 0.42q^{-2}}{1 + 0.5q^{-1} + 0.01q^{-2}} y_r(t+2) + \frac{0.001}{1 + 0.5q^{-1} + 0.01q^{-2}}$$

För att se om denna regulator ger rätt stationära värde studerar jag liksom i ex.5.1 systemet:

$$A y(t) = B u(t-k) + C e(t)$$

vilket styrs med regulatorn

$$\begin{aligned} u(t) &= -\frac{G}{BF} y(t) + \frac{C}{BF} y_r(t+k+1) - \frac{J}{BF} \\ &\equiv \frac{A}{B} y(t) + \frac{C}{B} y_r(t+k+1) - \frac{J}{B} \end{aligned}$$

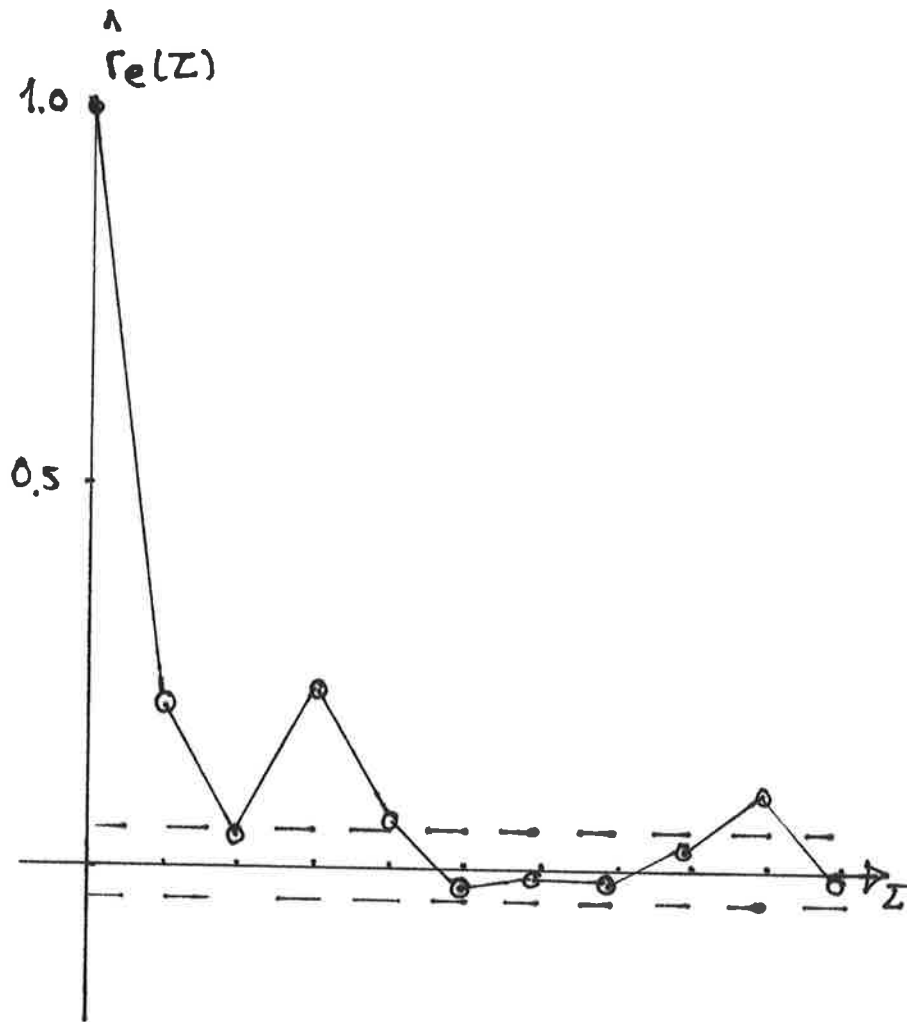
det återkopplade systemet blir då

$$y(t) = \frac{BC}{BA - B A q^{-k-1}} y_r(t) + \frac{BC}{BA - B A q^{-k-1}} e(t) - \frac{J}{BA - B A q^{-k-1}}$$

då erhåller vi rätt stationära värde om

$$\begin{aligned} \frac{B(1)C(1)}{B(1)A(1) - B(1)A(1)} &= 1 \Rightarrow \\ B(1) &= \frac{B(1) - A(1)}{B(1)} - A(1) = C(1) \end{aligned}$$

Jag erhåller vänstra ledet ovan lika med 0.027 och det högra blir 0.026, d.v.s. mycket god överensstämmelse.



Figur 3.30. $r_e(z)$ beräknad över 2000-steg. De streckade linjerna är 5%-konfidens-nivå

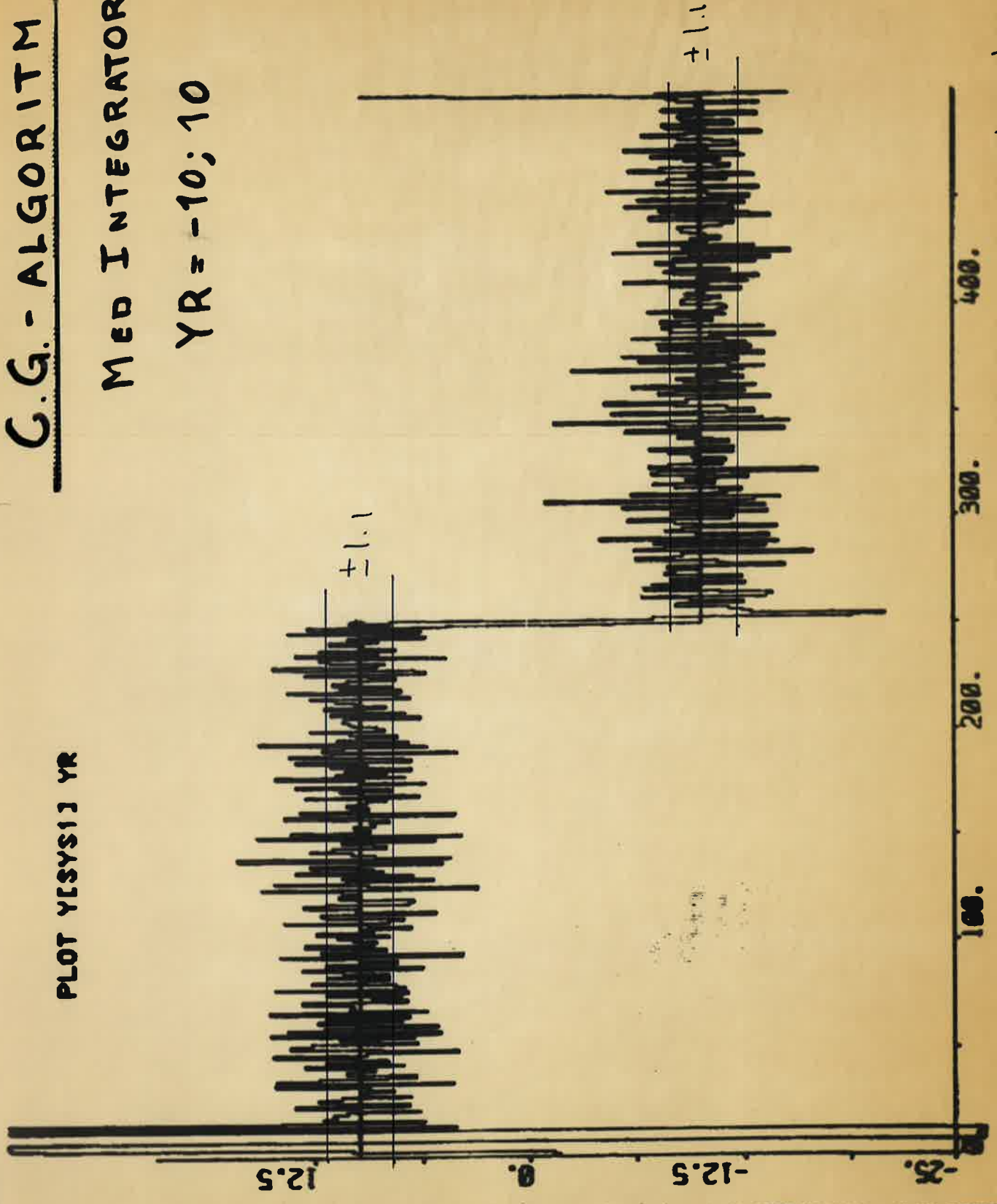
MA 2.3.2
14.0.1974

G.G. - ALGORITM

MED INTEGRATOR

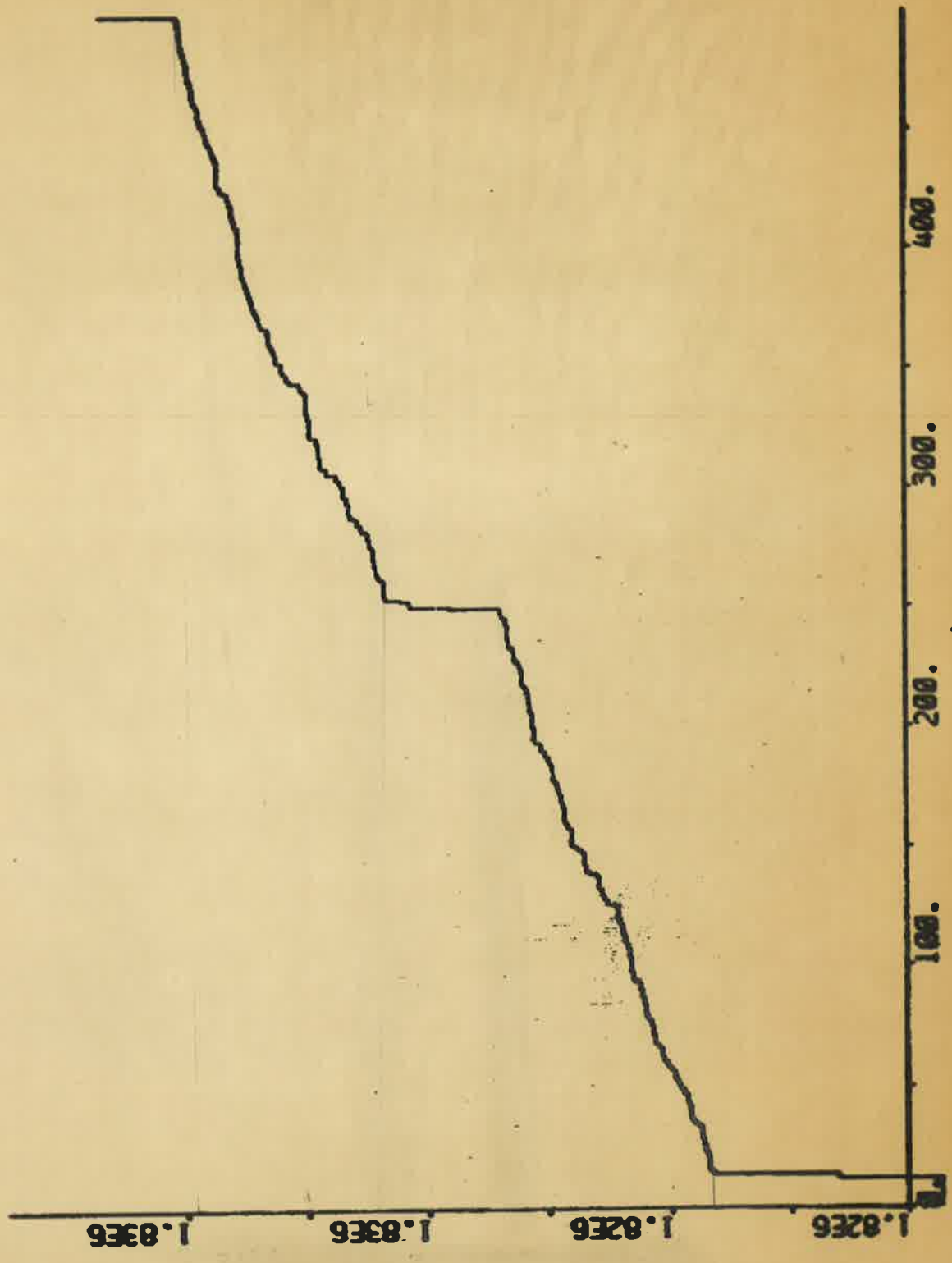
YR = -10; 10

PLOT Y[SYS] YR



Figur 3.28 Utsignal då vi kaskeadloopplar spälurinställaren med en integrator.

PLOT V
V=1.029E6

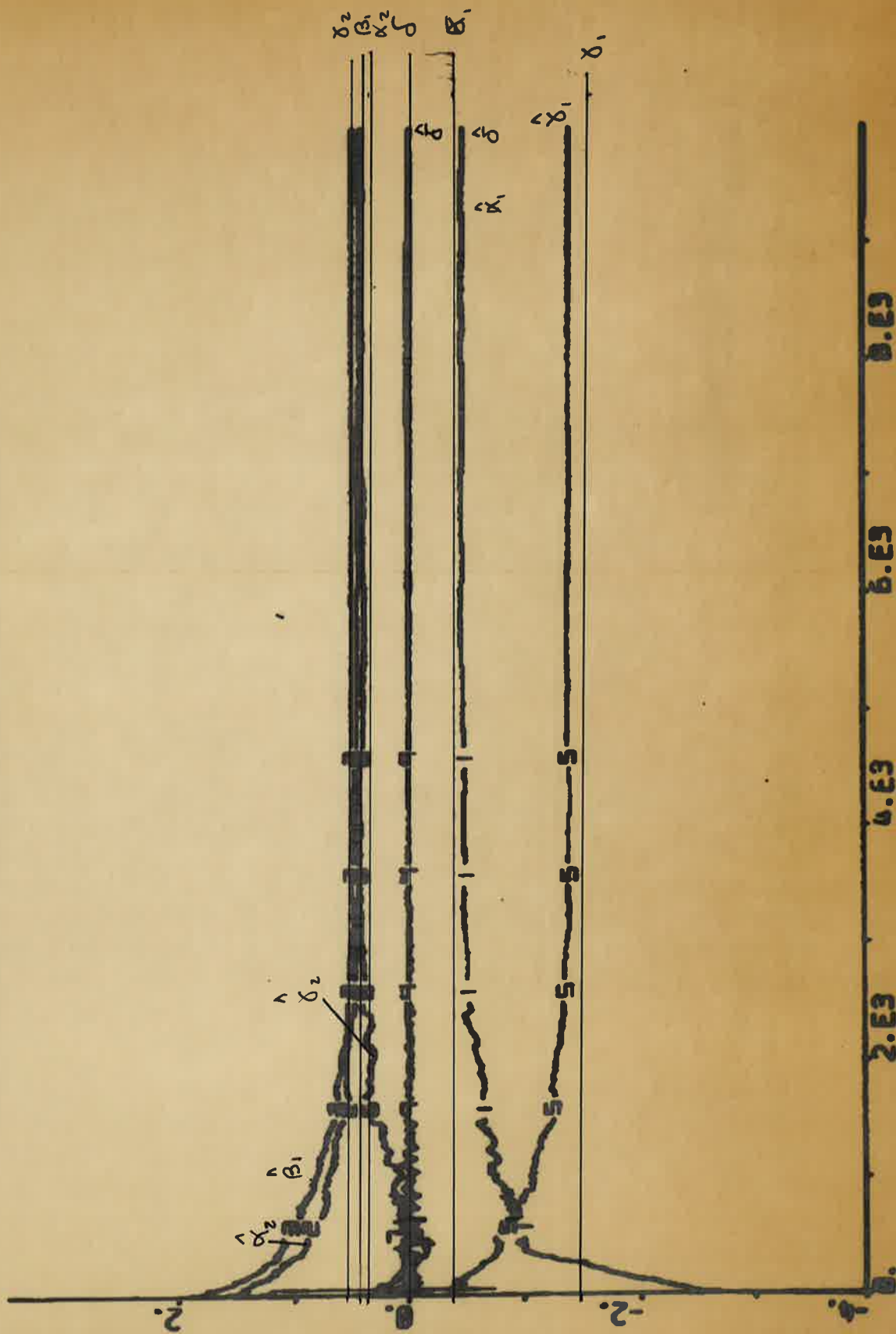


AD = 1.029
1.029E6

1.83E6
1.8280 1.8285
1.8290 1.8295
1.83E6

Figur 3.29. Varians för utsignalen i fig. 3.28.

PLOT TH1 TH2 TH3 TH4 TH5 TH6 TH7
 TH1 = -0.4861 TH2 = 0.4391 TH3 = 0.5025 TH4 = 0.9002 - 3
 TH5 = -1.300 TH6 = 0.4239 TH7 = -1.3002 - 3



Figur 3.31.

Estimeringens tidsförlopp. Glömskeprofil $\lambda_0 = 0.999$ och $\lambda = 0.99$ även här har en liten avvikelse uppkommit, men y erhåller sitt rätta storträkta värde

4. SAMMANFATTNING.

Då jag simulerade styrning av processerna utan integrator i kaskadkoppling, så blev konvergensen långsammare då gradtalet för processen ökade, detta bekräftades av att utsignalen i 3:e ordningens system blev ett glidande medelvärde och parameterpolynomen konvergerade långsamt till rätt värde (se fig. 3.9.), som sågs i exempel 2 med 3:e ordn. system fick vi problem med ett medelvärdesfel på styrsignalen vilket gjorde att $r_{eu}(z)$ ej låg i närheten av nollnivån, vilket den borde göra för att algoritmerna skall ha konvergerat.

För att undvika detta problem utökade vi algoritmernas gradtal, för att ge dessa möjlighet att estimeras fram en integrator i processen och såg då hur regleringen blev, då en konstant biasterm

$\delta = 1$ infördes på styrsignalen. Algoritm 1 uppvisade då en liten svårighet att trycka ner utsignalen till rätt nivå då referenssignalen ändrade nivå, men mellan referensvärdesändringarna var styrning^{en} mycket god med korrekt varians på utsignalen, för algoritm 2 däremot hade vi ett helt perfekt beteende, men detta var att vänta ty här estimeras biastermen fram.

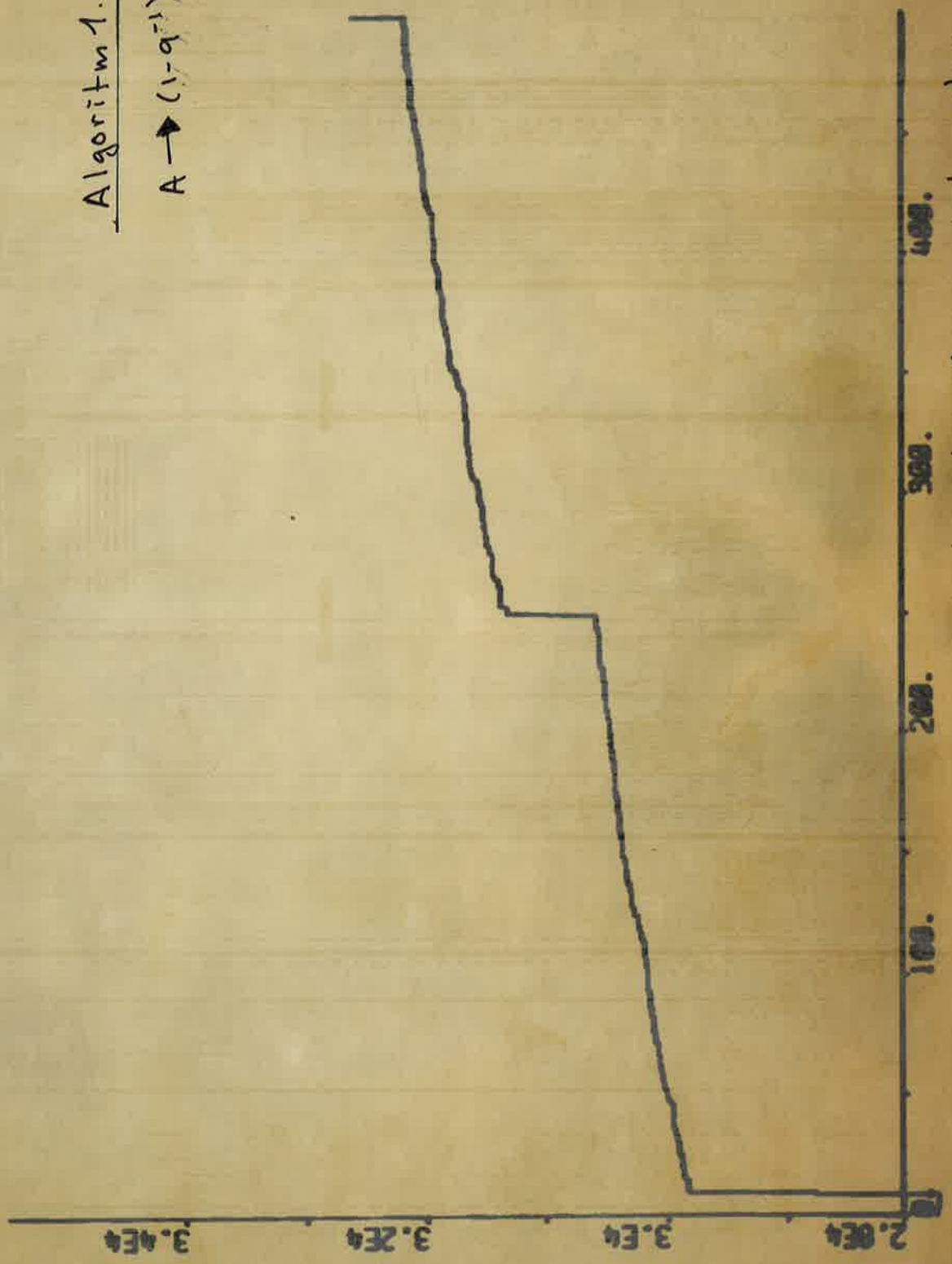
Ett annat sätt att försöka eliminera svårigheterna med konstanta störnivåer är att kaskadkoppla med en integrator, detta innebär att A - och B -polynomet kommer att innehålla en rot $z=1$ på enhetscirkeln och detta innebär att vi differentierar $y(t)$ och $y_f(t)$. I exempel 5 framgick då att vi fick en för hög varians i inledningskedet på utsignalen, men efterhand konvergerade regulatorn mycket nära den optimala och utsignalens beteende efter 10000 steg var mycket bra (se fig. 3.27). Trots att bägge algoritmerna bara konvergerade till nästan den optimala så erhöll vi rätt stationära värde på utsignalen.

TABELL 4.1.

1:a ordningens syst	teor. var.	exp. var.
Utan integrator	B.W. 1.16	1.14
	C.G. 1.16	1.15
Med integrator i kaskadkoppling	B.W. 1.16	≈ 5.8
	C.G. 1.16	≈ 7
Med utökad est.- modell.	B.W. 1.16	1.1
	C.G. 1.16	1.15
Utan integrator 3:e ordningens syst.	B.W. 2.96	≈ 6
	C.G. 2.96	≈ 6

PLOT V

Algorithm 1.
 $A \rightarrow (1-q^{-1})A$



Figur 3.32. Varrans då A-polynommet direkt tillförs en integrator.
Varrans: bli omkvarva 2.9 per steg. $\Delta v(250) = 750.$

5. Referenser.

- Clarke, D W, and P J Gawthrop (1975): Self-tuning Controller, Proc IEEE I22, 928-934.
- Gustavsson, I (1978): User's Guide for a Program package for Simulation of self-tuning Regulators. Department of Automatic Control, CODEN: LUTFD2/(TFRT-7149)/1-76/(1978),
- Wittenmark, B (1979): Personal communication.
- Wittenmark, B (1973): A Self-Tuning Regulator. TFRT-1003, Department of Automatic Control, Lund Institute of Technology, Lund, Sweden.
- Åström, K J and B Wittenmark (1973): On Self Tuning Regulators. Automatica 9, 185-189.
- Åström, K J, U Borisson, L LJUNG; and B Wittenmark (1977): Theory and Applications of Self-Tuning Regulators. Automatica 13, 457-476.
- Åström, K J and I Gustavsson (1978): Analysis of a Self Tuning Regulator in a servoloop. Department of Automatic Control Lund Institute of Technology, Lund, Sweden. CODEN: LUTFD2/(TFRT-3150)/1-058/(1978)