

ADAPTIV REGULERING AF IKKESTATIONÆRE PROCESSER

DEL 2

Niels Kjølstad Poulsen

**LYNGBY 1982
EKSAMENSPROJEKT
NR 5/82**

imsot

Trykt af **INSD**, DTH

Kapitel 7: IKKE STATIONÆRE FORSTYRRELSER -----	337
7.1 Standardalgoritmerne -----	341
7.2 Glømselsfaktor -----	367
7.3 Kalmanfiltrering -----	387
7.4 Afledt model -----	423
7.5 Integralvirkning -----	427
7.6 Fejldetektor -----	436
7.7 Robust identifikation -----	477
7.8 Robust adaptiv regulering -----	505
 Kapitel 8: KONKLUSION -----	 541
 APPENDIX -----	 545
1. Prediktor -----	546
3. Valg af model -----	549
4. Valg af observerpolynomie -----	555
5. Overføringsfunktioner -----	562
6. Specifikke overføringsfunktioner for objekt II -----	567
7. Simplificeret stabilitet -----	571
8. Brugervejledning -----	573
9. Programmel -----	582
 LITTERATUR LISTE -----	 620

KAPITEL 7

7. Ikke Stationære Forstyrrelser.

- 7.1 Standardalgoritmerne
- 7.2 Glemselsfaktor.
- 7.3 Kalmanfiltrering.
- 7.4 Afledt Model.
- 7.5 Integralvirkning
- 7.6 Fejldetektor.
- 7.7 Robust Identifikation
- 7.8 Robust Adaptiv Regulering.

7. Ikke Stationære Forstyrrelser.

I forbindelse med regulering af praktisk forekomne objekter, vil der være tilfælde hvor omgivelsernes påvirkninger, det vil sige forstyrrelserne, ikke kan modelleres som en stationær proces.

$$(1) \quad v(t) = C(q^{-1})e_o(t) + d$$

For konventionelle regulatorer er deres øjne til at reducere eller helt fjerne indflydelsen fra en belastning eller en variation i denne meget væsentlig. Det vil derfor være meget interessant at undersøge hvorledes en adaptiv regulator reagerer overfor en belastnings variation. I kapitel 5 blev de 3 selvindstillende regulator typer dimensioneret således, at de kunne udkompensere en konstant belastning. Belastnings variationerne vil her være opdelt i to kategorier. Den ene består af abrupte ændringer, som i den generelle formulering i kapitel 2 blev betegnet som spring. Disse spring kan skyldes bratte ændringer af en fysisk belastning, men kan også skyldes direkte fejl i styreorganer og måleudstyr. Den anden kategori består af mere lavfrekvente ændringer, som betegnes drift. Ligesom den første kategori kan drift have mange oprindelser. Temperaturforhold kan typisk afstedkomme sådanne langsomme ændringer af belastningen.

Som en følge af de anvendte estimationsalgoritmers uheldige egenskaber overfor meget store støjimpulser, også kaldet spidser, vil det være ret relevant at undersøge og om muligt forbedre disse forhold. I dette kapitel vil vi gå ud fra den modellering af forstyrrelserne, der er givet i kapitel 2. Her er forstyrrelserne, $v(t)$, til objektet

$$(2) \quad A(q^{-1})y(t) = q^{-k}B(q^{-1})u(t) + v(t)$$

givet ved:

$$(3) \quad v(t) = C(q^{-1})e(t) + d(t)$$

Selve støjen, $e(t)$, kan spaltes op i to dele sådan at:

$$(4) \quad e(t) = (1 - \delta_1(t)) e_0(t) + \delta_1(t) e_1(t)$$

hvor $e_0(t)$ er en sekvens af uafhængige $N(0, \sigma_0^2)$ fordelte stokastiske variable. Den stokastiske variabel, $\delta_1(t)$, er en markovkæde med overgangssansynligheden s_1 og den stokastiske matrice:

$$(5) \quad \pi_1 = \begin{bmatrix} 1-s_1 & s_1 \\ 1-s_1 & s_1 \end{bmatrix}$$

Fordelingen af $e_1(t)$ antages ukendt. Det vil dog være forudsat, at disse spidser indtræffer ret sjældent ($s_1 \ll 1$), hvis variansen af $e_1(t)$ ikke eksisterer. Belastningen kan j.v.f. kapitel 2 modelleres som:

$$(6) \quad d(t) = d(t-1) + \alpha_2(t)e_2(t) + \alpha_3(t)e_3(t)$$

Det antages, at springene ligesom spidserne er en sjælden begivenhed. Det vil sige, at overgangssansynligheden, s_2 , i den beskrivende stokastiske matrix

$$(7) \quad \pi_2 = \begin{bmatrix} 1-s_2 & s_2 \\ 1-s_2 & s_2 \end{bmatrix}$$

er ret lille. Der forudsættes heller ikke noget om fordelingen af $e_2(t)$, hvilket er i modsætning til $e_3(t)$, der skal have en

begrænset varians. Såfremt variansen af $e_2(t)$ eksisterer vil det i denne sammenhæng være naturligt, at $\sigma_3 \ll \sigma_2$, når overgangssansynlighederne v_1 og v_2 i

$$\pi_3 = \begin{bmatrix} 1-v_1 & v_1 \\ v_2 & 1-v_2 \end{bmatrix}$$

er små.

I det førtse afnit vil algoritmerne, sådan som de er præsenteret i kapitel 5, blive undersøgt overfor de nævnte typer på virkninger. For at adskille disse versioner, der også er anvendt i kapitel 6, fra de øvrige vil de blive betegnet som standard algoritmer, selvom en sådan betegnelse også kunne bruges om de øvrige. Dernæst vil der blive undersøgt to kendte metoder til at tilpasse algoritmerne til tidsvarierende objekter. Den ene af disse metoder går ud på at vægte målingerne sådan, at de sidst foretagne indgår med størst vægt. Den anden metode tager udgangspunkt i en opfattelse af estimationsalgoritmen som et udvidet kalmanfilter.

De to efterfølgende metoder bygger på et helt andet princip. Her er formålet at danne en model af objektet, sådan at belastningen slet ikke indgår. Disse metoder kræver uvægerligt, at regulatoren kan udkompensere belastningen uden at kende dennes størrelse. Den klassiske løsning er at anvende en integrator. Den efterfølgende metode går ud fra de foregående metoders dilemma, der skyldes to modstridende krav. Hensigten med den syvende metode er at forbedre de adaptive regulatorers egenskaber overfor spidser. Til sidst afprøves en endelig algoritme sammen med nogle af regulator typerne.

7.1 Standardalgoritmer.

I dette afsnit vil der blive demonstreret, hvorledes de adaptive regulatore, der er beskrevet i kapitel 5, reagerer overfor påvirkninger, som de ikke er dimensioneret til. Lad objektet være givet ved:

$$(1) \quad A(q^{-1})y(t) = q^{-k} B(q^{-1})u(t) + v(t)$$

hvor

$$(2) \quad v(t) = e(t) + d(t)$$

For på simpel og kontrolleret vis at simulere de i det foregående omtalte påvirkninger vil $d(t)$ være enten en trin- eller en rampefunktion. I simulationerne indsættes en enkelt spids i $e(t)$ direkte. Disse påvirkninger, der er ofte anvendt i den klassiske reguleringsteknik, adskiller sig bl. a. ved deres forskellige frekvensspektrum og dermed deres forskellige eksitationsgrad. Som referencesignal vil der i hele dette kapitel blive brugt et trin, sådan at $w(t) = 1$ for $t \geq 0$.

Til simulationerne vil der i det følgende blive anvendt objekt I, der er givet ved:

$$(3) \quad \begin{aligned} A(q^{-1}) &= 1 - 1.5q^{-1} + 0.7q^{-2} \\ B(q^{-1}) &= 1 + 0.5q^{-1} & k &= 1 \\ C(q^{-1}) &= 1 & \sigma_o^2 &= 0.1 \end{aligned}$$

Til reguleringen af dette objekt vil der i demonstrationerne blive anvendt en MVo - regulator, fordi denne er et especial - tilfælde samt et grundliggende eksempel for de implicite

regulatoralgoritmer. For denne regulator, der her anvendes i den generelle polplacerings formulering, sker estimationen med modellen:

$$(4) \quad y = (\hat{f}_0 + \hat{f}_1 q^{-1})y(t-k) + b_0 u(t-k) + \hat{g}_0 u(t-k-1) + \hat{\gamma} + \varepsilon(t)$$

mens j.v.f. afsnit 6.1.1 bestemmes af:

$$(5) \quad u(t) = \frac{1}{b_0} (w(t) - (\hat{f}_0 + \hat{f}_1 q^{-1})y(t) - g_0 u(t-k-1) - \hat{\gamma})$$

således at:

$$(6) \quad e_f(t) = e_m(t) = y(t) - w(t-k) = e(t)$$

for korrekte parameterestimer, der er givet ved:

$$F(q^{-1}) = 1.5 - 0.7q^{-1}$$

$$(7) \quad G(q^{-1}) = 0.5$$

$$\gamma = d = 0$$

Da tidsforsinkelsen gennem objektet er 1 vil belastningen blive estimeret direkte, fordi $E(q^{-1}) = 1$ for $k = 1$. For korrekte regulatorparameter vil variansen af styresignalet være 0.5053, når de øvrige påvirkninger, referencesignal og belastning, er konstante.

Deadbeat regulatorene adskiller sig på markante punkter fra de øvrige, hvorfor der i det følgende vil blive givet kuvert fra simulationer med disse. Deadbeat - algoritmerne er eksplisit-

te, der bliver derfor estimeret parameterne i modellen:

$$(8) \quad (1 + \hat{a}_1 q^{-1} + \hat{a}_2 q^{-2}) y(t) = q^{-1} (\hat{b}_0 + \hat{b}_1 q^{-1}) u(t) + \hat{d} + \varepsilon(t)$$

når regulatoren bruges overfor objekt I. I DBo - regulatoren bestemmes styresignalet som

$$(9) \quad u(t) = \hat{q}_0 (\hat{b}_0 + \hat{b}_1 q^{-1}) u(t-1) + \hat{q}_0 (1 + \hat{a}_1 q^{-1} + \hat{a}_2 q^{-2}) * (w(t) - y(t))$$

der som vist i kapitel 6 giver en middelmådig reguleringskvalitet. ($\text{Var}(y) = 0.31$ og $\text{Var}(u) = 0.0444$ for korrekte parameter). Regulatoren indeholder selv ved fejlagtige parameterestimerer en integrator, hvorfor der kan forventes specielle egenskaber i forbindelse hermed.

Spidser.

I fig. 1 og 2 er der angivet tidsforløbet af AC - forstærkningen og DC - fejlen for en simulation med et system bestående af objekt I og MVo - regulatoren. Her bevirker spidsen:

$$(10) \quad e(t) = \begin{cases} 30 & \text{for } t = 39 \\ e_0(t) & \text{ellers} \end{cases}$$

at RLS - estimereren, der minimerer størrelsen

$$(11) \quad I_e(t) = \sum_{i=0}^t \varepsilon^2(i)$$

flytter estimerne væk fra deres ellers rimelige værdier med det resultat, at systemet bliver ustabil. Den efterfølgende kraftige eksitation, der her både skyldes selve spidsen og at det samlede system er blevet ustabil i et mindre tidsrum, medfører at de dynamiske parameter (alle undtagen estimatet af belastningen) bliver meget godt bestemt, hvilket fremgår af fig. 1 og 4. Derimod medfører denne eksitation ikke, at estimatet af γ (= d) vender tilbage til udgangspunktet. Fig. 2 og 3 viser, at det fejlagtige estimat er meget længe om at blive korrigeret og giver derfor anledning til en markant stationær fejl og dermed en kraftig forøgelse af tabene. Den mest tabsgivende effekt er dog reguleringen med fejlagtige parameter selvom dette kun foregår i et kort tidsrum.

Såfremt spidsen ikke er nær så stor (f. eks. kun 5) vil parameterestimerne stadigvæk springe, men eksitationen af objektet er ikke nær så kraftig. Bliver det resulterende system ikke ustabil, vil estimerne af de dynamiske parameter ikke konvergerer tilnærmelig så godt, som tilfældet er i fig. 1. Tidspunktet hvori spidsen fremkommer er også af afgørende betydning for størrelsen af estimaternes spring og deres muligheder for hurtigt at vende tilbage.

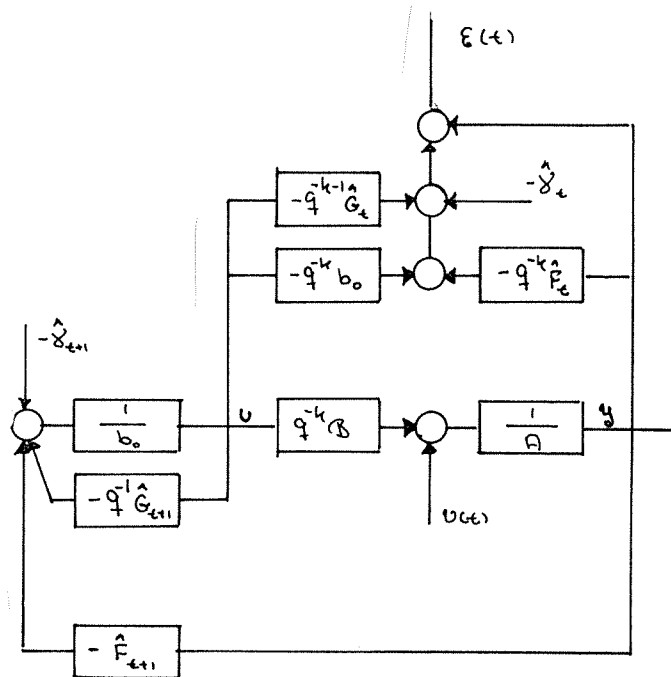
For deadbeatregulatorerne er forholdene ligeså uheldige. De relativt dårlige reguleringssegenskaber gives her til udtryk ved et ekstremt højt tab. Ved store spidser og når de dynamiske parameters estimer konvergerer ødelægger det fejlagtige estimat af d ikke reguleringen.

Spring.

Såfremt tidsforløbet af $d(t)$ ændres fra at være konstant til:

$$d(t) = \begin{cases} 0 & \text{for } t < 40 \\ 1 & \text{for } t \geq 40 \end{cases}$$

vil objektet ikke blive udsat for den samme eksitation. Som det fremgår af fig. 5 - 9 er MVo - regulatorens egenskaber overfor denne påvirkning ikke særligt gode. Tidsforløbet af parameter - estimaterne viser, at RLS - estimatoren imod en umiddelbar forventning ændrer estimatet af $F(1)$, sådan at springet udkompenseres.



Objekt med MVo - regulator og invers model.
Identisk med fig. 5.3.3.1,2.

Først senere bliver \hat{y} opdateret i overensstemmelse med springet i d . Som det fremgår af fig. 9 følger estimatet af d meget langsomt springet og starter i forkert retning. Estimatet af g_0 er næsten upåvirket af springet. De nævnte variationer i estimaterne af F - parameterne bevirker en forøgelse af

variansen for samtlige signaler, hvilket kommer til udtryk i fig. 5, 7 og 8. De store tab for $y(t)$ og $\varepsilon(t)$ opstår imidlertid som følge af den stationære fejl (fig. 6) som det fejlagtige estimat af $d(t)$ medfører. Ligesom tilfældet med de stationære påvirkninger trækker de to bidrag til DC - fejlen i modsat retning. Denne DC - fejl synes ikke at påvirke tabene for styresignalet.

Sænkes nu springets størrelse til 0.3 svarende til spredningen af det stationære støjsignal $e_o(t)$, vil den omtalte variation af $\hat{F}(q^{-1})$ ikke være særligt udpræget. AC - forstærkingen (fig. 12) er derfor ret upåvirket af springet i $d(t)$. Ydermere har estimatet af belastningen (fig. 14) ikke det ret uheldige udsving umiddelbart efter springet. Estimatet følger dog springet ret langsomt, hvilket dog kun giver anledning til en forøgelse af hældningen for tabskurven (fig. 13) for $y(t)$ på 2 % i forhold til tilfældet $d = 0$.

De her fundne egenskaber overfor mindre spring i $d(t)$ vil være forringet, hvis springet indtræffer på et senere tidspunkt. Når kovariansmatricen, $P(t)$, for parameterestimerne er reduceret, vil det ikke være muligt for RLS - algoritmen, at følge springet, men til gengæld ligger de øvrige parameter fast.

Med en deadbeatregulator er det på grund af intralvirkningen muligt at udregulere et spring i belastningen, $d(t)$, selvom estimatet af denne er fejlagtig, fordi dette estimat ikke indgår i beregningen af styresignalet. I fig. 15 og 16 er kurverne for tabene og tidsmiddelværdien for styrefejlen, $e_{st} = w - y$, optaget fra en simulation, hvor belastningen kun springer 0.3 til $t = 40$. Imidlertid påvirkes alle parameterestimerne, således at overføringsfunktionerne bliver meget fejlagtige (fig. 17).

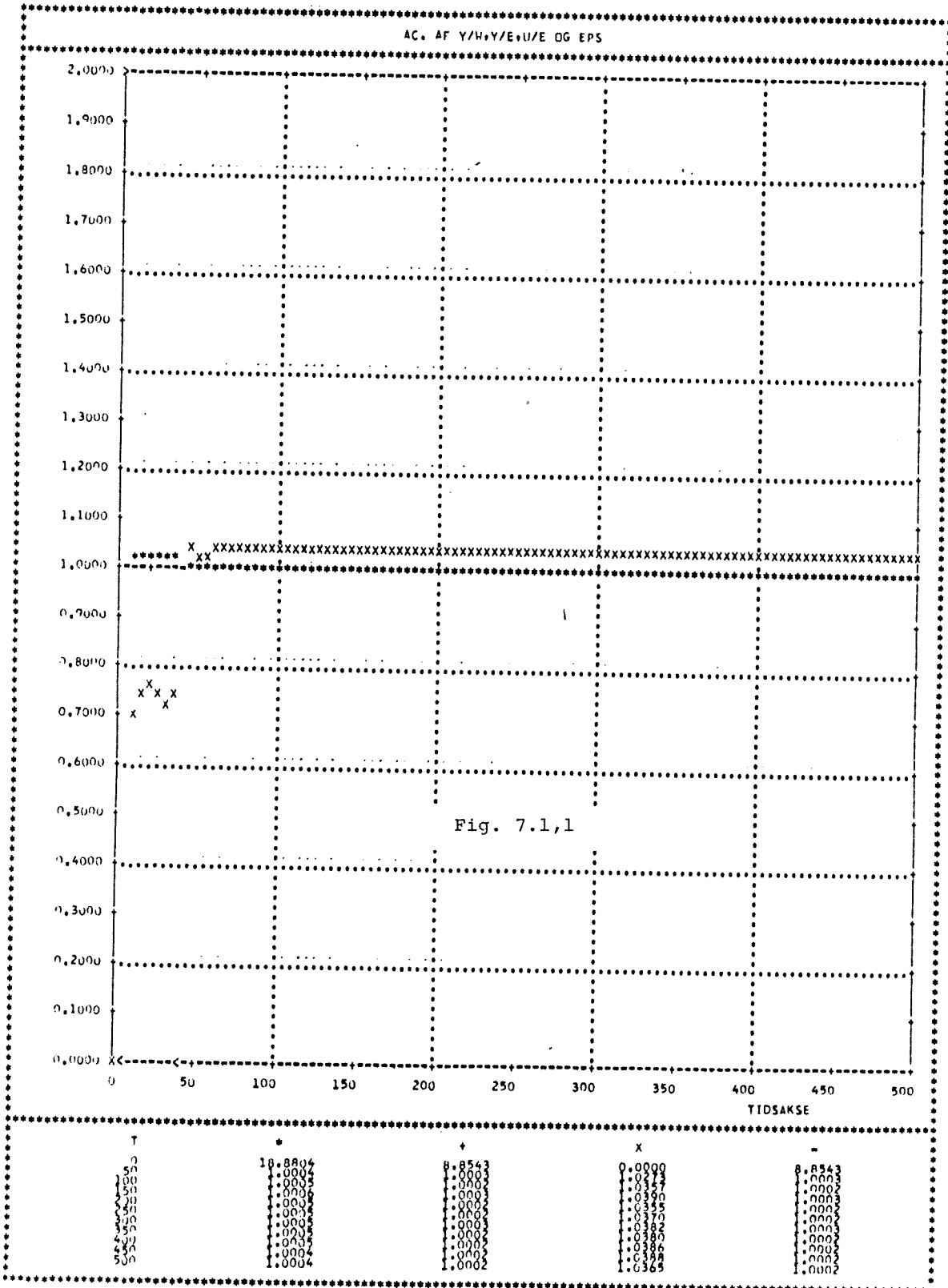
Drift.

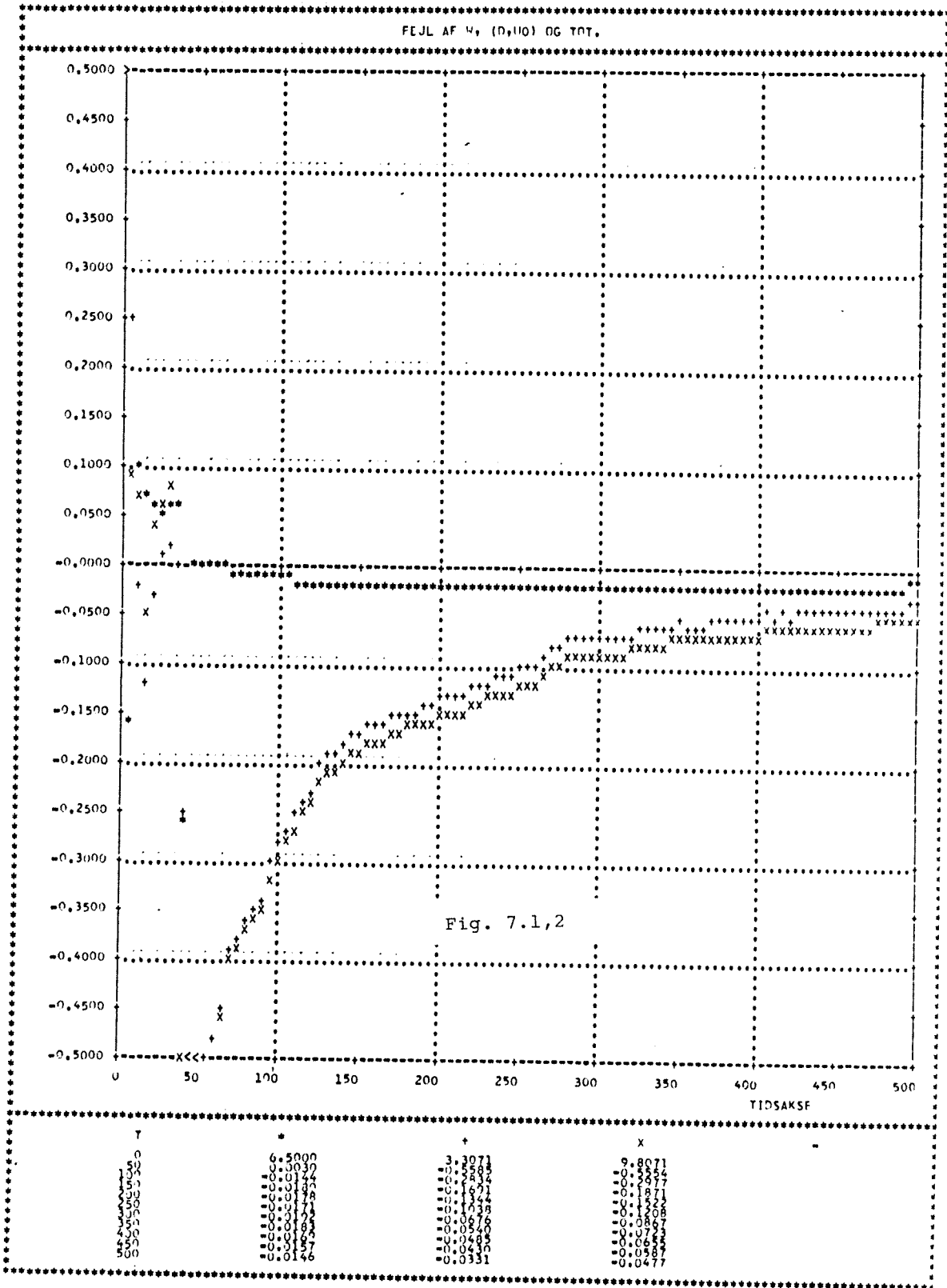
Den adaptive MYO - regulator har kvalitativt de samme egenskaber overfor en rampefunktion i $d(t)$ som overfor et spring.

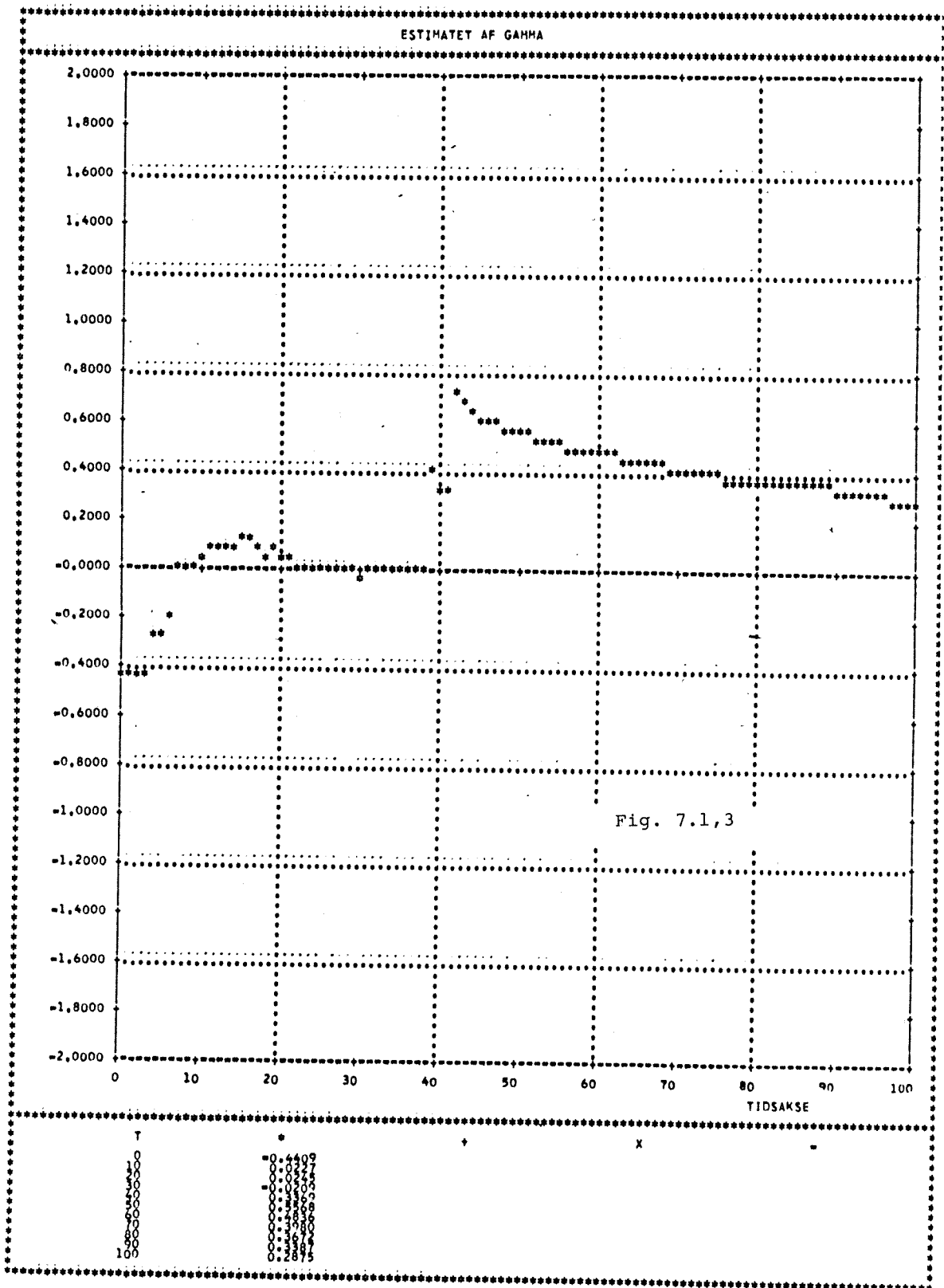
$$d(t) = \begin{cases} 0 & \text{for } t < 40 \\ 0.01(t-40) & \text{for } 40 \leq t < 140 \\ 1 & \text{for } t \geq 140 \end{cases}$$

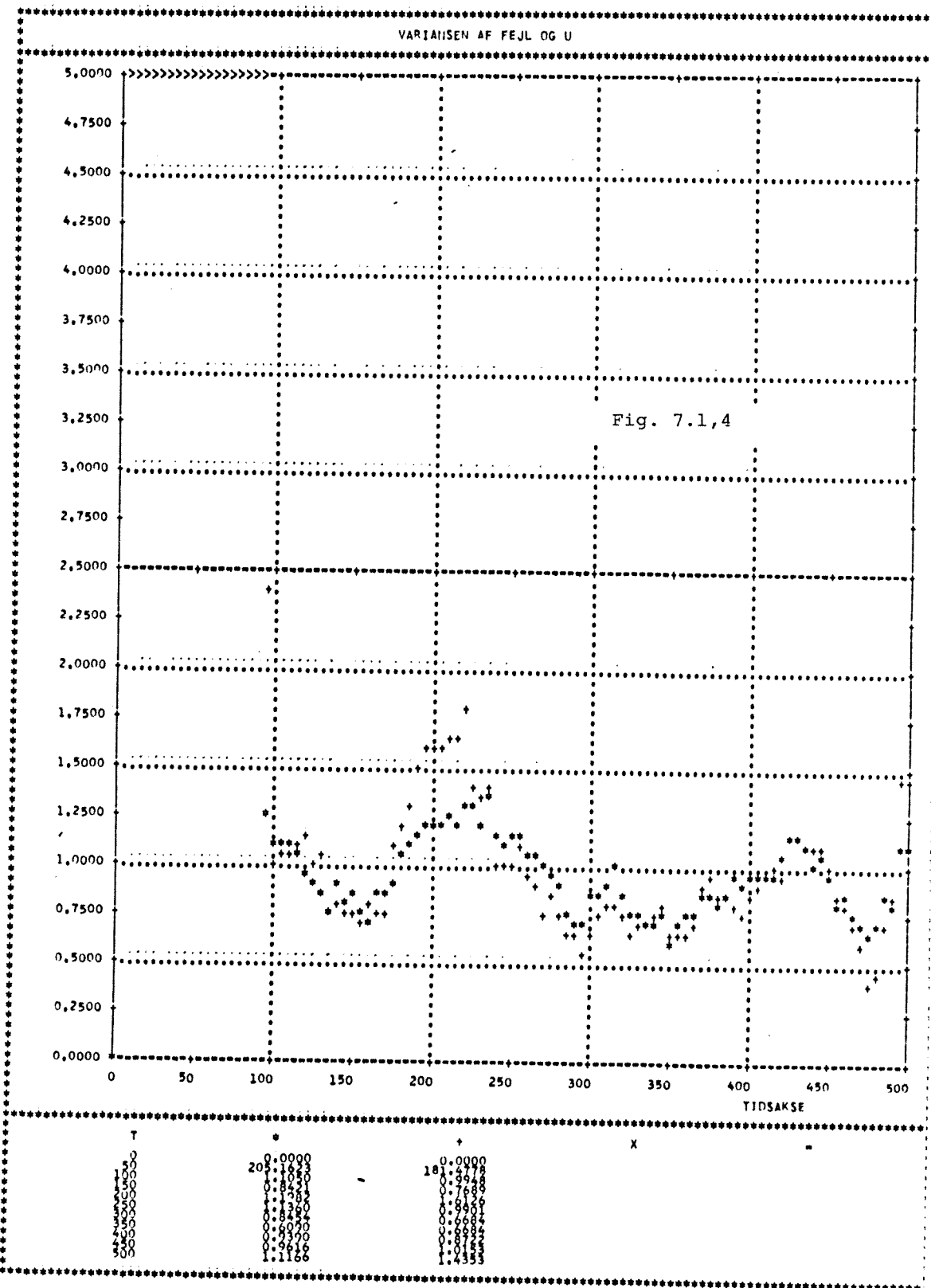
Rampefunktionens lavfrekvente karakter giver udslag i f. eks. tidsforløbet af AC - forstærkningen (fig. 18). Den adaptive deadbeatregulator har ikke bedre egenskaber overfor denne påvirkning end overfor springet i nævneværdig grad.

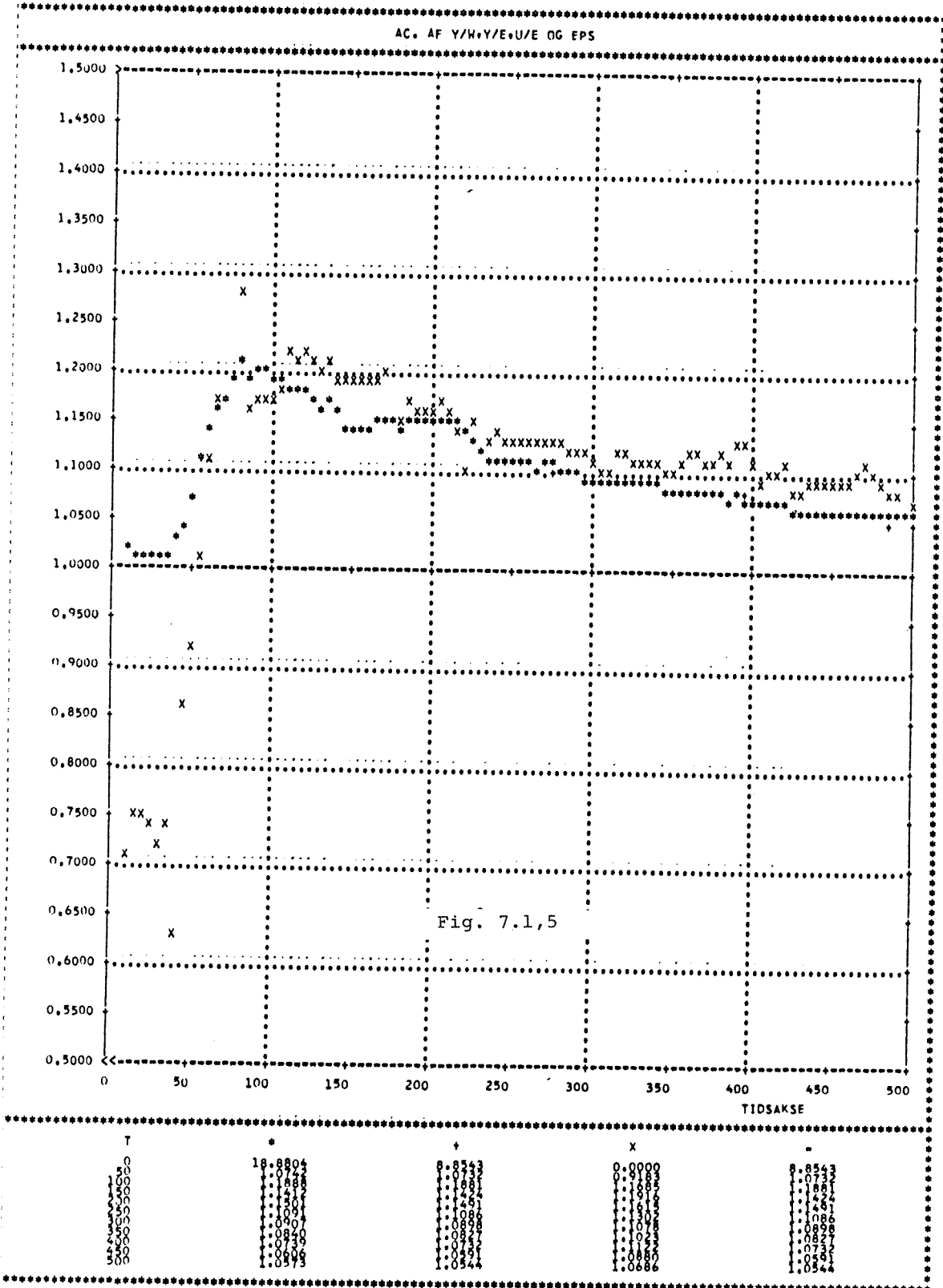
---ooo0ooo---

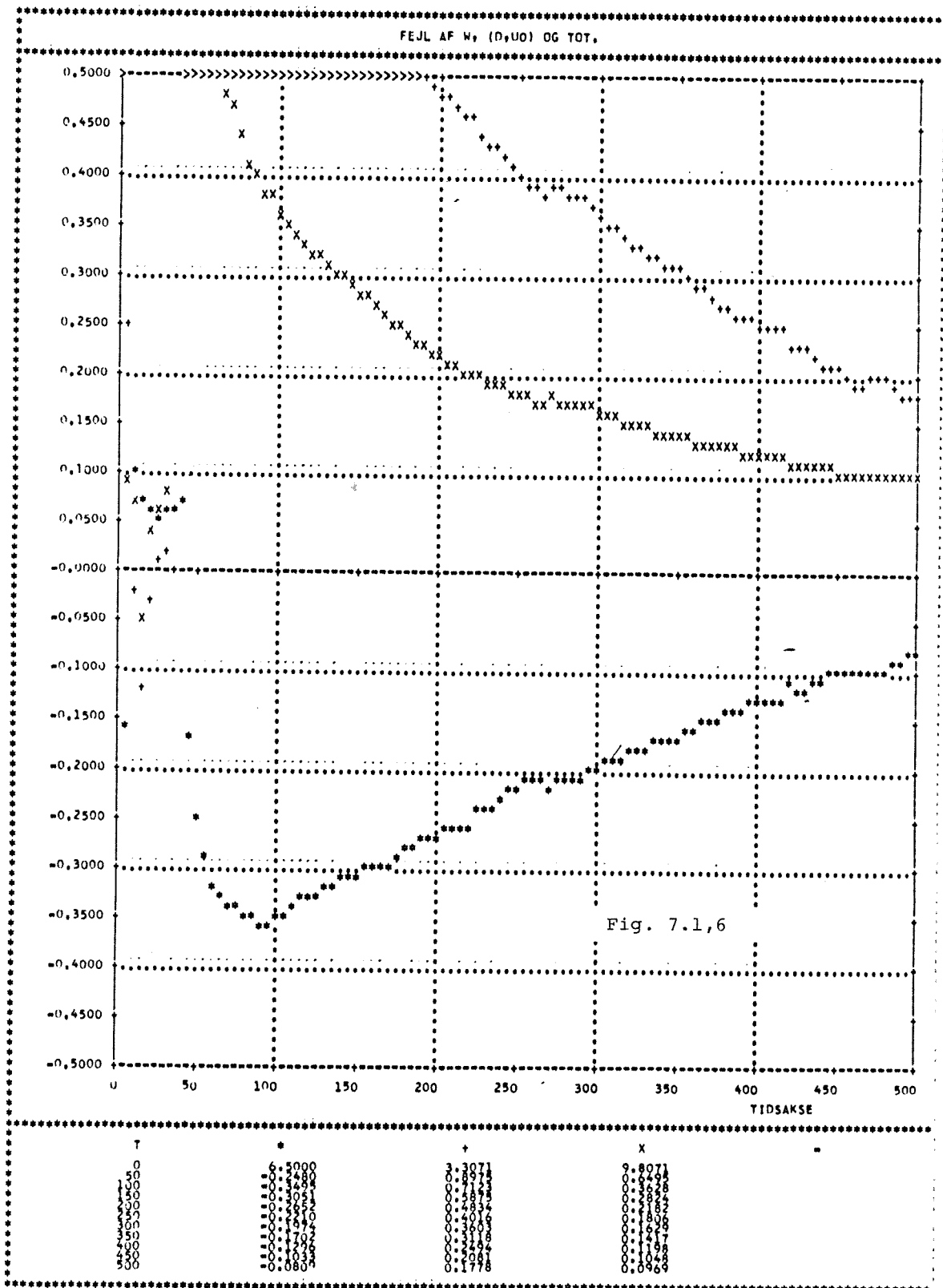


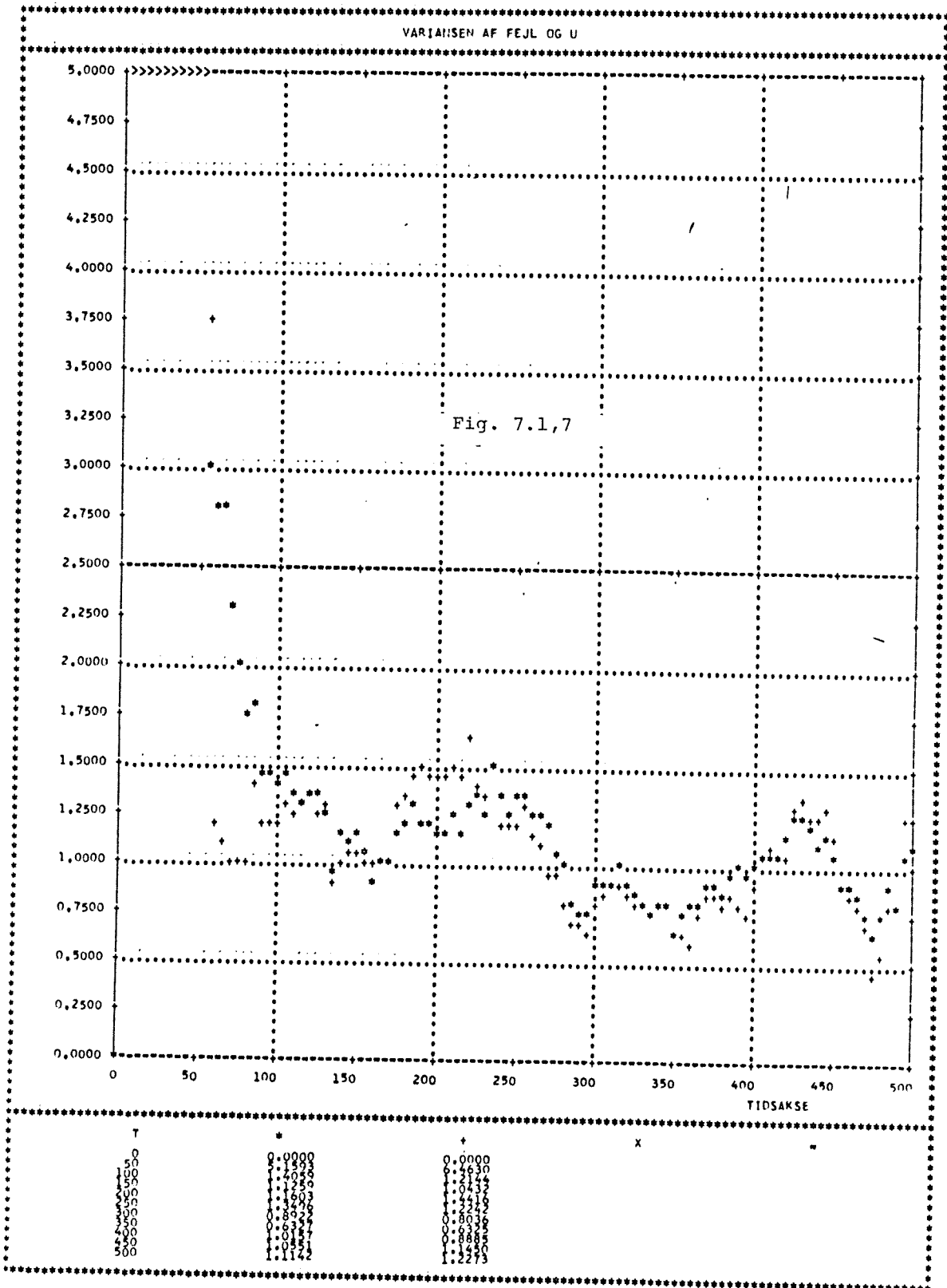


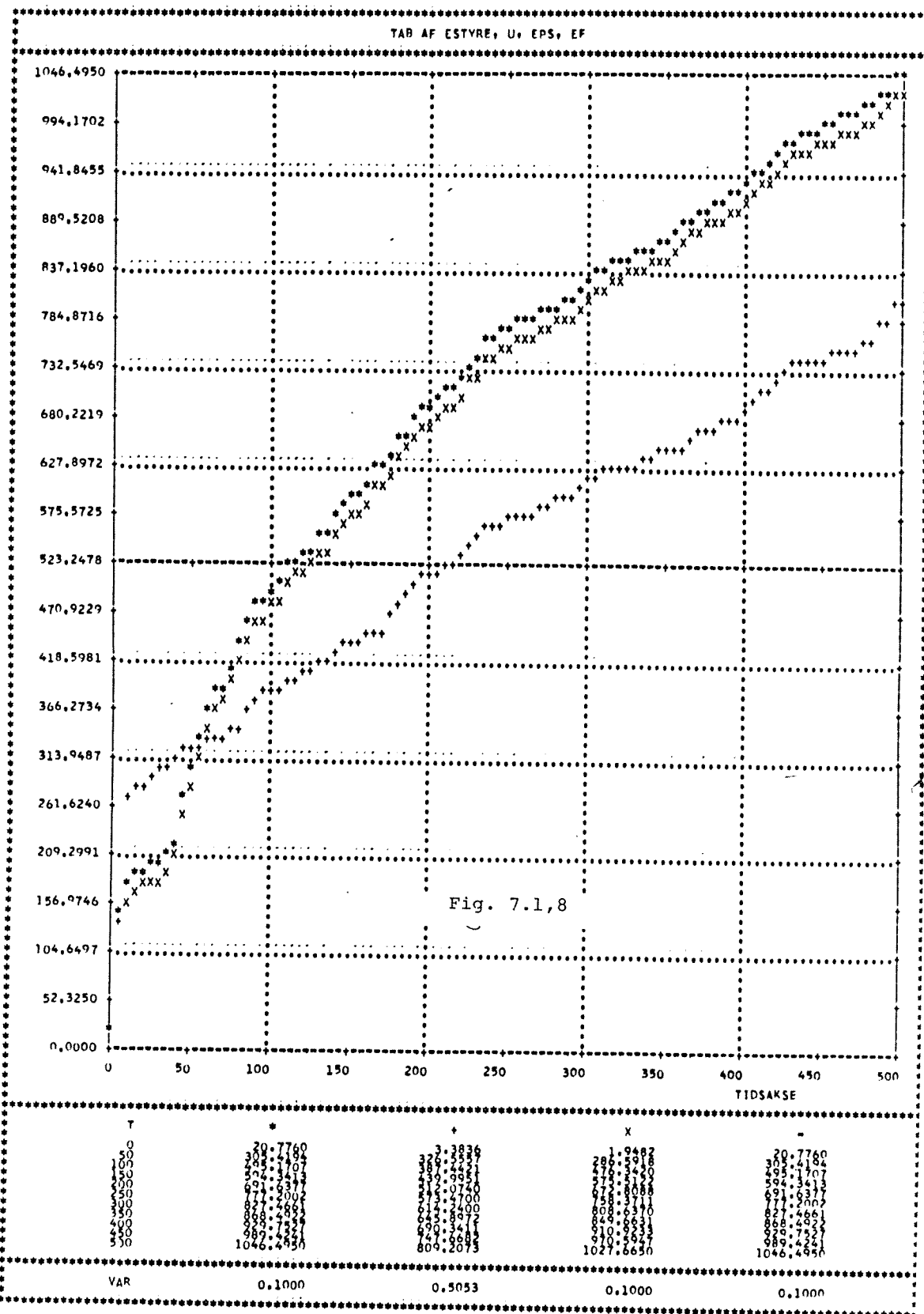


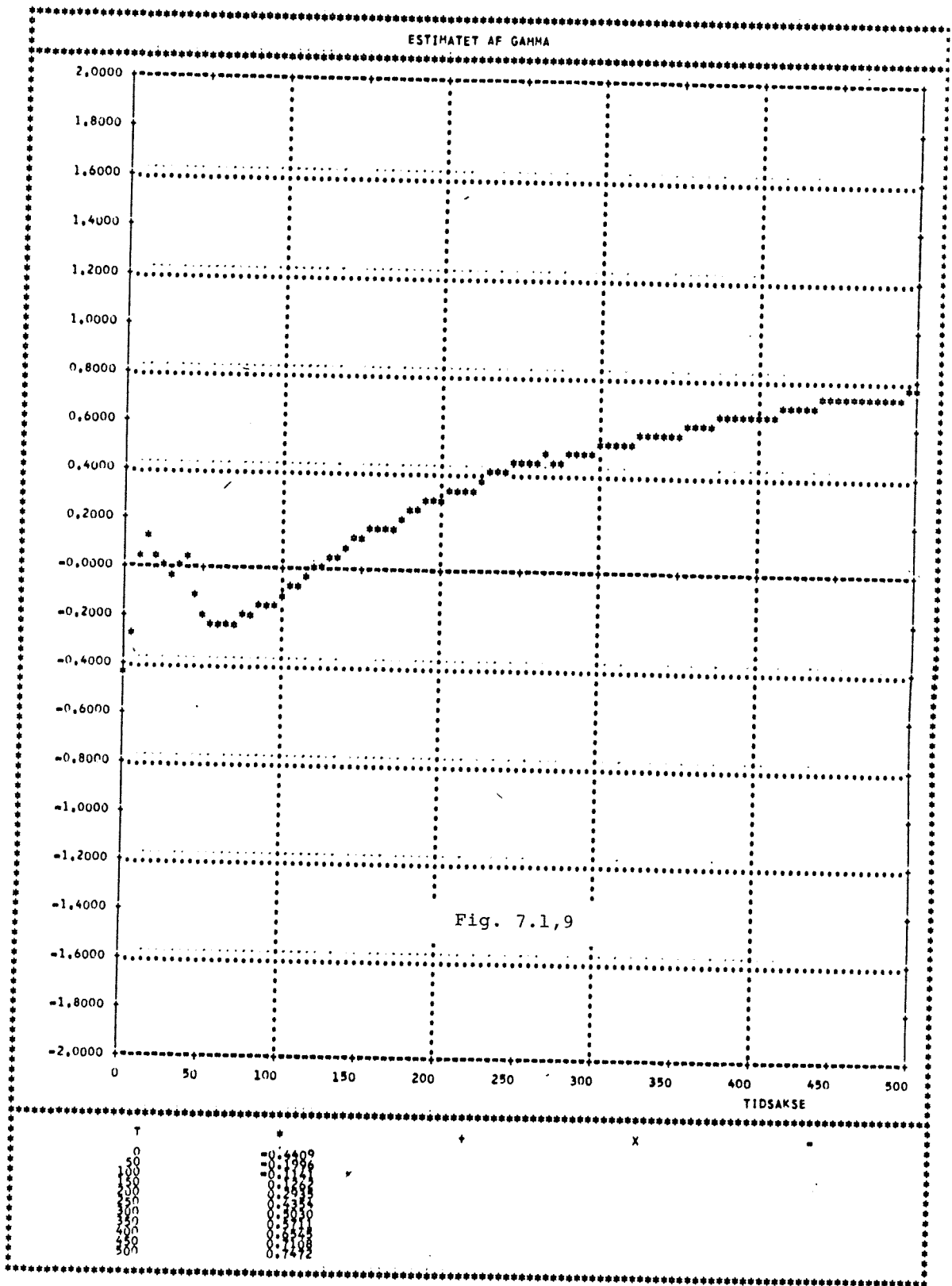


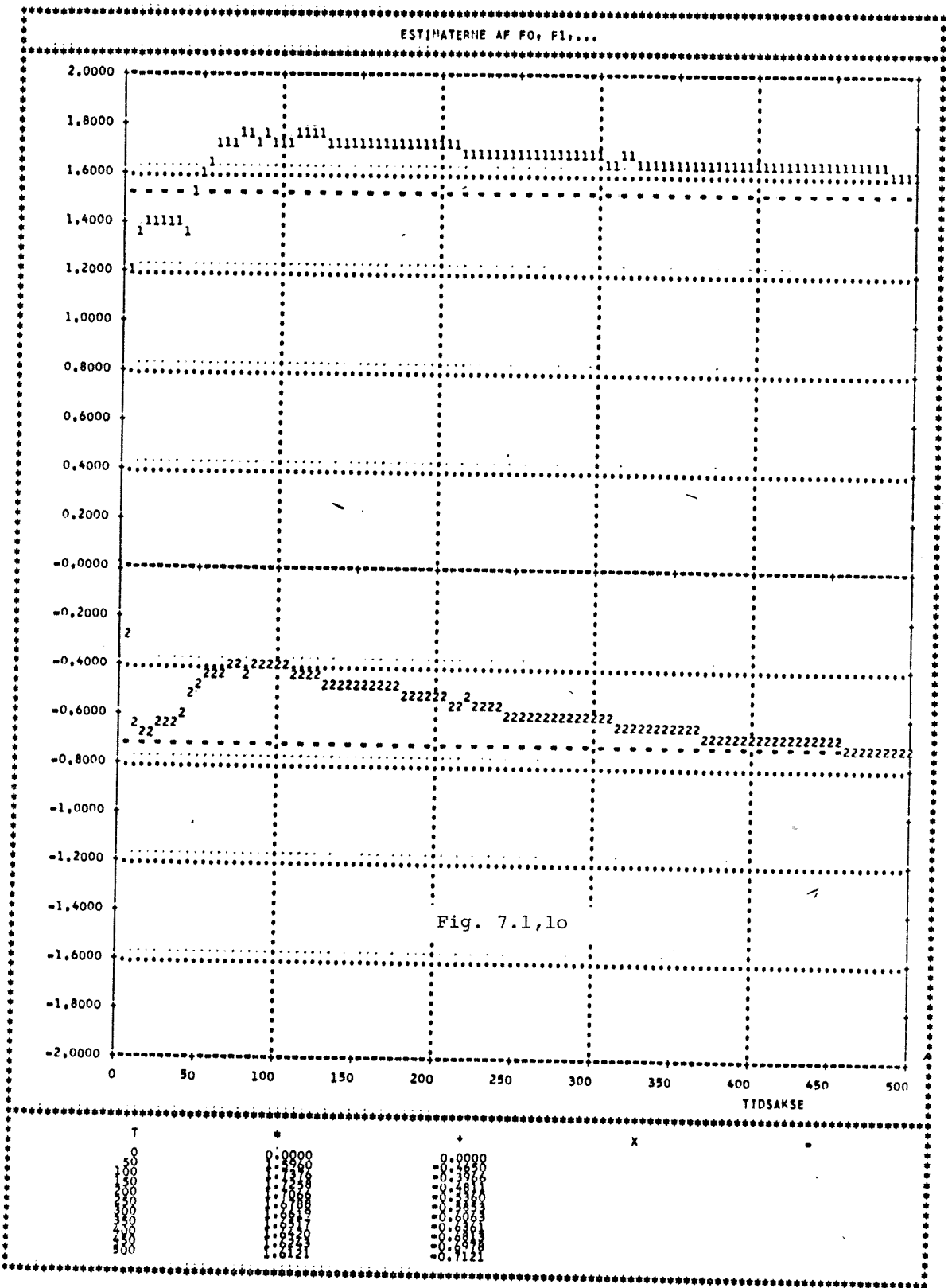


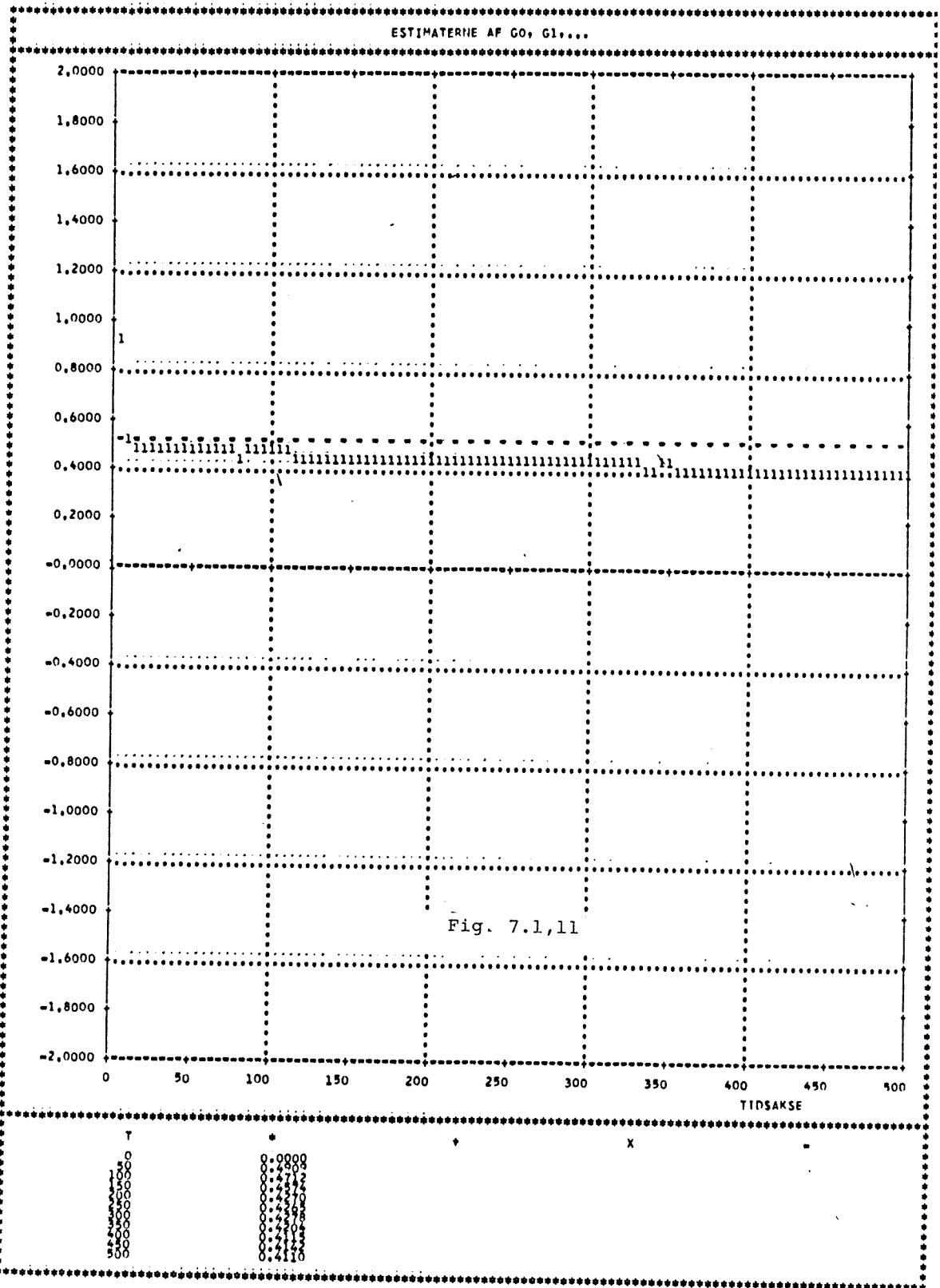


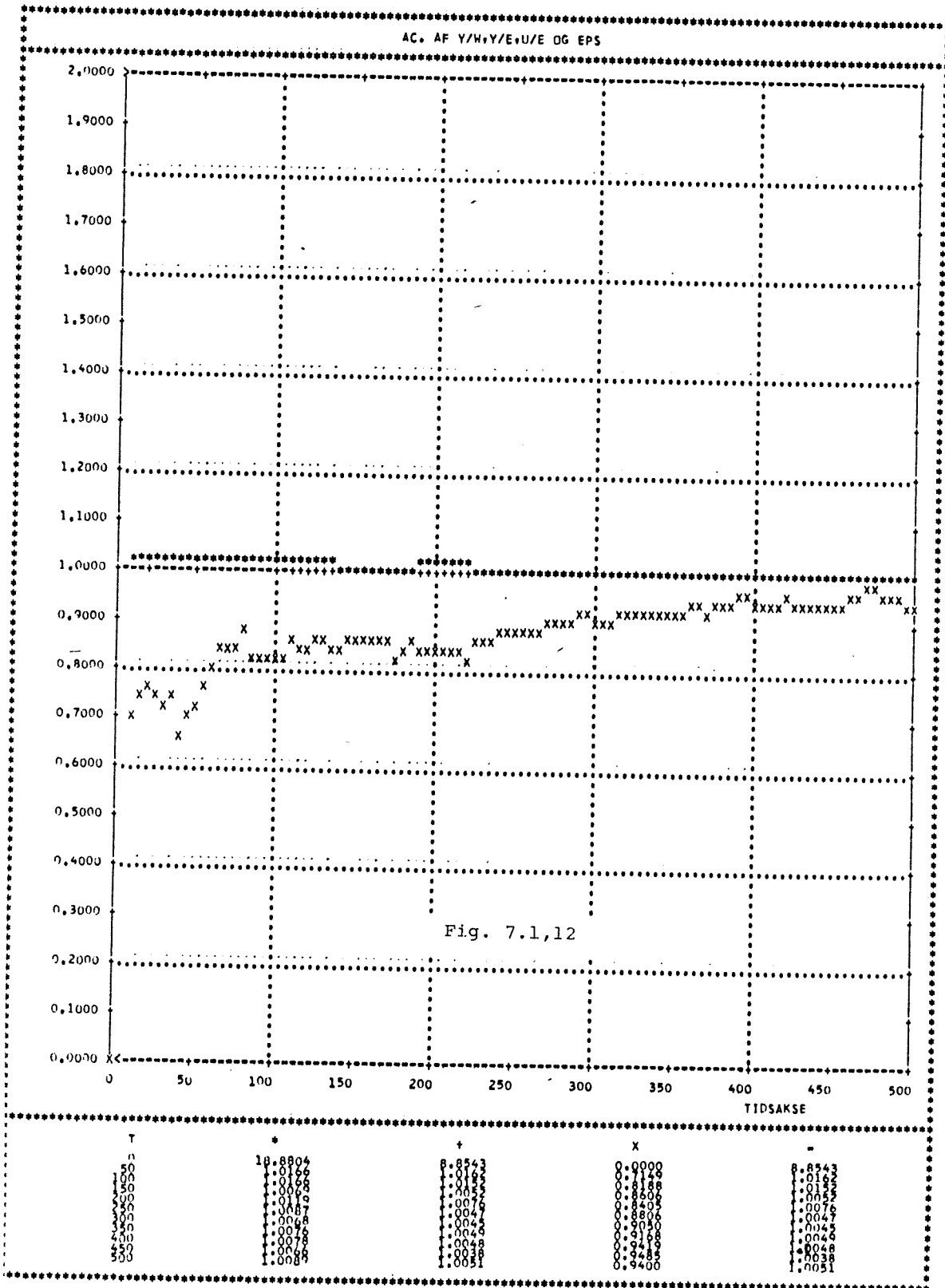


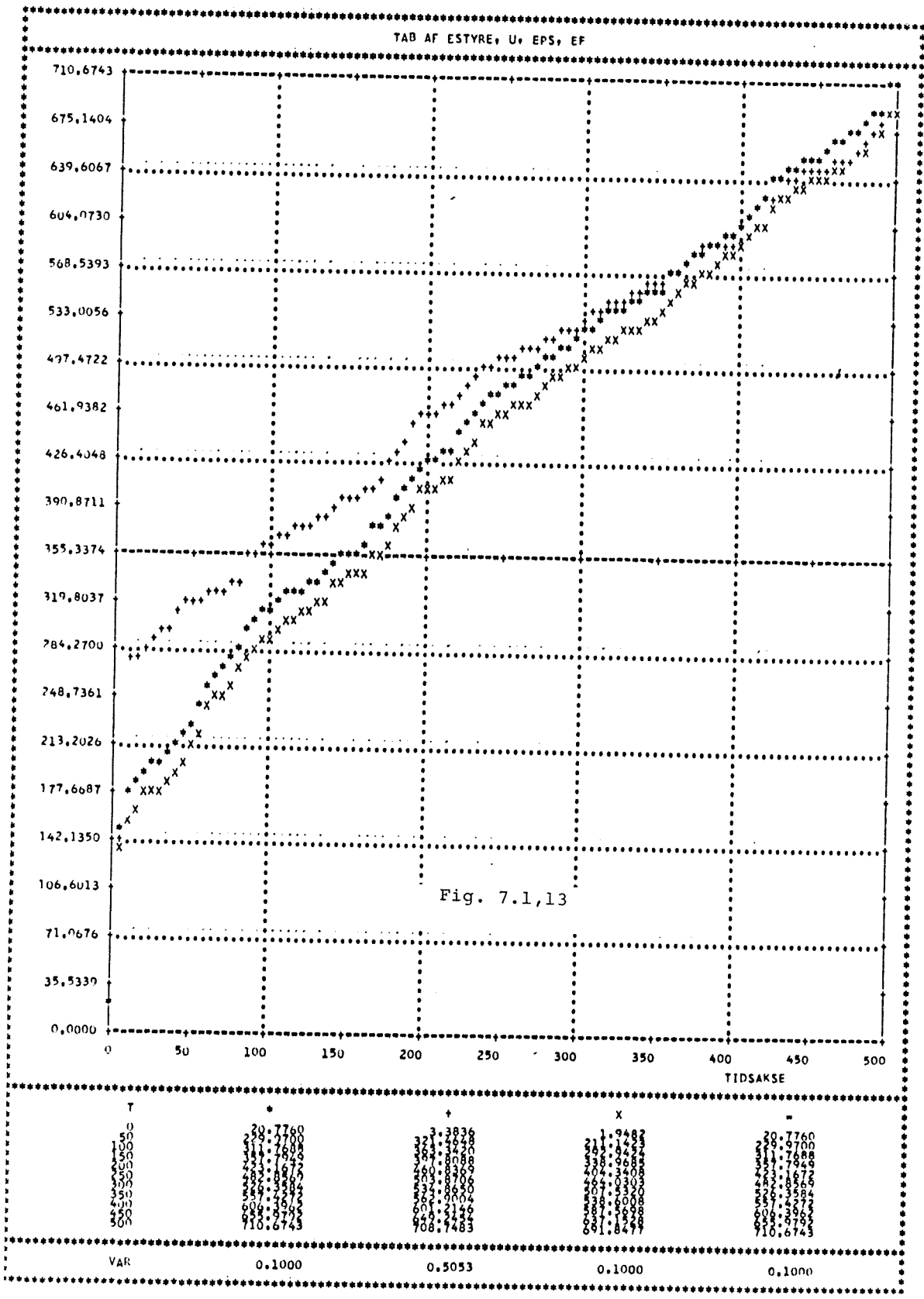


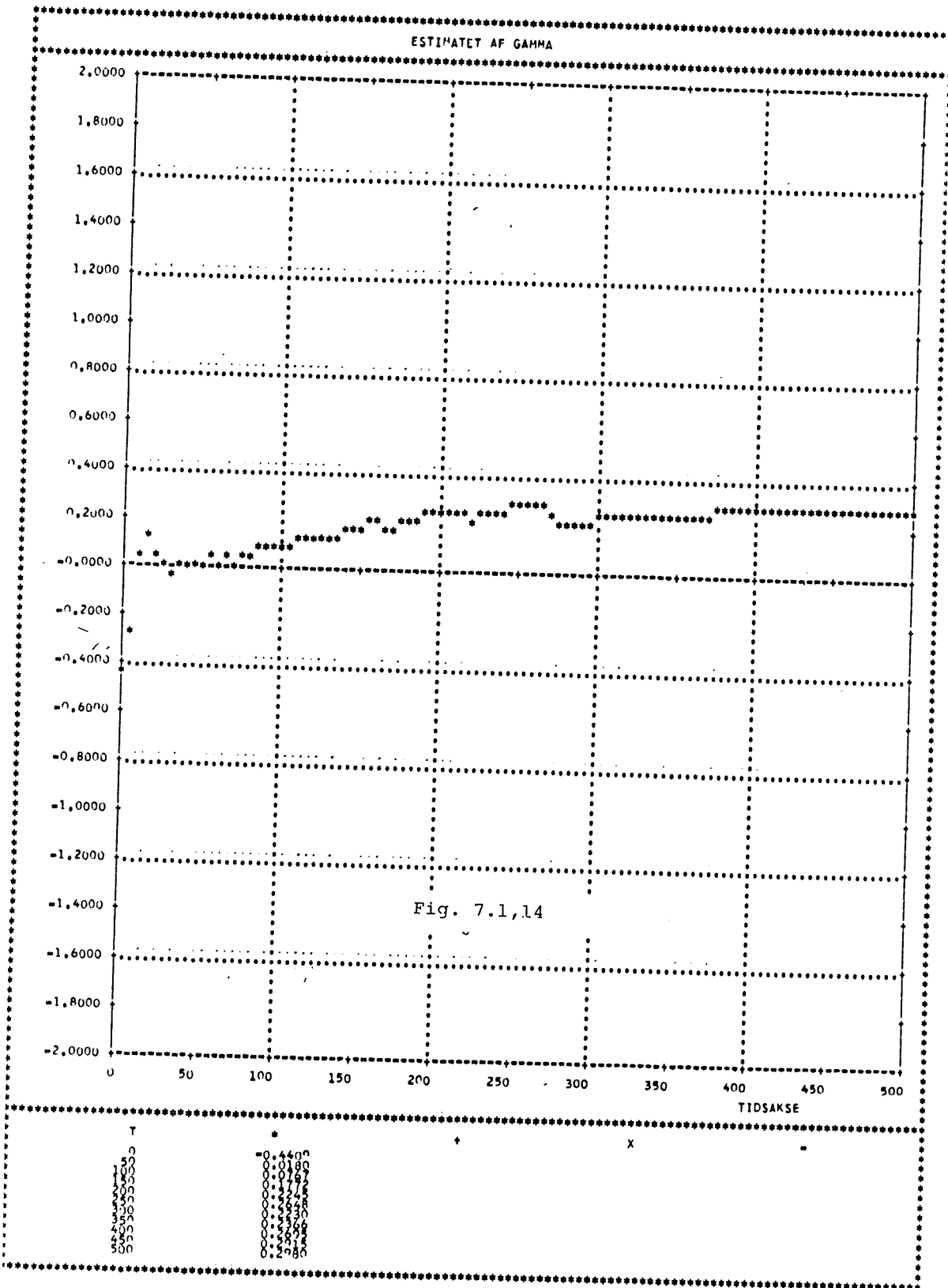


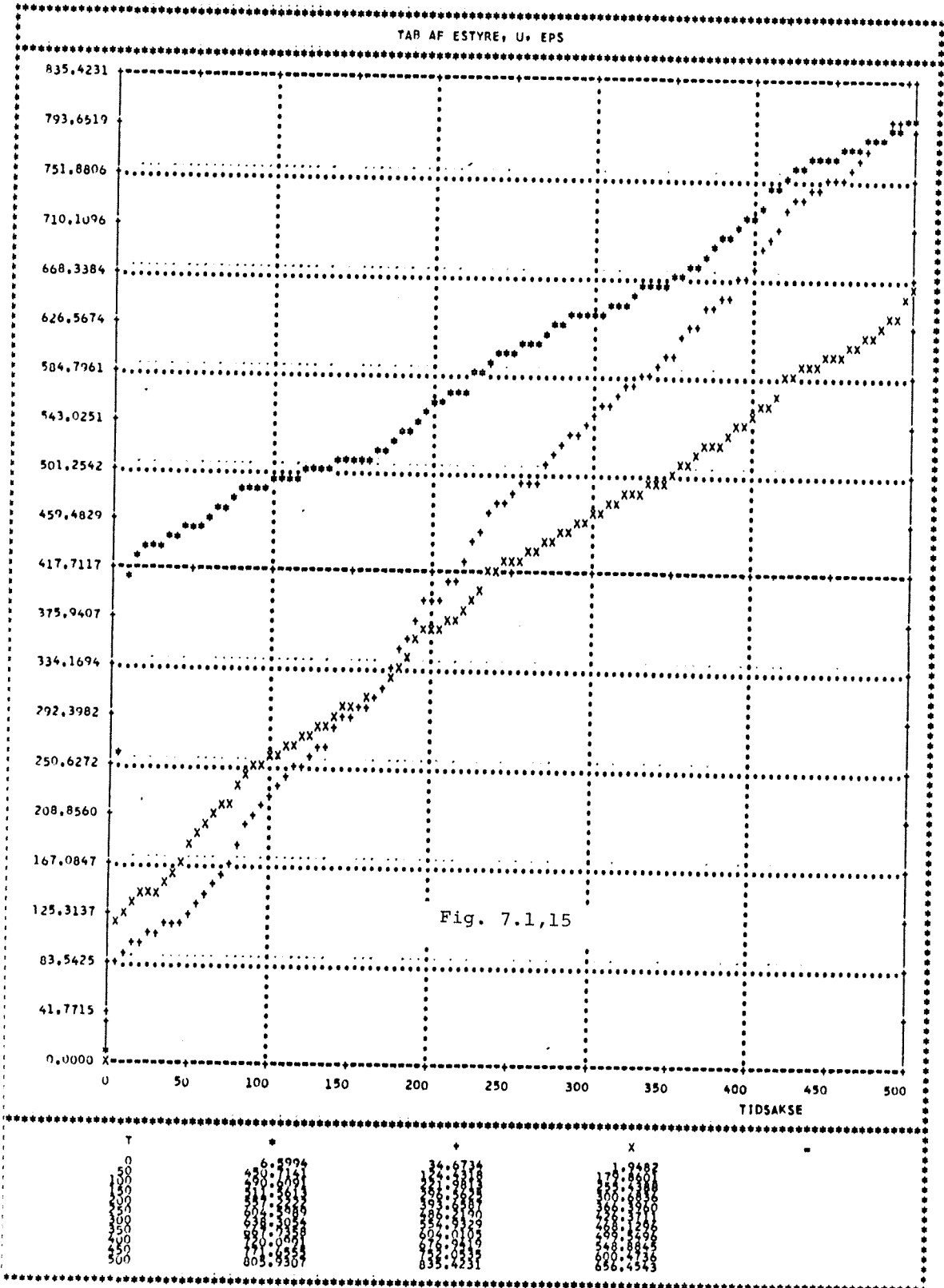


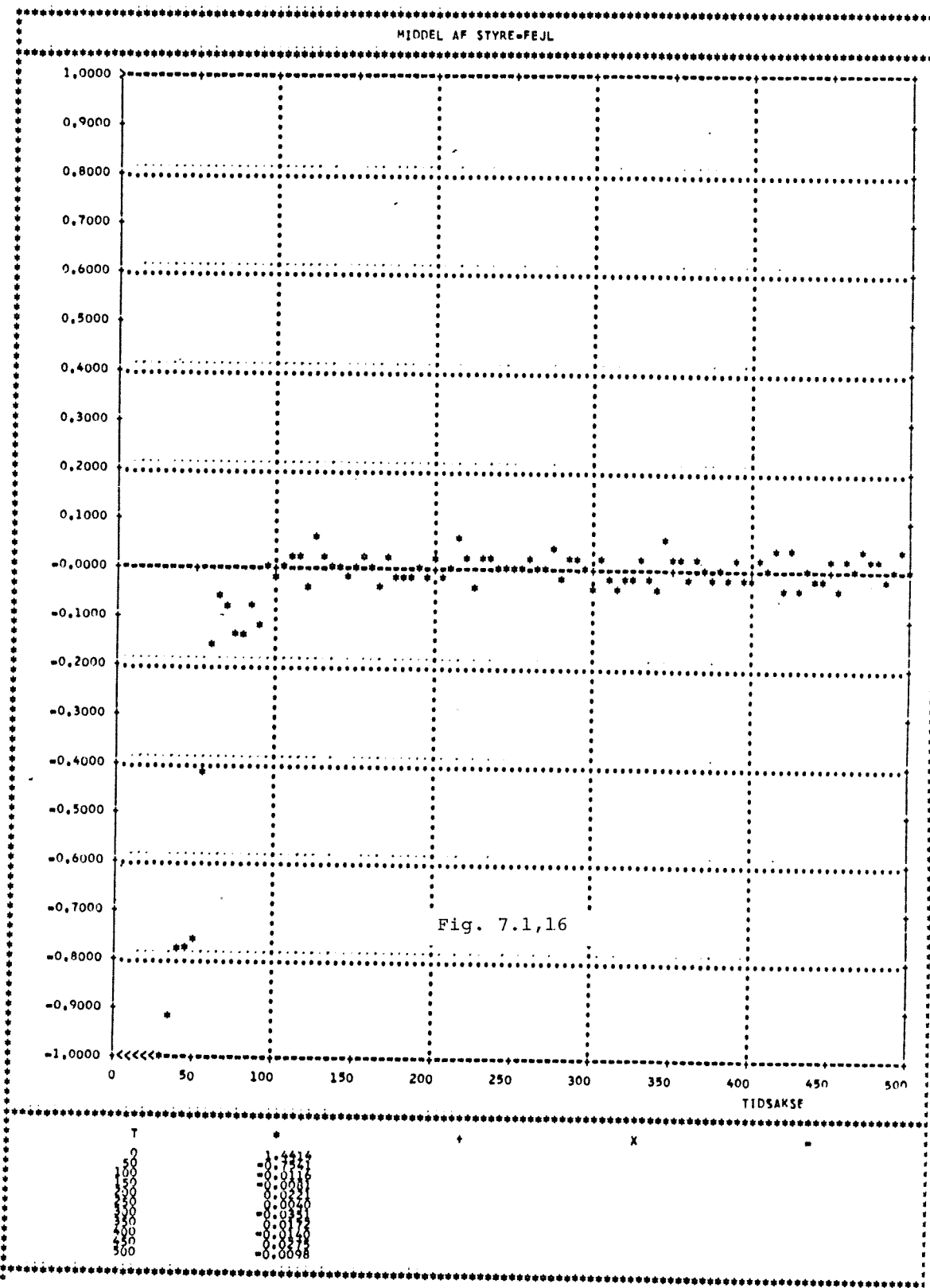


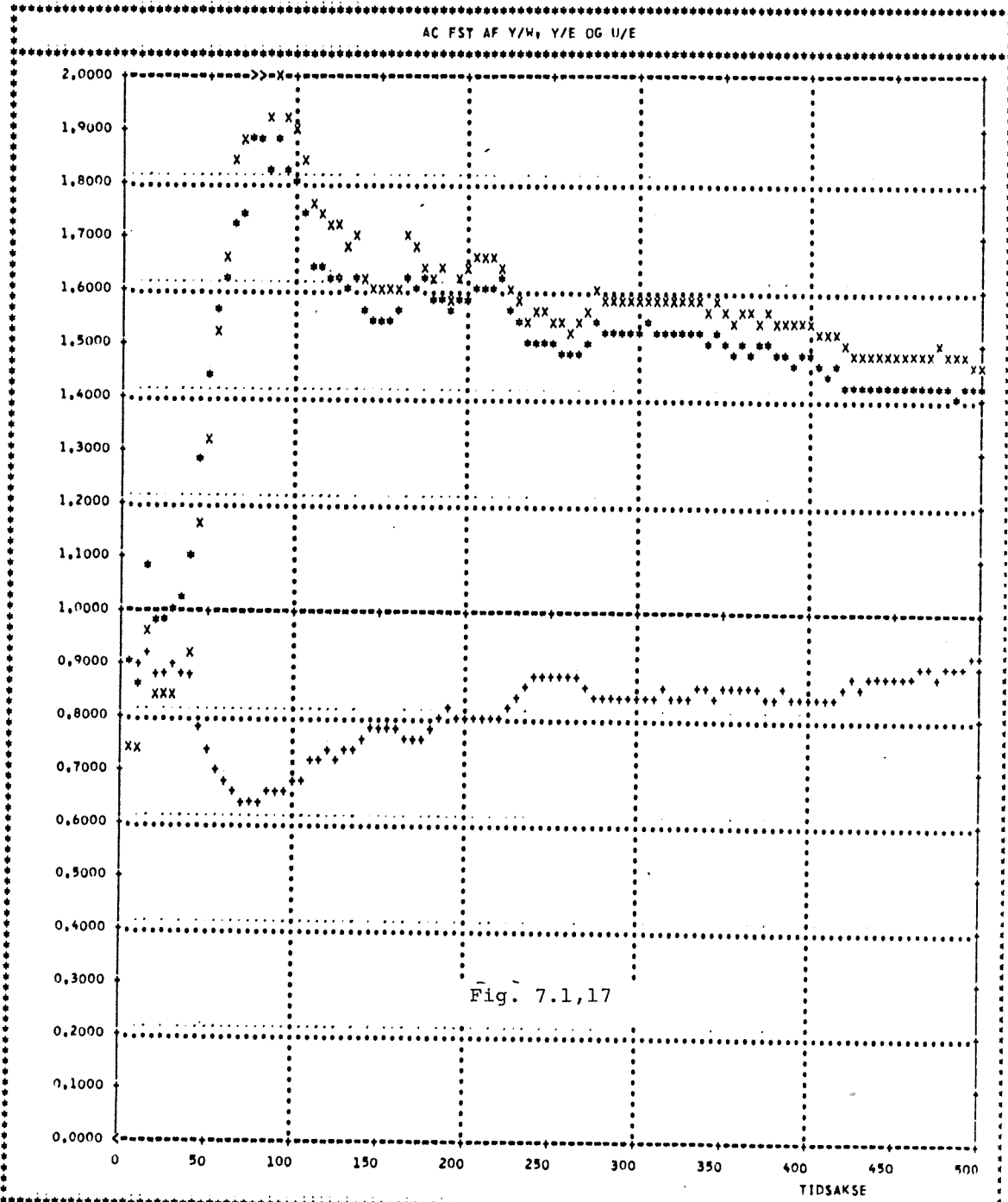




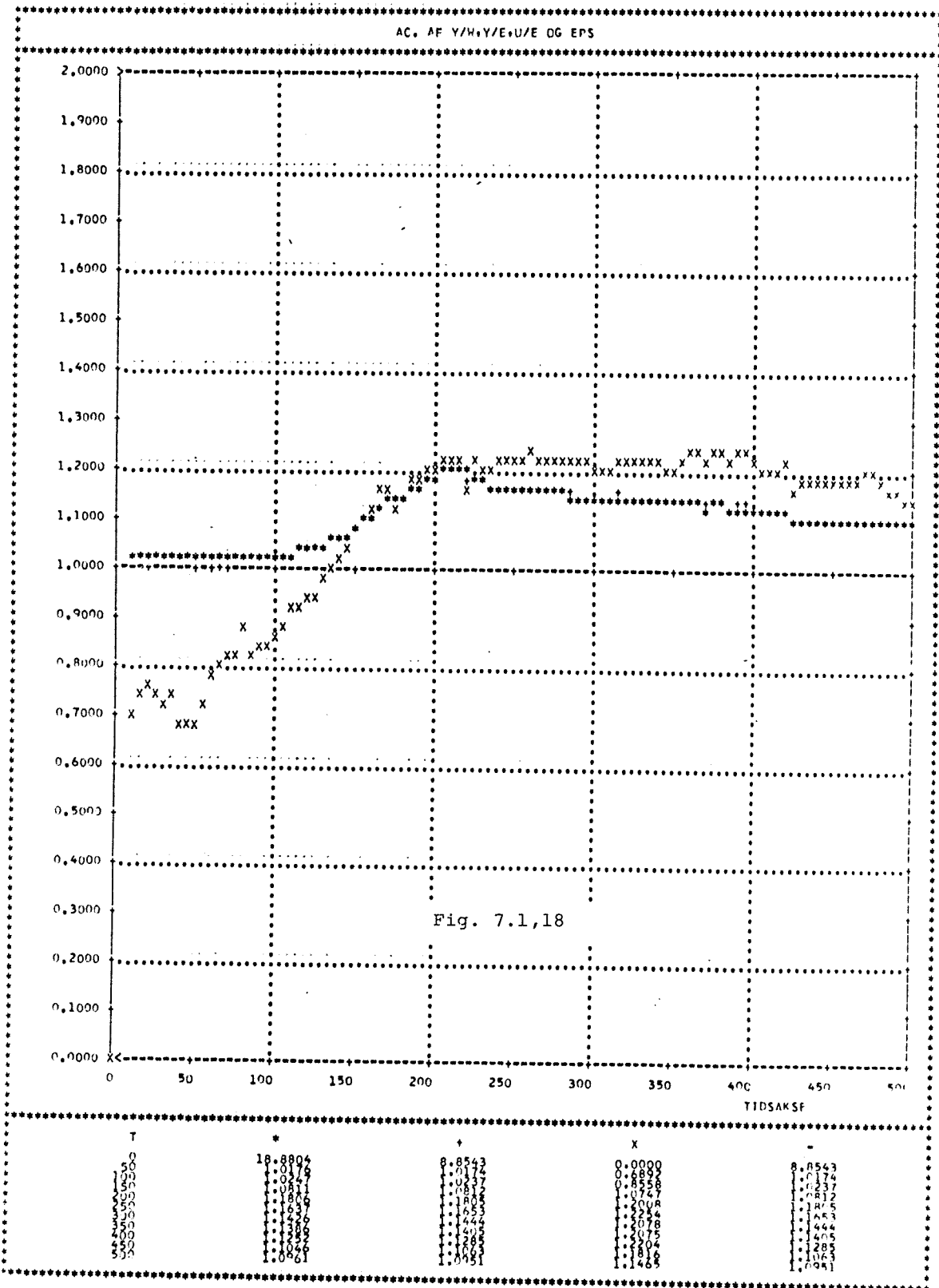








T	*	+	X
0	-18.0000	+3.1765	-22.5000
100	+1.4304	+0.7421	+1.3279
200	+0.8024	+0.6729	+1.0667
300	+0.3028	+0.7797	+1.5964
400	+2.7311	+0.8033	+1.6302
500	+1.3137	+0.8458	+1.7728
600	+0.9119	+0.8450	+1.5644
700	+2.7274	+0.8417	+1.3391
800	+1.4105	+0.8189	+1.4551



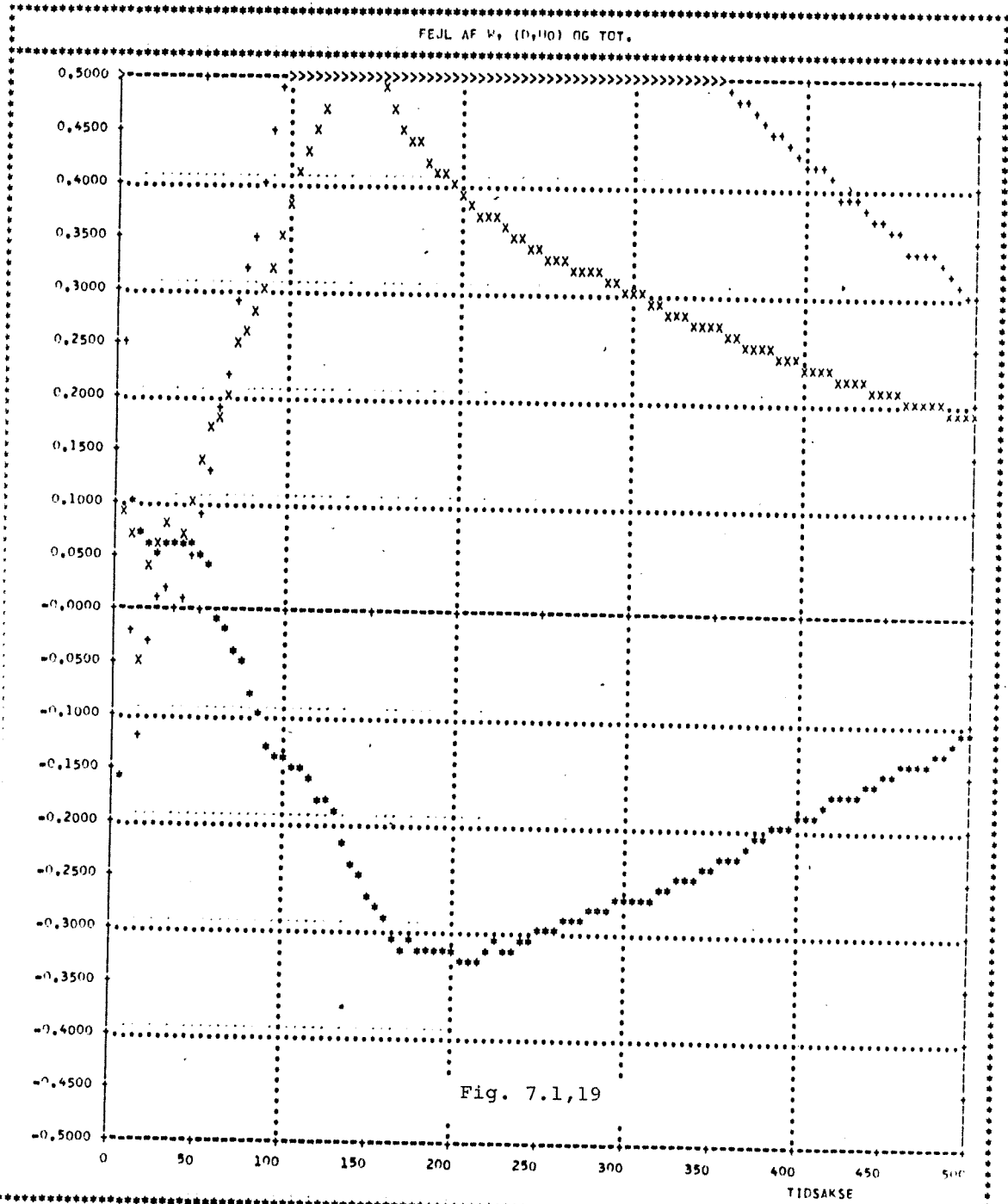


Fig. 7.1,19

T	+	+	x
0	0,3000	3,3071	9,8071
10	-0,0500	0,0885	0,1354
15	-0,0500	0,2477	0,3805
20	-0,0500	0,5707	0,5043
25	-0,0500	0,6378	0,3807
30	-0,0500	0,5749	0,3006
35	-0,0500	0,5078	0,2006
40	-0,0500	0,3240	0,0606
45	-0,1087	0,2474	0,1088
50			

7.2 Glemselsfaktor.

Som nævnt i kapitel 3 vil en ændring af estimationens kriteriefunktion til:

$$(1) \quad I_e = \sum_{i=0}^t \lambda_1^{t-i} \varepsilon^2(i) \quad 0 < \lambda_1 < 1$$

bevirke, at målinger indgår med en vægt i estimationen, der er eksponentielt aftagende. De estimerede parameter vil derfor med denne metode være mest påvirket af de sidst foretagne målinger. Glemselsfaktoren, λ_1 , skal vælges sådan, at horisonten, β , der kan defineres som løsningen til:

$$(2) \quad \lambda_1^\beta = 0.1$$

står i et rimeligt forhold til støjniveauet. Af hensyn til varianserne på parameterestimererne må λ_1 ikke vælges for lavt, specielt ikke når støjniveauet er højt. Ved lavere støjvarianser kan estimationshorisonten, β , formindskes, blot ikke så meget at der kan opstå windup i kovariansmatricen. Når der anvendes en glemselsfaktor ($\lambda_1 < 1$) er der skabt mulighed for at følge langsomt varierende parameter. Til det problem, der her foreligger, hvor kun belastningen $\hat{d}(t)$ varierer, har metoden den ulempe, at forstærkningen til estimererne af de konstante parameter ikke konvergerer mod 0. En ændring af belastningen vil derfor ikke alene give anledning til en variation af estimatet af denne men også af de øvrige parameterestimerer.

En metode til at undgå dette er at estimere $\hat{d}(t)$ separat efter udtrykket:

$$(3) \quad \hat{d}(t+1) = \hat{d}(t) \lambda_0 \left(1 - \frac{1}{t} \right) + \varepsilon(t) \left(1 + \frac{1}{t} - \lambda_0 \right)$$

der stationært svarer til et første ordens filter med overføringsfunktionen:

$$(4) \quad \hat{d}(t) = q^{-1} \frac{1 - \lambda_0}{1 - \lambda_0 q^{-1}} \varepsilon(t)$$

De øvrige parameter i θ opdateres på sædvanlig vis ($\lambda = 1$). Metoden lider imidlertid ved at give en temmelig langsom og iøvrigt dårlig konvergens. Dette kan skyldes, at i starten indgår residualerne med for stor vægt ($\lambda \neq 1$) i forhold til:

$$(5) \quad \hat{d}(t) = \frac{1}{t} \sum_{i=0}^t \varepsilon(i)$$

Den dårlige konvergens skyldes, at residualerne:

$$(6) \quad \varepsilon_0(t) = e(t) - \frac{1 - \lambda_0}{1 - \lambda_0 q^{-1}} e(t-1)$$

for korrekte estimater (korrekt middelværdi for \hat{d}) er korrelerede, hvorved de korrekte parameter ikke er et sandsynligt konvergenspunkt, når $\lambda_0 \neq 1$.

I simulationerne er der opnået bedre resultater med en kombination af (5) for $t < 50$ og :

$$(7) \quad \hat{d}(t) = \frac{1}{50} \frac{1 - q^{-49}}{1 - q^{-1}} \quad t \geq 50$$

Også her er residualerne

$$(8) \quad \varepsilon_0(t) = e(t) - 0.02 \frac{1 - q^{-49}}{1 - q^{-1}} e(t-1)$$

korreleret for korrekte estimater.

Udover fejlagtige estimater lider disse to metoder af, at

estimationen ikke kan følge med spring i belastningen. Er de øvrige parameterestimerer ikke konvergeret så godt, at forstærkingen til disse er neglisibel, vil de blive drevet bort fra deres oprindelige leje. Opmærksomheden vil derfor være centreret om den første metode, der er anvendt i de følgende dokumenterede simulationer med MVo - regulatoren.

Formålet med at bruge en glemselsfaktor er at følge lang -
somme variationer og mindre spring i belastningen, $d(t)$. Det er desuden ret interessant at se, hvilken skadelig indvirkning spidser har på reguleringen. For at følge et spring skal der vælges en tilpas lav værdi for λ_1 . Er denne på den anden side valgt for lille, vil den stationære reguleringskvalitet for konstant belastning være forringet på grund af fluktuationer i parameterestimererne. Denne forringelse afhænger naturligvis af, hvor følsom den pågældende regulatorstruktur er for parametervariationer i forbindelse med det foreliggende objekt. I fig. 1 og 2 er tidsforløbet DC - fejlen og AC - forstærkingen vist for en simulation, hvor der anvendes en glemselsfaktor på 0.985, hvilket svarer til en tidshorisont på 150 tidsenheder. Det ses, at i forhold til $\lambda = 1$ (fig. 6.1.1,9 - 10) fluktuerer specielt DC - fejlen temmelig meget. Med denne værdi for glemselsfaktoren er tabet for udgangssignalet, $y(t)$ målt over de sidste 400 trin steget 2.5%.

Drift.

Når objektet påtrykkes den i afsnit 7.1 omtalte rampefunktion, der kan henregnes til de langsomme variationer i $d(t)$, har estimererne i dette tilfælde bedre mulighed for at følge denne påvirkning. Sammenlignes specielt DC - egenskaberne (fig. 3 og fig. 7.1,19) er dette åbenlyst. Fig. 4 viser AC - forstærkingen (sml. 7.1,18). Tabene ($0 \leq t \leq 500$), hvori start - tabene er inkluderet, fremgår af nedenstående tabel.

	tab(y)	tab(u)
med rampe og λ_1	823,3	784,6
med rampe	1170,4	798,0
uden rampe	666,1	694,3

For udgangssignalet er der vundet en del. Det vil senere vise sig, at der med andre metoder kan opnås bedre resultater.

Spring.

Sammenlignet med standardalgoritmen er der, som det fremgår af fig. 5 - 11, også opnået forbedrede egenskaber overfor et enhedsspring i $d(t)$. Sammenlignes tabene bemærkes, at disse for udgangssignalets vedkommende er blevet mindre, samt at forstyrrelsernes indflydelse er bortkompenseret efter ca. 100 trin. Forholdene er iøvrigt som ved standard algoritmen, når man ser bort fra flutationerne i parameterestimerne samt den opnåede hurtighed, hvormed springet absorberes i parameterestimerne. Mest markant er den hurtighed, hvormed DC - fejlen returnerer (sml. fig. 6 med 7.1,6). Mindre spring af samme størrelsesorden som støjniveauet kan også følges i dette tilfælde.

Spids.

Ved anvendelse af en glemselsfaktor vil determinanten til parameterestimaternes kovariansmatrice, $P(t)$, ret hurtigt antage en stationær værdi, der dels afhænger af den brugte glemselsfaktor og dels afhænger af støjniveauet. Denne effekt giver anledning til den forøgede hastighed, hvormed springet i belastningen absorberes i $\hat{y}(t)$ (for MVo - regulatorens vedkommende). Denne effekt afstedkommer desværre også, på grund

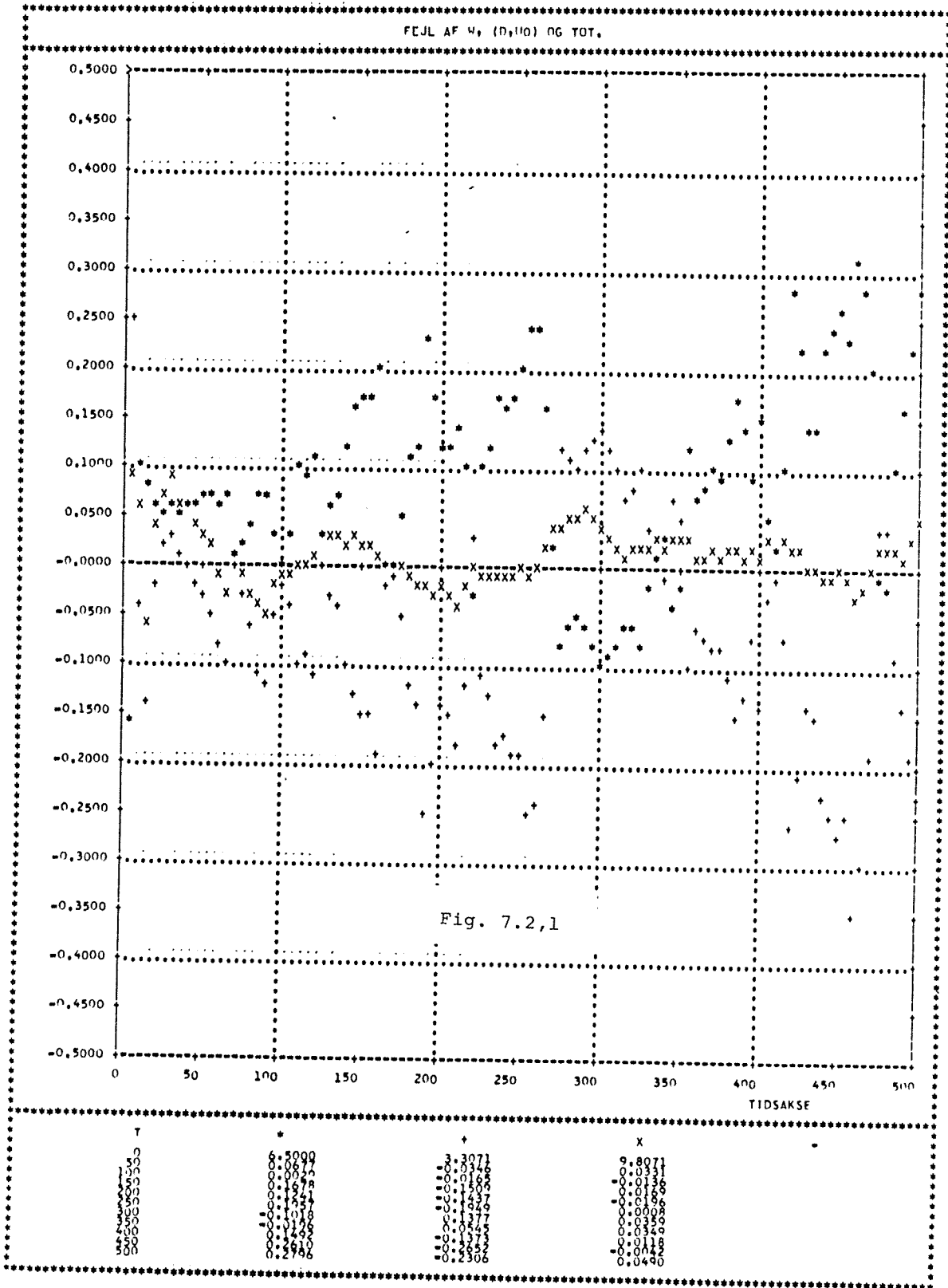
af den forøgede forstærkning, at en spids får parameterestimerne til at springe længere væk fra deres oprindelige værdi. Til gengæld er indsvingningsforløbet for $\hat{\gamma}(t)$ hurtigere end i standard tilfældet. De øvrige parameter bringes som regelen på plads af den efterfølgende eksitation.

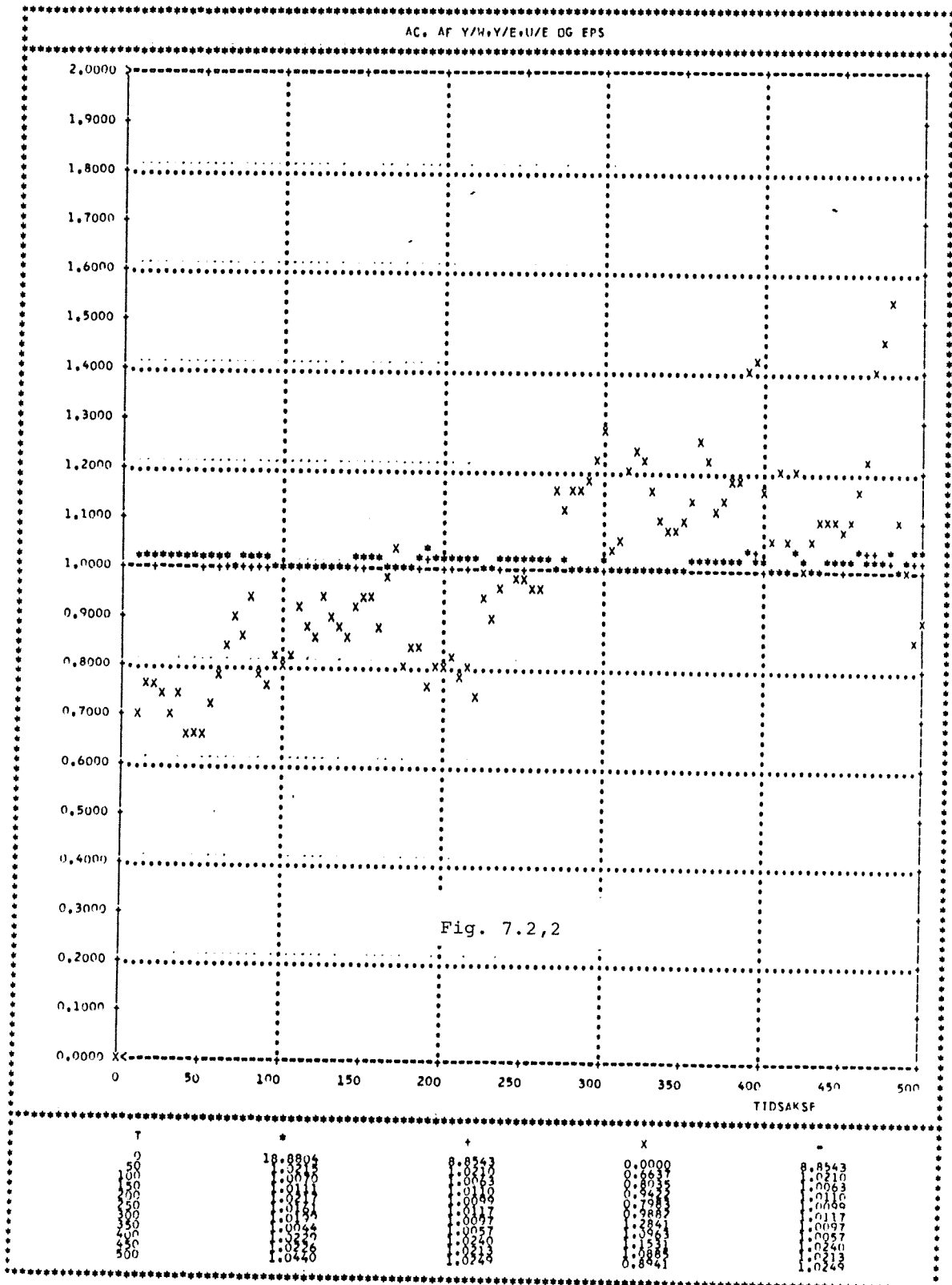
Deadbeat - regulatoren er øjensynligt mere følsom overfor fluktuationer i parameterestimerne, når de anvendes overfor dette testobjekt. For at rimeligt stabile estimer er det nødvendigt at hæve glemselsfaktoren til 0.99, hvilket svarer til en tidshorisont på 230 trin. I fig. 12 ses, at de kraftige variationer i AC - forstærkningen tyder på, at denne værdi for glemselsfaktoren ligger i underkanten. Fig. 13 og 14 viser de tilsvarende forløb, når objektet påvirkes af en rampe og et spring i $d(t)$ på 0.3. Som en simpel følge af at det ikke er muligt, at bruge en lavere værdi for glemselsfaktoren uden en kraftig forringelse af reguleringskvaliteten, er deadbeat - regulatorens egenskaber overfor disse to påvirkninger ret ringe med dette testobjekt. For det lille spring der kan absorberes af MVo - regulatorens estimator uden glemselsfaktor, har AC - forstærkningen her et indsvingsforløb på ca. 250 trin svarende til tidshorisonten for estimationen. Sammenlignet med DBo - regulatorens standard algoritme er der dog opnået en betydelig forbedring (sml. 7.1,18). Sammenholdes MVo - med DBo - regulatoren skal det huskes, at i de adaptive deadbeat - regulatore estimeres en parameter mere end i de øvrige, der kræver større a priori information.

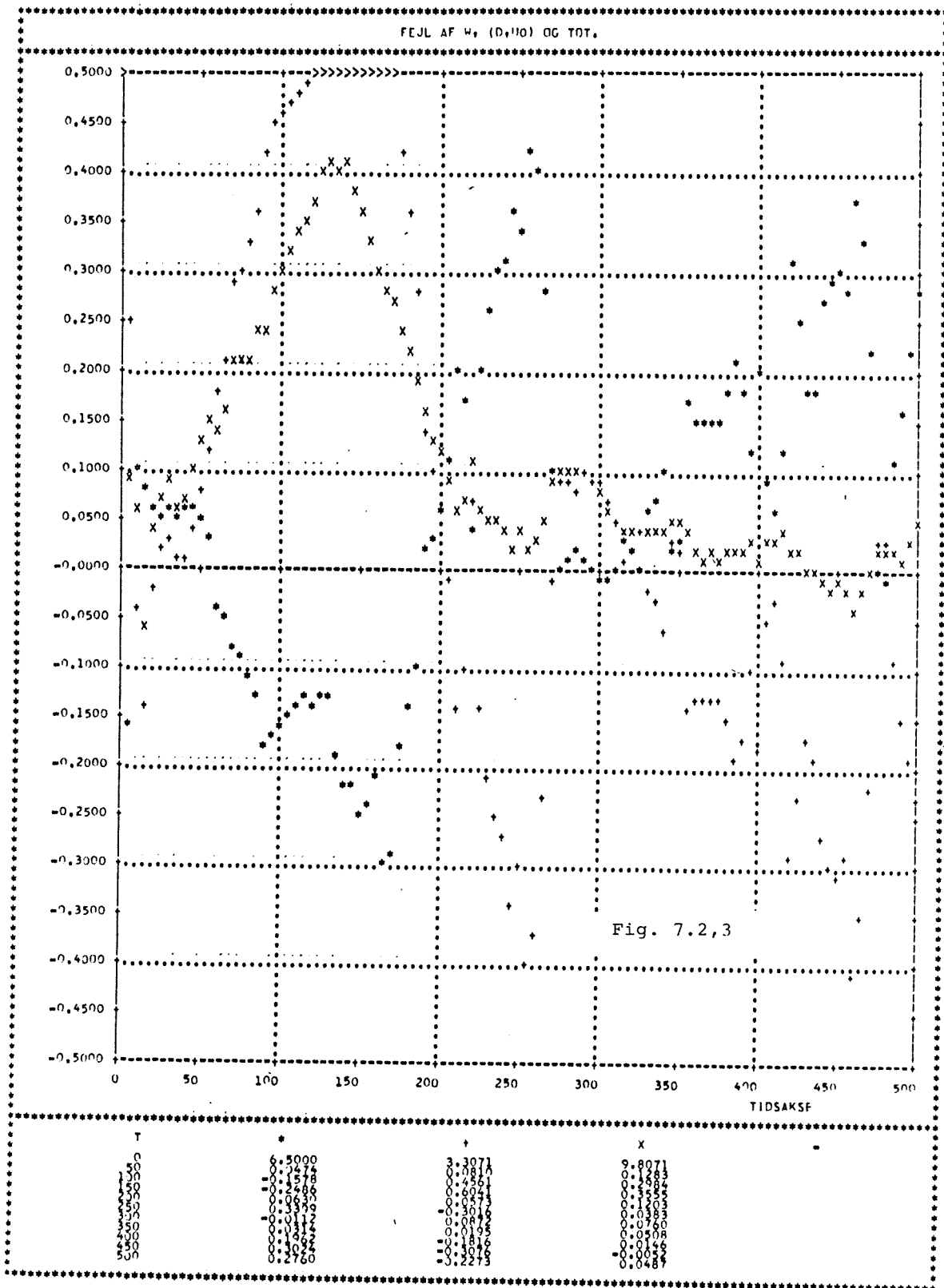
Som ventet er der med en glemselsfaktor opnået en forøget hastighed for absorption af en variation i belastningen, men samtidig en øget sårbarhed overfor spidser. En fejl som følge heraf er under alle omstændigheder hurtigere udkompenseret. Ydermere er der med en glemselsfaktor opnået, at forstærkningen til parameterestimerne ikke er vedvarende aftagende, men

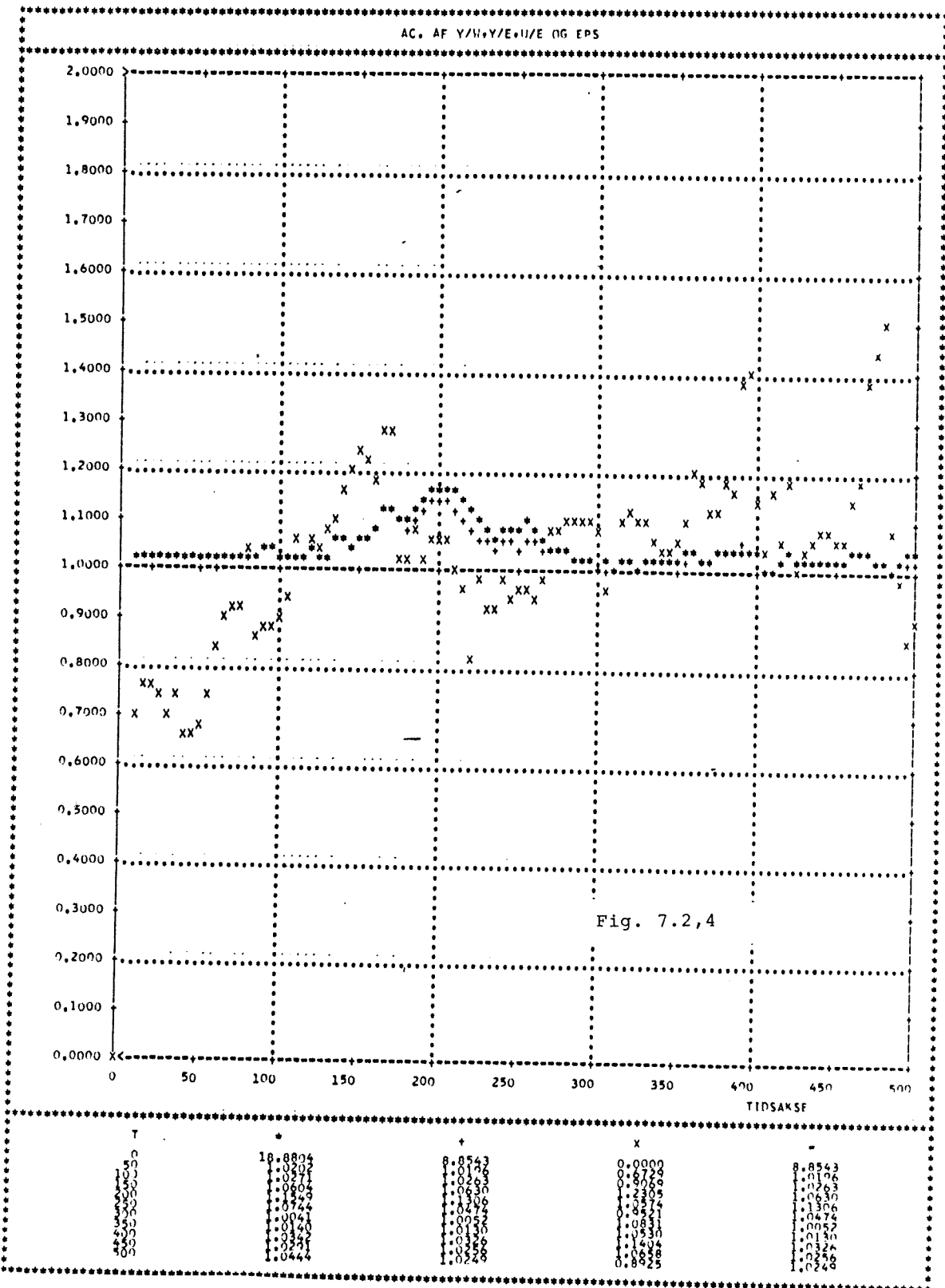
antager en stationær værdi. Desværre er dette ikke kun tilfældet for estimatet af belastningen, men for forstærkningen til samtlige estimater. Glemselsfaktoren skal vælges, sådan at fluktuationerne i parameterestimerne ikke forringer reguleringskvaliteten for konstant belastning. Dette kan være et problem, hvis den selvindstillende regulator er ret følsom overfor parameterfejl. Ved høje støjniveauer må glemselsfaktoren vælges i overensstemmelse med dette, og den dermed følgende forøgelse af estimationshorisonten bevirker en langsommere absorption af markante ændringer i belastningen, $d(t)$. Samtidig med støjniveauet øges algoritmens evne til at optage mindre variationer, når disse er mindre end spredningen på støjsignalet $e_o(t)$. På grund af faren for Windup i kovariansmatricen, $P(t)$, for parameterestimerne, kan glemselsfaktoren vælges vilkårligt lavt ved støjssvage tilfælde, med mindre der tages specielle forholdsregler. Referencesignalet er selv i servotilfældet konstant en del af tiden og kan derfor ikke forventes alene at kunne eksitere objektet i en sådan grad, at windup undgås.

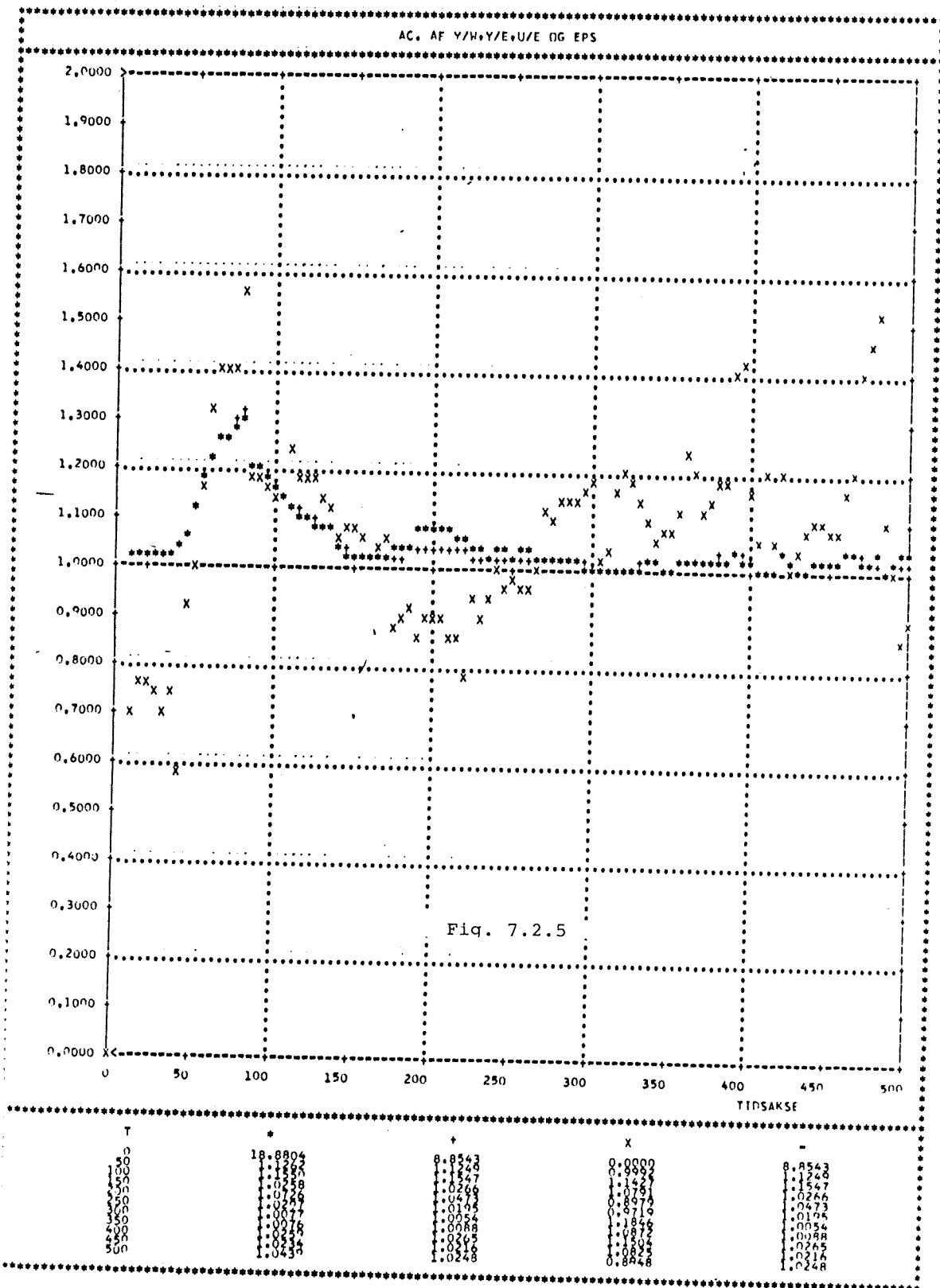
---ooo0ooo---

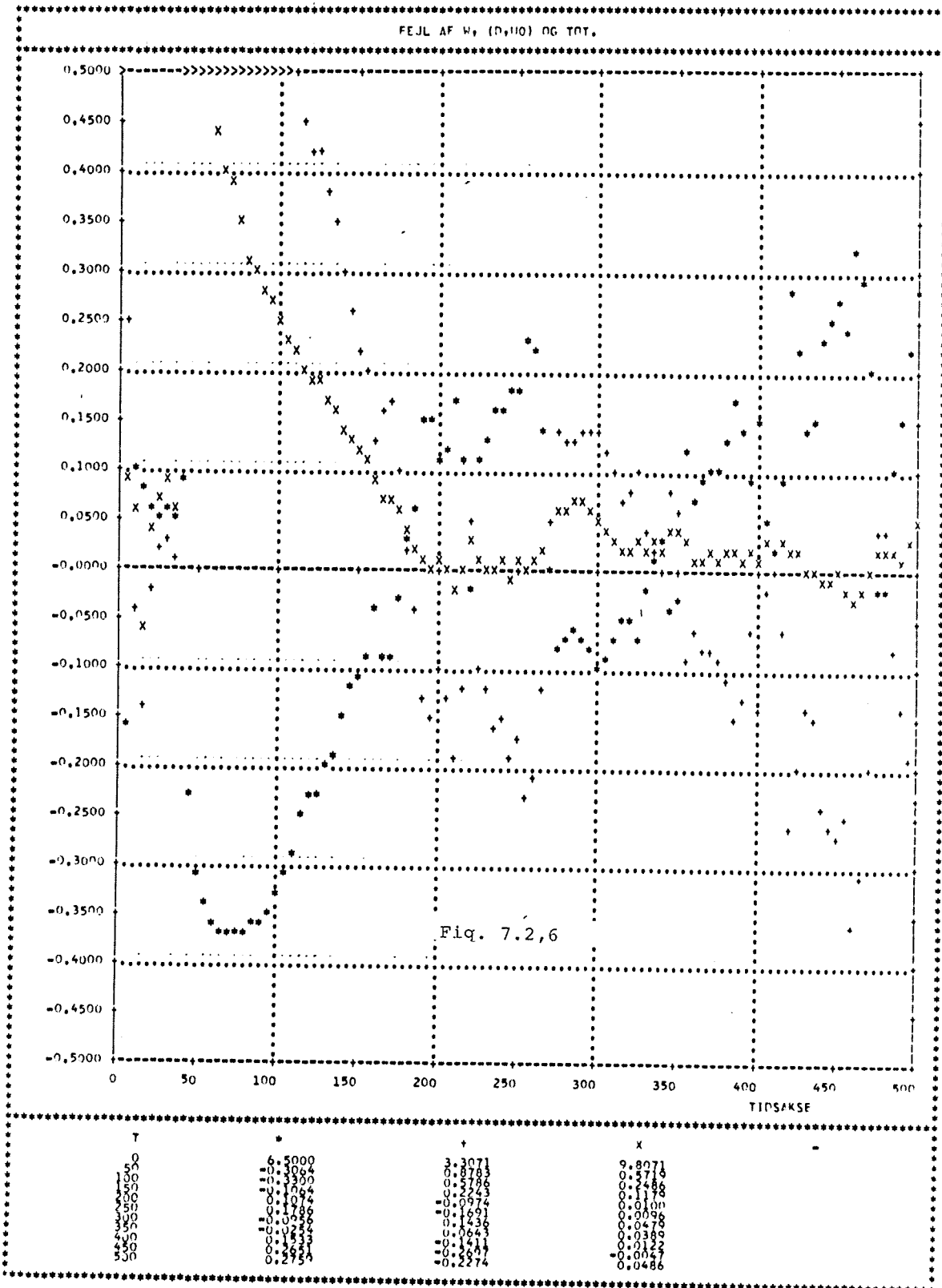


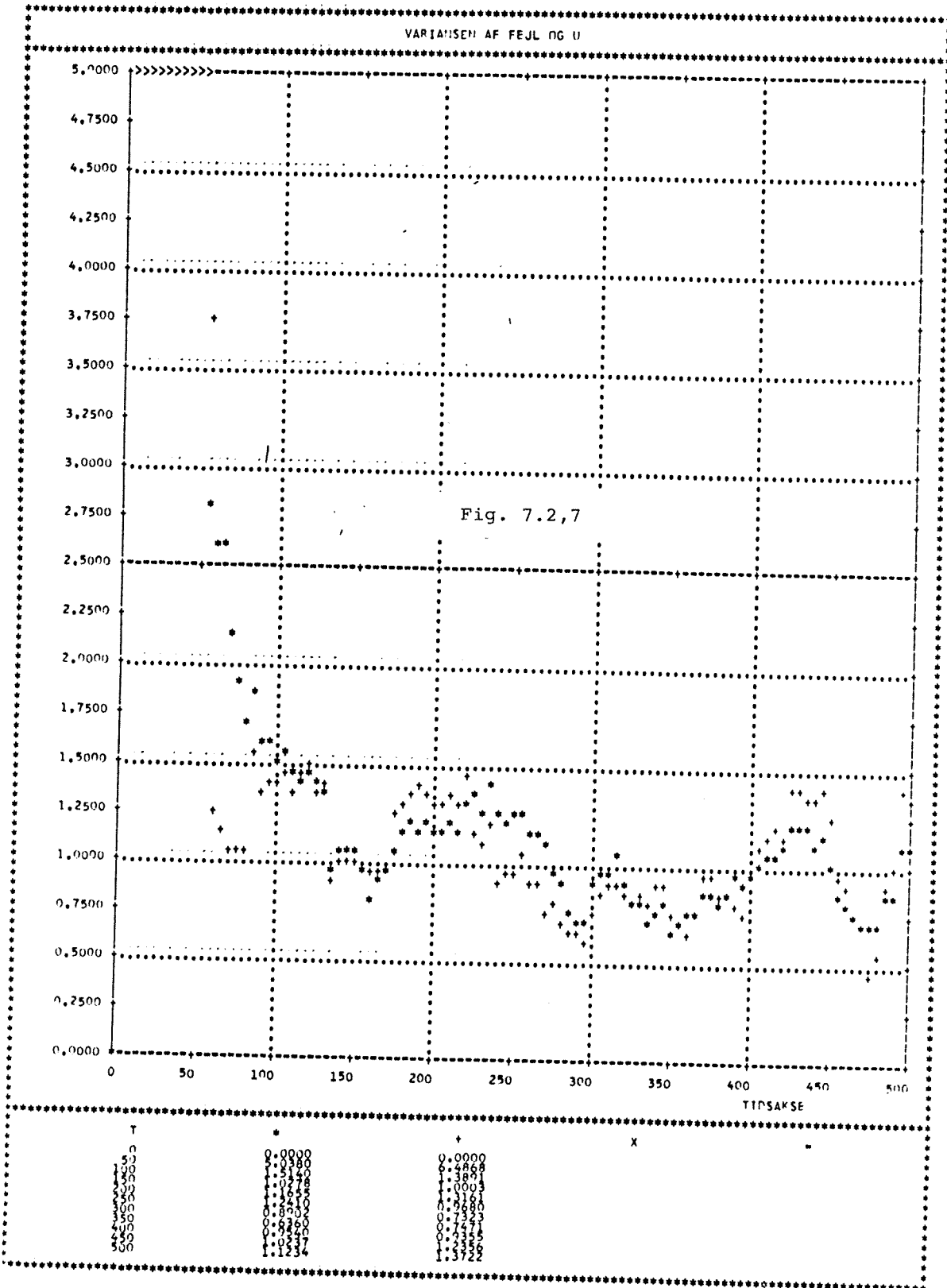


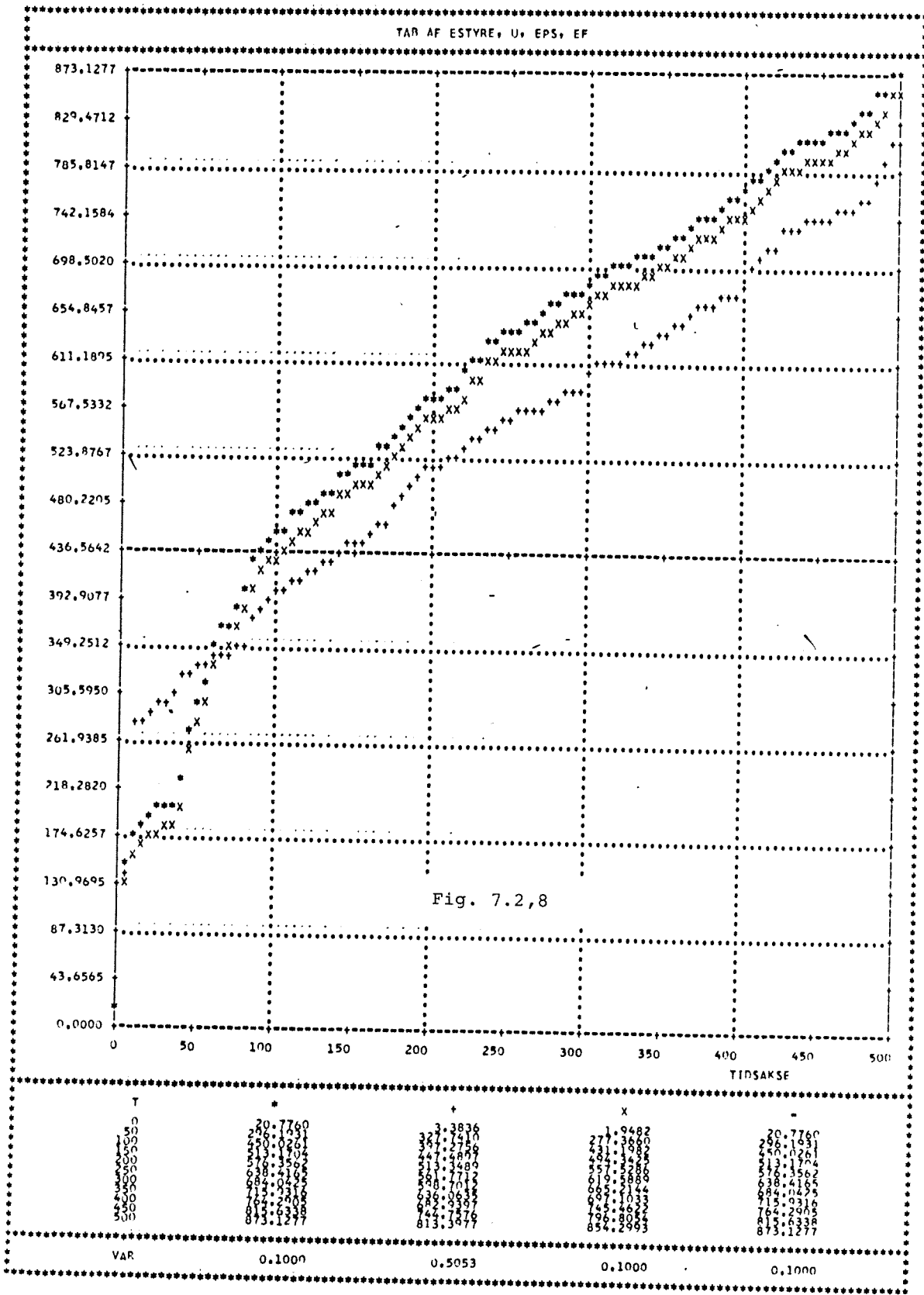


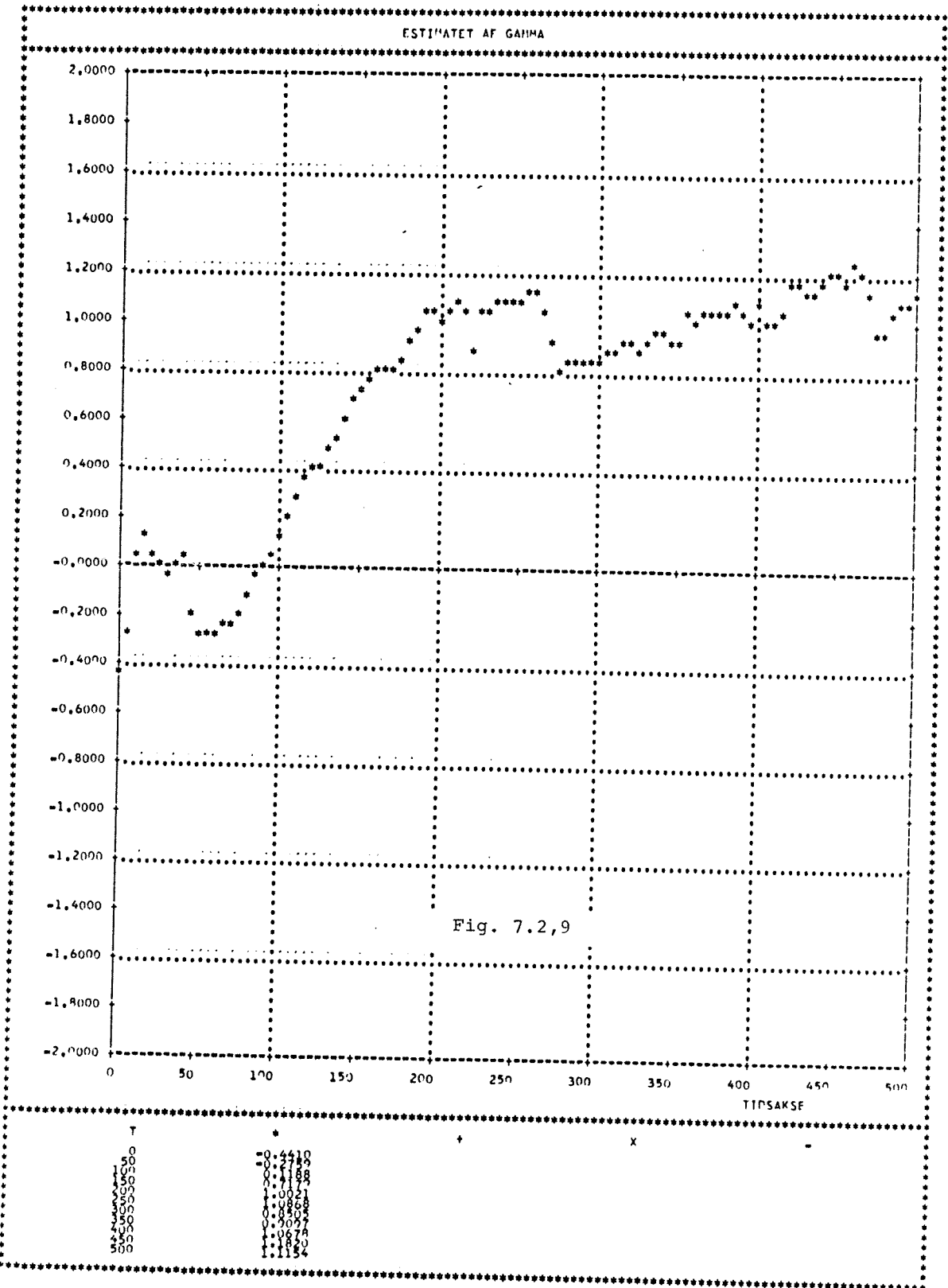


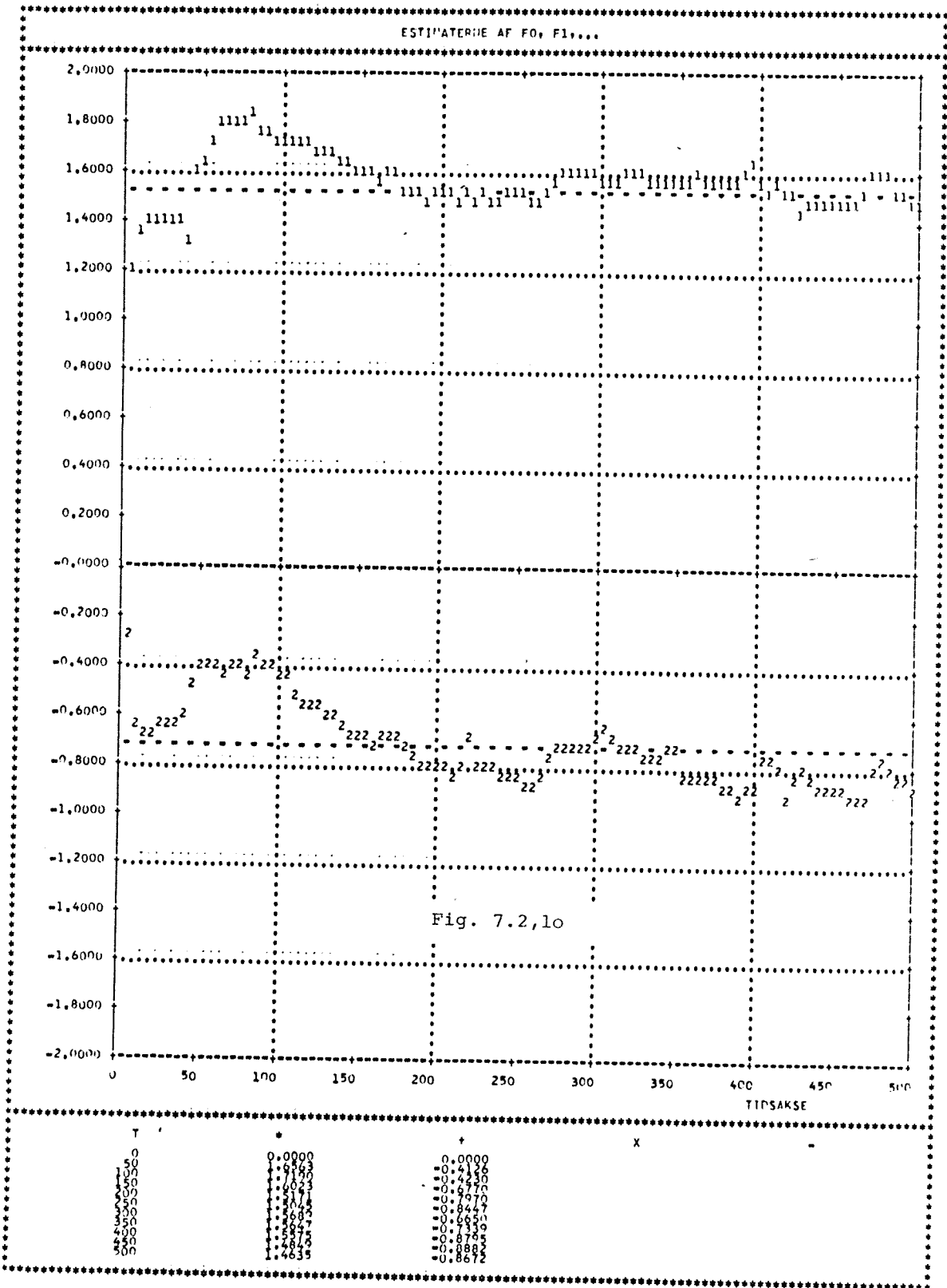


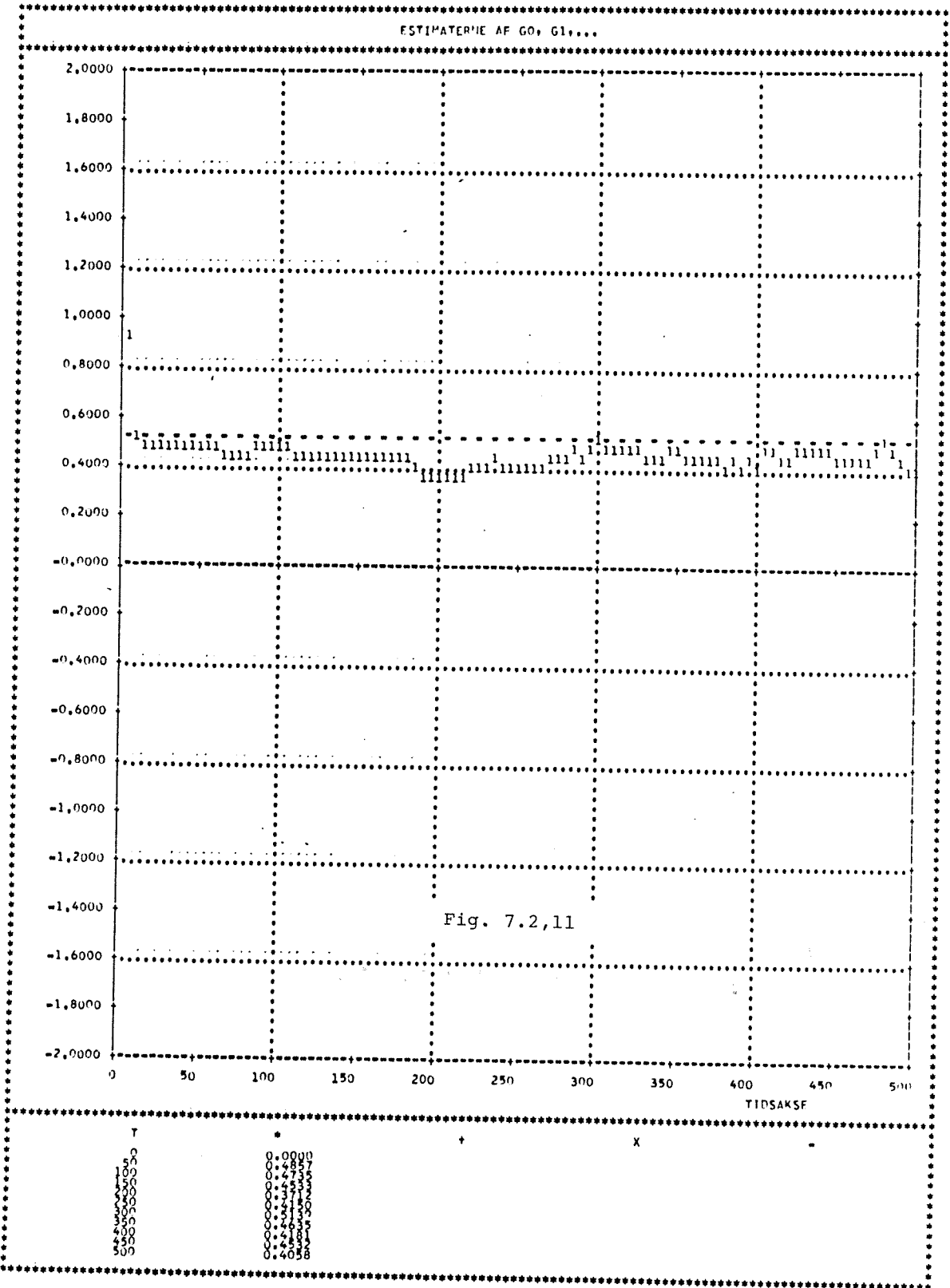


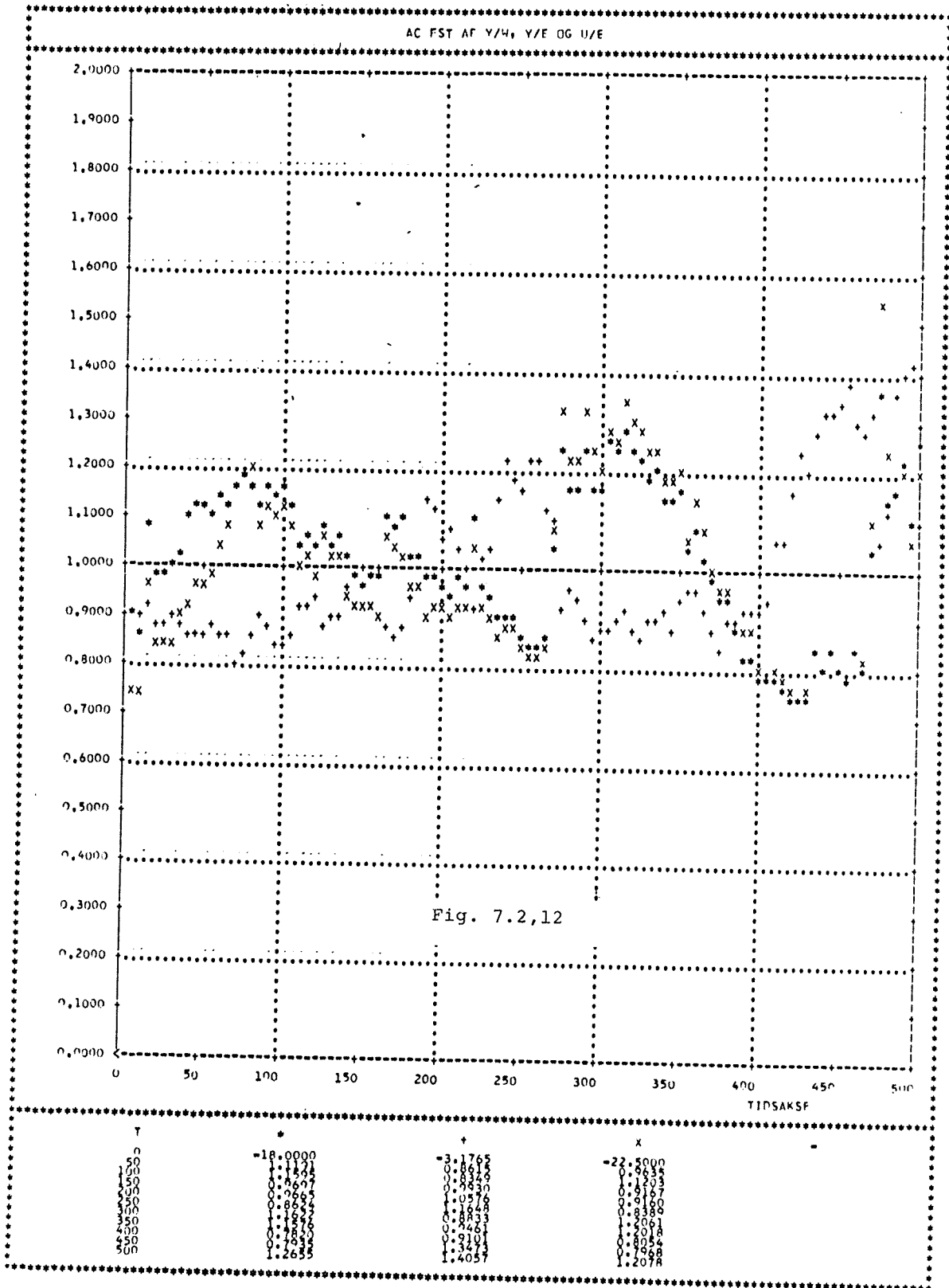


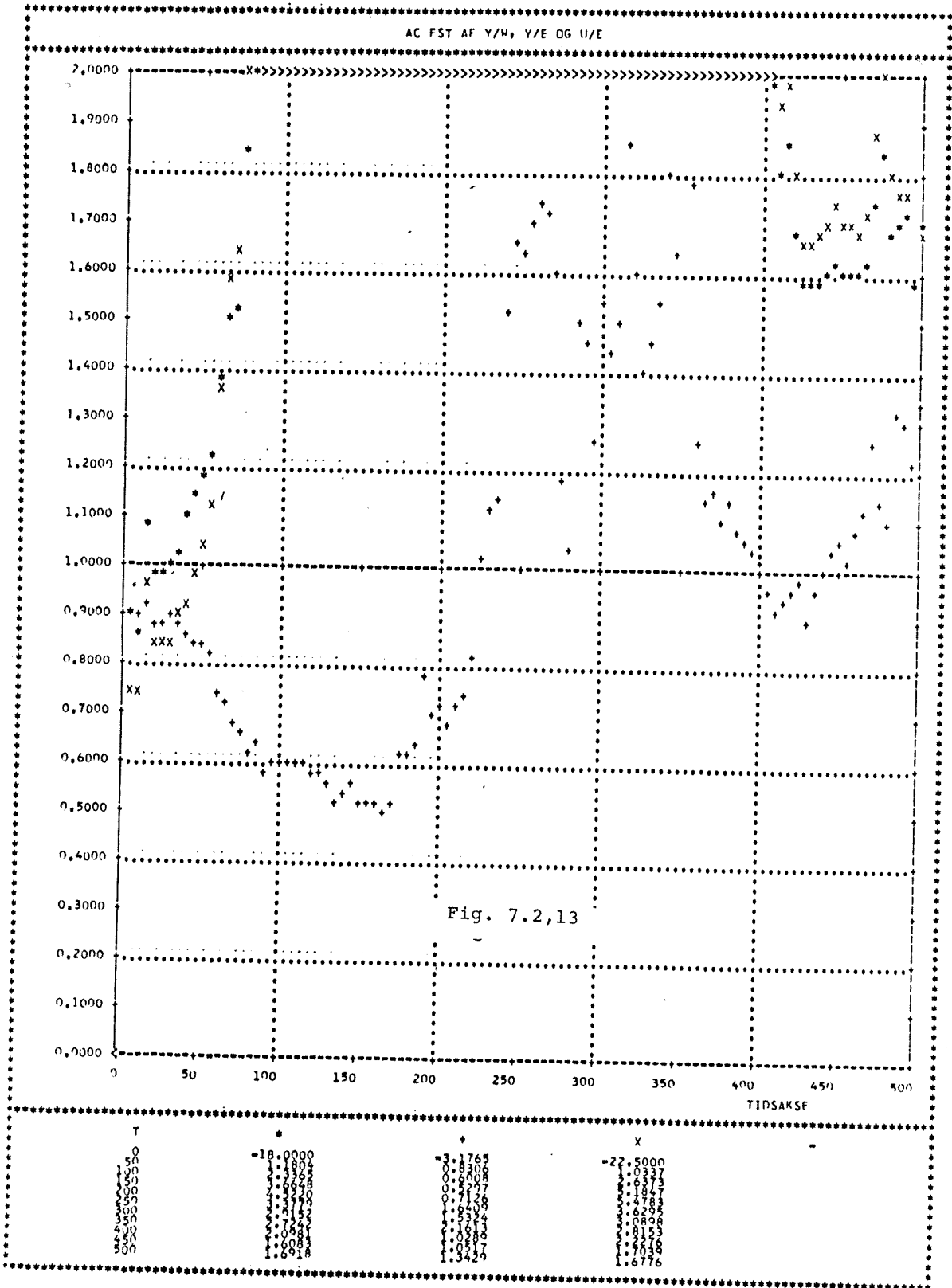


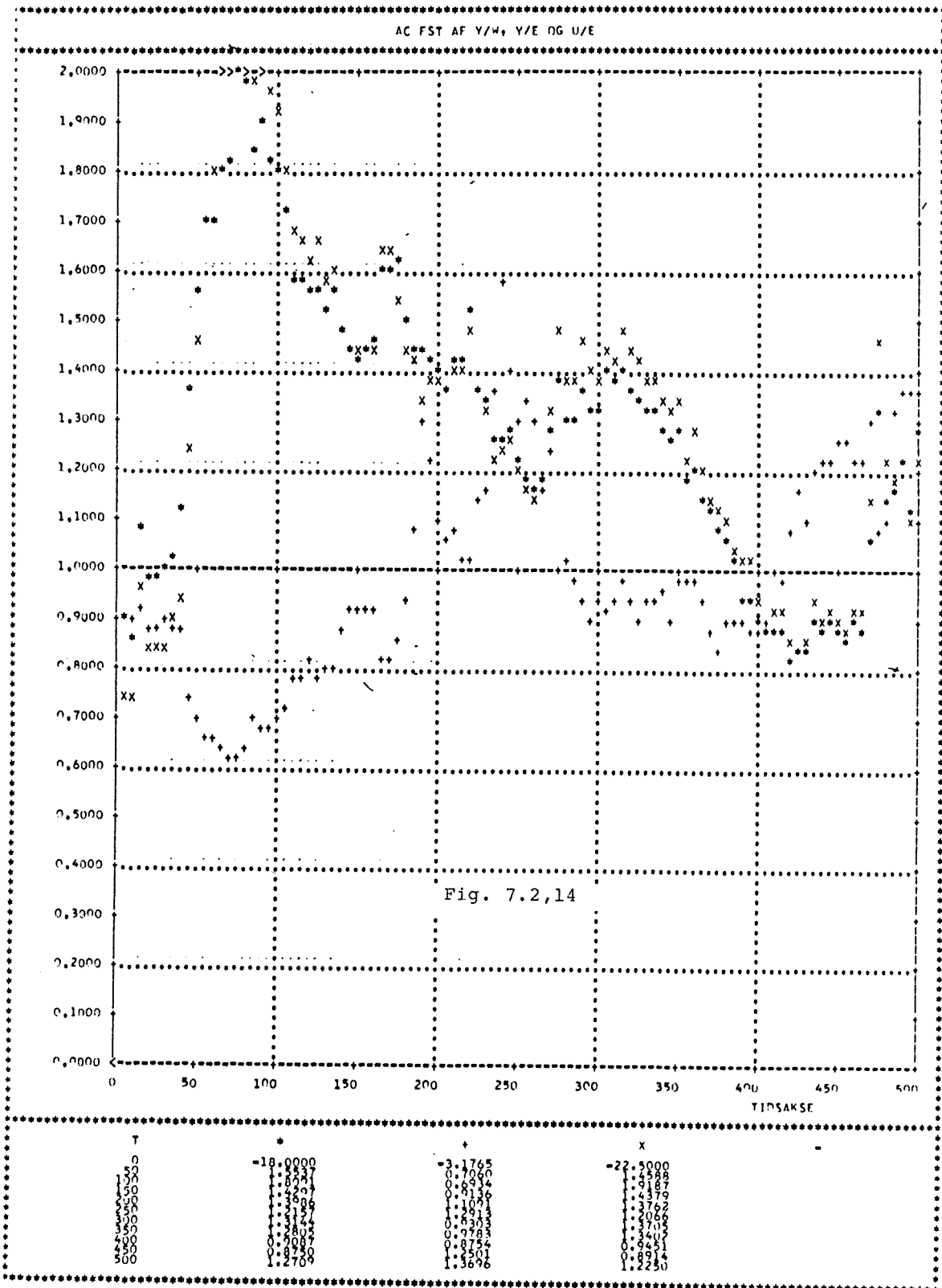












7.3 Kalmanfiltrering.

I de foregående afsnit er der undersøgt, hvorledes de adaptive regulatorer reagerer overfor forstyrrelser de ikke er dimensioneret til. Selve standard algoritmerne er udledt under forudsætning af, at parameterne heriblandt belastningen, d , er konstante. Der er fundet, at ved anvendelse af en eksponentiel vægtning kunne regulatorernes evne til at opfange ændringer i belastningen forbedres. Når der i estimationsalgoritmen bruges en glemselsfaktor fremkommer en metode, der ofte anvendes i forbindelse med langsomt varierende parameter. I dette afsnit vil opfattelsen af $d(t)$ som en langsomt varierende parameter blive forladt og i stedet vil belastningen blive opfattet som en ikke styrbar tilstand på linie med de andre i objektet. I så fald kan $d(t)$ modelleres ved:

$$(1) \quad d(t) = \frac{1}{1 - q^{-1}} e_d(t)$$

hvor $e_d(t)$ er hvid støj med variansen Q_d og middelværdi nul. Eftersom (1) er en stationær proces er det ikke helt simpelt at vælge Q_d , sådan at (1) er en god model for den ukendte ikke stationære proces (7,6) der foreligger.

En metode til at følge $d(t)$ er at kombinere en observer med estimationsalgoritmen, der opdaterer de øvrige parameter, som både observer og regulator anvender. Ud fra omformningen af systemligningen

$$(2) \quad A(q^{-1})y(t) = q^{-k} B(q^{-1})u(t) + C(q^{-1})e(t)$$

til tilstandsligningerne i den kanoniske observerbare form,

kan objektbeskrivelsen

$$(3) \quad A(q^{-1})y(t) = q^{-k} B(q^{-1})u(t) + C(q^{-1})e(t) + d(t)$$

omskrives til tilstandsligningerne

$$(4) \quad x(t+1) = F x(t) + G u(t+1-k) + \Gamma e(t) + \Gamma_d e_d(t)$$

$$(5) \quad y(t) = H^T x + e(t)$$

hvor

$$x(t) = \{ x_1, \dots, x_n, d(t) \}^T \quad (n = n_A = n_B + 1)$$

$$F = \begin{bmatrix} -a_1 & 1 & 0 & \dots & 0 & -a_1 \\ -a_2 & 0 & 1 & \dots & 0 & -a_2 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ -a_{n-1} & 0 & 0 & \dots & 1 & -a_{n-1} \\ -a_n & 0 & 0 & \dots & 0 & -a_n \\ 0 & 0 & 0 & \dots & 0 & 1 \end{bmatrix}$$

$$G = (b_0, \dots, b_{n-1}, 0)^T$$

$$H = (1, 0, \dots, 0, 1)^T$$

$$= (c_1^{-a_1}, c_2^{-a_2}, \dots, c_n^{-a_n}, 0)^T$$

$$= (0, \dots, 0, 1)^T$$

Denne repræsentation er vist i fig. 1. Antages det, at objektet

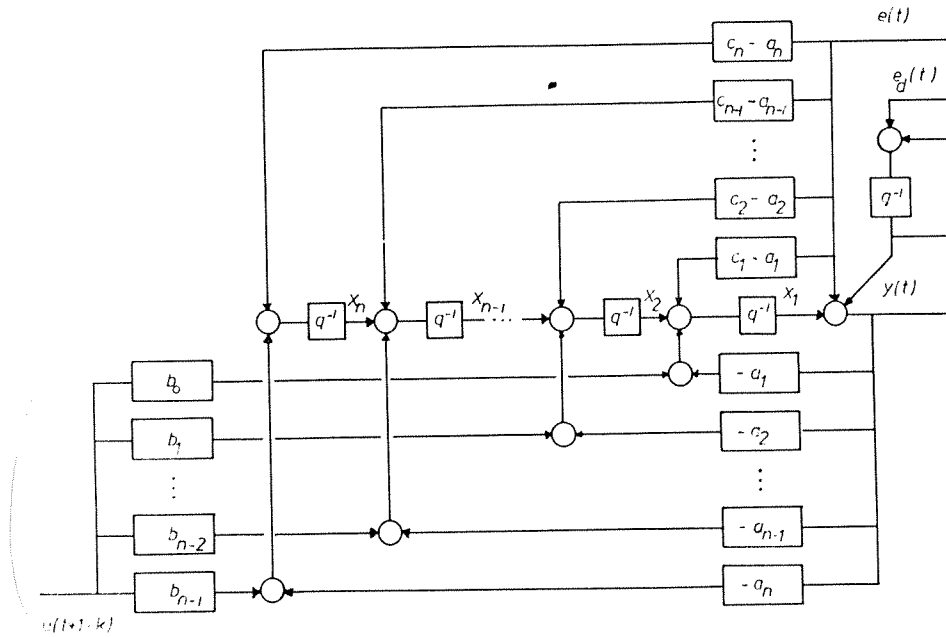


Fig. 1: Tilstandsrepræsentationen for objektet.

er kendt, kan et kalmanfilter til estimation af tilstandene x_1, x_2, \dots, x_n , $d(t)$ dimensioneres j.v.f. /21/ ud fra

$$(7) \quad \hat{x}(t+1) = F \hat{x}(t) + G u(t+1-k) + K(t)\vartheta(t)$$

hvor residualet, $\vartheta(t)$, og forstærkningen, $K(t)$, er givet ved

$$(8) \quad \vartheta(t) = y(t) - H^T \hat{x}(t)$$

og

$$(9) \quad K(t) = [FP(t)H + R_{12}] [H^T P(t)H + R_2]^{-1}$$

$$P(t+1) = E\{ (x-x)(x-x)^T \}$$

$$= FP(t)F^T + R_1 - K(t)[H^T P(t)H + R_2]K^T(t)$$

I dette specielle tilfælde er

$$(10) \quad R_1 = \Gamma \Gamma^T \sigma^2 + \Gamma_d \Gamma_d^T Q_d, \quad R_2 = \sigma^2$$

$$R_{12} = \Gamma \sigma^2$$

hvor σ er spredningen på $e(t)$. Den stationære forstærkning kan skrives på formen:

$$(11) \quad K_0 = \begin{bmatrix} \beta_1 + \alpha c_1 - a_1 \\ \beta_2 + \alpha \bar{c}_2 - a_2 \\ \vdots \\ \beta_{n-1} + \alpha c_{n-1} - a_{n-1} \\ \alpha c_n - a_n \\ \beta_n \end{bmatrix}$$

hvor

$$\alpha = \frac{\sigma^2}{N}$$

$$\beta_{i-1} = \frac{P_{1,i} + P_{i,d}}{N} \quad i = 2, \dots, n+1$$

$$N = \sigma^2 + P_{1,1} + 2P_{1,d} + P_{d,d}$$

Her er f.eks. $P_{1,d}$ et element i matricen $P(t)$ fra (9). I det adaptive tilfælde, hvor parameterestimerne opdateres efter hver måling, vil en stationær løsning til (9) være

ret tidskrævende og usikker i startperioden. Det vil da være ønskeligt, at der eksisterer en approximation til (11), så - at der med en observer (fig 2) kan opnås et rimeligt estimat af $d(t)$.

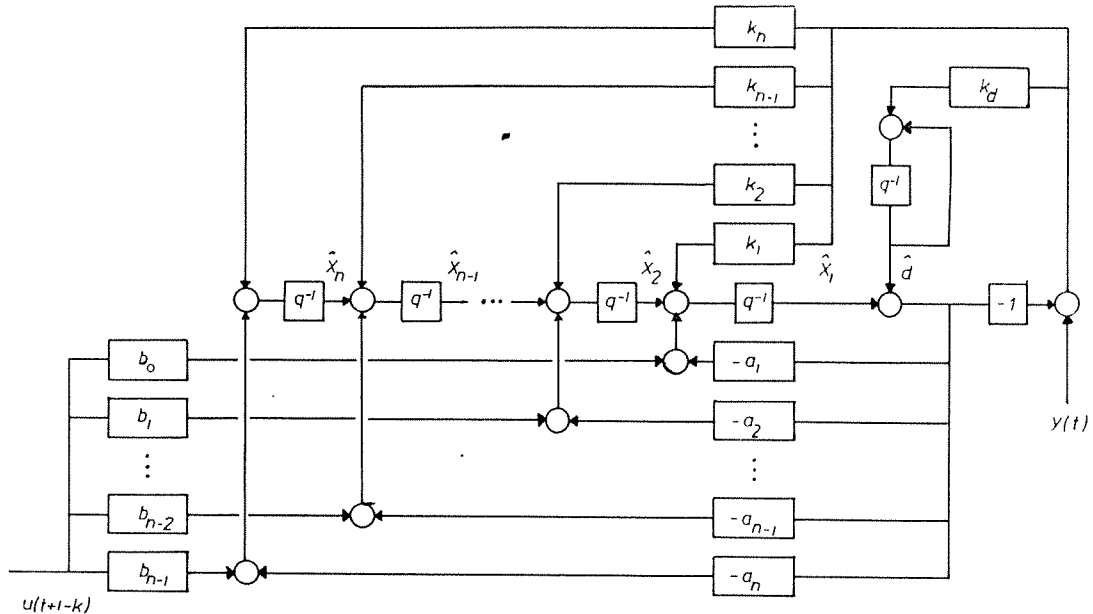


Fig. 2: Struktur i observeren.

En observer kan f. eks. dimensioneres ved at bestemme for - stærkningen, sådan at egenverdierne til matricen

$$(12) \quad F - K_O H^T$$

har nogle fastsatte værdier. I det generelle tilfælde vil dette i forbindelse med adaptiv regulering dog også kræve et tidskrævende regnearbejde. For at argumentere for en anden fremgangsmåde betragtes to ydertilfælde.

Lad nu $\sigma^2 = 0$, hvor de stationære løsninger til $P(t)$ og

K(t) er:

$$P_{\infty} = \Gamma_d \Gamma_d^T \sigma_d^2$$

(13)

$$K_{\infty} = (-a_1, -a_2, \dots, -a_n, 1)^T$$

Dette svarer til at placere egenverdierne for observeren i origo. Hvadenten estimatet af $d(t)$ anvendes ved estimationen af de øvrige parameter eller til at bortkompensere $d(t)$ udenom estimationsalgoritmen, vil de øvrige estimater blive fejlagtige. Dette skyldes, at når de øvrige parameter er korrekte er

$$(14) \quad \hat{d}(t) = q^{-1} (C(q^{-1})e(t) + d(t))$$

og dermed er residualerne

$$(15) \quad \varepsilon(t) = e(t) + C(q^{-1})e(t-1)$$

for estimationsalgoritmen korrelerede. Såfremt estimationsalgoritmen tilføres $\hat{d}(t+1)$, der er kendt j.v.f. fig. 2, vil denne ikke opdatere parameterestimerne, fordi observeren anvender hele innovationen

$$\eta(t) = y(t) - y^*(t|t-1)$$

til at korrigere $\hat{d}(t+1)$. Ved at betragte fig. 2 kan det let ses, at innovationen for observeren er givet ved:

$$\eta(t) = \hat{A}(q^{-1})y(t) - q^{-k} \hat{B}(q^{-1})u(t) - \hat{d}(t)$$

når forstærkningsvektoren er givet ved (13). Hvis der bruges en RLS - algoritme til at estimere de øvrige parameter og $C = 1$ er modellen der anvendes her givet ved:

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

hvor $\hat{d}(t+1)$ er dannet i observeren. Da nu

$$\hat{d}(t+1) = \hat{d}(t) + \eta(t)$$

ses det meget let at

$$\varepsilon(t) = 0$$

Det andet ydertilfælde fremkommer når $Q_d = 0$. Her er den stationære løsning til (9):

$$(16) \quad K_0 = (c_1 - a_1, \dots, c_n - a_n, 0)^T$$

hvilket svarer til en placering af egenværdierne i 1 og i nulpunkterne til C^* (sml. afsnit 5.3)

En approximation kan være at bestemme forstærkningsvektoren som en lineær interpolation mellem de to yderpunkter.

$$(17) \quad K = \begin{bmatrix} \alpha c_1 - a_1 \\ \vdots \\ \alpha c_{n-1} - a_{n-1} \\ 1 - \alpha \end{bmatrix} \quad 0 \leq \alpha \leq 1$$

Denne kan kun umiddelbart anvendes sammen med eksplicitte regulator algoritmer, fordi observeren er formuleret i objekt - parameterne. Problemet er nu at vælge α , sådan at der opnås et rimeligt kompromis mellem fejlagtige estimater, der skyldes den ucentrale estimator, og en evne til at optage variationer i $d(t)$ tilstrækkeligt hurtigt. Dette valg er langt fra trivielt og kræver i princippet information om støjvariansen og den

værdi af Q_d med hvilken (1) beskriver den faktiske proces bedst mulig. Det omtalte dilemma, hvor ønsket om tilfredstillende estimation og en god evne til at følge spring i $d(t)$ er et grundlæggende problem for de metoder, der ikke behandler spring i forstyrrelsernes middelværdi som et specialtilfælde, der relativt sjældent indtræffer. I senere afsnit vil der blive angivet metoder, der er baseret på en separation af de to krav, således at der opnås høj estimationspræcision i det normale tilfælde og høj sporingshastighed i specialtilfældene.

I dette projekt er det blandt de afprøvede regulatoralgoritmer kun deadbeatregulatorerne, der er eksplicitte. På grund af DBO - regulatorens ret høje følsomhed overfor estimationsfejl, når den anvendes med testobjektet, var det ikke muligt at finde en værdi for α , der gav et passende kompromis mellem estimationsfejl og sporingshastighed. Dette kan bl.a. skyldes, at den lineære approximation (17) er for dårlig. Denne metode er ydermere behæftet med en række ulemper, hvilket er omtalt nedenfor, og er derfor ikke behandlet yderligere.

Metoden har den begrænsning at kun kunne anvendes i forbindelse med eksplicitte algoritmer. I det følgende vil der blive undersøgt en metode, der kan anvendes for begge typer. I stedet for at estimere $d(t)$ sammen med en række tilstande, der ikke bliver brugt ved de her undersøgte regulatorer, kan denne ved hjælp af et udvidet kalmanfilter estimeres sammen med systemparameterne. Ved denne metode undgås også det ekstra filter, som observeren udgør. Denne metode har ydermere den fordel at være en meget lille udvidelse af standard estimationsalgoritmen.

Lad et ulineært og tidsvarierende system med de ukendte

parameter og tilstande samlet i vektoren θ være givet ved:

$$\theta(t+1) = f(\theta(t), \omega(t)) + v(t)$$

(19)

$$y(t) = h(\theta(t)) + e(t)$$

hvor $\omega(t)$ er en kendt påvirkning. Det udvidede kalmanfilter er da givet ved:

$$\hat{\theta}(t+1) = f(\hat{\theta}(t), \omega(t)) + K(t)\varepsilon(t)$$

(19)

med:

$$\varepsilon(t) = y(t) - h(\hat{\theta}(t))$$

$$K(t) = [FP(t)H + R_{12}] [H^T P(t)H + R_2]^{-1}$$

(20)

$$P(t+1) = FP(t)F^T - K(t)[H^T P(t)H + R_2]K^T(t)$$

Her indgår størrelserne:

$$H = H(\hat{\theta}(t)) = \nabla_{\theta} h(\theta) \Big|_{\hat{\theta}=\theta}$$

(21)

$$F = F(\hat{\theta}(t), \omega(t)) = \nabla_{\theta} f(\theta, \omega) \Big|_{\hat{\theta}=\theta}$$

Lad i det følgende θ betegne en af de tre parametervektore fra kapitel 3. Da $d(t)$ nu opfattes som en tidsvariabel tilstand opspaltes θ i θ^{\sim} og $d(t)$ således at:

$$\theta = (\theta^{\sim T}, d(t))^T$$

(22)

hvorved objektet kan modelleres som givet ved (23).

$$\theta^{\sim}(t+1) = \theta^{\sim}(t)$$

$$(23) \quad d(t+1) = d(t) + e_d(t)$$

$$y(t) = x^T(t)\theta(t) + e(t)$$

Betydningen af vektoren $x(t)$ er uændret i forhold til kapitel 3. Da nu:

$$(24) \quad R_1 = \Gamma_d \Gamma_d^T \sigma_d^2 \quad R_2 = \sigma^2 \quad R_{12} = 0$$

$$F = I$$

er det udvidede kalmanfilter (UKF) for systemet (23) givet ved:

$$\hat{\theta}(t+1) = \hat{\theta}(t) + K(t)\varepsilon(t)$$

$$\varepsilon(t) = y(t) - x^T(t)\hat{\theta}(t)$$

$$(25) \quad K(t) = P(t)H [H^T P(t)H + \sigma^2]^{-1}$$

$$P(t+1) = P(t) - K(t) [H^T P(t)H + \sigma^2]^{-1} K^T(t) + R_1$$

Som det fremgår af (25) kræver algoritmen et kendskab til støjsspredningen og Q_d . Det er dog muligt at estimere σ^2 . Der kan j.v.f. /11/ opnås en simplifikation ved at sætte $\sigma = 1$. Såfremt $C(q^{-1}) = 1$ giver (21) at

$$(26) \quad H = x(t)$$

hvorved algoritmen pånar bidraget R_1 er identisk med en RLS - algoritme. Betragtes residualerne i $x(t)$ som faste tal vil (26) også gælde for det generelle tilfælde $C(q^{-1}) \neq 1$.

Med denne betragtning er der ingen ulineariteter i objektbeskrivelsen (23). Inddrages det nu, at residualerne, der er elementer i $x(t)$, er afhængige af parameterestimerne, kan vektoren H udregnes til:

$$\begin{aligned} H &= \nabla_{\theta} (x^T(t) \theta) \Big|_{\theta=\hat{\theta}} = - \nabla_{\theta} \varepsilon(t) \Big|_{\theta=\hat{\theta}} \\ &= \frac{x(t)}{\hat{C}(q^{-1})} \end{aligned}$$

I dette tilfælde fremkommer RML - algoritmen, hvis der igen ses bort fra bidraget R_1 . Den bekrevne metode er en kendt udvidelse af standardalgoritmerne for at følge langsomt varierende paramter. Metoden er refereret i afsnit 3.4.

Ligesom i det foregående afsnit er der her opnået at forstærkningen i estimationen antager en stationær værdi. Herved bliver egenskaberne overfor ekstreme påvirkninger tidsuafhængige undtagen i startperioden. I modsætning til situationen ved brug af en glemselsfaktor er det her kun forstærkningen til estimatet af belastningen, der har en forstærkning væsentligt forskellig fra nul. Der vil derfor ikke forventes at finde udsving for estimatet af $d(t)$ til forkert side, når $d(t)$ springer. Denne løsning (UKF) har ligesom de i det foregående omtalte kalmanfiltre den ulempe, at såfremt værdien af Q_d er for stor, vil de øvrige estimater være fejlagtige på grund af en dermed indført korrelation i støjen. Når belastningen er konstant på nær i nogle meget få spring vil (1), der forudsætter stationaritet, være en temmelig ringe model for springprocessen. Følgelig vil det være problematisk at vælge Q_d sådan, at der som før nævnt i forbindelse med de lineære kalmanfiltre opnås et rimeligt kompromis. Såfremt springene optræder med en større hyppighed er modellen, (1), bedre og der kan da ud fra kendt hyppighed og fordeling af springen gives en værdi af Q_d , der giver antagelige resultater. Et

sådant kendskab til processen kan ikke forudsættes ved adaptiv regulering. Ved lave støjniveauer kan der også være fare for Windup i kovariansmatricen for parameterestimererne, men her vil $P(t)$ dog kun stige lineært og ikke eksponentiel som ved brug af en glemselsfaktor.

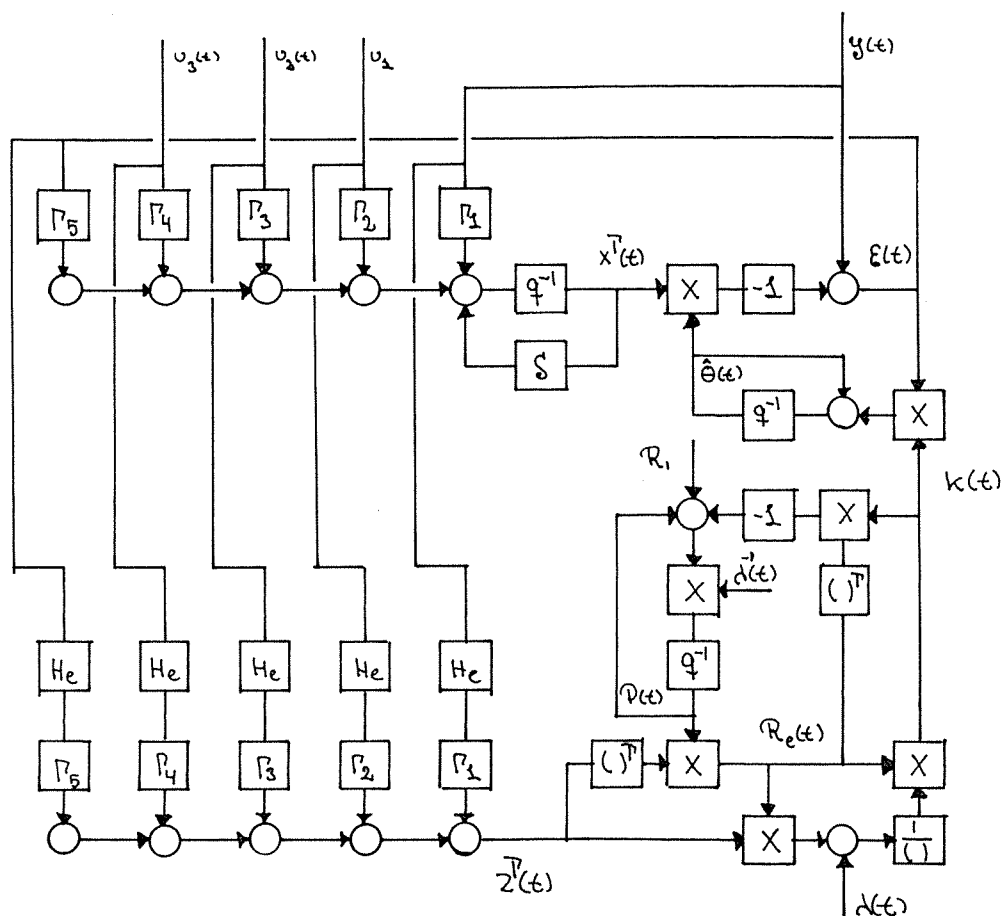


Fig. 3: Struktur i udvidet kalmanfilter.

Spring.

Som det fremgår af de foregående overvejelser egner denne løsning sig ikke til at klare spring, der er langt større end spredningen på $e(t)$. Formålet med dette filter kan da være at absorbere mindre spring af samme størrelsesorden som støjspredningen og samtidig estimere de øvrige parameter med en sådan

sikkerhed, at reguleringen ikke er svækket bemærkelsesværdigt. Filteret er desuden velegnet til at opfange de mere langsomme variationer i belastningen så som drift, fordi modellen (1) bedre kan tilpasses disse forstyrrelser. Når $d(t)$ er konstant vil i princippet enhver værdi for Q_d give fejlagtige estimer, hvorfor denne må vælges meget mindre end den ukendte varians på $e(t)$. Det kunne tænkes at fastsætte Q_d ud fra et estimat af σ f. eks. $Q_d = 0.01\sigma^2$. Denne tanke er dog ikke blevet fulgt op af eksperimenter. I det følgende er $Q_d = 0.005$, der er tilstrækkeligt lavt til ikke at give fejlagtige parameterestimer. På grund af den stationære forstærkning til estimatet af belastningen er DC - fejlen mere fluktuerende, hvilket fremgår af fig. 4, der sammen fig. 5 er optaget for en simulation med testobjektet reguleret af en MVo -regulator. Reference - signalet er stadig en trinfunktion. Sammenholdes fig. 4 og med fig. 6.1.1, 10 og 11 ses en lille forbedring i AC - forstærkningens konvergens og den før omtalte variation i DC - fejlen.

Kurverne i fig 6 - 12 viser udvalgte tidsforløb i simulationen, når belastningen, $d(t)$, springer til $t = 40$ fra 0 til 1. Dette spring er lidt større end algoritmen kan forventes at klare med godt resultat. Sammenligninger med standard algoritmen og med brug af glemselsfaktor viser betydelig forbedringer.

	Tab (y)	Tab (u)	
Standard	1046.7	809.2	fig. 7.1,8
med $\lambda = 0.985$	873.1	813.4	fig. 7.2,8
UKF ($Q_d = 0.005$)	774.3	774.4	fig. 7.3,6

Man kan på tabskurverne se, at springets indflydelse er forsvundet efter ca. 30 trin. Denne forbedring afspejles også i

kurven over DC - fejlen (fig. 9, sml. 7.2,6) og AC - forstærkningen (fig. 8, sml. 7.2,5). Specielt AC - forstærkningens forløb er forbedret væsentligt. Parameterestimaternes forløb er vist i fig. 10 - 12. Sammenholdes disse med fig. 7.2,9 - 11 bemærkes dels, at estimerne af koefficienterne i regulatorens F - polynomie er ret upåvirket af springet, dels at parameterestimerne er langt mere rolige ($\hat{\gamma}$ dog undtaget) som en følge af, at den stationære forstærkning kun gælder for estimatet af belastningen. Det bemærkes endvidere, at $\hat{\gamma}$ har korrekt retning efter springet (de negative estimer af belastningen er før $t = 40$).

Som det fremgår af fig. 13 (sml. 6.1.3,9) påvirkes DBo - regulatoren, der er følsom overfor parameterfejl, af den bias som $Q_d = 0.005$ afstedkommer. Denne kurve (fig. 13) er optaget fra en simulation, hvor objektet kun påvirkes af referencesignalet og den hvide støj, $e_o(t)$. ($\sigma^2 = 0.1$) I fig. 14 kan de samme kurver ses, når objektet også påvirkes af et mindre spring på 0.2 for $t = 40$. Også for denne regulatortype er der opnået en forbedring i forhold til glemsels - faktorløsningen (sml. fig. 7.2,14), men sammenlignet med MVo -regulatoren er forholdene temmelig dårlige konvergens mæssigt. Ser man på tabene (fig. 15), bemærker man, at fejlen i $H_{y,e}$ afstedkommer lavere tab for udgangen. Dette skyldes både springets indvirkning på $H_{y,e}$ og det i afsnit 6.1.3 beskrevne forhold ved selvindstillende deadbeatregulering, at AC - forstærkningen for $H_{y,e}$ er lavere end ventet. I forhold til ventet er hældningen af tabskurven 15% lavere for udgangssignalet, mens styresignalets målte varians kun er 3% mindre. Billedet ændres dog lidt, når tabene sammenlignes med fig. 6.1.3,5, som er optaget for en simulation med objekt I kun påvirket af den hvide støj, $e_o(t)$, og referencesignalet. ($w(t) = 1, Q_d = 0$). Det er bemærkelsesværdigt, at disse tab kun er meget lidt større end, når $d(t)$ er konstant, og at forløbet af tabene ikke er markeret af springet.

Drift.

For at undersøge det udvidede kalmanfilters påståede evne til at spore langsomme variationer i belastningen, er der foretaget en række simulationer, hvor objektet udover de stokastiske forstyrrelser, $e_o(t)$, påvirkes af en rampefunktion i $d(t)$. I fig. 16 - 18 er nogle af tidsforløbene afbildet ved regulering med en MVo - regulator. Tabene (fig. 16) er kun lidt større end, når $d(t)$ er konstant og $Q_d = 0$. (sml. 6.1.1,6). Metodens egnethed fremgår tydeligt af fig. 17 og 18, hvor tidsforløbet af AC - forstærkningen og DC - fejlen er afbildet. (sml. 7.1,17 - 18). For en tilsvarende simulation med DBo - regulatoren er der optaget en kurve for AC - forstærkningerne, der er afbildet i fig. 19. Også her er der en væsentlig forbedring at spore i forhold til det foregående.

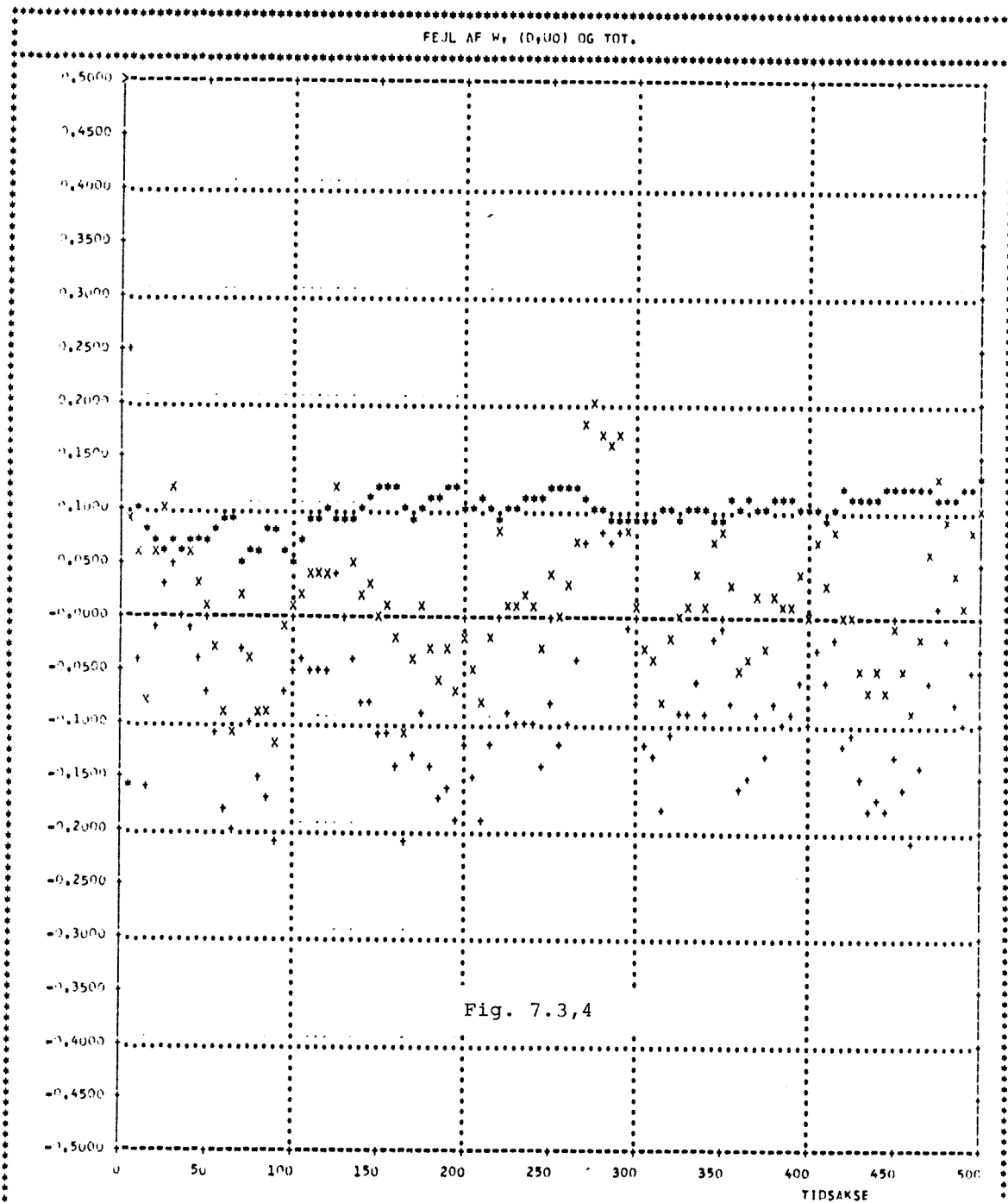
Spidser.

I det ovenstående er nogle af metodens fordele illustreret. Metoden har imidlertid den ulempe, at filteret også forsøger at spore en spids. På grund af den store forstærkning til estimatet af belastningen, vil en spids få dette til at springe ret voldsomt. Den dertil svarende fejl er på den anden side hurtigere udkompenseret igen. Dette kan ses ved sammeligning af fig. 20 og fig. 7.1,2, der er afbildninger af tidsforløbet af DC - fejlen, når systemet bestående af objekt I med MVO - regulator udsættes for en spids på 30.

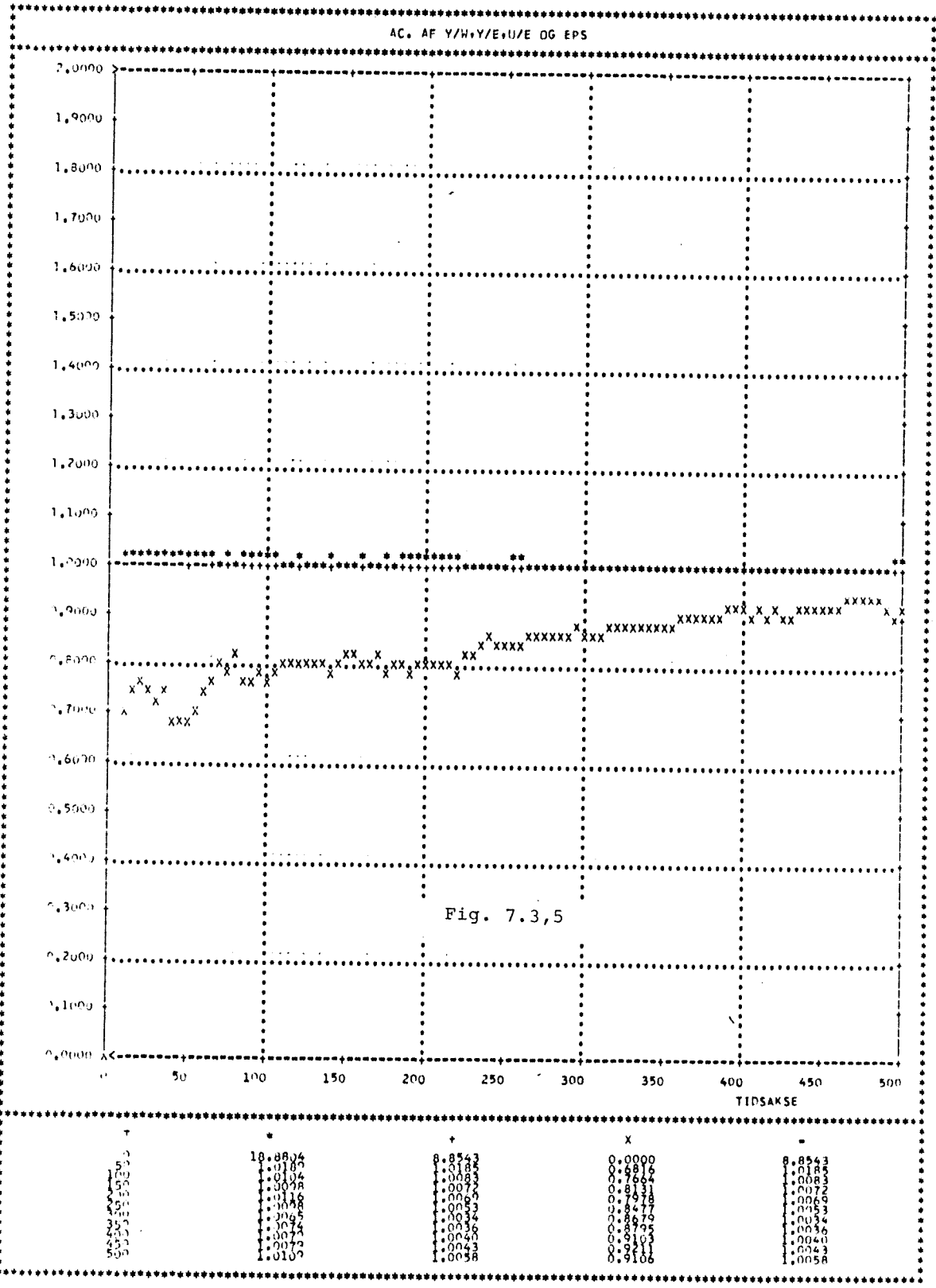
Dilemmaet, der bevirker, at der ikke samtidig kan opnås stor sporingshatighed og bias - fri estimater, eksisterer dog kun, når forstyrrelsernes eksitation af objektet indgår i estimationen med en ikke neglisibel vægt relativt til referencesignalets (eller andet tilsat tilstrækkelig eksiterende

signal). I støjsvage tilfælde (eller deterministiske) kan Q_d fastsættes efter et skøn over størrelsen af springene. Kurverne for signalerne, DC - fejlen samt AC - forstærkningen fra en simulation af et sådant tilfælde ($\sigma^2 = 0.01$) er vist i fig. 21 - 23. Regulatoren er en MVo - type og Q_d er valgt til 10. Det ses, at enhedsspringet ($T_j = 40$) allerede efter et trin ($k = 1$) er bortreguleret og har ingen større indflydelse på estimerne.

---ooo0ooo---



T	*	+	x
0	6.5000	3.3271	9.8071
50	0.0734	-0.0669	0.0065
100	0.0559	-0.0469	0.0080
150	0.0178	-0.1141	0.0037
200	0.1044	-0.1149	-0.0204
250	0.1175	-0.0813	0.0367
300	0.0877	-0.0188	0.0121
350	0.0377	-0.0100	-0.0448
400	0.1037	-0.1041	-0.0003
450	0.1168	-0.1300	-0.0132
500	0.1316	-0.0276	0.1040



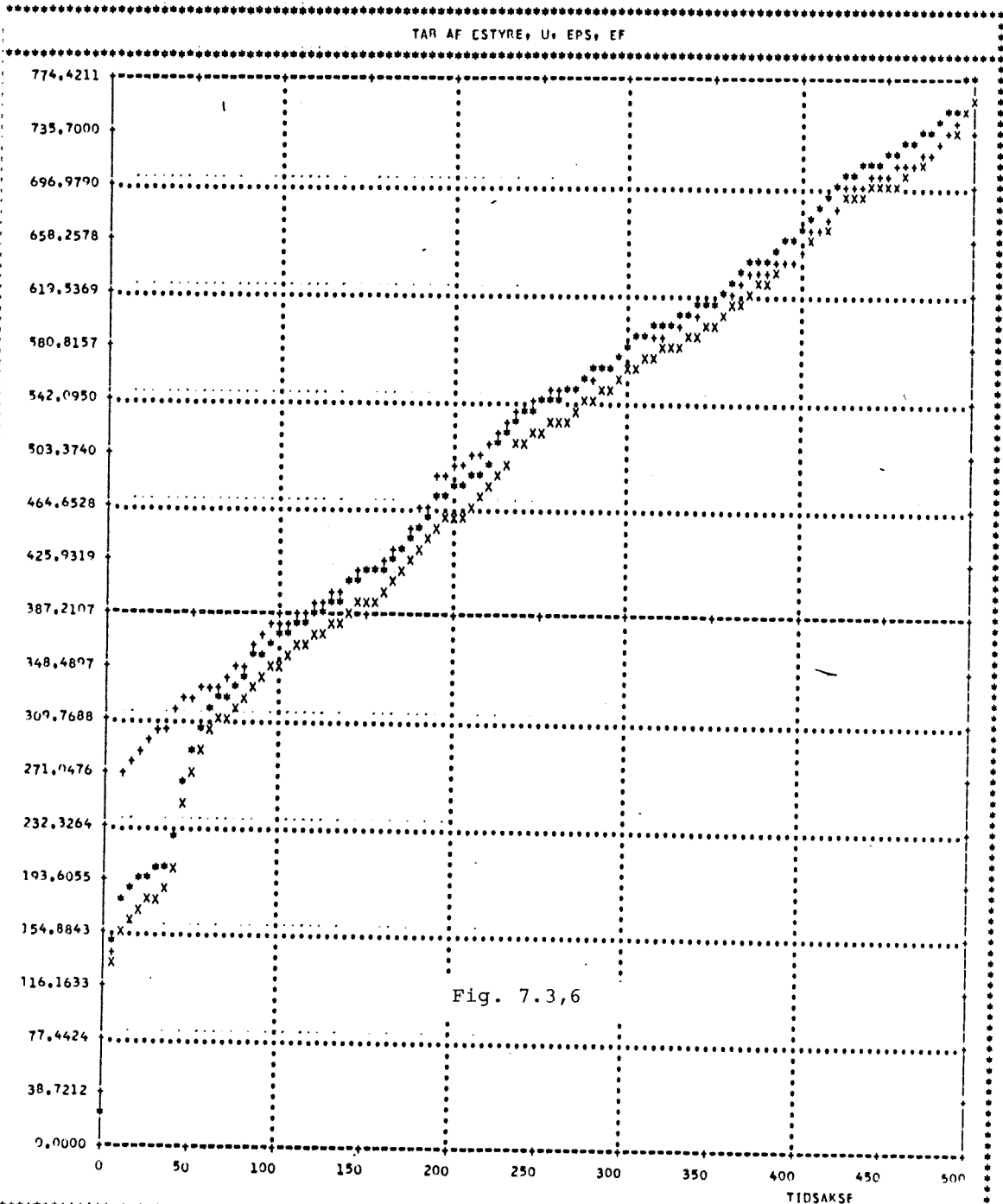
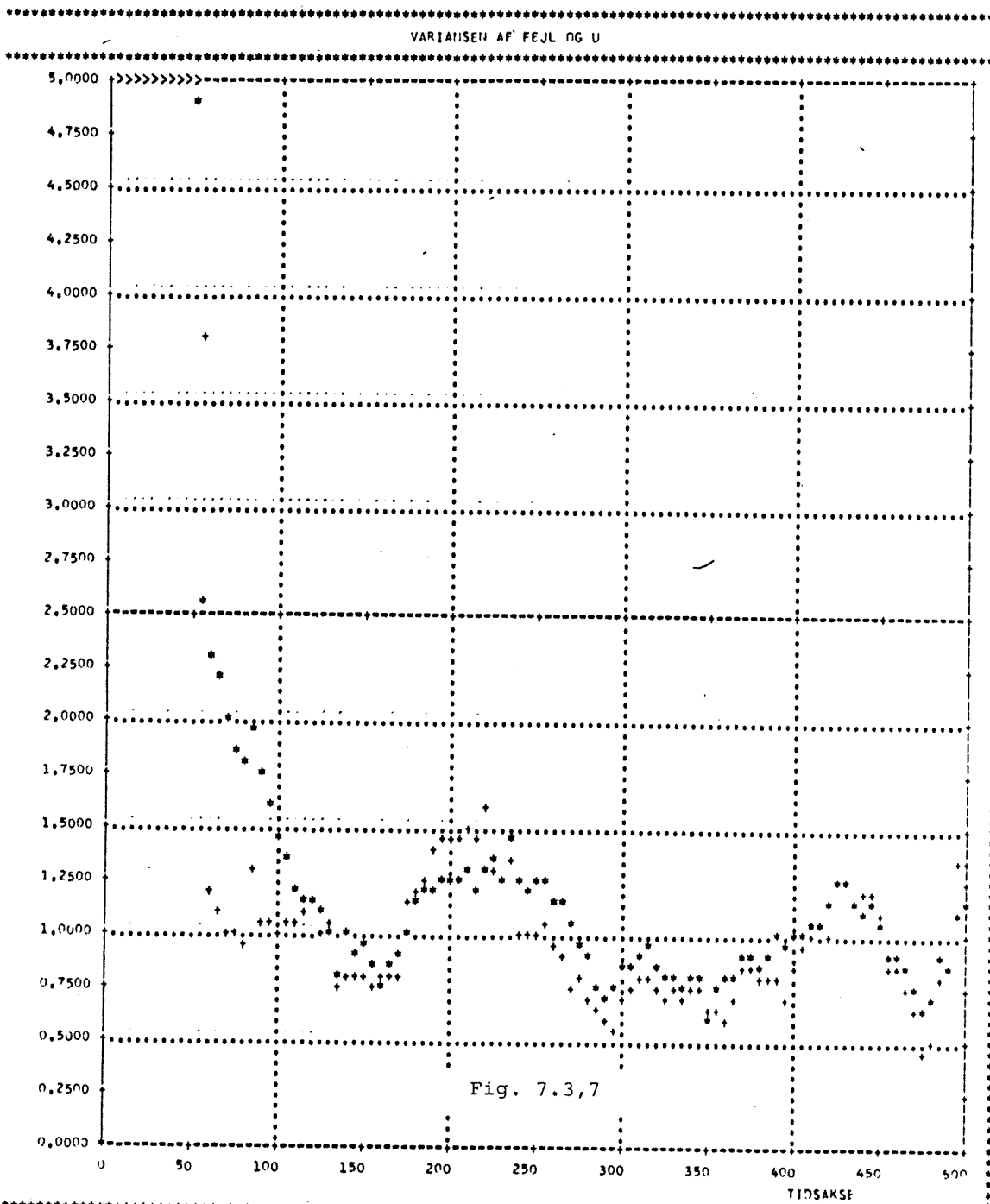
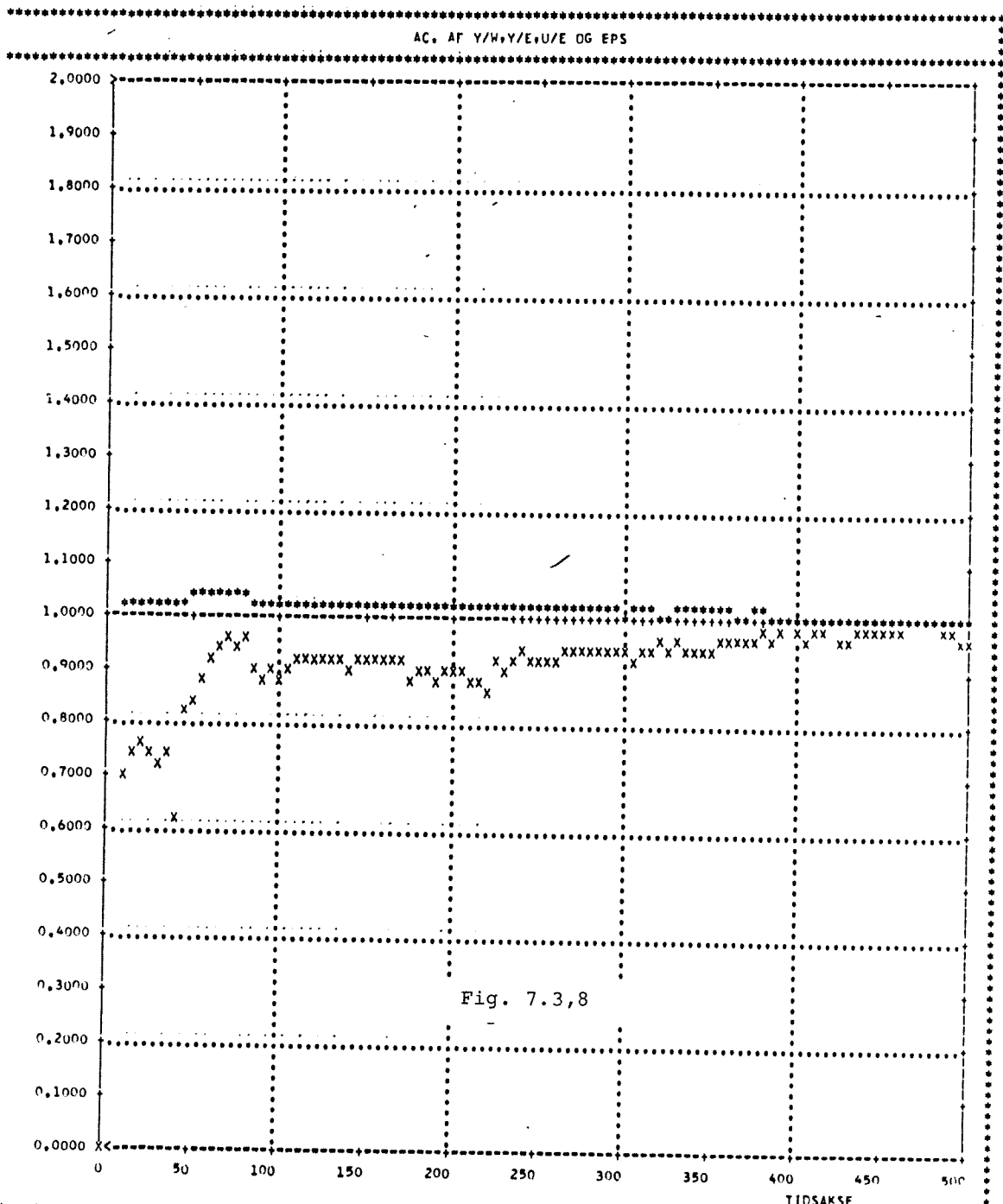


Fig. 7.3,6

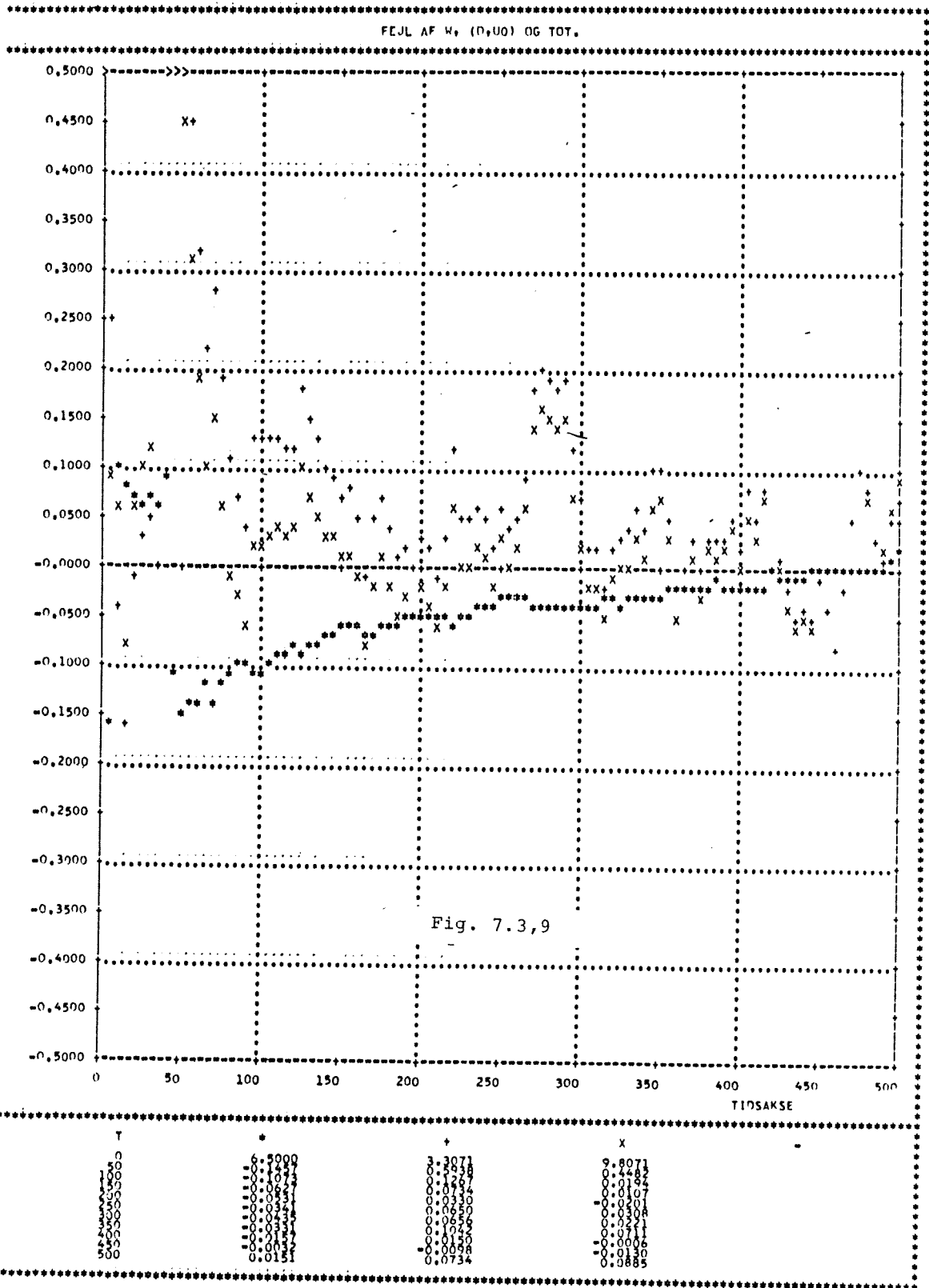
T	*	+	X	-
0	20.7760	3.3836	1.9482	20.7760
50	287.3293	37.5918	24.8594	287.3293
100	369.1487	42.6738	30.3703	369.1487
150	417.7658	47.8874	35.7541	417.7658
200	440.0884	52.3154	40.9346	440.0884
250	543.4827	60.8774	47.2603	543.4827
300	612.6475	65.3709	50.6548	612.6475
350	664.9336	69.7949	54.1221	664.9336
400	714.3210	74.4211	57.5624	714.3210
450			60.9747	
500			65.4929	
VAR	0.1000	0.5053	0.1000	0.1000

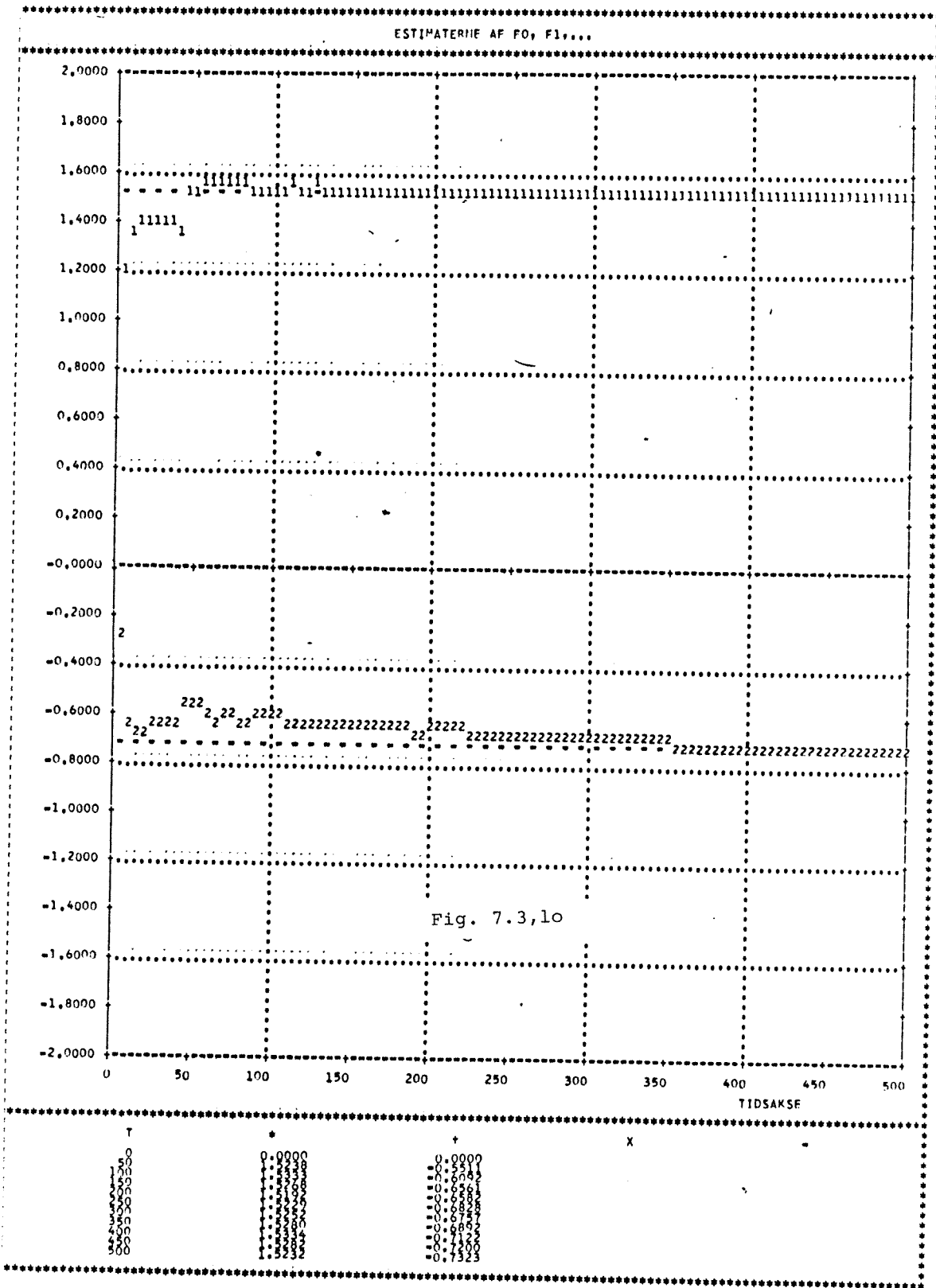


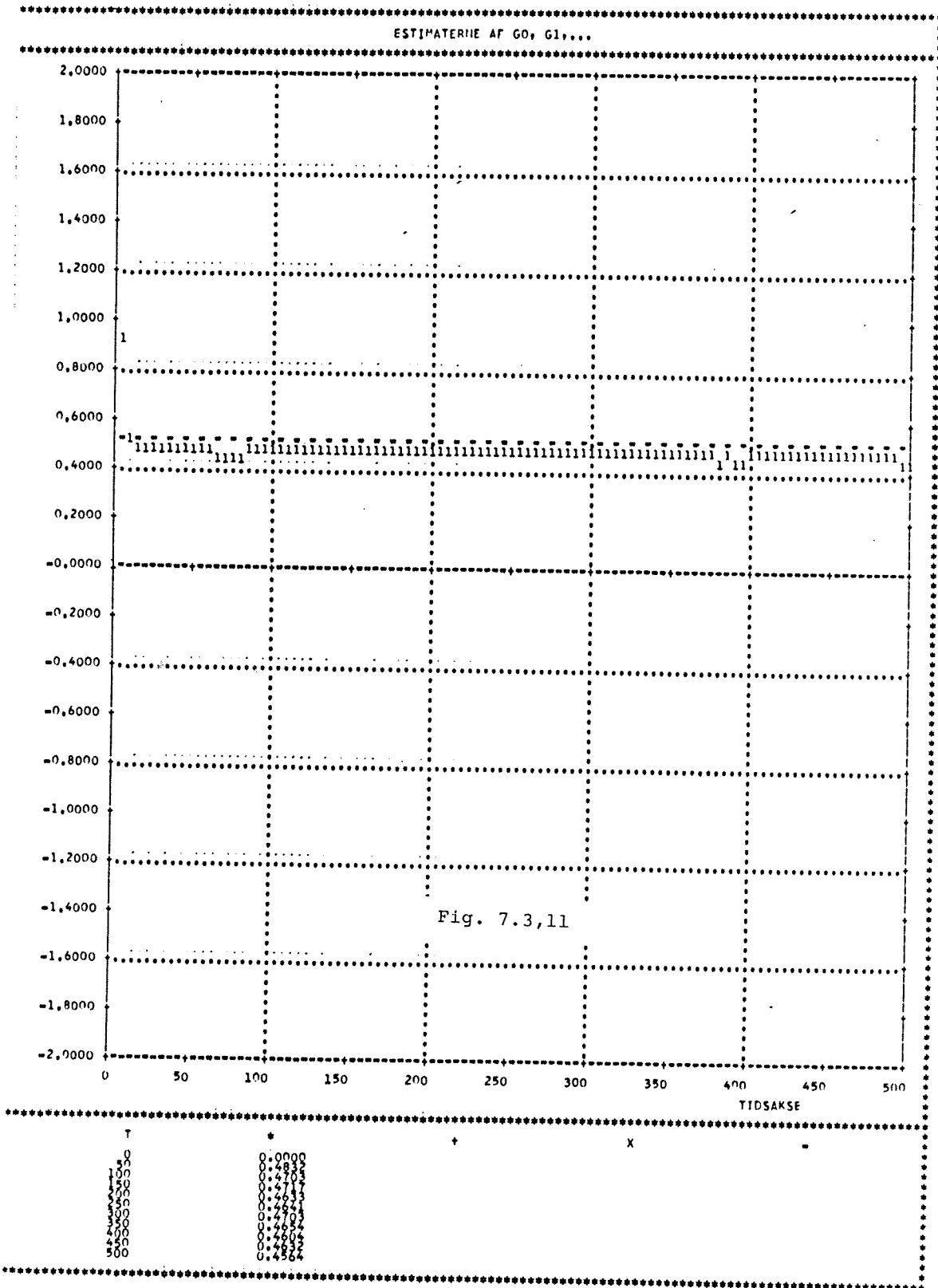
T	*	+	X	-
0	0.0000	0.0000		
50	0.8936	0.5080		
100	0.4324	1.0120		
150	0.7372	0.8243		
200	1.2250	0.4412		
250	0.2441	0.9978		
300	0.8678	0.7030		
350	0.8110	0.6315		
400	1.0372	1.0628		
450	1.1257	1.3342		
500				

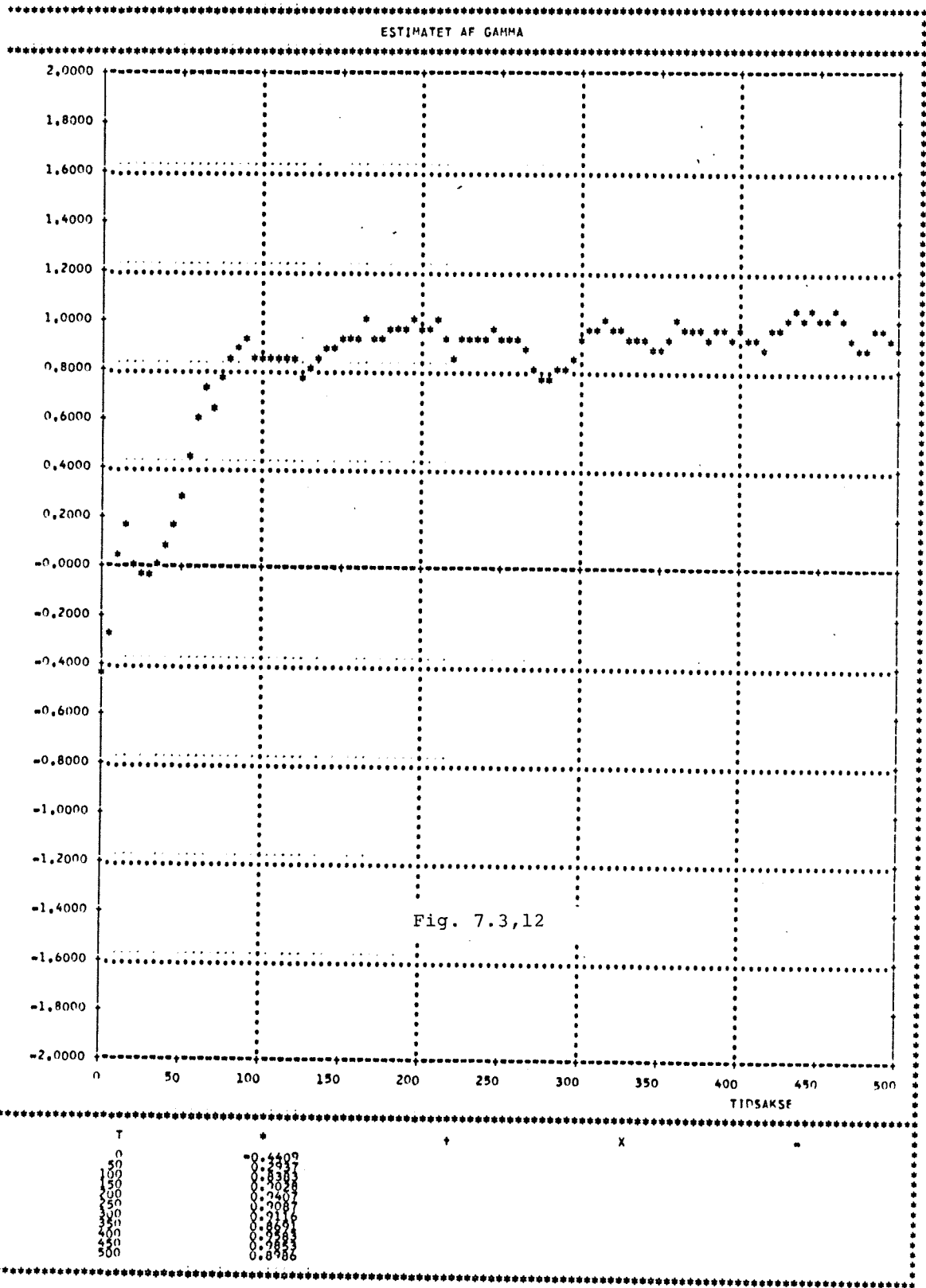


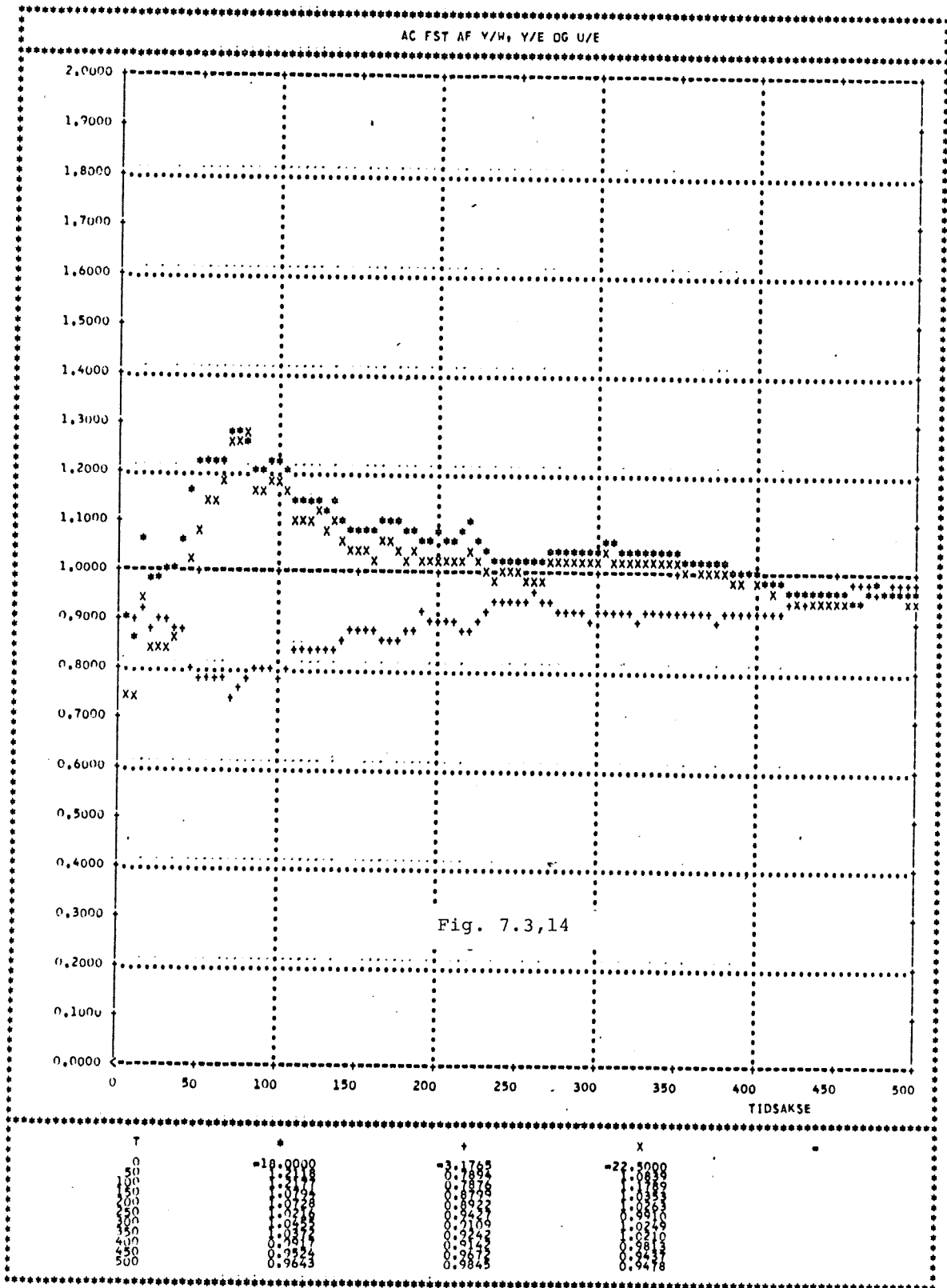
T	*	+	x	-
50	18.8804	8.8553	0.0000	8.8543
100	1.0327	1.0218	0.8369	1.0321
150	1.0252	1.0102	0.8234	1.0218
200	1.0146	1.0089	0.8099	1.0102
250	1.0120	1.0077	0.8023	1.0126
300	1.0048	1.0077	0.8279	1.0049
350	1.0103	1.0079	0.8386	1.0077
400	1.0098	1.0077	0.8761	1.0079
450	1.0076	1.0058	0.9807	1.0077
500	1.0046	1.0067	0.9683	1.0058
				1.0067

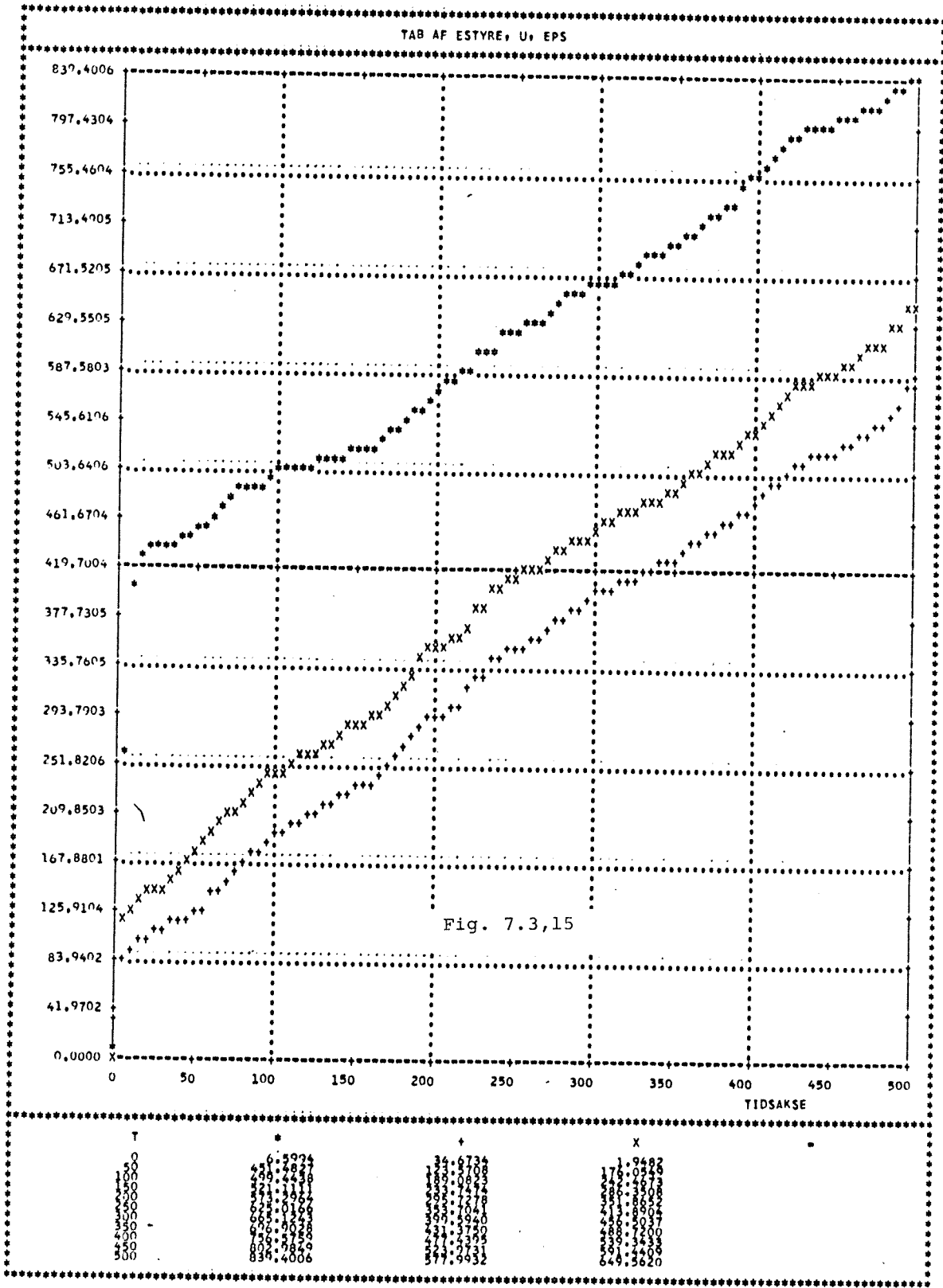












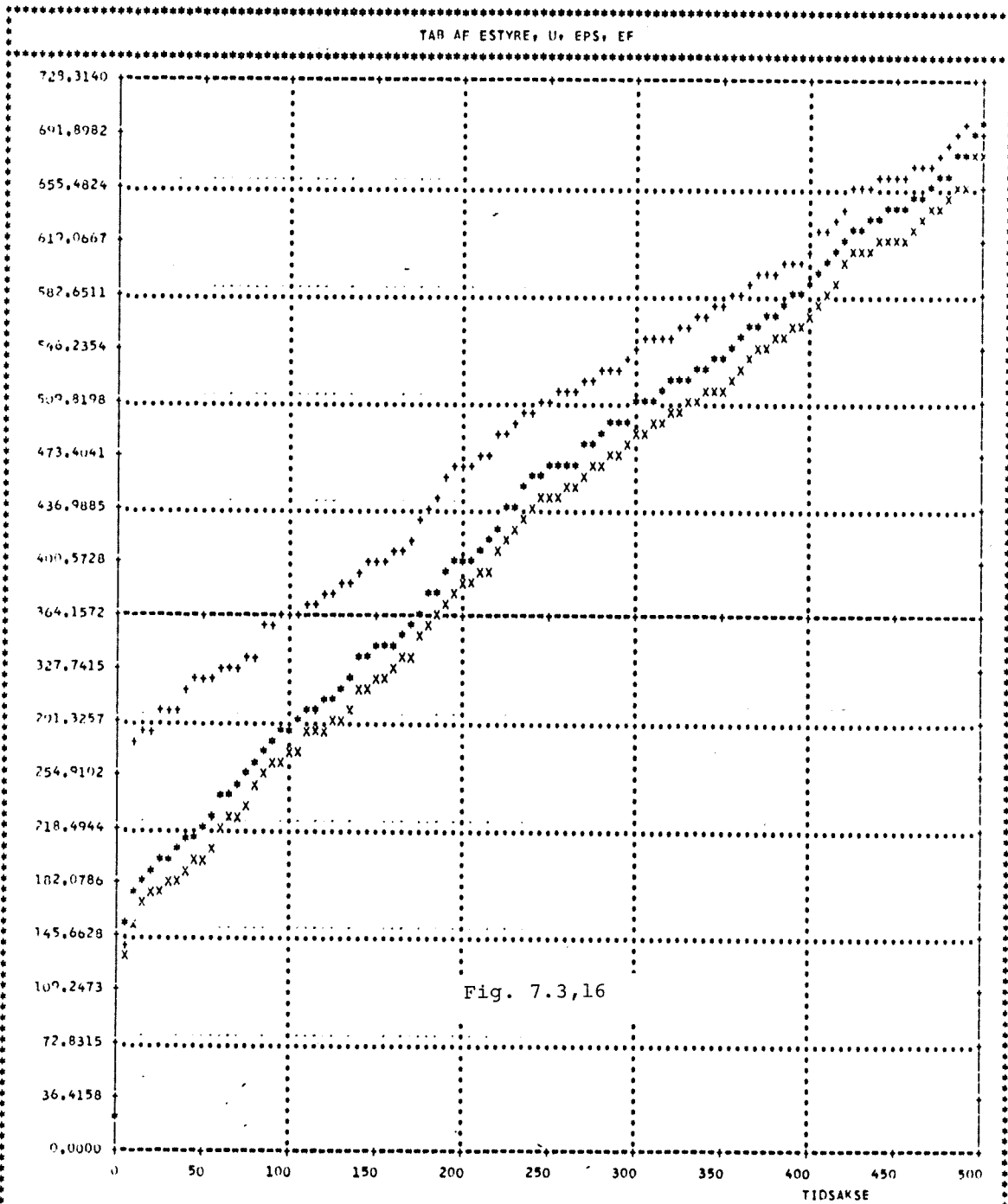
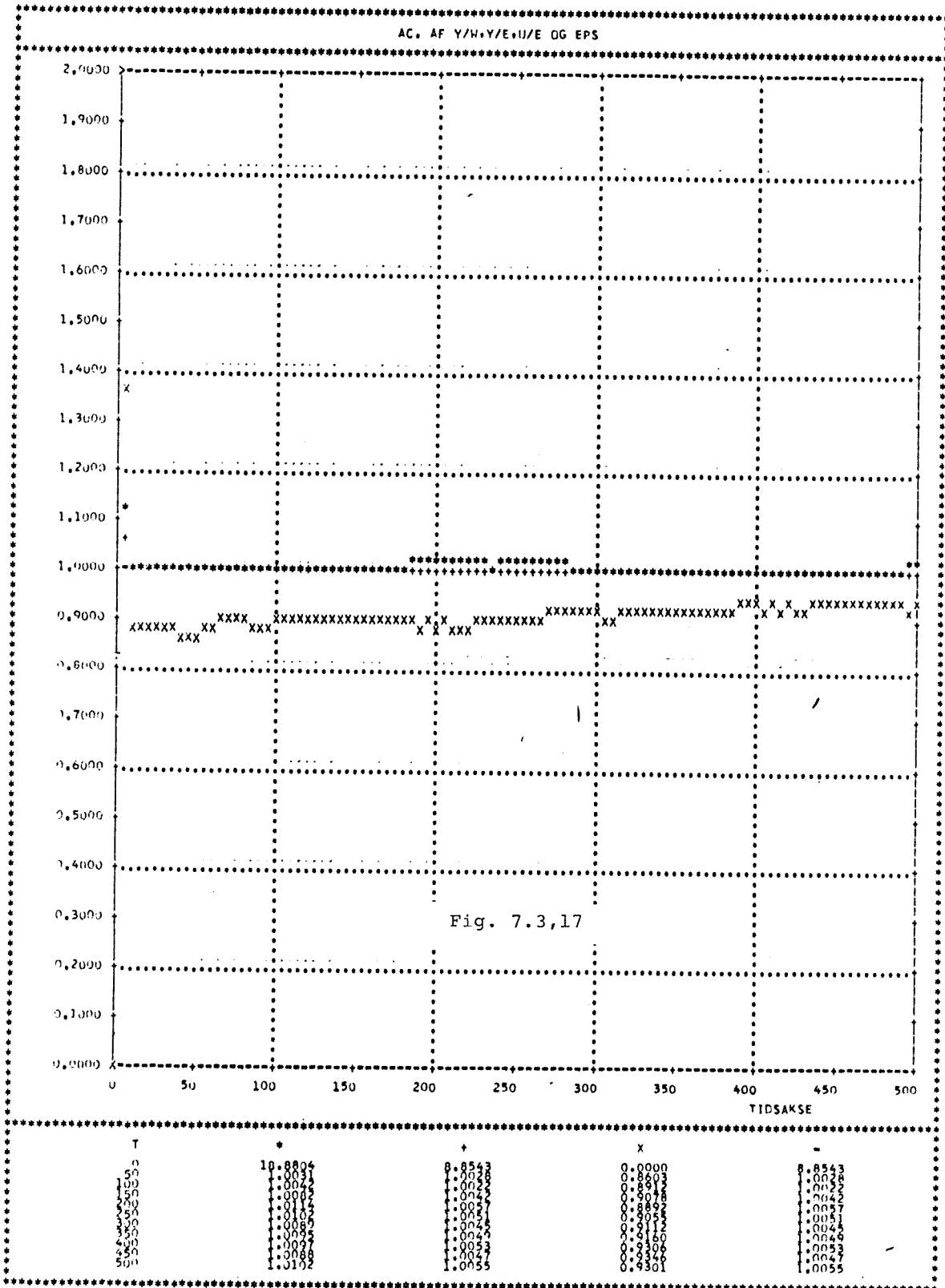
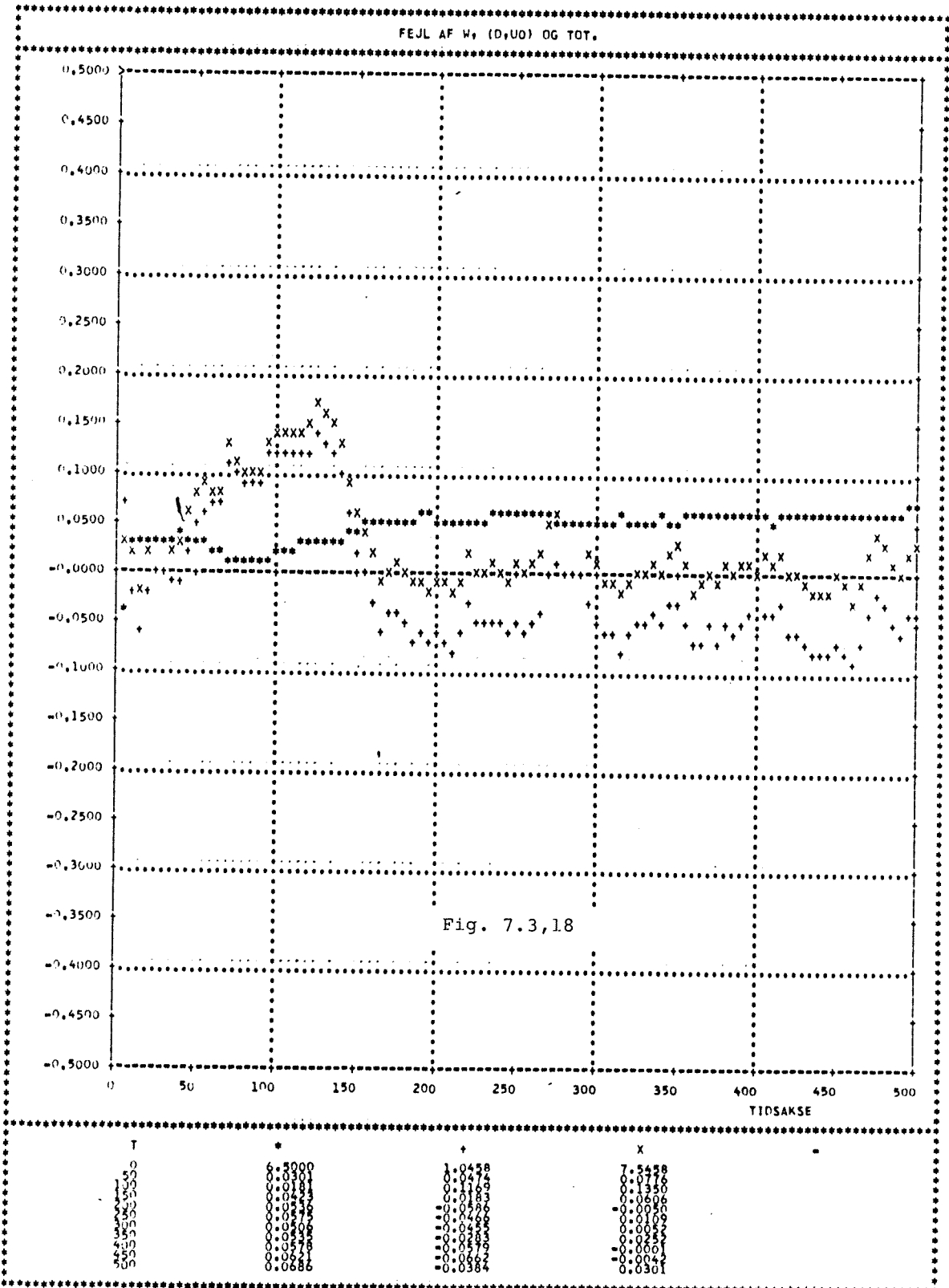
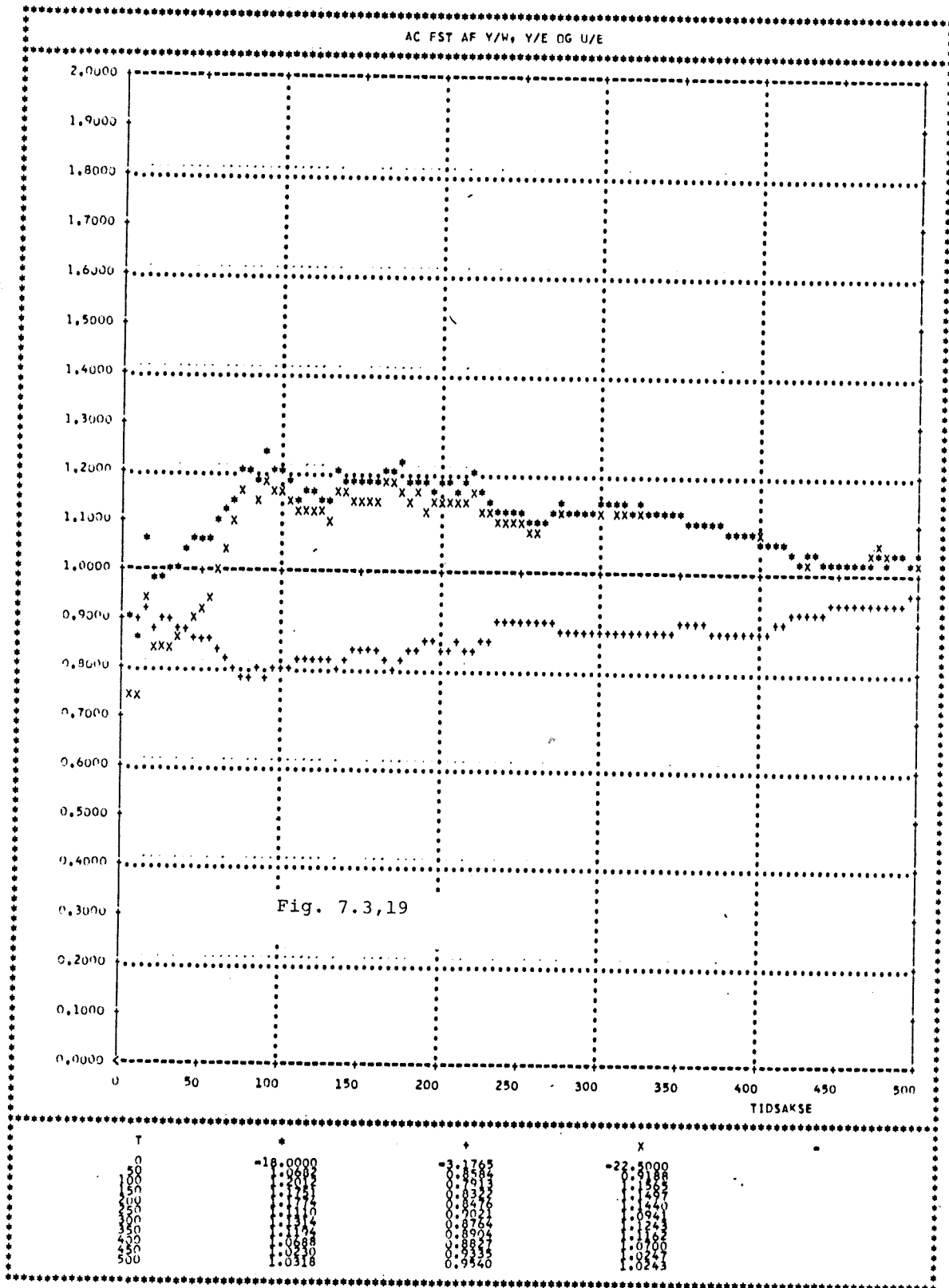


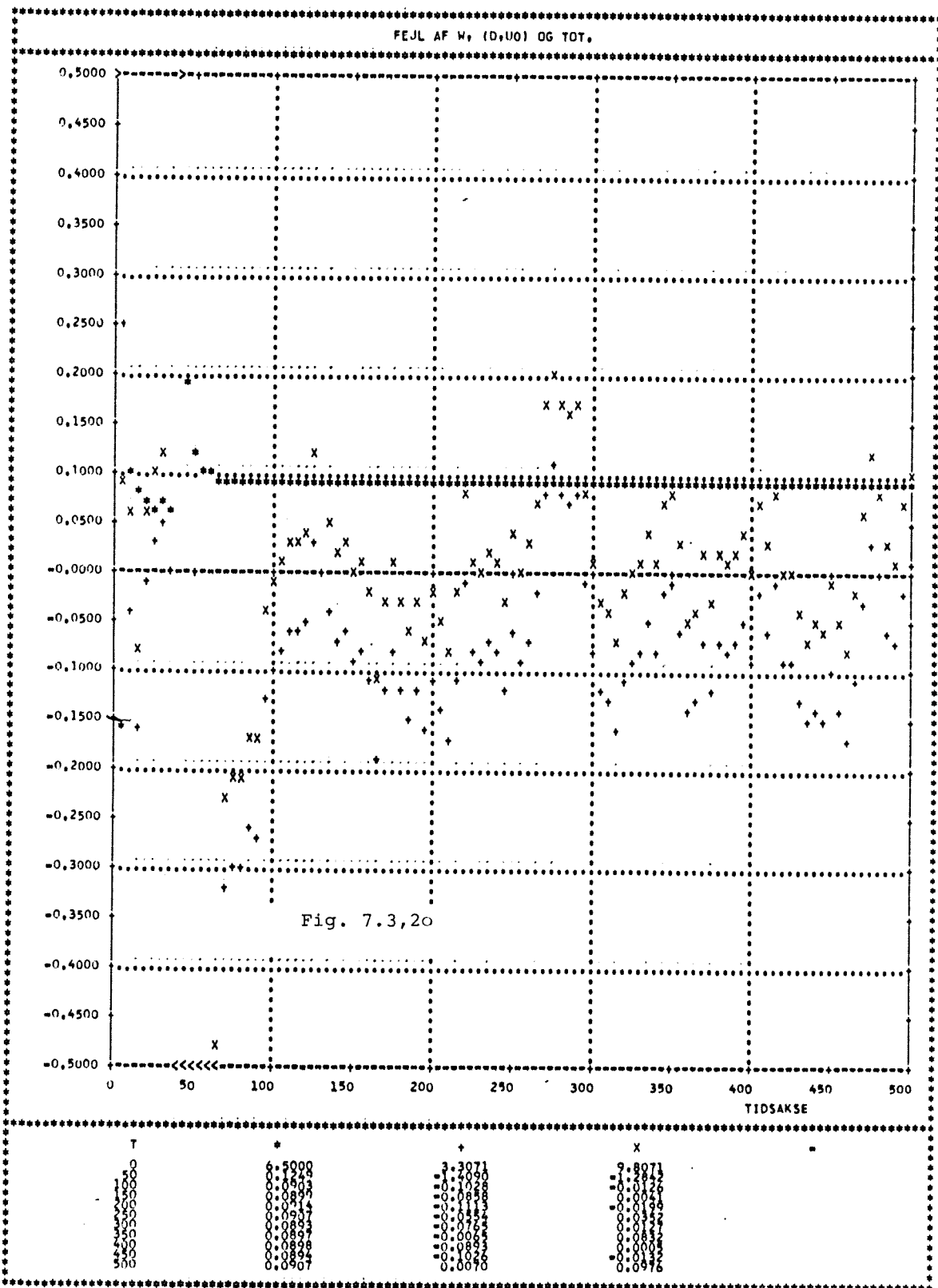
Fig. 7.3,16

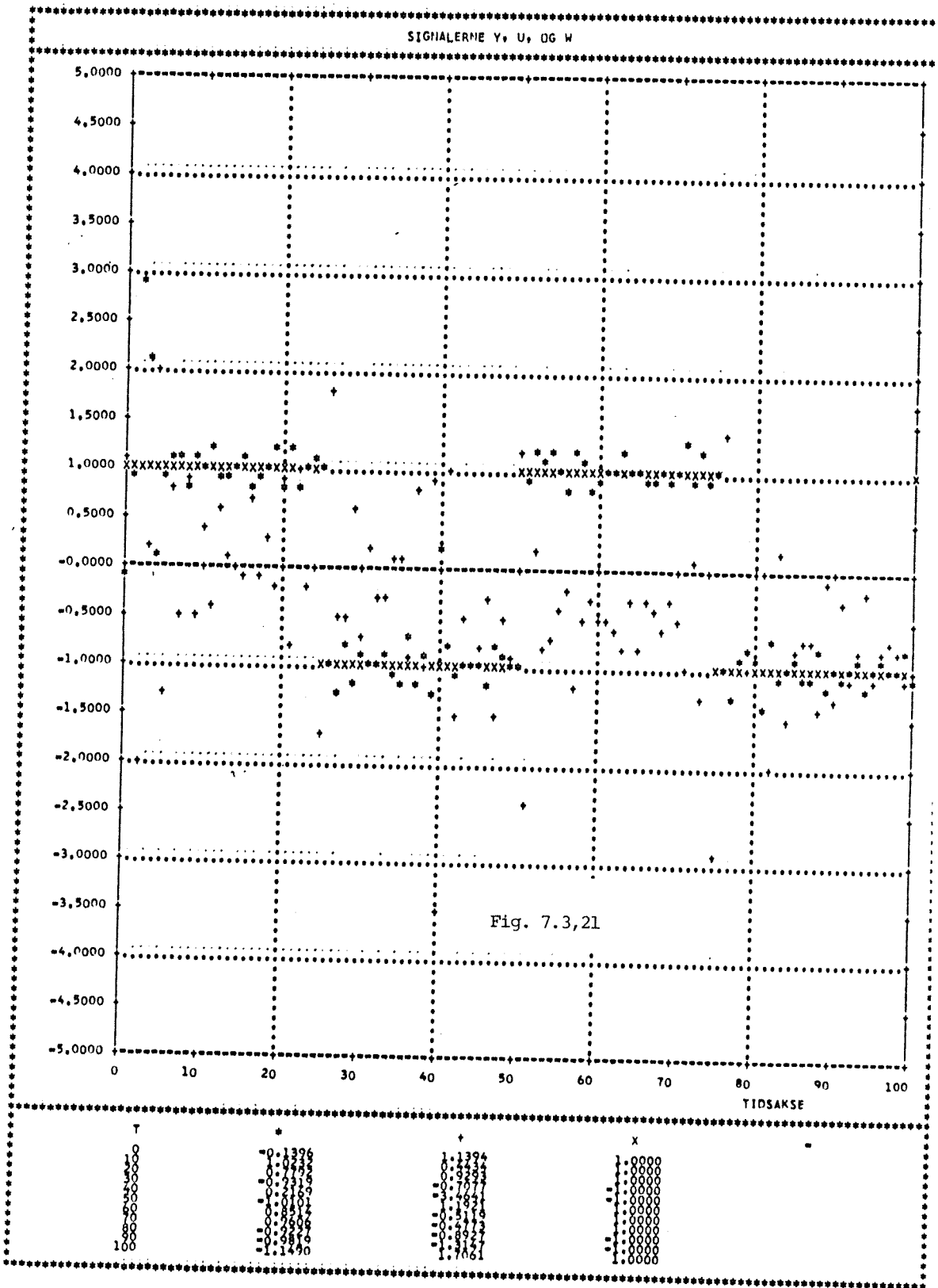
T	*	+	X	=
0	20,7760	3,3836	1,9482	20,7760
50	218,4944	321,8520	199,3480	218,4944
100	416,9888	639,7040	397,6960	416,9888
150	615,4832	957,5560	596,0440	615,4832
200	813,9776	1275,4080	794,3920	813,9776
250	1012,4720	1593,2600	992,7400	1012,4720
300	1210,9664	1911,1120	1191,0880	1210,9664
350	1409,4608	2228,9640	1389,4360	1409,4608
400	1607,9552	2546,8160	1587,7840	1607,9552
450	1806,4496	2864,6680	1786,1320	1806,4496
500	2004,9440	3182,5200	1984,4800	2004,9440
VAR	0,1000	0,5033	0,1000	0,1000

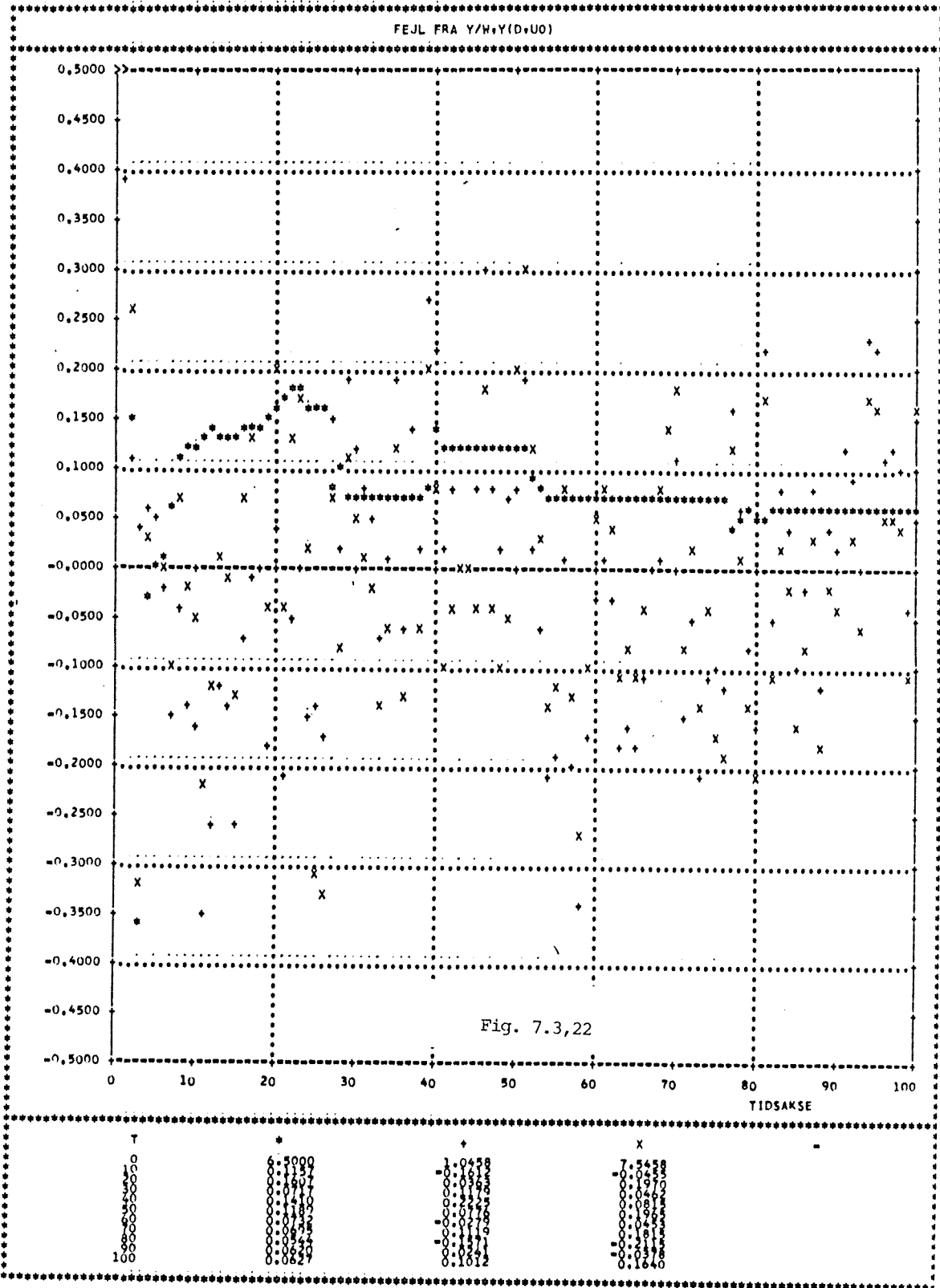


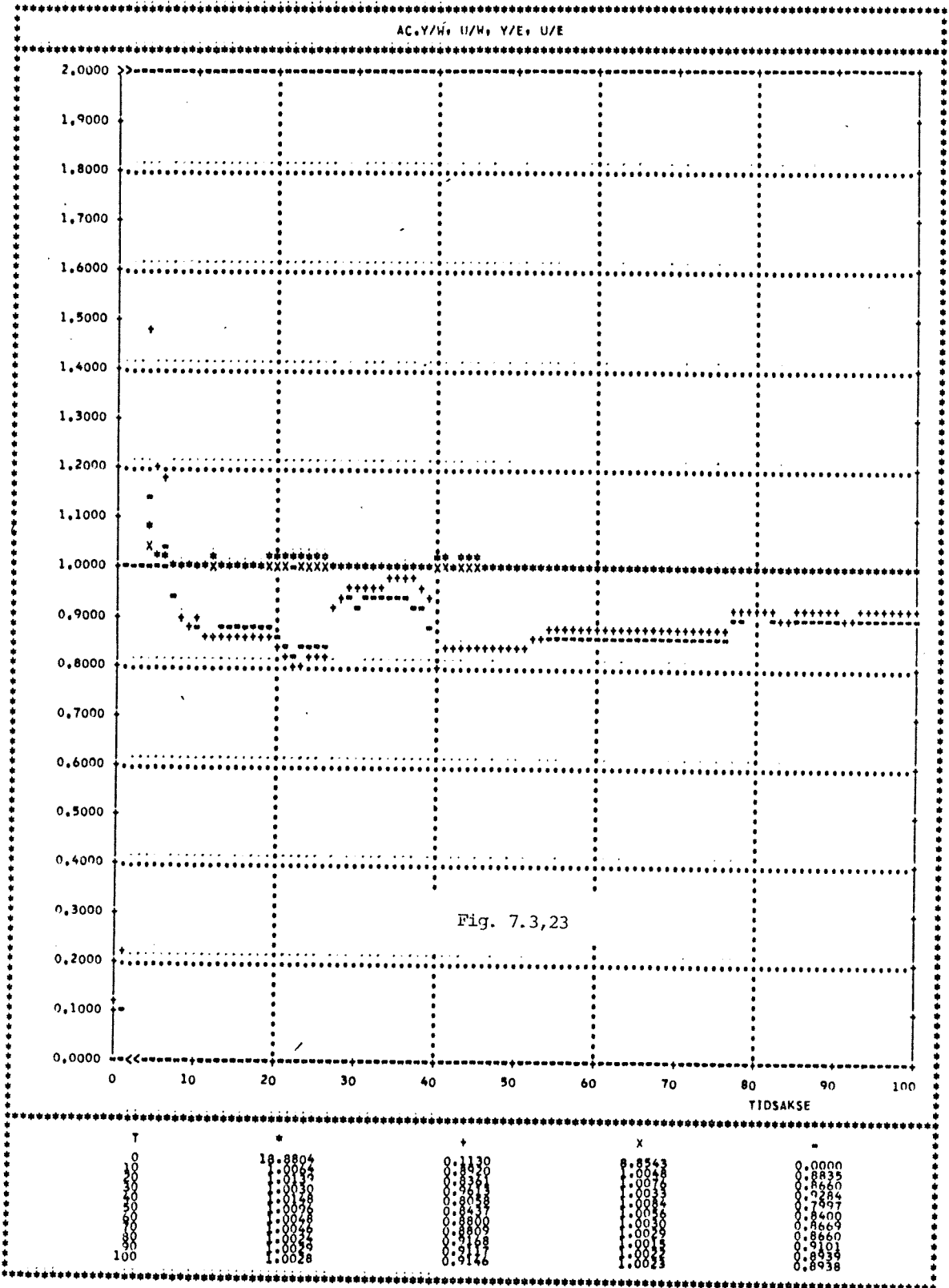












7.4 Afledt Model.

I dette afsnit vil der blive undersøgt en metode til at forbedre de adaptive regulatores egenskaber til at fjerne stationære fejl, der skyldes variationerne i belastningen, $d(t)$. Metoden er anvendt og beskrevet af Åstrøm og Wittenmark i /19/.

En forudsætning for, at en selvindstillende regulator kan anvendes til at bortregulere stationære fejl hidrørende fra $d(t)$, er naturligvis, at den faste regulator, som den er afledt af, kan dimensioneres til at klare dette problem. De generelle polplacering - og minimalvariansregulatorer opfylder denne betingelse ved en simpel tilbagekobling fra et estimat af belastningen. Selvom denne ikke blev opfattet som en konstant objektparameter, men en tilstand der kunne estimeres med et (udvidet) kalmanfilter, kunne dette estimat ikke af hensyn til præcisionen ikke spore $d(t)$, hvis denne springer for kraftigt. Deadbeat regulatorerne, som indeholder en integrator, bruger ikke estimatet af $d(t)$ i reguleringen. Imidlertid vil en fejl i dette estimat hurtigt brede sig til de øvrige, der bruges til beregning af regulatorparameterne. Det er derfor ønskeligt at finde en anden model for objektet, hvori netop $d(t)$ ikke indgår.

Hvis signalerne $y(t)$ og $u(t)$ filtreres gennem et filter med den simple overføringsfunktion:

$$(1) \quad H_1(q^{-1}) = 1 - q^{-1}$$

vil den hidtil anvendte model

$$(2) \quad \hat{A} y = q^{-k} \hat{B} u + \hat{C} \varepsilon + \hat{d}$$

blive ændret til:

$$(3) \quad \hat{A} y^r = q^{-k} \hat{B} u^r + \varepsilon_1$$

hvor $u^r = H(q^{-1})u$. Man ser meget hurtigt at for korrekte estimater er

$$(4) \quad \varepsilon_1^0(t) = (1 - q^{-1})C(q^{-1})e(t) + (1 - q^{-1})d(t)$$

Er $C = 1$ vil et RLS - estimat af parameterne være fejlagtigt på grund af de korrelerede residualer. Der kan desværre ikke bruges nogen form for compensation (som der normalt gøres med RELS blot med \hat{C}) fordi det kendte polynomie $H_1(q^{-1})$ har et nulpunkt på stabilitetsgrænsen. Følgelig kan RELS og RML ikke anvendes, når $C \neq 1$.

Ideen kan viderføres ved at ændre filteret til

$$(5) \quad H_2(q^{-1}) = \frac{1 - q^{-1}}{1 - \alpha q^{-1}} \quad 0 \leq \alpha < 1$$

hvorved der fremkommer en model analog til (3). Med korrekte estimater bliver residualerne:

$$(6) \quad \varepsilon_2^0(t) = \frac{1 - q^{-1}}{1 - \alpha q^{-1}} C(q^{-1})e(t) + \frac{1 - q^{-1}}{1 - \alpha q^{-1}} d(t)$$

For at få en tilstrækkelig god præcision af estimaterne i det stokastiske tilfælde skal filterets pol, α , vælges nær 1. Er $d(t)$ en trinfunktion vil residualerne få et bidrag, der svarer til et eksponentielt henfald med tidskonstanten

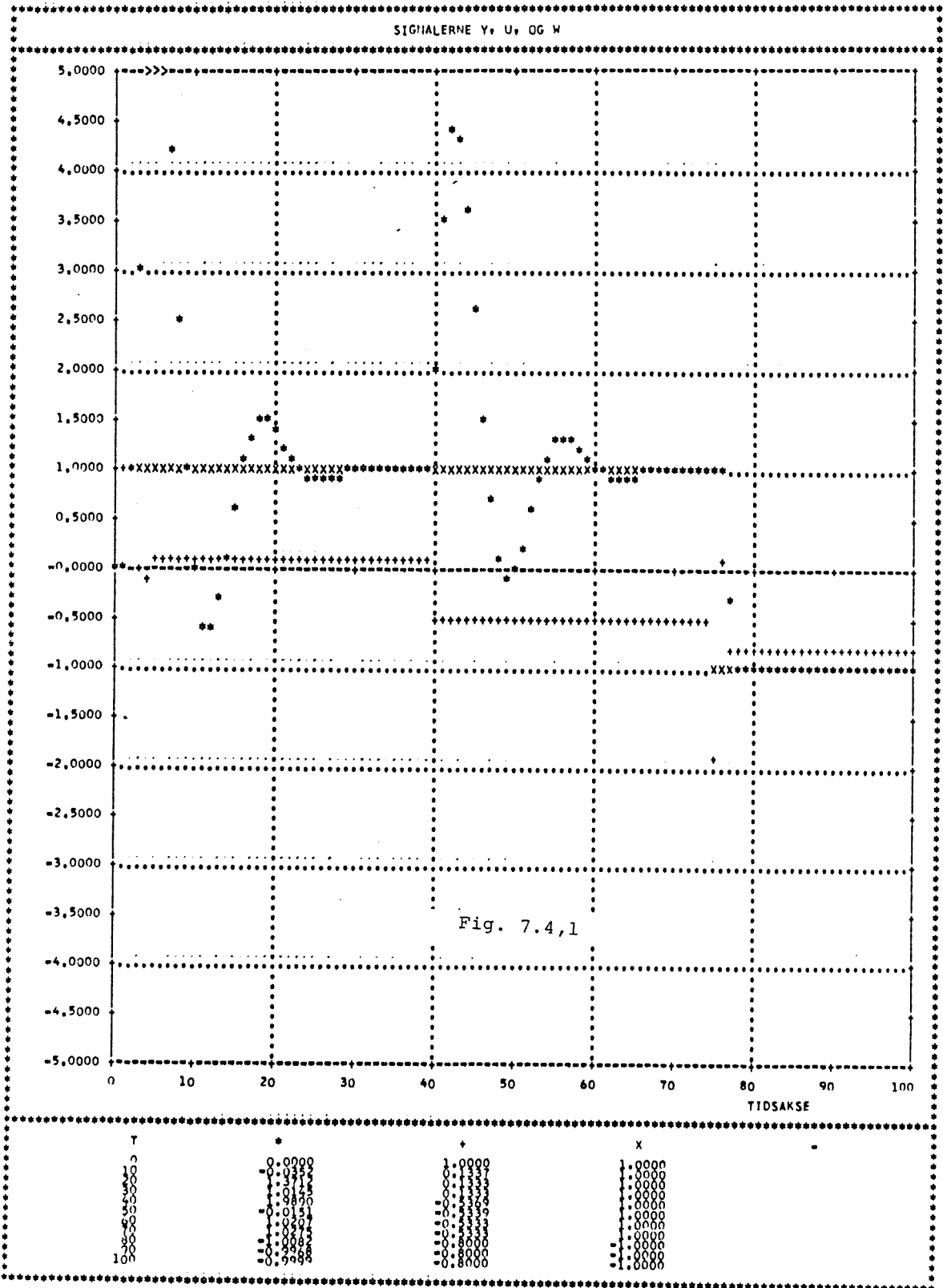
$$(7) \quad \tau = \frac{T_s}{\log(1/\alpha)}$$

hvor T_s er sampling intervallet. Vælges α for tæt på 1 er der ikke vundet meget i forhold til standardalgoritmen. Dilemmaet

fra kalmanfiltrene kan med andre ord ikke omgås med denne teknik. Ved simulationer er der fundet, at for objekt I giver $\alpha = 0.95$ en rimelig konvergens. Tidskonstanten i (7) bliver da $19.5 T_s$, hvorved der kan opnås væsentlige bedre resultater med eksponentiel glemsel.

Metoden kan med $\alpha = 0$ ligesom kalmanfilterne med fordel anvendes ved situationer, der er karakteriseret ved et stort signal - støjforhold, til at bortskaffe selv store spring i belastningen. Referencesignalet skal være tilstrækkelig eksisterende, men der er ikke som i de foregående afsnit et krav om, at det til stadighed skal have denne egenskab, fordi der ikke her er fare for Windup i kovariansmatricen for parameterestimerne. Som det fremgår af (4) er metoden bedst overfor abrupte ændringer i belastningen.

I fig. 1 er signalerne fra en deterministisk simulation med en deadbeatregulator (DBo) vist. Belastningen, $d(t)$, springer i $t = 40$ fra 0 til 1.



7.5 Integralvirkning.

Der eksisterer flere alternativer, når det gælder om at eliminere en stationær fejl fra en konstant belastning. En metode er at estimere belastningen og så koble direkte tilbage fra dette. Som det fremgår af de foregående afsnit, er det svært på sædvanlig vis at øge sporingshastigheden uden at ødelægge estimationspræcisionen. Et alternativ er at kombinere en selvindstillende regulator med en konventionel PI i en indre sløjfe som foreslået i /22/.

I dette afsnit vil der blive forsøgt at indlægge en integrator i en polplaceringsregulator, således at belastningen ikke indgår i modellen. Som i de øvrige tilfælde i kapitel 5 findes først en dimensionering for kendt objekt. Udgangspunktet er regulatorstrukturen i fig 5.3.1.1,3 for en polplacerings type, der plomberer alle objektets nulpunkter. Med betegnelserne:

$$(1) \quad I = 1 - q^{-1} \quad D = 1 - \alpha q^{-1}$$

kan der i strukturen indføres en integrator på en måde, som er anskueliggjort i fig. 1.

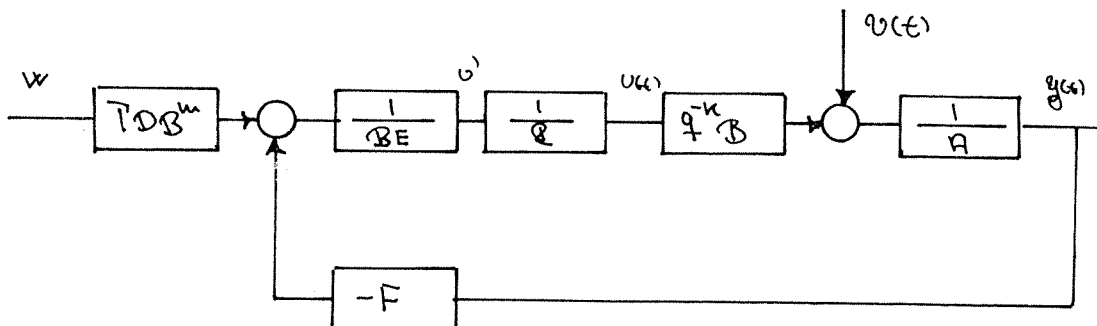


Fig. 1: Regulator med integralvirkning.

Det ekstra nulpunkt, der er indført med D, er medtaget for at give en yderligere frihedsgrad. Det vil senere i afsnittet vise sig, at man med valget af α kan dæmpe integralvirkningen. Målet med denne deterministiske dimensionering er, ligesom i 5.3 at opnå en overføring fra referencen, $w(t)$, til objektets udgang, $y(t)$, der kan beskrives med:

$$(2) \quad H_{y,w} = q^{-k} \frac{B^m(q^{-1})}{A^m(q^{-1})}$$

Derfor udregnes den faktiske overføringsfunktion i fig. 1 til

$$(3) \quad H_{y,w} = q^{-k} \frac{TDB^m}{AIE + q^{-k} F}$$

Ud fra kravet om identitet mellem (2) og (3) kan dimensioneringsligningen udtrykkes som:

$$(4) \quad TDA^m = AIE + q^{-k} F$$

Vælges der ligesom i de foregående tilfælde (undtagesn dead - beat regulatorene) at:

$$(5) \quad n_E = k - 1$$

skal:

$$(6) \quad n_F = \max(n_T + n_{A^m}^{-k}, n_A^{-1}) + 1$$

Såfremt (4) er opfyldt vil overføringsfunktionerne være givet ved (2) og

$$(7) \quad H_{y,v} = \frac{E}{TA^m} \frac{I}{D}$$

$$(8) \quad H_{u,w} = \frac{AB^m}{A^m B}$$

$$(9) \quad H_{u,v} = - \frac{F}{TDA^m B}$$

Ved sammenligning med de tilsvarende udtryk for polplaceringsregulatoren (5.3.1.1,32 - 35) bemærkes det, at med denne regulator vil en konstant belastning ikke give anledning til en stationær fejl, selvom denne ikke bliver estimeret. Man kan også se, at det kun er systemets egenskaber overfor forstyrrelserne, der er ændret. Overføringsfunktionen i (7) viser at indlæggelsen af integratoren forringer systemets dynamiske egenskaber overfor støjpåvirkninger. Forberingen af de statiske egenskaber sker øjensynligt på bekostning af de dynamiske.

Til at finde en model i regulatorparameterne følges fremgangsmåden, der er anvist af Egardt /5/. Der indføres følgelig to polynomier Q_4 og P_4 ved

$$(10) \quad \begin{aligned} Q_4(q^{-1}) &= 1 + q_1 q^{-1} + \dots + q_{n_Q} q^{-n_Q} \\ P_4(q^{-1}) &= P^1(q^{-1}) P^2(q^{-1}) \\ &= 1 + p_1 q^{-1} + \dots + p_{n_P} q^{-n_P} \\ P^1(0) &= P^2(0) = 1 \end{aligned}$$

Både Q_4^* og P_4^* skal have alle rødder indenfor enhedscirklen. Den filtrerede modelfejl defineres som

$$(11) \quad e_f(t) = \frac{Q_4(q^{-1})}{P_4(q^{-1})} e_m(t)$$

hvor modelfejlen er

$$e_m(t) = y(t) - q^{-k} \frac{B^m(q^{-1})}{A^m(q^{-1})} w(t)$$

Den adaptive regulator bestemmes ud fra

$$\begin{aligned} \text{TDA}^m e_m(t) &= \text{TDA}^m y(t) - q^{-k} \text{TDB}^m w(t) \\ &= q^{-k} [Fy(t) + (BE - b_o P^2) Iu(t) \\ &\quad + b_o P^2 Iu - \text{TDB}^m w(t)] \\ &\quad + IEv(t) \end{aligned} \tag{12}$$

samt definitionen af polynomiet G

$$\begin{aligned} G(q^{-1}) &= q(BE - b_o P^2) \\ n_G &= \max(n_B + k - 1, n_{P^2}) \end{aligned}$$

og signalererne:

$$\begin{aligned} y_2(t) &= \frac{Q_4}{\text{TDA}^m P_4} y(t) & u_2(t) &= \frac{Q_4}{\text{TDA}^m P_4} Iu(t) \\ u_1(t) &= P^2 u_2(t) & w_2(t) &= \frac{B^m Q_4}{A^m P_4} w(t) \end{aligned} \tag{13}$$

Hermed kan den filtredede fejl skrives som

$$e_f(t) = q^{-k} [Fy_2(t) + Gu_2(t) + b_o u_1(t) - w_2(t)] + \varepsilon_o(t) \tag{14}$$

hvor $\varepsilon_0(t)$ er givet ved:

$$\varepsilon_0(t) = \frac{Q_4 I E}{T D A^m P_4} v(t)$$

Dermed er der mulighed for at estimere parameterne i G og F - polynomiet ud fra modellen

$$(15) \quad y_f(t) = q^{-k} (\hat{F}Y_2 + \hat{G}u_2 + b_0 u_1) + \varepsilon(t)$$

hvor

$$(16) \quad y_f(t) = \frac{Q_4(q^{-1})}{P_4(q^{-1})} y(t)$$

I den selvindstillende regulator bestemmes styresignalet ved

$$(17) \quad u_1(t) = \frac{1}{b_0} [w_2 - \hat{F}Y_2 - \hat{G}u_2]$$

$$u(t) = \frac{T A^m P^l}{Q_4} \cdot \frac{D}{I} u_1(t)$$

Strukturen i den selvindstillende regulator er vist i fig. 2.

Som i det ordinære tilfælde kan der dimensioneres en selvindstillende regulator ud fra et stokastisk grundlag ved at sætte $T = C$ i (12) og udføre en beregning analog til den givne i afsnit 5.3. I dette tilfælde bliver modellen:

$$(18) \quad y_f = \frac{q^{-k}}{C} [\hat{F}Y_2 + \hat{G}u_2 + b_0 u_1] + \varepsilon_1$$

Residualerne, $\varepsilon_1(t)$, er for korrekte estimater givet ved:

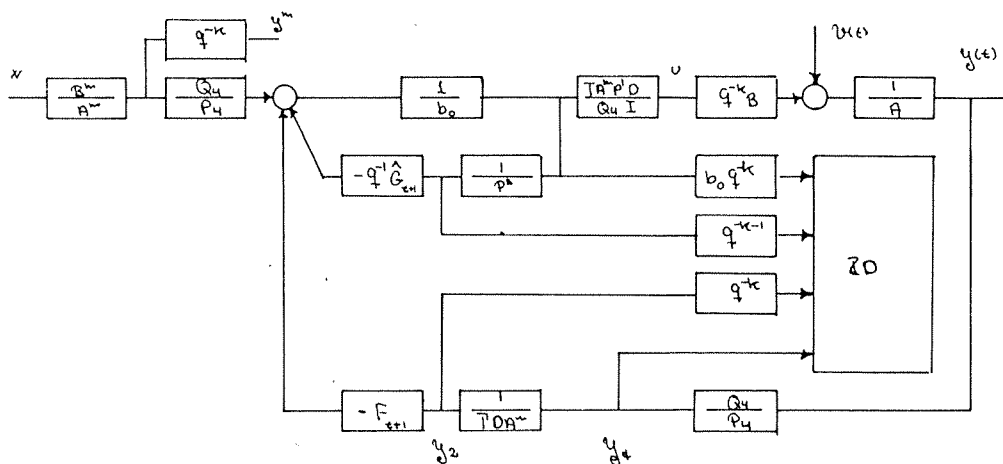


Fig. 2: Blokdiagram for selvindstillende regulator med integralvirkning

$$(19) \quad \varepsilon_o(t) = \frac{Q_4 E}{A^m P_4 C} \frac{I}{D} v(t)$$

Da nulpunktet i $I(q^{-1})$ sikrer at konstante belastninger ikke påvirker residualerne, vil det ikke have nogen mening at op - hæve integralvirkningen ved at indlægge et tilsvarende nul - punkt i f. eks. $D(q^{-1})$. Dette betyder samtidigt, at resi - dualerne er korreleret med regressorene, hvorfor en integralvirkning er mest relevant ved støjsvage tilfælde. Det er dog muligt at finde en værdi for α , det vil sig nulpunktet for D^* , der bevirker at denne korrelation er minimal og på den anden side ikke helt ophæver integralvirkningen. Dette nulpunkt indgår j.v.f. (19) i tidskonstanten for tidsforløbet af et springs indvirkning på residualerne. I projektet blev

der fundet, at $\alpha = 0.985$ giver tilfredstillende resultater, når testobjektet (objekt I) bliver udsat for et enheds - spring i $t = 40$. Valget er ret kritisk og sandsynligvis afhængig af objektet. Herefter vil kun den deterministiske version blive undersøgt. For denne type kan valget af tuningparameter træffes på samme måde som i afsnit 5.3. Dette gælder også for den stokastisk dimensionerede regulator.

Når der er et stort signal - støjforhold, og når reference - signalet eller et andet kendt tilsat signal er tilstrækkelig eksiterende, kan den deterministisk dimensionerede version anvendes. Bruges en sådan regulator overfor objekt I med polynomierne:

$$(20) \quad A^m = B^m = T = 1 \quad \text{og} \quad D = 1$$

vil de korrekte regulatorparameter være givet ved

$$(21) \quad \begin{aligned} F(q^{-1}) &= 2.5 - 2.2q^{-1} + 0.7q^{-2} \\ G(q^{-1}) &= 0.5 \end{aligned}$$

Hvis støjvariansen af $e_0(t)$ er 0.1 vil denne regulator give anledning til at

$$(22) \quad \text{Var}(y) = 0.2 \quad \text{Var}(u) = 2.6$$

når $w(t) = 0$. Forøgelsen i forhold til MVo - reguleringen skyldes for udgangens vedkommende nulpunktet i $I(q^{-1})$. Med det indførte nulpunkt, α , i D^* kan denne varians sænkes, men samtidig vil algoritmens evne til at klare spring blive nedsat. Indsættelsen af integratoren har medført en 5 - dobling af styrevariansen i forhold til MVo - regulatoren. Også i dette tilfælde vil en flytning af nulpunktet, α , ud fra origo mod 1 betyde mindre styreeffekt, hvilket fremgår af (4) der i

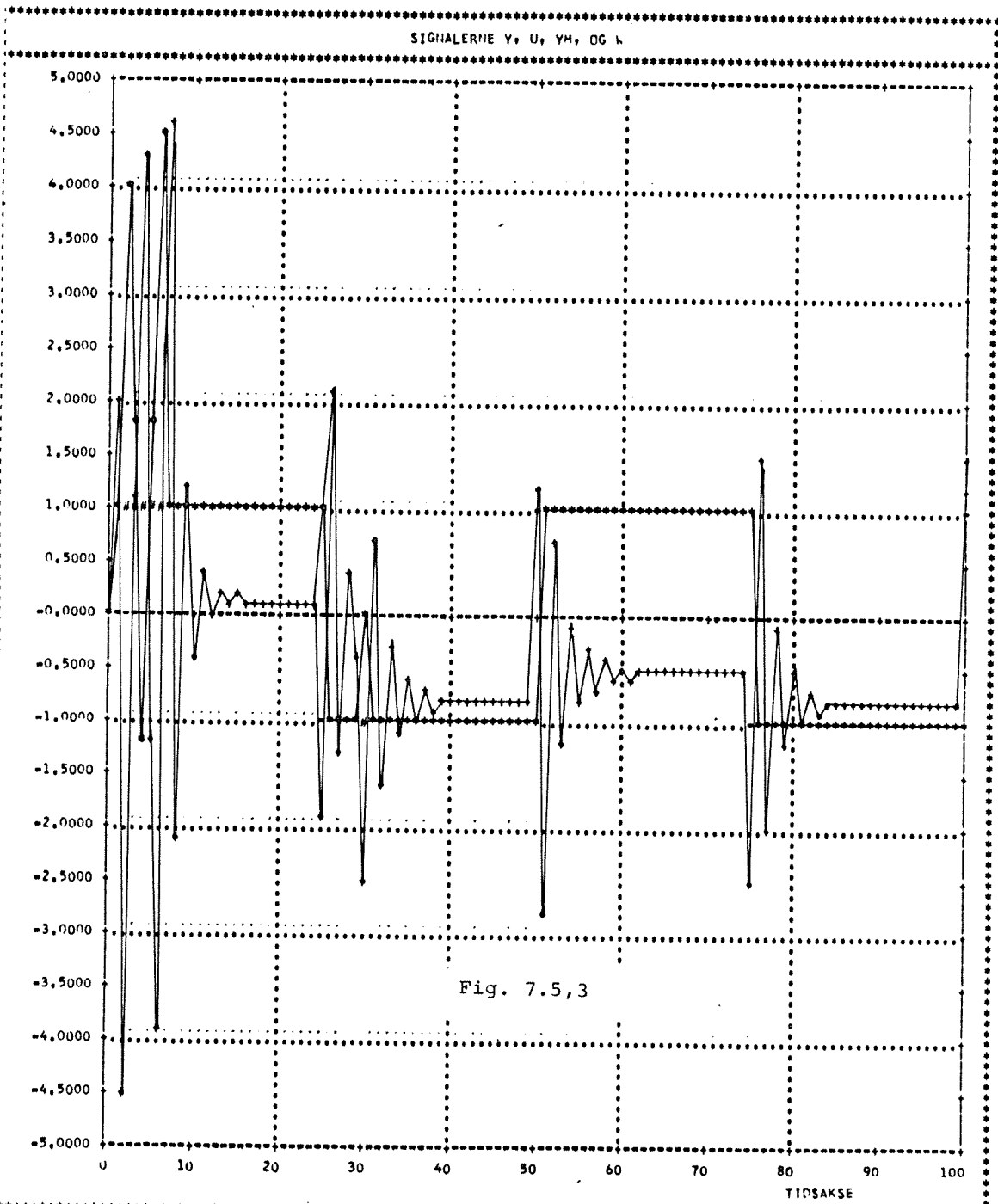
dette specialtilfælde lyder:

$$(23) \quad F = q(AI - D)$$

Vælges f. eks. $\alpha = 0.9$ vil varianserne være reduceret til 0.105 og 0.58 for henholdsvis udgangssignalet og styresignal. Styre - effekten kan også reduceres ved at vælge en anden model be - stemt ved A^m og B^m .

Fig. 3 viser signalerne for en støjfri simulation med den omtalte regulator, når referencesignalet er en firkantbølge og $d(t)$ springer fra 0 til 1 i $t = 30$. Det ses at springet ud - reguleres i løbet af 1 trin og at reguleringen er uændret herefter. De kraftige oscillationer i styresignalet skyldes objektets nulpunkt i venstre halvplan ($- 0.5$). Som en følge af at overføringsfunktionerne fra referencesignalet for kor - rekte estimater er ens for denne og pølplaceringsregulatorerne er tabene over de sidste 60 simulationstrin lige så store som opnået med en MVo -regulator, når belastningen er konstant. Denne regulator kan derfor godt anvendes til at klare be - lastningsvariationer, når støjens indflydelse er neglisibel.

---ooo0ooo---



T	#	+	#	=
0	0.0000	1.0000	0.0000	1.0000
10	1.0017	-0.2261	1.0000	1.0000
20	-1.0000	0.1328	-1.0000	-1.0000
30	-0.0001	-2.4225	-1.0000	-1.0000
40	-1.0520	-1.8320	-1.0000	-1.0000
50	-1.0121	-0.2484	-1.0000	-1.0000
60	-1.0124	-0.5209	-1.0000	-1.0000
70	-1.0124	-0.4186	-1.0000	-1.0000
80	-1.0111	1.1982	-1.0000	-1.0000
100			1.0000	1.0000

7.6 Fejldetektor.

I de foregående afsnit er der beskrevet nogle få metoder, der kan anvendes når forstyrrelserne ikke er stationære. Karakteristisk for nogle af disse metoder er, at deres egenskaber kun er antagelige overfor langsomme variationer, der her er betegnet som drift. Der er naturligvis stor forskel på egenskaberne, og der er i nogle tilfælde problemer med valg af parametre, der skal ansættes a priori. Overfor spring er deres egenskaber ikke særligt gode, når disse er større end støjniveauet. Dette skyldes, at man ikke samtidig kan opnå en rimelig estimationspræcision og en høj sporingshastighed, der er nødvendig ved større spring, for at undgå, at disse driver parameterestimererne bort fra deres oprindelige leje.

Fælles for de før omtalte metoder er, at filteret, der varetager estimationen, kun har en arbejdstilstand. Det er derfor begrænset hvor gode egenskaber, der kan opnås i specielle situationer, når reguleringen under normale omstændigheder ikke må forringes. Dette er det grundlæggende dilemma ved de nævnte metoder. For at opnå en separation mellem de to modstridende krav, vil der i dette afsnit blive angivet en metode, der bygger på, at estimationsfilteret har to arbejdstilstande hver tilpasset de to situationer. Metoden er, som følge af de øvrige metoders rimelige egenskaber overfor drift, kun dimensioneret overfor abrupte ændringer af belastningen, $d(t)$. Samtidig er den tredje type påvirkning, spidserne, ikke direkte inkluderet i dette problem. Dette skyldes, at der er opnået en kombination mellem den her angivne metode og en senere forbedring af identifikationens robusthedsegenskaber.

Udgangspunktet har været metoder beskrevet af Willsky og Jones (/17/ og /18/) samt af Sonalkar og Shen (/13/). Metoderne, der er udviklet til kalmanfiltrering af et lineært

systems tilstande, afviger følgelig en del fra den her beskrevne.

I det følgende specialiseres der til det tilfælde, hvor forstyrrelserne kun er en sekvens af spring overlejret af hvid støj, hvilket med notationen fra kapitel 2 vil sige at:

$$(1) \quad v(t) = C(q^{-1})e_0(t) + d(t)$$

og

$$(2) \quad d(t) = d(t-1) + \delta_2(t)e_2(t)$$

For at lette forholdene fokuseres der kun på et spring, hvorved (2) ændres til:

$$d(t) = \begin{cases} d(t-1) & t < T_j \\ d(t-1) + d_0 & t \geq T_j \end{cases}$$

Springene i $d(t)$ tolkes ved denne metode som deciderede fejl, der skal elimineres. Hermed er der brudt med de mere eller mindre stationære modeller fra de forrige afsnit. Både springets størrelse, d_0 , samt tidspunktet, T_j , på hvilket springet indtræffer er stokastiske variable. Problemet har derfor to dimensioner. En del af problemet er at estimere T_j , et andet er at følge springet, det vil sige at estimere d_0 . I mere generelle problemstillinger med flere fejlkilder vil isolationen af den pågældende fejlkilde udgøre en tredje del. Denne tanke er dog ikke ført videre i dette projekt. I figur 1 er den principielle opbygning af en selvindstillende regulator, der er baseret på denne metode, vist.

Da standardalgoritmen er den bedste overfor stationære forstyrrelser, vil en sådan svare til estimationsfilterets (ID) normale arbejdstilstand. Hvis der forekommer drift kan f. eks. et udvidet kalmanfilter anvendes. En speciel enhed (DET.)

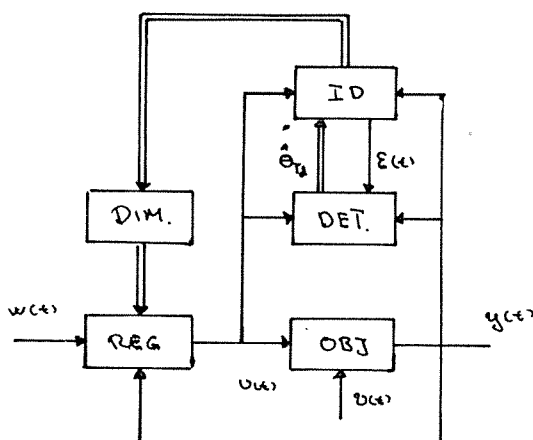


Fig. 1: Struktur i selvindstillende regulator med fejldetektor.

vil på grundlag af residualerne og andre signaler varetage detektionen af springet samt estimationen af d_0 . De øvrige dele, d.v.s. dimensioneringsdelen (DIM.) samt regulatoren med justerbare parameter, er uændret.

Antag nu at den adaptive regulator er i den normale arbejdstilstand for $t < T_j$. Herefter vil residualerne være givet ved:

$$(4) \quad \varepsilon_1(t) = d_0 + e_0(t) \quad \text{for } t \geq T_j$$

for faste og korrekte parameterestimer. Der er her anvendt den generelle estimationsmodel fra kapitel 3. Når parameterne estimeres er residualerne:

$$(5) \quad \varepsilon(t) = \Delta(t, T_j, d_0) + \varepsilon_0(t) \quad \text{for } t \geq T_j$$

hvor $\varepsilon_0(t)$ er residualerne for $d_0 = 0$. Under forudsætning af konvergens, vil disse residualer kort tid efter opstart kunne approximeres med et hvidt normalfordelt støjsignal med middelværdi nul og konstant varians. Hvor hurtigt $\Delta(t, T_j, d_0)$ aftager, er beskrivende for algoritmens evne til at følge springet. Ved metodens anvendelse i forbindelse med kalmanfiltrering af et

systems tilstande, kan denne udregnes a priori på nær den multiplikative d_0 , når systemets (ulinær og tidsafhængig) dynamik er kendt. Selv om estimationsalgoritmen kan fortolkes som et udvidet kalmanfilter kan dette ikke gøres i forbindelse med dette problem. Det skyldes, som det fremgår af (7.3,23-24) og (7.3,26-27), at de indgående elementer er stokastiske. Fremgangsmåden vil derfor afvige kraftigt fra den i /17/ beskrevne. Grundprincippet er dog det samme.

I detektoren skal der foretages et estimat af T_j samt d_0 . Til dette formål kan der anvendes forskellige principper. Lad H_0 og H_1 betegne henholdsvis hypotese og alternativ med definitionen:

$$H_0(T) : \quad \text{intet spring til } t = T$$

og

$$H_1(T) : \quad \text{spring fremkommet til } t = T$$

For hver valgt T ($0 < T < t$) kan der på basis af residualerne $\varepsilon(0), \dots, \varepsilon(t)$ dannes et estimat (f. eks. LS) af d_0 under forudsætning af at $T = T_j$ (d.v.s. $H_1(T)$). På grundlag af dette estimat kan der foretages en likelihood ratio - test. Kaldes teststørrelsen λ , kan beslutningsregelen være følgende simple:

$$\lambda(t, T) \leq L' \quad \text{antages } H_0(T)$$

$$\lambda(t, T) > L' \quad \text{antages } H_1(T)$$

Her er L' (threshold value) valgt således, at der er et rimeligt forhold mellem falske alarmer og oversete spring. Denne metode involverer imidlertid et antal estimationer, der er lineært voksende. Ved implementering vælges derfor et data - vindue af endelig længde, hvorover estimationen af T_j foregår. Et sådant vindue kan være på formen:

$$t - M < T \leq t - N + 1$$

hvor $N \leq M$. Vælges speciel $N = M$, forsvinder optimeringen over T helt, og problemet er reduceret til en test i $T = T_j$

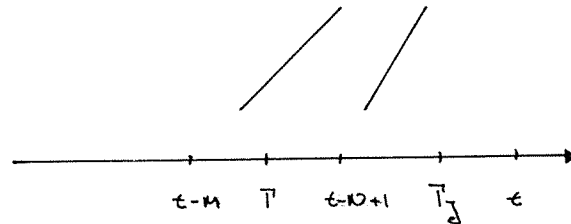


Fig. 2: Illustration af et datavindue.

Herved bliver L' en del lettere at vælge. Til gengæld er sandsynligheden for at overse et spring større.

Antag nu at $N = M$ og at parameterestimererne er konstante og korrekte for $t < T_j$. Da vil størrelserne i (5) være:

$$(6) \quad \varepsilon_0(t) = e_0(t) \quad \Delta(t, T_j, d_0) = d_0$$

Testen i $T = T_j$ kan alternativt formuleres i en test i H_0 kontra H_1 , hvor:

$$H_0: \quad \mu = 0 \quad H_1: \quad \mu \neq 0$$

i dette tilfælde, når μ betegner middelværdien af residualerne. Antages det yderligere at variansen σ_0^2 af $e_0(t)$ er kendt, kan der j.v.f. /4/ udføres en meget simpel test i H_0 mod H_1 ved at undersøge teststørrelsen:

$$(7) \quad Z = \frac{\bar{x}}{\sigma_0} \sqrt{N}$$

der under de givne forudsætninger er en normalfordelt stokastisk variabel med spredning 1 og middelværdi:

$$\sqrt{N} \frac{\mu}{\sigma_0}$$

Hypotesen H_0 forkastes; det vil sige, at den normale arbejdstilstand forlades, såfremt udfaldet, z , af teststørrelsen opfylder uligheden:

$$(8) \quad |z| > g_r u_{1-\frac{\alpha}{2}}$$

hvor u_p betegner p - fraktilen i den normerede normalfordeling. Sandsynligheden P_f for at fejldetektere et spring; det vil sige at forkaste H_0 , når $\mu = 0$, er fastsat via signifikationsniveauet, α , og er dermed en konstant af støjniveauet og horisonten N . Derimod afhænger sandsynligheden P_d for ikke at detektere et givet spring både af horisontens længde og støjsspredningen samt naturligvis af springets størrelse. Disse forhold kan undersøges ved en analyse af testens styrkefunktion $p(d_0)$,

$$(9) \quad p(d_0) = 1 - \Phi \left\{ g_r - \sqrt{N} \frac{d_0}{\sigma_0} \right\} + \Phi \left\{ -g_r - \sqrt{N} \frac{d_0}{\sigma_0} \right\}$$

hvor Φ betegner fordelingsfunktionen for den normerede normalfordeling. Styrkefunktionen er skitseret i fig. 3 for følgende værdier af N : 2, 5, 10 og 25 for et signifikationsniveau på 1 o/oo.

Selve detektoren har øjensynligt to parameter, der skal vælges. Dette kan ske ud fra bl. a. P_d og P_f . Alt efter den aktion der er knyttet til H_1 , vil det give anledning til ekstra tab, hvis P_f er uforholdsmæssig stor. Det skønnes, at et signifikationsniveau, $\alpha = P_f$, på 0.001 er passende.

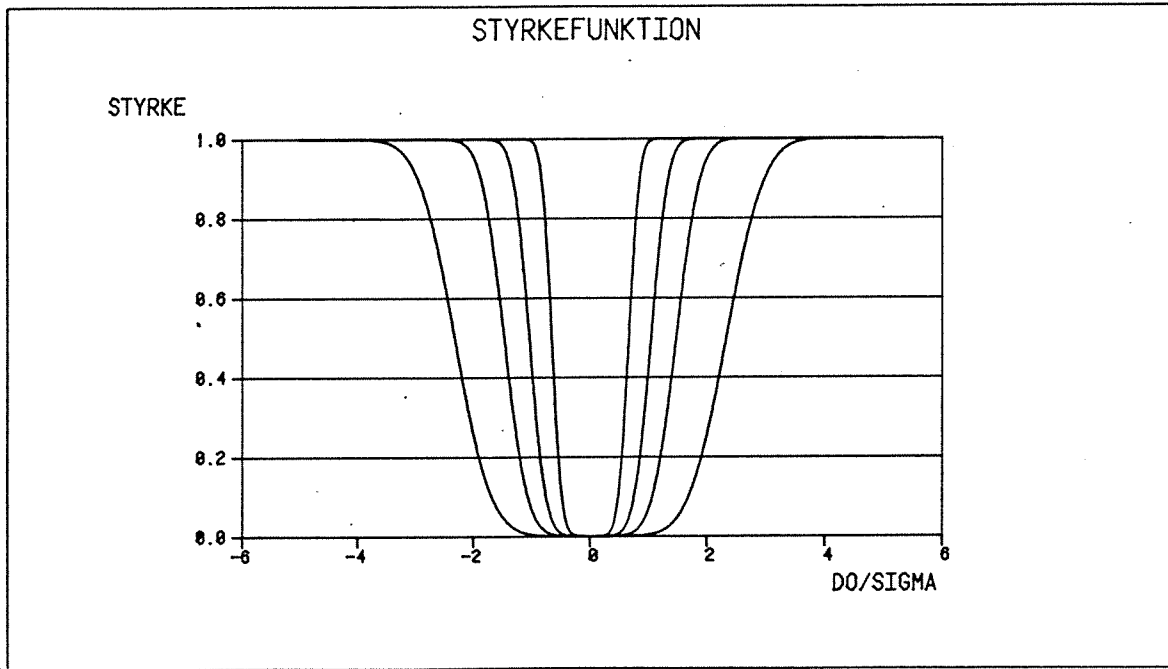


Fig. 3: Styrkefunktionen for testen med en horisont på 2,5,10 og 25 og $\alpha = 0.001$.

Horisontens længde, N , kan fastsættes ud fra et krav til P_d . Et krav om at P_d skal være meget lille, vil være i modstrid med et ønske om hurtig detektion. Dette dilemma er dog ikke tilnærmelig så uomgængelig som det problem, der skyldes den manglende separation mellem estimationspræcision og springshastighed. Er der fremkommet et spring som anskueliggjort på fig. 2, således at $T_1 > N$, kan dette, hvis d_0 er tilstrækkelig stor, blive detekteret før $t = T_1 + N - 1$. I praksis vil dette være det mest hyppige tilfælde. Konklusionen på afsnit 7.1 var blandt andet, at standard algoritmerne havde mulighed for at optage mindre spring. Ved at anvende f. eks. en UKF tilføjelse til estimationsalgoritmen (afsnit 7.3) kan der opnås en stationær forstærkning til estimatet af d (eller γ). Der kan desuden opnås, at spring af samme størrelsesorden som

σ_0 fint kan absorberes i estimatet af d . Der vil derfor ikke være nogen begrundelse for at vælge N så stor, at styrkefunktion er meget stor omkring $\frac{1}{2}\sigma_0$. Hvis den normale arbejdstilstand udgøres af den i afsnit 7.3 beskrevne UKF - algoritme med en lille værdi for Q_d , vil denne både kunne absorbere drift og små spring, som detektoren ikke reagerer overfor. Vælges her $N = 10$ vil styrken under forudsætningerne beskrevet ved (6) i σ_0 og i $2\sigma_0$ være henholdsvis 0.4483 og 0.9988, når $\alpha = 0.001$. Dette svarer til, at P_d er 55,2 % og 0.12 % for spring med størrelserne σ_0 og $2\sigma_0$. Det påpeges, at der i afsnit 7.3 er vist ret gode resultater for en UKF-algoritme med en MVo - regulator, når springet var over $3\sigma_0$. For deadbeatregulatoren er forholdene ikke ligeså gode.

For at lette beskrivelsen var det forudsat, at residualerne, $\varepsilon_0(t)$, for $d_0 = 0$ er et hvidt normalfordelt støjsignal med konstant spredning og middelværdi 0. Der var desuden forudsat, at $\Delta(t, T_j, d_0)$ er konstant, hvilket naturligvis specielt ikke er tilfældet, når der anvendes en UKF - algoritme. I fig. 7.3, 6, 8, 9 og 12 er angivet nogle kurver for en simulation med en MVo - regulator anvendt på objekt I. Estimationen er foretaget med et udvidet kalmanfilter med $Q_d = 0.005$. For $t = 40$ indtræffer et spring på 1 ($3.16 \sigma_0$), der bliver udkompenseret på relativ kort tid. I fig. 7.3, 8 ses, at de store tab umiddelbart efter springet skyldes DC - fejl. Det bemærkes, at tabstilvæksten fra $t = 35$ til $t = 40$ er relativ stor i forhold til de efterfølgende intervaller, når det erindres at $T_j = 40$. Kvalitativt kan denne realisation af Δ beskrives som et eksponentielt henfald med en tidskonstant på ca. 6. Den pågældende forudsætning om, at Δ er konstant, kan ikke siges at holde med en rimelig tilnærmelse, når $N = 10$. De anførte overvejelser kan dog holde såfremt d_0 substitueres med tidsmiddelværdien af Δ taget over N trin. Hvis denne tidsmiddelværdi er meget lille er det kun et tegn på, at algoritmen kan absorbere springet, forudsat at det hovedsageligt er estimatet af d , der ændres.

Forudsætning om at $\varepsilon_o(t) = e_o(t)$, gælder ikke i teoretisk forstand, idet $\hat{\theta}(t)$ er tidsvariende. Specielt i startperioden, hvor grovestimationen foregår, varierer $\hat{\theta}(t)$ meget, og $\varepsilon(t)$ er bestemt ikke stationær her. Under forudsætning af konvergens vil $\hat{\theta}(t)$ være langsomt varierende efter opstartsfasen, hvorved $\varepsilon(t)$ approximativt er stationær. I de simulationer der er foretaget i forbindelse med kapitel 6 antager tabet af residualerne meget hurtigt en hældning, der indikerer at residualerne har samme varians som $e_o(t)$. I de fleste simulationer er tilfældet efter ca. 10 trin. Det samme gælder om kurverne for AC - forstærkningen af $H_{e,e}$, der ligeledes efter nogle få trin antager den forventede værdi på 1. Da den korrekte overføringsfunktion fra støjen til residualerne gennem den inverse model er 1, antyder det ovenstående, at residualerne konvergerer mod støjsignalet, $e_o(t)$, der er hvid. Man kan naturligvis ikke forvente den samme høje konvergensthastighed for et vilkårligt objekt.

Den ovenstående diskussion har været baseret på den generelle estimationsmodel fra kapitel 3, hvor residualerne er lig $e_o(t)$ for korrekte estimater under forudsætning af, at der er valgt en algoritme, der er egnet til den foreliggende situation (f. eks. RELS eller RML hvis $C \neq 1$). De modeller, der anvendes til at estimere regulatorparameterne i den genrelliserede minimalvarians- og polplaceringsregulator afviger fra den generelle model ved at $\underline{A} = 1$. Dette bevirker at regressorene er forsinket k tidsenheder, og man kan derfor tillade, at residualerne er en glidende middelværdi af formen:

$$(10) \quad \varepsilon_o(t) = E(q^{-1}) e_o(t)$$

for korrekte estimater. Polynomiet E har ordenen $k-1$, hvor k er tidsforsinkelsen gennem objektet. Det er imidlertid ret uheldigt for detektoren, at residualerne er korreleret, fordi testen forudsætter, at de er uafhængige. Simulationer viser, at hvis

der ikke tages forholdsregler mod dette, vil hyppigheden af fejldetektioner stige. I simulationerne kunne der ikke spores en meget markant forøgelse af tabene. Dette vil afhænge af den aktion der udløses ved en forkastelse af H_0 . En metode til at omgå dette problem er at estimere parameterne i E - polynomiet, når tidsforsinkelsen, k , er større end 1. Dette kan f. eks. gøres med en RELS - eller en RML - algoritme. Der vil dog gå længere tid inden forholdene bliver stationære. Erfaringen viser, at parameterne i støjmodellen (polynomiet C i kapitel 3, der svarer til E - polynomiet i afsnit 5.1 og 5.3) er ret langsomme til at konvergere mod de korrekte. Når E - polynomiet estimeres vil residualerne også være approximativt uafhængige for $k \neq 1$ ved anvendelse af en generaliseret minimalvarians - og en polplaceringsregulator.

Forudsætningen om, at spredningen, σ_0 , er kendt for støjsignalet, $e_0(t)$, som residualerne om muligt konvergerer imod i det stationære tilfælde, vil nu blive genstand for opmærksomheden. Et kendskab til denne størrelse kan eksistere i et givet problem. Ved adaptiv regulering er det i modstrid med grundtanken at forudsætte et sådant kendskab, og desuden er det vanskeligt at få et a priori estimat af denne størrelse, når objektets dynamik er ukendt. Alternativet er at estimere variansen på $\varepsilon_0(t)$, når dette kan approximeres med et hvidt stationært signal. Problemet er blot at vide, hvornår de tilgængelige estimater, $\varepsilon(t)$, kan opfattes som stationære i en tilnærmet betydning indenfor det datavindue, der anvendes til estimationen af σ_0 .

Udgangspunktet for estimationen af variansen for de approximativt stationære residualer, $\varepsilon_0(t)$, er en omskrivning af en LS - estimator Q_e for en sekvens af uafhængig ensfordelte stokastiske variable f. eks. $e_0(t)$. Denne estimator, hvor:

$$(12) \quad Q_e(t) = \frac{1}{t+1} \sum_{i=0}^t e_0(i)^2 \quad (E\{e(t)\} = 0)$$

kan omskrives til rekursiv form:

$$\begin{aligned} (13) \quad Q_e(t+1) &= Q_e(t) \left(1 - \frac{1}{t} \right) + \frac{1}{t} e_o^2(t) \\ Q_e(0) &= e_o^2(0) \end{aligned}$$

Der er to typer perioder i forløbet af parameterestimationen, hvor approximationen til stationærtilstand er temmelig dårlig. Den ene er opstarten, hvor grovestimationen af parameterne finder sted. Den anden type periode er umiddelbart efter et spring. Hvor lange disse perioder er, afhænger af mange ting. Heriblandt er konvergenshastigheden for den adaptive regulator samt hvilken aktion en detektion af et spring initierer samt i hvor grad algoritmen i dens unormale arbejdstilstand er i stand til at spore variationen i $d(t)$.

Med hensyn til opstartperioden findes der flere måder at gribe problemet an på. En variant er simpelthen at vente et på forhånd valgt tidsrum, og først derefter anvende detektoren, der blandt andet foretager en estimation af støjvariansen på grundlag af de tilgængelige residualer. Argumentet for denne metode er, at testens forudsætninger slet ikke er opfyldt i startperioden. Her vil problemet blive grebet lidt anderledes an, fordi der ellers vil gå en temmelig lang tid, inden der kan foretages en rimelig sikker test. Variansen på estimatoren Q_e er omvendt proportional med antallet af observationer. Hvis estimationen af σ_o^2 startes sammen med den øvrige del af regulatoren, vil udfaldene være yderst fejlagtige på grund af store residualer i startfasen. Estimatoren må derfor arbejde med eksponential glemsel. Hvis estimatoren $Q(t)$ opdateres efter:

$$\begin{aligned} (14) \quad Q(t+1) &= Q(t) \lambda_q(t) \left\{ 1 - \frac{1}{t} \right\} \\ &+ \left\{ 1 - \lambda_q(t) + \frac{1}{t} \right\} \varepsilon^2(t) \end{aligned}$$

hvor

$$\lambda_q(t+1) = \lambda_0^q \lambda_q(t) + \lambda_1^q (1 - \lambda_0^q)$$

$$(15) \quad \lambda_0^q = 0.97 \quad \lambda(0) = 0.8$$

$$\lambda_1^q = 0.985$$

vil de forkerte startværdier af $Q(t)$ hurtigt blive glemt, fordi tidshorizonten for dette estimat langsomt forøges fra ca. 40 til 150 i løbet af 200 tidsenheder. Dette indebærer blandt andet, at detektoren kan anvendes ved problemer, hvor variansen, σ_0^2 , af støjsignalet, $e_0(t)$, er tidsvarierende. Anvendes detektoren alene skal disse tidsvariationer være så langsomme, at forholdene tilnærmet er stationære indenfor tidshorizonten for estimatoren $Q(t)$. I et efterfølgende afsnit vil de hurtigere variationer i σ blive behandlet. Der vil der blive angivet udvidelser til at inkludere de hurtige variationer i σ .

Der er ikke i (14) taget hensyn til problemer i forbindelse med den anden type periode, hvori der ikke tilnærmet gælder stationaritet. Uligheden (8) kan med de omtalte tilnærmelser omformes til:

$$(16) \quad |S(t)| > g_r(t) \sqrt{NQ(t)} = C_1(t)$$

hvor $S(t)$ er en sum over de sidste N tilgængelige residualer.

$$(17) \quad S(t) = \frac{1 - q^{1-N}}{1 - q^{-1}} \varepsilon(t)$$

Efter et detekteret spring nulstilles alle tilstandene i filteret fra (17). Der er en risiko for, at et spring kan løfte estimatet $Q(t)$ sådan, at springet ikke bliver detekteret. Dette problem kan undgås ved at forsinke residualerne i (14) med N trin, hvorved korrelationen mellem $S(t)$ og $Q(t)$ fjernes,

såfremt $\varepsilon(t)$ er hvid. Da residualerne, i tidsrummet mellem et spring opstår, T_j , og det bliver detekteret, T_d , ikke er stationære, kan disse ikke anvendes til estimationen af støj - variansen. Med den omtalte forsinkning på N trin af residualerne kan springets indflydelse på $Q(t)$ fjernes ved at standse opdateringen af $Q(t)$ i mindst N tidsenheder efter T_d . Såfremt den af detektoren initierede aktion ikke kan spore springet helt perfekt, kan der vælges en standsning af længere varighed. På grund af disse forsinkelser og standsninger kan tiden ikke anvendes som tæller. Der må anvendes en separat tæller, $I(t)$. Den endelige estimator lyder:

$$(18) \quad Q(t+1) = \begin{cases} Q(t) \lambda_q(t) \left\{ 1 - \frac{1}{I(t)} \right\} + \varepsilon^2(t) \left\{ 1 - \lambda_q(t) + \frac{1}{I(t)} \right\} & \text{for } t \leq N \\ Q(t) & \text{for } \begin{cases} N < t \leq 2N \\ T_{d,i} \leq t \leq T_{d,i} + N \end{cases} \\ Q(t) \lambda_q(t) \left\{ 1 - \frac{1}{I(t)} \right\} + \varepsilon^2(t-k) \left\{ 1 - \lambda_q(t) + \frac{1}{I(t)} \right\} & \text{ellers} \end{cases}$$

Her betegner $T_{d,i}$ de tidspunkter, hvor der er detekteret et spring. Glemselsfaktoren opdateres efter (14) mens

$$(19) \quad I(t+1) = \begin{cases} I(t) & \text{for } N < t < 2N \text{ og } T_{d,i} < t < T_{d,i} + N \\ I(t) + 1 & \text{ellers} \end{cases}$$

Valget af $Q(0)$ er her uden betydning, idet $I(0)=1$, men i de senere afsnit vil det være nødvendigt at vælge $Q(0)$ til et meget stort tal. Det er en erfaring, at residualernes størrelse i de fleste tilfælde er meget store i startøjeblikket og derefter er aftagende til en værdi, der svarer til $\varepsilon(t) \approx e_0(t)$. Den opdatering, der er angivet ved (18), vil bevirke, at hvis objektet bliver udsat for ekstremt store spring under opstarten, kan disse blive detekteret, selvom de indløber før $t = N$.

Når der i uligheden, (16), anvendes estimatet, $Q(t)$, kan der ikke anvendes værdien $u_{1-\frac{1}{2}\alpha}$ for $g_r(t)$. Asymptotisk er horisonten, β , for estimatoren $Q(t)$ bestemt af λ_1^q ved:

$$(20) \quad (\lambda_1^q)^\beta = 0.1$$

hvormed $g_r(t)$ asymptotisk skal være:

$$g_r(t) \rightarrow t(\beta-1)_{1-\frac{1}{2}\alpha}$$

Egentlig kunne $g_r(t)$ følge tidsvariationen af $\min(\beta, t)$, men her er der valgt at modellere $g_r(t)$ med et eksponentielt henfald

$$(21) \quad g_r(t+1) = g_r(t)g_r^0 + g_r^1(1-g_r^0)$$

med talværdierne:

$$g_1^0 = 0.75 \quad g_r(0) = 10$$

$$g_r^1 = t(150)_{0.9995} = 3.39$$

Ved valget af talværdierne er der lagt størst vægt på det asymptotiske forløb. Den endelige detektor er vist i fig. 4 i den normale arbejdstilstand.

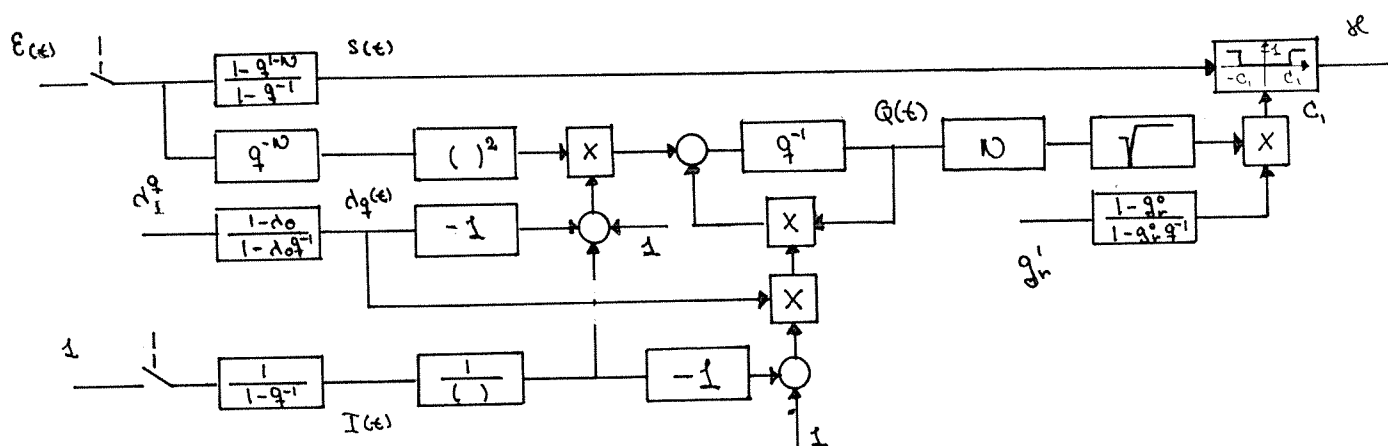


Fig. 4: Detektor i normale arbejdstilstand.

Til detektionen kunne der anvendes andre test som f. eks. en χ^2 - test. De aktioner, som et detekteret spring giver anledning til, udbyder en lige så rig variationsmulighed. Den simpleste løsning er en genstartning af parameterestimationen i det resulterende punkt, $\hat{\theta}(T_{d,i})$. Denne løsning giver dog nogle mindre tab på grund af variationer i $\hat{d}(t)$ (eller γ) samt de fejlagtige estimater af de dynamiske parameter, der skyldes springets påvirkning. Denne metode kan udbygges på mange måder, f. eks. kan der til $\hat{d}(T_d)$ lægges $S(T_d)/N$ og der kan startes i et punkt givet ved $\hat{\theta}(T_d-N)$. Den endelige metode, der er beskrevet nedenfor, udviser i modsætning til disse en god øyne til at klare spring, der indløber med en tidsafstand lidt større end N . Denne afstand er ret uheldig for de nævnte metoder.

For metoden, der er beskrevet i afsnit 7.3, blev der fundet,

at et udvidet kalmanfilter med en meget høj værdi for Q_d kunne spore selv temmeligt store spring. Uheldigvis for denne metode kunne et højt Q_d ikke opretholdes i længere tid, da parameter - estimerne ellers blev fejlagtige på grund af bias. Ud fra den betragtning at man ikke på tidspunktet kan afgøre om der er sket et spring, kan der derfor startes to udvidede kalmanfiltre. Det ene filter arbejder under hypotesen H_0 , d.v.s. i den normale arbejdstilstand, og har derfor en meget lav værdi for Q_d . Det andet filter arbejder under hypotesen H_1 med en høj værdi for Q_d for at spore springet. Efter N tidsenheder afgøres det ved hjælp af testen hvilken af de to hypoteser, der er den mest sandsynlige. Da testen kun anvender residualerne fra filteret, der arbejder under H_0 , behøves der ikke en realisation af de ekstra $N-1$ filtre, men kun en start af et filter i et punkt, der er passeret til $t = T_d - N + 1$, hvis et spring er detekteret. Dette filter arbejder med et højt Q_d fra start og til T_d , hvorefter det overgår til den normale arbejdstilstand. I fig. 5 er der for $N = 3$ skitseret multiple spring.

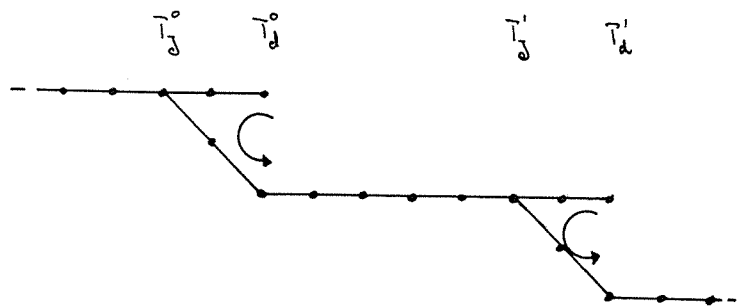


Fig. 5: Skitse af beslutningstræet for den anvendte metode.

hvor en skrå linie symboliserer, at det udvidede kalmanfilter

arbejder under hypotesen H_1 . Når de $N-1$ ekstra filtre ikke realiseres kræves der dog, at tilstandene i det udvidede kalman-filter (eller standalgoritmen) bliver lagret. Det vil med notationen fra kap. 3 sige at signalerne $y(t-1)$, $\varepsilon(t-1)$, vektorerne $x(t)$, $z(t)$ og $\hat{\theta}(t)$ samt matricen $P(t)$ skal lagres. Desuden skal alle indgående signaler til estimationsalgoritmen lagres fra $t-N+1$ til t .

I fig. 5 er der skitseret tilfældet $T_d = T_j + N - 1$. Dette er sjældent forekommende. Normalt er T_d mindre, alt afhængig af, hvor stort springet er. Dette betyder at tabene, der skyldes at springet ikke bliver udkompenseret, med det samme bliver mindre.

En mere tidskrævende metode, der ikke involverer et estimat af variansen, er at realisere alle N filtre og ved hvert tids - punkt sammenligne tabet af hver filtergren. Den filtergren med størst tab vil blive afsluttet og den gren med mindst tab vil blive kløftet. De samme to spring vil give anledning til et beslutningstræ, som er skitseret i fig. 6

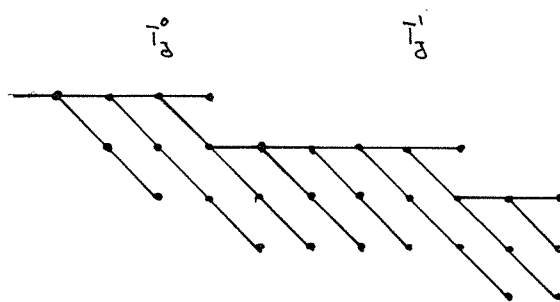


Fig. 6: Beslutningstræ for alternativ metode.

Der kan tænkes vanskeligheder når der skal vælges mellem to

filtre, der er startet i henholdsvis T_{j-1} og T_j . Problemet er nok af mindre betydning, idet begge filtre er i stand til at følge springet. Denne alternative metode er ikke undersøgt i dette projekt og i det følgende vil den forrige metode blive afprøvet.

Der vil nu blive kommenteret nogle simulationer foretaget med den beskrevne metode. Ved simulationerne er talværdierne

$$N = 10$$
$$\alpha = 0.001$$
$$Q_d = \begin{cases} Q_d^1 = 100 & \text{under } H_1 \\ Q_d^0 = 0.005 & \text{under } H_0 \end{cases}$$

anvendt for detektoren. For at have et sammenligningsgrundlag er de viste kurver optaget for objekt I reguleret af en MVo - regulator. De øvrige forhold er uændret og beskrevet i afsnit 7.1 og 6.1.1.

Spring.

I figurene 7 - 16 er vist kurverne for en simulation, når der forekommer et spring i $d(t)$ til $t = T_j = 40$. Signalerne i fig. 7 viser en ret hurtig udkompensering af springet som en følge af at springet detekteres tidligt ($T_d = 42$). Dette giver sig yderligere til udtryk i figurene 8 og 9, der viser tidsforløbet af tabene og varianserne for signalerne. Disse kurver kan sammenlignes med fig. 7.3,6 - 7 og fig. 7.2,7 - 8. En sådan sammenligning viser, at der med denne metode er opnået en markant forbedring. Dette kan gøres kvantitativt ved at sammenholde tabene for styre - og udgangssignalet med andre løsninger.

	tab(y)	tab(u)
UKF ($Q_d=0.005$)	774.3	774.4
Detektor	716.0	758.2

Ved simulationer foretaget uden spring i $d(t)$ kunne følgende tab måles:

	tab(y)	tab(u)
standard	666.1	694.3
UKF, detektor.	677.8	726.5

For udgangssignalet er der vundet 7.6%, mens styretabet kun er forbedret 2.1% i forhold til anvendelse af metoden beskrevet i afsnit 7.3 (UKF), der svarer til at lade den her beskrevne algoritme blive i den normale arbejdstilstand. I de ovenstående tabeller indgår desuden starttabene, der giver et ikke negligibelt bidrag. Til sammenligning er der angivet tabene for simulationer uden spring, der selvfølgelig giver langt mindre tab. Det skal erindres at det udvidede kalmanfilter i forbindelse med MVo - regulering har en rimelig evne overfor spring af denne størrelsesorden. Ved større spring vil denne detektor metode være mere overlegen, hvilket blandt andet skyldes at springene detekteres tidligere. I nedenstående tabel er angivet tabene (fig. 10) for en simulation med en detektor, hvor den normale arbejdstilstand udgøres af en standardalgoritme ($Q_d^0 = 0$)

	tab(y)	tab(u)
detektor ($Q_d=0$)	717.1	735.7

Denne simulation adskiller sig ikke fra tilfældet med $Q_d = 0.005$ ved detektionen af springet, men ved at fejldetektere et spring, der ikke forekommer, til $t = 273$. Detektoren anvender et signifikationsniveau på $\alpha = P_f = 0.001$. At anvende en værdi for Q_d^0 der er forskellig fra nul begrænser detektorens følsomhed overfor mindre variationer i tidsmiddelværdien (over N tidsenheder) af residualerne, der skyldes tilfældigheder i den faktiske støjrealisation eller mindre spring, som det udvidede kalman -

filter kan optage i den normale arbejdstilstand. Hvis man ser bort fra disse fejldetektioner, kan der opnås lavere tab j.v.f. afsnit 7.3 med $Q_d^0 = 0$, men hermed afskrives muligheden for at optage drift og mindre spring i $d(t)$.

I fig 11 er forløbet af AC - forstærkningerne angivet med en tidsopløsning, der er større end i de forrige afsnit. Simulationen er ligesom i det følgende foretaget med et spring til $t = T_j = 40$ i $d(t)$ på 1. Her ses det, at overføringsfunktionen til udgangen er upåvirket indenfor kurvens opløselighed af springet, mens $H_{u,e}$ kun påvirkes i tidsrummet, imellem hvilket springet opstår, T_j , og hvor det bliver detekteret, T_d . I fig. 11 a er tidsforløbet af grænsen,

$$C_1(t) = g_r(t) \sqrt{N Q(t)},$$

angivet (THRESH,*). Det bemærkes, at hvis springet var fremkommet tidligere skulle det være kraftigere for at blive detekteret lige så hurtigt som i dette tilfælde. Algoritmens evne til at optage udetekterede spring afhænger blandet af størrelserne af elementerne i parameterestimaternes kovariansmatrice $P(t)$. Selv om disse er store i starten er det svært at sige noget generelt om hvorledes spring i startperioden påvirker algoritmen, når disse ikke bliver detekteret. Hvis dette er et problem, kan man forsyne estimationen med en glemselsfaktor, der hurtigt svinger ind mod en asymptoteværdi på λ_1 .

Til sammenligningsformål er forløbet af DC - fejlen angivet i både stor (fig. 13) og lille (fig. 12) tidsopløselighed. Fig. 12 kan sammenlignes med fig. 7.3,9 (UKF) og fig. 7.3,6 (glemselsfaktor). Forbedringen er tydelig, når man ser bort fra indflydelsen fra $Q_d^0 \neq 0$, der ligesom i UKF - tilfældet giver anledning til fluktuationer i estimatet af $d(t)$. Både i fig. 12 og i fig. 7.3,9 kan det fænomen i støjrealisationen, der udløste en fejldetektion i tilfældet $Q_d^0 = 0$. I fig. 13 kan tids-

forløbet af DC - fejlen iagttages i detaljer. På grund af den temmelig store værdi for Q_d^1 , der anvendes i den unormale arbejdstilstand, har estimatet af $d(t)$ (eller $\gamma(t)$) et mindre oversving, der forsvinder i de almindelige fluktuationer efter en tidsenhed.

I den adaptive MVo - regulators model af objektet har den generelle modelparameter \underline{d} betegnelsen γ . Tidsforløbet af estimatet for denne er angivet i figur 16. Som det fremgår af fig. 12 og 13 fluktuerer denne en del som følge af at $Q_d^0 = 0.005$. Tidsforløbet af de øvrige parameterestimer er sammenlignelige med de tilsvarende i afsnit 6.1.1 (fig. 6.1.1,11 og 12), der er optaget med en standardalgoritme og med stationær hvid støj.

I fig. 17 er vist signalerne fra en tilsvarende støjfri simulation. Dog er referencesignalet ændret til en firkant - bølge. Her bliver springet detekteret i samme øjeblik det op - står, hvilket giver sig til udtryk ved den efterfølgende regulering.

Drift.

Den rampefunktion, der er anvendt til simpelt at simulere en drift, er i de foretagne simulationer med regulatoralgoritmen blevet absorberet af det udvidede kalmanfilter ($Q_d^0 = 0.005$), hvorved der ikke er udløst nogen detektion af et spring. Dette antyder, at løsningen, bestående af en detektor med tilhørende unormal arbejdstilstand til at følge de abrupte ændringer og et udvidet kalmanfilter med en meget lille forstærkning til at følge drift og estimere de øvrige parameter i modellen, er god.

Spidser.

Spidser havde en ret uheldig indvirkning på standardalgoritmerne og på det udvidede kalmanfilter. Dette vil derfor også være tilfældet, hvis der indtræffer en spids, når denne algoritme er i den normale arbejdstilstand. I de andre tilfælde var problemet, at regulatoren styrede med forkert estimat af $d(t)$ gennem længere tid. De øvrige faldt, hvis spidsen var stor nok, relativt hurtigt på plads som følge af eksitationen. I dette tilfælde er problemet, at spidser udløser detektoren prompte og i den unormale arbejdstilstand forsøger det udvidede kalmanfilter at følge spidsen. Ved næste sampling har estimatet af $d(t)$ en fejlagtig værdi og har været anvendt til styringen. På grund af det fejlagtige estimat bliver residualerne her af samme størrelsesorden som spidsen, hvis denne ikke er større end at det udvidede kalmanfilter kan følge den. Detektoren bliver derfor aktiveret igen for at korrigere estimerne. Det er derfor interessant at se, om den efterfølgende regulering lider skade, efter at algoritmen har forsøgt at følge opover en sådan spids. I fig 18 og 19 er der vist tidsforløbet af DC - fejlen og AC - forstærkningen for det tilfælde, at et spring opstår i $t = 40$, og en spids på 7 fremkommer i $e(t)$ for $t = 41$. Det er indres, at springet under normale omstændigheder først detekteres til $t = 42$. Detektoren er aktiveret i $t = 41$ og $t = 42$, hvilket skyldes spidsen. De to figurer viser, at den efterfølgende regulering ikke har lidt skade, fordi spidsen er blevet opfanget i estimatet af $d(t)$ og ikke i de øvrige parameterestimer. I dette tilfælde bliver de dynamiske parameter som følge af eksitationen korrigeret, hvilket fremgår af fig. 7.6,19. Selv om den efterfølgende regulering ikke her er spoleret vil tidsforløbet af estimatet for $d(t)$ (eller $y(t)$) i uheldige tilfælde give anledning til større tab. I de næste afsnit vil en metode til at undgå dette blive undersøgt.

Springprocessen.

I det foregående blev den samlede proces, der er beskrevet i kapitel 2, simuleret ved hjælp af en rampe - , en trinfunktion samt en enkelt spids. Nu vil denne algoritme med de angivne talværdier blive testen overfor en proces på formen:

$$v(t) = C(q^{-1})e_0(t) + d(t)$$

$$d(t) = d(t-1) + \delta_2 e_2(t)$$

Ved denne simulation er $e_2(t)$ en sekvens af uafhængige normalfordelte stokastiske variable med middelværdi 0 og varians σ_2^2 . Som nævnt i kapitel 2 er det underordnet, hvilken fordeling springstørrelserne har. Processen er simuleret med følgende talværdier:

$$\sigma_2 = 5.0$$

$$\pi_2 = \begin{bmatrix} 0.99 & 0.01 \\ 0.99 & 0.01 \end{bmatrix}$$

og den anvendte realisation af belastningen er afbildet i fig. 20. I nedenstående tabel er forløbet af denne realisation kvantitativt præsenteret.

T_1	-	31	241	273	275	293
T_d	-	35	247	-	277	294
d_0	-	1.4	-0.77	-0.39	2.7	4.25
$d(t)$	o	1.4	0.63	0.24	2.94	7.19

Samtidigt er detektionstidspunkterne angivet. I figur 21 og 22

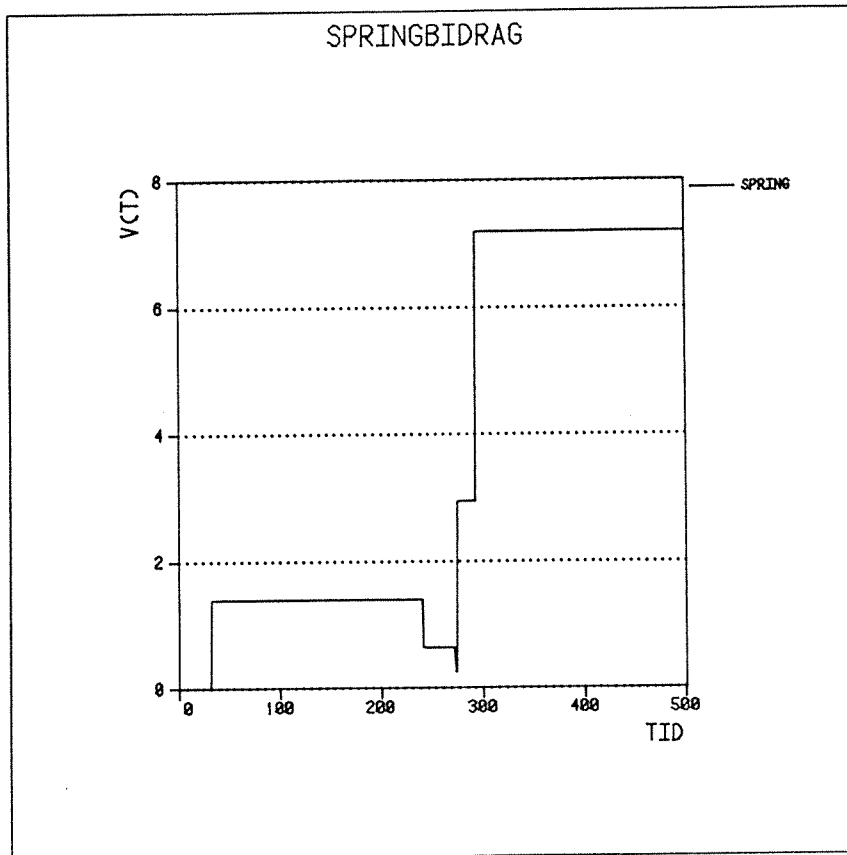
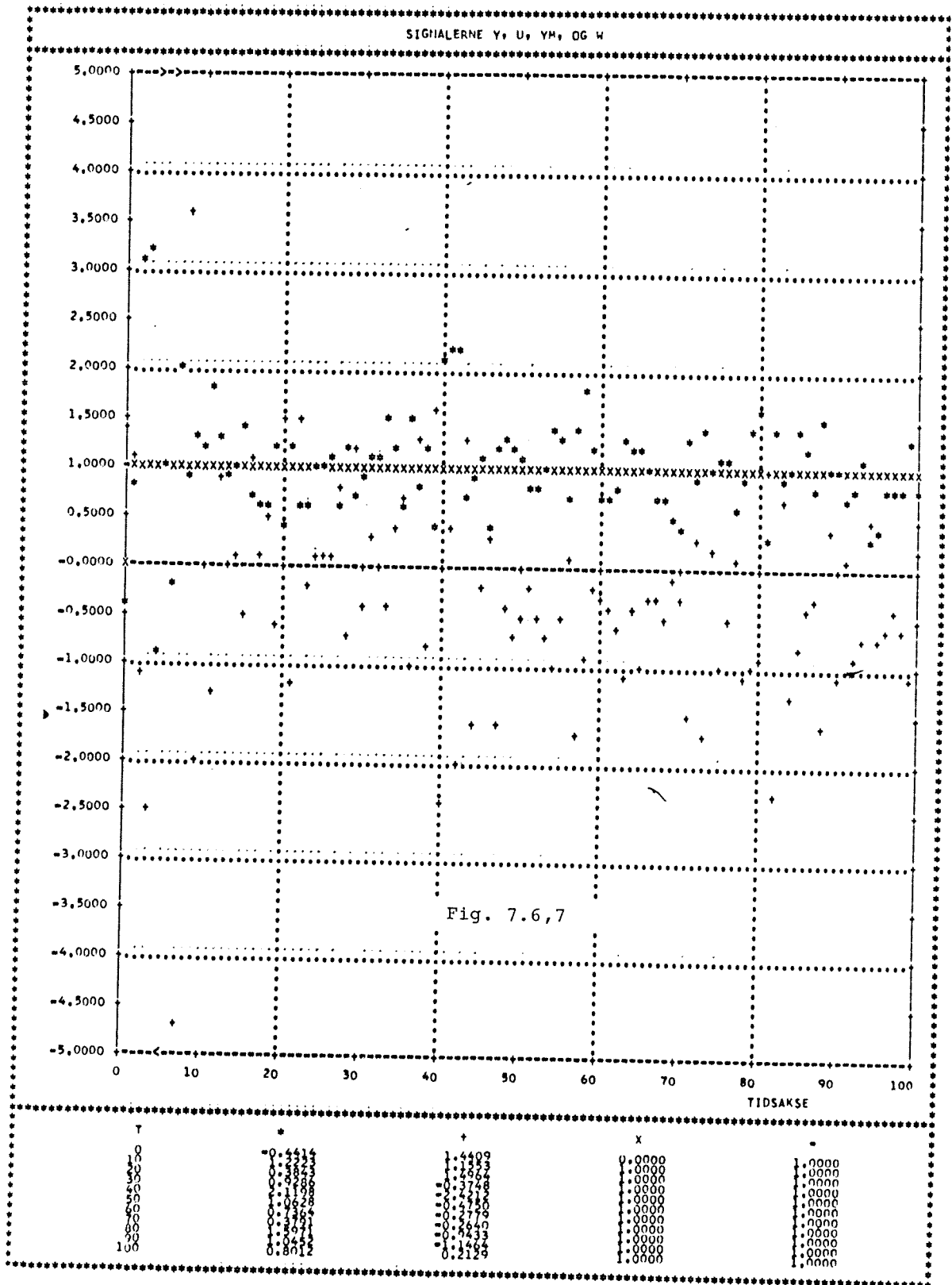
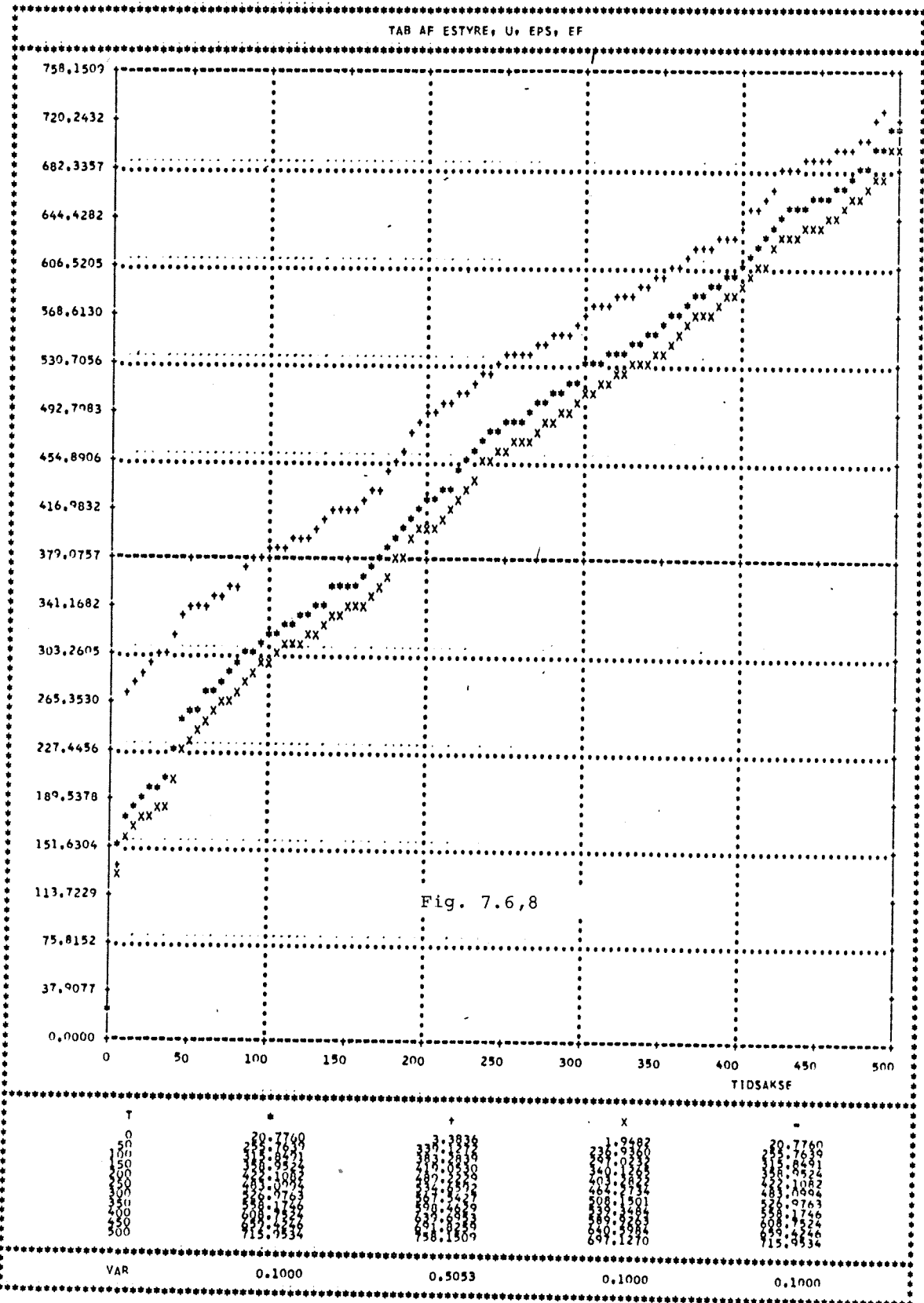


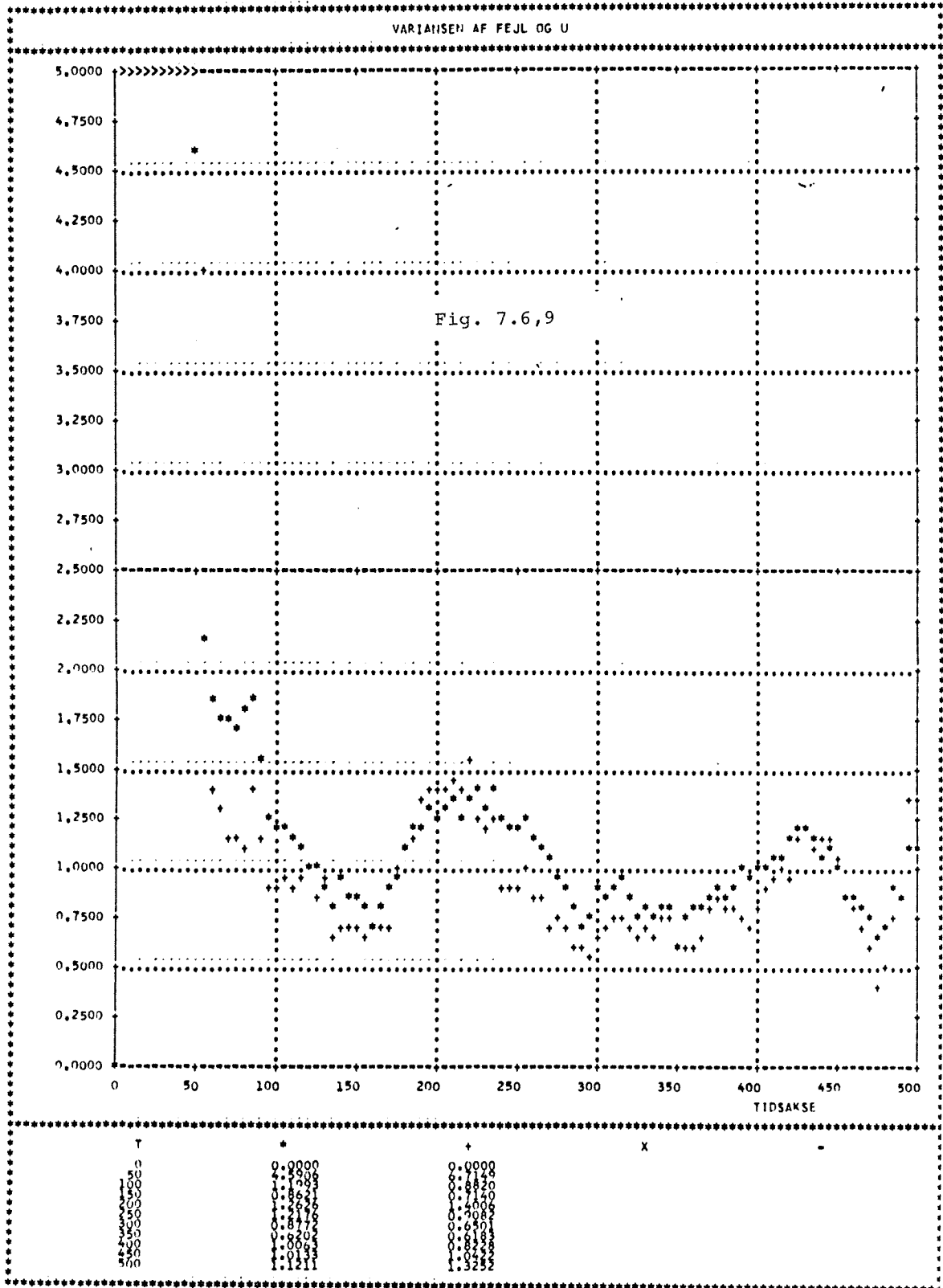
Fig. 2o: Den anvendte belastnings realisation

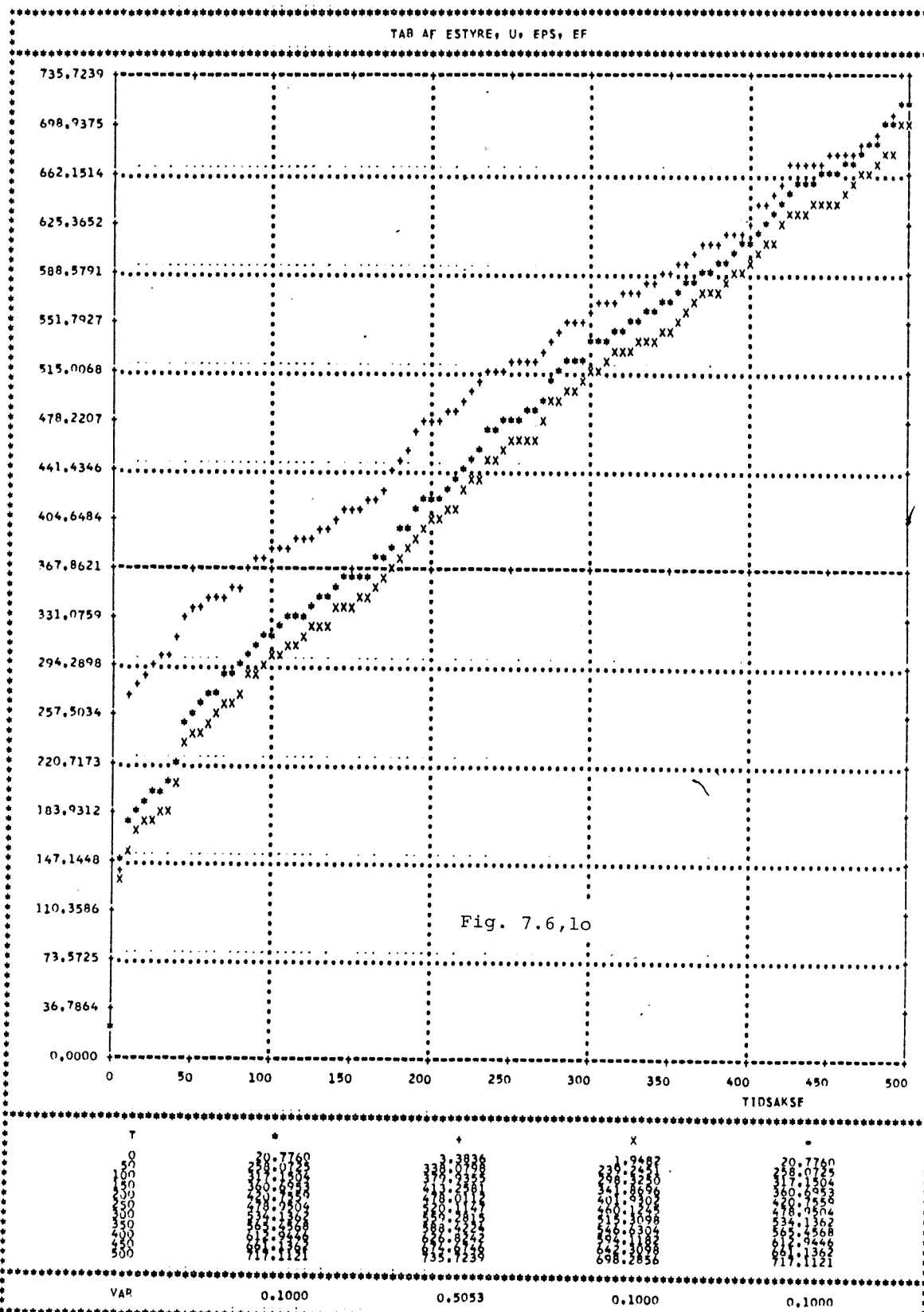
ses, at springenes indvirkning på parameterestimerne er reduceret på en effektiv måde. Figur 23 viser signalerne i tidsrummet fra $t = 230$ til 330 , hvor det fremgår, at de her forekommende spring udkompenseres kort tid efter deres opstående.

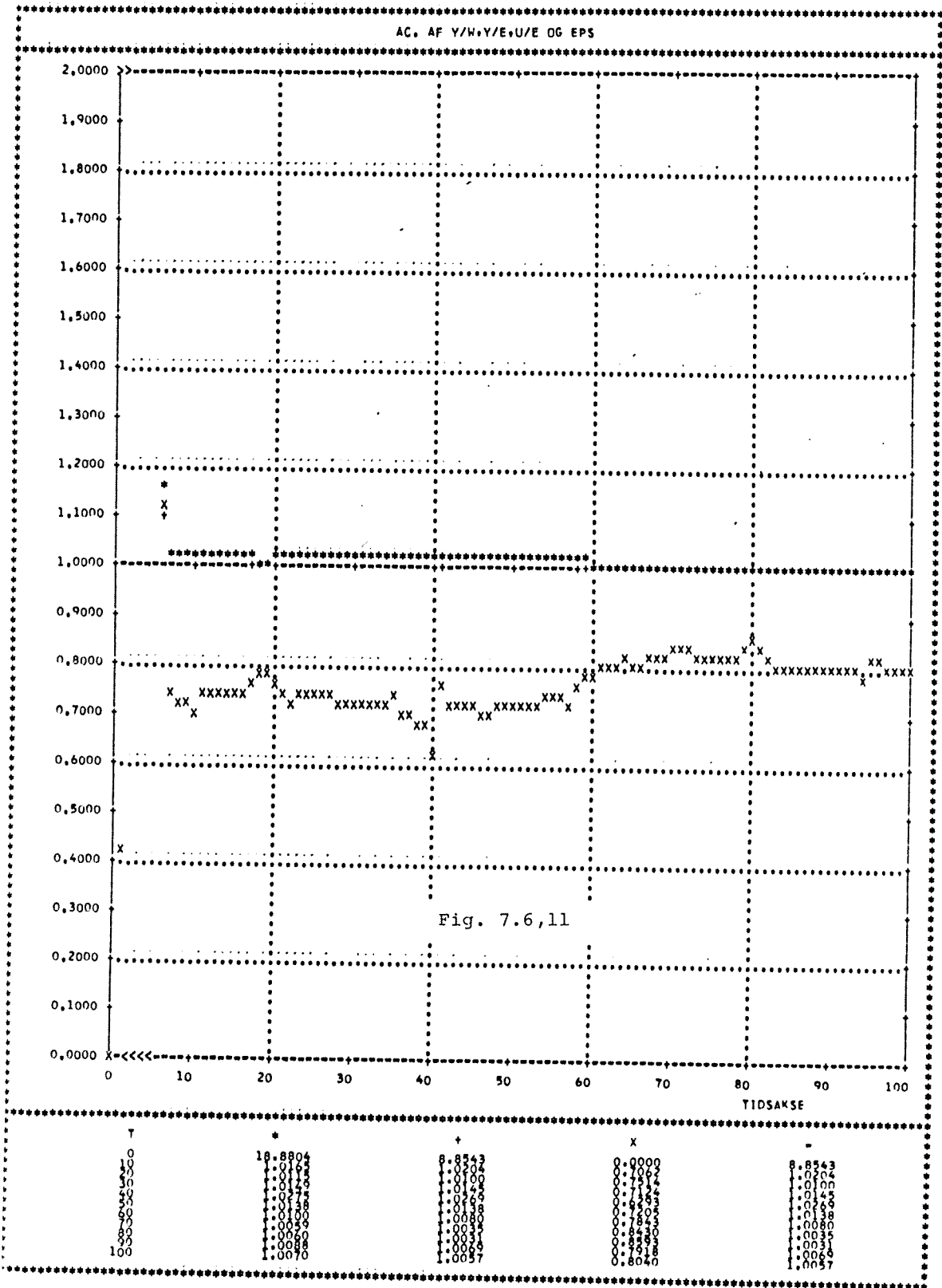
---ooo0ooo---

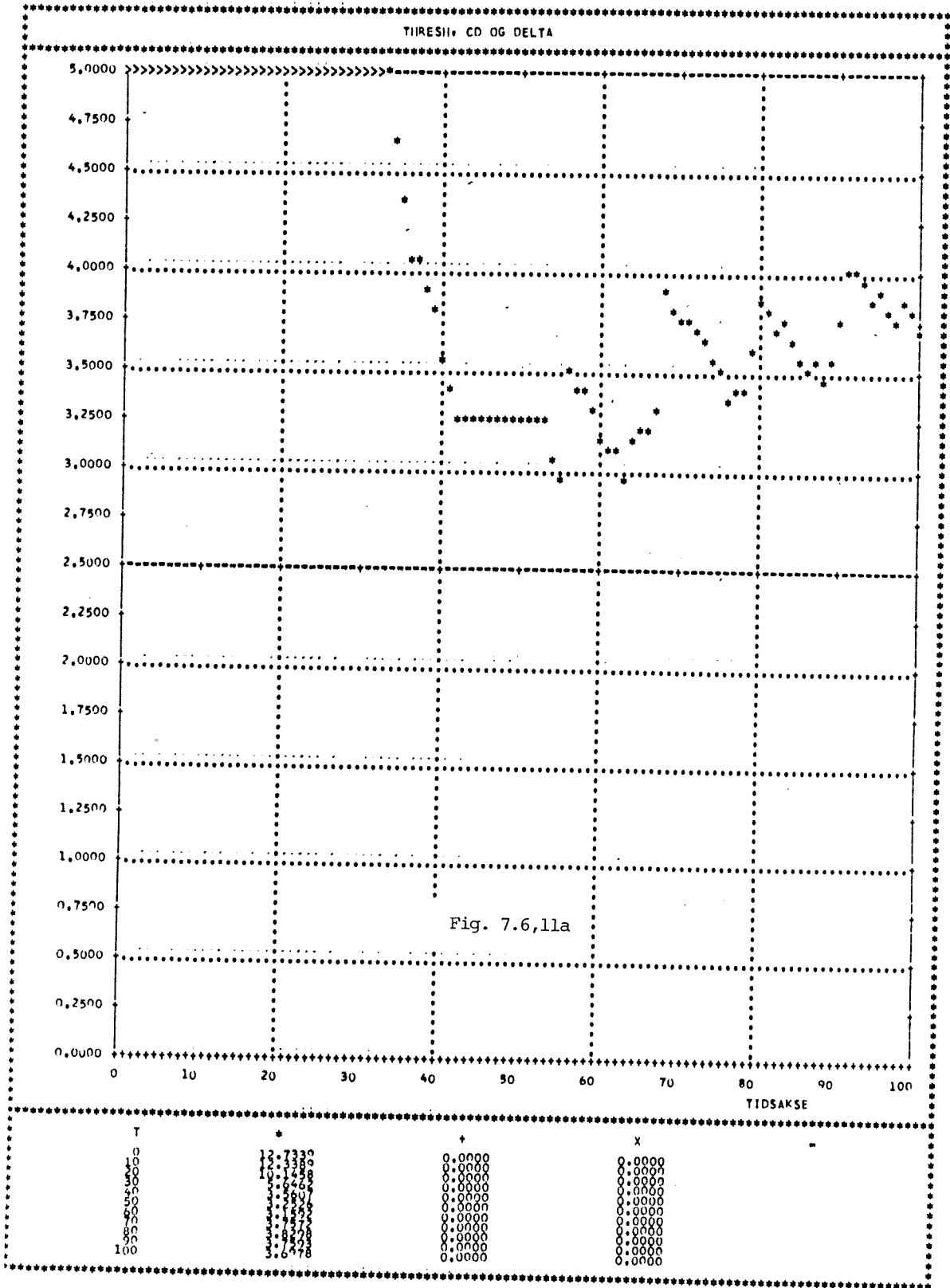




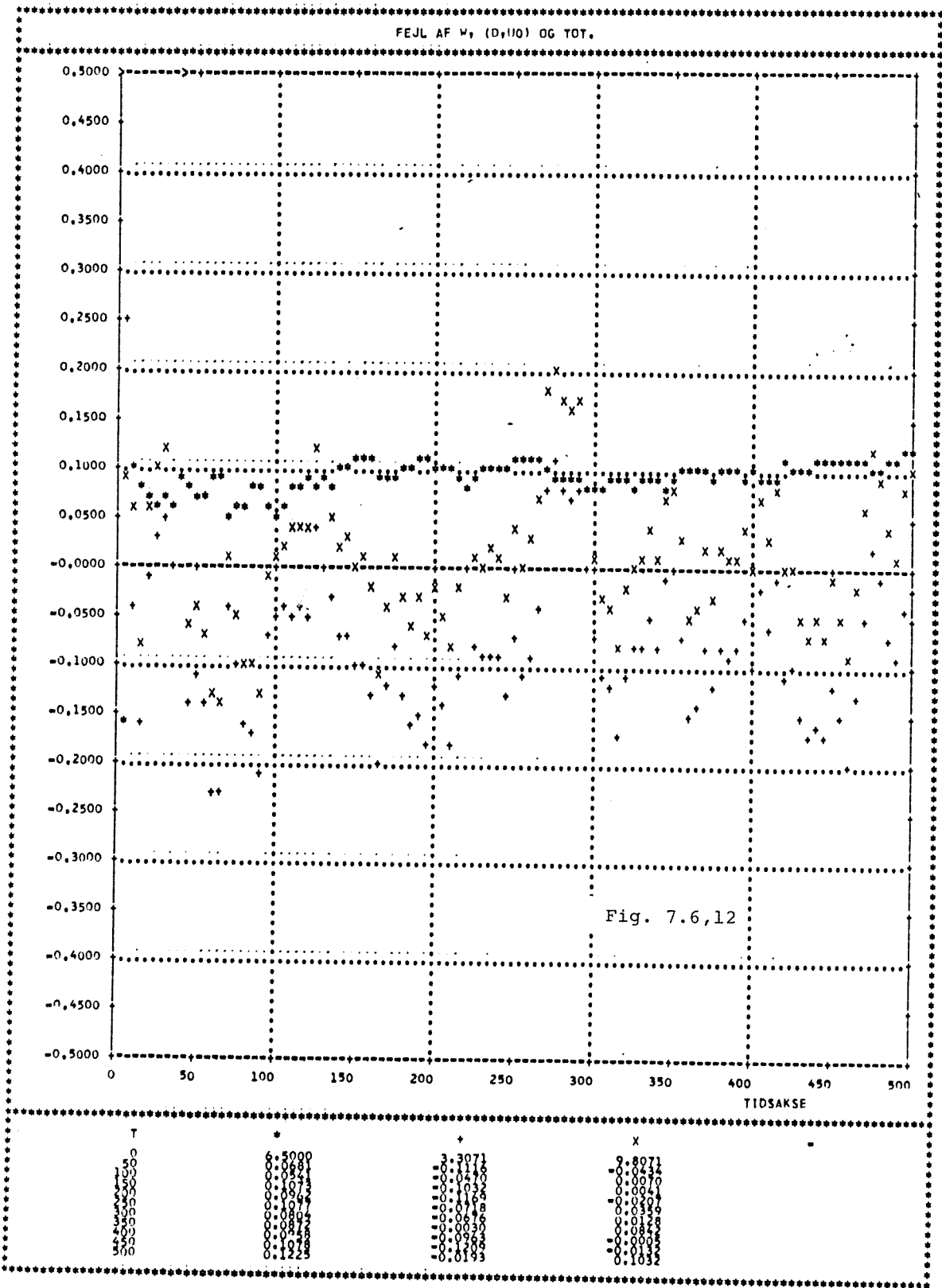


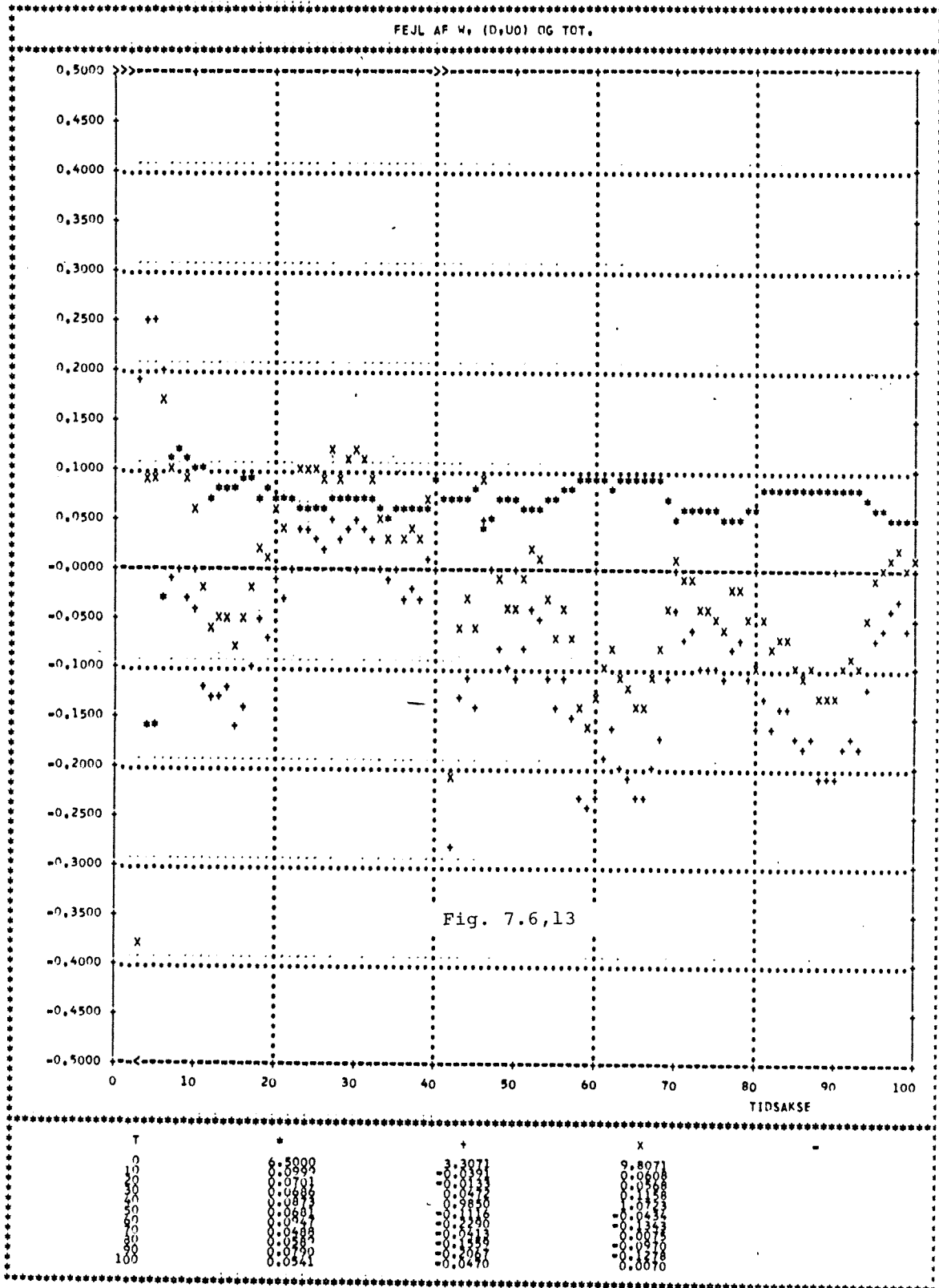


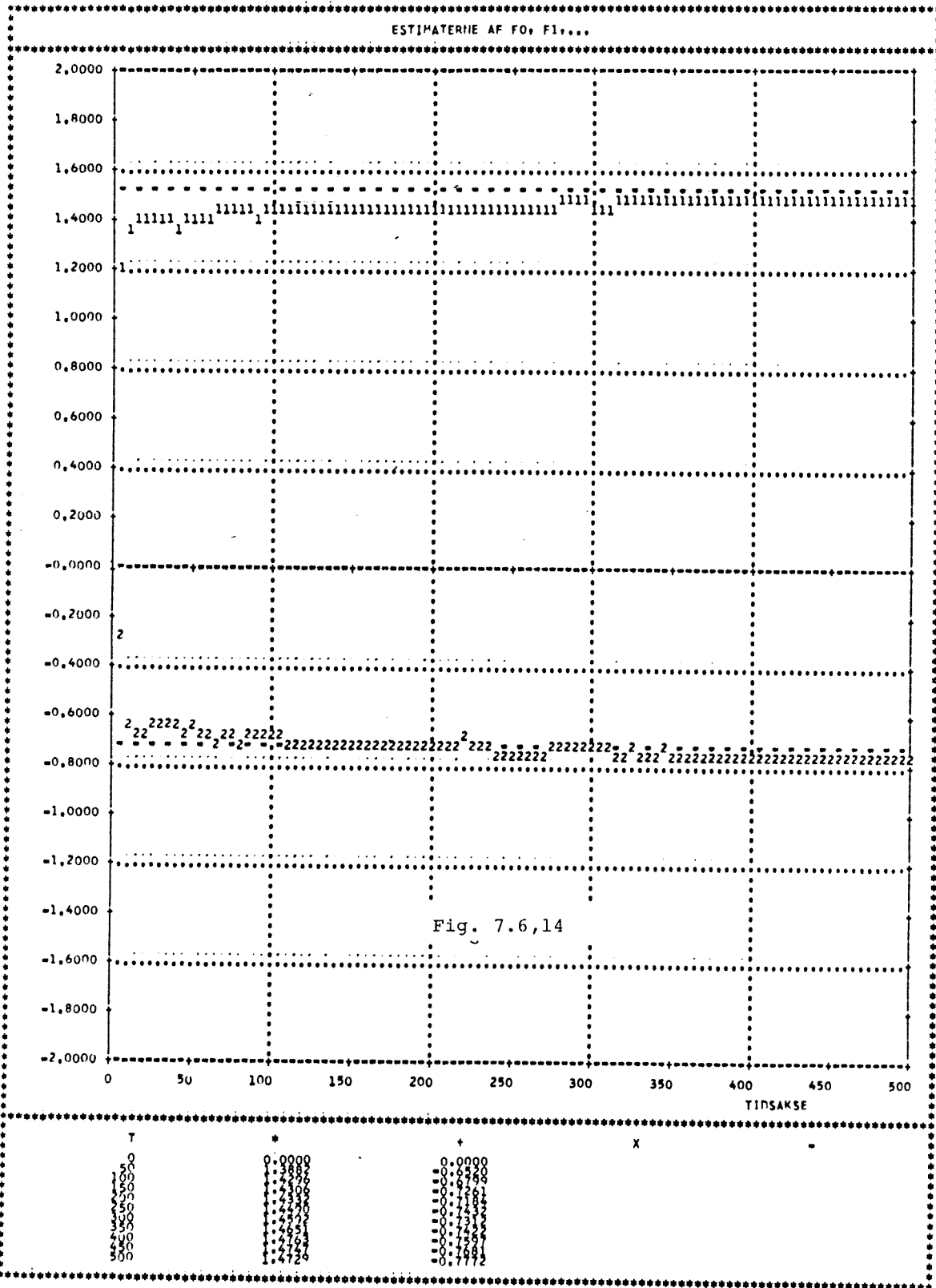


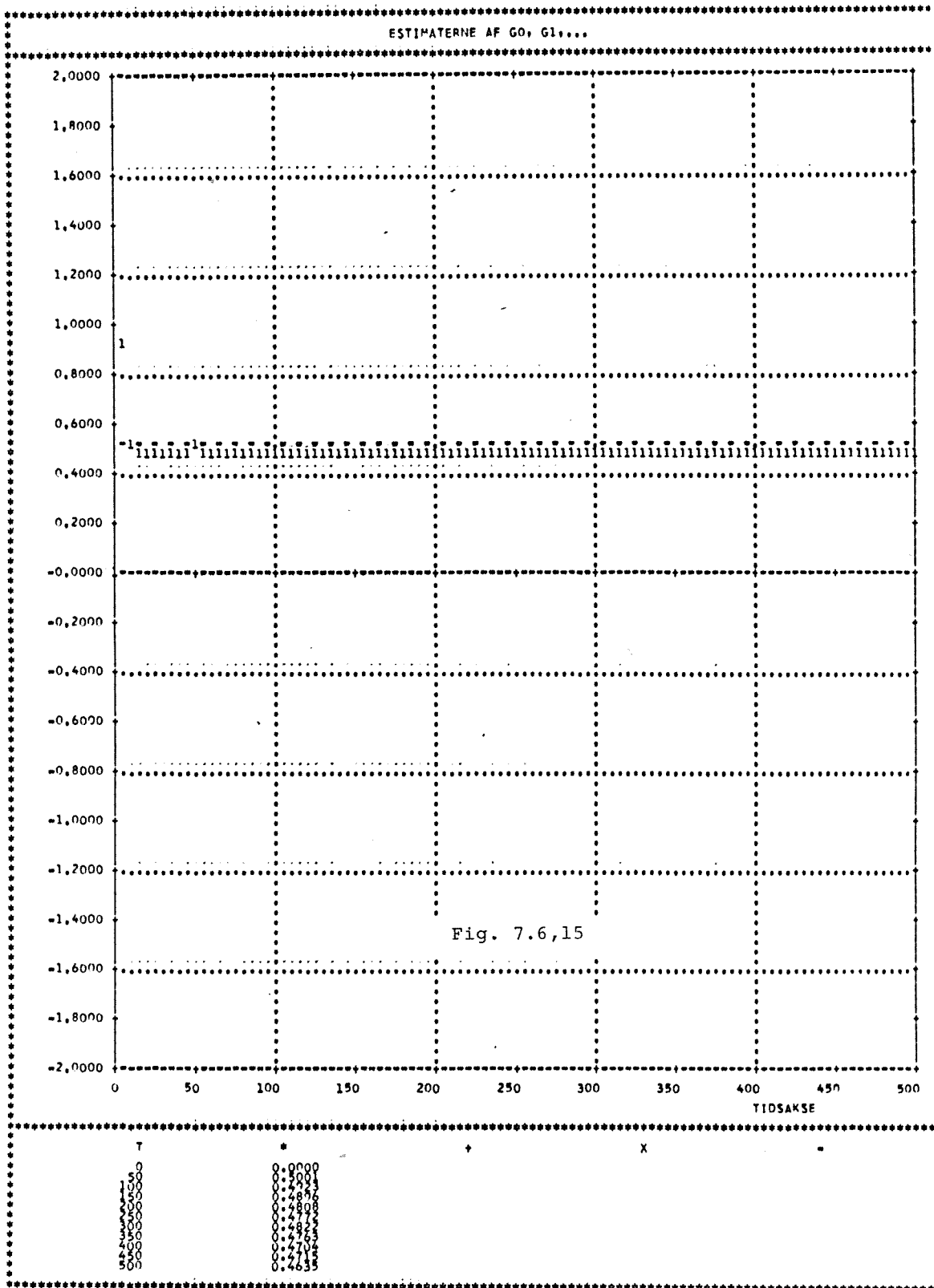


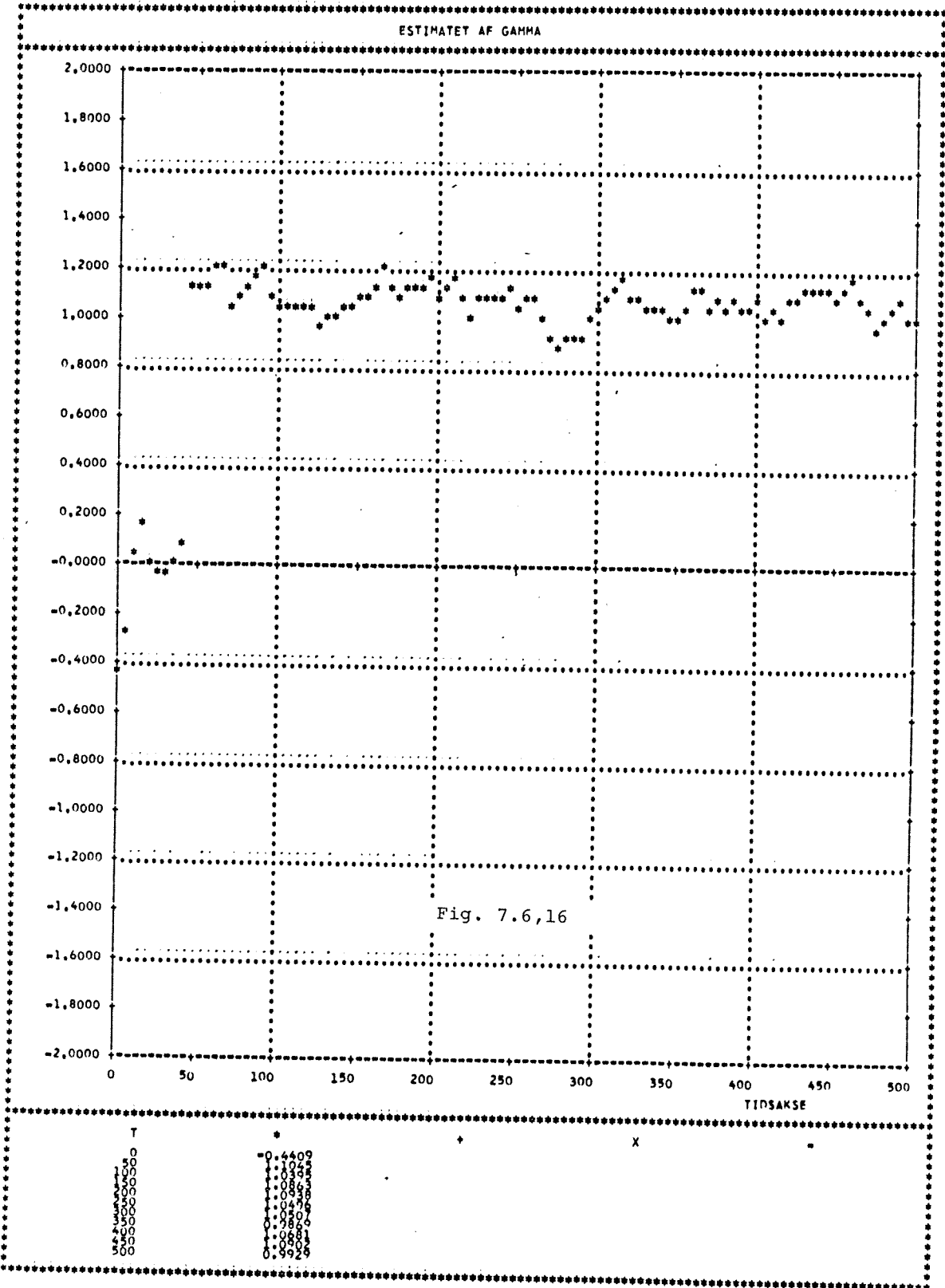
INITIAL GR#ISE TILF#LDE I 0

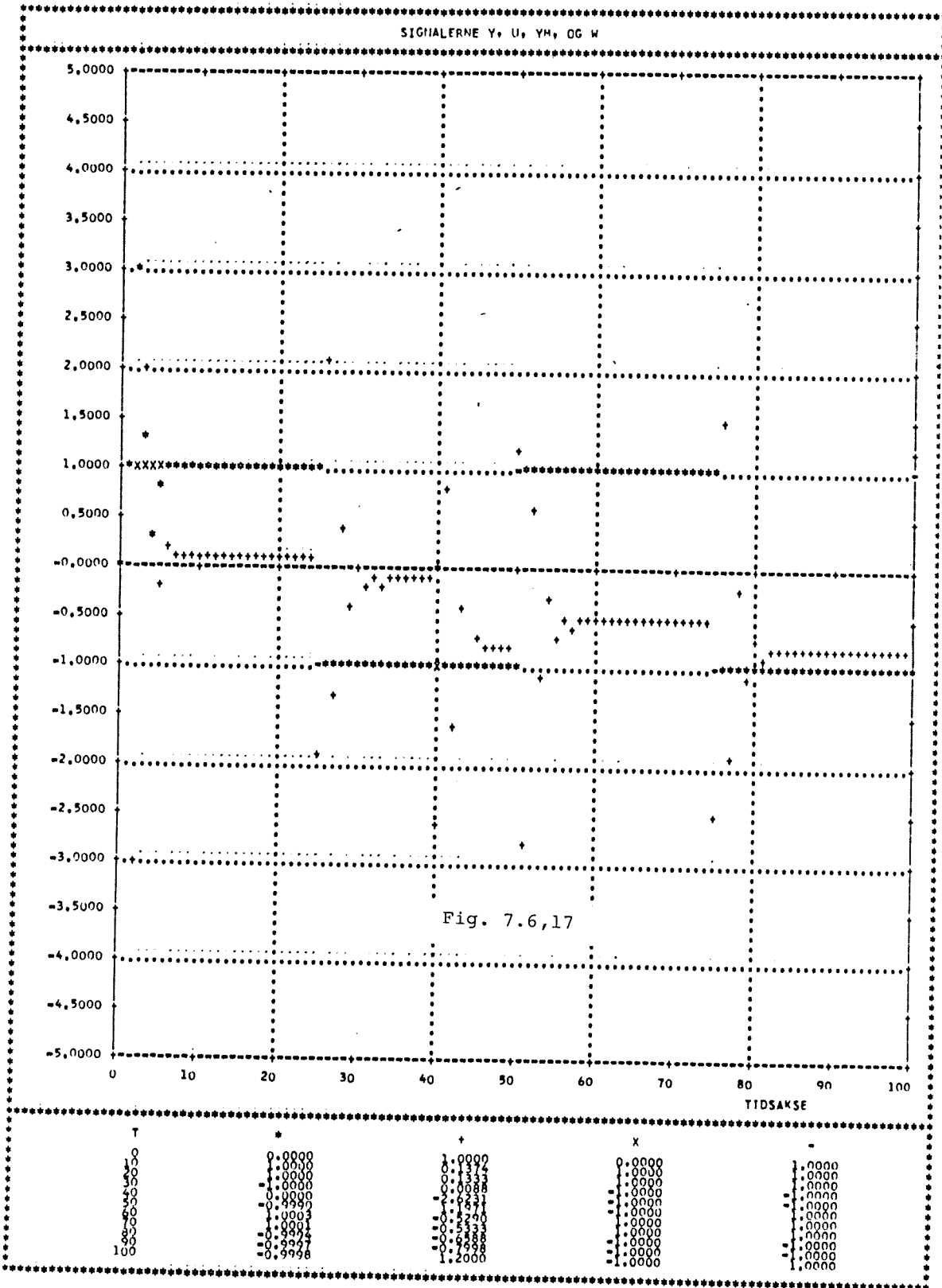


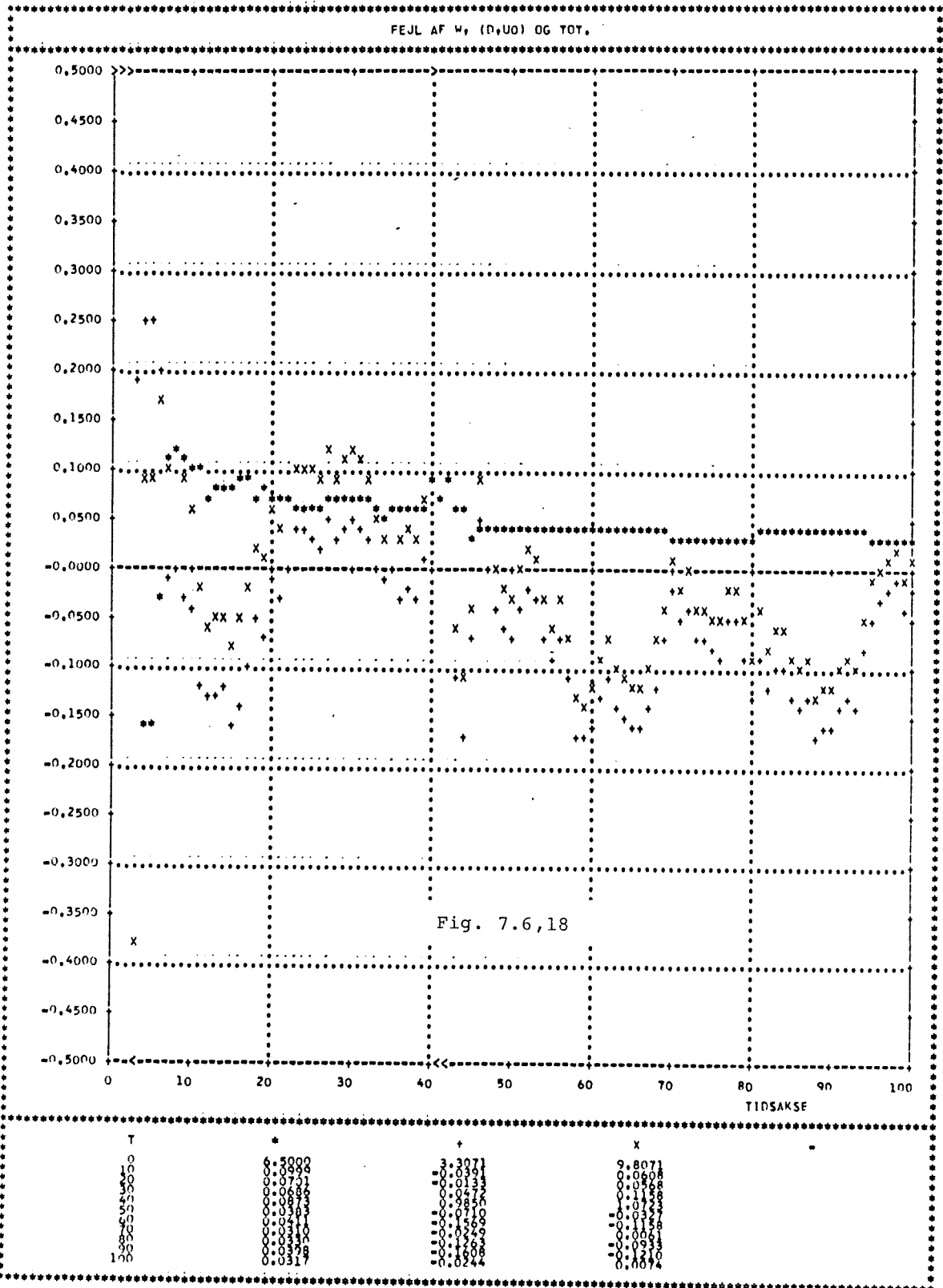


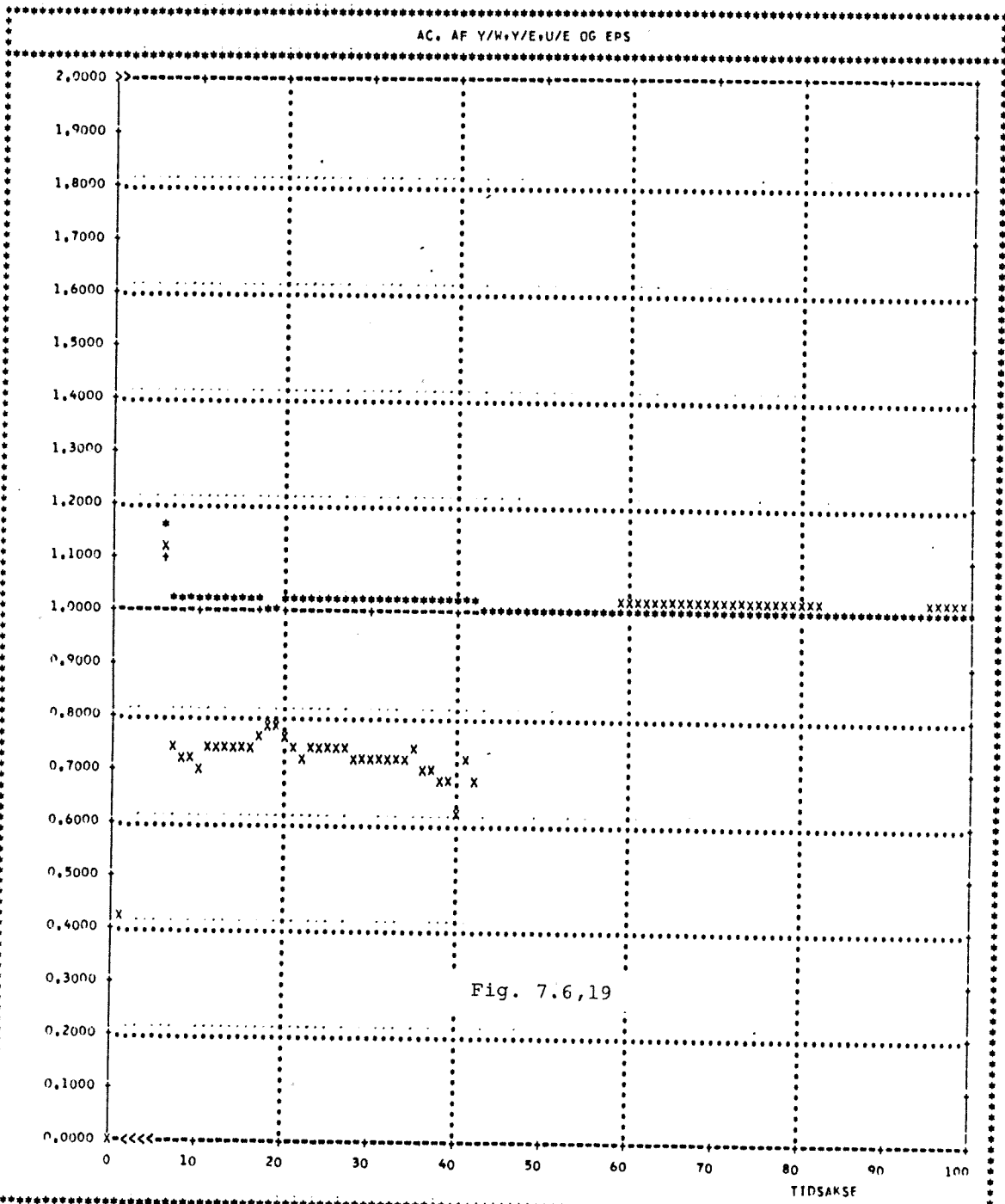




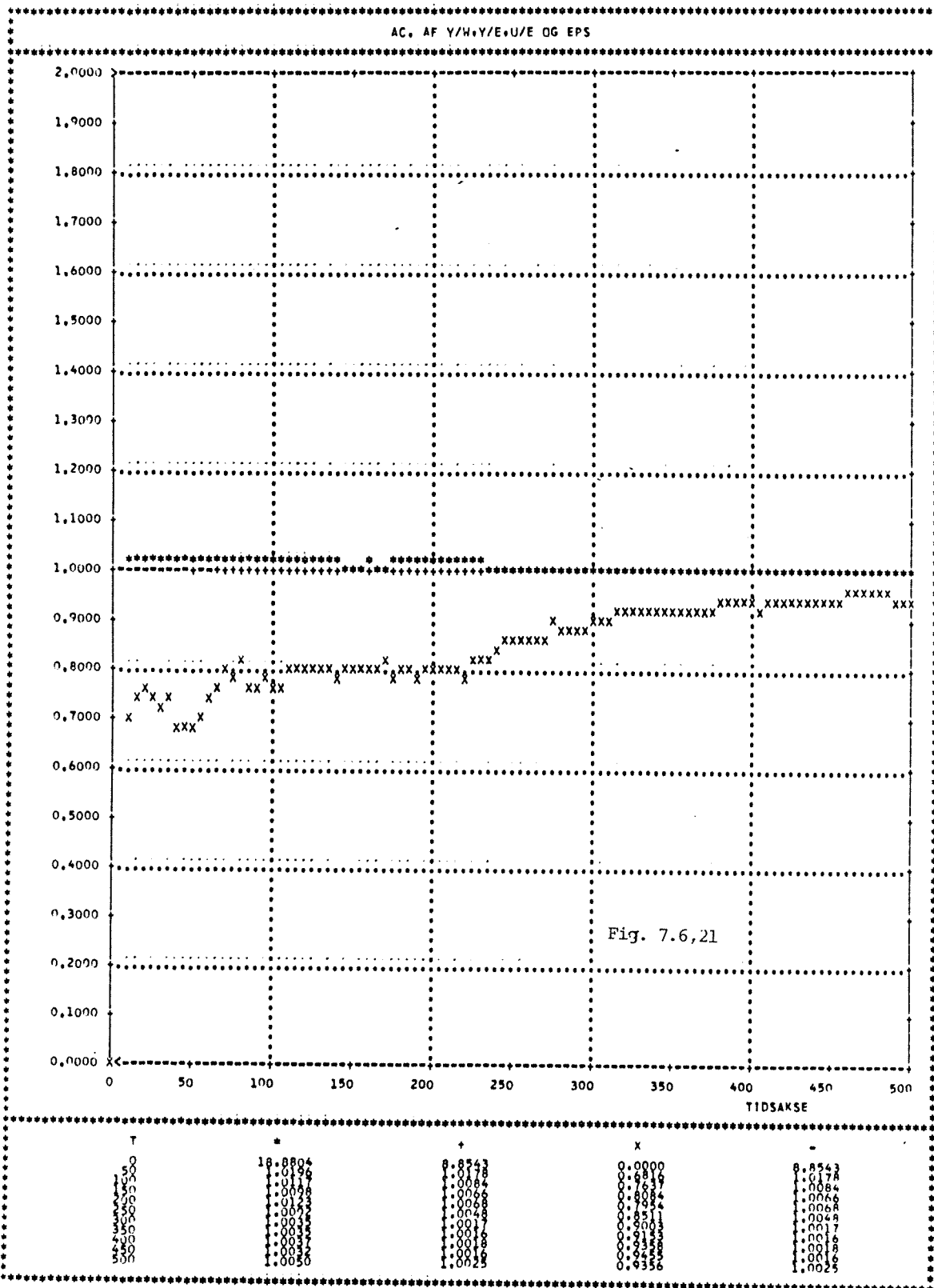


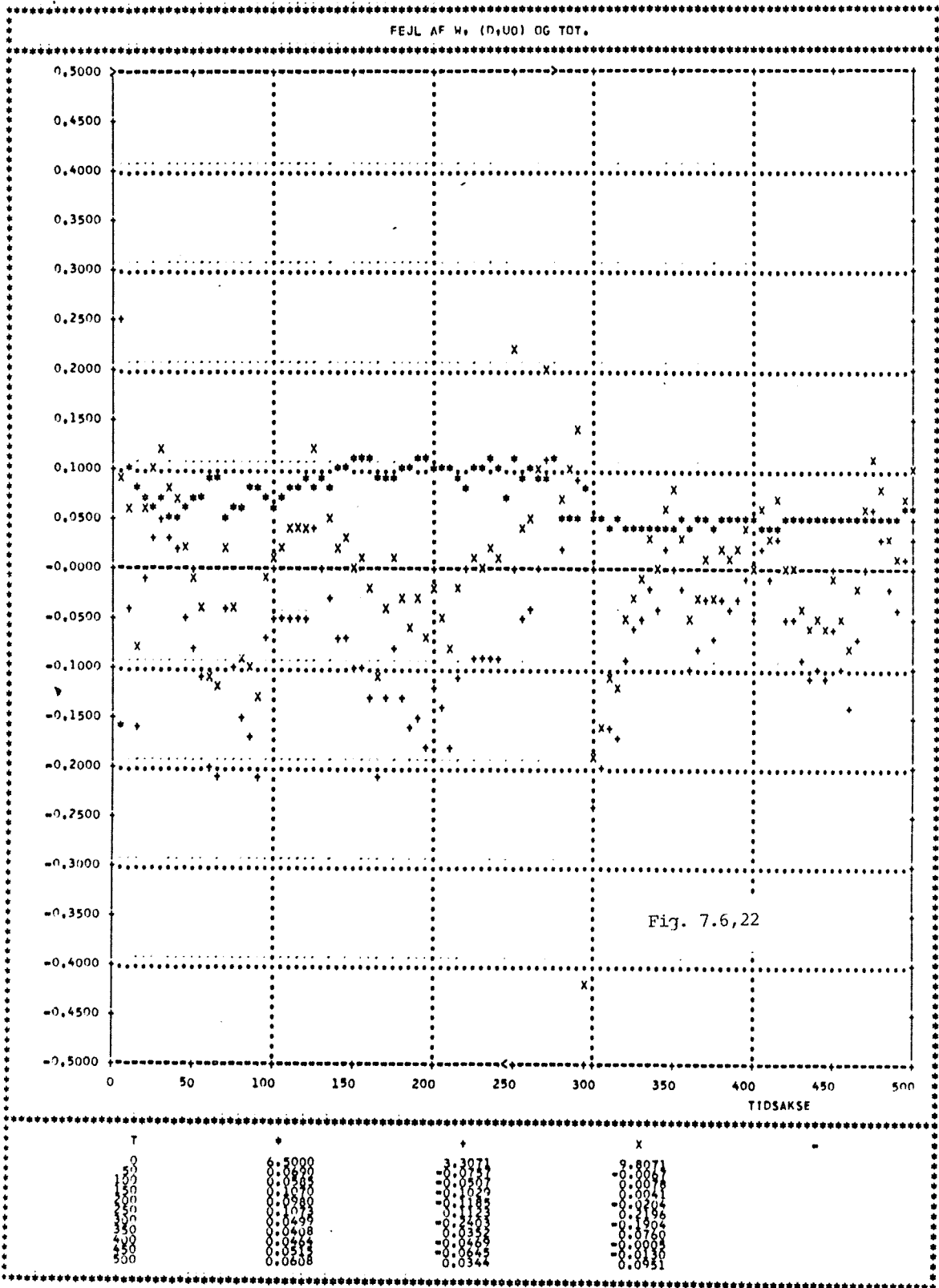






T	*	+	x	-
0	18.8804	8.8543	0.0000	8.8543
10	+0.0124	+0.0100	0.7042	+0.0204
20	+0.0114	+0.0100	0.7042	+0.0100
30	+0.0114	+0.0125	0.7114	+0.0145
40	+0.0098	+0.0064	0.6283	+0.269
50	+0.0098	+0.0008	+0.020	+0.006
60	+0.007	+0.0007	+0.109	+0.008
70	+0.007	+0.0007	+0.201	+0.007
80	+0.008	+0.0007	+0.206	+0.007
90	+0.008	+0.0007	+0.086	+0.007
100	+0.0006	+0.0005	+0.097	+0.005





af hvid støj, det vil sige at $C(q^{-1}) = 1$. For det tilfælde er iøvrigt de tre beskrevne estimationsalgoritmer, RLS, RELS og RML, identiske. De forrige simulationer viser tydeligt, at estimationsalgoritmerne er følsomme overfor spidser, hvorfor en mere robust estimations algoritme er påkrævet. Endvidere vides maximumlikelihood estimatorer at være følsomme overfor afvigelser fra den antagne fordeling, hvilket i adaptiv forbindelse underbygger ønsket om en mere robust estimator.

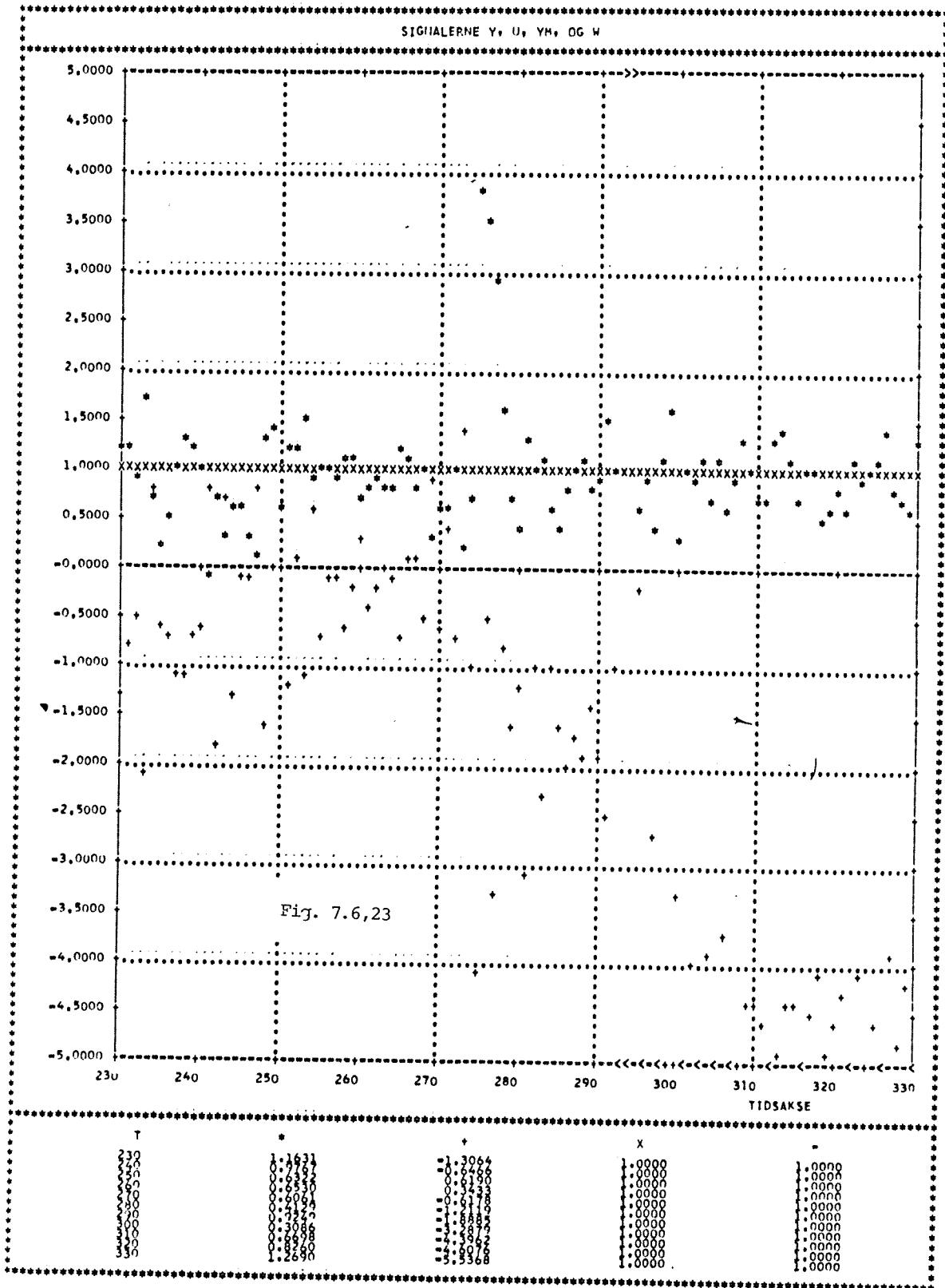
Udgangspunktet for den af Poljak og Tsyarkin i /12/ foreslåede metode til konstruktion af en robust estimator er ikke helt det samme som vores ikke stationære problem. Her og i det følgende antages det, at fordelingen af støjen, $e(t)$, er en stationær perturbation af en normalfordeling.

$$(3) \quad P_e(\xi) = (1-\alpha) P_0(\xi) + \alpha P_1(\xi)$$

hvor P_1 er en ukendt fordeling, og hvor α karakteriserer graden af kendskabet til fordeling P . Fordelingen P_0 er en $N(0, \sigma_0^2)$ - fordeling. Det antages altså, at fordelingen af støjen vides at tilhøre en klasse, \underline{P} , af fordelinger, der kan bringes på formen (3). Det oprindelige ikke stationære problem er her omformuleret til et stationært. Metodens anvendelighed vil afhænge af, hvor god denne tilnærmelse er. Den samme situation forelå i afsnit 7.3 med UKF - metoden. Samtidig skal det nævnes, at metoden til bestemmelse af robuste estimatorer har langt større generellitet end illustreret ved dette problem.

I det følgende vil Poljak og Tsypkins metode til dimensionering af en robust maximumlikelihood estimator blive refereret. Metoden vil på grund af de fælles træk mellem RLS og RML blive generelliseret til de fordelingfri estimationsalgoritmer, RLS og RELS.

Det kan vises j.v.f. /12/, at under visse forudsætninger



7.7 Robust Identifikation.

De foregående afsnit har været centreret om metoder til forbedring af algoritmernes egenskaber overfor variationer i belastningen $d(t)$. Formålet i dette afsnit er at undersøge metoder, der kan forbedre standardalgoritmernes egenskaber overfor støj af typen:

$$(1) \quad e(t) = (1 - \delta_1(t))e_0(t) + \delta_1(t)e_1(t)$$

hvor $e_0(t)$ er en sekvens af uafhængige $N(0, \sigma_0^2)$ fordelte stokastiske variable. Som angivet i kapitel 2 er $\delta_1(t)$ en markovkæde med to tilstande, E_0 og E_1 , hvor $\delta_1(t)$ antager værdierne 0 og 1. Overgangssandsynligheden s_1 indgår i den stokastiske matrix for processen:

$$(2) \quad \pi_1 = \begin{pmatrix} 1-s_1 & s_1 \\ 1-s_1 & s_1 \end{pmatrix} \quad 0 \leq s_1 \ll 1$$

Fordelingen af $e_1(t)$ kan i princippet være ukendt, men er i simulationerne modelleret med en $N(0, \sigma_1^2)$ - fordeling, hvor

$$\sigma_1 \gg \sigma_0.$$

Herved bliver variansen, σ^2 , af støjsignalet, $e(t)$, kraftigt tidsvarierende. Metodens løsning hindrer ikke at σ_1 er tidsvarierende. Det er endvidere formålet, at udforme metoden sådan at den kan kombineres til en samlet algoritme med de i det foregående beskrevne løsninger specielt detektor - og UKF løsningen.

Til de simulationer, der er anvendt til dokumentationen, er der anvendt en RLS - algoritme til at estimere parameterne med. Dette skyldes at det brugte testobjekt (objekt I) er påvirket

lyder Cramer - Rao uligheden for estimatoren $\hat{\theta}$

$$(4) \quad E\{ (\hat{\theta} - \theta)(\hat{\theta} - \theta)^T \} \geq \frac{\Omega^{-1}}{I(P_e)}$$

for identifikationsproblemet (3,1). Matrixuligheden symboliserer på sædvanlig måde, at matrix differencen ikke er negativ definit. Her er den skalare størrelse, $I(P_e)$, Fisher informationen:

$$(5) \quad I(P_e) = \int_{-\infty}^{\infty} \frac{P_e'(\xi)^2}{P_e(\xi)} d\xi$$

og matricen Ω er her en system afhængig matrix. Når P_e vides at tilhøre klassen \underline{P} af fordelinger, vil det værst tænkelige tilfælde indtræffe, hvis Fisher informationen, $I(P)$, er minimal. Der kan findes en fordeling, P^* , i \underline{P} , der minimerer Fisher informationen, det vil sige at:

$$(6) \quad P^* = \arg \min_{P \in \underline{P}} I(P)$$

Såfremt der dimensioneres en maximumlikelihood estimator over for denne fordeling, vil denne estimator være robust. Det kan endvidere vises, at den dermed opnåede estimator er asymptotisk optimal i en minimax opfattelse. Det kan nemlig under visse forudsætninger j.v.f. /12/ vises, at estimatoren er konsistent og at den asymptotiske kovariansmatrice opfylder at:

$$(7) \quad V(P_e) \leq V(P^*)$$

for alle fordelinger P i klassen \underline{P} . Variationsproblemet (6) kan være vanskeligt at løse. Løsningen for klassen \underline{P} givet ved (3) er j.v.f. /12/.

$$(8) \quad P^*(\xi) = \begin{cases} k_1 \exp\left(-\frac{1}{2} \left(\frac{\xi}{\sigma_0}\right)^2\right) & \text{for } |\xi| \leq \Delta \\ k_2 \exp\left(-\frac{|\xi|}{k_3}\right) & \text{for } |\xi| > \Delta \end{cases}$$

hvor konstanterne er positive og afhænger af α og σ . I dimensioneringen af maximumlikelihood estimatoren er kernen, $\chi(\xi)$, i kriteriefunktionen $J(t, \theta)$ givet ved:

$$(9) \quad \begin{aligned} \chi(\xi) &= -\log P^*(\xi) \\ &= \begin{cases} -\log(k_1) + \frac{1}{2} \left(\frac{\xi}{\sigma_0}\right)^2 & \text{for } |\xi| \leq \Delta \\ -\log(k_2) + \frac{|\xi|}{k_3} & \text{for } |\xi| > \Delta \end{cases} \end{aligned}$$

Den øvrige dimensionering følger kapitel 3. Det vil sige at:

$$(10) \quad \hat{\theta} = \arg \min J(t, \theta)$$

hvor

$$(11) \quad \begin{aligned} J(t, \theta) &= \sum_{i=0}^t \chi(\varepsilon(t)) \\ \varepsilon(t) &= y(t) - f(x(t), \theta) = y(t) - x^T(t) \end{aligned}$$

Anvedens der en gradient metode til minimering af kriteriefunktionen, J , kan denne tilnærmet bringes på rekursiv form

$$(12) \quad \begin{aligned} \hat{\theta}(t+1) &= \hat{\theta}(t) - P(t+1) \chi'(\underline{\varepsilon}) \nabla_{\theta} f(x(t), \theta) \\ \underline{\varepsilon}(t) &= -\varepsilon(t) \end{aligned}$$

Matricen $P(t)$ opdateres ved:

$$(13) \quad P(t+1) = P(t) - \frac{P(t) \nabla_{\theta} f(\nabla_{\theta} f)^T P(t)}{\chi''(\underline{\varepsilon})^{-1} + (\nabla_{\theta} f)^T P(t) \nabla_{\theta} f}$$

Tolkningen af $\nabla_{\theta} (x^T(t) \theta)$ er kendt for de tre tilfælde, RLS, RELS og RML. I RML tilfældet er:

$$(14) \quad \nabla_{\theta} (x^T(t) \theta) = z(t) = \frac{x(t)}{\hat{C}(q^{-1})}$$

Udtrykkene (11) og (9) giver nu at

$$\hat{\theta}(t+1) = \begin{cases} \hat{\theta}(t) + P(t+1) z(t) \varepsilon(t) & \text{for } |\varepsilon| \leq \Delta \\ \hat{\theta}(t) + P(t+1) z(t) \Delta & \text{for } \varepsilon > \Delta \\ \hat{\theta}(t) - P(t+1) z(t) \Delta & \text{for } \varepsilon < -\Delta \end{cases} \quad (15)$$

$$P(t+1) = \begin{cases} P(t) - \frac{P(t) z(t) z^T(t) P(t)}{1 + z^T(t) P(t) z(t)} & \text{for } |\varepsilon| \leq \Delta \\ P(t) & \text{for } |\varepsilon| > \Delta \end{cases} \quad (16)$$

Det bemærkes, at denne estimator indeholder den sædvanlige RML-algoritme, når residualerne er indenfor grænsen Δ . Hvis en spids indtræffer eller hvis residualerne af anden grund er udenfor grænsen, opdaters $P(t)$ ikke og kun en begrænset del af residualen anvendes til opdatering af $\hat{\theta}(t)$. Som det fremgår af udtrykket (17) for forstærkningen

$$(17) \quad K(t) = P(t+1) z(t) = \begin{cases} P(t) z(t) & \text{for } |\varepsilon| > \Delta \\ \frac{P(t) z(t)}{1 + z^T(t) P(t) z(t)} & \text{for } |\varepsilon| \leq \Delta \end{cases}$$

er der specielt for kraftig eksitation ($z^T z \gg 1$) en betydelig

forskel på forstærkningen overfor et begrænset residual og et lige indenfor. Anvendes denne metode også i forbindelse med de fordelingsfri estimationsalgoritmer, kan den generelle algoritme fra afsnit 3.6 udvides til nedenstående, der vil blive benævnt RII (Robust identifikation, version 1). Tolkningen af $z(t)$ og θ er uændret i forhold til afsnit 3.6.

$$(18) \quad \varepsilon(t) = y(t) - x^T(t)\hat{\theta}(t)$$

$$(19) \quad R(t) = P(t)z(t)$$

$$(20) \quad \mu(t) = \begin{cases} \lambda(t) + z^T(t)R(t) & \text{for } |\varepsilon| \leq \Delta \\ 1 & \text{for } |\varepsilon| > \Delta \end{cases}$$

$$(21) \quad K(t) = \frac{R(t)}{(t)}$$

$$(22) \quad \hat{\theta}(t+1) = \begin{cases} \hat{\theta}(t) - K(t)\Delta & \text{for } \varepsilon < -\Delta \\ \hat{\theta}(t) + K(t)\varepsilon(t) & \text{for } |\varepsilon| \leq \Delta \\ \hat{\theta}(t) + K(t)\Delta & \text{for } \varepsilon > \Delta \end{cases}$$

$$(23) \quad P(t+1) = \begin{cases} P(t) + R_1 - K(t)R^T(t) & \text{for } |\varepsilon| \leq \Delta \\ P(t) & \text{for } |\varepsilon| > \Delta \end{cases}$$

Denne algoritme er p. gr. af en fejltolkning af rekursionerne ikke blevet afprøvet ordenligt. Derimod er en anden algoritme, der kun afviger ved, at udtrykket (20) er erstattet med:

$$(24) \quad \mu(t) = \lambda(t) + z^T(t)R(t)$$

Forstærkningen $K(t)$, der i så fald kun afhænger af forholdene op til og med tidspunktet $t-1$, udviser følgelig ikke den før

nævnte markante forskel overfor residualernes størrelse: Ved kraftige eksitationer vil RI1 have en stor forstærkning overfor begrænsede residualer, mens den sidst beskrevne vil have samme lave forstærkning overfor såvel begrænsede som residualer indenfor grænserne. Man kan betegne denne som mere konservativ og vil blive refereret som RI2.

En tredje algoritme (RI3), der hviler på et heuristisk grundlag, fremkommer ved at ændre (22) i RI2 til:

$$\hat{\theta}(t+1) = \begin{cases} \hat{\theta}(t) + K(t)\varepsilon(t) & \text{for } |\varepsilon| \leq \Delta \\ \hat{\theta}(t) & \text{for } |\varepsilon| > \Delta \end{cases}$$

hvilket svarer til at se bort fra outliers, der er en ofte anvendt teknik ved estimation. Set fra et reguleringsteknisk synspunkt udviser denne metode imidlertid visse problematiske forhold. Hvis det samlede system er blevet ustabilt p. gr. af fejlagtige parameterestimer vil $\hat{\theta}(t)$ blive fastlåst i dette for systemet ustabile punkt. Metoden vil derimod have ret gode egenskaber overfor enkelte spidser. Algoritmen, RI3, er derfor en kombination mellem den rene afvisning af spidser og RI2, således at 3 på hinanden følgende begrænsede residualer afvises, mens den fjerde og de efterfølgende bliver anvendt som i RI2.

Poljak og Tsytkin angiver ingen metode til at fastsætte grænsen, Δ , En håndregel siger, at den skal vælges mellem $1.5\sigma_0$ og $2\sigma_0$. Da spredningen, σ_0 , for det stationære støjsignal ikke kendes er den eneste udvej at estimere den f. eks. med den estimator, $Q(t)$, der er angivet i afsnit 7.6. Anvendes denne undgås indvirkning fra spring, såfremt de detekteres. Denne estimator arbejder på residualerne direkte, hvorfor spidserne vil påvirke estimerne. For korrekte estimer er:

$$(26) \quad \varepsilon(t) = e(t) = (1 - \delta_1(t)) e_0(t) + \delta_1(t) e_1(t)$$

For at undgå dette kan estimereren, $Q(t)$, arbejde på nogle begrænsede residualer, $\epsilon_2(t)$.

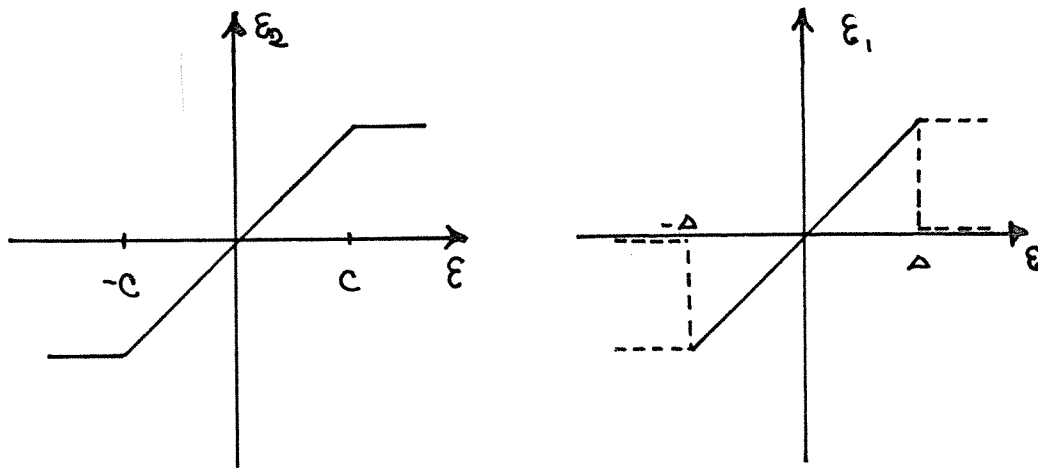


Fig. 1: Begrænserkarakteristikker

Grænsen $c(t)$, som $\epsilon_2(t)$ afhænger af, må nødvendigvis også bestemmes ud fra σ_0 . Sættes denne grænse til:

$$(27) \quad c(t) = 2 \sqrt{Q(t)}$$

hvor $Q(t)$ er estimeret på grundlag af $\epsilon_2(t-k)$ j.y.f. (7.6,18). Herved opstår en tilbagekobling, hvis stabilitet er vanskelig at fastslå formelt. Dette fremgår tydeligt hvis fig. 2 sammenholdes med fig. 7.6,4.

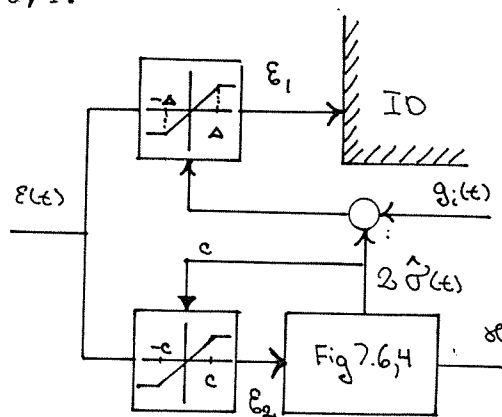


Fig. 2: Struktur i tilbagekoblingen

I appendix 7 er der angivet nogle meget heuristiske argumenter for stabilitet af tilbagekoblingsløjfen, der både er tidsaf - hængig og ikke lineær. Desuden er påvirkningerne stokastiske. Af hensyn til de ustationære forhold anvendes der to begrænsede residualer, $\varepsilon_1(t)$ og $\varepsilon_2(t)$. Til den egentlige estimation af parameterne bruges $\varepsilon_1(t)$ mens $\varepsilon_2(t)$ som nævnt tilføres estimatoren $Q(t)$. Såfremt $Q(0)$ initialiseres til en så stor værdi at $\varepsilon(0)$ ikke bliver begrænset, vil $Q(t)$ i startperioden hovedsagelig være bestemt af residualer, der skyldes parameterfejl. Enkelte store spidser vil på trods af den store forstærkning gennem estimatoren (7.6,18) til $c(t)$ og $Q(t)$ ikke påvirke disse i højere grad end parameterfejlene. Hvis der havde været anvendt en anden løsning, hvor f. eks. begrænsningen til estimatoren, $Q(t)$, havde været indsat senere i forløbet, ville en enkelt spids i startperioden kunne sætte hele begrænsnings - systemet ud af funktion i et tidsrum, der er bestemt af glemselsfaktoren, λ^Q , og spidsens størrelse.

Til den egentlige estimation af parameterne (22 - 25) vil det ikke være særligt hensigtsmæssig at anvende samme begrænsning som til $Q(t)$ mens grovestimationen foregår. Derfor er grænsens tidsforløb, $\Delta(t)$, forsynet med en ruse.

$$\Delta(t) = c(t) + g_i(t)$$

Her er $g_i(t)$ valgt som en linært aftagende funktion i t .

$$g_i(t) = \begin{cases} 10 - t & \text{for } t < 10 \\ 5 + (T_{d,i} - t) & \text{for } T_{d,i} < t < T_{d,i} + 5 \\ 0 & \text{ellers} \end{cases}$$

Umiddelbart efter et detekteret spring vil estimatet af $d(t)$ (d.v.s. \hat{d} eller $\hat{\gamma}$) have et mindre oversving som følge af den store forstærkning i det udvidede kalmanfilter, der foretager sporingen af springet. Det kan tænkes, at et sådant oversving kan give anledning til større residualer, hvorfor begrænsningen

$\Delta(t)$ også her forsynet med en ruse.

I de følgende simulationer er detektoren, der er beskrevet i afsnit 7.6, inaktiv, sådan at det kun er de robuste identifikationsalgoritmer, der bliver undersøgt. Endvidere er:

$$\lambda(t) = 1 \quad R_1 = \Gamma \Gamma^T Q_d^0 = 0$$

Først vil det blive undersøgt om reguleringskvaliteten og lign. ændres ved at anvende en robust identifikationsalgoritme, når objektet påvirkes af normalfordelt hvid støj med konstant middelværdi og spredning. Når der som her indføres ændringer for at øge den adaptive regulators robusthed, er det af stor interesse, om den indførte ændring forringer regulatorens egenskaber overfor konventionelle påvirkninger. Der er derfor foretaget en række simulationer med MVO - regulatoren overfor objekt I, når dette påvirkes af støjen $e(t) = e_0(t)$. Parameterne i estimationsmodellen bliver her estimeret med RI3. I fig. 3 - 7 er nogle af tidsforløbene vist. I tabene for y , hvis kurver er afbildet i fig. 3, er der en ringe stigning (0.5 %) i forhold til standard algoritmen (fig. 6.1.1,6). For styresignalets vedkommende er tabet blevet formindsket i forhold til standard tilfældet. Dette skyldes en ringere konvergens af AC - forstærkningen for $H_{u,e}$. På nær en trivial startkonstant er tabet for $y(t)$ og $\varepsilon(t)$ identiske. Årsagen til den tilsyneladende afvigelse herfra i fig. 3 er, at her indgår kun de begrænsede residualer, $\varepsilon_1(t)$, i tabet. I fig 4 og 5 er tidsforløbet af varianserne for $y(t)$, $u(t)$, $\varepsilon_1(t)$ samt

$$e_f(t) = y(t) - w(t-1) = y(t) - 1$$

afbildet. Disse kurver antyder, at hverken reguleringskvaliteten eller konvergenshastigheden er nedsat i højere grad. AC - forstærkningen for $H_{u,e}$ er dog undtaget. Fig. 7 viser, at med den anvendte støjsekvens og med det anvendte testobjekt er

konvergens af $H_{u,e}$ temmelig langsom også i forhold til standard tilfældet (fig. 6.1.1,10). En sammenligning mellem fig. 6 og fig. 6.1.1,9 viser, at DC - forholdene er forringet lidt. I fig. 8 er tidsforløbet af grænserne, $\Delta(t)$ (DELTA, X) og $c(t)$ (CD, +), angivet. Det ses, at disse stabiliserer sig lidt under $2\sigma_0$ (= 0.63) som forudsagt i appendix 7. Detektor - grænsen $c_1(t)$ (THRESH, *) er som nævnt ikke anvendt. Konklusionen er, at ved at anvende RI3 bortkastes 5,6% af målingerne, der ikke forringer reguleringskvaliteten væsentlig. Derimod er den ret følsomme $H_{u,e}$ i dette tilfælde påvirket en del.

Nu vil objektet blive udsat for en støj på formen:

$$e(t) = (1 - \delta_1(t)) e_0(t) + \delta_1(t) e_1(t)$$

hvor overgangssansynligheden, s_1 , for δ_1 er 2% og hvor $e_1(t)$ er en følge af uafhængige $N(0, \sigma_1^2)$ fordelte stokastiske variable. Simulationen er foretaget med $\sigma_1 = 25$ og $\sigma_0^2 = 0.1$. Estimationen af parameterne vil stadigvæk foregå med RI3. Den anvendte støjsekvens er angivet i fig. 9. Bemærk de store spidser til $t = 6$ og $t = 9$. Tidsforløbet af DC - fejlen og AC - forstærkningen, der er afbildet i fig. 10 og 11, udviser en meget sikker og hurtig konvergens. Afbildningen i fig. 12 viser, at grænserne $c(t)$ og $\Delta(t)$ er steget meget lidt som følge af spidserne. Den gode konvergens, som også fremgår af fig. 13 - 15, hvor parameterestimerne er afbildet, skyldes, at RI3 på den ene side hindrer den negative indflydelse fra spidserne men på den anden side anvender den eksitation, som spidserne til $t = 6$ og $t = 9$ giver anledning til. Det skal samtidig nævnes at en standard algoritme (e.v.t. med en glemselsfaktor eller et udvidet kalmanfilter) også vil opnå en meget fin konvergens, hvis DC - fejlen er uden betydning for reguleringen. I deadbeat-regulatorerne anvendes estimatet af d ikke til reguleringen. Men på grund af, at parameterestimerne springer, når objektet

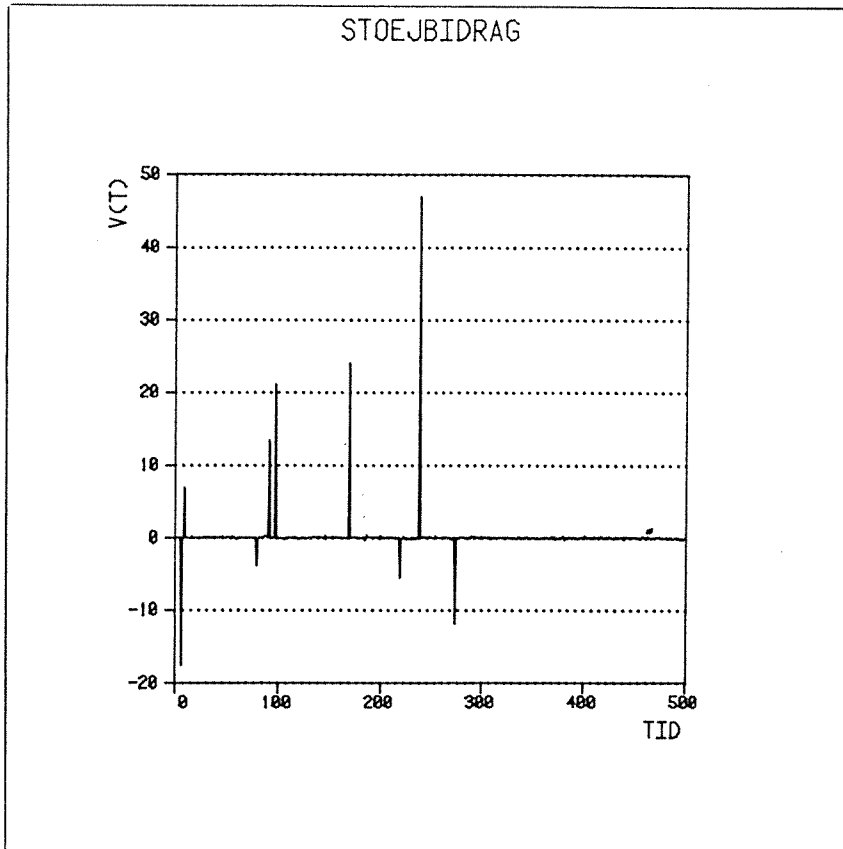


Fig. 9: Den anvendte støjrealisation.

påvirkes af en spids, vil reguleringskvaliteten være kraftigt nedsat. For adaptive regulatore, hvor estimatet af $d(t)$ anvendes i reguleringen, vil der opstå temmeligt store tab som følge af DC - fejlen.

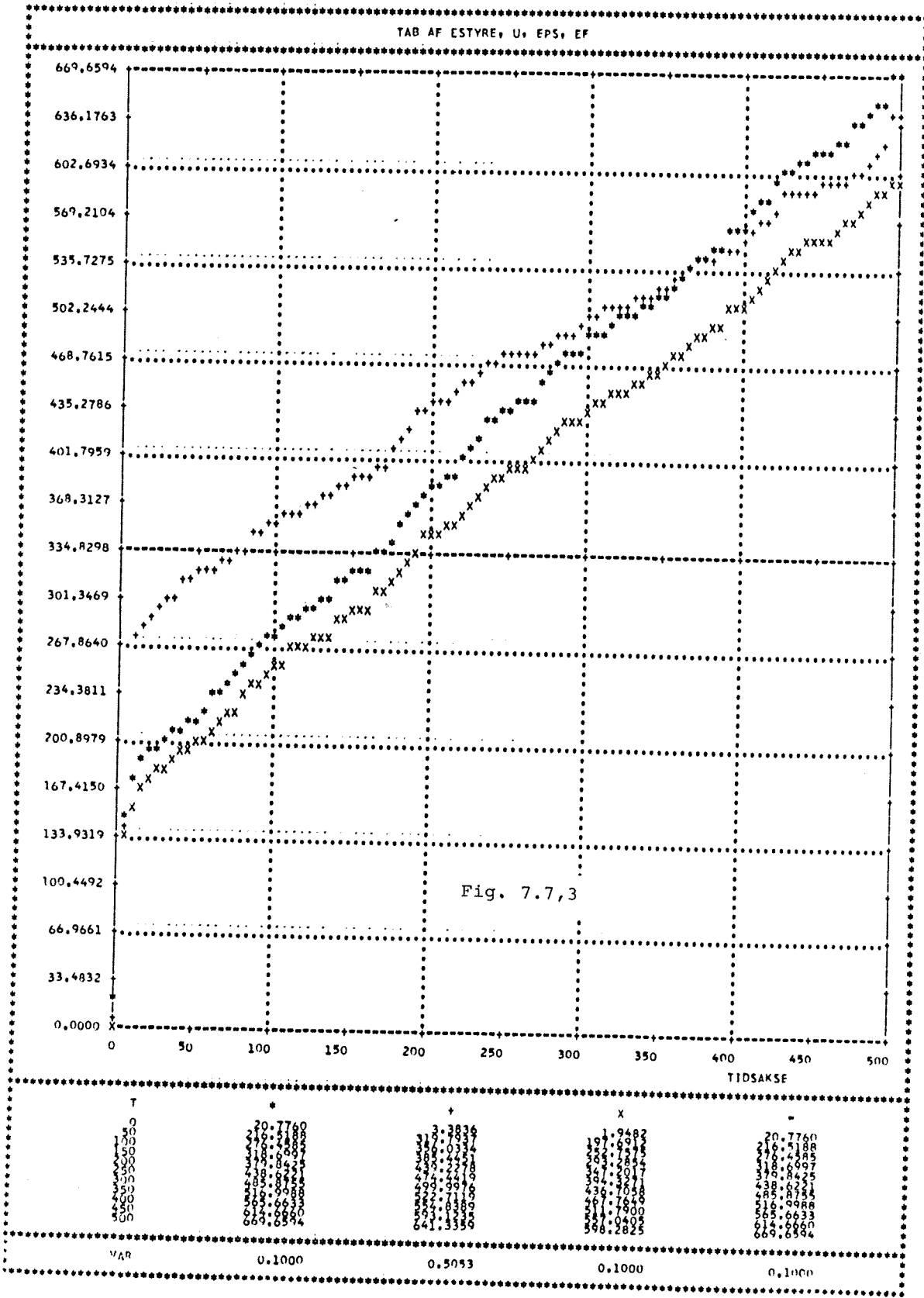
Hvis mistilliden til store residualer slækkes, således at residualerne anvendes i begrænset omfang til opdatering af parameterestimerne, fremkommer RI2 - algoritmen. Anvendes denne algoritme, når objektet udsættes for testsekvensen, opnås også meget fine resultater. Det fremgår af fig. 16 og til dels også af fig. 17, at med denne algoritme er den adaptive regulator mere følsom overfor spidser i opstartsperioden. Dette skyldes, at selvom de to spidser i $t = 6$ og i $t = 9$ bliver begrænset, driver de dog parameterne et stykke, som det fremgår af fig.18, hvor de estimerede F - parameter er angivet.

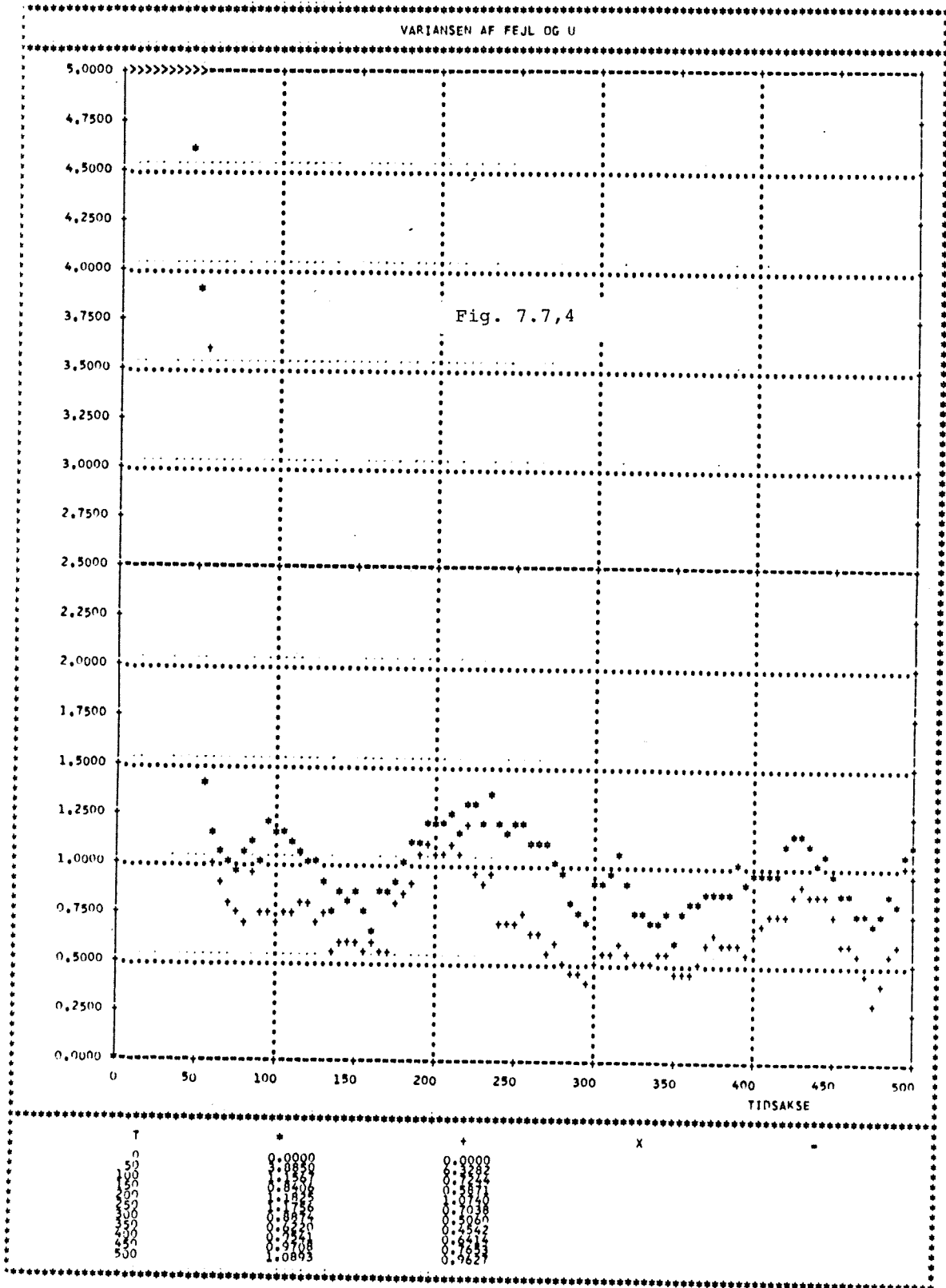
RI1 - algoritmen er desværre ikke blevet tilstrækkelig under -

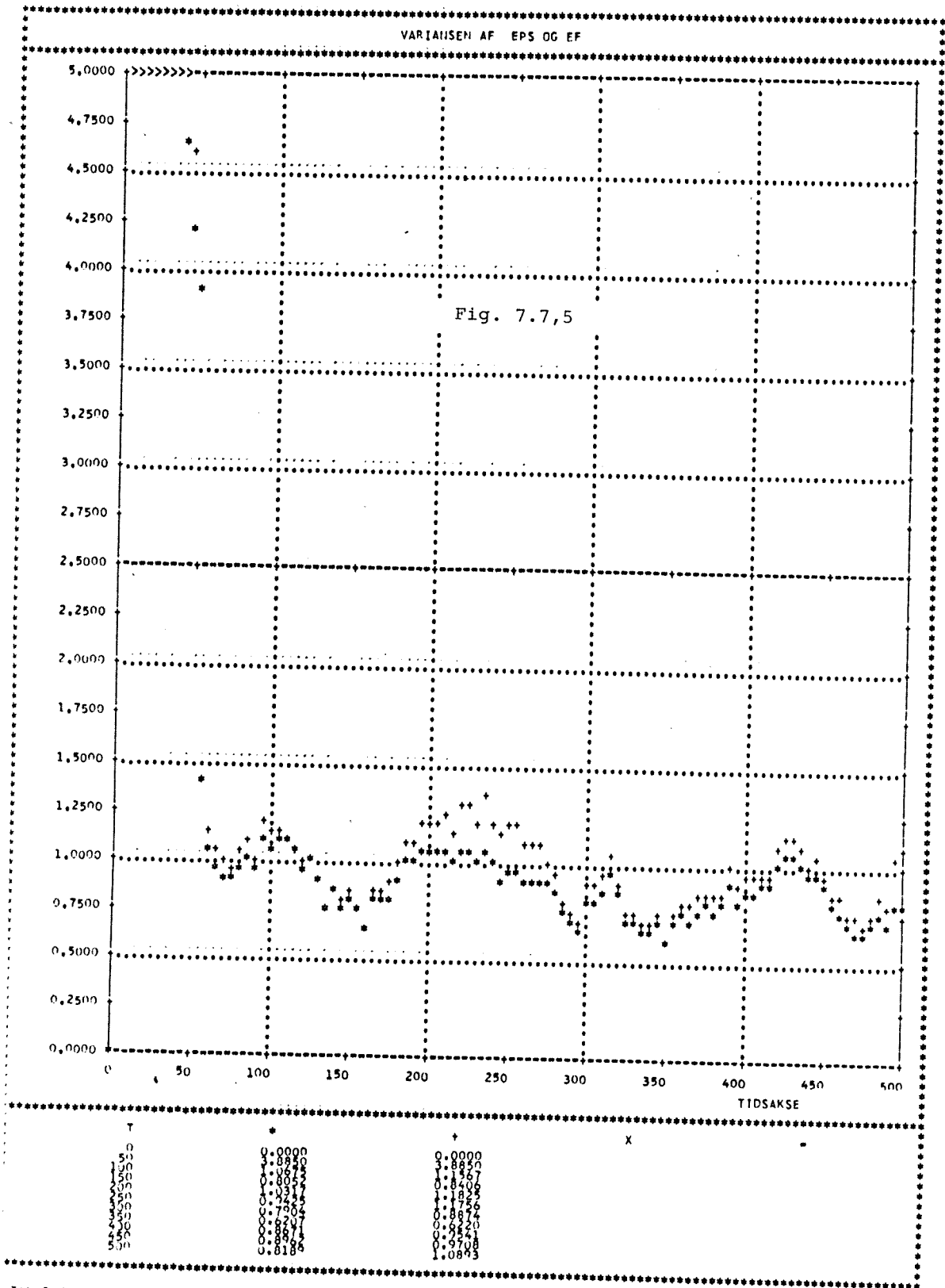
søgt. Nogle få simulationer tyder på, at denne algoritme er væsentlig mere følsom overfor spidser end de to forrige. Diskontinuiteten i forstærkningen, $K(t)$, er sandsynligvis årsag hertil.

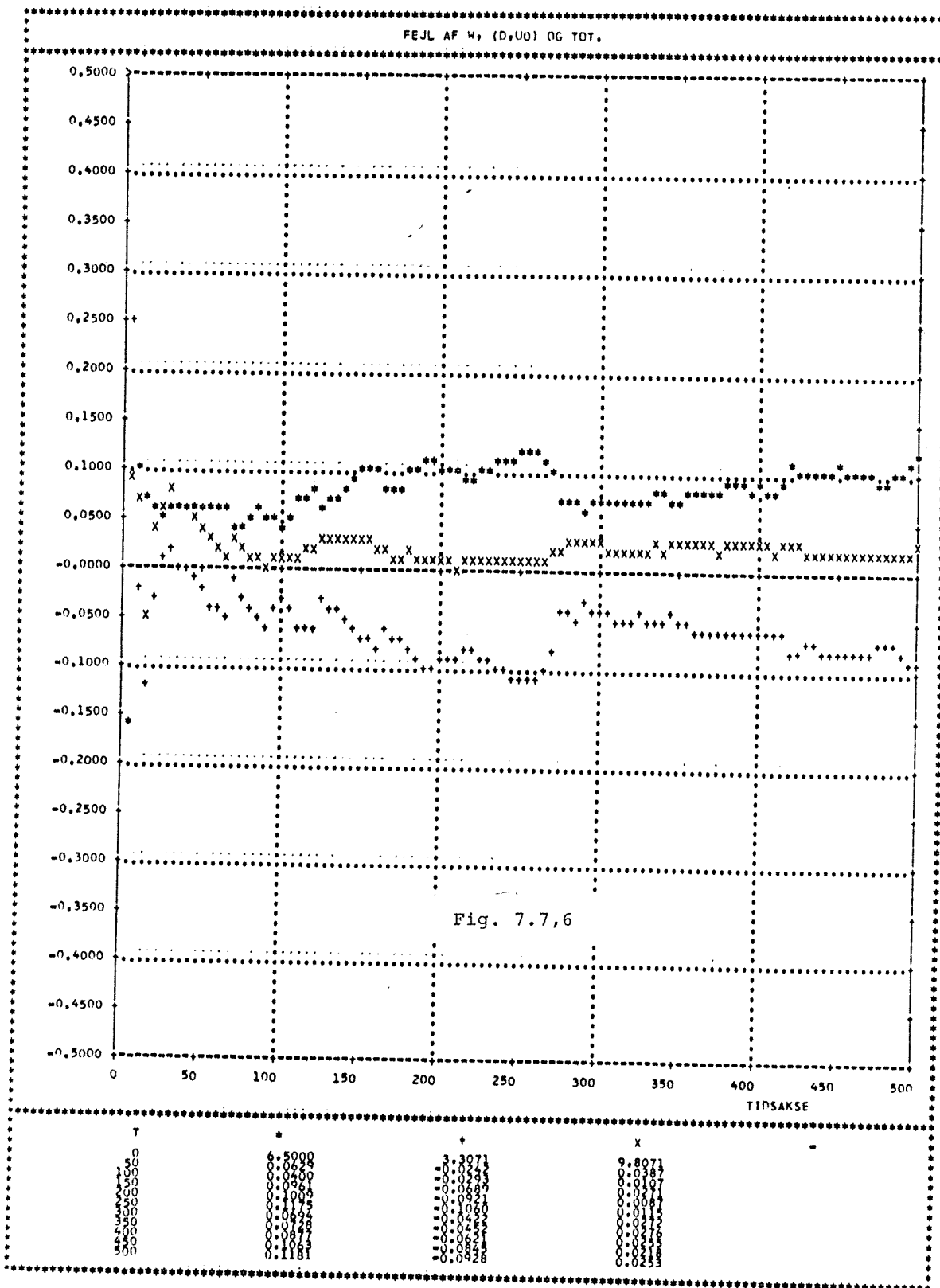
På et ret sent tidspunkt i projektet blev der fundet problemer med stabiliteten, når tidsforsinkelsen gennem objektet blev ændret til $k = 3$. Dette hænger nøje sammen med de to meget markante spidser i starten af simulationen, hvilket øjensynligt er temmeligt uheldigt for både RI2 og RI3. Hvis disse spidser bliver forsinket 20 tidsenheder forsvinder problemet helt for begge de to nævnte algoritmer. Når $k \neq 1$ vil residualerne fra den brugte MVO-regulator være korreleret for korrekte estimater. Dette er i modstrid med forudsætningerne for metoderne. Selvom dette opstartsproblem ikke er undersøgt fuldt ud, kan det tænkes, at fænomenet kan undgås ved at estimere paramterne i den glidende middelværdi med ordenen $k-1$, som RLS-residualerne udgør. Anvendes en RELS- eller en RML-algoritme med ovenstående udvidelser, vil de derved fremkomne residualer være ukorrelerede for korrekte estimater. At denne løsning er mulig understreges af, at fænomenet ikke optræder i forbindelse med de eksplicitte deadbeat algoritmer, hvor residualerne er ukorreleret for korrekte estimater uafhængig af at $k \neq 1$. At det er korreleret residualer, der er årsag til problemet, underbygges også af detaljeret undersøgelser af estimationsalgoritmernes signaler og størrelser i startperioden.

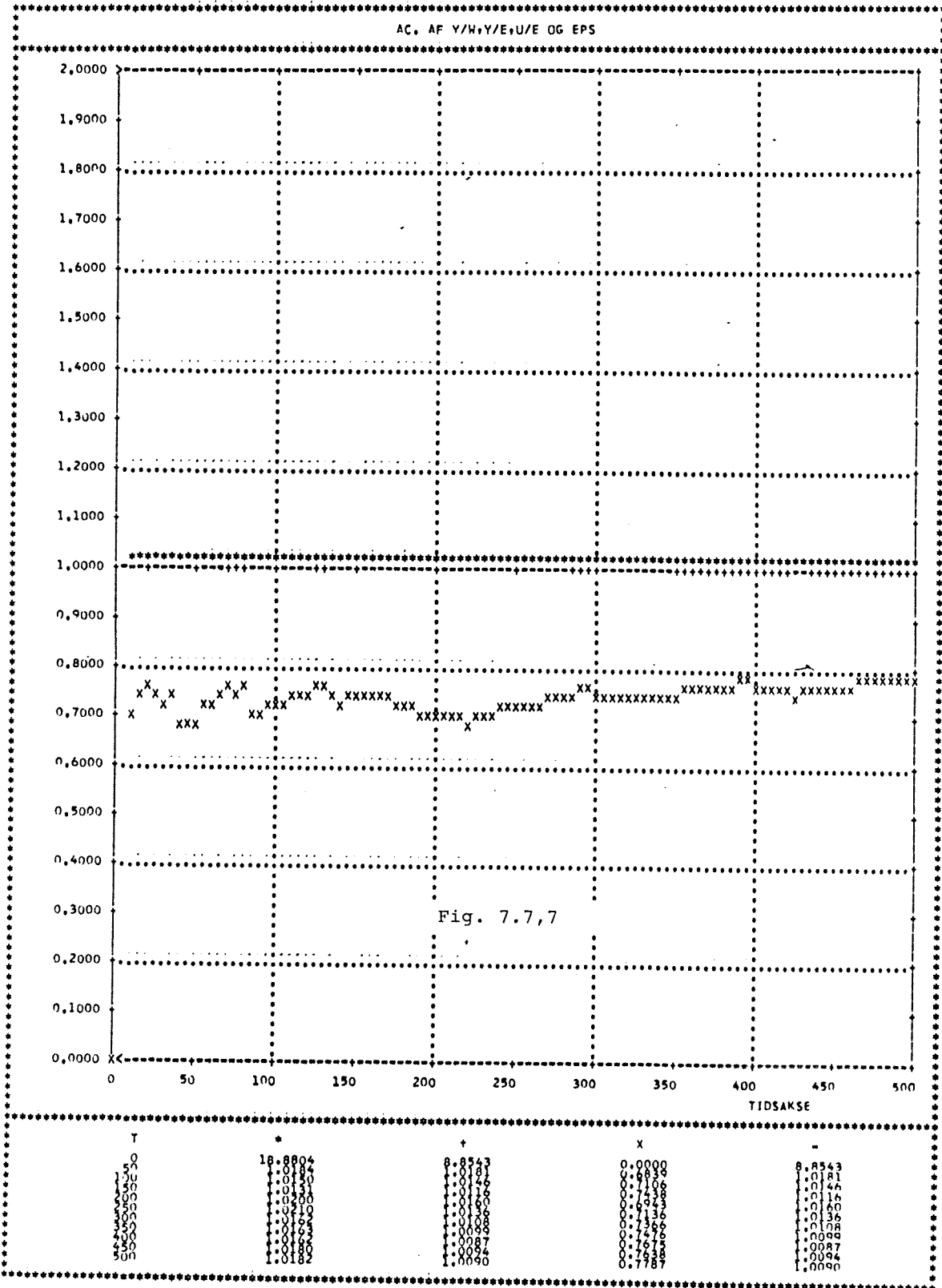
Konklusionen er, at de to afprøvede algoritmer har meget fine egenskaber overfor spidser. For de implicitte regulatorer, hvor residualerne ikke er ukorreleret, eksisterer der situationer, hvor spidser i startperioden kan give problemer. Dette problem er dog ikke undersøgt til bunds.

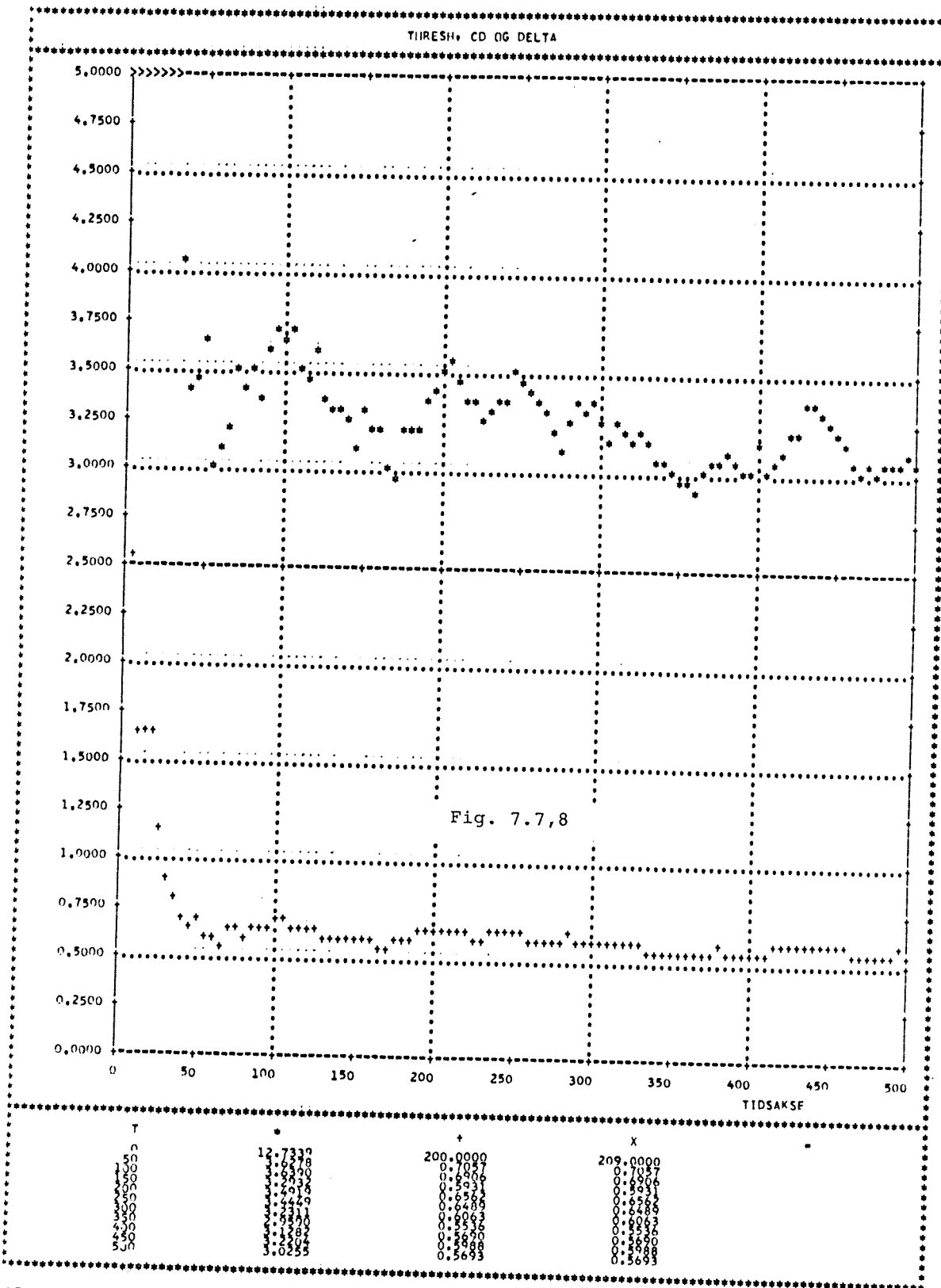












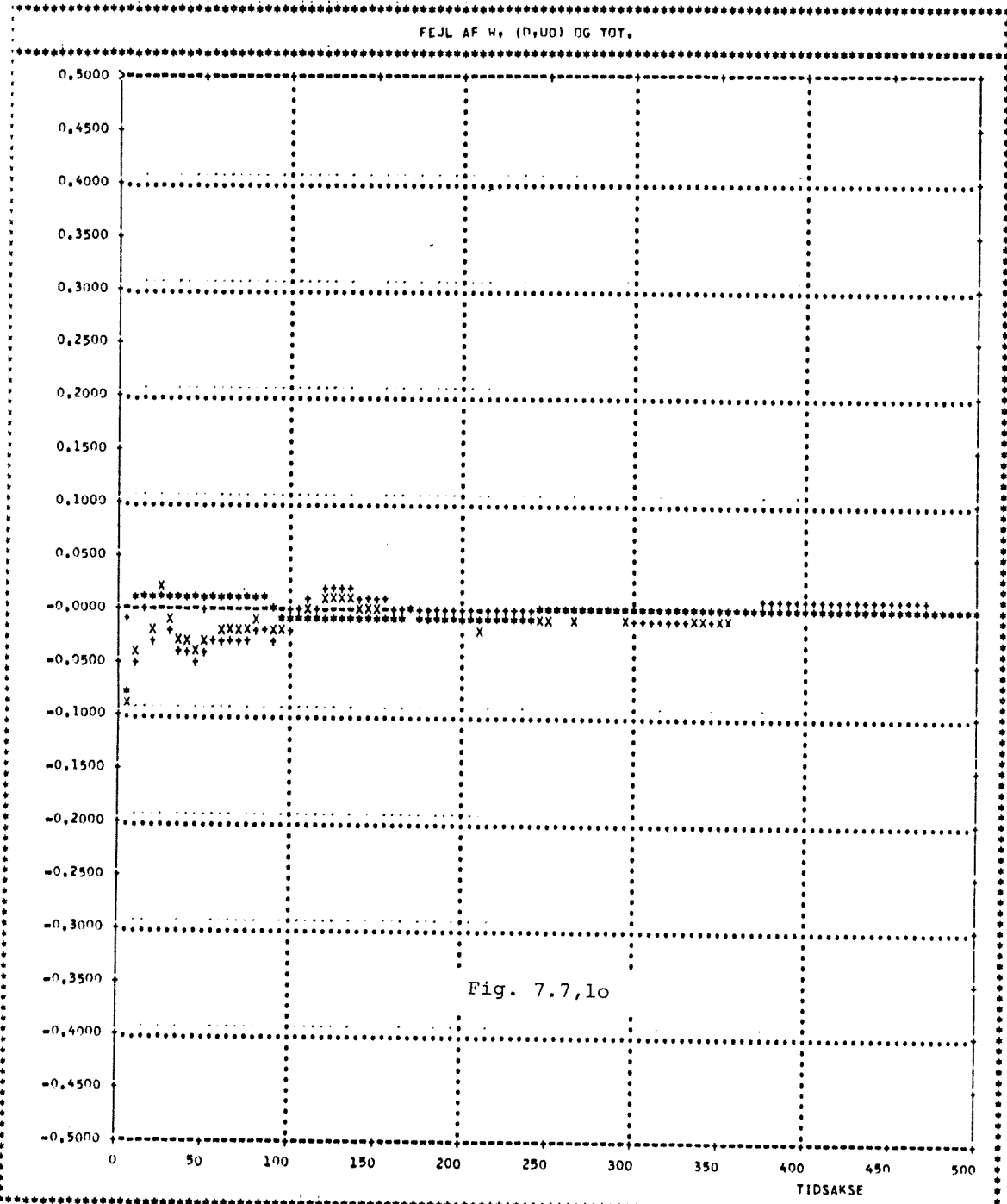
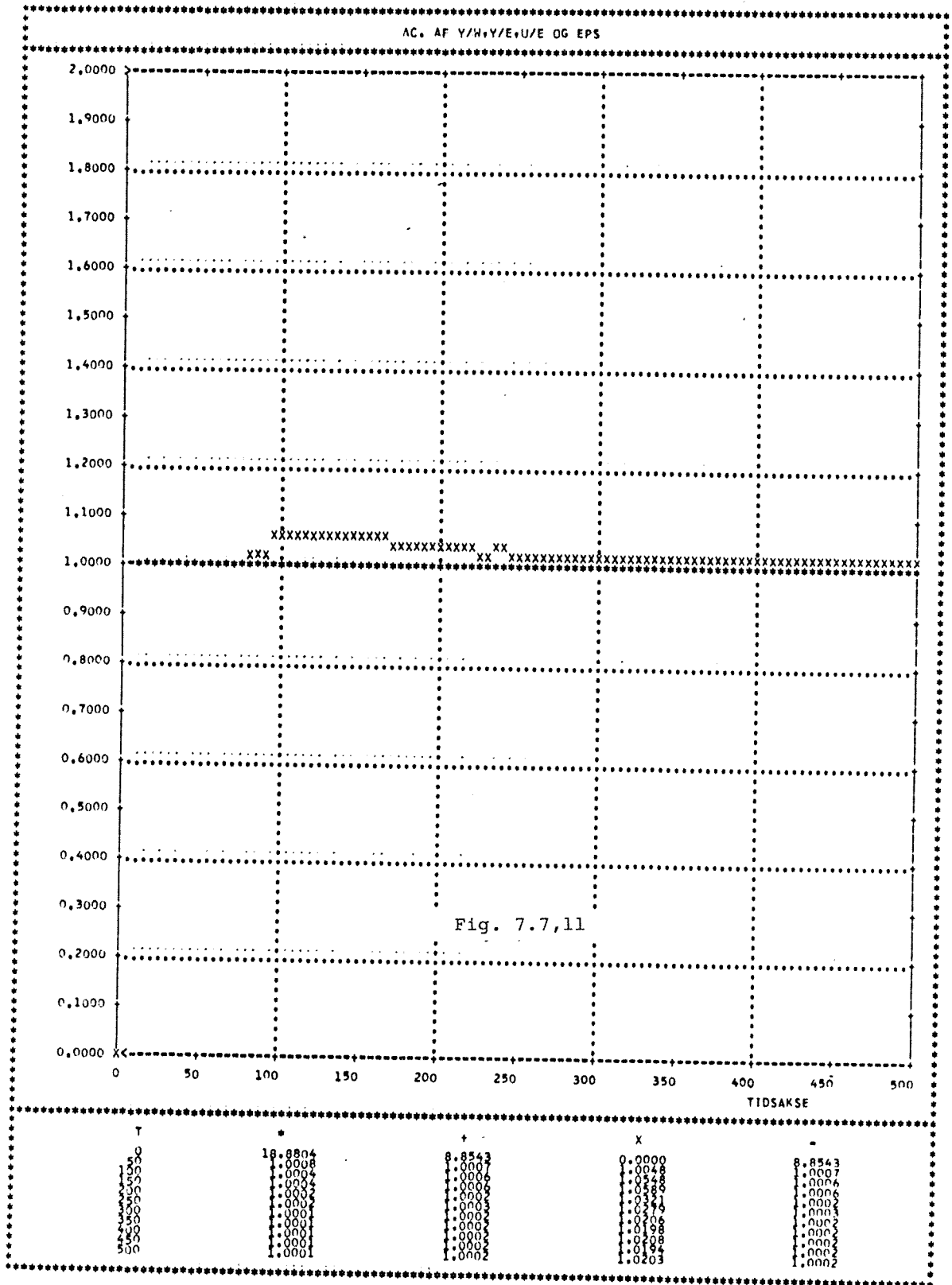
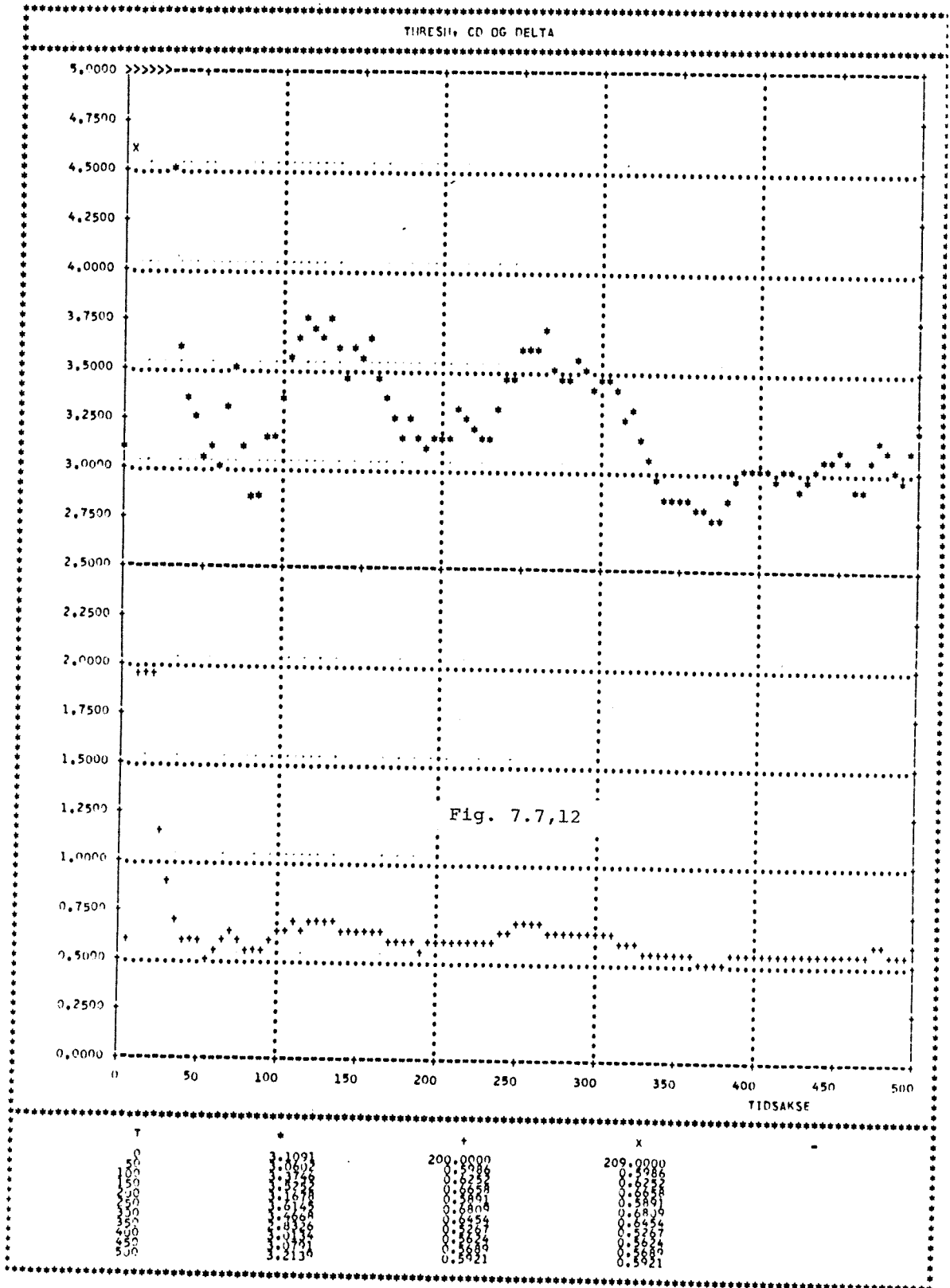
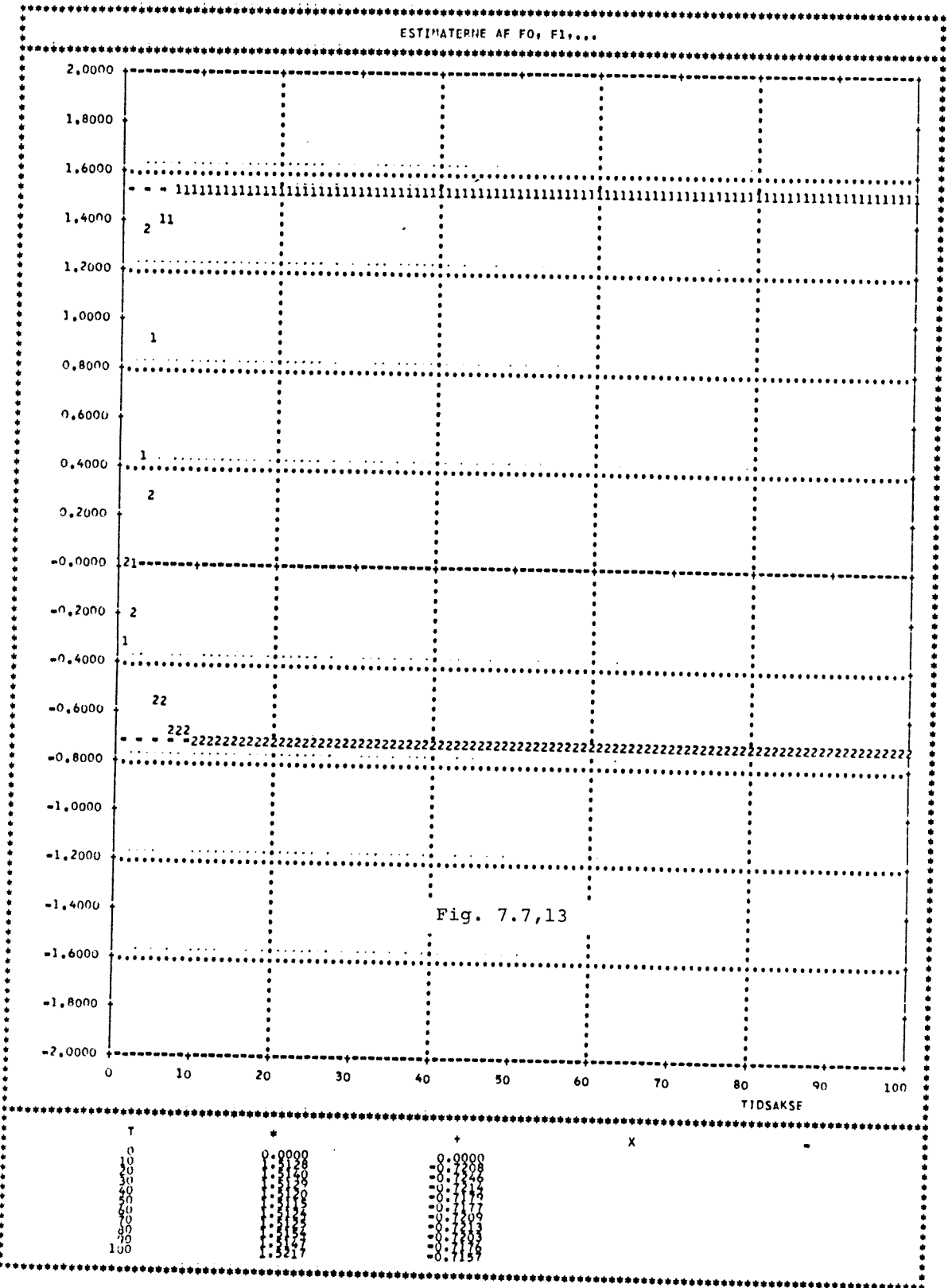


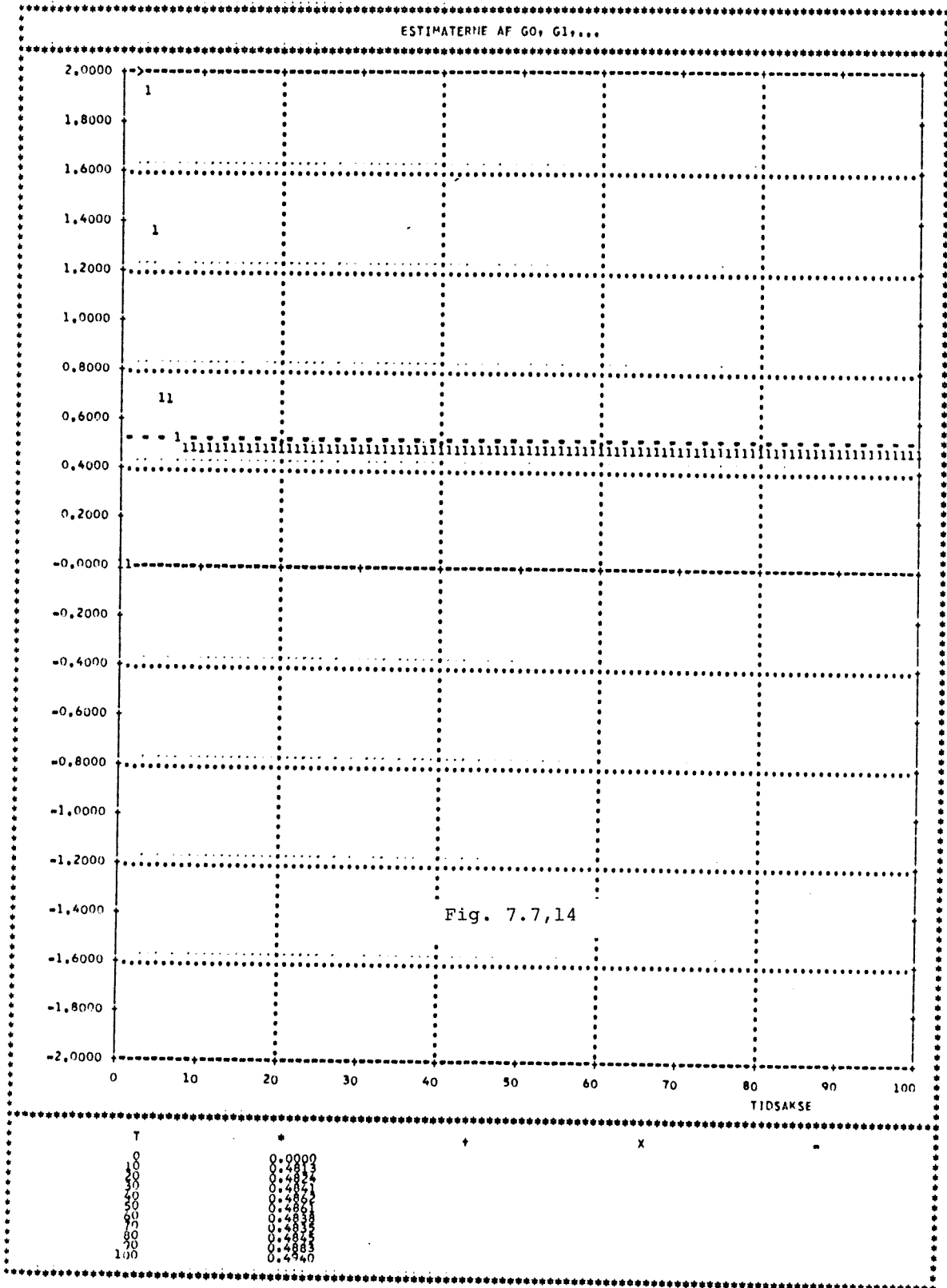
Fig. 7.7,10

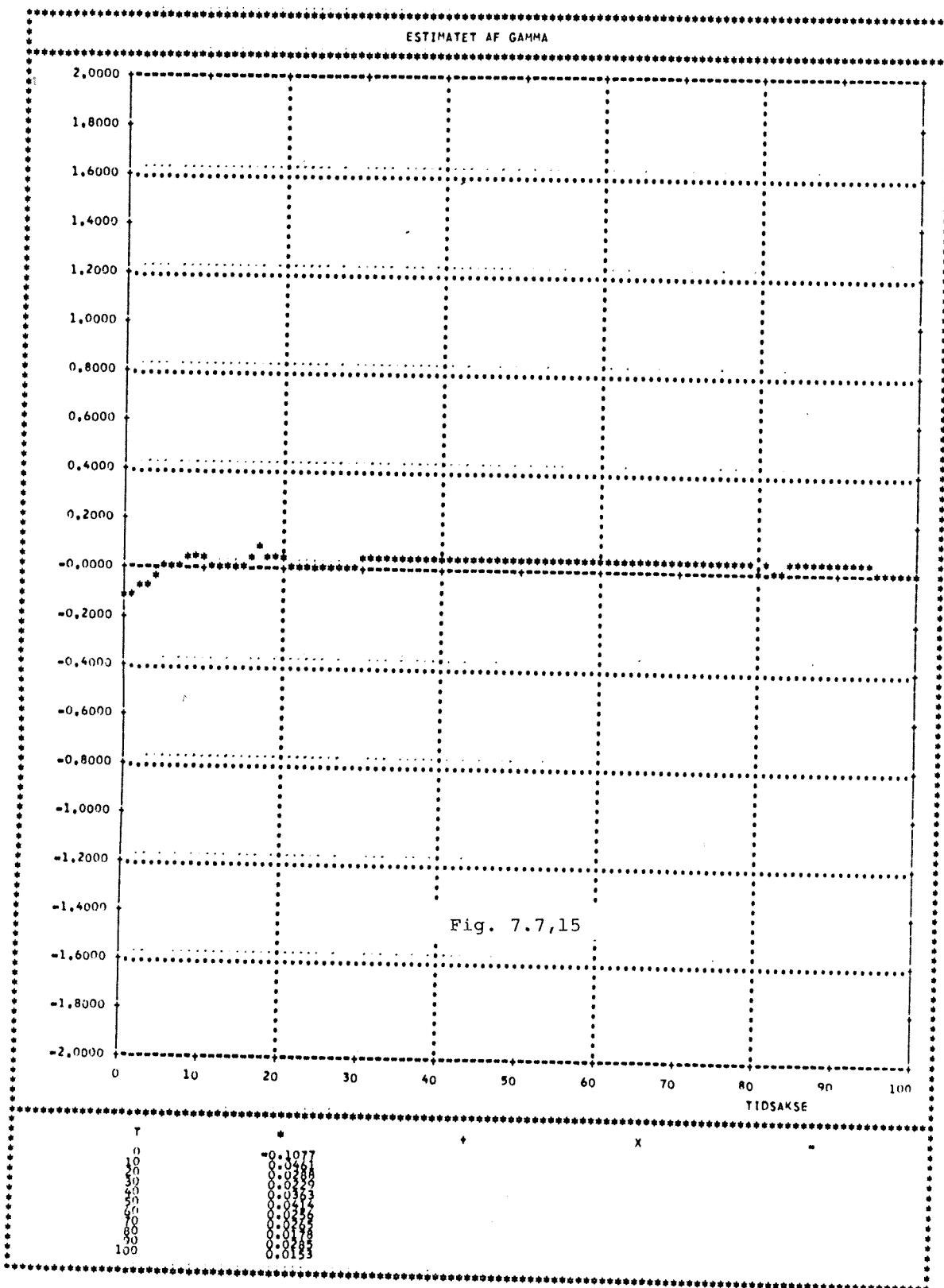
T	+	-	X
0	6,5000	0,8075	7,3075
50	-0,0081	-0,0417	-0,0337
100	-0,0052	-0,0152	-0,0203
150	-0,0113	-0,0103	-0,0004
200	-0,0088	-0,0038	-0,0128
250	-0,0041	-0,0029	-0,0188
300	-0,0045	-0,0038	-0,0183
350	-0,0042	-0,0037	-0,0080
400	-0,0043	0,0089	0,0039
450	-0,0041	0,0061	-0,0020
500	-0,0043	0,0013	-0,0033

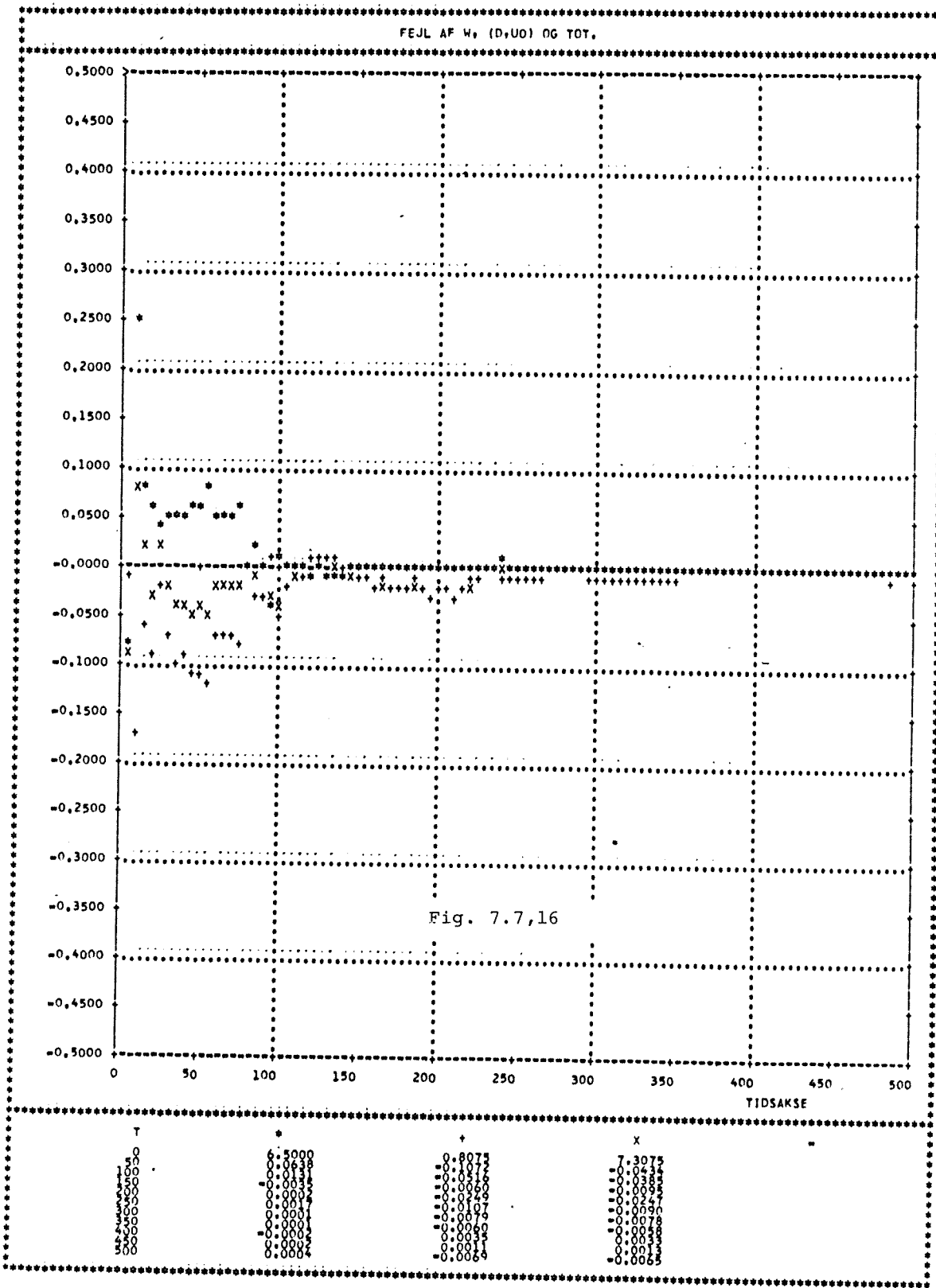


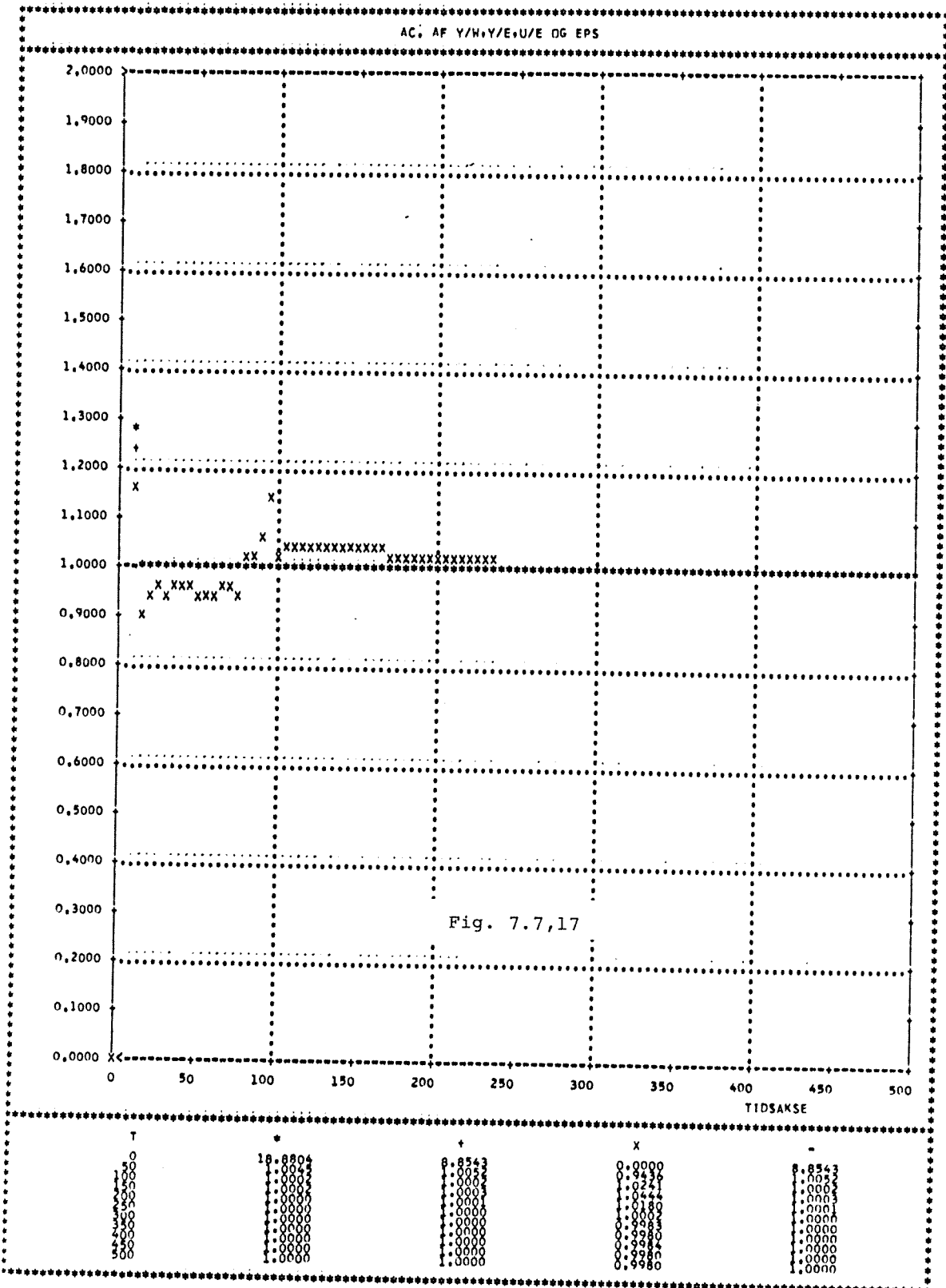


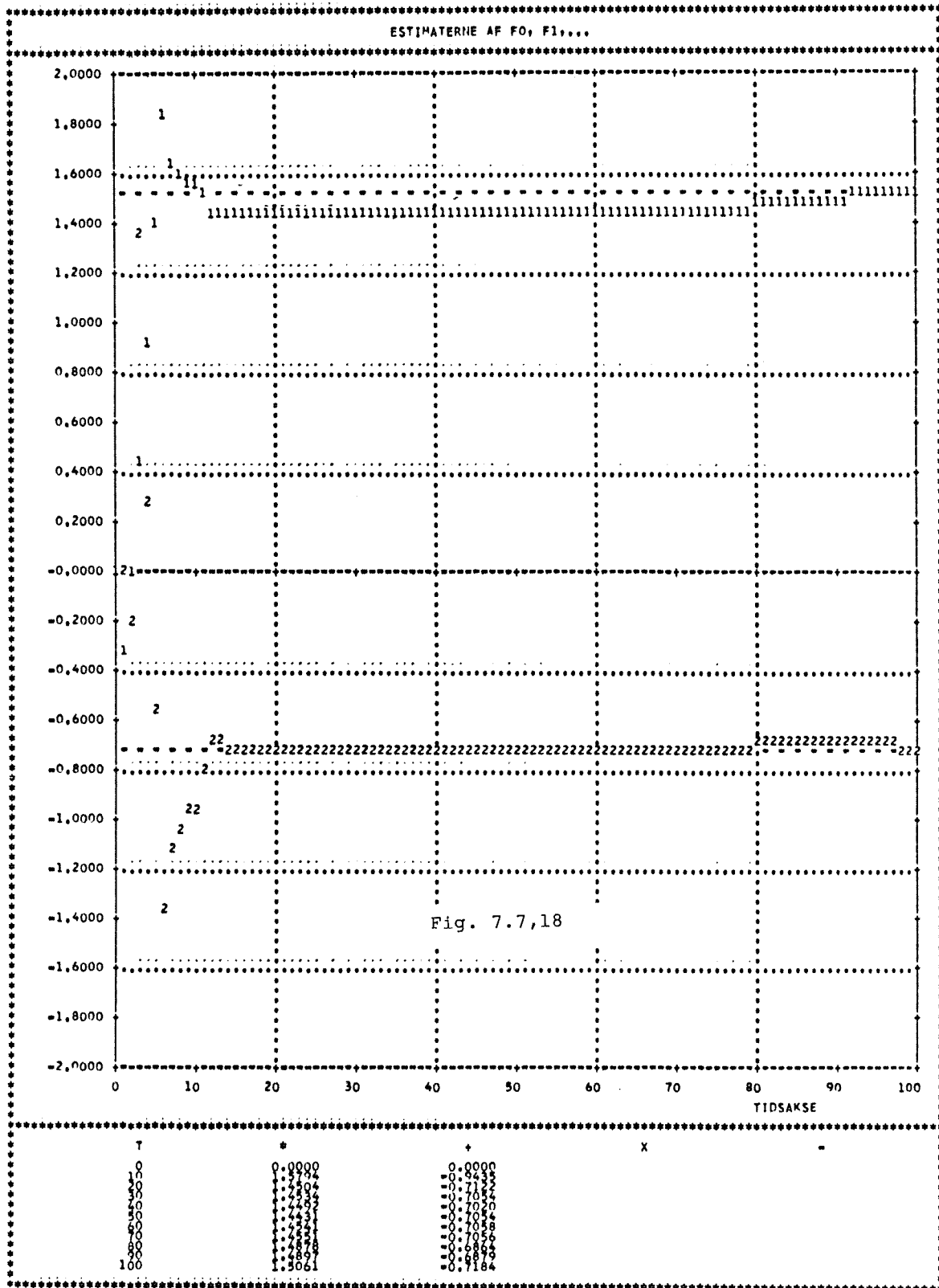












7.8 Robust Adaptiv Regulering.

Dette afsnit indeholder en beskrivelse af de simulationer, der er foretaget med forskellige adaptive regulatorer, når forstyrrelserne til objektet

$$(1) \quad A(q^{-1})y(t) = q^{-k} B(q^{-1})u(t) + v(t)$$

er på den generelle form

$$v(t) = C(q^{-1})e(t) + d(t)$$

$$(2) \quad e(t) = (1 - \delta_1(t)) e_0(t) + \delta_1 e_1(t)$$

$$d(t) = d(t-1) + \delta_2 e_2(t) + \delta_3 e_3(t)$$

I simulationerne er driften repræsenteret i (2) ved $e_3(t)$ kun blive kombineret med spidser. Endvidere vil driften blive simuleret simpelt ved en rampefunktion. De øvrige forstyrrelsesbidrag, $e_0(t)$, $e_1(t)$ og $e_2(t)$, bliver simuleret som en sekvens af uafhængige $N(0, \sigma_i^2)$ fordelte stokastiske variable, hvor:

$$\sigma_0^2 = 0.1$$

$$(3) \quad \sigma_1 = 25.0$$

$$\sigma_2 = 5.0$$

Overgangssandsynlighederne, s_1 og s_2 , er for δ_1 og δ_2 henholdsvis 2% og 1%. Realisationen for tilfældet $\sigma_3 = 0$ er vist i fig.1. Testobjektet er ligesom i de øvrige afsnit objekt I, der er givet ved:

$$A(q^{-1}) = 1 - 1.5q^{-1} + 0.7q^{-2} \quad C(q^{-1}) = 1$$

$$B(q^{-1}) = 1 + 0.5q^{-1} \quad k = 1$$

De enkelte selvindstillende regulatorer, der er anvendt i det følgende, er beksrevet i afsnit 6.1.

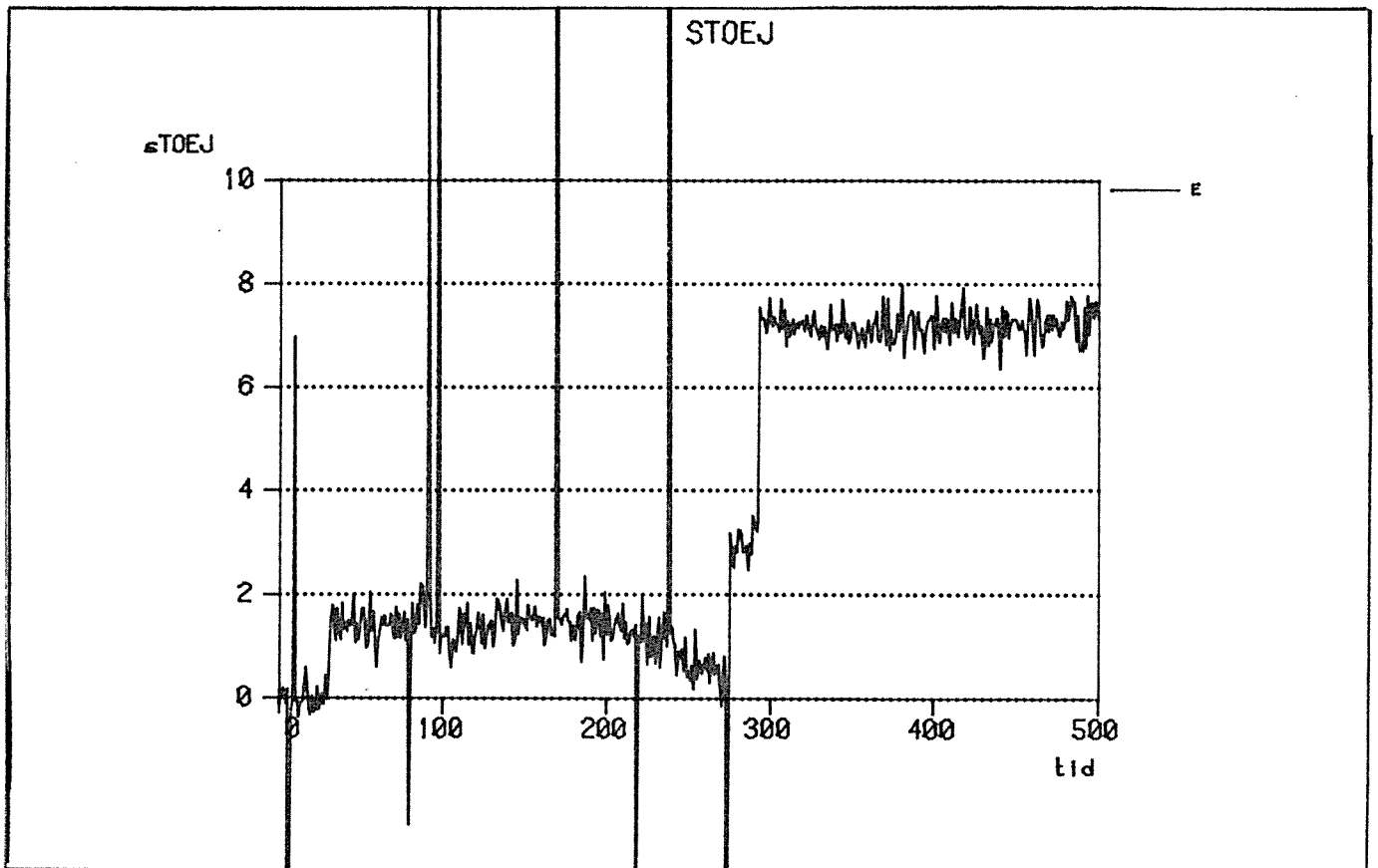


Fig. 1: Støjrealisationen for $\sigma_3 = 0$.

Den endelige estimations algoritme, der ikke er specifikt dimensioneret til de enkelte selvindstillende regulatorer, består som antydnet med formålet i afsnit 7.7 af en kombination af en detektor og en robust identifikationsalgoritme. Der er i det følgende valgt at anvende RI3. Algoritmen har to arbejdsstillerstande. I den normale udføres en robust estimation med en stationær forstærkning i kalmanfilteret, der gennem (7.7,23) er bestemt af:

$$(5) \quad R_1 = \Gamma \Gamma^T Q_d^0$$
$$\Gamma = (0, \dots, 0, 1)^T$$

Som nævnt i afsnit 7.6 skal Q_d^0 vælges så lille, at estimations - præcisionen ikke forringes mærkbart. Der er her ligesom i afsnit 7.6 valgt, at $Q_d^0 = 0.005$. I denne tilstand kan det udvidede kalmanfilter opfange mindre drift og små spring, som ikke bliver detekteret. Hvis driften er voldsom eller ved direkte abrupte spring vil detektoren udløse den anden arbejds - tilstand. Denne består i, at algoritmen på basis af lagrede data udregner parameterestimererne, $\hat{\theta}(t)$, som ville have været, hvis de i tidsrummet fra $t-N+1$ til t var estimeret med et udvidet kalmanfilter med en meget stor forstærkning til estimatet af $d(t)$. I simulationerne er Q_d^1 valgt til 100. Hvis der indtræffer en spids, umiddelbart efter et spring er opstået vil dette få begrænset indflydelse på parameterestimererne, når algoritmen er i den normale arbejdstilstand. Men når detektoren som følge af springet aktiverer den unormale arbejdstilstand, vil spidsen indgå med fuld styrke, hvis den indtræffer uheldigt.

Problemet er måske ikke så stort, men der kan forventes så store spidser, at deres indflydelse ikke kan opfanges i estimatet af $d(t)$. I figurene 7.6, 18 - 19 er der vist kurver for et tilfælde, hvor spidsen kan opfanges i estimatet af $d(t)$. Specielt ved anvendelse af lavere værdier for Q_d^1 kan der dog forventes problemer. Hvis der også i denne arbejdstilstand skal anvendes en robust estimator, opstår et problem med valget af grænsen Δ^1 . Der er i projektet forsøgt med en succesiv udvidelse af Δ^1 , indtil kvadratsummen over de N residualer har en passende lav værdi. Denne løsning gav ikke en acceptabel virkning. Man kan forestille sig, at der kan bruges statistiske metoder til at udskille outliers. En sådan metode skal kunne adskille spidser fra de residualer som et

spring giver anledning til. I simulationerne er der anvendt en fast værdi for Δ^1 på 10, hvilket sætter en øvre grænse for hvor store spring, der kan følges. Det eneste krav til valget af Δ^1 er, at dette er stort nok til, at algoritmen kan følge springene. Valget er derfor ikke kritisk, fordi Q_d^1 vælges i overensstemmelse hermed, hvorved spidserne i det kritiske tids - område kan optages i estimatet af $d(t)$, hvis de har en sådan størrelse, at de ikke bliver begrænset.

Prisen, der må betales for at kombinere en detektorløsning med en robust estimator, er at store spring ikke bliver detekteret hurtigere end et spring, der lige netop er større end grænsen, $c(t)$. Dette skyldes, at detektoren kun for tilført de be - grænsede residualer, $\varepsilon_2(t)$, som også er tilfældet for estima - toren $Q(t)$ j.v.f afsnit 7.7. Til gengæld vil der ikke, når der som her bruges RI3, blive reguleret med fejlagtige dynamiske parameter (alle undtagen estiamtet af $d(t)$). I fig. 2 er den samlede struktur illustreret. Numrene i blokkene svarer til formeludtrykkene, der beskriver blokkene.

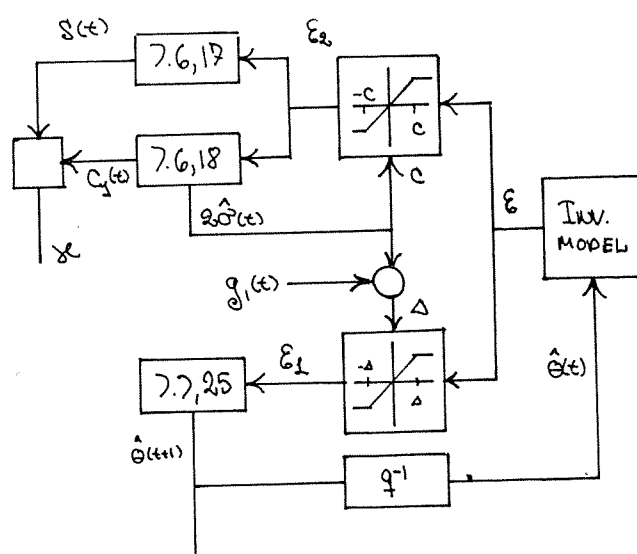


Fig. 2: Samlet struktur for RI3 og detektor.

Drift.

Først undersøges MVO - regulatoren, når objektet bliver ud - sat for forstyrrelser fra normal støj, $e_o(t)$, spidser samt drift. Driften er simuleret med den før anvendte rampefunktion, der starter i $t = 40$ og har en hældning på 0.01 . Belastningen er for $t > 140$ konstant lig 1 . På grund af begrænsningen af residualerne er estimatet, $Q(t)$, mindre end i det urobuste tilfælde, hvorfor detektoren er mere følsom. Dette giver sig til udslag i, at detektoren aktiverer den unormale arbejdstilstand til $t = 90$ og $t = 139$. I fig. 3 - 5 er vist forløbet af DC - fejlen, AC - forstærkningen samt estimatet af $d(t)$. Ses der bort fra de noget store oversving i $\hat{y}(t)$, følger estimatet ret godt med $d(t)$. Det ses, at såfremt algoritmen skal bruges overfor større drift, skal Q_d^1 vælges mere fornuftig. En håndregel er, at vælge Q_d^1 ud fra et a priori estimat af det maksimale spring, som $d(t)$ med rimelig sansynlighed kan forventes at foretage. Ud over dette må det siges, at den robuste estimator arbejder fint sammen med de foregående løsninger.

Spring.

For at undersøge spidsernes indflydelse på reguleringen, når disse ankommer sammen med et spring, er der udført samme eksperiment som referet i afsnit 7.6. Eksperimentet, der her er udført med en regulator af POL - typen (afsnit 6.1.2), består i at påvirke objektet med en spids på 7 til $t = 41$ og med et enhedsspring til $t = 40$. Desuden påvirkes objektet af støjsignalet $e_o(t)$. Regulatoren, der er dimensioneret med

$$A^m(q^{-1}) = 1 - 0.398q^{-1} + 0.1353q^{-2}$$

$$B^m(q^{-1}) = 0.582 + 0.1552q^{-1}$$

$$T(q^{-1}) = 1$$

er beskrevet i afsnit 6.1.2. Kurverne optaget ved eksperimentet er afbildet i fig. 6 - 8. Ved brugen af robust estimator bliver springet først detekteret i $t = 48$, hvilket naturligvis giver en ringere regulering end hvis det blev detekteret tidligere. Hvilke konsekvenser det får for signalerne fremgår af fig. 6. Da spidsen kun har højden 7 bliver den ikke begrænset i algoritmens unormale arbejdstilstand. Fig. 7 viser, at estimatet af $d(t)$ har et oversving af mindre format. Hvis fig. 8, hvor forløbet af AC - forstærkningen er afbildet, sammenlignes med en tilsvarende (fig. 9) for det tilfælde, at objektet kun påvirkes af hvid støj, $e_o(t)$, vil det bemærkes, at forløbet stort set er uændret bortset fra punktet $t = 46$. Dette skyldes, at parameterestimerne kun bliver opdateret i $t = 43$ og i $t = 46$ i tidsrummet mellem springet opstår og det bliver detekteret. Som før nævnt bliver der altså ikke i større omfang reguleret med fejlagtige estimater mellem $T_{j,i}$ og $T_{d,i}$. Det ses også, at spidsen ikke påvirker parameterestimerne i større omfang.

Nu vil objektet, der er reguleret af MYO - regulatoren, blive udsat for testsekvensen, der er afbildet i fig 1. Ind - føres betegnelserne T_{sp} og S_1 for tidspunktet hvori springet fremkommer og dens størrelse, kan sekvensen karakteriseres ved følgende tabel.

T_{sp}	S_p	T_j	d_o	$d(t)$	T_d
6	-17.61	-	-	o	-
9	6.99			o	
		31	1.40	1.40	34
79	- 3.85				
91	13.52				
97	21.24				
169	24.11				

T_{sp}	S_p	T_j	d_o	$d(t)$	T_d
219	-5.57				
238	47.06				
		241	-0.77	0.63	250
		273	-0.39	0.24	-
274	-11.95				
		275	2.70	2.94	281
		293	4.25	7.19	296

Her også tidpunkterne for detektion af spring (T_d) anført. Sammenlignet med forholdene uden spidser er springene som forusagt detekteret senere, hvilket skyldes, at residualerne er begrænset inden de indgår til detektionen. Herved undgås, at en spids udløser detektoren. Fra den genrelle regel afviger det første spring, der faktisk bliver detekteret før i dette tilfælde. Fordi de store residualer, der fremkommer i startperioden kun indgår i begrænset omfang til detektoren og estimatoren, $Q(t)$, vil estimatet af støjspredningen hurtigere svinge ned mod en stationær værdi, der iøvrigt er lidt mindre end uden robusthedsgrænserne. Af tabellen fremgår det, at det mest spændende område er fra $t = 230$ til 330 , hvorfor de følgende undersøgelser er centreret om dette. Signalerne for dette tidsrum er angivet i fig. 10, hvor man ser, at reguleringen er meget tilfredsstillende bortset fra tidspunkter mellem, springen opstår, og de bliver detekteret. Det samme kan konkluderes ud fra kurven over DC - fejlen (fig. 11). AC - forstærkningen er upåvirket af forstyrrelserne. Fig. 13 viser, at estimatet af støjvariansen, σ_o^2 , er rimelig. Tidsvariationerne fremgår tydeligt af kurven over

$$c_1(t) = g_r(t) \sqrt{10 Q(t)}$$

(THRESH , *), der for korrekt estimat skal være 3.39 ($\sigma_o^2 = 0.1$)

I figurene 14 - 23 ses, at disse forhold ikke er et specielt tilfælde for MVo - regulatoren. Udsættes objektet for den samme forstyrrelse og reguleres det af henholdsvis en POL -, MV2 -, DBo - samt en DBI - reguletor kan kurver for samme tidsrum (230 - 330) optages og sammenlignes. Som ved den foregående omtale er POL - regulatoren dimensioneret med:

$$A^m(q^{-1}) = 1 - 0.398q^{-1} + 0.1353q^{-2}$$

$$B^m(q^{-1}) = 0.582 + 0.1552q^{-2}$$

$$T(q^{-1}) = 1$$

Kurverne optaget med denne regulering er afbildet i fig 14 - 16. Sammenlignet med MVo - regulering er forholdene nogenlunde de samme. MV2 - regulatorens kriteriefunktion er bestemt af polynomierne:

$$R_1(q^{-1}) = 0.582$$

$$P_1(q^{-1}) = 1 - 0.398q^{-1} + 0.1353q^{-2}$$

$$P_2(q^{-1}) = 1 + 0.267q^{-1}$$

$$Q_1(q^{-1}) = 0$$

og er ansvarlig for kurverne i fig. 17 - 19. Disse kurver er lidt bedre ved MVo - regulering.

En sammenligning mellem MVo - og DBo - regulering (fig. 20 21) vil også i dette tilfælde falde ud til fordel for MVo - regualtoeren. Som i afsnit 6.1.3 ser man også her, at AC - forstærkningen for $H_{y,e}$ er lavere end den teoretisk forventede med en DBo - regulator. Konvergens synes ikke særlig god. DBI varianten, der er dimensioneret, sådan at det ekstra ind -

førte nulpunkt er beliggende i -1.5 for korrekte parameterestimer, har i dette tilfælde bedre konvergens, hvilket fremgår af fig. 23.

For at undersøge regulatortypernes indvirkning på robusthedsforholdene er der i nedenstående tabel opført detektionstidspunkterne, T_d , samt antallet af begrænsede residualer over 500 simulationstrin.

T_d	31	241	273	275	293	antal begrænsede residualer
MVo	34	250	-	281	296	56
DB1	35	251	-	281	295	57
DBo	35	248	-	281	296	59
MV2	35	250	-	281	296	
POL	34	250	-	281	296	55

Ud fra denne tabel kan der ikke med større sikkerhed siges, at der er forskel i regulator algoritmernes robusthedsegenskaber overfor spidser og spring. Der selvfølgelig en betydelig forskel i deres evne til at regulere disse forstyrrelser bort samt en forskel i deres konvergens egenskaber.

Undersøgelserne vil nu blive suppleret med et andet testobjekt. I det kontinuere tidsdomæne er objektet givet ved overføringsfunktionen:

$$H_{y,u}(s) = \frac{(1-4s)}{(1+4s)(1+10s)}$$

Obejektet, der er et ikke minimum - fase objekt, kan, hvis der anvendes en samplingperiode på 2 sek., beskrives i det tidsdiskrete område ved:

$$A(q^{-1}) = 1 - 1.425q^{-1} + 0.496q^{-2}$$

$$B(q^{-1}) = -0.102 + 0.173q^{-1} \quad k = 1$$

Polerne for objektet er beliggende i

$$q_p = 0.71 \pm j 0.108 = 0.721 \angle 8.6^\circ$$

mens nulpunktet ligger udenfor enhedscirklen:

$$q_n = -1.7$$

Dette objekt kan derfor ikke styres af en regulator, der plomberer nulpunkterne. Antages det at:

$$C(q^{-1}) = 1 \quad \text{og} \quad \sigma_o^2 = 0.1$$

vil $e_o(t)$ alene give anledning til størrelserne:

$$\text{Var}(u) = 10.22$$

$$\text{Var}(y) = 1.31$$

når objektet reguleres af en DB1 - regulator ($\beta = 1$), der har korrekt justeret parameter. Dette skal sammeholdes med, at ureguleret er

$$\text{Var}(y) = 1.43$$

Den pågældende støjreduktion er øjensynlig temmelig beskeden. Udsættes dette system bestående af objekt og regulator:

$$D(q^{-1}) = \frac{\hat{q}_o (1 + \hat{a}_1 q^{-1} + \hat{a}_2 q^{-2}) (1 - \hat{a}_1 q^{-1})}{1 - q^{-1} \hat{q}_o (\hat{b}_0 + \hat{b}_1 q^{-1}) (1 - \hat{a}_1 q^{-1})}$$

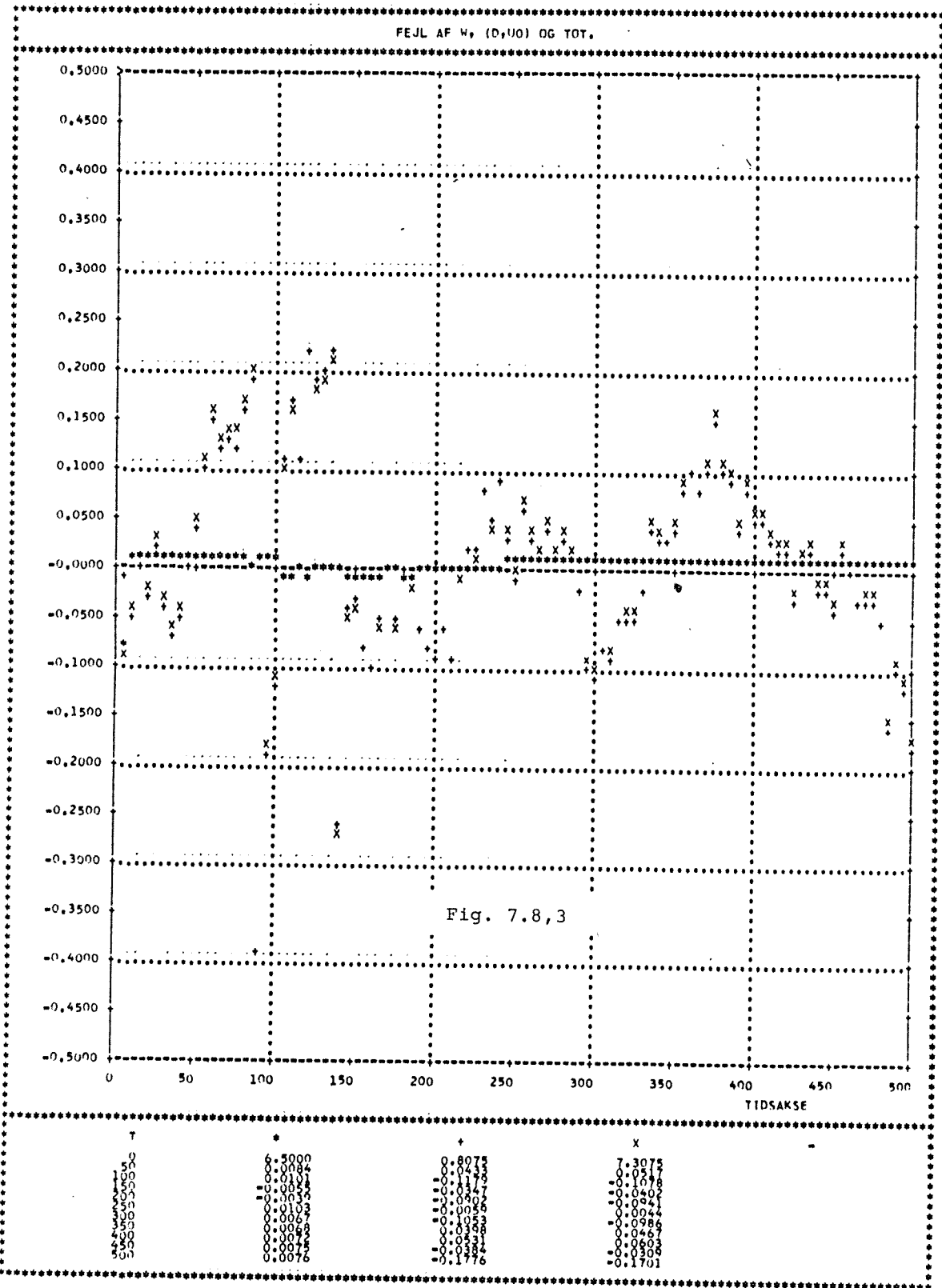
hvor

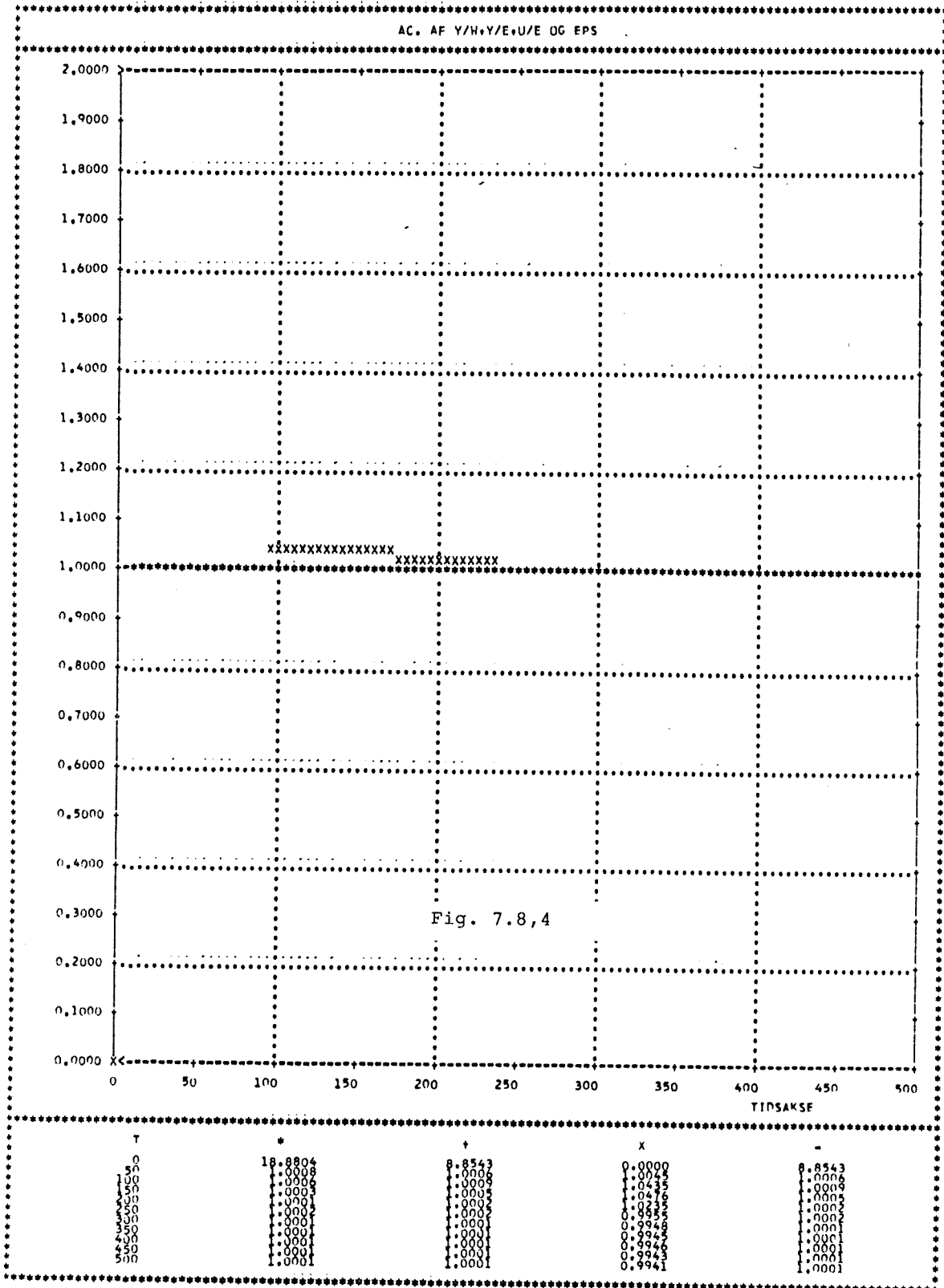
$$\hat{q}_0 = \frac{1}{(1 - \hat{a}_1)(\hat{b}_0 + \hat{b}_1)}$$

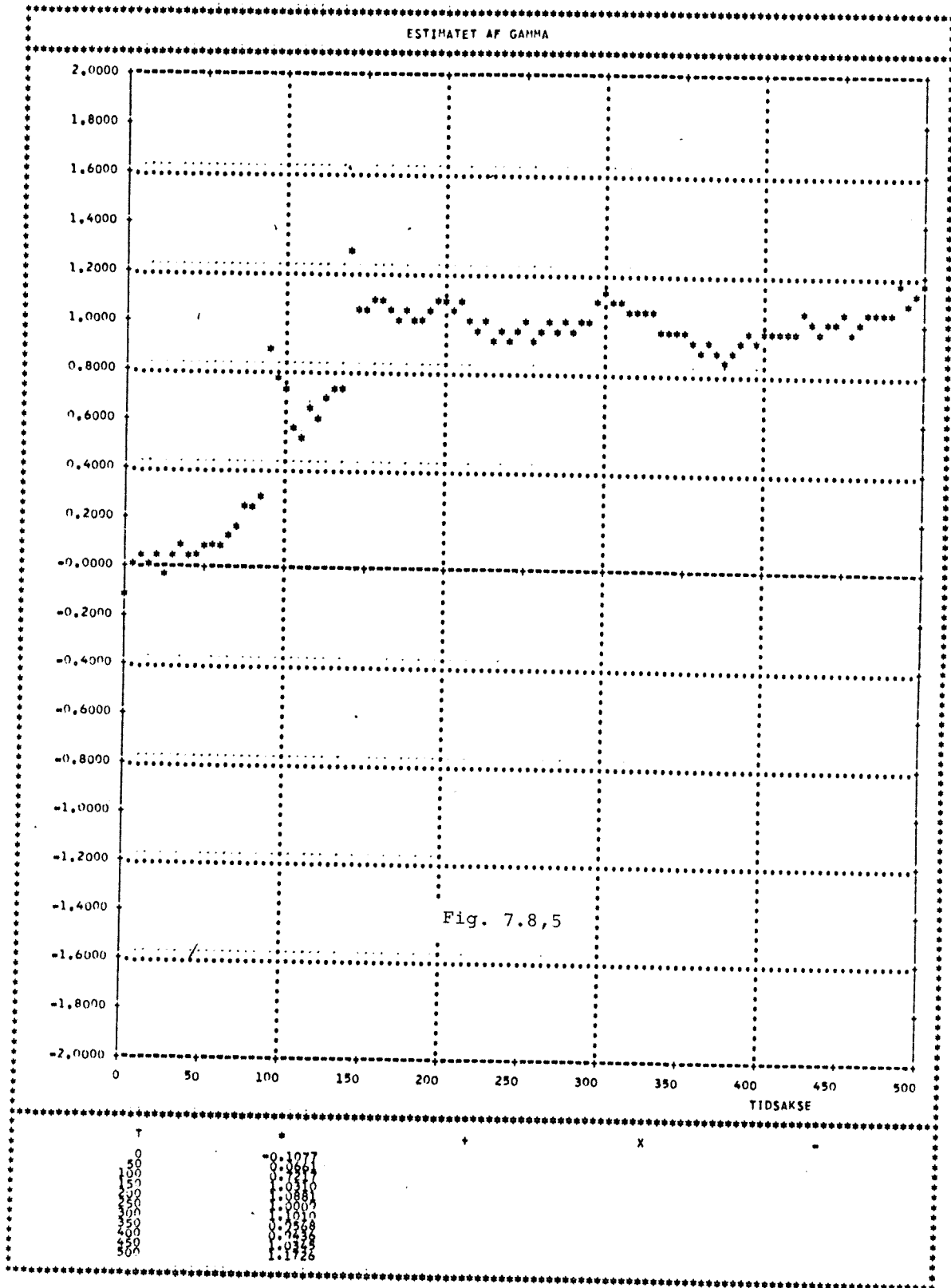
for den anvendte testsekvens af spidser og spring fås signalerne, der er afbildet i fig. 24. Parameter estimaterne er afbildet i fig. 25 og 26, hvor det bemærkes, at estimaterne er næsten upåvirket af de kraftige spring og spidser, der forefindes i det fremtrukne tidsrum (230 - 330).

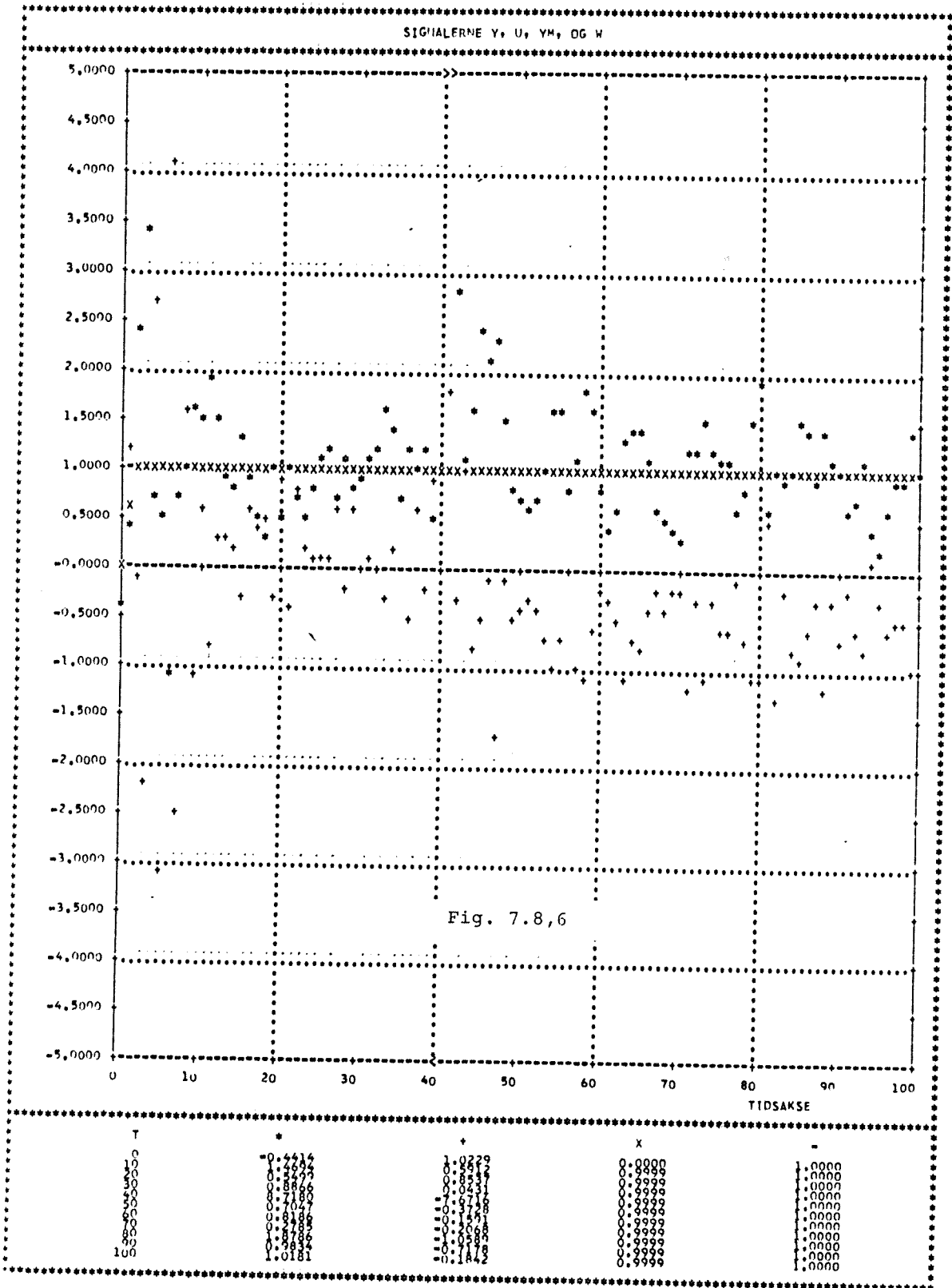
Ud fra de foretagne simulationer, hvoraf en del er dokumenteret i det ovenstående, kan det konkluderes, at den her undersøgte metode, der er en kombination af detektorløsningen og RI3, har meget fine egenskaber overfor den brugte testsekvens af spring og spidser. Simulationen foretaget med rampefunktionen antyder, at dette også er tilfældet overfor drift og spidser. Når $k \neq 1$ vil residualerne for de implicitte algoritmer være en glidende middelværdi, hvorfor forudsætningen for detektoren samt den robuste parameterestimator ikke er til stede. I projektet er der ikke forsøgt, at estimere parameterne i den glidende middelværdi, hvorved de fremkomne residualer er ukorrelerede for korrekte estimater. For de eksplicitte algoritmer er residualerne ukorrelerede for korrekte estimater, sålænge der er valgt en estimationsalgoritme i overensstemmelse med ordenen af C - polynomiet i modelbeskrivelsen. Ovenstående løsningsforslag bygger bl. a. på, at der ikke er stabilitets eller detektor problemer for de eksplicitte deadbeat algoritmer uanset værdien af tidsforsinkelsen gennem objektet.

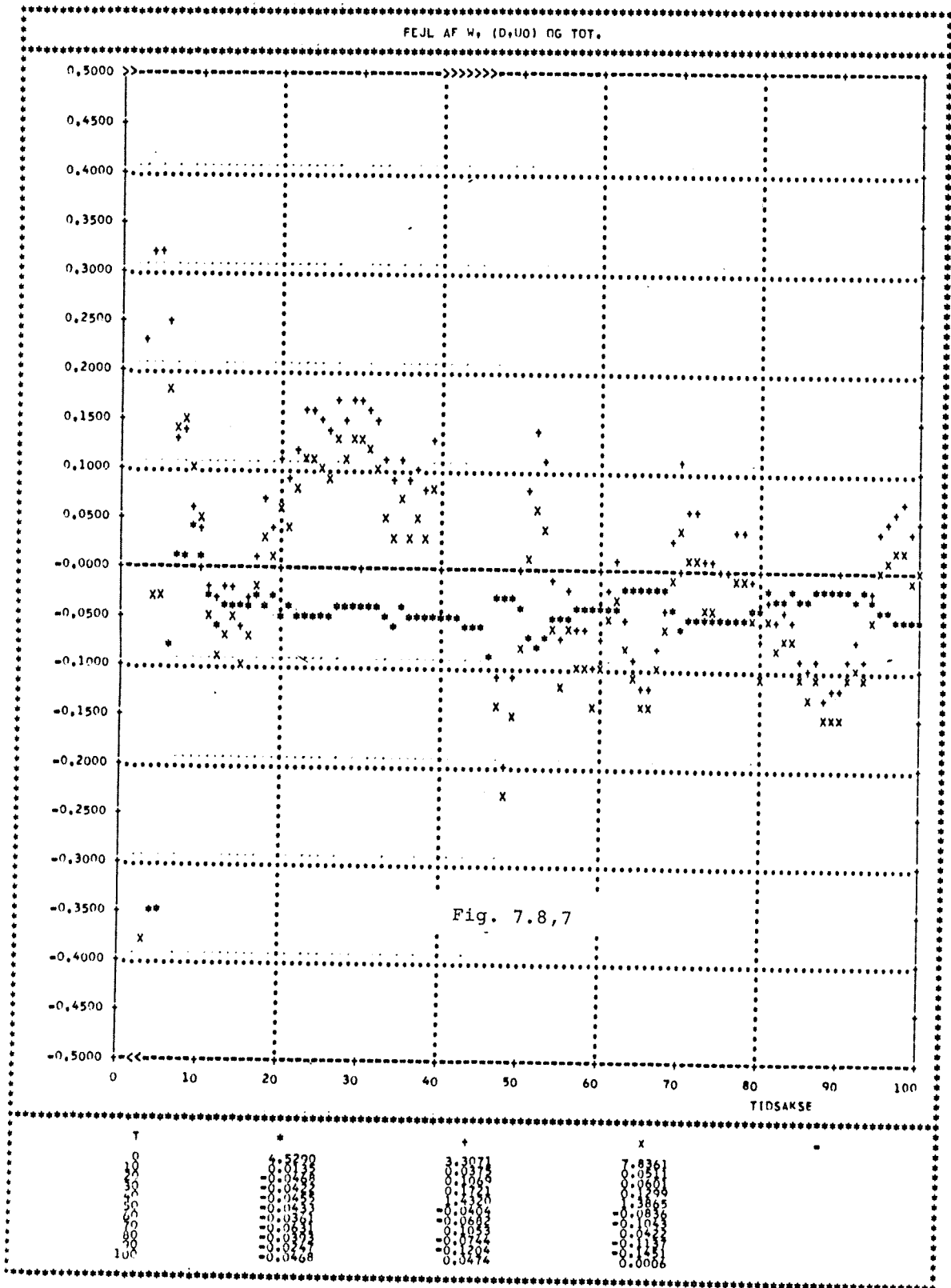
---ooo0ooo---

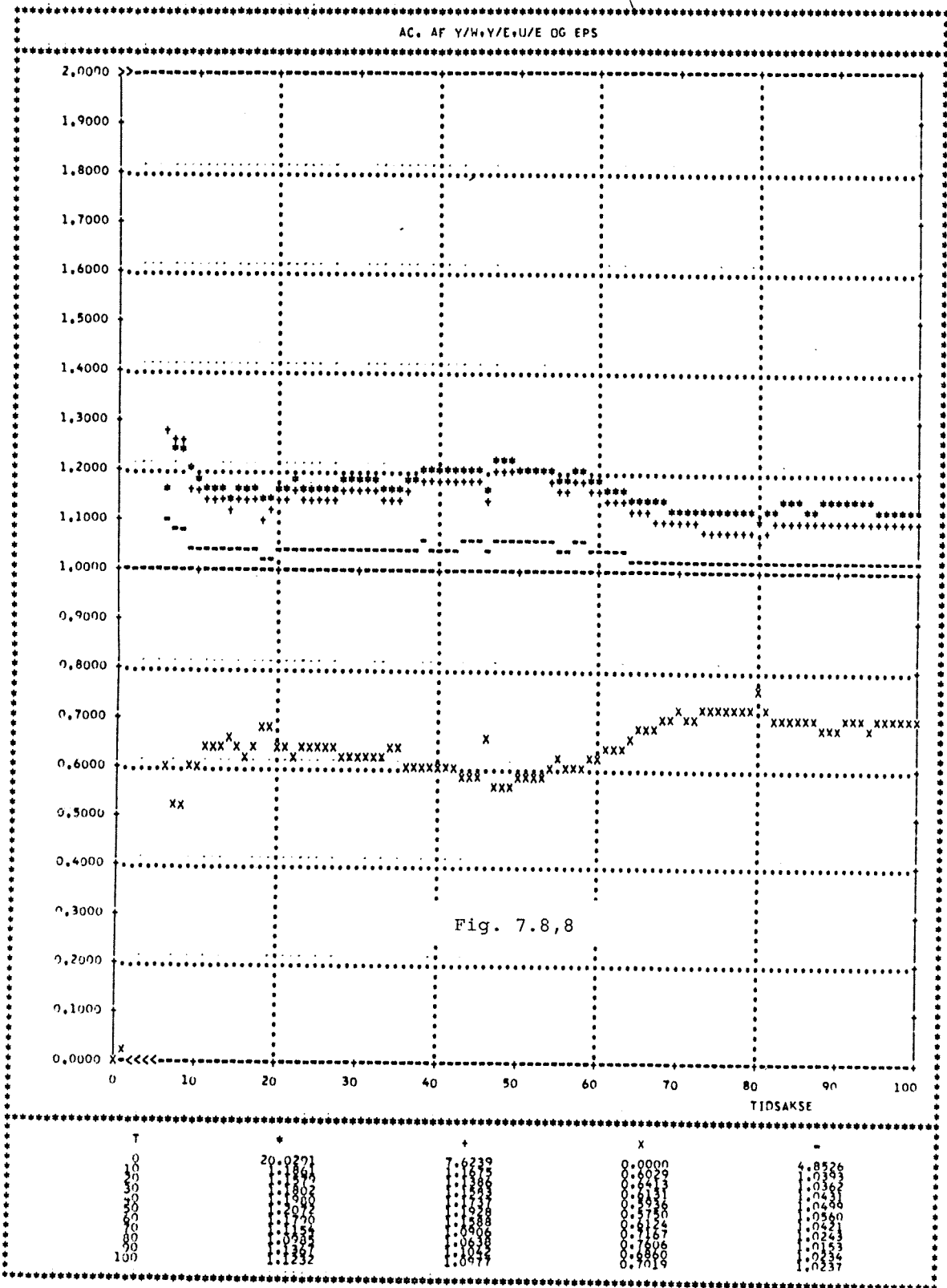












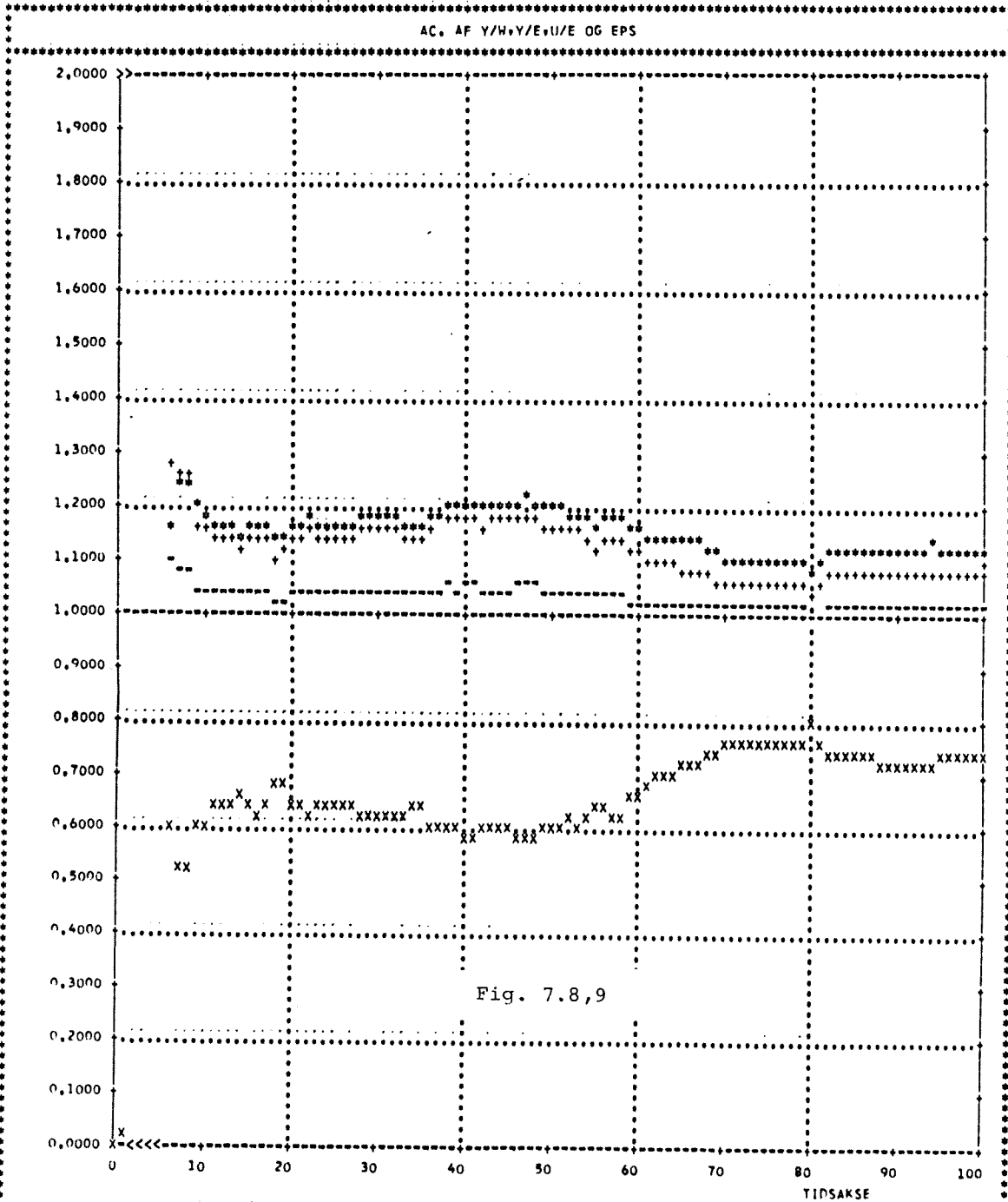
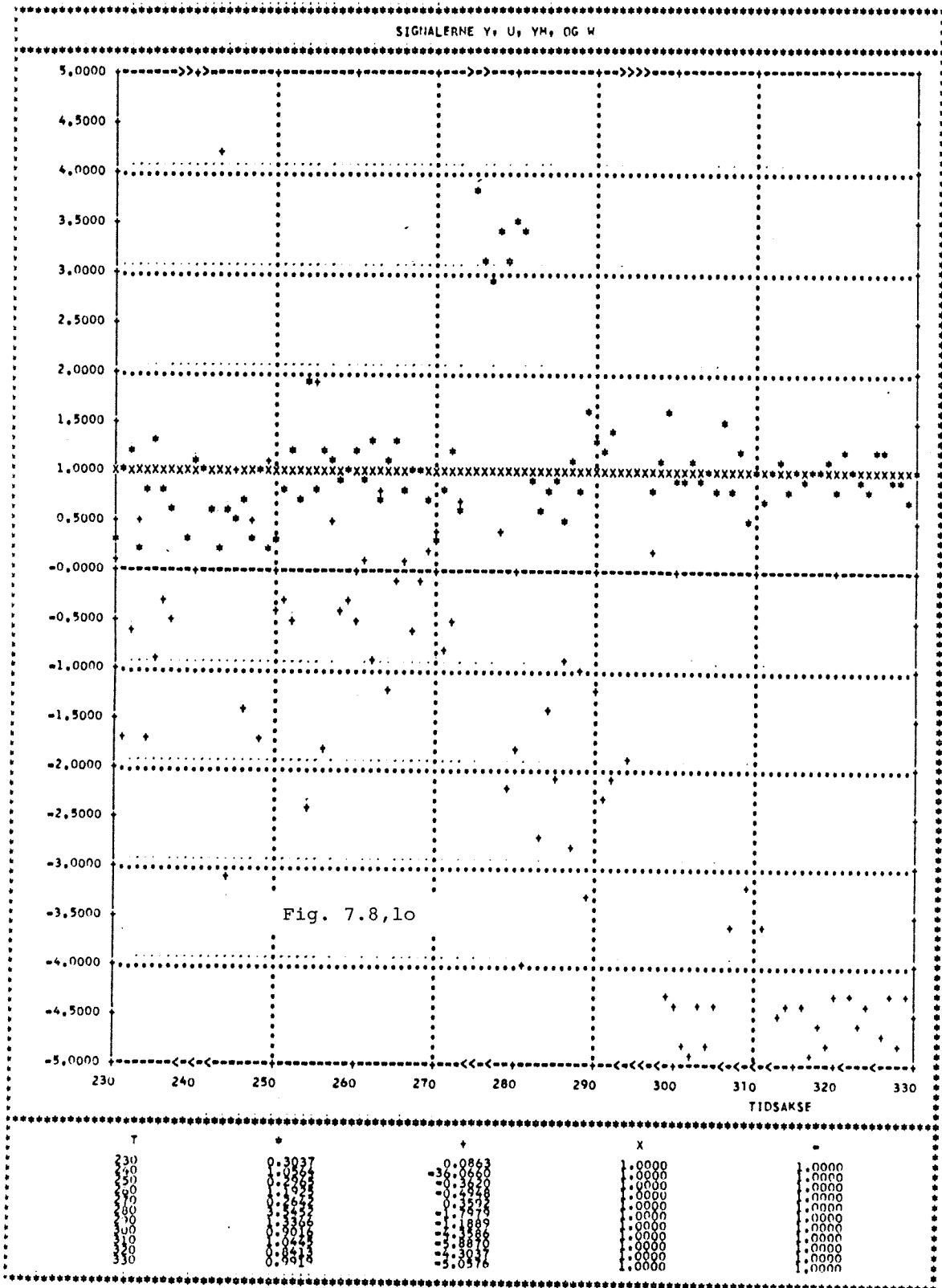
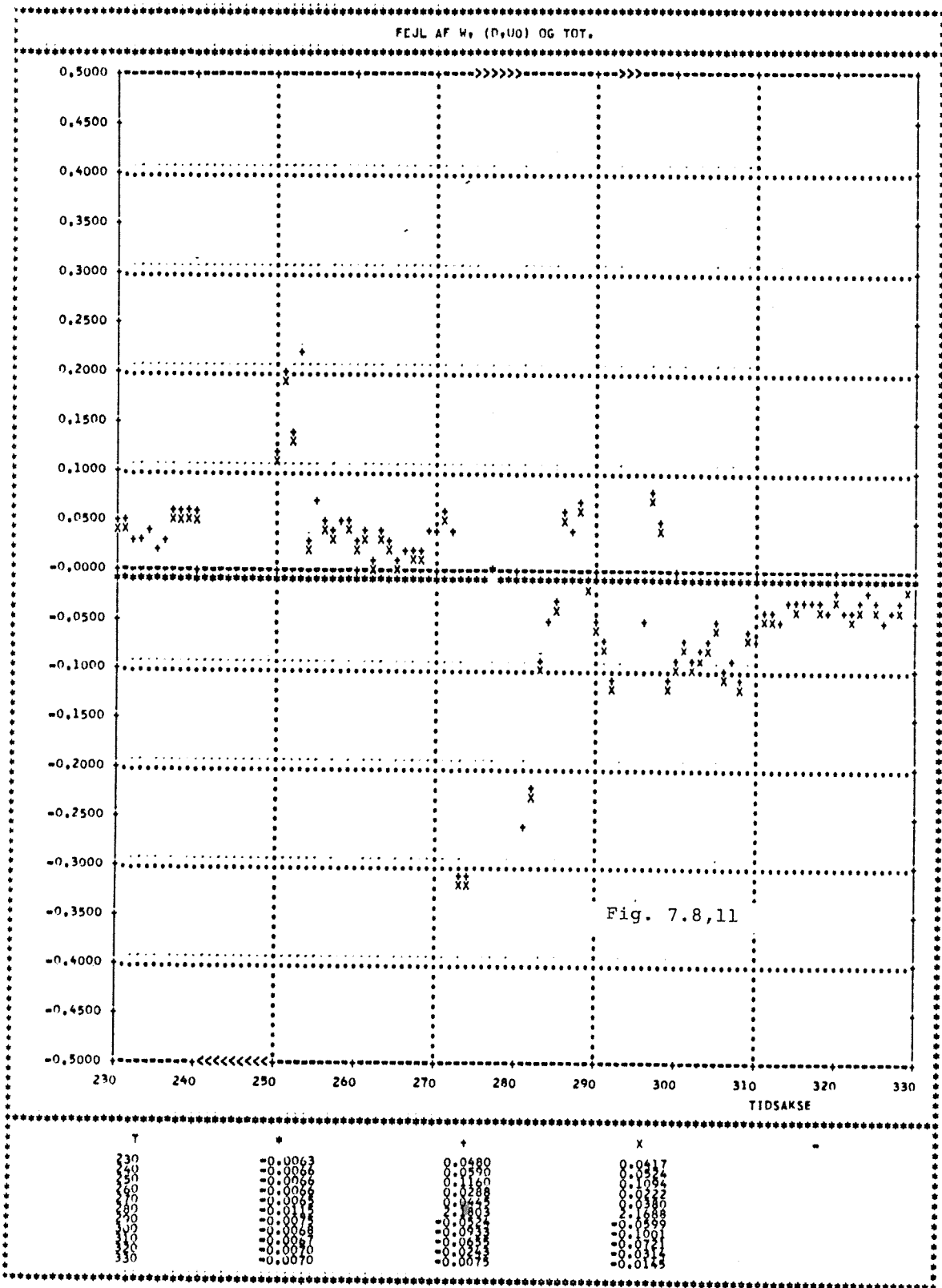
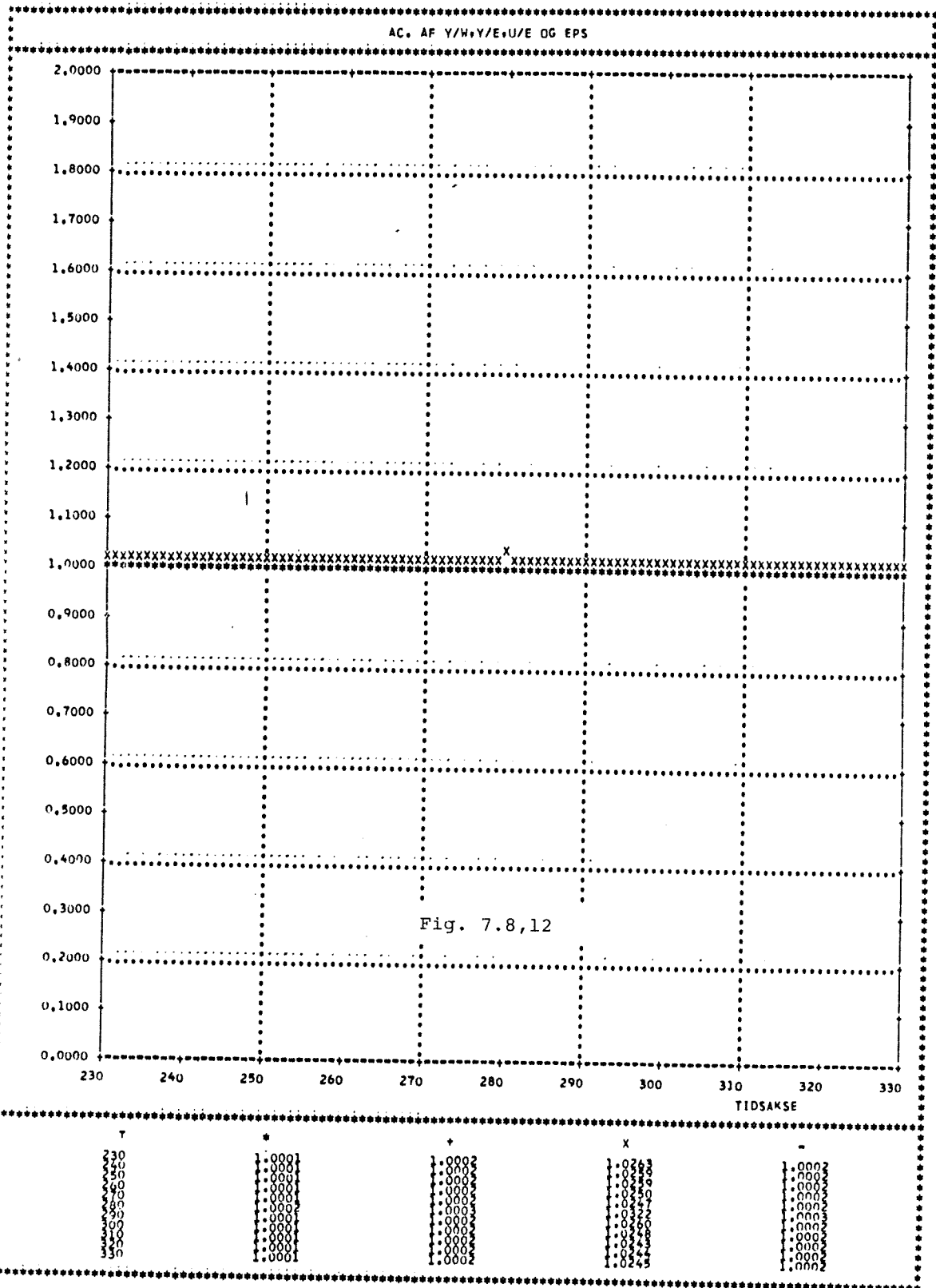


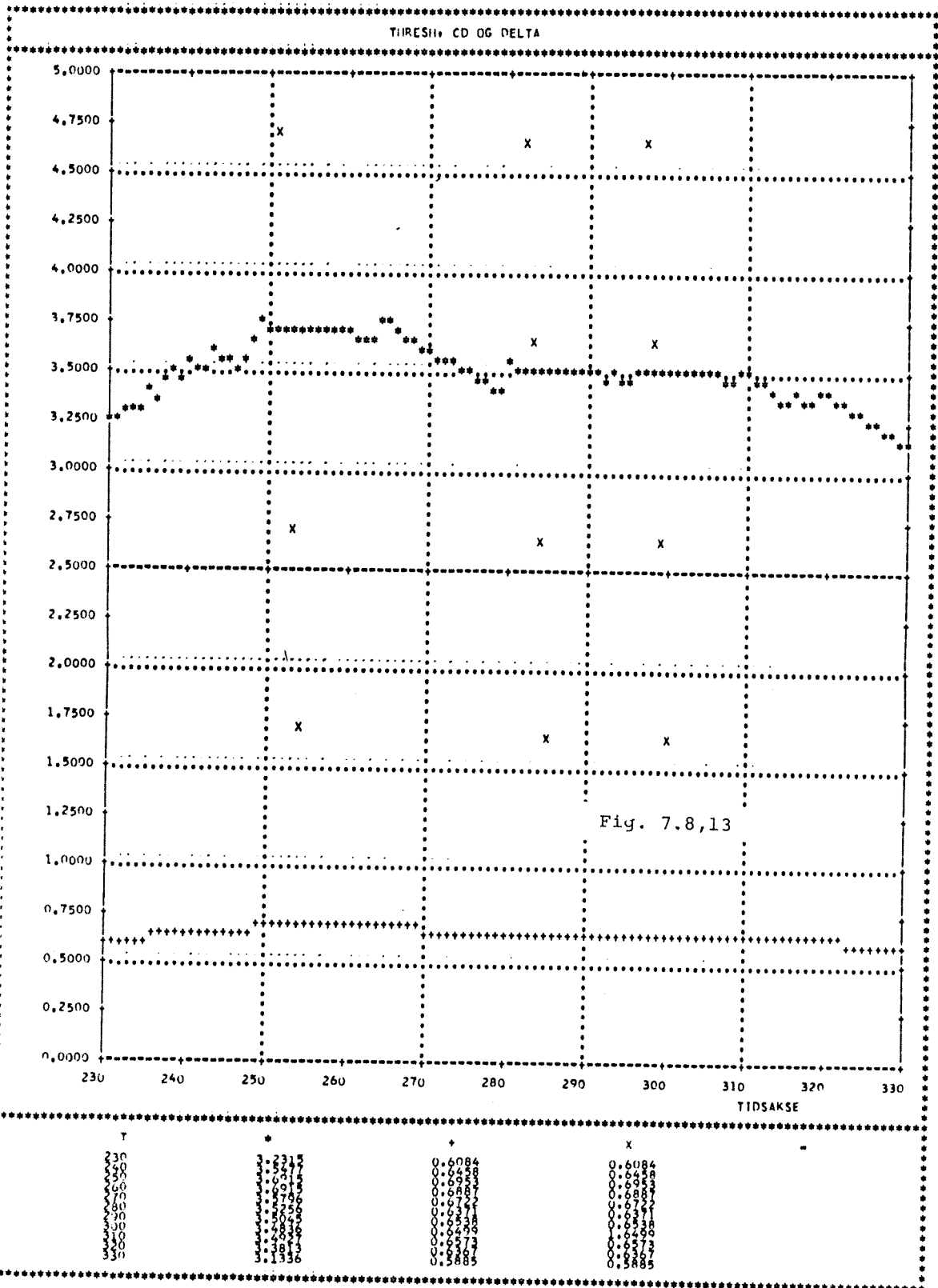
Fig. 7.8,9

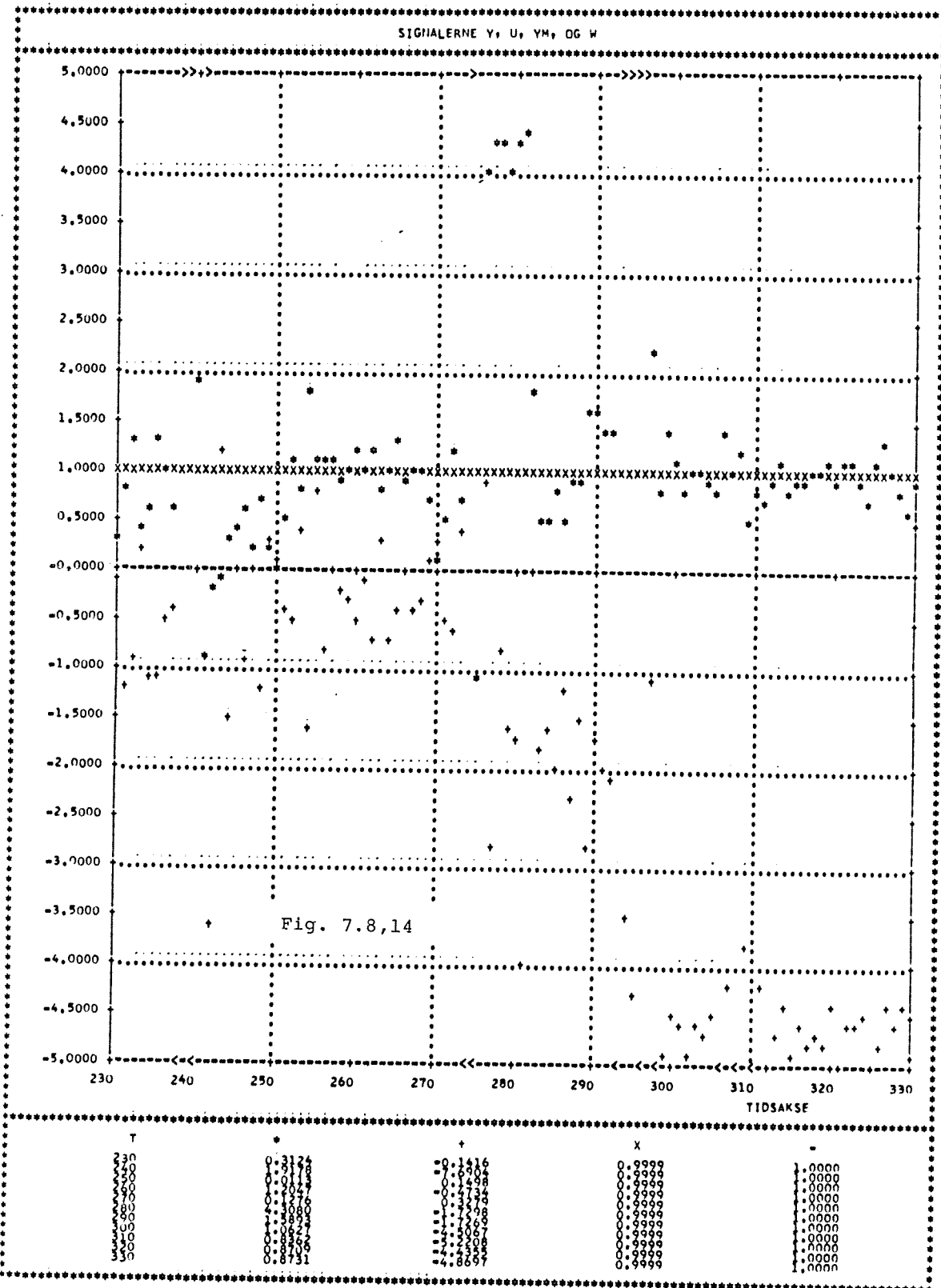
T	*	+	x	-
0.0000	20.0291	7.6233	0.0000	4.8526
1.0000	1.1294	1.1573	0.6029	1.0393
2.0000	1.1802	1.4898	0.5413	1.0302
3.0000	1.2002	1.7806	0.5113	1.0431
4.0000	1.2000	1.6733	0.5876	1.0462
5.0000	1.1800	1.6098	0.5976	1.0462
6.0000	1.0800	1.0600	0.6524	1.0274
7.0000	1.0800	1.0410	0.6524	1.0130
8.0000	1.0800	1.0830	0.6524	1.0143
9.0000	1.0800	1.0830	0.7181	1.0143
10.0000	1.1130	1.0761	0.7347	1.0150

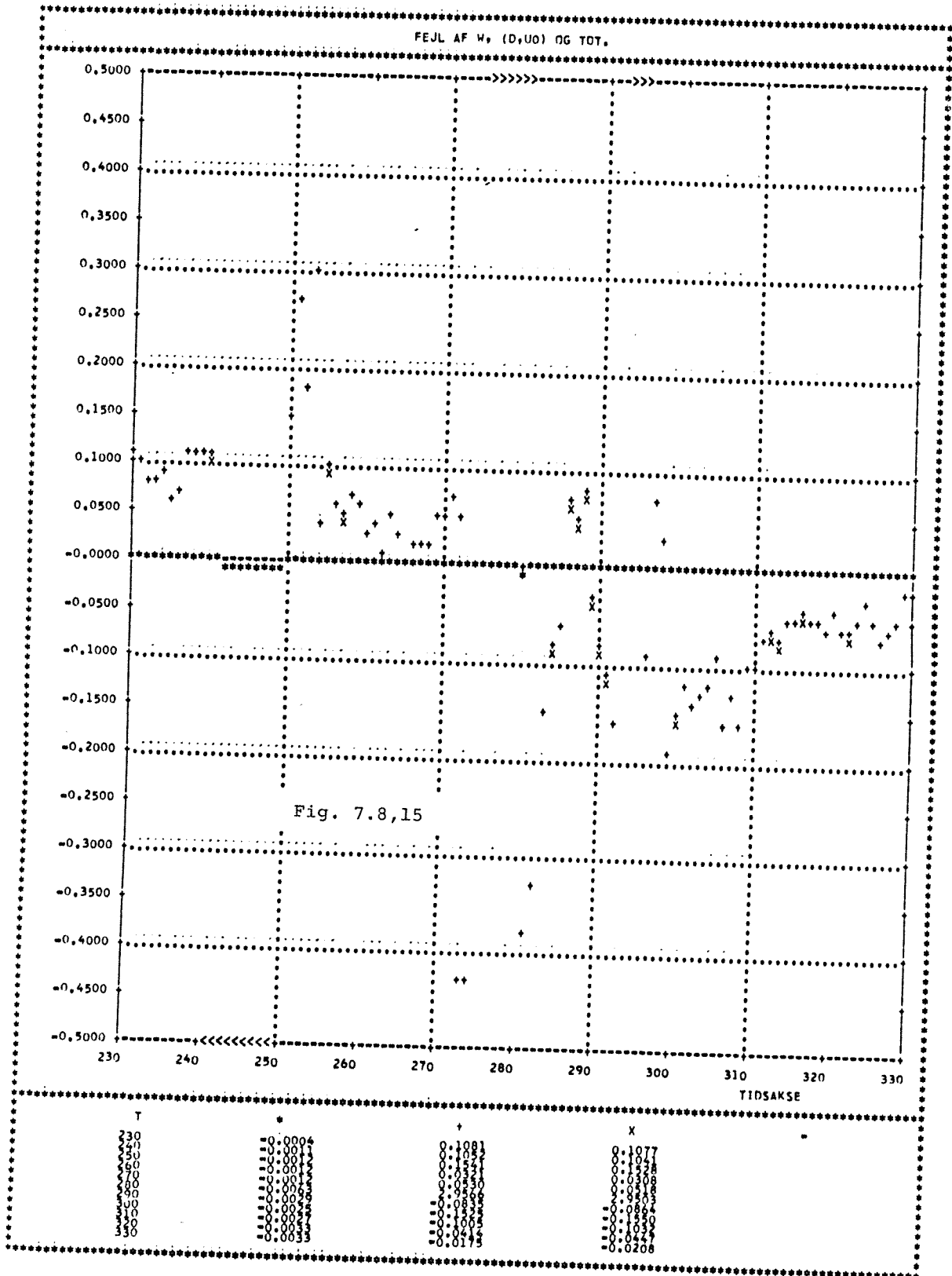


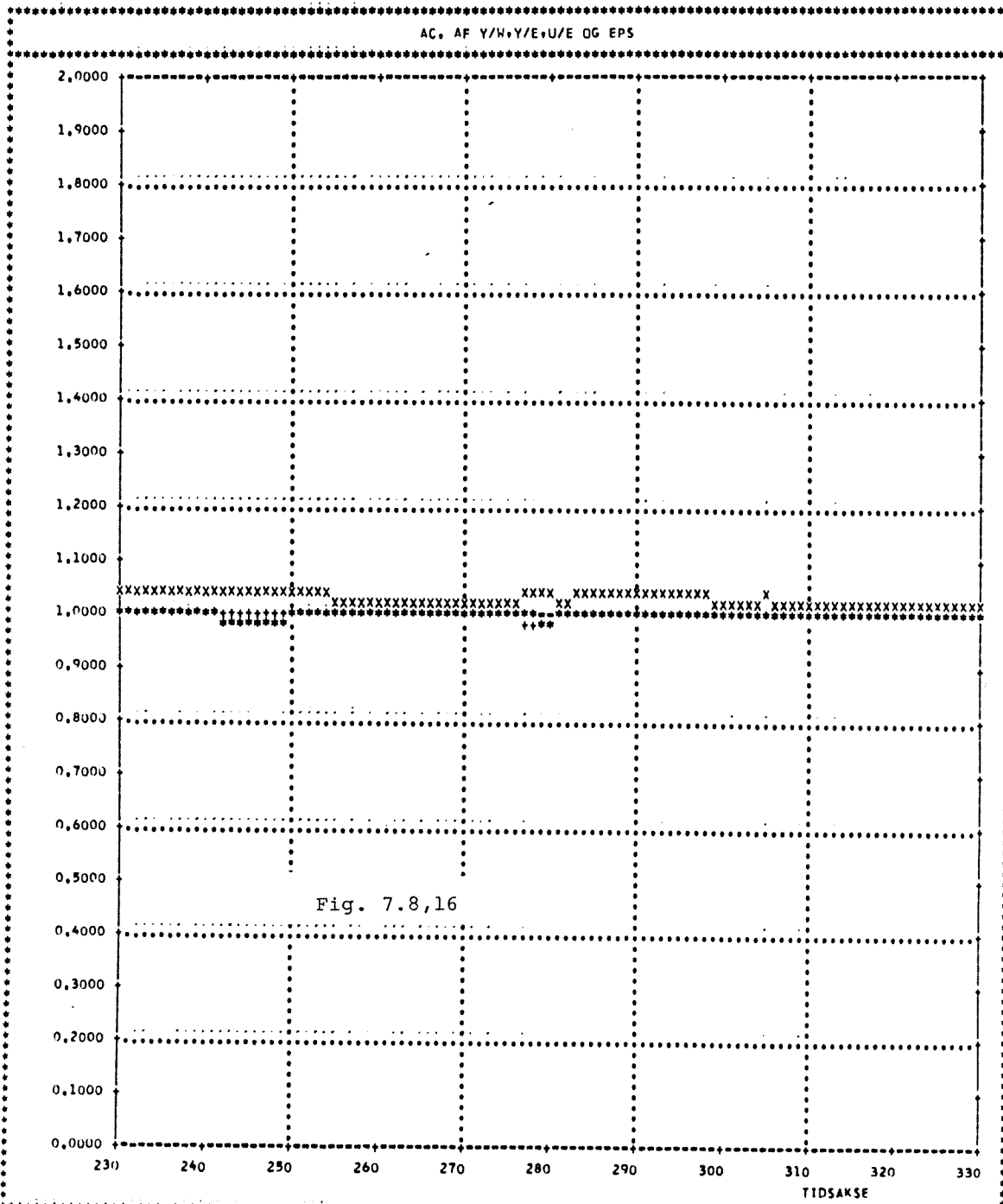




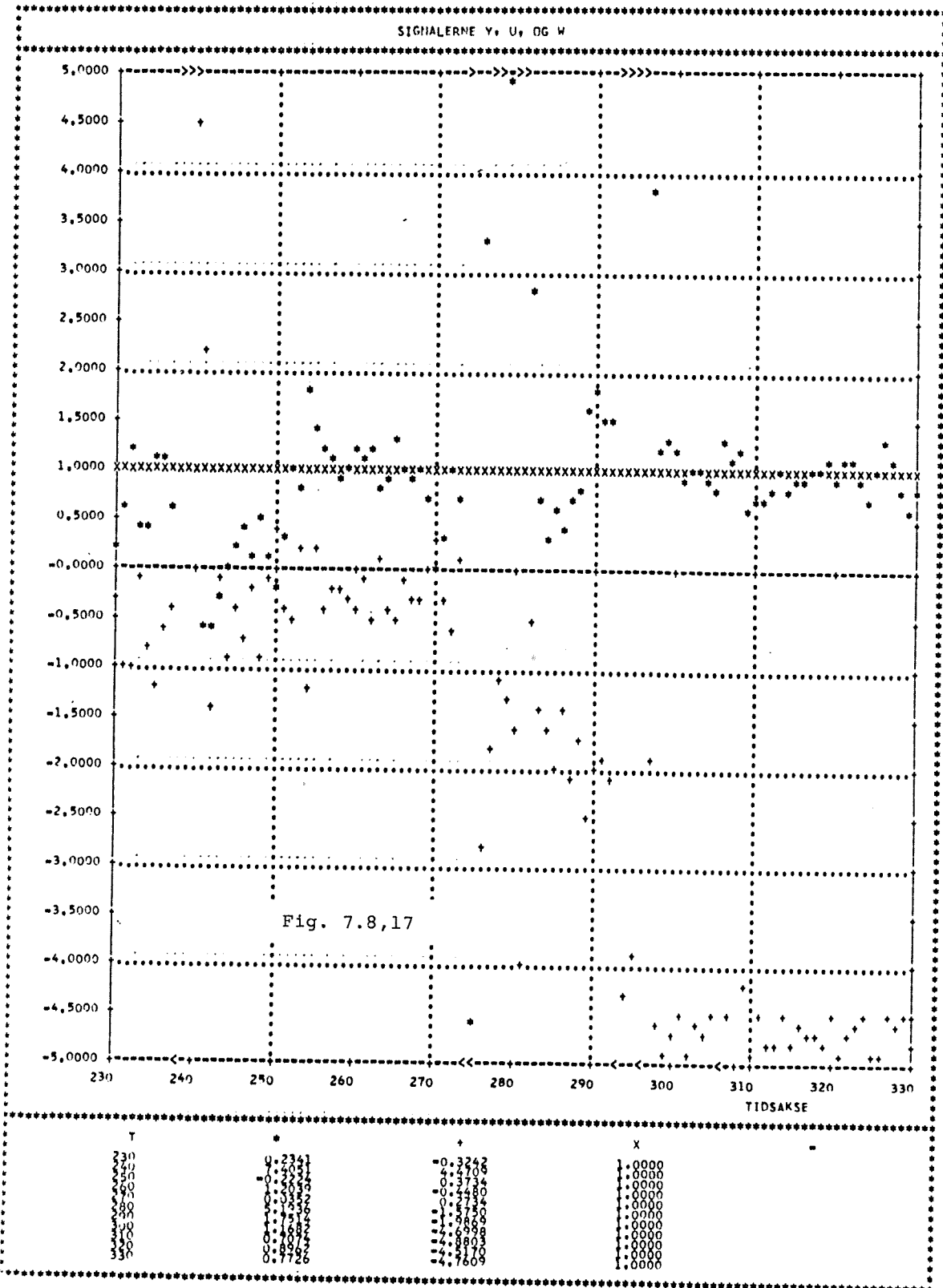


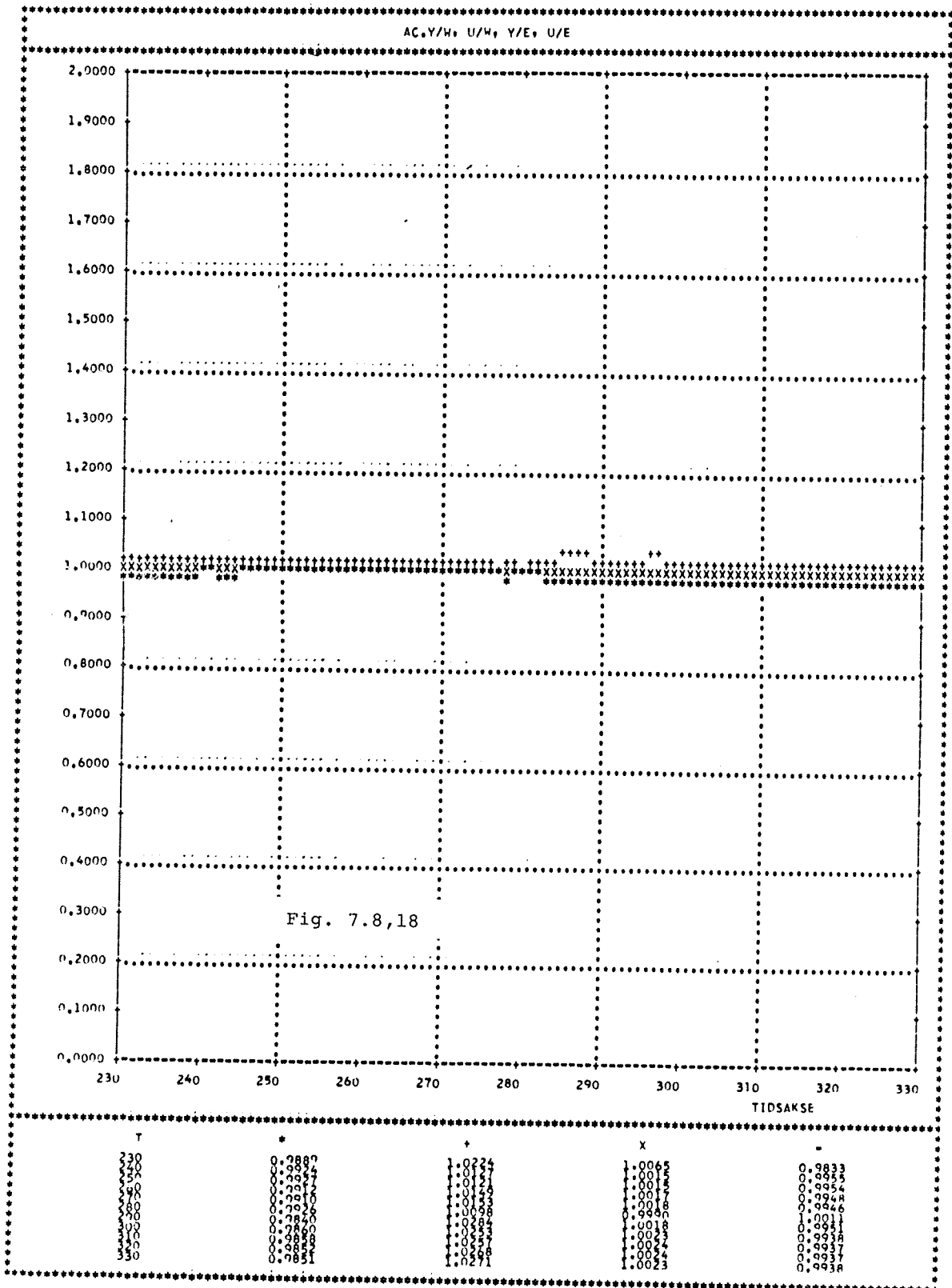


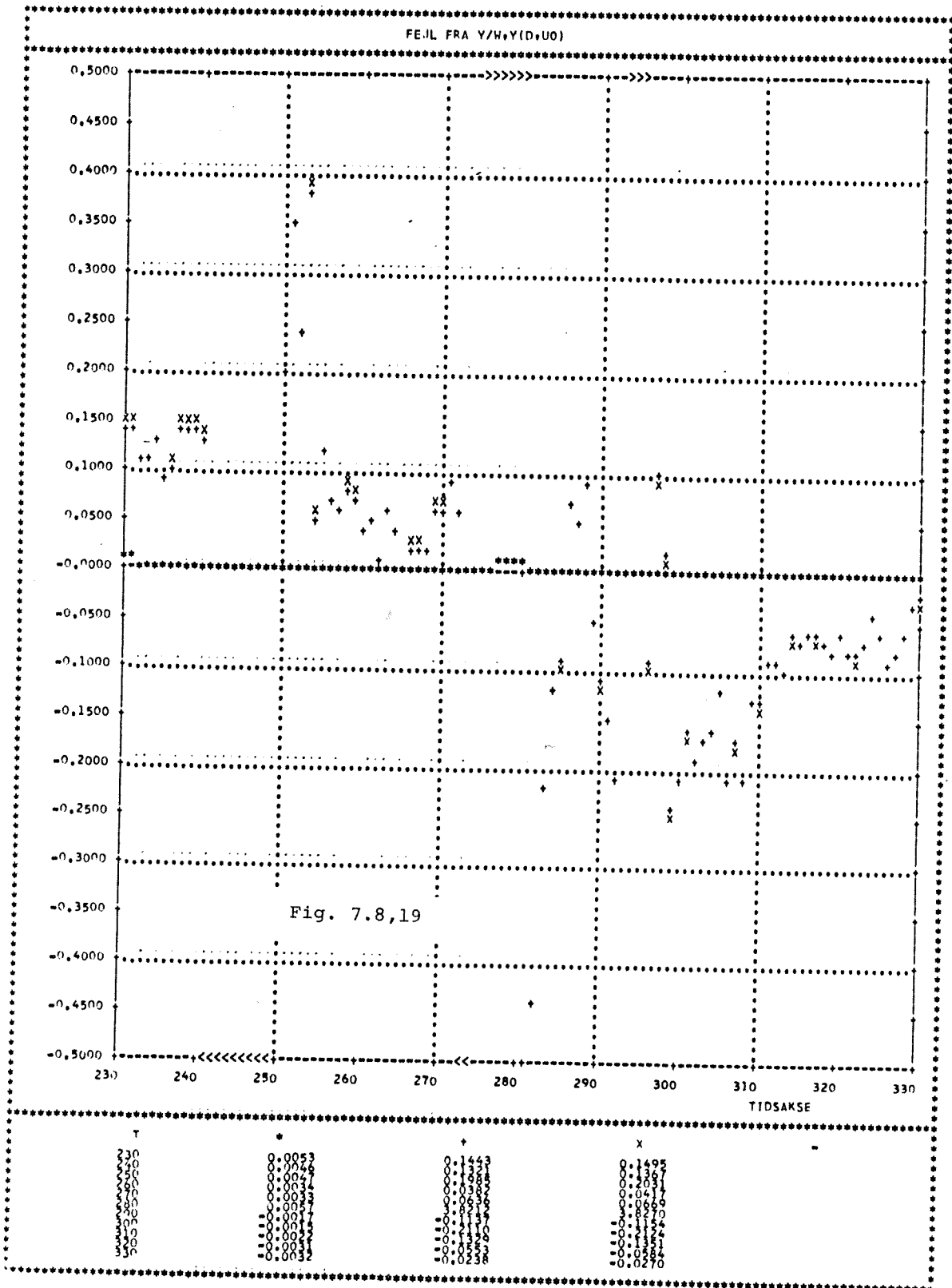


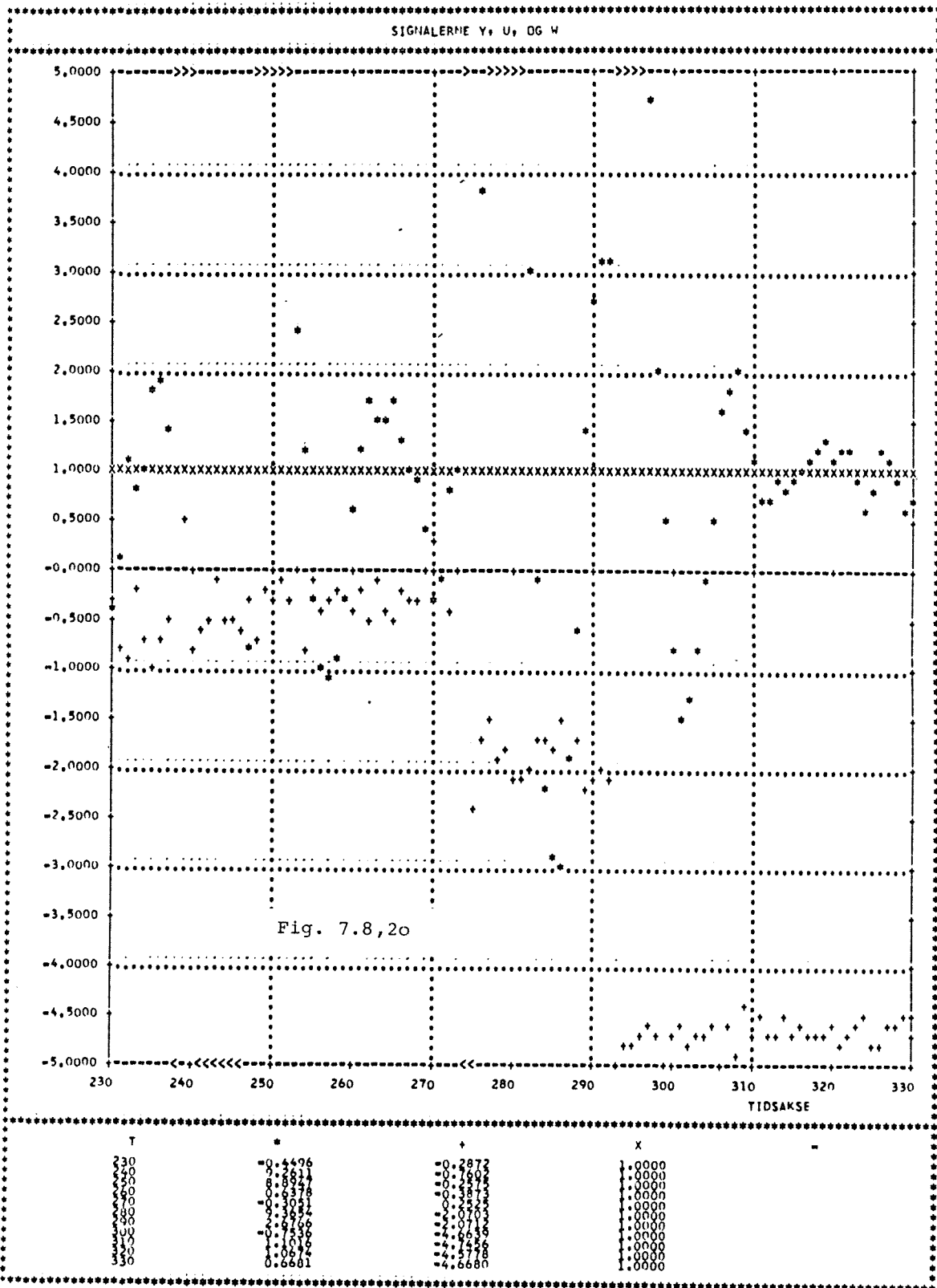


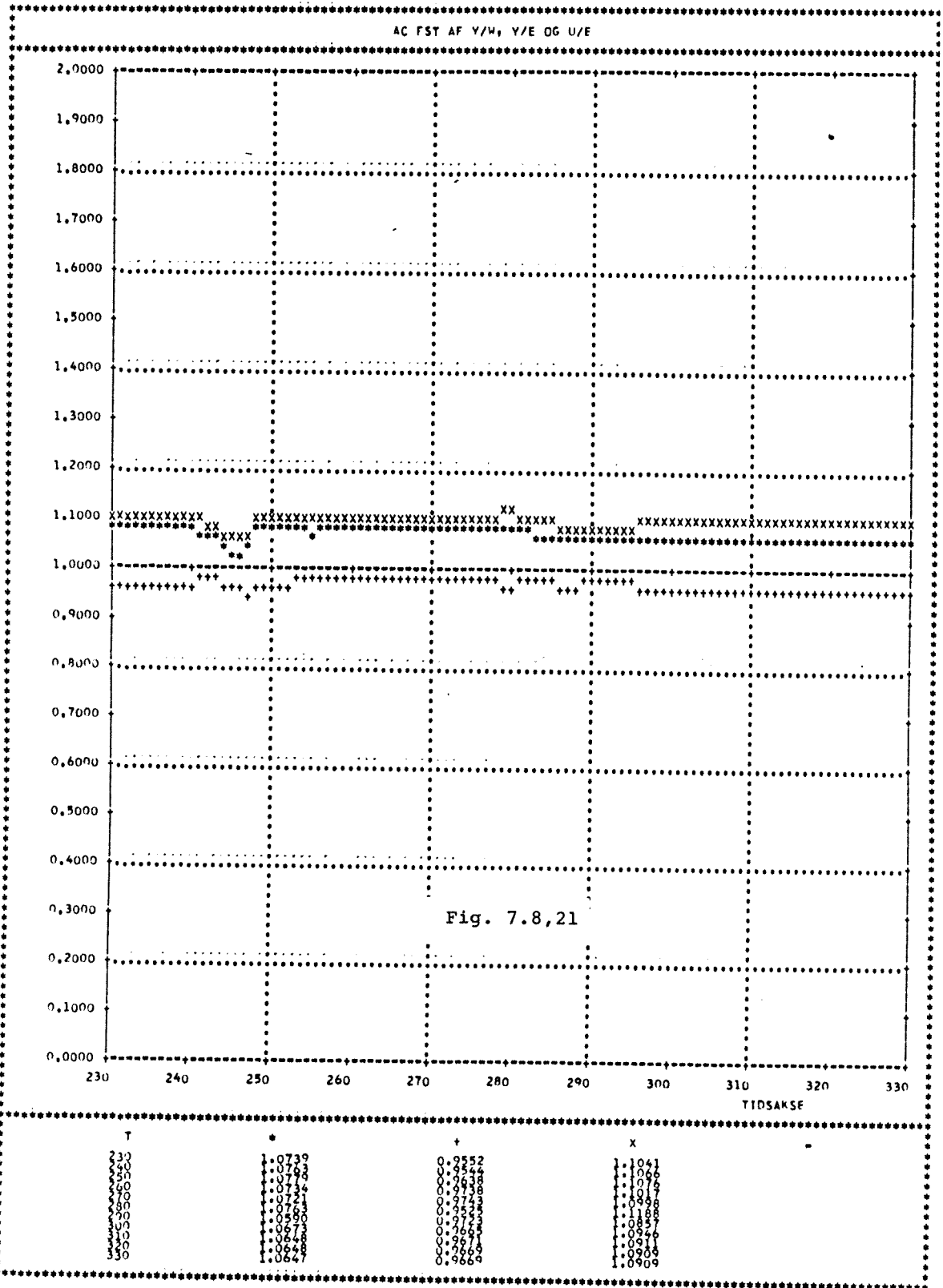
T	*	†	X	-
230	0.9920	0.9907	1.0315	1.0002
240	0.9920	0.9907	1.0308	1.0002
250	0.9919	0.9907	1.0308	1.0002
260	0.9919	0.9910	1.0297	1.0002
270	0.9918	0.9899	1.0296	1.0002
280	0.9918	0.9905	1.0317	1.0002
290	0.9914	0.9914	1.0291	1.0002
300	0.9918	0.9912	1.0287	1.0002
310	0.9918	0.9912	1.0290	1.0002
320	0.9915	0.9914	1.0291	1.0002
330	0.9915	0.9914	1.0291	1.0002

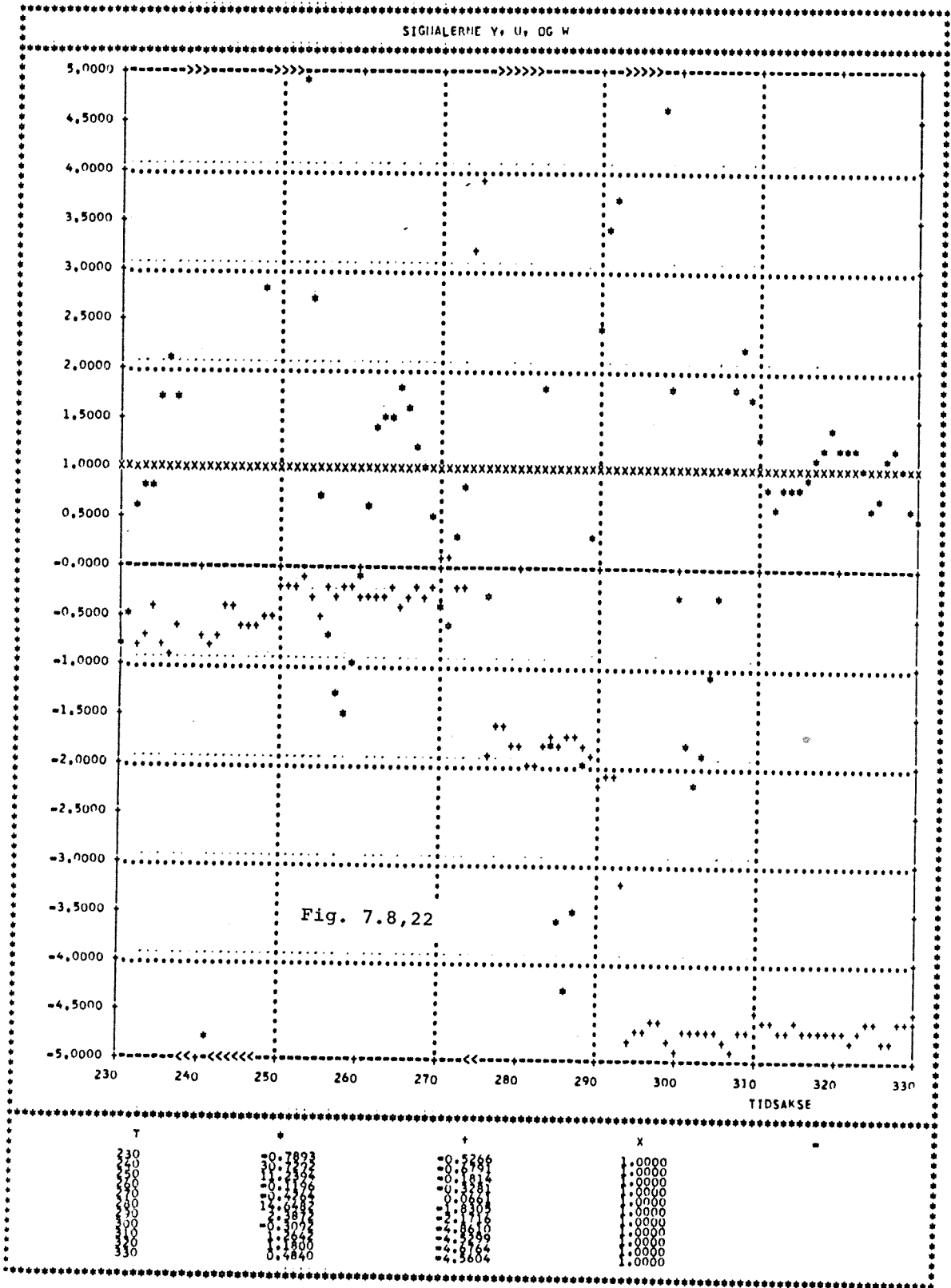




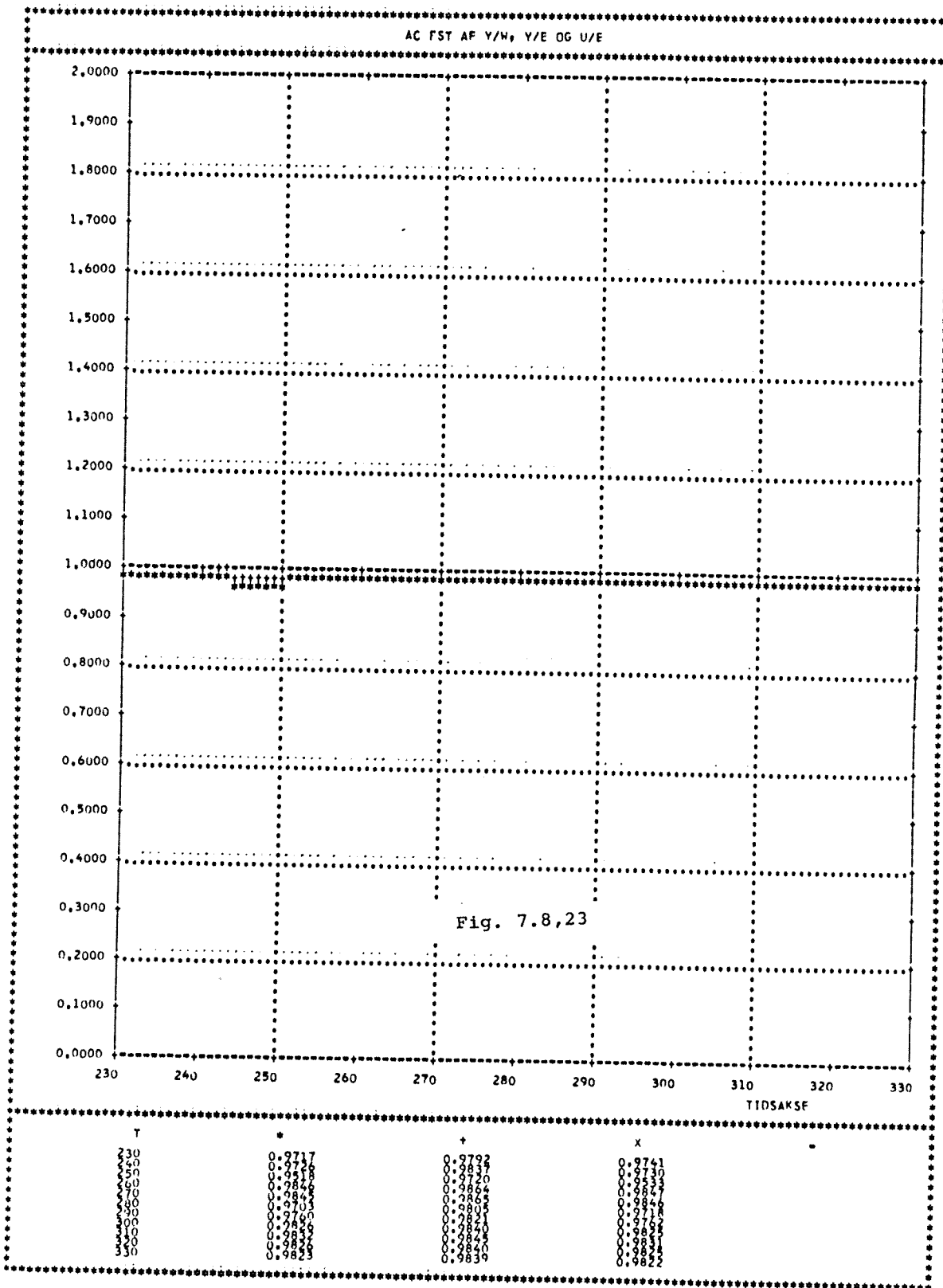


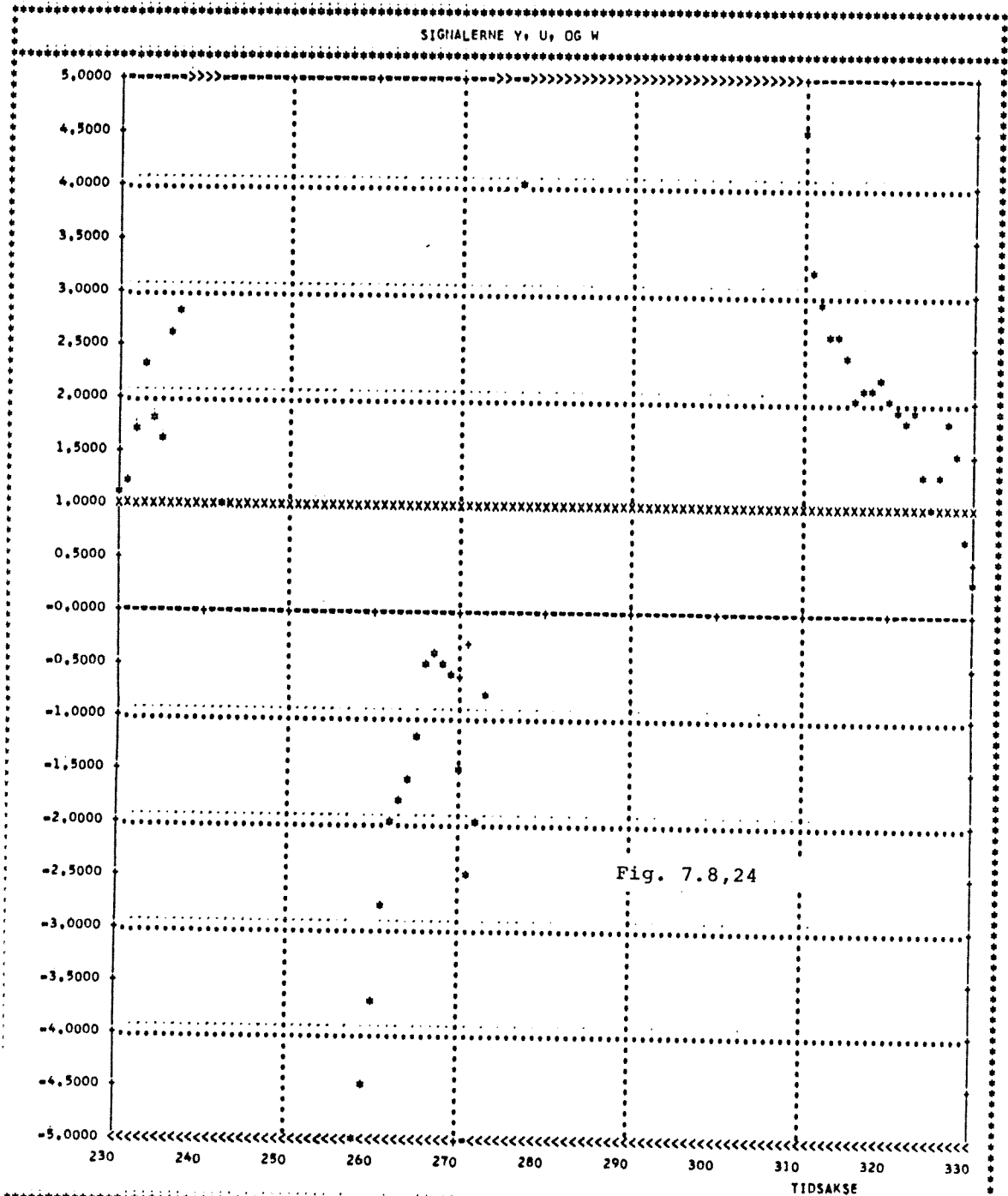




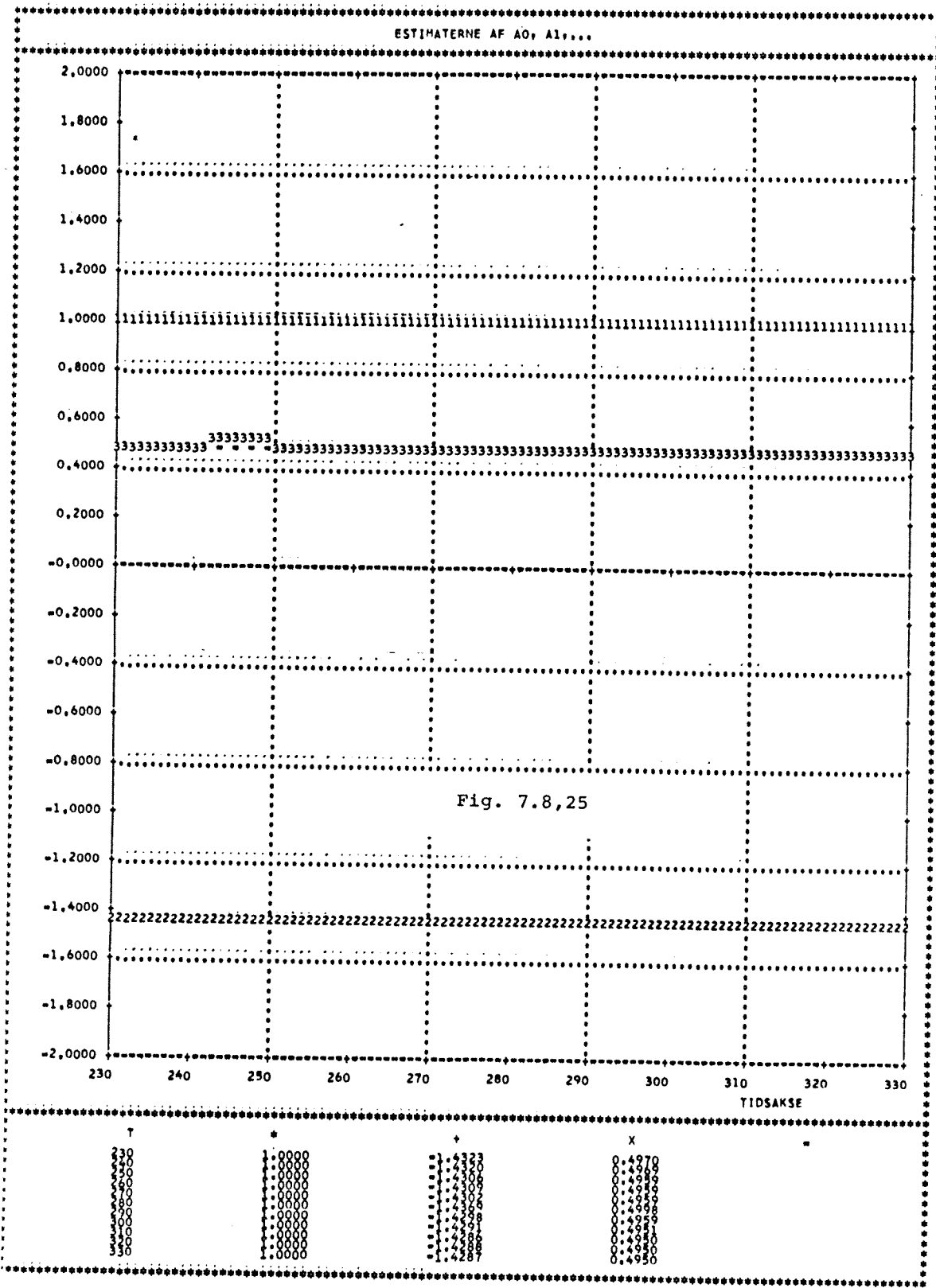


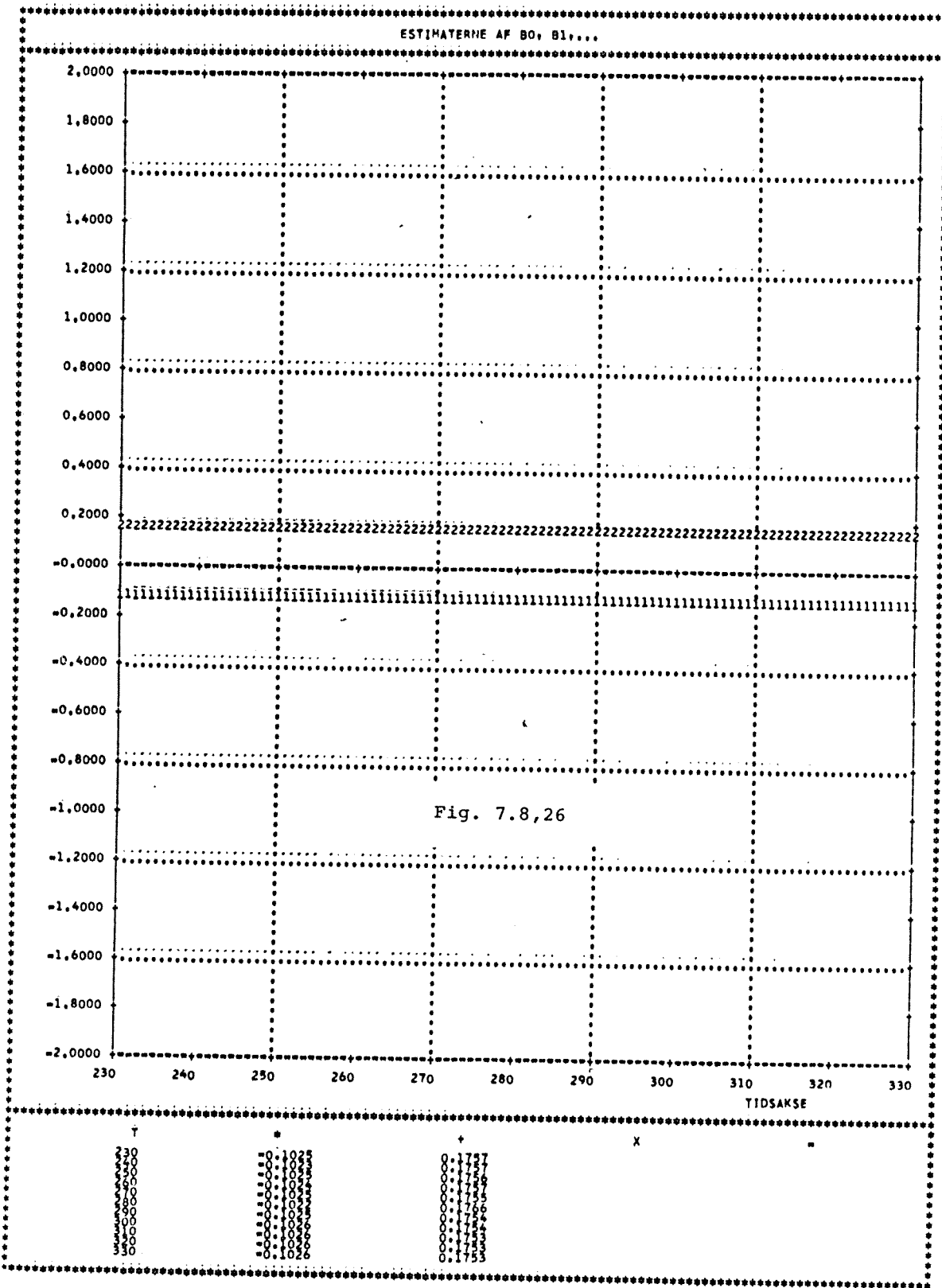
T	*	+	X
230			
240			
250			
260			
270			
280			
290			
300			
310			
320			
330			





T	*	+	X
230	1,0728	=12,8827	1,0000
240	=13,4295	=13,3387	1,0000
250	=1,0719	=7,1165	1,0000
260	=1,4899	=0,1306	1,0000
270	24,0380	=38,9803	1,0000
280	20,0634	=146,7371	1,0000
290	4,0129	=109,3271	1,0000
300	2,0147	=100,8043	1,0000
310	0,2775	=98,2373	1,0000





KAPITEL 8

8. Konklusion.

8. Konklusion.

Formålet med dette projekt er at undersøge adaptive regulatorer specielt med henblik på ikke medtagne model ufuldstændigheder. I projektet er der fokuseret på forstyrrelsesprocessen.

For deadbeatregulatorerne, der er dimensioneret som et rent deterministisk servoproblem, er stokastiske forstyrrelser en model ufuldstændighed. Dette blev afspejlet i regulatortypens moderate reguleringssegenskaber i forbindelse med de to testobjekter. Den deterministiske version af polplaceringsalgoritmerne har særdeles gode reguleringssegenskaber overfor ukorreleret gausisk støj. Dette skal ses i sammenhæng med, at for uafhængige støjpåvirkninger er den deterministiske og den stokastiske version sammenfaldende. For den stokastisk dimensioneret generelliserede minimalvariansregulator er støjpåvirkningerne naturligvis ikke en modelufuldstændighed.

Når der i forstyrrelserne både forekommer spidser og spring i middelværdien samt, det der i kapitel 2 er betegnet som drift, vil der foreligge en situation, som ingen af de her angivne algoritmer er konstrueret til at klare. Undersøgelserne viser, at i visse tilfælde kan standardalgoritmerne godt anvendes i forbindelse med ikke stationær støj, når blot afvigelserne fra stationære forhold ikke er stor. Evnen, til at udkompensere disse påvirkninger, er for standardalgoritmerne tidsvarierende og aftagende.

I projektet blev der derefter forsøgt med to ofte anvendte metoder til at tilpasse identifikationen til situationer med tidsvarierende objektparameter. Ved brug af eksponentiel glemsel kan enkelte fejl udkompenseres i løbet af et tidsrum, der svarer til tidshorizonten for estimationen. I dette tilfælde er egen-skaberne ikke mere tidsafhængige, men det er kun forholdsvis

små variationer i belastningen der kan udkompenseres i løbet af rimelig tid, fordi af hensyn til estimationspræcisionen kan tidshorizonten ikke gøres vilkårlig lille.

Ved at forlade opfattelsen af belastningen som en tidsvarierende parameter og i stedet modellere denne som en ikke styrbar tilstand, der drives af en stationær proces, kan estimationen foretages med et udvidet kalmanfilter. Også i dette tilfælde stødtes på dilemmaet, der skyldes de modstridende krav om en rimelig estimationspræcision og en høj sporingshastighed. Overfor svag drift og mindre spring havde denne metode fine egenskaber, hvilket skyldes, at for disse påvirkninger kan moddelen af belastningen bringes til at passe rimeligt. I støjsvage eller direkte deterministiske tilfælde, hvor estimationen foregår på grundlag af eksitationen fra referencesignalet, kan man med denne metode klare selv store spring, hvis blot modellen af belastningen er passende.

Deadbeatregulatorerne adskilte sig yderligere fra de øvrige ved at indeholde en integralvirkning, hvorfor estimatet af belastningen ikke indgår i reguleringen. Af hensyn til de øvrige estimater er det imidlertid nødvendigt at estimere dem, når standard algoritmen bruges. For at undgå skadelig indvirkning fra et fejlestimat af belastningen, når den springer, kan der til estimationen bruges en model hvori belastningsestimatet ikke indgår. Ud fra en lignende filosofi kan der indføres en integrator i en selvindstillende polplaceringsregulator. I det deterministiske tilfælde er disse løsninger gode, men i en stokastisk situation vil de give fejlagtige estimater på grund af en indført korrelation mellem residualerne, som ikke kan udkompenseres. Det er muligt at dæmpe intralvirkningen, men samtidig nedsættes sporingshastigheden.

Det gennemgående dilemma, der karakteriserer de metoder, hvis udgangspunkt er en stationær proces, kan elimineres ved

at separere de to modstridende krav. En metode, der er konstrueret med henblik på abrupte variationer i belastningen, er at kombinere et udvidet kalmanfilter, der er i stand til at opfange mindre variationer i belastningen, med en detektor, der, hvis der er sket en tilstrækkelig stor ændring, udløser en procedure til at spore denne ændring. I simulationerne er der opnået meget fine resultater med denne metode, når der kun er tale om belastningsvariationer. Overfor spidser har metoden ringe egenskaber, specielt hvis de er hyppigt forekommende. Metoden er ikke konstrueret til selvindstillende regulatorer, hvor residualerne fra estimationsdelen er korreleret, hvilket er tilfældet ved visse anvendelser af polplacerings- og minimalvariansalgoritmerne. Der er i projektet foreslået, men ikke afprøvet, en metode til at omgå dette problem.

I projektet er der afprøvet to metoder til at forøge de selvindstillende regulatorers egenskaber overfor støjspidser. Begge de to metoder har i simulationerne udvist særdeles fine egenskaber, når de anvendes i overensstemmelse med deres forudsætninger. Der er observeret problemer, når residualerne fra estimationen er korreleret, hvilket er et brud med forudsætningerne. Måden hvorpå man kan omgå dette problem er det samme som ved problemet fra detektoren. Denne løsning er som nævnt ikke undersøgt tilstrækkeligt endnu.

Den endelige estimationsalgoritme, der kan opfattes som et udvidet kalmanfilter suppleret med en detektor og nogle robusthedsfaciliteter, er til sidst kombineret med de tre generelle regulatorer og afprøvet. Resultaterne opnået i disse simulationer er ret lovende for metodens anvendelser.

APPENDIX

1. Prediktor.
3. Valg af model.
4. Valg af observerpolynomie.
5. Overføringsfunktioner.
6. Specifikke overføringsfunktioner for objekt II.
7. Simplificeret stabilitet.
8. Brugervejledning.
9. Programmel.

APPENDIX I.

=====

1. Prediktoren.

I dette appendix gennemgås en kort udledning af en mindstekvadrat-prediktor for et lineært system. Lad dette være givet ved:

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

hvor $e(t)$ er et stationært, hvidt $N(0, \sigma^2)$ fordelt støjsignal. Størrelsen d er en konstant. Lad videre j betegne prediktions horisonten, der indledningsvis er positiv. Da er:

$$(2) \quad y(t+j) = \frac{B}{A} u(t+j-k) + \frac{C}{A} e(t+j) + \frac{1}{A} d$$

Hvis de to polynomier E_j og F_j defineres ved:

$$(3) \quad C = A E_j + q^{-j} F_j$$

hvor:

$$(4) \quad \begin{aligned} E_j(q^{-1}) &= 1 + e_1^j q^{-1} + \dots + e_{j-1}^j q^{1-j}, \quad 0 < j \leq k \\ F_j(q^{-1}) &= f_0 + f_1^j q^{-1} + \dots + f_{n_F}^j q^{-n_F}, \quad 0 < j \leq k \end{aligned}$$

kan støjbidraget spaltes i forhold til t :

$$(5) \quad \begin{aligned} \frac{C}{A} e(t+j) &= E_j e(t+j) + \frac{F_j}{A} e(t) \\ &= h_1(t+j, \dots, t+1) + h_2(t, \dots, -\infty) \end{aligned}$$

For at (3) skal kunne løses skal :

$$(6) \quad j + n_F = \max(n_C, n_A + j - 1)$$

hvilket giver, at ordenen på F_j skal være:

$$(7) \quad n_F = \max(n_A - 1, n_C - j)$$

når ordenen af E_j er fastsat til:

$$n_E = j - 1$$

Af udtrykkene (2) og (5) ses, at mindste-kvadrat-predik-
tionen er givet ved:

$$(8) \quad y^*(t+j|t) = E\{ y(t+j) \mid t \}$$

$$= \frac{B}{A} u(t+j-k) + \frac{F_j}{A} e(t) + \frac{1}{A} d$$

med fejlen:

$$\underline{\varepsilon}(t+j|t) = y(t+j) - y^*(t+j|t)$$

$$(9) \quad = E_j e(t+j)$$

$$= e(t+j) + \dots + e_{j-1} e(t+1)$$

Elimineres $e(t)$ i (9) og indsættes i (8) giver dette sammen med
prediktorligningen, (3), at:

$$(10) \quad C y^*(t+j|t) = q^{j-k} B E_j u + F_j y + E d$$

For $j \leq 0$ gælder simpelt, at:

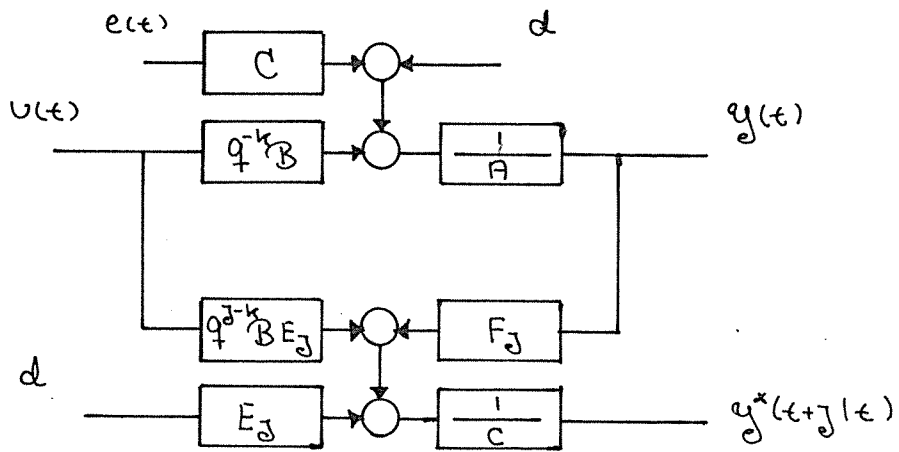
$$(11) \quad y^*(t+j|t) = y(t+j)$$

Dette er indeholdt i (10), hvis E_j og F_j for $j \leq 0$ defineres

som:

$$(12) \quad \begin{aligned} E_j &= 0 & j &\leq 0 \\ F_j &= q^j C & j &\leq 0 \end{aligned}$$

Når $j = k$ undlades indexet for E og F. Prediktionen kan også angives ved et blokdiagram.



APPENDIX III

=====

3. Modelvalg.

I dette appendix er der søgt illustreret, hvorledes modellens poler påvirker reguleringens egenskaber for en polplacerings regulator, der plomberer alle objektets nulpunkter. Da polerne i modellen påvirker overføringsfunktionen $H_{u,e}$ på en ret kompliceret måde, vil undersøgelserne blive foretaget på et objekt(objekt I) givet ved:

$$A = 1 - 1.5q^{-1} + 0.7q^{-2} \quad k = 1$$

$$B = 1 + 0.5q^{-1} \quad C = 1$$

Regulatoren dimensioneres ud fra modellen:

$$A^m = 1 - 2\alpha q^{-1} + (\alpha + \beta) q^{-2}$$

$$B^m = (1 - \alpha)^2 + \beta^2$$

Denne model er valgt således, at den har to kompleks konjugeret poler, ingen nulpunkter samt en DC - forstærkning på 1. Overføringsfunktionerne $H_{y,w}$, $H_{u,w}$, $H_{y,e}$ og $H_{u,e}$ kan udregnes ved (5.3.1.1,32 - 35). I figur 1 - 4 er angivet et konjunkturplot for varianserne (i logaritmisk skala) af y og u , når referencesignalet, $w(t)$, og støjsignalet, $e(t)$, er hyvide normalfordelte støjsignaler med spredning 1 og middelværdi nul. Konjunkturerne er fremkommet ved at lade polerne i modellen vandre rundt indenfor enhedscirklen. Selve plottet er fremstillet med programmet TAB (appendix 7).

ST. VARIANS FOR Y/W (I DB)

REG. TYPE I POL = TP2
ABCISSE I RE
ORDINAT I IM

REF = 1.0000
SKALERINGSFAKTOR I 1.0000
AKSEFORHOLD I 1.0000

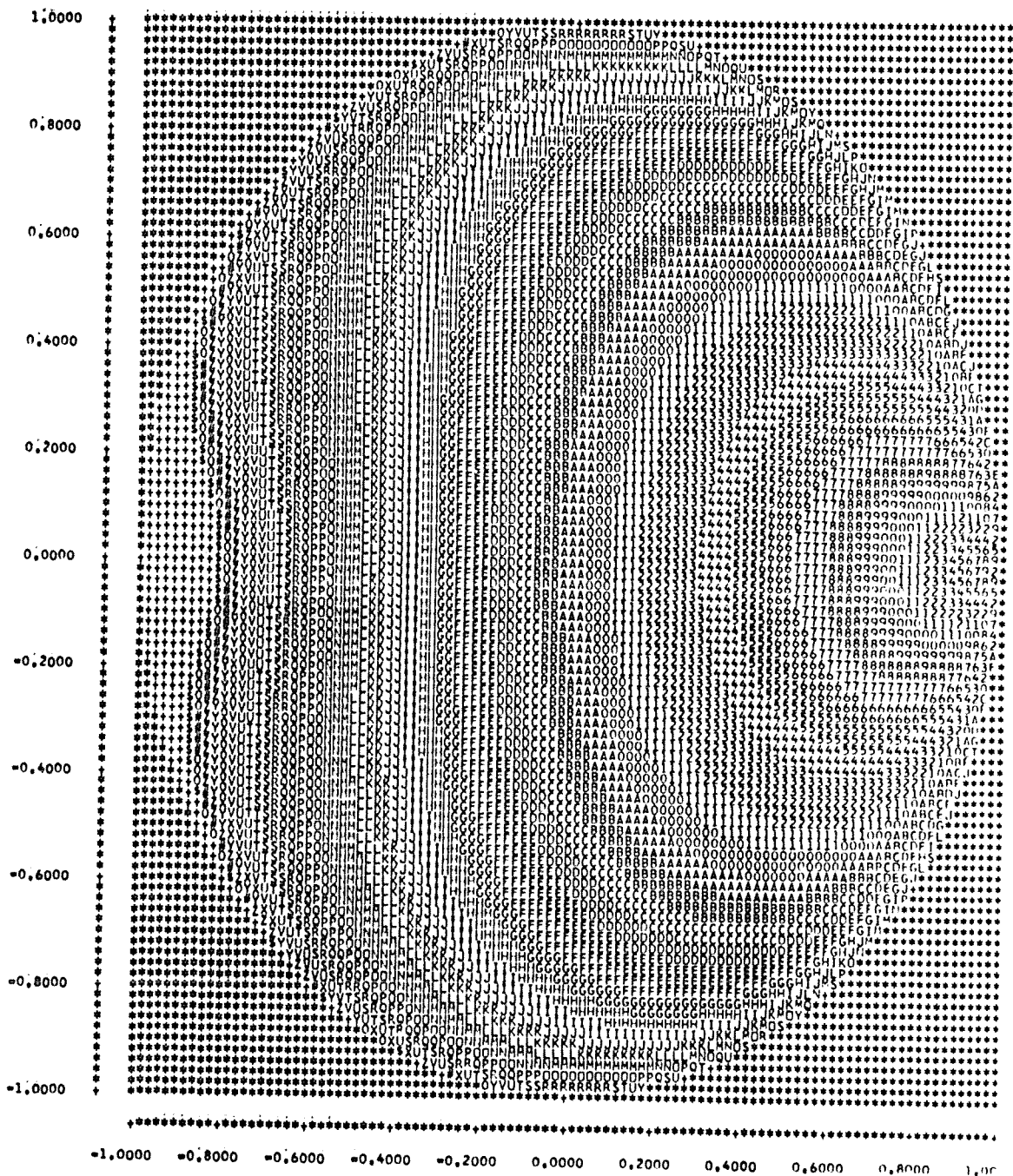


Fig. 1: Konjunkturplot for den stationære varians af y, når w er hyld støj og e = 0, og når α (realdel) og β (imaginærdel) varierer.

ST. VARIANS FOR Y/E (I DB)

REG. TYPE I POL = TP2
ABCISSE I RE
ORDINAT I IM

REF = 1,0000
SKALERTINGSFAKTOR I 1,0000
AKSEFORHOLD I 1,0000

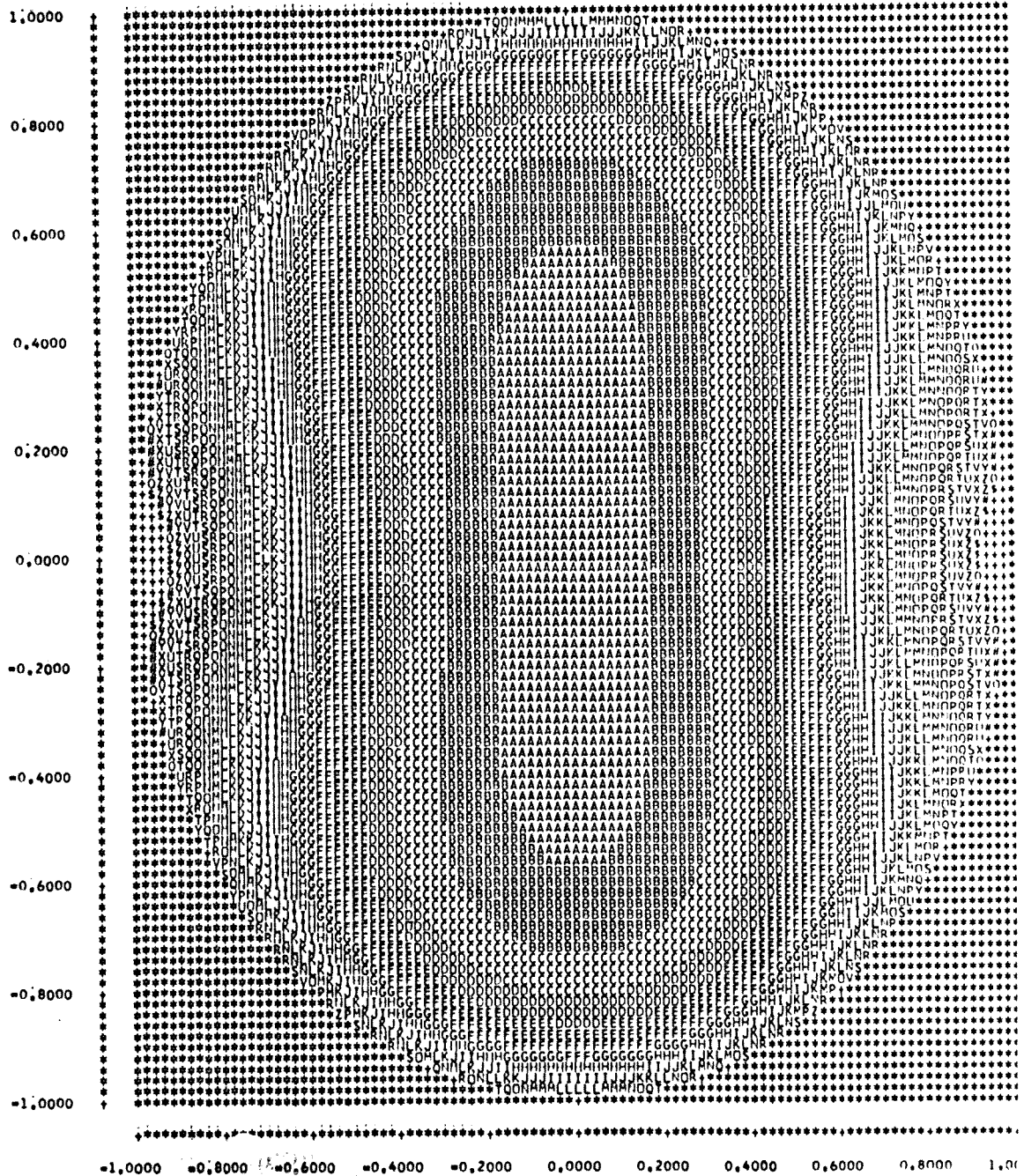


Fig. 3: Konjunkturplot for den stationære varians af y, når $w = 0$ og e er hvid støj.

ST. VARIANS FOR U/E (I DB)

REG. TYPE : POL = TP2
ABSCISSE : RE
ORDINAT : IM

REF = 1,0000
SKALERINGSFAKTOR : 1,0000
AKSEFORHOLD : 1,0000

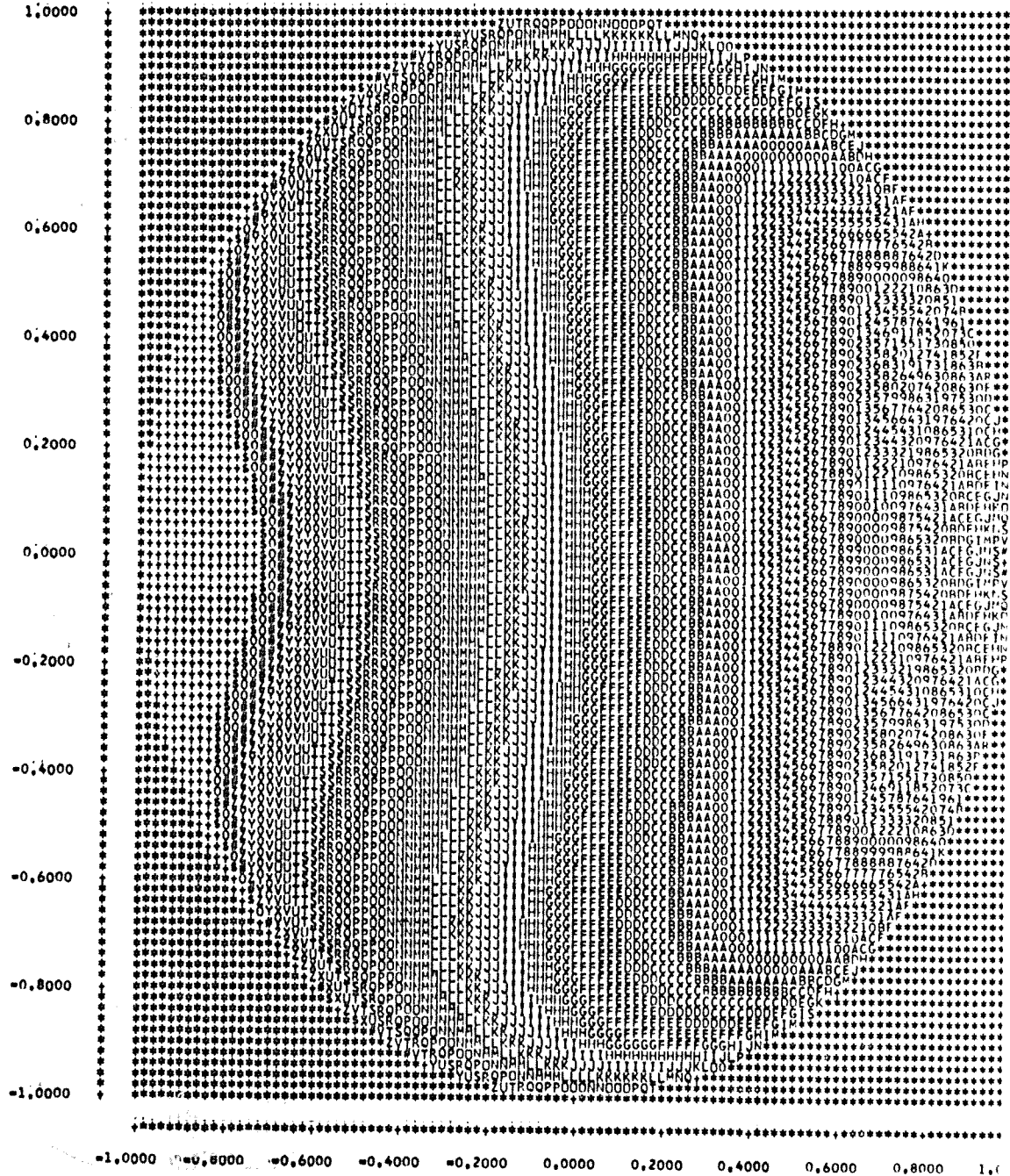


Fig. 4: Konjunkturplot for den stationære varians af u, når w = 0 og e er hvid støj.

Hvis polerne vandrer fra origo (MVo - regulering) og ud i højre halvplan mod 1 vil modellen blive mere lavfrekvent og styringen mere moderat. Af figurene fremgår det, at man kan opnå en ret betydelig reduction af styreeffekten i forhold til MVo - regulering, uden at forringe reguleringskvaliteten væsentligt. Dette skyldes, at tabsfladerne for $H_{y,w}$ og $H_{y,e}$ er relativt flade omkring origo. Når modellens poler er sammen - faldende med objektets, forsvinder tilbagekoblingen helt, hvilket fremgår af fig. 4 samt af, at polynomiet F bestemmes af udtrykket:

$$\begin{aligned} F &= q(A^m - A) \\ &= (a_1^m - a_1) + (a_2^m - a_2) + \dots \end{aligned}$$

når T og k som her er lig 1.

---ooo0ooo---

APPENDIX IV

=====

4. Valg af observerpolynomie.

I dette appendix vil der blive angivet og undersøgt nogle metoder til valg af observerpolynomiet T for en deterministisk dimensioneret polplaceringsregulator. Man kan måske mene, at dette ikke er relevant i forbindelse med adaptive regulatorer, hvor en model af støjen sjældent haves eller kan forudsættes. Denne undersøgelse kan imidlertid give en retningslinie for, hvordan man med valgt model (A^m og B^m) kan vælge T . Det antages, at støjens bidrag til styresignalet er forsvindende sammenlignet med referencens (j.v.f. den deterministiske dimensionering).

Udgangspunktet er et valgt polynomie, A^m , og overføringsfunktionerne fra kapitel 5.3:

$$(5.3.1.1, 32) \quad H_{y,e} = \frac{C}{TA^m} E$$

$$(5.3.1.1, 35) \quad H_{u,e} = - \frac{F}{BTA^m}$$

samt dimensioneringsligningen:

$$(5.3.1.1, 31) \quad TA^m = AE + q^{-k} F$$

Ses der i det følgende bort fra referencen, vil der kunne findes et minimum for variansen på $y(t)$ ved at foretage en numerisk optimering på baggrund af (33) og (31). Minimaet kan også

findes ved en approximativ metode. Såfremt

$$(1) \quad C = TA^m$$

fremkommer den basale minimalvarians regulering (MVo), idet da er:

$$(2) \quad H_{y,e} = E_o$$

$$(3) \quad H_{u,e} = - \frac{F_o}{BC}$$

$$(4) \quad C = AE_o + q^{-k} F_o$$

Ligningen (1) kan desværre kun opfyldes i visse specielle tilfælde. Udskrives (1) i komponenter, fås at:

$$\begin{aligned} & 0 = c_1 - (a_1 + t_1) \\ & \quad \vdots \\ & 0 = c_n - (a_n + a_{n-1}t_1 + \dots + a_1t_{n-1} + t_n) \\ (5) \quad & 0 = 0 - (a_n t_1 + \dots + a_1 t_n) \\ & \quad \vdots \\ & 0 = 0 - (a_n t_n) \end{aligned}$$

En approximation kan gå ud på at bestemme T, således at kvadratsummen af højresiderne i (5) minimeres. En anden metode er simpelthen at se bort fra de n sidste ligninger. Begge approximationer vil kun være gode, hvis alle a_i , c_i og t_i er små sammenlignet med 1. Lad et objekt være givet ved:

$$(6) \quad \begin{aligned} A &= 1 - 1.5q^{-1} + 0.7q^{-2} & k &= 1 \\ B &= 1 + 0.5q^{-1} & C &= 1 + c_1q^{-1} \end{aligned}$$

og lad:

$$(7) \quad T = 1 - \beta q^{-1} \quad \text{og} \quad A^m = 1 - \alpha q^{-1}$$

Den simple lineære, hvor de sidste n ligninger i (5) bortkastes, giver at:

$$(8) \quad \beta_\ell = - (c_1 - \alpha)$$

mens den kvadratiske lyder:

$$(9) \quad \beta_k = - \frac{c_1 + \alpha}{1 + \alpha^2}$$

I figurene 1 - 3 er vist 3 konjunkturplot for variansen af udgangssignalet, $y(t)$, som funktion af α og β , når $e(t)$ har variansen 1 og $w = 0$. De tre figurer adskiller sig ved, at c_1 er lig henholdsvis 0.4, 0 og -0.4. Figurene, der er produceret med programmet TAB, er forsynet med en kurve (+) over den optimale værdi for β som funktion af α . Samtidig er β_k afbildet (*). Det ses af disse figurer, at approximationerne er gode så længe α og β er numerisk mindre end ca. 0.4. Figur 4 viser variansen af styresignalet, $u(t)$, som funktion af α og β , hvor også β_k er indtegnet for $c_1 = 0$. Styreeffekten er næsten konstant for β_k og β_ℓ indenfor det område hvori approximationerne er gode.

ST. VARIANS FOR Y/E (I DB)

REG. TYPE I POL = TP2
ABCISSE I ALFA
ORDINAT I BETA

REF = 1,0000
SKALRINGSFAKTOR I 1,0000
AKSEFORHOLD I 1,0000

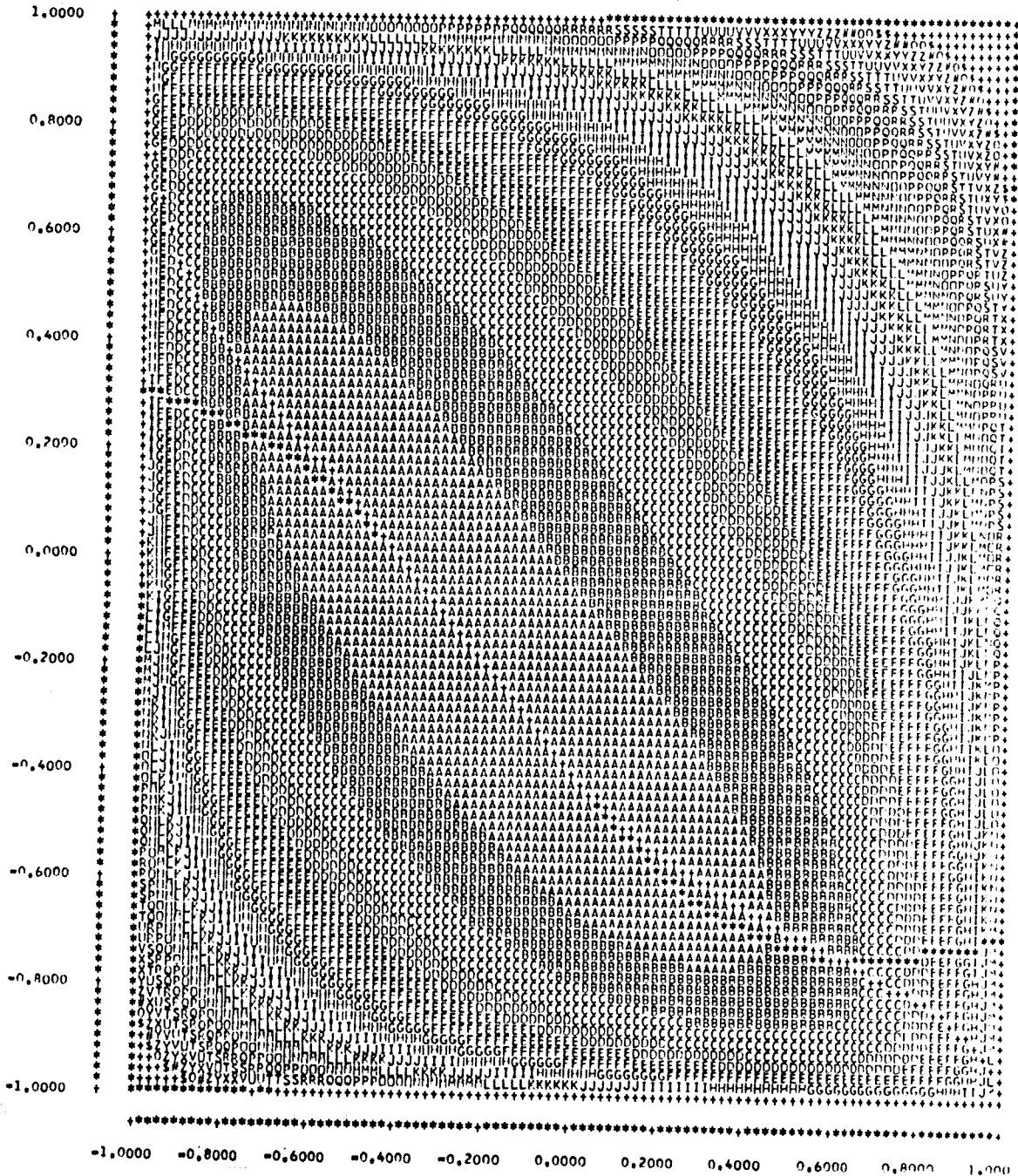


Fig. 1: Variansen af $y(t)$ som funktion af nulpunkterne i T^* og A^{m*} for $c_1 = 0.4$

ST. VARIANS FOR Y/E (I DB)

REG. TYPE I POL = TP2
ABCISSÉ I ALFA
ORDINAT I BETA

REF = 1,0000
SKALERINGSFAKTOR I 1,0000
AKSEFORHOLD I 1,0000

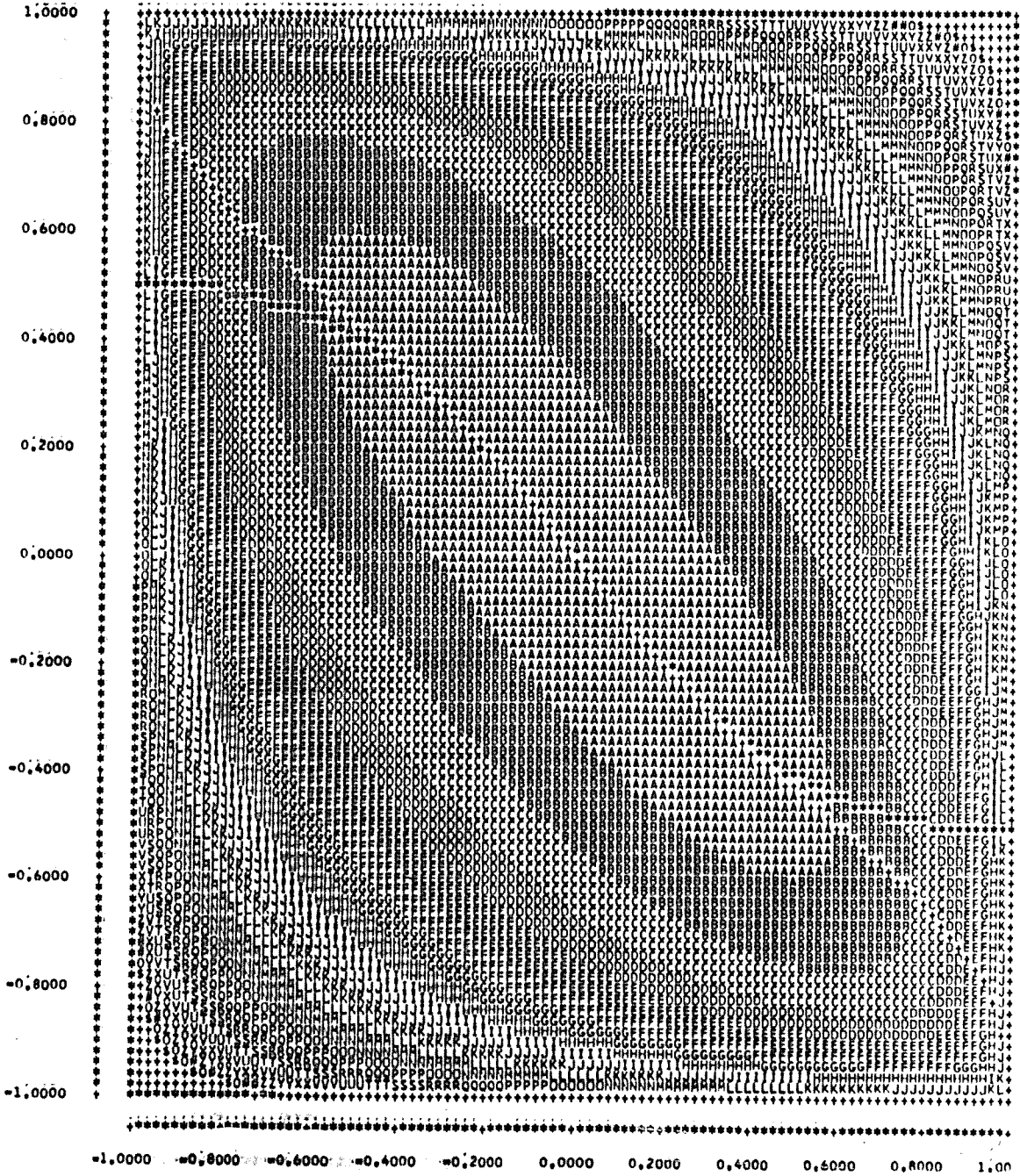


Fig. 2: Variansen af y(t) som funktion af α og β for $c_1 = 0$.

ST. VARIANS FOR Y/E (I DB)

REG. TYPE I POL = TP2
ABCISSE I ALFA
ORDINAT I BETA

REF = 1,0000
SKALERINGSFAKTOR I 1,0000
AKSEFORHOLD I 1,0000

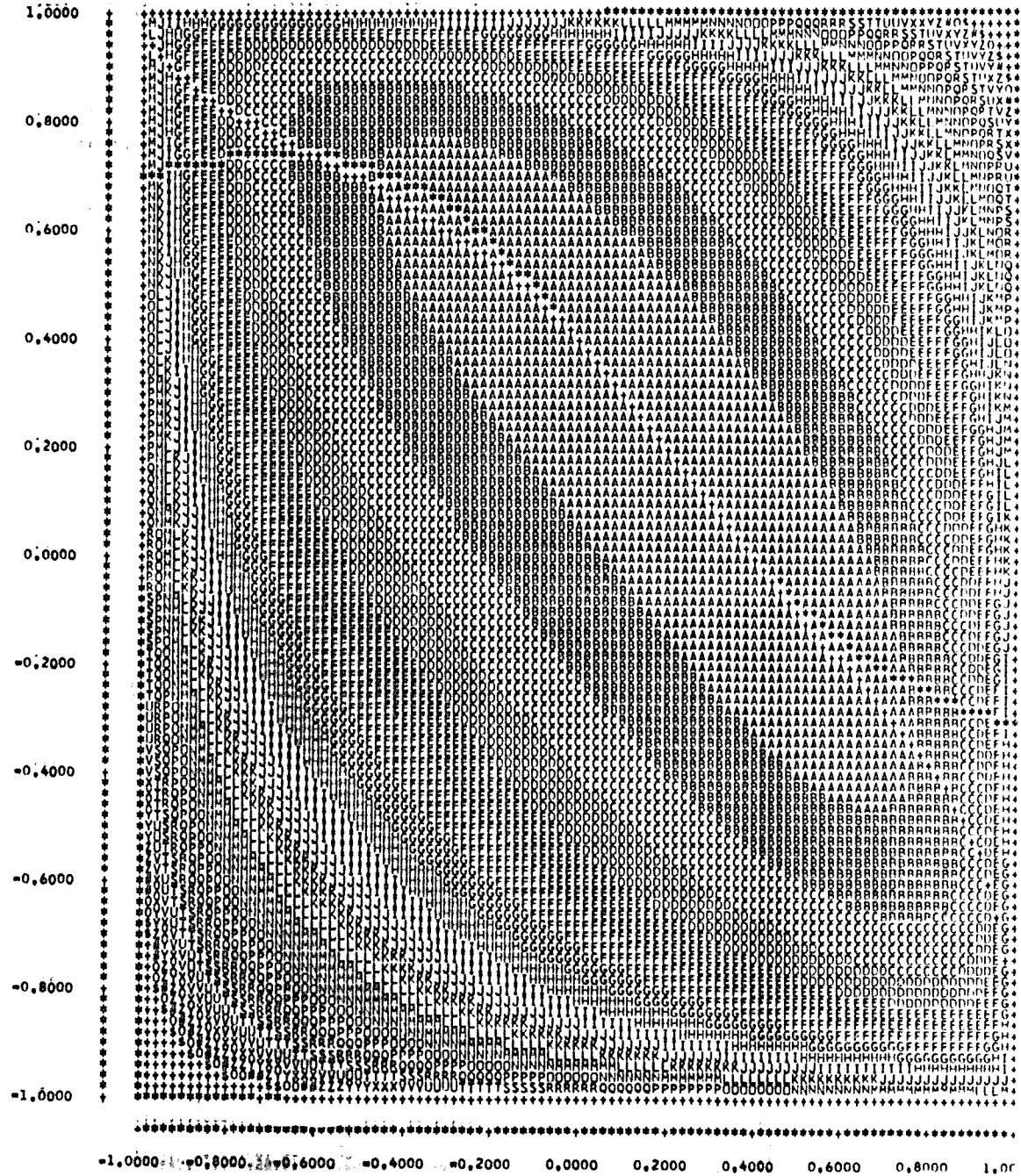


Fig. 3: Variansen af $y(t)$ som funktion af α og β for $c_1 = -0.4$.

ST. VARIANS FOR U/E (DB)

REG: TYPE I POL = TP2
ABCISS: ALFA
ORDINAT: BETA

REF = 1,0000
SKALERINGSFAKTOR: 1,0000
AKSEFORHOLD: 1,0000

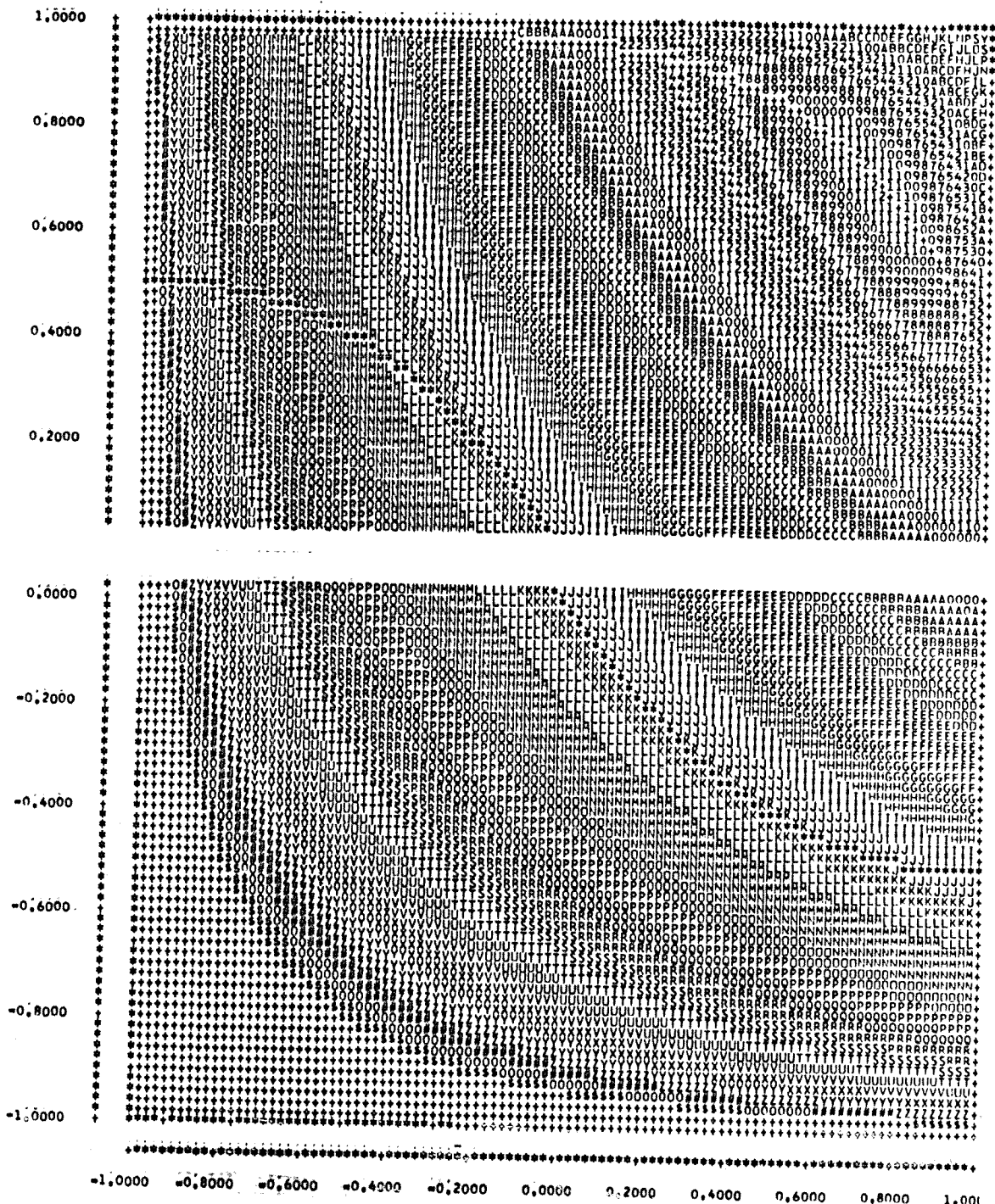


Fig. 4: Variansen af $u(t)$ som funktion af α og β for $c_1 = 0$.

APPENDIX 5.

=====

5. Overføringsfunktioner.

I dette appendix vil der blive angivet de overføringsfunktioner, der er anvendt i programmet (KONV) til udregning af AC - forstærkning og DC - fejl. Overføringsfunktionerne vil desuden blive anvendt til en simplificeret følsomhedsanalyse. Det antages, at forstyrrelserne er på formen:

$$(1) \quad v(t) = C(q^{-1})e(t) + d$$

og der vil blive anvendt den notation, der i det øvrige er brugt i forbindelse med den aktuelle regulatortype.

Deadbeatregulatorerne.

Da disse indeholder en integralvirkning, der eliminerer en stationær fejl, er DC - fejlen uinteressant. Dette er naturligvis ikke tilfældet for AC - forstærkningen. Antages det at forstyrrelserne er på formen (1), kan overføringsfunktionerne udregnes på grundlag af fig. 5.5,1, hvor

$$(2) \quad \begin{aligned} \hat{P}_3 &= \hat{q}_0 \hat{B} (1 - \alpha q^{-1}) \\ \hat{Q}_3 &= \hat{q}_0 \hat{A} (1 - \alpha q^{-1}) \end{aligned}$$

På grund af en senere anvendelse vil der blive givet flere overføringsfunktioner, end der er anvendt i kurveudskrifterne. Hvis polynomiet N_1 betegner

$$(3) \quad N_1 = A + q^{-k} (\hat{Q}_3 B - A \hat{P}_3) \rightarrow A$$

kan alle overføringsfunktioner bringes på formen:

$$(4) \quad H_{Y,e} = \frac{T_{Y,e}}{N_1}$$

Tællerpolynomierne er givet ved:

$$(5) \quad T_{Y,e} = C(1 - q^{-k} \hat{P}_3) \quad (+)$$

$$(6) \quad T_{u,e} = - \hat{Q}_3 C \quad (-)$$

$$(7) \quad T_{Y,w} = q^{-k} \hat{Q}_3 B \quad (*)$$

$$(8) \quad T_{u,w} = A \hat{Q}_3$$

I forbindelse med følsomhedsanalysen er overføringsfunktionerne:

$$(9) \quad H_{\epsilon,e} = \frac{CA}{\hat{C}N_1}$$

$$(10) \quad H_{\epsilon,w} = \frac{q^{-k} \hat{Q}_3 (\hat{A}B - A\hat{B})}{\hat{C} N_1}$$

af speciel interesse. De korrekte overføringsfunktioner er anført i afsnit 5.2.1.2 (d.v.s i udtrykkene (5.2.1.2,39 - 42)), idet DBo - regulatoren kan opfattes som et specialtilfælde af DBI - regulatoren. For de øvrige overføringsfunktioner er de korrekte:

$$(11) \quad H_{\epsilon,e} = 1 \quad H_{\epsilon,w} = 0$$

Polplaceringsregulatorerne.

Her er udregningerne kun foretaget for den deterministisk

dimensioneret version. For at lette notationen indføres to hjælpe polynomier:

$$(12) \quad \hat{R}^2 = b_0 p^2 + q^{-1} \hat{G} \quad \rightarrow BE$$

$$(13) \quad N_2 = \hat{R}^2 A + q^{-k} B \hat{F} \quad \rightarrow BTA^m$$

Hermed kan de interessante overføringsfunktioner skrives på formen (4) med N_2 og

$$(14) \quad T_{Y,W} = q^{-k} B^m T_B$$

$$(15) \quad T_{u,W} = A T_A^m$$

$$(16) \quad T_{Y,V} = \hat{R}^2$$

$$(17) \quad T_{u,V} = - \hat{F}$$

$$(18) \quad T_{\varepsilon,V} = T_A^m \hat{R}^2 \quad \rightarrow BTA^m E$$

$$(19) \quad T_{\varepsilon,V} = q^{-k} T_B^m (T_A^m B - N_2) \quad \rightarrow 0$$

Da

$$(20) \quad H_{Y,U_0} = q^{-k} \frac{P_4 T_A^m B}{Q_4 N_2}$$

er belastningens bidrag til fejlen for korrekte parameter

$$(21) \quad e_{st}(t) = \frac{E}{T_A^m} (1 - q^{-k}) d(t)$$

forudsat at estimatet af belastningen kan følge med denne. Ovenstående følge af nogle beregner ud fra fig 5.3.2,1. De korrekte overføringsfunktioner (5.3.1.1,32 - 35) er angivet i afsnit

5.3.1.1.

Minimalvariansregulatorerne.

Denne genrelle regulator type er temmelig kompliceret, hvorfor der bl. a af hensyn til beregningstiden under EXPRESS systemet er foretaget en opspaltning. Først betragtes det tilfælde, at alle objektets nulpunkter plomberes, det vil sigs, at $Q_1 = 0$. Dernæst vil tilfældet, hvor alle kriteriefunktionens overføringer er polynomier blive behandlet.

$Q_1 = 0$

Indføres polynomiet:

$$(22) \quad N_3 = A\hat{G}P_2 + q^{-k}BF \quad \rightarrow P_1B$$

kan overføringsfunktionerne bringes på formen:

$$(23) \quad H_{Y,w} = q^{-k} \frac{BP_2R_1}{N_3 R_2} \quad \rightarrow q^{-k} \frac{P_2R_1}{P_1R_1}$$

$$(24) \quad H_{u,w} = \frac{AP_2R_1}{N_3 R_2} \quad \rightarrow \frac{A}{B} \frac{P_2R_1}{P_1R_1}$$

$$(25) \quad H_{Y,e} = \frac{\hat{G}P_2}{N_3} \quad \rightarrow \frac{P_2}{P_1} E$$

$$(26) \quad H_{u,e} = - \frac{\hat{F}}{N_3} \quad \rightarrow - \frac{F}{BP_1}$$

Den stationære fejl bestemmes af (23) samt af

$$(27) \quad H_{Y,U_0} = q^{-k} \frac{BP_2}{N_3} \rightarrow q^{-k} \frac{P_2}{P_1} \quad (U_0 = - Ed)$$

$$(28) \quad H_{Y,d} = \frac{\hat{GP}_2}{N_3} \rightarrow \frac{EP_2}{P_1}$$

$$P_2 = Q_2 = R_2 = 1$$

Også i dette tilfælde indføres et hjælpepolynomie

$$(29) \quad N_4 = \hat{AG} + q^{-k} \hat{BF} \rightarrow BP_1 + AQ_1$$

hvorved de brugte overføringsfunktioner er

$$(30) \quad H_{Y,w} = q^{-k} \frac{BR_1}{N_4} \rightarrow q^{-k} \frac{BR_1}{BP_1 + AQ_1}$$

$$(31) \quad H_{u,w} = \frac{AR_1}{N_4} \rightarrow \frac{AR_1}{BP_1 + AQ_1}$$

$$(32) \quad H_{Y,e} = \frac{\hat{G}}{N_4} \rightarrow \frac{BE + CQ_1}{BP_1 + AQ_1}$$

$$(33) \quad H_{u,e} = - \frac{\hat{F}}{N_4} \rightarrow - \frac{F}{BP_1 + AQ_1}$$

samt

$$(34) \quad H_{Y,d} = \frac{\hat{G}}{N_4} \rightarrow \frac{BE + CQ_1}{BP_1 + AQ_1}$$

$$(35) \quad H_{Y,U_0} = q^{-k} \frac{B}{N_4} \rightarrow q^{-k} \frac{B}{BP_1 + AQ_1}$$

---ooo0ooo---

APPENDIX 6

=====

6. Specifikke overføringsfunktioner for objekt II.

I dette appendix angives overføringsfunktionerne, der er anvendt i afsnit 6.2. Notationen skal derfor ses i sammenhæng med de respektive underafsnit.

MVo - regulering.

$$\begin{aligned}
 (1) \quad H_{Y,w} &= q^{-1} \frac{b_o}{\beta + q^{-1}(\beta a + \hat{f}_o b_o)} \rightarrow q^{-1} \\
 (2) \quad H_{u,w} &= \frac{1 + aq^{-1}}{\beta + q^{-1}(\beta a + \hat{f}_o b_o)} \rightarrow \frac{1 + aq^{-1}}{b_o} \\
 (3) \quad H_{Y,e} &= \frac{\beta}{\beta + q^{-1}(\beta a + \hat{f}_o b_o)} = H_{\epsilon,e} \rightarrow 1 \\
 (4) \quad H_{u,e} &= - \frac{\hat{f}_o}{\beta + q^{-1}(\beta a + \hat{f}_o b_o)} \rightarrow \frac{a}{b_o} \\
 (5) \quad H_{\epsilon,w} &= \frac{b_o - \beta - q^{-1}(\beta a + \hat{f}_o b_o)}{\beta + q^{-1}(\beta a + \hat{f}_o b_o)} \rightarrow 0
 \end{aligned}$$

→ Når objektet er givet ved:

$$A = 1 + aq^{-1}$$

$$B = b_o$$

og β betegner a priori estimatet af b_o

Polplacering.

Med modellen

$$A^m = 1 + \alpha q^{-1}$$

$$B^m = 1 + \alpha \quad (T = 1)$$

er overføringsfunktionerne

$$(9) \quad H_{Y,W} = q^{-1} \frac{b_0(1+\alpha)}{\beta + q^{-1}(\beta a + \hat{f}_0 b_0)} \rightarrow q^{-1} \frac{1 + \alpha}{1 + \alpha q^{-1}}$$

$$(10) \quad H_{u,W} = \frac{(1+\alpha q^{-1})(1+\alpha)}{\beta + q^{-1}(\beta a + \hat{f}_0 b_0)} \rightarrow \frac{(1+\alpha q^{-1})(1+\alpha)}{(1+\alpha q^{-1})b_0}$$

$$(11) \quad H_{Y,e} = \frac{\beta}{\beta + q^{-1}(\beta a + \hat{f}_0 b_0)} \rightarrow \frac{1}{1 + \alpha q^{-1}}$$

$$(12) \quad H_{u,e} = - \frac{\hat{f}_0}{\beta + q^{-1}(\beta a + \hat{f}_0 b_0)} \rightarrow - \frac{a - \alpha}{b_0}$$

samt

$$(13) \quad H_{\varepsilon,W} = (1 +) \frac{b_0 - \beta + q^{-1}(\beta a + \hat{f}_0 b_0)}{\beta + q^{-1}(\beta a + \hat{f}_0 b_0)} \rightarrow 0$$

$$(14) \quad H_{\varepsilon,e} = \frac{\beta(1+\alpha q^{-1})}{\beta + q^{-1}(\beta a + \hat{f}_0 b_0)} \rightarrow 1$$

DBo - regulering.

Hvis \hat{a} og b betegner estimaterne

$$\hat{A} = 1 + \hat{a}q^{-1}$$

$$\hat{B} = b$$

kan der anvendes samme notation som i appendix 5 og der fås:

$$(16) \quad N_1 = 1 + (a + \frac{b_0}{b} - 1)q^{-1} + (a\frac{b_0}{b} - a)q^{-2} \\ \rightarrow 1 + aq^{-1}$$

og videre at:

$$(17) \quad T_{y,e} = 1 - q^{-1}$$

$$(18) \quad T_{u,e} = \frac{1}{b} (1 + \hat{a}q^{-1}) \rightarrow \frac{1}{b_0} (1 + aq^{-1})$$

$$(20) \quad T_{y,w} = q^{-1} \frac{b_0}{b} (1 + \hat{a}q^{-1}) \rightarrow q^{-1} (1 + aq^{-1})$$

$$(21) \quad T_{u,w} = \frac{1}{b} (1 + (\hat{a} + a)q^{-1} + \hat{a}aq^{-2}) \rightarrow \frac{1}{b_0} (1 + aq^{-1})^2$$

$$(22) \quad T_{\epsilon,e} = 1 + \hat{a}q^{-1} \rightarrow (1 + aq^{-1})$$

$$(23) \quad T_{\epsilon,w} = q^{-1} [(\frac{b_0}{b} - 1) + (2a\frac{b_0}{b} - \hat{a} - a)q^{-1} \\ + (\hat{a}^2\frac{b_0}{b} - \hat{a}a)q^{-2}] \rightarrow 0$$

DB1 - regulering.

Med det pågældende valg af tuningparameter er forholdene for denne type mere kompliceret end for DBo - regulatoren. Her er det ekstra indførte nulpunkt

$$\alpha = \hat{a}$$

Næyrerpolynomiet er i dette tilfælde:

$$(28) \quad N_1 = 1 + \left(\frac{ab(1-\hat{a}) + b_0 - b}{b_0(1-\hat{a})} \right) q^{-1} + \frac{\hat{a} - a}{1 - \hat{a}} q^{-2} \\ + \frac{\hat{a}ba - \hat{a}^2 b_0}{(1-\hat{a})b} q^{-3} \rightarrow 1 + aq^{-1}$$

mens tallerpolynomierne er:

$$(29) \quad T_{y,w} = q^{-1} b_0 \frac{1 - \hat{a}^2 q^{-2}}{(1-\hat{a})b} \rightarrow q^{-1} (1 - a^2 q^{-2})$$

$$(30) \quad T_{u,w} = \frac{1 + aq^{-1} - \hat{a}^2 q^{-2} - \hat{a}^2 a q^{-3}}{(1-\hat{a})b}$$

$$(31) \quad T_{y,e} = 1 + \frac{1}{\hat{a}-1} q^{-1} + \frac{\hat{a}}{1-\hat{a}} q^{-2}$$

$$(32) \quad T_{u,e} = \frac{1 - \hat{a}^2 q^{-2}}{(1-\hat{a})b}$$

$$(33) \quad T_{\epsilon,e} = 1 + \hat{a}q^{-1}$$

$$(34) \quad T_{\epsilon,w} = q^{-1} \frac{(b_0 - b) + (\hat{a}b_0 - ba)q^{-1} - \hat{a}^2(b_0 - b)q^{-2}}{(1-\hat{a})b} \\ - \frac{\hat{a}^2(\hat{a}b_0 - ba)q^{-3}}{(1-\hat{a})b} \rightarrow 0$$

---ooo0ooo---

APPENDIX 7

=====

7. Simplificeret stabilitet.

I dette appendix vil der blive givet nogle heuristiske argumenter for at tilbagekoblingen i fig. 7.7,2 er stabil. En egentlig stabilitetsanalyse vil kræve anvendelse af Lyapunov - teorien for tidsdiskrete systemer. En sådan analyse er temmelig tidskrævende. Da eksperimentelle undersøgelser viser, at systemet er stabilt er en undersøgelse af den robuste identifikation prioriteret højere end en Lyapunov undersøgelse. I det følgende vil der blive anvendt nogle stationære betragtninger på det simplificerede system i fig. 1.

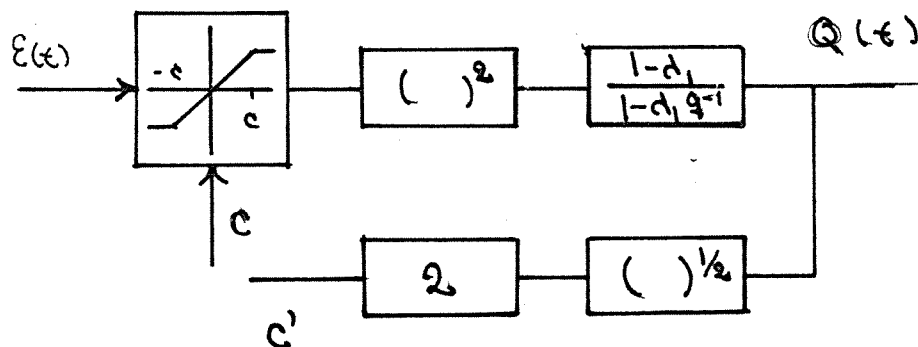


Fig. 1: Simplificeret robust estimator af støjvariansen.

Systemet er karakteriseret ved at være meget langsomt idet $\lambda_1 = 0.985$.

$$\epsilon(t) = A$$

Antag først af $\epsilon(t)$ er konstant og at $c > A$. Med åben sløjfe vil $Q(t)$ asymptotisk nærme sig til A^2 , hvor c' vil nærme sig til $2A$. Såfremt $\epsilon(t)$ springer til A_1 ($c > A_1$) vil c' følge langsomt efter til $2A_1$. Antag nu, at $\epsilon(t)$ springer til A_2 hvor $c < A_2$. Da vil $Q(t)$, såfremt sløjfen er åben, stige eksponentielt til c^2 . Med lukket sløjfe vil $Q(t)$ stige mod A_2^2 . Udgangen $Q(t)$ vil være fastlåst hvis $c(0) = 0$.

$$\epsilon(t) = A \sin(\omega t)$$

Antag at ω er meget større end $(\log(1/\lambda)/T_2)$ og at $c > A$. Da vil $Q(t)$ indstille sig på $\frac{1}{2}A^2$ (åben sløjfe) og dermed vil c' svinge ind mod $\sqrt{2}A > A$. Antag nu, at $0 < c \ll A$. Da kan $\epsilon_2(t)$ approximeres med en firkantbølge. For en sådan vil $Q(t)$ svinge op mod c^2 og dermed vil c' tilnærme værdien $2c$. Hvis sløjfen lukkes vil c' derfor stige indtil $c = \sqrt{2}A$. Hvis frekvensen sænkes vil c kunne varieres med tidsvariationerne på $\epsilon(t)$

$$\epsilon(t) \in N(0, \sigma^2)$$

Hvis $\epsilon(t)$ er en sekvens af uafhængige normalfordelte stokastiske variable vil $Q(t)$ være et rimeligt estimat for σ^2 såfremt $c > \sigma$ og sløjfen er åben. Størrelsen c' vil derfor tilnærmet antage værdien 2σ . Hvis sløjfen lukkes vil den lukke sammen. Antag nu at sløjfen er åben og at $c \ll \sigma$. Da vil $Q(t)$ tilnærmet være $2c^2 \Phi(-c/\sigma)$, hvilket svarer til, at kun signaler i grænsen indgår. Hjælpe størrelsen c' vil have værdien $2c$. Følgelig vil grænsen i lukket sløjfesystemet udvides. Resultatet er kvalitativt, at den lukkede sløjfe vil indstille sig således, at c er lidt mindre end 2σ

APPENDIX 8

=====

8. Brugervejledning.

Dette appendix skal ses som en meget kort brugervejledning til de fire programmer, der er udviklet og anvendt i projektet. Hovedprogramdelen for simulationsprogrammerne, PZ (polplacering), DB (deadbeatregulering) og MV (minimalvariansregulering), er listet i fig. 1 - 3. Samtidig er inddata til programmerne listet. De indgående underprogrammer, så som f. eks. FILTER og RML (estimationsprocedure), er ligesom selve programmerne listet i appendix 9. Herunder er givet en oversigt over underprogrammernes funktion.

INPUT	Varetager indlæsning, initialisering samt indledende beregninger af korrekte regulatorparameter samt forventede varianser. Desuden foretages en udskrivning af bl. a. inddata.
DELAY	Procedure til forsinkelse af signaler.
REF	Generator af en referencesignal
STOEJ	Støjgenerator. Spidserne indgår også heri.
MARKOV	Genererer belastningsspring.
FILTER	Et digitalt filter.
RML	Generel estimationsprocedure. Indeholder både RLS, RELS samt RML. Proceduren kan også foretage en robust estimation.

- GLR2 Procedure der både varetager estimationen af støjvariansen og detektionen af spring i belastningen.
- OUT1 Udskrivning af signaler samt analyse herpå
- KONV Udskriver en kurve over AC - forstærkningen samt DC - fejlen.
- OUT2 Udskriver kurver over de estimerede parameter.

I hovedprogrammet optræder 5 styrevariable, som sættes umiddelbart efter inputroutinen. Deres virkemåde er forklaret herunder.

- ISTOP Angiver stoptidspunktet for simulationen. stoptidspunktet er ISTOP - 1.
- ISTART Angiver starttidspunktet for udskriften af kurver fra analyseret data.
- IP Styrevariabel til bestemmelse af den på gældende analyse, der ønskes foretaget
- | | |
|---|-------------------|
| 1 | Signalanalyse |
| 2 | Konvergensanalyse |
| 3 | Parameteranalyse |
- IR Hvis denne variabel er 1 fremkommer den robuste parameterestimation.
- IST Hvis denne er 1 vil detektoren kunne aktivere den unormale arbejdstilstand for estimations algoritmen.

Inddata består generelt af objektets karakteristika så som polynomierne A, B og C efterfulgt af tidsforsinkelsen, k, støj - variansen, VAR, samt den konstante belastning, D,. Objektbeskrivelsen er efterfulgt af en angivelse af regulatorens dimensioneringskriterie samt a priori information til estimationsalgoritmen. For indlæsning af polynomier gælder at antallet af koefficienter indlæses umiddelbart før selve polynomiet. Indlæsningen af disse data er specifik for de tre programmer.

MV

I dette program er indlæsningen af objektets karakteristika efterfulgt af kriteriefunktionens polynomier P_1 , P_2 , Q_1 , Q_2 , R_1 og R_2 . Hvis styrevariablen IREGTP er tildelt en negativ vil reguleringen blive foretaget med en fast korrekt justeret regulator. Er dennes værdi positiv vil den selvindstillende regulator fremkomme. Den indlæste værdi for QD svarer til Q_d^0 i det udvidede kalmanfilter. Herefter indlæses informationen til estimationsalgoritmen indlæst. NRELS er en styrevariabel, der bestemmer hvilken estimationsmetode, der ønskes anvendt.

o RELS

< o RLS

> o RML

De indgående parameterantal bliver justeret i overensstemmelse med dette valg. Størrelsen G0 svarer naturligvis til a priori estimatet af G_0 . Herefter følger indlæsningen af ordenerne, n_A , n_B , tidsforsinkelsen, k, samt n_C . Disse er ikke nødvendigvis de samme som de korrekte.

DB

I forhold til MV er angivelsen af tuningparameter begrænset til Q0, der i procent svarer til β for DB1 - regulatoren. Hvis denne er 0 fås som nævnt en DBo - regulering.

PZ

Udover tuningparameterne er størrelserne BETA og DELTA specielle for dette program. Hvis BETA er tildelt værdien nul vil der fremkomme en almindelig polplaceringsregulering. Er den derimod 1 vil der være indlagt en integralvirkning i regulatoren. DELTA er nulpunktet svarende til polynomiet D fra afsnit 7.5. Ordenen NCE anvendes her som en styrevariabel. Er værdien af denne negativ vil den deterministiske version fremkomme.

Programmet TAB er listet i fig. 4. Som det fremgår heraf er den undesøgte overføringsfunktion indsat i linierne 460 - 520. Den korrekte er angivet i linierne 360 - 390. Parameter - variationerne kan ændres i linie 320 - 330.

```

00460 C
00470 C$PRINTON
00480 C*****
00490 C*****
00500 C**
00510 C**
00520 C**
00530 C**
00540 C*****
00550 C
00560 C
00570     ISTOP=501
00580     IP=1
00590     ISTART=1
00600     IINT=(ISTOP-ISTART)/100
00610     CALL INPUT(NN,A,N,B,NB,C,NC,D,K,VAR,Q1,IREGTP)
00620     SPR=SQRT(VAR)
00621     IR=1
00622     IST=0
00630     DO 10 I=1,ISTOP
00640         CALL DELAY(K,XX,Y,U)
00650         W=REF(I,25,0)
00660         CALL STOEJ(I,SPR,25.0,1,0.02,E)
00670 C         CALL MARKOV(I,5.0,0.01,D)
00680 C         CALL PERTUB(I,E,VAR,1.5,5.0)
00690 C         IF (I.GT.40) D=1.00
00700 C         IF (I.GT.40) D=AMIN1(D+0.01,1.00)
00710 C         CALL GAUSS(NN,SPR,0.0,E)
00720 C         IF (I.EQ.7) E=30.00
00730         Y=FILTER(1,A,N,B,NB,XX(K,2))+FILTER(2,A,N,C,NC,E)
00740     >         +FILTER(3,A,N,ET,0,D)
00750         ESTYRE=W-Y
00770         IF (IREGTP.GE.0) CALL RML(Y,XX(K,2),EPS,EPS2,IR,GR,IGR,QE,CD,CE
00780     >         ,TRP,PD)
00790         IF (IGR.EQ.1) WRITE(6,1020) I
00810         CALL GLR2(I,10,100.0,EPS2,QE,GR,Y,XX(K,2),IST,IR,THRESH)
00820         CALL REGPAR(Q1,Q,NQ,P,NP,IREGTP)
00830         U=FILTER(8,P,NP,Q,NQ,ESTYRE)
00840         IF (IP.EQ.1) CALL KONV(I,ISTART-1,IINT,P,NP,Q,NQ,A,N,B,NB,C,NC
00850     >         ,K,VAR,CD,CE,THRESH)
00860         IF (IP.EQ.2) CALL OUT1(I,ISTART-1,IINT,Y,U,W,ESTYRE,EPS,VAR,D)
00870         IF (IP.EQ.3) CALL OUT2(I,ISTART-1,IINT,A,B,C,D,TRP,PD)
00880         IG=IG+IGR
00890 10     CONTINUE
00900         WRITE(6,1010) IG
00910         WRITE(6,1000)
00920 1000     FORMAT(1H1)
00930 1010     FORMAT(1H-, 'ANTAL GRÆNSETILFELDE : ',I3)
00940 1020     FORMAT(1H-,T90, 'GRÆNSETILFELDE P$ TRIN : ',I3)
00950         STOP
00960         END
00970 C
00980 C$PRINTOFF
00990 C
01000 C*****
01010 C*****
01020 C**
01030 C**
01040 C**
01050 C*****
01060 C*****
01070 C

12480 C*****
12490 C
12500 $ENTRY
12510 3     1.0000    -01.5000    00.7000    -00.0000    (N + 1),A0,A1,...,AN
12520 2     1.0000     00.5000    -01.8920
12530 1     1.0000    -00.9000
12540 3     0.1000     0.0000
12550 5     000.0
12560 -5    1.0000     00.0000
12570 2
12580 1
12590 3
12600 0

```

Fig. 1: DB

=====


```

00440 C*****
00450 C*****
00460 C**
00470 C**
00480 C**
00490 C*****
00500 C*****
00510 C
00520 C
00530 CALL INPUT(NN,A,N,B,NB,C,NC,D,K,VAR,P1,NP1,Q1,NQ1,R1,NR1,
00540 > P2,NP2,Q2,NQ2,R2,NR2,IREGTP)
00550 ISTART=1
00560 ISTOP=301
00570 IP=2
00580 IST=0
00590 SPR=SQRT(VAR)
00600 IR=0
00610 IINT=(ISTOP-ISTART)/100
00620 DO 10 I=1,ISTOP
00630 CALL DELAY(K,XX,Y1,U1,W1,U)
00640 W=REF(I,0)
00650 C CALL STOEL(I,SPR,25.0,1,0.02,E)
00660 C IF (I.GT.40) D=AMIN1(D+0.01,1.00)
00670 C CALL MARKOV(I,5.0,0.01,D)
00680 C IF (I.GT.40) D=1.00
00690 C IF (I.EQ.41) E=30.00
00700 C CALL PERTUB(I,E,VAR,1.5,5.0)
00710 CALL GAUSS(NN,SPR,0.0,E)
00720 Y=FILTER(1,A,N,B,NB,XX(K,2))+FILTER(2,A,N,C,NC,E)
00730 > +FILTER(3,A,N,ET,0,D)
00740 ESTYRE=Y-W
00750 W1=FILTER(16,R2,NR2,R1,NR1,W)
00760 Y1=FILTER(17,P2,NP2,ET,0,Y)
00770 FI=FILTER(4,ET,0,P1,NP1,Y1)+FILTER(5,ET,0,Q1,NQ1
00780 > ,XX(K,2))-XX(K,3)
00790 YE1=FI
00800 YE2=XX(K,1)
00810 YE3=XX(K,2)
00820 YE4=XX(K,3)
00830 CALL RML(FI,XX(K,1),XX(K,2),XX(K,3),EPS,
00840 > EPS2,IR,GR,IGR,QE,CD,DELTA,TRP,PD)
00850 CALL GLR2(I,10,100.0,EPS2,QE,GR,YE1,YE2,YE3,YE4,IST,IR,THRESH)
00860 IG=IG+IGR
00870 IF (IGR.EQ.1) WRITE(6,1020) I
00880 CALL REGPAR(IREGTP,G,NG,F,NF,H,NH,GAMMA)
00890 U1=-FILTER(7,G,NG,F,NF,Y1)-FILTER(8,G,NG,H,NH,W1)
00900 > -FILTER(9,G,NG,ET,0,GAMMA)
00910 U=FILTER(6,ET,0,Q2,NQ2,U1)
00920 IF (IP.EQ.1) CALL OUT1(I,ISTART-1,IINT,Y,U,W,ESTYRE,FI,
00930 > EPS,VAR,D,E)
00940 IF (IP.EQ.3) CALL OUT2(I,ISTART-1,IINT,K,TRP,PD)
00950 IF (IP.EQ.2) CALL KONV1(I,ISTART-1,IINT,A,N,B,NB,K,VAR,G,NG,F,NF,
00960 > P1,NP1,Q1,NQ1,R1,NR1,D,GAMMA,W,THRESH,CD,DELTA)
00970 IF (IP.EQ.4) CALL KONV2(I,ISTART-1,IINT,A,N,B,NB,K,VAR,G,NG,F,NF
00980 > ,P1,NP1,P2,NP2,R1,NR1,R2,NR2,D,GAMMA,W,THRESH,CD,
00990 > DELTA)
01000 10 CONTINUE
01010 WRITE(6,1000) IG
01020 WRITE(6,1010)
01030 1000 FORMAT(1H-, 'ANTAL GRÆNSE TILFÆLDE : ',I4)
01040 1010 FORMAT(1H1)
01050 1020 FORMAT(1H-,T90, ' GRAENSE P$ TRIN : ',I3)
01060 STOP
01070 END

```

Fig. 2: MV

=====

Line	Column	Value	Value	Value	Value	Description
14370	\$ENTRY					
14380	3	1.0000	-01.5000	00.7000	-00.9277	(NA+1),AO,A1,
14390	2	01.0000	00.5000	00.1892		(NB+1),BO,B1,...
14400	1	1.0000	-00.9000			(NC+1),CO,C1,...
14410	1	0.1000	0.0000			K,VAR,D
14420	3	1.0000	-0.3980	0.1353		(NP1+1),P10,P11,...
14430	1	0.0000	-00.2500			(NQ10+10),Q100,Q1010,
14440	1	0.5820	00.1555			(NR1+1),R10,R11,...
14450	2	1.0000	0.2670			(NP2+1),P20,P21,...
14460	1	1.0000				(NQ2+1),Q20,Q21,...
14470	1	1.0000	-0.3980	0.1353		(NR2+1),R20,R21,...
14480	5	0.0000				IREGTP,QD
14490	-5	1.0000				NRELS,G0
14500	2					NE
14510	1					NBE
14520	1					KE
14530	0					NCE

```

00480 C*****
00490 C**
00500 C**                HOVEDPROGRAM                **
00510 C**                **
00520 C*****
00530 C*****
00540 C
00550 C
00560 CALL INPUT(NN,A,N,B,NB,C,NC,D,K,VAR,P,NP,Q,NQ,AM,NAM,BM,NBM,
00570 > TAM,NTAM,TAMP1,NTAMP1,P2,NP2,B0,BETA,IREGTP,
00580 > T,NT)
00590 ISTOP=401
00600 ISTART=1
00610 IR=1
00620 IP=2
00630 IST=0
00640 IINT=(ISTOP-ISTART)/100
00650 SPR=SQR(VAR)
00660 DO 10 I=1,ISTOP
00670 CALL DELAY(K,XX,U,U1,U2,YM1,Y2,WF)
00680 W=REF(I,25,0)
00690 IF (I.GE.20) CALL STOEJ(I,SPR,25.0,1,0.02,E)
00700 C IF (I.GT.40) D=AMIN1(D+0.01,1.00)
00710 C CALL MARKOV(I,5.0,0.01,D)
00720 IF (I.LT.20) CALL GAUSS(NN,SPR,0.0,E)
00730 C IF (I.GT.40) D=1.0
00740 C IF (I.EQ.41) E=30.00
00750 C CALL PERTUB(I,E,VAR,1.5,5.0)
00760 Y=FILTER(1,A,N,B,NB,XX(K,1))+FILTER(2,A,N,C,NC,E)
00770 > +FILTER(3,A,N,ET,0,D)
00780 ESTYRE=W-Y
00790 YM1=FILTER(4,AM,NAM,BM,NBM,W)
00800 YM=XX(K,4)
00810 YF=FILTER(5,P,NP,Q,NQ,Y)
00820 Y2=FILTER(6,TAM,NTAM,ET,0,YF)
00830 WF=FILTER(7,P,NP,Q,NQ,YM1)
00840 YE1=YF-B0*XX(K,2)
00850 YE2=XX(K,5)
00860 YE3=XX(K+1,3)
00870 YE4=XX(K,6)
00880 CALL RML(YE1,YE2,YE3,YE4,EPS,EPS2,IR,GR,TRP,PD,IGR,QE,CD,CE)
00890 IF (IGR.EQ.1) WRITE(6,6060) I
00900 6060 FORMAT(1H-,T90,' GRAENSE P$ TRIN : ',I3)
00910 CALL GLR2(I,10,100.0,EPS2,QE,GR,YE1,YE2,YE3,YE4,IST,IR,THRESH)
00920 CALL REGPAR(IREGTP,G,NG,F,NF,H,NH,UO)
00930 U1=FILTER(18,ET,0,H,NH,WF)-FILTER(8,ET,0,G,NG,XX(1,3))
00940 > -FILTER(9,ET,0,F,NF,Y2)+UO
00950 U1=U1/B0
00960 U2=FILTER(17,P2,NP2,ET,0,U1)
00970 U =FILTER(16,Q,NQ,TAMP1,NTAMP1,U1)+BETA*U
00980 EF=YF-WF
00990 IF (IP.EQ.1) CALL OUT1(I,ISTART-1,IINT,Y,YM,U,W,ESTYRE,EF,EPS,
01000 > VAR,D,E)
01010 IF (IP.EQ.2) CALL KONV(I,ISTART-1,IINT,A,N,B,NB,C,NC,K,P2,NP2,
01020 > Q,NQ,T,NT,CD,CE,THRESH,
01030 > P,NP,AM,NAM,BM,NBM,F,G,VAR,UO,D,W)
01040 IF (IP.EQ.3) CALL OUT2(I,ISTART-1,IINT,K,TRP,PD)
01050 IGRAEN=IGRAEN+IGR
01060 10 CONTINUE
01070 WRITE(6,1000) IGRAEN
01080 WRITE(6,1010)
01090 1000 FORMAT(1H-, 'ANTAL GRÆNSE TILFÆLDE : ',I4)
01100 1010 FORMAT(1H1)
01110 STOP
01120 C$PRINTOFF
01130 END
14540 C*****
14550 $ENTRY
14560 3 1.0000 -01.5000 00.7000 -00.2019 (N+1),A0,A1,...,AN
14570 2 1.0000 00.5000 -00.2290 (NB+1),B0,B1,...,BNB
14580 1 1.0000 -00.8000 00.1500 (NC+1),C0,C1,...,
14590 3 0.1000 0.0000 K,VAR,D
14600 1 1.0000 (NP1+1),P1
14610 1 1.0000 00.1000 (NP2+1),P2
14620 1 1.0000 -00.3980 00.1353 0.0447 00.0031 Q
14630 1 1.0000 00.3980 00.0231 (NT+1),T
14640 1 1.0000 -00.3980 00.1353 (NAM+1),AM
14650 1 1.0000 00.1552 00.0000 (NBM+1),BM
14660 5 0.0000 -00.0000 IREGTP,BETA(1/0),DELTA
14670 -5 1.0000 0.0050 NRELS,B0,QD
14680 2 NE
14690 1 NBE
14700 3 KE
14710 -1 NCE

```

Fig. 3: PZ

=====


```

01090 RHJ=FLOAT(IO-1)*TRINO+STARTO
01100 WRITE(6,1010) RHJ,KANT2(IO),(PAGE(IO,I1),I1=1,101)
01110 DO 50 J=1,9
01120 WRITE(6,1020) KANT2(IO+J),(PAGE(IO+J,I1),I1=1,101)
01130 CONTINUE
01140 RHJ=100.0*TRINO+STARTO
01150 WRITE(6,1010) RHJ,KANT2(101),(PAGE(101,I1),I1=1,101)
01160 WRITE(6,1030) (KANT1(I),I=1,101),(R(K),K=1,11)
01170 WRITE(6,1000)
01180 FORMAT(1H1)
01190 FORMAT(1H ,F8.4,3X,A1,3X,101A1)
01200 FORMAT(1H ,11X,A1,3X,101A1)
01210 FORMAT(1H-,15X,101A1//11X,11(F8.4,2X))
01220 FORMAT(1H0,T30,'AC FOELSMHEDEN AF OVERFOERINGEN ',A5,//
01230 > T30,'-----')
01240 FORMAT(1H0,ORDINAT : FO,T60,'AKSEFORHOLD : ',F8.4)
01250 FORMAT(1H0,ABCISSE : F1,T60,'SKALERINGSFAKTOR : ',F8.4)
01260 FORMAT(1H0,REG. TYPE : MV2,T60,REF = ',F8.4)
01270 FORMAT(1H-)
01280 FORMAT(1H0,T30,'DC FOELSMHEDEN AF OVERFOERINGEN ',A5,//
01290 > T30,'-----')
01300 STOP
01310 END
01320 C
01330 C
01340 C*****
01350 REAL FUNCTION SALOSS(A1,B1,N,IERR)
01360 C*****
01370 C
01380 C
01390 C
01400 C 1/(2*PI*I)*B(Z)*B(1/Z)/(A(Z)*A(1/Z)*Z)
01410 C
01420 C RUNDT LANGS ENHEDSCIRKLEN. HVIS A IKKE HAR ALLE ROEDDER INDENFOR
01430 C ENHEDSCIRKLEN RETURNERES VAERDIEN -10.00
01440 C*****
01450 C*****
01460 C
01470 C VARIABELLISTE:
01480 C =====
01490 C
01500 C N ORDER AF SYSTEM
01510 C A ARRAY INDH. NAEVNERPOLYNOMIET
01520 C B ARRAY INDH. TAELLERPOLYNOMIET
01530 C
01540 C*****
01550 DIMENSION A(6),A1(6),B(6),B1(6),AS(7)
01560 IERR=0
01570 IF (A1(1).GT.0.0) GO TO 70
01580 IN=N+1
01590 DO 80 I=1,IN
01600 B1(I)=-B1(I)
01610 A1(I)=-A1(I)
01620 70 CONTINUE
01630
01640
01650
01660 A(I)=A1(I)
01670 B(I)=B1(I)
01680 V=0.00
01690 DO 10 K=1,N
01700 L=N+1-K
01710 L1=L+1
01720 ALFA=A(L1)/A(1)
01730 BETA=B(L1)/A(1)
01740 V=V+BETA*B(L1)
01750 DO 20 I=1,L
01760 M=L+2-I
01770 AS(I)=A(I)-ALFA*A(M)
01780 B(I)=B(I)-BETA*A(M)
01790 IF (AS(1)) 50,50,30
01800 30
01810 40
01820 10 CONTINUE
01830 V=V+B(1)**2/A(1)
01840 SALOSS=V/A0
01850 RETURN
01860 50 CONTINUE
01870 IERR=1
01880 SALOSS=-10.0
01890 RETURN
01900 END
01910 C
01920 C
01930 REAL FUNCTION DCAMP(A,B,N)
01940 DIMENSION A(6),B(6)
01950 REAL RKA,RKB
01960 RKA=0.0
01970 RKB=0.0
01980 IN=N+1
01990 DO 10 I=1,IN
02000 RKA=RKA+A(I)
02010 10 RKB=RKB+B(I)
02020 IF (ABS(RKA).LT.0.0000001) GO TO 20
02030 DCAMP=RKB/RKA
02040 RETURN
02050 20 CONTINUE
02060 DCAMP=10000000.0
02070 RETURN
02080 END
END OF DATA

```

APPENDIX 9

=====

9. Programmel.

I dette appendix er selve programteksten listet for de tre programmer. Af pladshensyn er underprogrammer, der er anvendt i alle tre programmer, listet separat.

- Fig. 1: Hovedprogrammet MV.
- Fig. 2: Hovedprogrammet DB.
- Fig. 3: Hovedprogrammet PZ.
- Fig. 4: Fælles underprogrammer.

---ooo0ooo---

MV

```
00010 //A159270 JOB (***,NEU), 'N*K*P', CLASS=W
00020 /*ROUTE PRINT REMOTE3
00030 $JOB WATFIV, L=0, P=30
00040 C$PRINTOFF
00050 BLOCK DATA
00060 COMMON /RMLDAT/P, THETA, X, Z, EPS1, Y1, RL, RLO, RL1, NRELS, N1, IRHO, NF, NG,
00070 >NH, NCC, QW
00080 COMMON /FILDAT/X1
00090 COMMON /TPARM/TG, TF, TH, E, TGAMMA, NF2, NG2, NH2, TVAR, RKA, RKB
00100 C+++++
00110 COMMON /LGDATA/D1, SUM
00120 C COMMON /HISTD/IDATA, NHIST, R, N1HIST
00130 C+++++
00140 C COMMON /CORDAT/GEM, COR, N, RMID, NSEL, NCOR
00150 C*****
00160 REAL D1(8,50)/400*0.00/, SUM(8)/8*0.00/
00170 C REAL R(4,4)/16*0.0/
00180 C INTEGER IDATA(103,3)/309*0/, NHIST(3)/3*0/, N1HIST(3)/3*0/
00190 C+++++
00200 C REAL GEM(41,4), COR(41,10)/410*0.00/, RMID(4)/4*0.00/
00210 C INTEGER N/0/, NCOR/4/, NSEL(10,2)/1,1,2,2,3,3,4,4,12*0/
00220 C+++++
00230 REAL P(23,23)/529*0.00/, THETA(23)/23*0.00/, X(23)/23*0.00/,
00240 >Z(23)/23*0.00/, X1(6,2,17)/204*0.00/, TG(6)/6*0.00/,
00250 >TF(6)/6*0.00/, TH(6)/6*0.00/, E(6)/6*0.00/, QW(23,23)/529*0.0/
00260 REAL EPS1/0.00/, Y1/0.00/, RL/1.00/, RL1/1.00/, RLO/0.00/,
00270 >TGAMMA/0.00/, RKA/0.00/, RKB/0.00/
00280 REAL TVAR(4)/4*0.00/
00290 INTEGER NRELS/0/, N1(5)/5*0/, NF/1/, NG/1/, NH/1/, NF2/2/, NG2/2/, NH2/2/
00300 END
00310 C
00320 C
00330 REAL XX(6,4)/24*0.00/, A(6)/6*0.00/, B(6)/6*0.00/, C(6)
00340 >/6*0.00/, F(6)/6*0.00/, G(6)/6*0.00/, H(6)/6*0.00/, O(6)
00350 >/6*0.00/, P1(6)/6*0.00/, Q1(6)/6*0.00/, R1(6)/6*0.00/, D/0.00/,
00360 >VAR/0.00/, G0/1.00/, GAMMA/0.00/, Y1/0.00/, U1/0.00/, W1/0.00/, E/0.00/
00370 >, EPS/0.00/, FI/0.00/, ET(6)/1.00, 5*0.00/, ESTYRE/0.00/, U/0.0/
00380 INTEGER NN/-1/, K/-1/, IREGTP/-1/, IG/0/
00390 REAL P2(6)/6*0.00/, Q2(6)/6*0.00/, R2(6)/6*0.00/
00400 REAL GR/10.0/, QE/100000.0/
00410 C
00420 C$PRINON
00430 C
00440 C*****
00450 C*****
00460 C**
00470 C**
00480 C**
00490 C*****
00500 C*****
00510 C
00520 C
00530 CALL INPUT(NN, A, N, B, NB, C, NC, D, K, VAR, P1, NP1, Q1, NQ1, R1, NR1,
00540 > P2, NP2, Q2, NQ2, R2, NR2, IREGTP)
00550 ISTART=1
00560 ISTOP=301
00570 IP=2
00580 IST=0
00590 SPR=SQRT(VAR)
00600 IR=0
```

```
00600      IR=0
00610      IINT=(ISTOP-ISTART)/100
00620      DO 10 I=1,ISTOP
00630          CALL DELAY(K,XX,Y1,U1,W1,U)
00640          W=REF(I,0)
00650 C      CALL STOEJ(I,SPR,25.0,1,0.02,E)
00660 C      IF (I.GT.40) D=AMIN1(D+0.01,1.00)
00670 C      CALL MARKOV(I,5.0,0.01,D)
00680 C      IF (I.GT.40) D=1.00
00690 C      IF (I.EQ.41) E=30.00
00700 C      CALL PERTUB(I,E,VAR,1.5,5.0)
00710      CALL GAUSS(NN,SPR,0.0,E)
00720      Y=FILTER(1,A,N,B,NB,XX(K,2))+FILTER(2,A,N,C,NC,E)
00730      >          +FILTER(3,A,N,ET,0,D)
00740      ESTYRE=Y-W
00750      W1=FILTER(16,R2,NR2,R1,NR1,W)
00760      Y1=FILTER(17,P2,NP2,ET,0,Y)
00770      FI=FILTER(4,ET,0,P1,NP1,Y1)+FILTER(5,ET,0,Q1,NQ1
00780      >          ,XX(K,2))-XX(K,3)
00790      YE1=FI
00800      YE2=XX(K,1)
00810      YE3=XX(K,2)
00820      YE4=XX(K,3)
00830      CALL RML(FI,XX(K,1),XX(K,2),XX(K,3),EPS,
00840      >          EPS2,IR,GR,IGR,QE,CD,DELTA,TRP,PD)
00850      CALL GLR2(I,10,100.0,EPS2,QE,GR,YE1,YE2,YE3,YE4,IST,IR,THRESH)
00860      IG=IG+IGR
00870      IF (IGR.EQ.1) WRITE(6,1020) I
00880      CALL REGPAR(IREGTP,G,NG,F,NF,H,NH,GAMMA)
00890      U1=-FILTER(7,G,NG,F,NF,Y1)-FILTER(8,G,NG,H,NH,W1)
00900      >          -FILTER(9,G,NG,ET,0,GAMMA)
00910      U=FILTER(6,ET,0,Q2,NQ2,U1)
00920      IF (IP.EQ.1) CALL OUT1(I,ISTART-1,IINT,Y,U,W,ESTYRE,FI,
00930      >          EPS,VAR,D,E)
00940      IF (IP.EQ.3) CALL OUT2(I,ISTART-1,IINT,K,TRP,PD)
00950      IF (IP.EQ.2) CALL KONV1(I,ISTART-1,IINT,A,N,B,NB,K,VAR,G,NG,F,NF,
00960      >          P1,NP1,Q1,NQ1,R1,NR1,D,GAMMA,W,THRESH,CD,DELTA)
00970      IF (IP.EQ.4) CALL KONV2(I,ISTART-1,IINT,A,N,B,NB,K,VAR,G,NG,F,NF,
00980      >          ,P1,NP1,P2,NP2,R1,NR1,R2,NR2,D,GAMMA,W,THRESH,CD,
00990      >          DELTA)
01000 10      CONTINUE
01010          WRITE(6,1000) IG
01020          WRITE(6,1010)
01030 1000      FORMAT(1H-, 'ANTAL GRÆNSE TILFÆLDE : ',I4)
01040 1010      FORMAT(1H1)
01050 1020      FORMAT(1H-,T90, ' GRAENSE P$ TRIN : ',I3)
01060          STOP
01070          END
01080 C
01090 C$PRINTOFF
01100 C
01110 C*****
01120 C*****
01130 C**
01140 C**          UNDERPROGRAMMER
01150 C**
01160 C*****
01170 C*****
06210      DO 80 I=1,INH
06220 80      H(I)=THETA(N1(4)-1+I)
06230          GAMMA=THETA(IRHO)
06240          NG=NG1
06250          NF=NF1
06260          NH=NH1
06270 90      CONTINUE
06280          RETURN
06290          END
06300 C
06310 C
06320      SUBROUTINE INPUT(NN,A,N,B,NB,C,NC,D,K,VAR,P1,NP1,Q1,NQ1,R1,NR1,
06330      >          P2,NP2,Q2,NQ2,R2,NR2,IREGTP)
06340 C*****
06350 C
06360 C          INDLAESNING OG
06370 C          INITIALISERING
06380 C
06390 C*****
06400      COMMON /RMLDAT/PP(23,23),THETA(23),X(23),Z(23),EPS1,Y1,RL,RLO,
06410      >          RL1,NRELS,N1(5),IRHO,NF,NG,NH,NCC,QW(23,23)
06420      COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TGAMMA,NF2,NG2,NH2,TVAR(4),
```

```
06430 > RKA,RKB
06440 DIMENSION P2(6),Q2(6),R2(6)
06450 DIMENSION A(6),B(6),C(6),P1(6),Q1(6),R1(6)
06460 REAL EC(6)/6*0.0/,F(6)/6*0.0/,G(6)/6*0.0/,H(6)/6*0.0/
06470 REAL RH1(6)/6*0.0/,RH2(6)/6*0.0/,RH3(6)/6*0.0/
06480 LOGICAL*1 STJERN/'*'/,BLK/' '/
06490 LOGICAL*1 ESTM(6,6)/36*' '/
06500 INTEGER INDEX(6)/0,1,2,3,4,5/
06510 C+++++ INDLAESNING AF SYSTEMSATA ++++++
06520 READ(5,1010)N,(A(I),I=1,N)
06530 READ(5,1010)NB,(B(I),I=1,NB)
06540 READ(5,1010)NC,(C(I),I=1,NC)
06550 N=N-1
06560 NB=NB-1
06570 NC=NC-1
06580 READ(5,1010)K,VAR,D
06590 NN=10317
06600 RKA=0.00
06610 IN=N+1
06620 DO 140 J=1,IN
06630 140 RKA=RKA+A(J)
06640 IN=NB+1
06650 RKB=0.00
06660 DO 150 J=1,IN
06670 150 RKB=RKB+B(J)
06680 C+++++ INDLAESNING AF KOSTFUNKTION ++++++
06690 READ(5,1010) NP1,(P1(I),I=1,NP1)
06700 READ(5,1010)NQ1,(Q1(I),I=1,NQ1)
06710 READ(5,1010)NR1,(R1(I),I=1,NR1)
06720 NP1=NP1-1
06730 NQ1=NQ1-1
06740 NR1=NR1-1
06750 READ(5,1010)NP2,(P2(I),I=1,NP2)
06760 READ(5,1010)NQ2,(Q2(I),I=1,NQ2)
06770 READ(5,1010)NR2,(R2(I),I=1,NR2)
06780 READ(5,1010) IREGTP,QD
06790 NP2=NP2-1
06800 NQ2=NQ2-1
06810 NR2=NR2-1
06820 C+++++ ESTIMATIONSDATA ++++++
06830 RL1=1.00
06840 RL =0.95
06850 READ(5,1010)NRELS,GO
06860 READ(5,1010) NE
06870 READ(5,1010) NBE
06880 READ(5,1010) KE
06890 READ(5,1010) NCE
06900 ALFA =1000.00
06910 ALFAD=1000.00
06920 NF=MAX0(NP2+NE-1,NP1+NCE-1)
06930 NG=MAX0(NQ2+NBE+KE-1,NQ1+NCE)
06940 NH=NCE
06950 NCC=KE-1
06960 C----- EST. AF F - PARAMETERNE -----
06970 N1(2)=1
06980 INF=NF+1
06990 DO 10 I=1,INF
07000 10 ESTM(I,1)=STJERN
07010 C----- EST. AF G - PARAMETERNE -----
07020 N1(3)=NF+2
07030 IF (NG.LT.1) GO TO 25
07040 ING=NG+1
07050 DO 20 I=2,ING
07060 20 ESTM(I,2)=STJERN
07070 C----- EST. AF H - PARAMETERNE -----
07080 25 N1(4)=N1(3)+NG+1
07090 IF (NH.LT.1) GO TO 35
07100 INH=NH+1
07110 DO 30 I=2,INH
07120 30 ESTM(I,3)=STJERN
07130 C----- EST. AF D - PARAMETEREN -----
07140 35 IRHO=N1(4)+NH+1
07150 IF (ALFAD .NE. 0.00) ESTM(1,6)=STJERN
07160 IF (NRELS) 40,50,50
07170 C- - - - - RLS - TILFAELDET - - - - -
07180 40 RLO=0.00
07190 GO TO 75
07200 C- - - - - RELS OG RML TILFAELDET - - - - -
07210 50 RLO=0.99
07220 C----- E - PARAMETERNE -----
```



```
07230      N1(5)=IRHO
07240      IRHO=IRHO+NCC
07250      IF (NCC.EQ.0) GO TO 75
07260      DO 70 I=1,NCC
07270 70    ESTM(I+1,5)=STJERN
07280 C+++++ INITIALISERING AF THETA OG P ++++++
07290 75    THETA(N1(3))=G0
07300      THETA(N1(4))=-1.00
07310      EC(1)=1.00
07320      F(1)=THETA(N1(2))
07330      G(1)=THETA(N1(3))
07340      H(1)=THETA(N1(4))
07350      DO 80 I=1,IRHO
07360 80    PP(I,I)=ALFA
07370      PP(IRHO,IRHO)=ALFAD
07380      PP(N1(3),N1(3))=0.00
07390      PP(N1(4),N1(4))=0.00
07400      QW(IRHO,IRHO)=QD
07410      X(IRHO)=1.00
07420 C+++++ KONTOL UDSKRIFT ++++++
07430      WRITE(6,502)
07440      WRITE(6,503)
07450      WRITE(6,504) (INDEX(I),A(I),B(I),C(I),INDEX(I),F(I),ESTM(I,1),
07460 >G(I),ESTM(I,2),H(I),ESTM(I,3),I=1,6)
07470      WRITE(6,505) N,K,VAR,D,THETA(IRHO),ESTM(1,6),NE,QD
07480      TVAR(1)=SALOSS(A,C,N)*VAR
07490      WRITE(6,506) TVAR(1),NN,ALFA,NBE
07500      WRITE(6,507) ALFAD,NCE,NRELS,KE
07510      WRITE(6,508) RL1,RLO,RL
07520      WRITE(6,509)
07530 C----- BEREKNING AF REG. PARAMETER -----
07540      CALL MULT(P1,NP1,C,NC,RH1,NH1)
07550 C          RH1=P1*C
07560      CALL MULT(A,N,P2,NP2,RH2,NH2)
07570 C          RH2=A*P2
07580      CALL PREDIK(RH2,NH2,RH1,NH1,K,E,TF,NF2)
07590 C
07600      CALL MULT(Q2,NQ2,B,NB,RH1,NH1)
07610 C          RH1=Q2*B
07620      CALL MULT(RH1,NH1,E,K-1,RH2,NH2)
07630 C          RH2=Q2*B*E
07640      CALL MULT(Q1,NQ1,C,NC,RH3,NH3)
07650 C          RH3=Q1*C
07660      CALL ADD(RH2,NH2,RH3,NH3,TG,NG2)
07670 C          TG=Q2*B*E+Q1*C
07680      INH=NC+1
07690      DO 90 I=1,INH
07700 90    TH(I)=-C(I)
07710      TGAMMA=E(1)
07720      DO 100 I=2,K
07730 100   TGAMMA=TGAMMA+E(I)
07740      TGAMMA=TGAMMA*D
07750      WRITE(6,5010) (INDEX(I),TF(I),TG(I),TH(I),INDEX(I),EC(I),ESTM(I,4)
07760 >,I=1,6)
07770 C----- BEREKNING AF VARIANSERNE -----
07780      CALL MULT(RH1,NH1,P1,NP1,RH2,NH2)
07790 C          RH2=Q2*B*P1
07800      CALL MULT(Q1,NQ1,A,N,RH1,NH1)
07810 C          RH1=Q1*A
07820      CALL MULT(RH1,NH1,P2,NP2,RH3,NH3)
07830 C          RH3=Q1*A*P2
07840      CALL ADD(RH3,NH3,RH2,NH2,RH1,NH1)
07850 C          RH1=Q1*A*P2+Q2*B*P1
07860      CALL MULT(TG,NG2,P2,NP2,RH3,NH3)
07870 C          RH3=TG*P2
07880      TVAR(1)=SALOSS(RH1,RH3,MAXO(NH3,NH1))*VAR
07890      CALL MULT(TF,NF2,Q2,NQ2,RH3,NH3)
07900 C          RH3=TF*Q2
07910      TVAR(2)=SALOSS(RH1,RH3,MAXO(NH1,NH3))*VAR
07920      DO 110 J=1,K
07930 110   TVAR(4)=TVAR(4)+E(J)**2
07940      TVAR(4)=TVAR(4)*VAR
07950      TVAR(3)=VAR
07960      IF (NRELS.LT.0) TVAR(3)=TVAR(4)
07970      DO 120 J=1,4
07980 120   IF (TVAR(J).LT.0.00001) TVAR(J)=1.00
07990      CONTINUE
08000      WRITE(6,5013) TVAR(1),TVAR(2),TGAMMA
08010      WRITE(6,5015) IREGTP
08020      WRITE(6,5011)
```

```
08030 ESTM(1,6)=BLK
08040 WRITE(6,504) (INDEX(I),P1(I),Q1(I),R1(I),INDEX(I),P2(I),BLK
08050 > Q2(I),BLK,R2(I),BLK,I=1,6)
08060 WRITE(6,5016) TVAR(4),TVAR(3)
08070 C+++++ FORMAT SAETNINGERNE +++++
08080 1010 FORMAT(I3,2X,10(F8.4,3X))
08090 C
08100 C
08110 502 FORMAT(1H1,2(117(1H*)/1H ),'***',T117,'**'/1H ,'**',T45,
08120 >'SELVINDSTILLENDEN MINIMALVARIANS REGULATOR',T117,'**'/1H ,
08130 >'***',T117,'**'/1H ,2(117(1H*)/1H ),'**',T60,'**',T118,'**'/1H ,
08140 >'**',T25,'OBJEKT BESKRIVELSEN',T60,'**',T80,'ESTIMATIONS DATA',T118,
08150 >'**'/1H ,'**',T60,'**',T118,'**'/1H ,117(1H*)/1H ,'**',T60,'**',
08160 >T118,'**',/)
08170 C
08180 503 FORMAT(1H+,'**',T14,'N',T22,'A',T33,'B',T44,'C',T60,'**',T71,'N',
08190 >T79,'F',T91,'G',T103,'H',T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08200 C
08210 504 FORMAT(1H+,6('**',T13,I2,3(3X,F8.4),T60,'**',T70,I2,3X,3(F8.4,1X,
08220 >1A1,2X),T118,'**'/1H ,'**',T60,'**',T118,'**'/1H ))
08230 C
08240 505 FORMAT(1H+,'**',5X,'N = ',I2,3X,'K = ',I2,3X,'VAR = ',F8.4,3X,
08250 >'D = ',F8.4,T60,'**',T65,'GAMMA = ',F8.4,1X,A1,2X,'N = ',I2,
08260 >2X,'QD = ',F8.4, T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08270 C
08280 506 FORMAT(1H+,'**',T8,'VAR(Y) = ',F8.4,7X,'NN = ',I6,
08290 > T60,'**',T65,'ALFA = ',F8.1,
08300 >4X,'NB = ',I2,T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08310 C
08320 507 FORMAT(1H+,59(1H*),T65,'ALFAD = ',F8.1,4X,'NC = ',I2,T118,'**'/1H ,
08330 >'**',T60,'**',T118,'**'/1H ,'**',T20,'KORREKTE PARAMETER',T60,'**',T65,
08340 >'NRELS = ',I2,10X,'K = ',I2,T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08350 C
08360 508 FORMAT(1H+,59(1H*),T63,' L1 = ',F8.4,4X,'LO = ',F8.4,
08370 >2X,'L = ',F8.4,T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08380 C
08390 509 FORMAT(1H+,'**',T14,'N',T22,'F',T33,'G',T44,'H',T60,'**',T71,'N',T79
08400 >,'E',T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08410 C
08420 5010 FORMAT(1H+,6('**',T13,I2,3(3X,F8.4),T60,'**',T70,I2,3X,
08430 >1(F8.4,1X,1A1,2X),T118,'**'/1H ,'**',T60,'**',T118,'**'/1H ))
08440 C
08450 5011 FORMAT(1H+,117(1H*)/1H ,'**',T60,'**',T118,'**'/1H ,'**',T25,
08460 >'KOSTFUNKTIONEN',T60,'**',T118,'**'/1H ,'**',T60,'**',
08470 >T118,'**'/1H ,117(1H*)/1H ,'**',T60,'**',T118,'**'/1H ,'**',T14,
08480 >'N',T22,'P1',T33,'Q1',T44,'R1',T60,'**',T71,'N',T79,'P2',T91,'Q2',
08490 >T103,'R2',T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08500 C
08510 5013 FORMAT(1H+,'**',2X,'VARY = ',F8.4,3X,'VARU = ',F8.4,3X,'GAMMA = ',
08520 >F8.4,T60,'**',T118,'**'/1H ,'**',T60,'**',T118,'**',/)
08530 C
08540 C
08550 5015 FORMAT(1H+,'**',18X,'IREGTP = ',I3,T60,'**',T118,'**'/1H ,
08560 > T60,'**',T118,'**',/)
08570 C
08580 5016 FORMAT(1H+,'**',16X,'VAR(FI) = ',F8.4,T60,'**',3X,'VAR(EPS) = ',F8.4,
08590 > T118,'**'/1H ,'**',T60,'**',T118,'**'/1H ,117(1H*)/1H ,
08600 RETURN
08610 END
08620 C
08630 C
08640 C
08650 SUBROUTINE OUT1(I,IO,I1,Y,U,W,ESTYRE,FI,EPS,VAR,D,ESTOEJ)
08660 C*****
08670 C
08680 C PROCEDURE TIL OPSAMLING, BEHANDLING OG UDSKRIVNING AF SIGNAL
08690 C
08700 C*****
08710 COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TGAM,NG2,NF2,NH2,TVAR(4),RKA,
08720 > RKB
08730 REAL SIGNAL(101,3),VAR1(101,2),VAR2(101,2),TAB(101,4),
08740 > RMID(101,3)
08750 REAL RKVAD1/0.0/,RKVAD2/0.0/,RKVAD3/0.0/,RKVAD4/0.0/
08760 REAL O2(2)/2*0.00/
08770 REAL O3(3)/3*0.00/
08780 REAL O4(4)/4*0.00/
08790 IF ((I.LT.IO+1).OR.(I.GT.IO+100*I1+1)) RETURN
08800 IF (I.EQ.IO+100*I1+1) GO TO 10
08810 C----- DATA - BEHANDLING -----
08820 U1=U-(RKA*W-D)/RKB
```



```

096  ^      DO 110 J=1,IN
09640 110  HUD(I2,J)=THETA(N1(4)+J-1)
09650      GUD(I2,1)=THETA(N1(3))
09660      IF (NRELS) 90,70,70
09670 70    IN=NCC+1
09680      DO 80 J=2,IN
09690 80    EUD(I2,J)=THETA(N1(5)+J-2)
09700      EUD(I2,1)=1.00
09710 90    GAMUD(I2)=THETA(IRHO)
09720      TRACE(I2,1)=ALOG10(TRP)
09730      TRACE(I2,2)=ALOG10(PD)
09740      IF (I2.NE.101) RETURN
09750 C-    - - - - - UDSKRIFT - - - - -
09760      CALL IPLT(NF+1,6,IO,I1,0,2.0,-2.0,FUD,TF,
09770 >      'ESTIMATERNE AF FO, F1,...')
09780      IF (NG.NE.0) CALL IPLT(NG+1,6,IO,I1,0,2.0,-2.0,GUD,TG,
09790 >      'ESTIMATERNE AF GO, G1,...')
09800      IF (NH.NE.0) CALL IPLT(NH+1,6,IO,I1,0,2.0,-2.0,HUD,TH,
09810 >      'ESTIMATERNE AF HO, H1,...')
09820      IF ((NCC.GT.0).AND.(NRELS.GT.-1)) CALL IPLT(NCC,6,IO,I1,0,
09830 >      2.0,-2.0,EUD,E,'ESTIMATERNE AF EO, E1,')
09840      CALL IPLT(1,1,IO,I1,1,2.0,-2.0,GAMUD,TGAMMA,
09850 >      'ESTIMATET AF GAMMA')
09860      CALL IPLT(2,2,IO,I1,1,5.0,-5.0,TRACE,TF,
09870 >      'TR(P) OG P(RHO,RHO) I LOG')
09880      RETURN
09890      END
09900 C
09910 C
09920 C*****
09930      SUBROUTINE KONV1(IT,IO,I1,A,NA,B,NB,K,VAR,G,NG,F,NF,P,NP,
09940 >      Q,NQ,R,NR,D,GAMMA,W,THRESH,CD,DELTA)
09950 C*****
09960 C
09970 C      PROCEDURE TIL TIL KONVERGENSANALYSE FOR DET TILFAELDE AT :
09980 C
09990 C          P2 = Q2 = R2 = 1  OG  C = 1
10000 C
10010 C*****
10020      COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TGAMMA,NF2,NG2,NH2,TVAR(4),
10030 >      RKA,RKB
10040      DIMENSION A(6),B(6),G(6),F(6),P(6),Q(6),R(6)
10050      REAL RN(6),RN1(6),RN2(6)/6*0.0/,RN3(6)
10060      REAL RT(6),RT1(6),RT2(6)
10070      INTEGER L/O/
10080      REAL GRAENS(101,3)
10090      REAL O4(4)/4*0.0/,O3(3)/3*0.0/
10100      REAL ACAMP(101,4)
10110      REAL DCAMP(101,3)
10120      IF ((IT.LT.IO+1).OR.(IT.GT.IO+100*I1+1)) RETURN
10130      IF (MOD(IT-IO-1,I1).NE.0) RETURN
10140      I2=(IT-IO-1)/I1+1
10150 C+++++ INITIALISERING ++++++
10160      IF (L.NE.0) GO TO 10
10170      L=11
10180      VAR1=VAR
10190      IF (VAR.EQ.0.0) VAR1=1.00
10200      CALL MULT(B,NB,R,NR,RT1,NRT1)
10210 C          ** RT1=B*R **
10220      CALL MULT(A,NA,R,NR,RT2,NRT2)
10230 C          ** RT2=A*R **
10240      CALL MULT(B,NB,P,NP,RN1,NRN1)
10250 C          ** RN1=B*P **
10260      CALL MULT(A,NA,Q,NQ,RN2,NRN2)
10270 C          ** RN2=A*Q **
10280      CALL ADD(RN1,NRN1,RN2,NRN2,RN,NRN)
10290 C          ** RN=B*P+A*Q **
10300      VARYW=SALOSS(RN,RT1,MAXO(NRN,NRT1))
10310      VARUW=SALOSS(RN,RT2,MAXO(NRN,NRT2))
10320      RK1=0.0
10330      RK2=0.0
10340      DO 20 I=1,6
10350      RK1=RK1+RT1(I)
10360 C          ** RK1=DC(B*R) **
10370 20    CONTINUE
10380      IN=NB+1
10390      DO 30 I=1,IN
10400 30    RN2(I+K)=B(I)
10410      DO 50 I=1,K
10420 50    RN2(I)=0.0

10430      RN2=NB+K
10440 C          ** RN2=B/Q**K **
10450 10    CONTINUE
10460 C+++++!
QED
1 10420
10420 50    RN2(I)=0.0
/99999

```

```

10420 50   RN2(I)=0.0
10430     NRN2=NB+K
10440 C
10450 10   CONTINUE                               RN2=B/Q**K **
10460 C+++++ DATA BEHANDLING ++++++
10470     CALL MULT(A,NA,G,NG,RN1,NRN1)
10480 C
10490     CALL MULT(RN2,NRN2,F,NF,RN3,NRN3)      ** RN1=A*G **
10500 C
10510     CALL ADD(RN1,NRN1,RN3,NRN3,RN,NRN)    -* RN3=B*F/Q**K **
10520 C
10530     ACAMP(I2,1)=SALOSS(RN,RT1,MAXO(NRN,NRT1))/VARYW
10540     ACAMP(I2,2)=SALOSS(RN,RT2,MAXO(NRN,NRT2))/VARUW
10550     ACAMP(I2,3)=SALOSS(RN,G,MAXO(NRN,NG))*VAR1/TVAR(1)
10560     ACAMP(I2,4)=SALOSS(RN,F,MAXO(NRN,NF))*VAR1/TVAR(2)
10570 C----- THRESHOLD GRÆNSER -----
10580     GRAENS(I2,1)=THRESH
10590     GRAENS(I2,2)=CD
10600     GRAENS(I2,3)=DELTA
10610     RK=0.0
10620     RK2=0.0
10630     DO 40 I=1,6
10640     RK=RK+RN(I)
10650     RK2=RK2+G(I)
10660 40   CONTINUE
10670     DCAMP(I2,1)=RK1/RK-1.00
10680     DCAMP(I2,2)=(RK2*D-RKB*GAMMA)/RK
10690     DCAMP(I2,3)=DCAMP(I2,1)*W+DCAMP(I2,2)
10700     IF (I2.NE.101) RETURN
10710 C+++++
10720     CALL IPLT(4,4,IO,I1,1,2.0,0.0,ACAMP,04,
10730     >      'AC.Y/W, U/W, Y/E, U/E      ')
10740     CALL IPLT(3,3,IO,I1,1,0.5,-0.5,DCAMP,03,
10750     >      'FEJL FRA Y/W,Y(D,UO)      ')
10760     CALL IPLT(3,3,IO,I1,1,5.0,0.0,GRAENS,04,
10770     >      'THRESH, CD OG DELTA      ')
10780     RETURN
10790     END
10800 C
10810 C
10820 C*****
10830     SUBROUTINE KONV2(IT,IO,I1,A,NA,B,NB,K,VAR,G,NG,F,NF,P1,NP1,P2,NP2,
10840     >      R1,NR1,R2,NR2,D,GAMMA,W,THRESH,CD,DELTA)
10850 C*****
10860 C
10870 C     PROCEDURE TIL TIL KONVERGENSANALYSE FOR DET TILFAELDE AT :
10880 C
10890 C           Q=0  OG  C=1
10900 C
10910 $PRINTON
10920 C*****
10930     COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TGAMMA,NF2,NG2,NH2,TVAR(4),
10940     >      RKA,RKB
10950     DIMENSION A(6),B(6),G(6),F(6),P1(6),P2(6),R1(6),R2(6)
10960     REAL RN(6),RN1(6),RN2(6)/6*0.0/,RN3(6),RN4(6)
10970     REAL RT(6),RT1(6),RT2(6),RT3(6)
10980     INTEGER L/O/
10990     REAL GRAENS(101,3)
11000     REAL O4(4)/4*0.0/,O3(3)/3*0.0/
11010     REAL ACAMP(101,4)
11020     REAL DCAMP(101,3)
11030     IF ((IT.LT.IO+1).OR.(IT.GT.IO+100*I1+1)) RETURN
11040     IF (MOD(IT-IO-1,I1).NE.0) RETURN
11050     I2=(IT-IO-1)/I1+1
11060     IF (L.NE.0) GO TO 10
11070     L=11
11080     VAR1=VAR
11090     IF (VAR.EQ.0.0) VAR1=1.00
11100     CALL MULT(R1,NR1,P2,NP2,RT,NRT)
11110 C
11120     CALL MULT(P1,NP1,R2,NR2,RN,NRN)          ** RT=R1*P2 **
11130 C
11140     VARYW=SALOSS(RN,RT,MAXO(NRN,NRT))
11150     LK=MAXO(NRN,NRT)
11160     CALL MULT(B,NB,RN,NRN,RN1,NRN1)
11170 C
11180     CALL MULT(A,NA,RT,NRT,RT1,NRT1)         ** RN1=B*P1*R2 **
11190 C
11200     VARUW=SALOSS(RN1,RT1,MAXO(NRN1,NRN1))
11210     CALL MULT(B,NB,RT,NRT,RT1,NRT1)         ** RT1=A*P2*R1 **

```

```

11220 C          RT1=B*P2*R1 **
11230          CALL MULT(A,NA,RT,NRT,RT2,NRT2)
11240 C          -- RT2=A*P2*R1 **
11250          CALL MULT(A,NA,P2,NP2,RN1,NRN1)
11260 C          ** RN1=A*P2 **
11270          RK1=0.0
11280          RK2=0.0
11290          RK4=0.0
11300          DO 20 I=1,6
11310          RK1=RK1+RT1(I)
11320 C          ** RK1=DC(B*P2*R1) **
11330          RK2=RK2+R2(I)
11340 C          ** RK2=DC(R2) **
11350          RK4=RK4+R1(I)
11360 20          CONTINUE
11370          RK3=RK1/RK4
11380 C          ** RK3=DC(B*P2) **
11390          IN=NB+1
11400          DO 30 I=1,IN
11410 30          RN2(I+K)=B(I)
11420          NRN2=NB+K
11430 C          ** RN2=B/Q**K **
11440 10          CONTINUE
11450 C+++++
11460          CALL MULT(F,NF,RN2,NRN2,RN3,NRN3)
11470 C          ** RN3=B*F/Q**K **
11480          CALL MULT(G,NG,RN1,NRN1,RN4,NRN4)
11490 C          ** RN4=A*G*P2 **
11500          CALL ADD(RN3,NRN3,RN4,NRN4,RN,NRN)
11510 C          ** RN=A*G*P2+B*F/Q**K **
11520          ACAMP(I2,4)=SALOSS(RN,F,MAXO(NRN,NF))*VAR1/TVAR(2)
11530          CALL MULT(G,NG,P2,NP2,RT,NRT)
11540 C          ** RT=G*P2 **
11550          R5=0.0
11560          DO 50 I=1,6
11570 50          R5=R5+RT(I)
11580          ACAMP(I2,3)=SALOSS(RN,RT,MAXO(NRN,NRT))*VAR1/TVAR(1)
11590          CALL MULT(RN,NRN,R2,NR2,RN3,NRN3)
11600 C          ** RN3=RN*R2 **
11610          ACAMP(I2,1)=SALOSS(RN3,RT1,MAXO(NRN3,NRT1))/VARYW
11620          ACAMP(I2,2)=SALOSS(RN3,RT2,MAXO(NRN3,NRT2))/VARUW
11630          R4=0.0
11640          DO 40 I=1,6
11650 40          R4=R4+RN(I)
11660          DCAMP(I2,1)=RK1/(R4*RK2)-1.00
11670          DCAMP(I2,2)=(R5*D-RK3*GAMMA)/R4
11680          DCAMP(I2,3)=DCAMP(I2,1)*W+DCAMP(I2,2)
11690 C----- THRESHOLD GRENSER -----
11700          GRAENS(I2,1)=THRESH
11710          GRAENS(I2,2)=CD
11720          GRAENS(I2,3)=DELTA
11730          IF (I2.NE.101) RETURN
11740 C+++++
11750          CALL IPLT(4,4,IO,I1,1,2.0,0.0,ACAMP,04,
11760          >          'AC.Y/W, U/W, Y/E, U/E ')
11770          CALL IPLT(3,3,IO,I1,1,0.5,-0.5,DCAMP,03,
11780          >          'FEJL FRA Y/W, Y(D,UO) ')
11790          CALL IPLT(3,3,IO,I1,1,5.0,0.0,GRAENS,04,
11800          >          'THRESH, CD OG DELTA ')
11810          RETURN
11820          END
14480 C
14490 $ENTRY
14500 3 1.0000 -01.5000 00.7000 -00.9277 (NA+1),A0,A1,
14510 2 01.0000 00.5000 00.1892 (NB+1),B0,B1,...
14520 1 1.0000 -00.9000 (NC+1),C0,C1,...
14530 1 0.1000 0.0000 K,VAR,D
14540 3 1.0000 -0.3980 0.1353 (NP1+1),P10,P11,...
14550 1 0.0000 -00.2500 (NQ10+10),Q100,Q1010,
14560 1 0.5820 00.1555 (NR1+1),R10,R11,...
14570 2 1.0000 0.2670 (NP2+1),P20,P21,...
14580 1 1.0000 (NQ2+1),Q20,Q21,...
14590 1 1.0000 -0.3980 0.1353 (NR2+1),R20,R21,...
14600 5 0.0000 IREGTP,QD
14610 -5 1.0000 NRELS,G0
14620 2 NE
14630 1 NBE
14640 1 KE
14650 0 NCE
END OF DATA

```

DB

```

00010 //A159270 JOB (***,NEU), 'N*K*P', CLASS=W
00020 /*ROUTE PRINT REMOTE3
00030 $JOB WATFIV, L=0, P=30, NOLIST
00040 BLOCK DATA
00050 COMMON /RMLDAT/P, THETA, X, Z, EPS1, Y1, RL, RLO, RL1, QW, QE, T, NRELS, N1,
00060 >IRHO, N, M, K, NC
00070 COMMON /FILDAT/X1
00080 COMMON /TPARM/TQ, TP, NQ1, NP1, TUO, TVAR, RKA, RKB
00090 C+++++
00100 COMMON /HISTD/IDATA, NHIST, R, N1HIST
00110 COMMON /UDDAT1/SIGNAL, VAR1, VAR2, TAB,
00120 >
KVAD1, KVAD2, KVAD3
00130 C+++++
00140 COMMON /CORDAT/GEM, COR, N2, RMID, NSEL, NCOR
00150 C*****
00160 REAL SIGNAL(101,3), VAR2(101), VAR1(101,2), TAB(101,3),
00170 >KVAD1/0.00/, KVAD2/0.00/, KVAD3/0.00/, R(4,4)/16*0.00/
00180 INTEGER IDATA(103,3)/309*0/, NHIST(3)/3*0/, N1HIST(3)/3*0/
00190 C+++++
00200 REAL GEM(41,4), COR(41,10)/410*0.0/, RMID(4)/4*0.00/
00210 INTEGER N2/0/, NCOR/4/, NSEL(10,2)/1,1,2,2,3,3,4,4,12*0/
00220 C+++++
00230 REAL P(23,23)/529*0.00/, THETA(23)/23*0.00/, X(23)/23*0.00/,
00240 >Z(23)/23*0.00/, X1(6,2,9)/108*0.00/, TQ(6)/6*0.00/,
00250 >TP(6)/6*0.00/
00260 REAL QW(23,23)/529*0.00/, QE/0.00/, T/1.00/
00270 REAL EPS1/0.00/, Y1/0.00/, RL/1.00/, RL1/1.00/, RLO/0.00/,
00280 >TUO/0.00/
00290 REAL TVAR(3)/3*0.00/, RKA/0.00/, RKB/0.00/
00300 C REAL X2(6)/6*0.00/, DEST/0.00/
00310 INTEGER NRELS/0/, N1(3)/3*0/, NQ1/1/, NP1/1/
00320 INTEGER IRHO/0/, N/0/, K/1/, M/0/, NC/0/
00330 END
00340 C
00350 C*****
00360 C
00370 REAL XX(6,2)/12*0.00/, A(6)/6*0.00/, B(6)/6*0.00/, C(6)
00380 >/6*0.00/, O(6)/6*0.00/
00390 >, P(6)/6*0.00/, Q(6)/6*0.00/, D/0.00/,
00400 >VAR/0.00/, UO/0.00/, Y/0.00/, U/0.00/, W/0.00/, E/0.00/
00410 >, EPS/0.00/, ET(6)/1.00, 5*0.00/, ESTYRE/0.00/
00420 >, BETA/0.00/, YID/0.00/, UID/0.00/
00430 INTEGER NN/-1/, K/-1/, IREGTP/-1/, ITYPE/0/, IG/0/
00440 REAL TRP/1000.0/, PD/1000.0/, QE/1000.0/, GR/10.0/
00450 C
00460 C
00470 C$PRINTON
00480 C*****
00490 C*****
00500 C**
00510 C** HOVEDPROGRAM **
00520 C**
00530 C*****
00540 C*****
00550 C
00560 C
00570 ISTOP=501
00580 IP=1
00590 ISTART=1
00600 IINT=(ISTOP-ISTART)/100

```

```
00600      IINT=(ISTOP-ISTART)/100
00610      CALL INPUT(NN,A,N,B,NB,C,NC,D,K,VAR,Q1,IREGTP)
00620      SPR=SQRT(VAR)
00621      IR=1
00622      IST=0
00630      DO 10 I=1,ISTOP
00640          CALL DELAY(K,XX,Y,U)
00650          W=REF(I,25,0)
00660          CALL STOEJ(I,SPR,25.0,1,0.02,E)
00670 C      CALL MARKOV(I,5.0,0.01,D)
00680 C      CALL PERTUB(I,E,VAR,1.5,5.0)
00690 C      IF (I.GT.40) D=1.00
00700 C      IF (I.GT.40) D=AMIN1(D+0.01,1.00)
00710 C      CALL GAUSS(NN,SPR,0.0,E)
00720 C      IF (I.EQ.7) E=30.00
00730      Y=FILTER(1,A,N,B,NB,XX(K,2))+FILTER(2,A,N,C,NC,E)
00740      >      +FILTER(3,A,N,ET,0,D)
00750      ESTYRE=W-Y
00770      IF (IREGTP.GE.0) CALL RML(Y,XX(K,2),EPS,EPS2,IR,GR,IGR,QE,CD,CE
00780      >      TRP,PD)
00790      IF (IGR.EQ.1) WRITE(6,1020) I
00810      CALL GLR2(I,10,100.0,EPS2,QE,GR,Y,XX(K,2),IST,IR,THRESH)
00820      CALL REGPAR(Q1,Q,NQ,P,NP,IREGTP)
00830      U=FILTER(8,P,NP,Q,NQ,ESTYRE)
00840      IF (IP.EQ.1) CALL KONV(I,ISTART-1,IINT,P,NP,Q,NQ,A,N,B,NB,C,NC,
00850      >      K,VAR,CD,CE,THRESH)
00860      IF (IP.EQ.2) CALL OUT1(I,ISTART-1,IINT,Y,U,W,ESTYRE,EPS,VAR,D)
00870      IF (IP.EQ.3) CALL OUT2(I,ISTART-1,IINT,A,B,C,D,TRP,PD)
00880      IG=IG+IGR
00890 10      CONTINUE
00900      WRITE(6,1010) IG
00910      WRITE(6,1000)
00920 1000      FORMAT(1H1)
00930 1010      FORMAT(1H-, 'ANTAL GRÆNSETILFÆLDE : ',I3)
00940 1020      FORMAT(1H-,T90,'GRÆNSETILFÆLDE P$ TRIN : ',I3)
00950      STOP
00960      END
00970 C
00980 C$PRINTOFF
00990 C
01000 C*****
01010 C*****
01020 C**
01030 C**
01040 C**
01050 C*****
01060 C*****
05610 C
05620 C*****
05630      SUBROUTINE REGPAR(Q1,Q,NQ,P,NP,IREGTP)
05640 C*****
05650 C
05660 C      PROCEDURE TIL UDREGNING AF REGULATORPARAMETERNE
05670 C
05680 C*****
05690 C
05700 C      VARIABELLISTE:
05710 C      =====
05720 C      Q1      REGULATORPARAMETEREN Q(0) I %
05730 C      Q      ARRAY (6) INDH. POLYNOM. Q = Q0*A(1-ALFA/Z)
05740 C      NQ      ORDEN AF Q
05750 C      P      ARRAY (6) INDH. POLYNOM. 1 - P/Z**K
05760 C      NP      ORDEN AF OVENSTAAENDE POLYNOM.
05770 C      UO      STATIONAER STYRESIGNAL TIL UDKOMPENSERING AF D
05780 C      ITYPE   STYREVARIABEL
05790 C      HVIS NUL UDLAESER PARAMETERNE FOR EN DB(V) REG.
05800 C      ELLERS  UDLAESER PARAMETERNE FOR EN DB(V+1) REG.
05810 C      IREGTP  STYREVARIABEL.
05820 C      HVIS IKKE NEG. UDLAESER ESTIMEREDE VAERDIER.
05830 C      HVIS NEG. UDLAESER KORREKTE VAERDIRER.
05840 C
05850 C*****
05860      COMMON /RMLDAT/PP(23,23),THETA(23),X(23),Z(23),EPS1,Y1,RL,RL0,
05870      >      RL1,QW(23,23),QE,T,NRELS,N1(3),IRHO,N,M,K,NC
05880      COMMON /TPARM/TQ(6),TP(6),NQ1,NP1,TUO,TVAR(3),RKA,RKB
05890      DIMENSION P(6),Q(6)
05900      P(1)=1.00
05910      IF (IREGTP) 90,100,100
05920 100      RKB=0.00
05930      J1=K+1
```



```
05940      IN=N+M+1
05950      IN2=N1(2)
05960      DO 10 J=IN2,IN
05970      RKB=RKB+THETA(J)
05980      P(J1)=THETA(J)
05990 10    J1=J1+1
06000      Q0=1.0/((1.0-Q1*THETA(N1(1))/100.0)*RKB)
06010      NQ=N
06020      DO 50 J=2,IN2
06030 50    Q(J)=Q0*THETA(J-1)
06040      Q(1)=Q0
06050      NP=M+K
06060      INP=NP+1
06070      IK=K+1
06080      DO 60 J=IK,INP
06090 60    P(J)=-Q0*P(J)
06100      IF (Q1.EQ.0.0) RETURN
06110 C----- KUN FOR DB(V+1) -----
06120      ALFA=1.00-1.00/(Q0*RKB)
06130      NQ=NQ+1
06140      Q(NQ+1)=-ALFA*Q(NQ)
06150      IN=NQ-1
06160      DO 70 J=1,IN
06170 70    Q(NQ+1-J)=Q(NQ+1-J)-ALFA*Q(NQ-J)
06180      NP=NP+1
06190      P(NP+1)=-ALFA*P(NP)
06200      IN=NP-1-K
06210      IF (IN.LT.1) RETURN
06220      DO 80 J=1,IN
06230 80    P(NP+1-J)=P(NP+1-J)-ALFA*P(NP-J)
06240      RETURN
06250 C+++++ KOPIERING AF KORREKTE PARAMETER ++++++
06260 90    CONTINUE
06270      DO 110 J=1,6
06280      Q(J)=TQ(J)
06290 110   P(J)=TP(J)
06300      NP=NP1
06310      NQ=NQ1
06320      UO=TUO
06330      RETURN
06340      END
06350 C
06360 C
06370 C*****
06380      SUBROUTINE INPUT(NN,A,N,B,NB,C,NC,D,K,VAR,Q1,IREGTP)
06390 C*****
06400 C
06410 C
06420 C
06430 C
06440 C*****
06450      COMMON /RMLDAT/PP(23,23),THETA(23),X(23),Z(23),EPS1,Y1,RL,RLO,
06460      \RL1,QW(23,23),QE,T,NRELS,N1(3),IRHO,NE,NBE,KE,NCE
06470      COMMON /TPARM/Q(6),P(6),NQ1,NP1,TUO,TVAR(3),RKA,RKB
06480      COMMON /FILDAT/XFILT(6,2,9)
06490      DIMENSION A(6),B(6),C(6)
06500      REAL EC(6)/6*0.0/,EA(6)/6*0.0/,EB(6)/6*0.0/
06510      REAL RHJ(6)/6*0.00/
06520      REAL ET(6)/1.00,5*0.00/
06530      LOGICAL*1 STJERN/'*'/
06540      LOGICAL*1 ESTM(6,4)/24*' '/
06550      INTEGER INDEX(6)/0,1,2,3,4,5/
06560 C+++++ INDLAESNING AF SYSTEMSATA ++++++
06570      READ(5,1010)N,(A(I),I=1,N)
06580      READ(5,1010)NB,(B(I),I=1,NB)
06590      READ(5,1010)NC,(C(I),I=1,NC)
06600      N=N-1
06610      NB=NB-1
06620      NC=NC-1
06630      READ(5,1010)K,VAR,D
06640      NN=10317
06650 C
06660      NN=7913
06660 C----- DIMENSIONS KRITERIET -----
06670      READ(5,1020) IREGTP,Q1
06680      ITYPE=0
06690      IF (Q1.NE.0.00) ITYPE=1
06700 C+++++ ESTIMATIONSATA ++++++
06710      RL1=1.00
06720      RL =0.95
06730      READ(5,1010)NRELS,B0,QD
```

```
06740 READ(5,1010) NE
06750 READ(5,1010) NBE
06760 READ(5,1010) KE
06770 READ(5,1010) NCE
06780 ALFA =1000.00
06790 ALFAD=1000.0
06800 C----- EST. AF A - PARAMETERNE -----
06810 IF (NE.LT.1) GO TO 15
06820 N1(1)=1
06830 DO 10 I=1,NE
06840 10 ESTM(I+1,1)=STJERN
06850 C----- EST. AF B - PARAMETERNE -----
06860 15 N1(2)=NE+1
06870 IN=NB+1
06880 DO 20 I=1,IN
06890 20 ESTM(I,2)=STJERN
06900 C----- EST. AF D - PARAMETEREN -----
06910 25 IRHO=NE+NBE+2
06920 IF (ALFAD+QD .NE. 0.00) ESTM(1,4)=STJERN
06930 IF (NRELS) 40,50,50
06940 C----- RLS - TILFAELDET -----
06950 40 RLO=0.00
06960 GO TO 75
06970 C----- RELS OG RML TILFAELDET -----
06980 50 RLO=0.99
06990 C----- C - PARAMETERNE -----
07000 N1(3)=IRHO
07010 IRHO=IRHO+NCE
07020 IF (NCC.EQ.0) GO TO 75
07030 DO 70 I=1,NCE
07040 70 ESTM(I+1,3)=STJERN
07050 C+++++ INITIALISERING AF THETA OG P ++++++
07060 75 THETA(N1(2))=B0
07070 EC(1)=1.00
07080 EB(1)=THETA(N1(2))
07090 EA(1)=1.00
07100 DO 80 I=1,IRHO
07110 80 PP(I,I)=ALFA
07120 PP(IRHO,IRHO)=ALFAD
07130 C PP(N1(2),N1(2))=0.00
07140 QW(IRHO,IRHO)=QD
07150 X(IRHO)=1.00
07160 C+++++ KONTOL UDSKRIFT ++++++
07170 WRITE(6,502)
07180 WRITE(6,503)
07190 WRITE(6,504) (INDEX(I),A(I),B(I),C(I),INDEX(I),EA(I),ESTM(I,1),
07200 >EB(I),ESTM(I,2),EC(I),ESTM(I,3),I=1,6)
07210 WRITE(6,505) N,K,VAR,D,THETA(IRHO),ESTM(1,4),NE
07220 TVAR(1)=SALOSS(A,C,N)*VAR
07230 WRITE(6,506) TVAR(1),ALFA,NBE
07240 WRITE(6,507) ALFAD,NCE,NRELS,KE
07250 WRITE(6,508) RL1,RLO,RL
07260 WRITE(6,509) QW(1,1),QW(IRHO,IRHO)
07270 IN=NB+1
07280 RKB=0.00
07290 DO 600 J=1,IN
07300 600 RKB=RKB+B(J)
07310 IN=N+1
07320 RKA=0.00
07330 DO 670 J=1,IN
07340 670 RKA=RKA+A(J)
07350 IF (Q1.EQ.0.00) GO TO 620
07360 C----- DB(V+1) PARAMETER UDREGNING -----
07370 Q0=1.00/((1.00-Q1*A(2)/100.0)*RKB)
07380 ALFA1=1.00-1.00/(Q0*RKB)
07390 RHJ(1)=Q0
07400 RHJ(2)=-Q0*ALFA1
07410 CALL MULT(A,N,RHJ,1,Q,NQ1)
07420 CALL MULT(B,NB,RHJ,1,P,NP1)
07430 IN=NP1+1
07440 NP1=NP1+K
07450 DO 610 J=1,IN
07460 610 P(NP1+2-J)=-P(NP1+1-J)
07470 P(1)=1.00
07480 GO TO 630
07490 620 CONTINUE
07500 C----- DB(V) PRAMETERUDREGNING -----
07510 Q0=1.00/RKB
07520 ALFA1=0.00
07530 NQ1=N
```

```
07540      IN=N+1
07550      DO 640 J=1,IN
07560 640   Q(J)=Q0*A(J)
07570      NP1=NB+K
07580      IN=NB+1
07590      DO 650 J=1,IN
07600 650   P(J+K)=-Q0*B(J)
07610      P(1)=1.00
07620 630   CONTINUE
07630      TUO=0.00
07640 C----- OPSTART AF SYSTEM -----
07650      DO 700 I=1,6
07660      XFILT(I,1,3)=D/RKA
07670 700   XFILT(I,2,3)=D
07680 C----- VARIANSBEREGNINGEN -----
07690      CALL MULT(P,NP1,C,NC,RHJ,N2)
07700      TVAR(1)=SALOSS(A,RHJ,MAX0(N2,N))*VAR
07710      IN=NC+2
07720      DO 660 J=2,IN
07730 660   RHJ(J)=Q0*(C(J)-ALFA1*C(J-1))
07740      RHJ(1)=Q0
07750      N2=NC+1
07760      TVAR(2)=SALOSS(ET,RHJ,N2)*VAR
07770      IN=NC+1
07780      DO 680 I=1,IN
07790 680   TVAR(3)=TVAR(3)+C(I)**2
07800      TVAR(3)=TVAR(3)*VAR
07810      IF (NRELS.GE.0) TVAR(3)=VAR
07820      WRITE(6,5010) (INDEX(I),P(I),Q(I),I=1,6)
07830 C----- UDSKRIVNING AF PARM. OG VAR. -----
07840      WRITE(6,5013) TVAR(1),TVAR(2),TUO
07850      RHJ(1)=1.00/RKB
07860      RHJ(2)=1.00/((1.00-A(2))*RKB)
07870      WRITE(6,5014) RHJ(1),Q0,RHJ(2)
07880      WRITE(6,5015) ALFA1
07890      WRITE(6,5016) ITYPE,IREGTP
07900      WRITE(6,5017)
07910      DO 120 J=1,3
07920      IF (TVAR(J).LT.0.00001) TVAR(J)=1.00
07930 120   CONTINUE
07940 C+++++ FORMAT SAETNINGERNE ++++++
07950 1010  FORMAT(I3,2X,10(F8.4,3X))
07960 C
07970 1020  FORMAT(I3,2X,10(F10.4,3X))
07980 C
07990 502   FORMAT(1H1,2(117(1H*)/1H ),'***',T117,'**'/1H , '**',T45,
08000     >'SELVINDSTILLENDEN DEADBEAT REGULATOR',T117,'**'/1H ,
08010     >'***',T117,'**'/1H ,2(117(1H*)/1H ), '**',T60,'**',T118,'**'/1H ,
08020     >'**',T25,'OBJEKT BESKRIVELSEN',T60,'**',T80,'ESTIMATIONS DATA',T118,
08030     >'**'/1H , '**',T60,'**',T118,'**'/1H ,117(1H*)/1H , '**',T60,'**',
08040     >T118,'**',/)
08050 C
08060 503   FORMAT(1H+,'**',T14,'N',T22,'A',T33,'B',T44,'C',T60,'**',T71,'N',
08070     >T79,'EA',T91,'EB',T103,'EC',T118,'**'/1H , '**',T60,'**',T118,'**',/)
08080 C
08090 504   FORMAT(1H+,6('**',T13,I2,3(3X,F8.4),T60,'**',T70,I2,3X,3(F8.4,1X,
08100     >1A1,2X),T118,'**'/1H , '**',T60,'**',T118,'**'/1H ))
08110 C
08120 505   FORMAT(1H+,'**',5X,'N = ',I2,3X,'K = ',I2,3X,'VAR = ',F8.4,3X,
08130     >'D = ',F8.4,T60,'**',T65,'D = ',F8.4,1X,A1,2X,'N = ',I2,T118,
08140     >'**'/1H , '**',T60,'**',T118,'**',/)
08150 C
08160 506   FORMAT(1H+,'**',T23,'VAR(Y) = ',F8.4,T60,'**',T65,'ALFA = ',F8.1,
08170     >4X,'NB = ',I2,T118,'**'/1H , '**',T60,'**',T118,'**',/)
08180 C
08190 507   FORMAT(1H+,59(1H*),T65,'ALFAD = ',F8.1,4X,'NC = ',I2,T118,'**'/1H ,
08200     >'**',T60,'**',T118,'**'/1H , '**',T25,'REGULATOR PARM.',T60,'**',T65,
08210     >'NRELS = ',I2,10X,'K = ',I2,T118,'**'/1H , '**',T60,'**',T118,'**',/)
08220 C
08230 508   FORMAT(1H+,59(1H*),T63,' L1 = ',F8.4,4X,'LO = ',F8.4,
08240     >2X,'L = ',F8.4,T118,'**'/1H , '**',T60,'**',T118,'**',/)
08250 C
08260 509   FORMAT(1H+,'**',T14,'N',T22,'P**',T33,'Q',T60,'**'
08270     >T63,' QW = ',F8.4,4X,'QD = ',F8.4
08280     >,T118,'**'/1H , '**',T60,'**',T118,'**',/)
08290 C
08300 5010  FORMAT(1H+,6('**',T13,I2,2(3X,F8.4),T60,'**',
08310     >T118,'**'/1H , '**',T60,'**',T118,'**'/1H ))
08320 C
08330 5013  FORMAT(1H+,'**',2X,'VARY = ',F8.4,3X,'VARU = ',F8.4,3X,' UO = ',
```

```
08340 >F8.4,T60,'*',T118,'*'/1H,'*',T60,'*',T118,'*'/)
08350 C
08360 5014 FORMAT(1H+,'*',9X,F8.4,2X,'>',2X,'Q0 =',F8.4,4X,'>',4X,F8.4,
08370 > T60,'*',T118,'*'/',T60,'*',T118,'*'/)
08380 C
08390 5015 FORMAT(1H+,'*',T23,'ALFA =',F8.4,T60,'*',
08400 > T118,'*'/',T60,'*',T118,'*'/)
08410 5016 FORMAT(1H+,'*',10X,'DB - TYPE :',I3,4X,'IREGTP =',I3,
08420 C
08430 > T60,'*',T118,'*'/',7('*',T60,'*',T118,'*'/')
08440 C
08450 5017 FORMAT(1H+,117(1H*))
08460 C
08470 RETURN
08480 END
08490 C
08500 C
08510 C*****
08520 SUBROUTINE OUT1(I,I0,I1,Y,U,W,ESTYRE,EPS,VAR,D)
08530 C*****
08540 C
08550 C PROCEDURE TIL OPSAMLING, BEHANDLING OG UDSKRIVNING AF SIGNAL
08560 C
08570 C*****
08580 COMMON /TPARM/TQ(6),TP(6),NQ1,NP1,TUO,TVAR(3),RKA,RKB
08590 COMMON /UDDAT1/SIGNAL(101,3),VAR1(101,2),VAR2(101,1),TAB(101,3),
08600 >RKVAD1,RKVAD2,RKVAD3
08610 REAL O2(2)/2*0.00/
08620 REAL O3(3)/3*0.00/
08630 REAL STFEJL(101,1)
08640 IF ((I.LT.I0+1).OR.(I.GT.I0+100*I1+1)) RETURN
08650 IF (I.EQ.I0+100*I1+1) GO TO 10
08660 C----- DATA - BEHANDLING -----
08670 U1=U-(RKA*W-D)/RKB
08680 RKVAD1=RKVAD1+ESTYRE**2/TVAR(1)
08690 RKVAD2=RKVAD2+U1**2/TVAR(2)
08700 RKVAD3=RKVAD3+EPS**2/TVAR(3)
08710 RMID=RINTEG(4,I,ESTYRE)
08720 IF (VAR.EQ.0.00) GO TO 10
08730 V1=RINTEG(1,I,ESTYRE**2)-RMID**2
08740 V2=RINTEG(2,I,U1**2) -RINTEG(5,I,U1)**2
08750 V3=RINTEG(3,I,EPS**2) -RINTEG(6,I,EPS)**2
08760 C CALL HIST1(1,Y)
08770 C CALL HIST1(2,U)
08780 C CALL HIST1(3,EPS)
08790 C----- DATA - OPSAMLING -----
08800 10 CONTINUE
08810 IF (MOD(I-I0-1,I1).NE.0) RETURN
08820 I2=(I-I0-1)/I1+1
08830 SIGNAL(I2,1)=Y
08840 SIGNAL(I2,2)=U
08850 SIGNAL(I2,3)=W
08860 STFEJL(I2,1)=RMID
08870 TAB(I2,1)=RKVAD1
08880 TAB(I2,2)=RKVAD2
08890 TAB(I2,3)=RKVAD3
08900 IF (VAR.EQ.0.00) GO TO 20
08910 VAR1(I2,1)=V1/TVAR(1)
08920 VAR1(I2,2)=V2/TVAR(2)
08930 VAR2(I2,1)=V3/TVAR(3)
08940 20 CONTINUE
08950 IF (I2.NE.101) RETURN
08960 C----- DATA UDSKRIFT -----
08970 IF ((I1.NE.1).AND.(VAR.NE.0.00)) GO TO 30
08980 CALL IPLT(3,3,I0,I1,1,5.00,-5.0,SIGNAL,O3,
08990 > 'SIGNALERNE Y, U, OG W ')
09000 C- - - - - TABENE - - - - -
09010 30 CONTINUE
09020 RAMAX=AMAX1(TAB(101,1),TAB(101,2),TAB(101,3),1.00)
09030 CALL IPLT(3,3,I0,I1,1,RAMAX,0.0,TAB,O3,
09040 > 'TAB AF ESTYRE, U, EPS ')
09050 CALL IPLT(1,1,I0,I1,1,1.00,-1.00,STFEJL,O2,
09060 > 'MIDDEL AF STYRE-FEJL ')
09070 WRITE(6,1010) (TVAR(L),L=1,3)
09080 IF (VAR.EQ.0.00) RETURN
09090 C- - - - - VARIANSERNE - - - - -
09100 CALL IPLT(2,2,I0,I1,1,5.0,0.0,VAR1,O2,
09110 > 'VARIANSEN AF FEJL OG U ')
09120 CALL IPLT(1,1,I0,I1,1,5.0,0.0,VAR2,O3,
09130 > 'VARIANSEN AF EPS ')
```

```

09140 1010 FORMAT(1H , ' * ', T118, ' * ' / ' * ', 13X, 'VAR', 14X, 4(F8.4, 14X), T118, ' * ' /
09150 > ' * ', T118, ' * ' / ' * ', 117(1H*))
09160 RETURN
09170 END
09180 C
09190 C
09200 C*****
09210 SUBROUTINE KONV(IT, IO, I1, P, NP, Q, NQ, A, NA, B, NB, C, NC, K, VAR, CD, CE, TH)
09220 C*****
09230 COMMON /TPARM/TQ(6), TP(6), NQ1, NP1, TUO, TVAR(3), RKA, RKB
09240 INTEGER L/0/
09250 REAL RB(6)/6*0.0/, RN(6)/6*0.0/, R1(6), R2(6)/1.0, 5*0.0/,
09260 > ACAMP(101, 3), O3(3)/3*0.0/
09270 REAL GRAENS(101, 3)
09280 DIMENSION A(6), B(6), C(6), P(6), Q(6)
09290 C+++++***** INITIALISERING *****
09300 IF (MOD(IT-IO-1, I1).NE.0) RETURN
09310 IF ((IT.LT.IO+1).OR.(IT.GT.IO+100*I1+1)) RETURN
09320 I2=(IT-IO-1)/I1+1
09330 IF (L.NE.0) GO TO 10
09340 L=1
09350 IN=NB+1
09360 DO 20 I=1, IN
09370 20 RB(I+K)=B(I)
09380 NRB=NB+K
09390 RVAR=VAR
09400 IF (VAR.EQ.0.0) RVAR=1.00
09410 IN=NP1+1-K
09420 DO 30 I=1, IN
09430 30 R1(I)=-TP(I+K)
09440 RVAR1=SALOSS(R2, R1, NP1-K)
09450 10 CONTINUE
09460 C+++++***** DATABEHANDLING *****
09470 CALL MULT(P, NP, A, NA, R1, NR1)
09480 CALL MULT(Q, NQ, RB, NRB, R2, NR2)
09490 CALL ADD(R1, NR1, R2, NR2, RN, NRN)
09500 CALL MULT(Q, NQ, RB, NRB, R1, NR1)
09510 ACAMP(I2, 1)=SALOSS(RN, R1, MAXO(NRN, NR1))/RVAR1
09520 CALL MULT(P, NP, C, NC, R1, NR1)
09530 ACAMP(I2, 2)=SALOSS(RN, R1, MAXO(NRN, NR1))*RVAR/TVAR(1)
09540 CALL MULT(Q, NQ, C, NC, R1, NR1)
09550 ACAMP(I2, 3)=SALOSS(RN, R1, MAXO(NRN, NR1))*RVAR/TVAR(2)
09560 GRAENS(I2, 1)=TH
09570 GRAENS(I2, 2)=CD
09580 GRAENS(I2, 3)=CE
09590 IF (I2.NE.101) RETURN
09600 C+++++***** UDSKRIVNING *****
09610 CALL IPLT(3, 3, IO, I1, 1, 2.00, 0.00, ACAMP, O3,
09620 > 'AC FST AF Y/W, Y/E OG U/E')
09630 CALL IPLT(3, 3, IO, I1, 1, 5.00, 0.00, GRAENS, A,
09640 > 'THRESH, CD OG DELTA ')
09650 RETURN
09660 END
09670 C
09680 C
09690 C*****
09700 SUBROUTINE OUT2(I, IO, I1, A, B, C, D, TRP, PD)
09710 C*****
09720 C
09730 C PROCEDURE TIL OPSAMLING AF ESTIMERET PARAMETER
09740 C
09750 C*****
09760 COMMON /RMLDAT/P(23, 23), THETA(23), X(23), Z(23), EPS1, Y1, RL, RLO, RL1,
09770 > QW(23, 23), QE, T, NRELS, N1(3), IRHO, N, M, K, NC
09780 DIMENSION A(6), B(6), C(6)
09790 REAL TRACE(101, 2), AUD(101, 6), BUD(101, 6), CUD(101, 6), DUD(101)
09800 C----- DATA OPSAMLINGEN -----
09810 IF ((I.LT.IO+1).OR.(I.GT.IO+100*I1+1)) RETURN
09820 IF (MOD(I-IO-1, I1).NE.0) RETURN
09830 I2=(I-IO-1)/I1+1
09840 IF (N1(1).EQ.0) GO TO 40
09850 IN=N1(2)-1
09860 DO 50 J=1, IN
09870 50 AUD(I2, J+1)=THETA(J)
09880 AUD(I2, 1)=1.00
09890 40 IN=M+1
09900 DO 60 J=1, IN
09910 60 BUD(I2, J)=THETA(N1(2)+J-1)
09920 IF (NRELS) 70, 90, 90
09930 90 CONTINUE

```

```

09940      IF (NC.EQ.0) GO TO 70
09950      IN=NC+1
09960      DO 30 J=2,IN
09970 30    CUD(I2,J)=THETA(N1(3)+J-2)
09980 70    DUD(I2)=THETA(IRHO)
09990      TRACE(I2,1)=ALOG10(TRP)
10000      IF (PD.NE.0.0) TRACE(I2,2)=ALOG10(PD)
10010      IF (I2.NE.101) RETURN
10020 C-    - - - - - UDSKRIFT - - - - -
10030      IF (N1(1).NE.0) CALL IPLT(N+1,6,IO,I1,0,2.0,-2.0,AUD,A,
10040 >      'ESTIMATERNE AF A0, A1,...')
10050      CALL IPLT(M+1,6,IO,I1,0,2.0,-2.0,BUD,B,
10060 >      'ESTIMATERNE AF B0, B1,...')
10070      IF ((NC.GT.0).AND.(NRELS.GT.-1)) CALL IPLT(NC,6,IO,I1,0,
10080 >      2.0,-2.0,CUD,C,'ESTIMATERNE AF C0, C1, ')
10090      CALL IPLT(1,1,IO,I1,1,2.0,-2.0,DUD,D,
10100 >      'ESTIMATET AF D ')
10110      CALL IPLT(2,2,IO,I1,1,5.00,-5.00,TRACE,A,
10120 >      'SPORET AF P SAMT P(D) ')
10130      RETURN
10140      END
10150 C
12480 C*****
12490 C
12500 $ENTRY
12510 3 1.0000 -01.5000 00.7000 -00.0000 (N+1),A0,A1,...,AN
12520 2 1.0000 00.5000 -01.8920 (NB+1),B0,B1,...,BNB
12530 1 1.0000 -00.9000 (NC+1),C0,C1,...,CNC
12540 3 0.1000 0.0000 K,VAR,D
12550 5 000.0 IREGTP,Q0( I % )
12560 -5 1.0000 00.0000 NRELS,B0,QD
12570 2 NE
12580 1 NBE
12590 3 KE
12600 0 NCE
END OF DATA

```

PZ

```

00010 //A159270N JOB (***,NEU), 'N*K*P', CLASS=W
00020 /*ROUTE PRINT R3
00030 $JOB WATFIV, L=0, P=30, NOLIST
00040 BLOCK DATA
00050 COMMON /RMLDAT/P, QW, THETA, X, Z, EPS1, Y1, RL, RLO, RL1, NRELS, N1, IRHO, NF,
00060 >NG, NH, NCC
00070 COMMON /FILDAT/X1
00080 COMMON /TPARM/TG, TF, TH, E, TGAMMA, NF2, NG2, NH2, TVAR, RKA, RKB
00090 C+++++
00100 COMMON /LGDATA/D1, SUM
00110 C COMMON /HISTD/IDATA, NHIST, R, N1HIST
00120 C+++++
00130 C COMMON /CORDAT/GEM, COR, N, RMID, NSEL, NCOR
00140 C*****
00150 REAL D1(8,50)/400*0.00/, SUM(8)/8*0.00/
00160 C REAL R(4,3)/12*0.00/
00170 C INTEGER IDATA(103,3)/309*0/, NHIST(3)/3*0/, N1HIST(3)/3*0/
00180 C+++++
00190 C REAL GEM(41,4), COR(41,10)/410*0.00/, RMID(4)/4*0.00/
00200 C INTEGER N/0/, NCOR/4/, NSEL(10,2)/1,1,2,2,3,3,4,4,12*0/
00210 C+++++
00220 REAL P(13,13)/169*0.00/, THETA(13)/13*0.00/, X(13)/13*0.00/,
00230 >Z(13)/13*0.00/, X1(6,2,18)/216*0.00/, TG(6)/6*0.00/,
00240 >TF(6)/6*0.00/, TH(6)/6*0.00/, E(6)/6*0.00/, QW(13,13)/169*0.00/
00250 REAL EPS1/0.00/, Y1/0.00/, RL/1.00/, RL1/1.00/, RLO/0.00/,
00260 >TGAMMA/0.00/
00270 REAL TVAR(4)/4*0.00/, RKA/0.00/, RKB/0.00/
00280 INTEGER NRELS/0/, N1(5)/5*0/, NF/0/, NG/0/, NH/0/, NF2/0/, NG2/0/, NH2/0/
00290 END
00300 C
00310 C*****
00320 C
00330 REAL XX(6,6)/36*0.00/, A(6)/6*0.00/, B(6)/6*0.00/, C(6)
00340 >/6*0.00/, F(6)/6*0.00/, G(6)/6*0.00/, H(6)/6*0.00/
00350 >, P(6)/6*0.00/, Q(6)/6*0.00/, D/0.00/,
00360 >VAR/0.00/, Y/0.00/, U/0.00/, W1/0.00/, E/0.00/
00370 >, EPS/0.00/, ET(6)/1.00, 5*0.00/, ESTYRE/0.00/
00380 REAL AM(6)/6*0.00/, BM(6)/6*0.00/, TAM(6)/6*0.00/, TAMP1(6)/6*0.00/
00390 >, P2(6)/6*0.00/
00400 REAL T(6)/6*0.00/, QE/10000.0/
00410 REAL GR/10.0/
00420 REAL U1/0.00/, U2/0.00/, YM1/0.00/, Y2/0.00/, WF/0.00/
00430 INTEGER NN/-1/, K/-1/, IREGTP/-1/, IGRAEN/0/
00440 C
00450 C$PRINTON
00460 C
00470 C*****
00480 C*****
00490 C**
00500 C** HOVEDPROGRAM **
00510 C** **
00520 C*****
00530 C*****
00540 C
00550 C
00560 CALL INPUT(NN, A, N, B, NB, C, NC, D, K, VAR, P, NP, Q, NQ, AM, NAM, BM, NBM,
00570 > TAM, NTAM, TAMP1, NTAMP1, P2, NP2, BO, BETA, IREGTP,
00580 > T, NT)
00590 ISTOP=401
00600 ISTART=1

```

```
00600      ISTART=1
00610      IR=1
00620      IP=2
00630      IST=0
00640      IINT=(ISTOP-ISTART)/100
00650      SPR=SQRT(VAR)
00660      DO 10 I=1,ISTOP
00670          CALL DELAY(K,XX,U,U1,U2,YM1,Y2,WF)
00680          W=REF(I,25,0)
00690          IF (I.GE.20) CALL STOEJ(I,SPR,25.0,1,0.02,E)
00700 C      IF (I.GT.40) D=AMIN1(D+0.01,1.00)
00710 C      CALL MARKOV(I,5.0,0.01,D)
00720          IF (I.LT.20) CALL GAUSS(NN,SPR,0.0,E)
00730 C      IF (I.GT.40) D=1.0
00740 C      IF (I.EQ.41) E=30.00
00750 C      CALL PERTUB(I,E,VAR,1.5,5.0)
00760          Y=FILTER(1,A,N,B,NB,XX(K,1))+FILTER(2,A,N,C,NC,E)
00770 >          +FILTER(3,A,N,ET,0,D)
00780          ESTYRE=W-Y
00790          YM1=FILTER(4,AM,NAM,BM,NBM,W)
00800          YM=XX(K,4)
00810          YF=FILTER(5,P,NP,Q,NQ,Y)
00820          Y2=FILTER(6,TAM,NTAM,ET,0,YF)
00830          WF=FILTER(7,P,NP,Q,NQ,YM1)
00840          YE1=YF-B0*XX(K,2)
00850          YE2=XX(K,5)
00860          YE3=XX(K+1,3)
00870          YE4=XX(K,6)
00880          CALL RML(YE1,YE2,YE3,YE4,EPS,EPS2,IR,GR,TRP,PD,IGR,QE,CD,CE)
00890          IF (IGR.EQ.1) WRITE(6,6060) I
00900 6060      FORMAT(1H-,T90,' GRAENSE P$ TRIN : ',I3)
00910          CALL GLR2(I,10,100.0,EPS2,QE,GR,YE1,YE2,YE3,YE4,IST,IR,THRESH)
00920          CALL REGPAR(IREGTP,G,NG,F,NF,H,NH,U0)
00930          U1=FILTER(18,ET,0,H,NH,WF)-FILTER(8,ET,0,G,NG,XX(1,3))
00940 >          -FILTER(9,ET,0,F,NF,Y2)+U0
00950          U1=U1/B0
00960          U2=FILTER(17,P2,NP2,ET,0,U1)
00970          U =FILTER(16,Q,NQ,TAMP1,NTAMP1,U1)+BETA*U
00980          EF=YF-WF
00990          IF (IP.EQ.1) CALL OUT1(I,ISTART-1,IINT,Y,YM,U,W,ESTYRE,EF,EPS,
01000 >          VAR,D,E)
01010 >          IF (IP.EQ.2) CALL KONV(I,ISTART-1,IINT,A,N,B,NB,C,NC,K,P2,NP2,
01020 >          Q,NQ,T,NT,CD,CE,THRESH,
01030 >          P,NP,AM,NAM,BM,NBM,F,G,VAR,U0,D,W)
01040          IF (IP.EQ.3) CALL OUT2(I,ISTART-1,IINT,K,TRP,PD)
01050          IGRAEN=IGRAEN+IGR
01060 10      CONTINUE
01070          WRITE(6,1000) IGRAEN
01080          WRITE(6,1010)
01090 1000      FORMAT(1H-, 'ANTAL GRÆNSE TILFÆLDE : ',I4)
01100 1010      FORMAT(1H1)
01110          STOP
01120 C$PRINTOFF
01130          END
01140 C
01150 C
01160 C*****
01170 C**
01180 C**          UNDERPROGRAMMER          **
01190 C**
01200 C*****
01210 C*****
05830 C
05840 C*****
05850          SUBROUTINE REGPAR(N,G,NG,F,NF,H,NH,U0)
05860 C*****
05870 C
05880 C          REGPAR
05890 C
05900 C*****
05910 C
05920 C      PROGRAMMET KOPIERE REGULATORPARAMETERNE FRA DEN ESTIMEREDE
05930 C      PARAMETER VEKTOR, THETA,HVIS N IKKE ER NEGATIV. HVIS N ER
05940 C      NEGATIV UDLAESER DE TEORETISKE PARAMETRE.
05950 C
05960 C*****
05970          COMMON /RMLDAT/P(13,13),QW(13,13),THETA(13),X(13),Z(13),EPS1,Y1,RL
05980 > ,RLO,RL1,NRELS,N1(5),IRHO,NF1,NG1,NH1,NCC
05990          COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TUO,NF2,NG2,NH2,TVAR(4),
06000 > RKA,RKB
```



```
06010      DIMENSION G(6),F(6),H(6)
06020      IF (N) 10,20,20
06030 C----- KOPIERING AF TEORETISKE PARM. -----
06040 10      CONTINUE
06050      ING=NG2+1
06060      DO 30 I=1,ING
06070 30      G(I)=TG(I)
06080      INH=NH2+1
06090      DO 40 I=1,INH
06100 40      H(I)=TH(I)
06110      INF=NF2+1
06120      DO 50 I=1,INF
06130 50      F(I)=TF(I)
06140      UO=TUO
06150      NG=NG2
06160      NF=NF2
06170      NH=NH2
06180      RETURN
06190 C----- KOPIERING AF ESTIMEREDE PARM. -----
06200 20      CONTINUE
06210      IF (N1(3).EQ.0) GO TO 65
06220      ING=NG1+1
06230      DO 60 I=1,ING
06240 60      G(I)=THETA(N1(3)-1+I)
06250 65      INF=NF1+1
06260      DO 70 I=1,INF
06270 70      F(I)=THETA(N1(2)-1+I)
06280      H(1)=1.00
06290      IF (N1(4).EQ.0) GO TO 100
06300      INH=NH1+1
06310      DO 80 I=1,INH
06320 80      H(I)=THETA(N1(4)-1+I)
06330 100     CONTINUE
06340      UO=-THETA(IRHO)
06350      NG=NG1
06360      NF=NF1
06370      NH=NH1
06380      RETURN
06390      END
06400 C
06410 C
06420 C*****
06430      SUBROUTINE INPUT(NN,A,N,B,NB,C,NC,D,K,VAR,P,NP,Q,NQ,AM,NAM,
06440 >      BM,NBM,TAM,NTAM,TAMP1,NTAMP1,P2,NP2,BO,BETA,
06450 >      IREGTP,TD,NTD)
06460 C*****
06470 C
06480 C          INDLAESNING OG
06490 C          INITIALISERING
06500 C
06510 C*****
06520      COMMON /RMLDAT/PP(13,13),QW(13,13),THETA(13),X(13),Z(13),EPS1,Y1,
06530 >RL,RLO,RL1,NRELS,N1(5),IRHO,NF,NG,NH,NCC
06540      COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TUO,NF2,NG2,NH2,TVAR(4),
06550 >RKA,RKB
06560      COMMON /FILDAT/XFILT(6,2,18)
06570      DIMENSION A(6),B(6),C(6),P(6),Q(6),TD(6)
06580      DIMENSION AM(6),BM(6),TAM(6),TAMP1(6),P2(6)
06590      REAL EC(6)/6*0.0/,F(6)/6*0.0/,G(6)/6*0.0/,H(6)/6*0.0/
06600      REAL T(6)/6*0.00/,P1(6)/6*0.00/
06610      REAL PC(6)/6*0.00/,BE(6)/6*0.00/
06620      REAL AI(6)/6*0.00/
06630      REAL RI(6)/1.00,-1.00,4*0.00/
06640      REAL RD(6)/1.00,-0.90,4*0.00/
06650      REAL TAM1(6)/6*0.00/
06660      LOGICAL*1 STJERN/'*'/
06670      LOGICAL*1 BLK/' '/
06680      LOGICAL*1 ESTM(6,5)/30*' '/
06690      INTEGER INDEX(6)/0,1,2,3,4,5/
06700 C+++++ INDLAESNING AF SYSTEMSATA ++++++
06710      READ(5,1010)N,(A(I),I=1,N)
06720      READ(5,1010)NB,(B(I),I=1,NB)
06730      READ(5,1010)NC,(C(I),I=1,NC)
06740      N=N-1
06750      NB=NB-1
06760      NC=NC-1
06770      READ(5,1010)K,VAR,D
06780      NN=10317
06790      RKA=0.00
06800      IN=N+1
```

```
06810      DO 230 J=1,IN
06820 230   RKA=RKA+A(J)
06830      RKB=0.00
06840      IN=NB+1
06850      DO 140 J=1,IN
06860 140   RKB=RKB+B(J)
06870      IF (VAR.NE.0.00) GO TO 190
06880      DO 150 I=1,5
06890 150   C(I+1)=0.00
06900 190   CONTINUE
06910 C----- OPSTART AF SYSTEM -----
06920      DO 350 I=1,6
06930      XFILT(I,1,3)=D/RKA
06940 350   XFILT(I,2,3)=D
06950 C+++++ INDLAESNING AF KOSTFUNKTION ++++++
06960      READ(5,1010)NP1,(P1(I),I=1,NP1)
06970      READ(5,1010)NP2,(P2(I),I=1,NP2)
06980      READ(5,1010)NQ,(Q(I),I=1,NQ)
06990      READ(5,1010) NT,(T(I),I=1,NT)
07000      READ(5,1010)NAM,(AM(I),I=1,NAM)
07010      READ(5,1010)NBM,(BM(I),I=1,NBM)
07020      READ(5,1010) IREGTP,BETA,RD(2)
07030      NP1=NP1-1
07040      NP2=NP2-1
07050      NQ=NQ-1
07060      NT=NT-1
07070      NAM=NAM-1
07080      NBM=NBM-1
07090      CALL MULT(P1,NP1,P2,NP2,P,NP)
07100 C+++++ ESTIMATIONS DATA ++++++
07110      RL=1.00
07120      RL =0.95
07130      READ(5,1010) NRELS,B0,QD
07140      READ(5,1010) NE
07150      READ(5,1010) NBE
07160      READ(5,1010) KE
07170      READ(5,1010) NCE
07180      ALFA =1000.00
07190      ALFAD=1000.00
07200      IF (NCE.LT.0) GO TO 200
07210      NT=NCE
07220      NH=NCE
07230      DO 30 I=2,6
07240 30   T(I)=0.00
07250      T(1)=1.00
07260 200  CONTINUE
07270      NF=MAX0(NE-1,NAM-KE+NT)
07280      NG=MAX0(NBE+KE-1,NP2)-1
07290      NCC=0
07300      NTD=NT
07310      DO 270 I=1,6
07320 270  TD(I)=T(I)
07330      IF (BETA.NE.1.00) GO TO 260
07340      CALL MULT(T,NT,RD,1,TD,NTD)
07350      NF=NF+1
07360      ALFAD=0.00
07370 260  CONTINUE
07380 C----- EST. AF F - PARAMETERNE -----
07390      N1(2)=1
07400      INF=NF+1
07410      DO 10 I=1,INF
07420 10   ESTM(I,1)=STJERN
07430 C----- EST. AF G - PARAMETERNE -----
07440      IF (NG.LT.0) GO TO 25
07450      N1(3)=NF+2
07460      ING=NG+1
07470      DO 20 I=1,ING
07480 20   ESTM(I,2)=STJERN
07490 C----- EST. AF H - PARAMETERNE -----
07500 25   NG=MAX0(0,NG)
07510      IRHO=NF+NG+3
07520      H(1)=1.00
07530      IF (NCE.LT.0) GO TO 35
07540      N1(4)=IRHO
07550      IRHO=IRHO+NCE+1
07560      THETA(N1(4))=1.00
07570      IN=NCE+1
07580      DO 240 I=2,IN
07590 240  ESTM(I,3)=STJERN
07600 C----- EST. AF D - PARAMETEREN -----
```

```
07610 35. CONTINUE
07620 IF (ALFAD .NE. 0.00) ESTM(1,4)=STJERN
07630 IF (NRELS) 40,50,50
07640 C----- RLS - TILFAELDET -----
07650 40 RLO=0.00
07660 GO TO 75
07670 C----- RELS OG RML TILFAELDET -----
07680 50 RLO=0.99
07690 C----- E - PARAMETERNE -----
07700 N1(5)=IRHO
07710 IRHO=IRHO+NCC
07720 IF (NCC.EQ.0) GO TO 75
07730 DO 70 I=1,NCC
07740 70 ESTM(I+1,5)=STJERN
07750 C+++++ INITIALISERING AF THETA OG P ++++++
07760 75 CONTINUE
07770 EC(1)=1.00
07780 F(1)=THETA(N1(2))
07790 DO 80 I=1,IRHO
07800 80 PP(I,I)=ALFA
07810 PP(IRHO,IRHO)=ALFAD
07820 QW(IRHO,IRHO)=QD
07830 IF (N1(4).NE.0) PP(N1(4),N1(4))=0.00
07840 IF (ALFAD.NE.0.00) X(IRHO)=1.00
07850 C+++++ KONTOL UDSKRIFT ++++++
07860 WRITE(6,501)
07870 IF (BETA.EQ.1.00) WRITE(6,502)
07880 IF (NCE.LT.0) WRITE(6,5018)
07890 IF (NCE.GE.0) WRITE(6,5017)
07900 WRITE(6,5019)
07910 WRITE(6,503)
07920 WRITE(6,504) (INDEX(I),A(I),B(I),C(I),INDEX(I),F(I),ESTM(I,1),
07930 >G(I),ESTM(I,2),H(I),ESTM(I,3),I=1,6)
07940 WRITE(6,505) N,K,VAR,D,THETA(IRHO),ESTM(1,4),NE,B0
07950 TVAR(1)=SALOSS(A,C,N)*VAR
07960 WRITE(6,506) TVAR(1),ALFA,NBE,QD
07970 WRITE(6,507) ALFAD,NH,NRELS,KE
07980 WRITE(6,508) RL1,RLO,RL
07990 WRITE(6,509)
08000 WRITE(6,5010) (INDEX(I),AM(I),BM(I),T(I),INDEX(I),EC(I),ESTM(I,5),
08010 >I=1,6)
08020 C----- INITIALISERING AF POLYNOMIER -----
08030 CALL MULT(TD,NTD,AM,NAM,TAM,NTAM)
08040 CALL MULT(P1,NP1,TAM,NTAM,TAMP1,NTAMP1)
08050 C----- BEREGNING AF REG. PARAMETERNE -----
08060 IF (NCE.LT.0) GO TO 210
08070 DO 60 I=1,6
08080 60 TD(I)=C(I)
08090 IF (BETA.EQ.1.00) CALL MULT(C,NC,RD,1,TD,NTD)
08100 210 CONTINUE
08110 DO 290 I=1,6
08120 290 AI(I)=A(I)
08130 IF (BETA.EQ.1.00) CALL MULT(A,N,RI,1,AI,IN)
08140 CALL MULT(TD,NTD,AM,NAM,TAM1,NTAM1)
08150 CALL PREDIK(AI,IN,TAM1,NTAM1,K,E,TF,NF2)
08160 CALL MULT(B,NB,E,K-1,BE,NBE1)
08170 NG2=MAX0(NB+K-1,NP2)-1
08180 NG2=MAX0(NG2,0)
08190 ING=NG2+1
08200 DO 170 I=1,ING
08210 170 TG(I)=BE(I+1)-B(1)*P2(I+1)
08220 IF (NCE.LT.0) GO TO 220
08230 INH=NC+1
08240 DO 90 I=2,INH
08250 90 TH(I)=C(I)
08260 220 CONTINUE
08270 TH(1)=1.00
08280 IF (BETA.EQ.1.00) GO TO 280
08290 DO 100 I=1,K
08300 100 TUO=TUO+E(I)
08310 PC(1)=0.00
08320 PC(2)=0.00
08330 PC(3)=0.00
08340 DO 180 I=1,6
08350 PC(1)=PC(1)+TAM(I)
08360 PC(2)=PC(2)+P(I)
08370 PC(3)=PC(3)+Q(I)
08380 180 CONTINUE
08390 TUO=-TUO*PC(3)*D/(PC(1)*PC(2))
08400 280 CONTINUE
```

```
08410 C----- BEREKNING AF VARIANSERNE -----
08420 CALL MULT(E,K-1,C,NC,EC,NEC)
08430 C %% EC = E*C %%
08440 DO 300 I=1,6
08450 300 PC(I)=EC(I)
08460 NPC=NEC
08470 C %% PC = E*C %%
08480 IF (BETA.EQ.1.00) CALL MULT(EC,NEC,RI,1,PC,NPC)
08490 C %% PC = C*E*I %%
08500 TVAR(1)=SALOSS(TAM1,PC,MAXO(NTAM1,NPC))*VAR
08510 CALL MULT(P,NP,TAM1,NTAM1,EC,NEC)
08520 C %% EC = T*D*AM*P %%
08530 CALL MULT(Q,NQ,PC,NPC,BE,NEB)
08540 C %% BE = C*E*I*Q %%
08550 TVAR(3)=SALOSS(EC,BE,MAXO(NEC,NEB))*VAR
08560 TVAR(4)=TVAR(3)
08570 CALL MULT(TF,NF2,C,NC,PC,NPC)
08580 C %% PC = F*C %%
08590 CALL MULT(B,NB,TAM1,NTAM1,EC,NEC)
08600 C %% EC = B*T*D*AM %%
08610 TVAR(2)=SALOSS(EC,PC,MAXO(NEC,NPC))*VAR
08620 WRITE(6,5016) TVAR(4),TVAR(3)
08630 WRITE(6,5011)
08640 WRITE(6,504) (INDEX(I),Q(I),P1(I),P2(I),INDEX(I),TF(I),BLK,
08650 > TG(I),BLK,TH(I),BLK,I=1,6)
08660 WRITE(6,5013) IREGTP,RD(2),TVAR(1),TVAR(2),TUO
08670 C WRITE(6,5014)
08680 DO 120 J=1,4
08690 IF (TVAR(J).LT.0.00001) TVAR(J)=1.00
08700 120 CONTINUE
08710 RETURN
08720 C+++++ FORMAT SAETNINGERNE ++++++
08730 1010 FORMAT(I3,2X,10(F8.4,3X))
08740 C
08750 501 FORMAT(1H1,2(117(1H*)/ ' '), '***',T117, '***/' ' ***',T25,
08760 > ' S E L V I N D S T I L L E N D E P O L P L A '
08770 > ', C E R I N G S R E G U L A T O R ',T117, '***/' ' ***',
08780 > T117, '***/')
08790 C
08800 502 FORMAT(1H+, '***',T50, 'MED INTEGRALVIRKNING',T117, '***/' ' ***',
08810 > T117, '***/')
08820 C
08830 5017 FORMAT(1H+, '***',T50, 'STOCHATISK DIMENSIONERING',T117, '***/' ' ***',
08840 > T117, '***/')
08850 C
08860 5018 FORMAT(1H+, '***',T50, 'DETERMINISTISK DIMENSIONERING',T117, '***/'
08870 > ' ***',T117, '***/')
08880 C
08890 5019 FORMAT(1H+,2(117(1H*)/1H ), '***',T60, '***',T118, '***/'1H ,
08900 > '***',T25, 'OBJEKT BESKRIVELSEN',T60, '***',T80, 'ESTIMATIONS DATA',T118,
08910 > '***/'1H , '***',T60, '***',T118, '***/'1H , 117(1H*)/1H , '***',T60, '***',
08920 > T118, '***',/)
08930 C
08940 503 FORMAT(1H+, '***',T14, 'N',T22, 'A',T33, 'B',T44, 'C',T60, '***',T71, 'N',
08950 > T79, 'F',T91, 'G',T103, 'H',T118, '***/'1H , '***',T60, '***',T118, '***',/)
08960 C
08970 504 FORMAT(1H+,6( '***',T13,I2,3(3X,F8.4),T60, '***',T70,I2,3X,3(F8.4,1X,
08980 > 1A1,2X),T118, '***/'1H , '***',T60, '***',T118, '***/'1H ))
08990 C
09000 505 FORMAT(1H+, '***',5X, 'N = ',I2,3X, 'K = ',I2,3X, 'VAR = ',F8.4,3X,
09010 > 'D = ',F8.4,T60, '***',T65, 'UO = ',F8.4,1X,A1,2X, 'N = ',I2,
09020 > 7X, 'BO = ',F8.4,T118,
09030 > '***/'1H , '***',T60, '***',T118, '***',/)
09040 C
09050 506 FORMAT(1H+, '***',T23, 'VAR(Y) = ',F8.4,T60, '***',T65, 'ALFA = ',F8.1,
09060 > 4X, 'NB = ',I2,7X, 'QD = ',F8.4,T118, '***/'1H , '***',T60, '***',T118, '***',/)
09070 C
09080 507 FORMAT(1H+,59(1H*),T65, 'ALFAD = ',F8.1,4X, 'NC = ',I2,T118, '***/'1H ,
09090 > '***',T60, '***',T118, '***/'1H , '***',T25, 'KOSTFUNKTIONEN',T60, '***',T65,
09100 > 'NRELS = ',I2,10X, 'K = ',I2,T118, '***/'1H , '***',T60, '***',T118, '***',/)
09110 C
09120 508 FORMAT(1H+,59(1H*),T63, ' L1 = ',F8.4,4X, 'LO = ',F8.4,
09130 > 2X, 'L = ',F8.4,T118, '***/'1H , '***',T60, '***',T118, '***',/)
09140 C
09150 509 FORMAT(1H+, '***',T14, 'N',T22, 'AM',T33, 'BM',T44, 'T',T60, '***',T71, 'N'
09160 > ,T79, 'E',T118, '***/'1H , '***',T60, '***',T118, '***',/)
09170 C
09180 5010 FORMAT(1H+,6( '***',T13,I2,3(3X,F8.4),T60, '***',T70,I2,3X,
09190 > 1(F8.4,1X,1A1,2X),T118, '***/'1H , '***',T60, '***',T118, '***/'1H ))
09200 C
```

```
09210 5011  FORMAT(1H+,117(1H*)/1H , '*' ,T60 , '*' ,T118 , '*' /1H , '*' ,T25,
09220  >'KOSTFUNKTIONEN',T60 , '*' ,
09230  >T80 , 'KORREKTE REGULATOR PARAMETRE',T118 , '*' /1H , '*' ,T60 , '*' ,
09240  >T118 , '*' /1H , 117(1H*)/1H , '*' ,T60 , '*' ,T118 , '*' /1H , '*' ,T14 ,
09250  >'N',T22 , 'Q',T33 , 'P1',T44 , 'P2',T60 , '*' ,T79 , 'F',T91 , 'G',T103 , 'H',
09260  >T118 , '*' / '*' ,T60 , '*' ,T118 , '*' /1H )
09270  C
09280 5013  FORMAT(1H+ , '*' ,17X , 'IREGTP = ',I2,
09290  > 2X , 'DELTA = ',F8.4 ,T60 , '*' ,
09300  >2X , 'VAR(Y) = ',F8.4 ,3X ,
09310  >'VAR(U) = ',F8.4 ,3X ,
09320  >'UO      = ',F8.4 ,T118 , '*' /1H , '*' ,T60 , '*' ,T118 , '*' /' ',117(1H*))
09330  C
09340 C5014  FORMAT(1H+,30( '*' ,T60 , '*' ,T118 , '*' /1H ) ,117(1H*))
09350  C
09360 5016  FORMAT(1H+ , '*' ,16X , 'VAR(EF) = ',F8.4 ,T60 , '*' ,4X , 'VAR(EPS) = ',F8.4 ,
09370  >      T118 , '*' /' ' ,T60 , '*' ,T118 , '*' /)
09380  END
09390  C
09400  C
09410 C*****
09420  SUBROUTINE OUT1(I,I0,I1,Y,YM,U,W,ESTYRE,EF,EPS,VAR,D,EN)
09430 C*****
09440  C
09450  C      PROCEDURE TIL OPSAMLING, BEHANDLING OG UDSKRIVNING AF SIGNAL
09460  C
09470 C*****
09480  COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TGAM,NF2,NG2,NH2,TVAR(4),
09490  >RKA,RKB
09500  REAL SIGNAL(101,4),VAR1(101,2),VAR2(101,2),TAB(101,4)
09510  REAL RKVAD1/0.0/,RKVAD2/0.0/,RKVAD3/0.0/,RKVAD4/0.0/
09520  REAL RMIDD(101,3)
09530  REAL O2(2)/2*0.00/
09540  REAL O3(3)/3*0.00/
09550  REAL O4(4)/4*0.00/
09560  IF ((I.LT.I0+1).OR.(I.GT.I0+100*I1+1)) RETURN
09570  IF (I.EQ.I0+100*I1+1) GO TO 10
09580 C----- DATA - BEHANDLING -----
09590  U1=U-(RKA*W-D)/RKB
09600  RKVAD1=RKVAD1+ESTYRE**2/TVAR(1)
09610  RKVAD2=RKVAD2+U1**2/TVAR(2)
09620  RKVAD3=RKVAD3+EPS**2/TVAR(3)
09630  RKVAD4=RKVAD4+EF**2/TVAR(4)
09640  R=RINTEG(5,I,ESTYRE)
09650  IF (VAR.EQ.0.00) GO TO 10
09660  V1=RINTEG(1,I,ESTYRE**2)-R**2
09670  V2=RINTEG(2,I,U1**2)-RINTEG(6,I,U1)**2
09680  V3=RINTEG(3,I,EPS**2)-RINTEG(7,I,EPS)**2
09690  V4=RINTEG(4,I,EF**2)-RINTEG(8,I,EF)**2
09700  C      CALL HIST1(1,Y)
09710  C      CALL HIST1(2,U)
09720  C      CALL HIST1(3,EPS)
09730 C----- DATA - OPSAMLING -----
09740 10  CONTINUE
09750  IF (MOD(I-I0-1,I1).NE.0) RETURN
09760  I2=(I-I0-1)/I1+1
09770  SIGNAL(I2,1)=Y
09780  SIGNAL(I2,2)=U
09790  SIGNAL(I2,3)=YM
09800  SIGNAL(I2,4)=W
09810  RMIDD(I2,3)=R
09820  RMIDD(I2,1)=D/10.0
09830  RMIDD(I2,2)=EN
09840  TAB(I2,1)=RKVAD1
09850  TAB(I2,2)=RKVAD2
09860  TAB(I2,3)=RKVAD3
09870  TAB(I2,4)=RKVAD4
09880  IF (VAR.EQ.0.00) GO TO 20
09890  VAR1(I2,1)=V1/TVAR(1)
09900  VAR1(I2,2)=V2/TVAR(2)
09910  VAR2(I2,1)=V3/TVAR(3)
09920  VAR2(I2,2)=V4/TVAR(4)
09930 20  CONTINUE
09940  IF (I2.NE.101) RETURN
09950 C----- DATA UDSKRIFT -----
09960  IF (I1.NE.1) GO TO 30
09970  CALL IPLT(4,4,I0,I1,1,5.00,-5.0,SIGNAL,04,
09980  >      'SIGNALERNE Y, U, YM, OG W')
09990 C----- TABENE -----
10000 30  CONTINUE
```

```

10010      RAMAX=AMAX1(TAB(101,1),TAB(101,2),TAB(101,3),TAB(101,4),1.00)
10020      CALL IPLT(4,4,IO,I1,1,RAMAX,0.0,TAB,04,
10030      >      'TAB AF ESTYRE, U, EPS, EF')
10040      WRITE(6,1010) (TVAR(L),L=1,4)
10050      CALL IPLT(3,3,IO,I1,1,1.0,-1.0,RMIDD,03,
10060      >      'D/10, E OG MID. AF ESTYRE')
10070      IF (VAR.EQ.0.00) RETURN
10080 C-      - - - - - VARIANSERNE - - - - -
10090      CALL IPLT(2,2,IO,I1,1,5.0,0.0,VAR1,02,
10100      >      'VARIANSEN-AF FEJL OG U ')
10110      CALL IPLT(2,2,IO,I1,1,5.0,0.0,VAR2,02,
10120      >      'VARIANSEN AF EPS OG EF ')
10130 C-      - - - - - HISTOGRAMMERNE - - - - -
10140      I3=0
10150 C      CALL HIST2(1,'Y ',I3)
10160 C      CALL HIST2(2,'U ',I3)
10170 C      CALL HIST2(3,'EPS ',I3)
10180      RETURN
10190 1010  FORMAT(1H,'*',T118,'*'/',13X,'VAR',14X,4(F8.4,14X),T118,'*'/
10200      >      ',*',T118,'*'/',117(1H*))
10210      END
10220 C
10230 C
10240 C*****
10250      SUBROUTINE OUT2(I,IO,I1,K,TRP,PD)
10260 C*****
10270 C
10280 C      PROCEDURE TIL OPSAMLING AF ESTIMERET PARAMETER
10290 C
10300 C*****
10310      COMMON /UDDAT2/
10320      >FUD(101,6),GUD(101,6),HUD(101,6),EUD(101,6)
10330      >,GAMUD(101)
10340      >,TRACE(101,2)
10350      COMMON /RMLDAT/P(13,13),QW(13,13),THETA(13),X(13),Z(13),EPS1,Y1,RL
10360      >      ,RLO,RL1,NRELS,N1(5),IRHO,NF,NG,NH,NCC
10370      COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TGAMMA,NF2,NG2,NH2,TVAR(4),
10380      >RKA,RKB
10390 C----- DATA OPSAMLINGEN -----
10400      IF ((I.LT.IO+1).OR.(I.GT.IO+100*I1+1)) RETURN
10410      IF (MOD(I-1-IO,I1).NE.0) RETURN
10420      I2=(I-IO-1)/I1+1
10430      IN=NF+1
10440      DO 60 J=1,IN
10450 60      FUD(I2,J)=THETA(N1(2)+J-1)
10460      IF (N1(3).EQ.0) GO TO 120
10470      IN=NG+1
10480      DO 100 J=1,IN
10490 100      GUD(I2,J)=THETA(N1(3)+J-1)
10500 120      CONTINUE
10510      IF (N1(4).EQ.0) GO TO 50
10520      IN=NH+1
10530      DO 110 J=1,IN
10540 110      HUD(I2,J)=THETA(N1(4)+J-1)
10550 50      CONTINUE
10560      IF (NRELS) 90,70,70
10570 70      CONTINUE
10580      IF (NCC.EQ.0) GO TO 90
10590      IN=NCC+1
10600      DO 80 J=2,IN
10610 80      EUD(I2,J)=THETA(N1(5)+J-2)
10620      EUD(I2,1)=1.00
10630 90      GAMUD(I2)=THETA(IRHO)
10640      TRACE(I2,1)=ALOG10(TRP)
10650      TRACE(I2,2)=ALOG10(PD)
10660      IF (I2.NE.101) RETURN
10670 C-      - - - - - UDSKRIFT - - - - -
10680      CALL IPLT(NF+1,6,IO,I1,0,2.0,-2.0,FUD,TF,
10690      >      'ESTIMATERNE AF FO, F1,...')
10700      IF (N1(3).EQ.0) NG=-10
10710      IF (NG.GE.0) CALL IPLT(NG+1,6,IO,I1,0,2.0,-2.0,GUD,TG,
10720      >      'ESTIMATERNE AF GO, G1,...')
10730      IF (NH.NE.0) CALL IPLT(NH+1,6,IO,I1,0,2.0,-2.0,HUD,TH,
10740      >      'ESTIMATERNE AF HO, H1,...')
10750      IF ((K.GT.1).AND.(NRELS.GT.-1)) CALL IPLT(K,6,IO,I1,0,
10760      >      2.0,-2.0,EUD,E,'ESTIMATERNE AF CCO, CC1,')
10770      IF (IRHO.NE.NF+NG+2) CALL IPLT(1,1,IO,I1,1,2.0,-2.0,GAMUD,TGAMMA,
10780      >      'ESTIMATET AF GAMMA ')
10790      CALL IPLT(2,2,IO,I1,1,5.0,-5.0,TRACE,TF,
10800      >      'TR(P) OG P(RHO,RHO) I DB ')

```

```
10810      RETURN
10820      END
10830 C
10840 C
10850 C*****
10860      SUBROUTINE KONV(IT,IO,I1,A,N,B,NB,C,NC,K,P2,NP2,Q,NQ,T,NT,
10870      >      CD,CE,THRESH,P,NP,AM,NAM,BM,NBM,F,G,VAR,UO,D,W)
10880 C*****
10890 C
10900 C      PROGRAM TIL UNDERSOEGELSE AF ADAPTIV REGULATORS KONVERGENS
10910 C      EGENSKABER. DER UNDERSOEGES KUN AC OG DC EGENSKABERNE
10920 C
10930 C+++++
10940      COMMON /TPARM/TG(6),TF(6),TH(6),E(6),TGAM,NF,NG,NH,TVAR(4),
10950      >RKA,RKB
10960      INTEGER L/O/
10970      DIMENSION A(6),B(6),C(6),P2(6),Q(6),T(6),P(6),AM(6),BM(6),
10980      >      F(6),G(6)
10990      REAL RP2(6)/6*0.0/,RT(6)/6*0.0/,RN(6)/6*0.0/,R1(6)/6*0.0/,
11000      >      R2(6)/6*0.0/,R(6)/6*0.0/,O4(4)/4*0.0/
11010      REAL ACAMP(101,4),DCAMP(101,3)
11020      REAL GRAENS(101,3)
11030      IF ((IT.LT.IO+1).OR.(IT.GT.IO+1+100*I1)) RETURN
11040      IF (MOD(IT-IO-1,I1).NE.0) RETURN
11050      I2=(IT-IO-1)/I1+1
11060      IF (L.NE.0) GO TO 10
11070 C+++++ INITIALISERING ++++++
11080      L=1
11090      RVAR=VAR
11100      IF (VAR.EQ.0.0) RVAR=1.00
11110      IN=NP2+1
11120      DO 20 I=1,IN
11130 20  RP2(I)=B(1)*P2(I)
11140      VAR1=SALOSS(AM,BM,MAXO(NAM,NBM))
11150      CALL MULT(BM,NBM,T,NT,R,NR)
11160      CALL MULT(R,NR,B,NB,RT,NRT)
11170      R(1)=RP2(1)
11180      NR=MAXO(NP2,NG+1)
11190      RK2=0.0
11200      RK3=0.0
11210      RK4=0.0
11220      RK5=0.0
11230      DO 80 I=1,K
11240      RK2=RK2+P(I)
11250      RK3=RK3+Q(I)
11260      RK4=RK4+T(I)
11270      RK5=RK5+AM(I)
11280 80  CONTINUE
11290      RK3=RKB*RK2*RK3*RK5*RK4/RK3
11300 C+++++ DATABEHANDLING ++++++
11310 10  CONTINUE
11320      IN=NR+1
11330      DO 30 I=2,IN
11340 30  R(I)=RP2(I)+G(I-1)
11350      CALL MULT(A,N,R,NR,R1,NR1)
11360      CALL MULT(B,NB,F,NF,R2,NR2)
11370      NRN=MAXO(NR1,K+NR2)
11380      IN=NRN+1
11390      DO 40 I=1,K
11400 40  RN(I)=R1(I)
11410      IK=K+1
11420      DO 50 I=IK,IN
11430 50  RN(I)=R1(I)+R2(I-K)
11440 C----- Y/W -----
11450      RK1=0.0
11460      RK2=0.0
11470      DO 60 I=1,6
11480      RK1=RK1+RN(I)
11490 60  RK2=RK2+RT(I)
11500      DCAMP(I2,1)=RK2/RK1-1.00
11510      ACAMP(I2,1)=SALOSS(RN,RT,MAXO(NRN,NRT))/VAR1
11520 C----- Y/E -----
11530      CALL MULT(C,NC,R,NR,R1,NR1)
11540      ACAMP(I2,2)=SALOSS(RN,R1,MAXO(NRN,NR1))*RVAR/TVAR(1)
11550      RK2=0.0
11560      DO 70 I=1,6
11570 70  RK2=RK2+R(I)
11580      DCAMP(I2,2)=RK2*D/RK1+RK3*UO/RK1
11590 C----- TOTAL ST. FEJL -----
11600      DCAMP(I2,3)=DCAMP(I2,1)*W+DCAMP(I2,2)
```

```
11610 C----- U/E -----
11620 ACAMP(I2,3)=SALOSS(RN,F,MAXO(NRN,NF))*RVAR/TVAR(2)
11630 C----- EPS/E -----
11640 CALL MULT(RN,NRN,P,NP,R1,NR1)
11650 CALL MULT(R,NR,Q,NQ,R2,NR2)
11660 ACAMP(I2,4)=SALOSS(R1,R2,MAXO(NR1,NR2))*RVAR/TVAR(3)
11670 C----- THRESHOLD GRÆNSER -----
11680 GRAENS(I2,1)=THRESH
11690 GRAENS(I2,2)=CD
11700 GRAENS(I2,3)=CE
11710 IF (I2.NE.101) RETURN
11720 C+++++ UDSKRIVNING +++++
11730 CALL IPLT(3,3,I0,I1,1,0.5,-0.5,DCAMP,04,
11740 > 'FEJL AF W, (D,UO) OG TOT.')
11750 CALL IPLT(4,4,I0,I1,1,2.0,0.0,ACAMP,04,
11760 > 'AC. AF Y/W,Y/E,U/E OG EPS')
11770 CALL IPLT(3,3,I0,I1,1,5.0,0.0,GRAENS,04,
11780 > 'THRESH, CD OG DELTA ')
11790 RETURN
11800 END
14530 C
14540 C*****
14550 $ENTRY
14560 3 1.0000 -01.5000 00.7000 -00.2019 (N+1),A0,A1, ... ,AN
14570 2 1.0000 00.5000 -00.2290 (NB+1),B0,B1, ... ,BNB
14580 1 1.0000 -00.8000 00.1500 (NC+1),C0,C1, ... ,
14590 3 0.1000 0.0000 K,VAR,D
14600 1 1.0000 (NP1+1),P1
14610 1 1.0000 00.1000 (NP2+1),P2
14620 1 1.0000 -00.3980 00.1353 0.0447 00.0031 Q
14630 1 1.0000 00.3980 00.0231 (NT+1),T
14640 1 1.0000 -00.3980 00.1353 (NAM+1),AM
14650 1 1.0000 00.1552 00.0000 (NBM+1),BM
14660 5 0.0000 -00.0000 IREGTP,BETA(1/0),DELTA
14670 -5 1.0000 0.0050 NRELS,B0,QD
14680 2 NE
14690 1 NBE
14700 3 KE
14710 -1 NCE
END OF DATA
```


Fig. 4

FÆLLES UNDERPROGRAMMER.

```
00010 C
00020 C
00030 C*****
00040 SUBROUTINE IPLT(N,IN,IO,II,ITYPE,RMAX,RMIN,DATA,PARM,TEKST)
00050 C*****
00060 C
00070 C PROGRAM TIL UDSKRIFT AF KURVER
00080 C
00090 C I A3 FORMAT
00100 C
00110 C*****
00120 C
00130 C VARIABELLISTE:
00140 C =====
00150 C
00160 C N ANTAL KURVER
00170 C IN DIMENSIONEN AF DATA
00180 C IO STARTTIDSPUNKT
00190 C II INTERVAL MELLEM KURVEPUNKTER
00200 C ITYPE VARIABEL BESTEMMENDE FOR KARAKTERENE
00210 C HVIS 1 FAES *,+,X,-
00220 C HVIS 0 FAES 1, 2, 3, ... OG NIVEAUER
00230 C RMAX MAXIMALNIVEAUET
00240 C RMIN MINIMALNIVEAUET
00250 C DATA ARRAY INDHOLDENDE DATA (101,IN)
00260 C PARM ARRAY INDEHOLDENDE DE N NIVEAUER
00270 C TEKST ARRAY (25 ) INDEHOLDENDE EN TEKSTSTRING
00280 C
00290 C+++++
00300 DIMENSION DATA(101,IN)
00310 DIMENSION PARM(IN)
00320 LOGICAL*1 TEKST(25)
00330 C ----- LOKALE VARIABLE -----
00340 DIMENSION IHJ3(11)
00350 LOGICAL*1 PAGE(101,101)
00360 LOGICAL*1 CHAR(20)/'1','2','3','4','5','6','7','8','9','0',' ',
00370 > '*','+','X','-','ø','&','>','<','/'
00380 C----- LODRETTE KANTER OG BLANKTEGN -----
00390 DO 10010 K=1,101
00400 PAGE(K,1)=CHAR(16)
00410 PAGE(K,101)=CHAR(16)
00420 IF (MOD(K-1,5).EQ.0) PAGE(K,101)=CHAR(13)
00430 IF (MOD(K-1,5).EQ.0) PAGE(K,1)=CHAR(13)
00440 DO 10010 I=2,100
00450 PAGE(K,I)=CHAR(11)
00460 IF ((MOD((I-1),20).EQ.0).OR.(MOD((K-1),10).EQ.0))
00470 > PAGE(K,I)=CHAR(20)
00480 10010 CONTINUE
00490 C----- INDSAET KURVER -----
00500 DO 10020 I=1,101
00510 PAGE(1,I)=CHAR(15)
00520 PAGE(101,I)=CHAR(15)
00530 PAGE(51,I)=CHAR(15)
00540 IF (MOD(I-1,10).EQ.0) PAGE(1,I)=CHAR(13)
00550 IF (MOD(I-1,10).EQ.0) PAGE(51,I)=CHAR(13)
00560 IF (MOD(I-1,10).EQ.0) PAGE(101,I)=CHAR(13)
00570 DO 10020 J=1,N
00580 KHJ=101-IFIX((DATA(I,N+1-J)-RMIN)*100.0/(RMAX-RMIN)+0.5)
00590 KHI=101-IFIX((PARM(N+1-J)-RMIN)*100.0/(RMAX-RMIN)+0.5)
00600 IF ((KHI.GT.101).OR.(KHI.LT.1)) GO TO 10015
```

```
00600      IF ((KHI.GT.101).OR.(KHI.LT.1)) GO TO 10015
00610      IF ((ITYPE.EQ.0).AND.(MOD(I,2).EQ.0)) PAGE(KHI,I)=CHAR(15)
00620 10015  IF ((KHJ.GT.101).OR.(KHJ.LT.1)) GO TO 10030
00630      IF (ITYPE.EQ.0) PAGE(KHJ,I)=CHAR(N+1-J)
00640      IF (ITYPE.EQ.1) PAGE(KHJ,I)=CHAR(N+12-J)
00650      GO TO 10020
00660 10030 CONTINUE
00670      IF (KHJ.GT.101) PAGE(101,I)=CHAR(19)
00680      IF (KHJ.LT.1) PAGE(1,I)=CHAR(18)
00690 10020 CONTINUE
00700 C----- EGENTLIG UDSKRIFT -----
00710      WRITE(6,90001)
00720      WRITE(6,90002)
00730      WRITE(6,90003)
00740      WRITE(6,90004) (TEKST(I),I=1,25)
00750      WRITE(6,90003)
00760      WRITE(6,90002)
00770      WRITE(6,90003)
00780      DO 10040 K=1,20
00790          RHJ1=RMAX-FLOAT(K-1)*(RMAX-RMIN)/20.0
00800          IHJ1=1+(K-1)*5
00810          WRITE(6,90007) RHJ1,(PAGE(IHJ1,I),I=1,101)
00820          DO 10040 L=1,4
00830              IHJ2=IHJ1+L
00840 10040      WRITE(6,90008) (PAGE(IHJ2,I),I=1,101)
00850          WRITE(6,90007) RMIN,(PAGE(101,I),I=1,101)
00860          WRITE(6,90003)
00870          DO 10050 I=1,11
00880 10050      IHJ3(I)=IO+(I-1)*10*II
00890          WRITE(6,90009) (IHJ3(I),I=1,11)
00900          WRITE(6,90003)
00910          WRITE(6,90005)
00920          WRITE(6,90003)
00930          WRITE(6,90002)
00940          IF (N.GT.4) RETURN
00950          WRITE(6,90010)
00960          DO 10070 I=1,101,10
00970              J=IO+(I-1)*II
00980 10070      WRITE(6,90011) J,(DATA(I,K),K=1,N)
00990          WRITE(6,90012)
01000          RETURN
01010 90001  FORMAT(1H1)
01020 90002  FORMAT(1H+,117(1H*),/)
01030 90003  FORMAT(1H+,1H*,T118,1H*,/)
01040 90004  FORMAT(1H+,1H*,T50,25A1,T118,1H*,/)
01050 90005  FORMAT(1H+,1H*,T95,'TIDSAKSE',T118,1H*,/)
01060 90007  FORMAT(1H+,2H*,F10.4,' ',101A1,T118,1H*,/)
01070 90008  FORMAT(1H+,1H*,T15,101A1,T118,1H*,/)
01080 90009  FORMAT(1H+,'*',T12,I4,10(' ',I4),T118,1H*,/)
01090 90010  FORMAT(1H+,'*',T118,'*'/ '*',15X,'T',17X,'*',22X,'+',21X,'X',
01100      > 21X,'-',T118,'*'/ '*')
01110 90011  FORMAT(1H+,T118,'*'/ '*',12X,I4,4(12X,F10.4))
01120 90012  FORMAT(1H+,T118,'*'/ '*',T118,'*'/ '*',117(1H*))
01130      END
01140 C
01150 C
01160 C
01170      REAL FUNCTION FILTER(I,A,N,B,M,U)
01180 C*****
01190 C
01200 C          FILTER MED LIGNINGEN
01210 C
01220 C          A Y = B U
01230 C
01240 C*****
01250 C
01260 C          VARIABELLISTE:
01270 C          =====
01280 C          I          ANVENDELSESNUMMER
01290 C          A          ARRAY(10) INDH. A - POLYNOMIET
01300 C          N          ORDEN AF A - POLYNOMIET
01310 C          B          ARRAY(10) INDH. ET TAE LLER POLYNOMIE
01320 C          M          ORDEN AF B - POLYNOMIET
01330 C          U          ET PAAVIRKNINGSSIGNAL
01340 C
01350 C+++++
01360      COMMON /FILDAT/X(6,2,17)
01370      DIMENSION A(6),B(6)
01380 C----- INDSAET PAAVIRKNING -----
01390      X(1,2,I)=U
```

```
01400 C----- FORETAG FILTRERINGEN -----
01410      Y=0.0
01420      IF (N.LT.1) GO TO 20
01430      DO 10 J=1,N
01440 10      Y=Y+A(J+1)*X(J,1,I)
01450 20      IM=M+1
01460      DO 30 J=1,IM
01470 30      Y=Y+B(J)*X(J,2,I)
01480      Y=Y/A(1)
01490 C----- OPDATER DATAVEKTORENE -----
01500      IF (N.LT.2) GO TO 50
01510      IN=N-1
01520      DO 40 J=1,IN
01530 40      X(N+1-J,1,I)=X(N-J,1,I)
01540 50      CONTINUE
01550      IF (M.LT.1) GO TO 70
01560      IM2=IM+1
01570      DO 60 J=1,M
01580 60      X(IM2-J,2,I)=X(IM-J,2,I)
01590 70      X(1,1,I)=-Y
01600      FILTER=Y
01610      RETURN
01620      END
01630 C
01640 C
01650      REAL FUNCTION RINTEG(IANV,ITRIN,U)
01660 C*****
01670 C
01680 C      DIGITAL INTEGRATOR MED ENDELIG HORISONT
01690 C
01700 C*****
01710 C
01720 C      VARIABELLISTE:
01730 C      =====
01740 C
01750 C      IANV   ANVENDELSES NUMMER
01760 C      ITRIN  SIMULATIONSTRIN
01770 C      U      INDGANGSSIGNAL
01780 C
01790 C+++++
01800      COMMON /LGDATA/D(8,50),Y(8)
01810      INDEX=MOD(ITRIN-1,50)+1
01820      Y(IANV)=Y(IANV)+U-D(IANV,INDEX)
01830      D(IANV,INDEX)=U
01840      IF (ITRIN.LT.51) GO TO 10
01850      RINTEG=Y(IANV)/50.0
01860      RETURN
01870 10      RINTEG=Y(IANV)/FLOAT(ITRIN)
01880      RETURN
01890      END
01900 C
01910 C
01920      SUBROUTINE RML(Y,U1,U2,U3,EPS,EPS2,IRR,GR,IGR,Q,CD,DELTA,TRP,PD)
01930 C*****
01940 C
01950 C      ESTIMATIONS ALGORITME
01960 C
01970 C      AY = B1 U1 + B2 U2 + B3 U3 + C E + D
01980 C
01990 C*****
02000 C
02010 C      VARIABELLISTE:
02020 C      =====
02030 C
02040 C      P      ARRAY (23,23) INDH. ESTIMATERNES KOVARIANS
02050 C      TRP     TR(P)
02060 C      PD     P(IRHO,IRHO)
02070 C      THETA  ARRAY (23)   INDH. ESTIMERET PARAMETERVEKTOR
02080 C      X      ARRAY (23)   INDH. MODELLENS TILSTANDE
02090 C      Z      ARRAY (23)   INDH. MODELLENS FILTREREDE TILSTANDE
02100 C      EPS    RESIDUALET
02110 C      EPS1   FORSINKET RESIDUAL
02120 C      EPS2   BEGRAENSET RESIDUAL
02130 C      IRR    STYREVARABEL, HVIS 1 FAAS ROBUST ESTIMATION
02140 C           HVIS 0 FAAS ALM. ESTIMATION
02150 C      GR     RUSEFUNKTION FOR ROBUSTHEDSGRAENSEN
02160 C      Q      ESTIMAT AF STOEJVARIANSEN
02170 C      IGR    HVIS 1 ER RESIDUALET BEGRAENSET
02180 C           HVIS 0 ER RESIDUALET MINDRE END DELTA
02190 C      CD     GRAENSE FOR RESIDUAL TIL GLR2
```

```

02200 C      DELTA  GRAENSE FOR RESIDUAL TIL ESTIMATION
02210 C      Y      UD GANGSSIGNAL ( FRA OBJEKTET )
02220 C      Y1     FORSINKET UD GANGSSIGNAL
02230 C      RL     FORGLEMMELSESFAKTOR
02240 C      RLO    POL I TIDSUDVIKLINGEN AF L
02250 C      RL1    ASYMTOTEVAERDI FOF L
02260 C      NRELS  STYREVARIABLE
02270 C           HVIS NEGATIV FAAS EN RLS ESTIMATION
02280 C           HVIS NUL      FAAS EN RELS ESTIMATION
02290 C           HVIS POSITIV FAAS EN RML ESTIMATION
02300 C      U1     ET PAAVIRKNINGSSIGNAL
02310 C
02320 C+++++
02330 C
02340 C      NOEDVENDIGE UNDERPROGRAMMER: FILTER
02350 C      =====
02360 C
02370 C+++++
02380      COMMON /RMLDAT/P(23,23),THETA(23),X(23),Z(23),EPS1,Y1,RL,RLO,
02390 >RL1,NRELS,N1(5),IRHO,NF,NG,NH,NCC,QW(23,23)
02400      REAL C(6)/1.00,5*0.00/
02410      REAL R(23)
02420      REAL K(23)
02430      INTEGER ISPIDS/0/
02440      REAL ET(6)/1.00,5*0.00/
02450      IRHO1=IRHO-1
02460      IRHO2=IRHO-2
02470 C----- OPDATERING AF X - VEKTOREN -----
02480      DO 10 I=1,IRHO2
02490 10      X(IRHO-I)=X(IRHO1-I)
02500      IF (N1(1).GT.0) X(N1(1))=-Y1
02510      X(N1(2))=U1
02520      X(N1(3))=U2
02530      X(N1(4))=U3
02540      IF (NRELS.GT.-1) X(N1(5))=EPS1
02550 C----- OPDATERING AF Z - VEKTOREN -----
02560      IF (NRELS) 20,20,30
02570 C - - - - - RLS, RELS TILFAELDET (KOPIERING) - - - - -
02580 20      DO 25 I=1,IRHO
02590 25      Z(I)=X(I)
02600      GO TO 35
02610 C - - - - - RML TILFAELDET (FILTRERING) - - - - -
02620 30      CONTINUE
02630      IF (NCC.EQ.0) GO TO 20
02640      DO 31 I=1,NCC
02650 31      C(I+1)=THETA(N1(5)-1+I)
02660      DO 32 I=1,IRHO2
02670 32      Z(IRHO-I)=Z(IRHO1-I)
02680      IF (N1(1).GT.0) Z(N1(1))=FILTER(10,C,NCC,ET,0,-Y1)
02690      Z(N1(2))=FILTER(11,C,NCC,ET,0,U1)
02700      Z(N1(3))=FILTER(12,C,NCC,ET,0,U2)
02710      Z(N1(4))=FILTER(13,C,NCC,ET,0,U3)
02720      Z(N1(5))=FILTER(14,C,NCC,ET,0,EPS1)
02730      Z(IRHO)=FILTER(15,C,NCC,ET,0,1.0)
02740 C----- OPDATER FORGLEMMELSESFAKTOREN -----
02750 35      RL=RLO*RL+RL1*(1.00-RLO)
02760 C----- BEREGNING AF TR(P) OG P(IRHO) -----
02770      PD=P(IRHO,IRHO)
02780      TRP=0.0
02790      DO 100 I=1,IRHO1
02800 100     TRP=TRP+P(I,I)
02810 C----- BEREGNING AF RESIDUALET -----
02820      EPS=Y
02830      DO 80 I=1,IRHO
02840 80      EPS=EPS-THETA(I)*X(I)
02850 C----- ROBUSTHEDSGRAENSER -----
02860      CD=0.0
02870      DELTA=0.0
02880      IGR=0
02890      EPS2=EPS
02900      E=EPS
02910      IF (IRR.NE.1) GO TO 110
02920      GR=AMAX1(GR-1.0,0.0)
02930      SPR=SQRT(Q)
02940      DELTA=GR+2.0*SPR
02950      CD=2.0*SPR
02960      IF (ABS(EPS).GT.CD) EPS2=SIGN(CD,EPS)
02970      IF (ABS(EPS).GT.DELTA) GO TO 140
02980      ISPIDS=0
02990      GO TO 110

```

```
03000 140 IGR=1
03010 ISPIDS=ISPIDS+1
03020 EPS=SIGN(DELTA, EPS)
03030 110 CONTINUE
03040 C WRITE(6, 6060) IGR, Q, SPR, EPS, E, EPS2, CD, DELTA
03050 C6060 FORMAT(1H-, 117(1H-), ///, ' IGR, Q, SPR, EPS, E, EPS2, DC, DELTA',
03060 C > I3, 3X, 7(3X, F8.4))
03070 C----- BEREKNINGEN AF HJAE LPE STOERRE LSEN R -----
03080 DO 40 IR=1, IRHO
03090 R(IR)=0
03100 DO 40 IS=1, IRHO
03110 40 R(IR)=R(IR)+P(IR, IS)*Z(IS)
03120 C----- BEREKNING AF HJAE LPE STOERRE LSEN MY -----
03130 RMY=RL
03140 DO 50 I=1, IRHO
03150 50 RMY=RMY+Z(I)*R(I)
03160 C----- BEREKNING AF FORSTAERKNING -----
03170 DO 60 IR=1, IRHO
03180 60 K(IR)=R(IR)/RMY
03190 C----- OPDATERING AF P MATRICEN -----
03200 IF (IGR.EQ.1) GO TO 120
03210 DO 70 IR=1, IRHO
03220 DO 70 IS=1, IRHO
03230 70 P(IR, IS)=(P(IR, IS)+QW(IR, IS)-K(IR)*R(IS))/RL
03240 C----- UDGLATNING AF P -----
03250 DO 130 IR=1, IRHO
03260 DO 130 IS=1, IR
03270 P(IR, IS)=0.5*(P(IR, IS)+P(IS, IR))
03280 P(IS, IR)=P(IR, IS)
03290 130 CONTINUE
03300 C----- OPDATERING AF PARAMETERVEKTOREN -----
03310 120 CONTINUE
03320 IF ((IGR.EQ.1).AND.(ISPIDS.LT.4)) GO TO 150
03330 DO 90 I=1, IRHO
03340 90 THETA(I)=THETA(I)+K(I)*EPS
03350 150 CONTINUE
03360 C----- GEMNING AF EPS OG Y -----
03370 Y1=Y
03380 EPS1=EPS
03390 RETURN
03400 END
03410 C
03420 C
03430 C
03440 C
03450 SUBROUTINE GLR2(J, N, ALFAD, EPS, QE, RMLGR, Y1, Y2, Y3, Y4, IST, IR, THRESH)
03460 C*****
03470 C
03480 C DETECTOR - VERSION 2
03490 C
03500 C*****
03510 C
03520 C VARIABELLISTE :
03530 C =====
03540 C
03550 C J SIMULATIONSTRIN ( T+1 )
03560 C N DETECTOR HORISONT
03570 C ALFAD BEGYNDELSESVAERDI FOR ESTIMERET VARIANS
03580 C AF ESTIMATET PAA D.
03590 C EPS RESIDUALET ( FRA RML )
03600 C Y1 SIGNAL TIL RML - ROUTINEN
03610 C IST STYREVARIABEL HVIS NUL UDFRES KUN ET ESTIMAT AF Q
03620 C QE ESTIMAT AF STOEJVARIANS
03630 C RMLGR RUSEFUNKTIONENS BEGYNDELSESVAERDI EFTER ET SPRING
03640 C IR STYREVARIABEL TIL RML - ROUTINEN ( ROBUSTHED )
03650 C THRESH THRESHOLD VALUE
03660 C
03670 C*****
03680 C
03690 C
03700 C NOEDVENDIGE PROCEDURER: RML
03710 C =====
03720 C
03730 C*****
03740 COMMON /RMLDAT/P(23, 23), THETA(23), X(23), Z(23), EPS1, YY, RL,
03750 >RLO, RL1, NRELS, N1(5), IRHO, NF1, NG1, NH1, NCC, QW(23, 23)
03760 REAL DATA(20)/20*0.0/, GR/10.0/, RTT/0.00/
03770 REAL XX(13, 10)/130*0.0/, YX(10)/10*0.0/
03780 REAL RLQ/0.8/, SUM/0.0/
03790 REAL PP(13, 13, 10), TH(13, 10), SIGN(4, 10)/40*0.0/
```

```
03800      INTEGER ITT/0/
03810      INDEX=MOD(J,N)+1
03820      SUM=SUM+EPS-DATA(INDEX)
03830      IF (J.LT.N+1) DATA(INDEX)=EPS
03840      IF (J.EQ.N+1) ITT=N+1
03850      SIGN(1,INDEX)=Y1
03860      SIGN(2,INDEX)=Y2
03870      SIGN(3,INDEX)=Y3
03880      SIGN(4,INDEX)=Y4
03890      ITT=MAX(0,ITT-1)
03900      IF (ITT.NE.0) GO TO 50
03910      RTT=RTT+1.00
03920      RLQ=0.97*RLQ+0.03*0.985
03930      QE=RLQ*QE*(1.00-1.00/RTT)+(DATA(INDEX)**2)*(1.00/RTT+
03940      >      1.00-RLQ)
03950 50    CONTINUE
03960      DATA(INDEX)=EPS
03970      GR=0.75*GR+0.25*3.39
03980      THRESH=GR*SQRT(FLOAT(N)*QE)
03990      IF (IST.EQ.0) RETURN
04000 C      WRITE(6,6060) J,INDEX,EPS,SUM,GR,QE,THRESH
04010 C6060  FORMAT(1H-, 'J,INDEX,EPS,SUM,GR,QE,TH',5X,2(I3,3X),5X,5(F8.4,3X))
04020      IF (ABS(SUM).GT.THRESH) GO TO 10
04030 C----- NORMALE TILFAELDE -----
04040      YX(INDEX)=YY
04050      DO 20 I=1,IRHO
04060      TH(I,INDEX)=THETA(I)
04070      XX(I,INDEX)=X(I)
04080      DO 20 K=1,IRHO
04090 20    PP(I,K,INDEX)=P(I,K)
04100      RETURN
04110 C----- SPRING DETEKTERET -----
04120 10    CONTINUE
04130      SUM=0.00
04140      ITT=N+1
04150      WRITE(6,1010) J
04160 1010  FORMAT(1H-,50(1H*)////, ' * * * S P R I N G * * * ',//, ' ',
04170      >      'PAA SIMULATIONSTRIN ',I5, ' DETECTET I GLR2',////,50(1H*))
04180      YY=YX(INDEX)
04190      DO 30 K=1,IRHO
04200      THETA(K)=TH(K,INDEX)
04210      X(K)=XX(K,INDEX)
04220      DO 30 L=1,IRHO
04230 30    P(K,L)=PP(K,L,INDEX)
04240      I=J-N+1
04250      DO 40 K=I,J
04260      INDEX=MOD(K,N)+1
04270      P(IRHO,IRHO)=ALFAD
04280      Q1=0.00
04290      RMLGR=10.0
04300      CALL RML(SIGN(1,INDEX),SIGN(2,INDEX),SIGN(3,INDEX),
04310      >      SIGN(4,INDEX),EPS,EPS2,IR,RMLGR,II,Q1,CD,CE,TRP,PD)
04320      DATA(INDEX)=0.00
04330 C      WRITE(6,6061) K,INDEX,EPS,SUM
04340 C6061  FORMAT(1H-,T60, 'K,INDEX,EPS,SUM',5X,2(I3,3X),5X,2(F8.4,3X))
04350      DO 40 K1=1,IRHO
04360      TH(K1,INDEX)=THETA(K1)
04370      DO 40 K2=1,IRHO
04380      PP(K1,K2,INDEX)=P(K1,K2)
04390 40    CONTINUE
04400      RMLGR=5.00
04410      RETURN
04420      END
04430 C
04440 C
04450      SUBROUTINE DELAY(K,XX,Y,U,W,U1)
04460 C*****
04470 C
04480 C      DELAY
04490 C
04500 C*****
04510      DIMENSION XX(6,4)
04520      IF (K.LT.2) GO TO 20
04530 C----- REOTER XX -----
04540      IK=K-1
04550      DO 10 I=1,IK
04560      DO 10 J=1,4
04570 10    XX(K+1-I,J)=XX(K-I,J)
04580 C----- INDSAET SIGNALER -----
04590 20    XX(1,1)=Y
```

```
04600      XX(1,2)=U
04610      XX(1,3)=W
04620      XX(1,4)=U1
04630      RETURN
04640      END
04650 C
04660 C
04670 C
04680 C
04690 C*****
04700      SUBROUTINE MARKOV(ITRIN,DSPR,PROB,D)
04710 C*****
04720 C
04730 C          PROCEDURE TIL SIMULATION AF EN MARKOV - PROCES
04740 C
04750 C+++++
04760 C
04770 C          VARIABELISTE :
04780 C          =====
04790 C
04800 C          ITRIN          SIMULATIONSTRINNET ( T+1 )
04810 C          D              UD GANGEN AF PROCESSEN
04820 C          PROB           OVERGANGSSANDSYNLIGHEDEN
04830 C          DSPR           SPREDNINGEN PAA SPRINGET
04840 C
04850 C+++++
04860      INTEGER IVEC(3)/7913,17,90846371/
04870      CALL RANDXR(X,1,IVEC)
04880      IF (X.GT.PROB) RETURN
04890      CALL RANDXR(X1,1,IVEC)
04900      CALL RANDXR(X2,1,IVEC)
04910      PHI=3.141592654
04920      Y=DSPR*SQRT(-2.0*ALOG(X1))*COS(2.0*PHI*X2)
04930      D=D+Y
04940      WRITE(6,1000) ITRIN,Y,DSPR,D
04950 1000  FORMAT(1H-,90(1H+),//, ' SPRING FRA MARKOV PAA TRIN ',I3,3X,
04960      >      ' AENDRING ',F8.4, ' MED SPREDNING ',F8.4, ' TIL ',F8.4,
04970      >      //, ' ',90(1H+))
04980      RETURN
04990      END
05000 C
05010 C*****
05020      SUBROUTINE STOEJ(ITRIN,SPR,SPR1,IPERT,DELTA,E)
05030 C*****
05040 C
05050 C          PROCEDURE TIL SIMULATION AF N STOEJPAAVIRKNING
05060 C
05070 C          P = (1-DELTA)*N(0,SPR)+DELTA*
05080 C
05090 C          HVOR PO = N(1,SPR1) FOR IPERT=1
05100 C+++++
05110 C
05120 C          VARIABELISTE :
05130 C          =====
05140 C
05150 C          ITRIN          SIMULATIONSTRINNET ( T+1 )
05160 C          E              UD GANGEN AF PROCESSEN
05170 C          DELTA          OVERGANGSSANDSYNLIGHEDEN
05180 C          SPR            SPREDNINGEN
05190 C          SPR1           SPREDNINGEN PAA DEN UKENDTE FORDELING
05200 C          IPERT         STYREVARIABEL
05210 C
05220 C+++++
05230 C          INTEGER IVEC(3)/1397,30117133,90846371/
05240      INTEGER IVEC(3)/7913,17,90846371/
05250      CALL RANDXR(X1,1,IVEC)
05260      CALL RANDXR(X2,1,IVEC)
05270      PHI=3.141592654
05280      E=SPR*SQRT(-2.0*ALOG(X1))*COS(2.0*PHI*X2)
05290      CALL RANDXR(X,1,IVEC)
05300      IF (IPERT.EQ.0) RETURN
05310      IF (X.GT.DELTA) RETURN
05320      CALL RANDXR(X1,1,IVEC)
05330      CALL RANDXR(X2,1,IVEC)
05340      E=SPR1*SQRT(-2.0*ALOG(X1))*COS(2.0*PHI*X2)
05350      WRITE(6,1000) ITRIN,E,SPR1
05360 1000  FORMAT(1H-,90(1H-),//, ' SPIDS FRA STOEJ P$ TRIN : ',I3,3X,
05370      >      ' E = ',F8.4, ' SPREDNING = ',F8.4,
05380      >      //, ' ',90(1H-))
05390      RETURN
```

```

05400      END
05410 C
05420 C
05430 C
05440      REAL FUNCTION REF(I,NR)
05450 C*****
05460 C
05470 C      PROCEDURE TIL GENERERING AF ET REFERENCE SIGNAL
05480 C
05490 C*****
05500      IF (NR) 10,20,30
05510 10      REF=0.00
05520      RETURN
05530 20      REF=1.00
05540      RETURN
05550 30      REF=1.00-FLOAT(MOD((I-1)/25,2))*2.00
05560      RETURN
05570      END
05580 C
05590 C
05600      REAL FUNCTION SALOSS(A1,B1,N)
05610 C*****
05620 C
05630 C      PROCEDURE TIL UDREGNING AF INTEGRALET AF FUNKTIONEN:
05640 C
05650 C          1/(2*PI*I)*B(Z)*B(1/Z)/(A(Z)*A(1/Z)*Z)
05660 C
05670 C      RUNDT LANGS ENHEDSCIRKLEN. HVIS A IKKE HAR ALLE ROEDDER INDENFOR
05680 C      ENHEDSCIRKLEN RETURNERES VAERDIEN -10.00
05690 C
05700 C*****
05710 C
05720 C      VARIABELLISTE:
05730 C      =====
05740 C
05750 C      N          ORDEN AF SYSTEMM
05760 C      A          ARRAY INDH. NAEVNERPOLYNOMIET
05770 C      B          ARRAY INDH. TAELLERPOLYNOMIET
05780 C
05790 C+++++
05800      DIMENSION A(6),A1(6),B(6),B1(6),AS(7)
05810      AO=A1(1)
05820      IN=N+1
05830      DO 60 I=1,IN
05840          A(I)=A1(I)
05850 60      B(I)=B1(I)
05860      IF (N.EQ.0) GO TO 70
05870      V=0.00
05880      DO 10 K=1,N
05890          L=N+1-K
05900          L1=L+1
05910          ALFA=A(L1)/A(1)
05920          BETA=B(L1)/A(1)
05930          V=V+BETA*B(L1)
05940          DO 20 I=1,L
05950              M=L+2-I
05960              AS(I)=A(I)-ALFA*A(M)
05970 20      B(I)= B(I)-BETA*A(M)
05980          IF (AS(1)) 50,50,30
05990 30      DO 40 I=1,L
06000 40      A(I)=AS(I)
06010 10      CONTINUE
06020      V=V+B(1)**2/A(1)
06030      SALOSS=V/AO
06040      RETURN
06050 50      SALOSS=-10.00
06060      RETURN
06070 70      SALOSS=(B(1)**2)/(A(1)**2)
06080      RETURN
06090      END
06100 C
06110 C
06120      SUBROUTINE ADD(A,N,B,M,C,NC)
06130 C*****
06140 C
06150 C          PROCEDURE TIL ADDITION AF
06160 C          TO POLYNOMIET
06170 C
06180 C*****
06190 C

```



```
06200 C      VARIABELLISTE:
06210 C      =====
06220 C
06230 C      A      ARRAY INDH. ET POLYNOMIE, A0, A1, A2,      AN
06240 C      B      ARRAY INDH. ET POLYNOMIE, B0, B1, B2,      BM
06250 C      N      ORDENEN AF A
06260 C      M      ORDENEN AF B
06270 C      C      ARRAY INDH. DET RESULTERENDE POLYNOMIE
06280 C      NC     ORDEN AF C POLYNOMIET
06290 C
06300 C+++++
06310      DIMENSION A(6),B(6),C(6)
06320      DO 20 J=1,6
06330 20      C(J)=0.00
06340      NC=MAXO(N,M)
06350      INC=NC+1
06360      DO 10 I=1,INC
06370 10      C(I)=A(I)+B(I)
06380      RETURN
06390      END
06400 C
06410 C
06420      SUBROUTINE MULT(A,N,B,M,C,NC)
06430 C*****
06440 C
06450 C      PROCEDURE TIL MULTIPLIKATION AF TO
06460 C      POLYNOMIER A OG B MED ORDENEN N OG
06470 C      M. C ER DET RESULTERENDE POLYNOMIE
06480 C      MED ORDEN NC.BEMAERK AT ALLE UBEENYTT -
06490 C      TEDE PLADSER I ARRAYS SKAL VAERE
06500 C      NUL
06510 C
06520 C*****
06530 C
06540 C      VARIABELLISTE:
06550 C      =====
06560 C
06570 C      A      ARRAY(6) INDH. POLYNOMIE DVS: A0, A1, ... , AN
06580 C      B      ARRAY(6) INDH. POLYNOMIE DVS: B0, B1, ... , BM
06590 C      C      ARRAY(6) INDH. POLYNOMIE DVS: C0, C1, ... , CNC
06600 C
06610 C+++++
06620      DIMENSION A(6),B(6),C(6)
06630      INC=N+M+1
06640      DO 20 J=1,6
06650 20      C(J)=0.00
06660      DO 10 J=1,INC
06670      IH=MINO(N+1,J)
06680      L=MAXO(1,J-M)
06690      DO 10 I=L,IH
06700 10      C(J)=C(J)+A(I)*B(J-I+1)
06710      NC=N+M
06720      RETURN
06730      END
06740 C
06750 C
06760      SUBROUTINE PREDIK(A,NA,C,NC,K,E,F,NF)
06770 C*****
06780 C
06790 C      PROCEDURE TIL LOESNING AF
06800 C
06810 C      C = A E + 1/(Q**K)F
06820 C
06830 C*****
06840      DIMENSION A(6),C(6),E(6),F(6)
06850 C----- E - PARAMETERNE -----
06860      E(1)=1.00
06870      IF (K.LT.2) GO TO 20
06880      DO 10 J=2,K
06890      IH=MINO(NA,J-1)
06900      E(J)=C(J)
06910      DO 10 I=1,IH
06920 10      E(J)=E(J)-A(I+1)*E(J-I)
06930 20      CONTINUE
06940 C----- F - PARAMETERNE -----
06950      NF=MAXO(NC-K,NA-1)
06960      INF=NF+1
06970      DO 30 J=1,INF
06980      F(J)=C(K+J)
06990      L=J
```

```
06990      L=J
07000      IH=MINO(NA,J+K-1)
07010      DO 30 I=L,IH
07020 30    F(J)=F(J)-A(I+1)*E(K+J-I)
07030      RETURN
07040      END
07050 C
07060 C
END OF DATA
```

L I T T E R A T U R L I S T E

- /1/ A. Y. Allidina, F. M. Huges (1980): Generalised self-tuning controller with pole assignment.
IEE PROC. Vol 127, jan 1980 s. 13
- /2/ D. W. Clarke, P. J. Gawthrop (1975): Self - tuning controller.
IEE PROC. Vol 122, sept. 1975 s. 929
- /3/ D. W. Clarke, P. J. Gawthrop (1980): Implementation and Applications of Microprocesser - based Self - Tuners.
Automatica Vol. 17, 1981 s. 233
- /4/ K. Condradsen: En Introduktion til Statistik.
IMSOR
- /5/ B. Egardt (1978): Stability of Adaptive Controllers
Lecture Notes in Control and Information Sciences.
Springer
- /6/ P. J. Gawthrop (1977): Some interpretations of the self - tuning controller.
IEE PROC. Vol 124, okt. 1977 s. 889
- /7/ J. Holst (1977): Adaptive prediction and recursive estimation. Report 1013, Dept. of Automatic Control, Lund Institute of Technology, Lund, Sweden
- /8/ J. Holst (1979): Noter til kursus 0414, Stokastiske systemer, identifikation og regulering. IMSOR.
- /9/ R. Isermann (1977): Digitale Regelsysteme.
Springer.

- /10/ H. Kurz, R. Isermann, R. Schumann (1979): Experi -
mental Comparison and Application of Various Parameter-
Adaptive Control Algorithms.
Automatica Vol. 16, s. 117
- /11/ V. Panuska (1980): A New Form of Extended Kalman
Filter for Parameter Estimation in Linear Systems
with Correlated Noise.
IEEE, Vol. AC - 25, apr. 1980 s.229
- /12/ B. T. Poljak, J. Z. Tsytkin (1978): Robust Identi -
fication.
Automatica Vol. 16, s. 53.
- /13/ R. V. Sonalkar, C. N. Shen (1974): Rapid Estimation
and Detection Scheme for Unknown Discretized Rectangu-
lar Inputs
IEEE, Vol. AC - , feb. 1975 s. 142
- /14/ P. Stoica, J. Holst, T. Söderström (1981): Eigen -
value Location of Certain Matrices arising in
Convergence Analysis Problems.
- /15/ T. Söderström, L. Ljung, I. Gustavson (1974):
A comparative study of recursive identification
methods.
Report 7427, Dept of Automatic Control, Lund, Sweden.
- /16/ T. Söderström, L. Ljung, I. Gustavson (1976):
A theroretical analysis of recursive identification
methods.
Automatica Vol. 14, s. 231.
- /17/ A. Willsky, H. Jones (1974): A generalized likeli -
hood ratio approach to the detection and estimation
of jumps in linear systems.
IEEE AC , feb. 1976.

- /18/ A. Willsky (1975): A survey of design methods for failure detection in dynamic systems.
Automatica Vol. 12, 1976, s. 601.
- /19/ B. Wittenmark, K. Åstrøm (1980): Simple self - tuning controllers.
Methods and applications in adaptive control.
Lecture notes in control and information sciences.
- /20/ B. Wittenmark (1975): Stochastic adaptive control methods: a survey.
Int. J. Control, 1975, Vol. 21, s. 705.
- /21/ K. Åstrøm (1970): Introduction to stochastic control theory.
Academic press.
- /22/ K. Åstrøm (1980): Design principles for self - tuning regulators.
Methods and applications in adaptive control.
Lecture notes in control and information sciences.
- /23/ K. Åstrøm, B. Wittenmark (1979): Self - tuning controllers based on pole - zero placement.
IEE PROC. Vol 127, maj 1980 s. 120.
- /24/ K. Åstrøm, B. Wittenmark (1972): On Self Tuning Regulators.
Automatica Vol. 9, s 185