CODEN:LUTFD2/(TFRT-5296)/1-93/(1983)

SIMULERING OCH FELDETEKTERING I ETT VENTILSERVO FÖR TRYCKREGLERING AV EN KOKARREAKTOR

PER PERSSON

INSTITUTIONEN FÖR REGLERTEKNIK
LUNDS TEKNISKA HÖGSKOLA

APRIL 1983

LUND INSTITUTE OF TECHNOLOGY
DEPARTMENT OF AUTOMATIC CONTROL
Box 725
S 220 07 Lund 7 Sweden

Document name
Report

Date of issue
April 1983

Document number
CODEN:LUTFD2/(TFRT-5296)/1-93/(1983)

Author(s)
Per Persson

Supervisor
Sten Bergman

Sponsoring organization

Title and subtitle
(Simulering och feldetektering i ett ventilservo för tryckreglering av en kokarreaktor.) Simulation and fault-detection of a pressure control servosystem in a Boiling Water Reactor.

Abstract

This master thesis describes a Simnon model of a boiling water reactor to be used in simulating faults and disturbances. These faults and disturbanses will be detected by noise analysis. Some methods in identification and noise analysis are also described and are applied on some malfunctions of a servo. A Pascal program for recursive parameter identification was also written and tested. This program is to be used in an expert system for noise analysis on the nuclear power plant Barsebäck.

Key words

Classification system and/or index terms (if any)

Supplementary bibliographical information

ISSN and key title

ISBN

Language
Swedish

Number of pages
93

Security classification

Recipient's notes

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

Simulering och feldetektering i ett ventilservo för tryckreglering av en kokarreaktor.

Per Persson

Institutionen för Reglerteknik
Tekniska Högskolan i Lund

April 1983

Handledare: Sten Bergman

INNEHÅLL

1 INLEDNING

Examensarbetet är en del i ett forskningsprojekt rörande brusanalys av ett kärnkraftverk. Man vill med studier av bruset i mätsignalerna kunna detektera olika felfunktioner, såväl plötsligt uppkomna som långsamma 'feltrender'. Med hjälp av ett expertsystem skall sedan operatören i kraftverket kunna få diagnosförslag då något händer. Syftet med examensarbetet var att med en tidigare skriven Simnonmodell av en BWR (se [9]) simulera tänkbara felfunktioner och utprova metoder för detektering av dessa. Vissa kompletteringar och korrektioner av den ursprungliga modellen har också gjorts.
Abetet har bedrivits i tre etapper, nämligen;

1. Hela reaktormodellen provades ut. En del nya delmodeller infördes och en vissa fysikaliska parametrar behövde korrigeras.

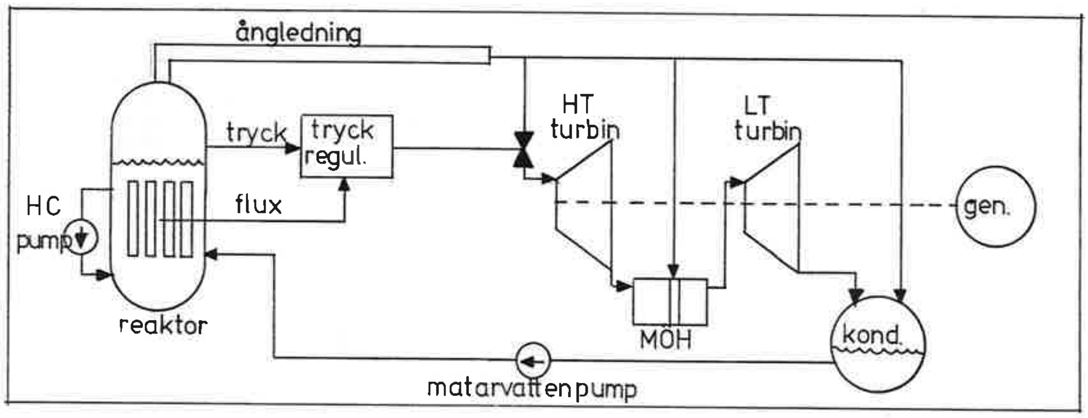
2. För att utprova några olika feldetekteringsmetoder har EHW-servot studerats närmare. Där har felfunktioner som glapp, friktion och ändrad dynamik införts. Med hjälp av Idpac har vi sedan försökt att detektera dessa felfunktioner.

3. Då hela brusanalys systemet är ämnat att fungera online i en BWR (Barsebäck) har ett Pascalprogram för rekursiv on-line identifiering skrivits och testats.

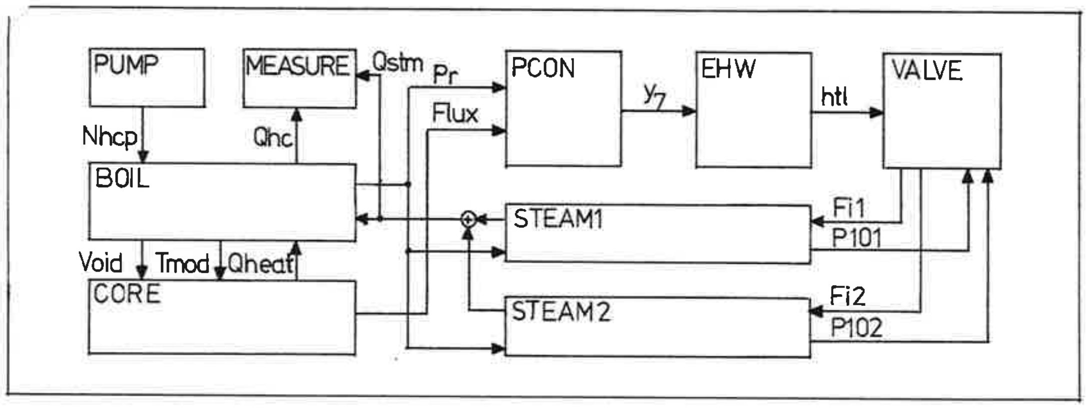
Jag vill inledningsvis också tacka civilingenjör Sten Bergman för sakkunnig handledning under examensarbetet.

2 BESKRIVNING AV EN BWR OCH SIMNON MODELLERNA

För simulerigen av en BWR har programpaketet Simnon (se Appendix 1) använts. Programpaketet, som är utvecklat vid Institutionen för Reglerteknik vid LTH, löser de ordinära differentialekvationer som används för modelleringen av reaktorn, och sköter utmatningen av resultaten. Den förenklade modell av reaktorn som här har använts har innehållit delmodeller av reaktorhärden, ånggenereringen, ångledningarna, ventiler och ventilservon samt en regulator. En schematisk bild av en BWR visas i fugur 1. Figur 2 visar ett blockschema av Simnonmodellen med dess signalflöden.



Figur 2.1 Schematisk bild av en BWR.



Figur 2.2 Simnonmodellens signalflöde.

Jag ger här en summarisk beskrivning av en BWRs funktion och beskriver samtidigt kort Simnonmodellen. Programkoden för Simnonmodellen återfinnes i Appendix 2.

Vid kärnreaktionen utvecklas värme i härden, vilken beskrives av systemet CORE. I härden kan neutronflödet och därmed effekten regleras med hjälp av styrstavarnas läge, vilket i modellen svarar mot en ändring av parametern Ross. Ross har vid simuleringen justerats så att härden ger effekten 1600 MW vid driftpunkt 100% och 330 MW vid driftpunkt 20%. I systemet CORE har även hänsyn tagits till termiska effekter i bränslet och Xenon förgiftning. Systemet CORE ger den i härden producerade effekten, Qheat, som utsignal. Qheat är i sin tur insignal till systemet BOIL vilket modellerar kokningen i reaktordomen. Systemet beskriver ångtrycksdynamiken och ångkvalitetsdynamiken i domen. Ångtrycket i reaktorn skall vara 70 bar. Detta regleras med hjälp av en PI-regulator i systemet PCON, vilken styr ånguttaget och därned domtrycket. Domtrycket och neutronflödet i reaktorn mätes och kommer som insignaler till regulatorn. Neutronflödet framkopplas genom ett första ordningens filter och adderas till PI-länkens utsignal. PI-länkens insignal är domtrycket, som regulatorn skall hålla konstant lika med 70 bar. En ökning av neutronflödet ger en ökning i domtrycket med en tidskonstant på 7-10 s. Framkopplingen av neutronflödet ger därmed en bättre reglering av trycket i och med att man härmed kan 'förutse' en tryckökning med hjälp av neutronflödet. Från domen går ångan i 4 st 120 m långa ångledningar via ventiler till högtrycksturbinerna och efter återupphettning till lågtryckturbinerna. Efter turbinerna kondenseras ångan och pumpas tillbaka till reaktorn. Utsignalen från regulatorn styr ett elektrohydrauliskt servo, systemet EHW, som påverkar ångledningsventilen och som därmed styr ångpådraget till turbinerna. I EHW-servot finns en olinjär länk i form av en kamskiva för att lineärisera ånguttaget som funktion av PI-regulatorns utsignal. Om man antar att servot kan vara en felkälla kan man tänka sig åtminstone tre felfunktioner, nämligen;

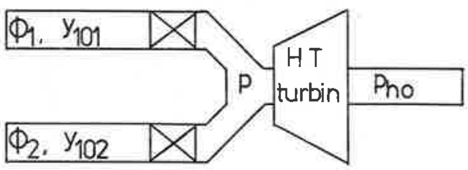
1. Glapp
2. Ökad friktion
3. Ändringar i dynamiken

Dessa tre felfunktioner är implementerade i modellen EHW. Glapp och friktion representeras av olika avsnitt i koden, ändrad dynamik modelleras genom att ändra tidskonstanterna. Ångledningarna, som för ångan från domen till turbinerna, beskrives av systemen STEAM1 och STEAM2. Dessa modeller är diskretiseringar av Bernoullis strömningsekvation. Denna är i sin kontinuerliga form en partiell differentialekvation och blir i diskretiserad form ett system av ordinära differentialekvationer. I vårt fall har fem nodpunkter i ångledningarna ansetts vara tillräckligt. Ekvationerna blir

$$\frac{d\dot{m}_1}{dt} = (p_{i-1} - p_1 - c_2 \cdot \dot{m}_i \cdot |\dot{m}_i|)/T_2 \qquad (1)$$

$$\frac{dp_1}{dt} = (\dot{m}_i - \dot{m}_{i-1})/T_1 \qquad (2)$$

p_i är trycket i nodpunkt i och $\dot{m}_i$ är massflödet i samma punkt. Parametrarna T_1, T_2 och c_2 bestämmes av rörets och det strömmande mediets egenskaper. Denna modell av ångledningarna ger ett tryckfall proportionell mot kvadraten på flödet. Med konstanten c2 kan man anpassa modellen så att man erhåller ett tryckfall som överensstämmer med det experimentellt uppmätta. I modellen har vi för enkelhets skull slagit ihop ledningarna två och två. I stället för de ursprungliga fyra ångledningarna har man i modellen fått två. EHW-servot påverkar två ventiler, och en möjlig felfunktion är att ventilerna påverkas olika av servot. En annan möjlig felfunktion är att de båda rören ej har samma tryck strax innan ventilen, detta kan uppstå genom t.ex. läckage. De båda ångledningarna har gemensam utgång och kommer därmed att växelverka. Balans för tryck och flöde ger, med antagandet att tryckfallet över en ventil är proportionellt mot kvadraten på flödet och att tryckfallet över en turbin är proportionellt mot flödet.



$$p - p_{ho} = K_{ht} \cdot (\Phi_1 + \Phi_2) \tag{3}$$

$$\Phi_1 = A_1 \cdot \sqrt{\frac{y_{101} - p}{k_{htv}}} \tag{4}$$

$$\Phi_2 = A_2 \cdot \sqrt{\frac{y_{102} - p}{k_{htv}}} \tag{5}$$

där variablerna har följande betydelse

p	trycket före högtrycksturbinen
p_{ho}	trycket efter högtrycksturbinen
A_1	öppningsarea för ventil 1
A_2	öppningsarea för ventil 2
k_{htv}	tryckfallskonstant för ventilerna
k_{ht}	tryckfallskonstant för högtrycksturbinen
y_{101}	trycket före ventil 1
y_{102}	trycket före ventil 2
Φ_1	ångflöde genom ventil 1
Φ_2	ångflöde genom ventil 2

Detta ger ett olinjärt ekvationssystem för de båda flödena Φ_1 och Φ_2.

$$\begin{cases} \Phi_1^2 \cdot k_{htv} = A_1^2 \cdot (y_{101} - k_{ht} \cdot (\Phi_1 + \Phi_2) - p_{ho}) \\ \Phi_2^2 \cdot k_{htv} = A_2^2 \cdot (y_{102} - k_{ht} \cdot (\Phi_1 + \Phi_2) - p_{ho}) \end{cases} \qquad (6)$$

Φ_1 löses ut och den uppkomna fjärdegradsekvationen löses i Simnon programmet med Newton-Raphsons metod, enligt formeln

$$\Phi_1(n+1) = \Phi_1(n) - \frac{\Phi_1(n)^4 + A\Phi_1(n)^3 + B\Phi_1(n)^2 + C\Phi_1(n) + D}{4\Phi_1(n)^3 + 3A\Phi_1(n)^2 + 2B\Phi_1(n) + C} \qquad (7)$$

$$A = 2\,\beta_1 \qquad (8)$$

$$B = -2\alpha_1 + \beta_1 - \beta_1\beta_2^2 \qquad (9)$$

$$C = -2\,\alpha_1\beta_1 \qquad (10)$$

$$D = \beta_1\beta_2\alpha_1 - \beta_1^2\alpha_2 + \alpha_1^2 \qquad (11)$$

$$\alpha_i = (y_{10i} - p_{ho})A_i^2/K_{htv} \qquad i=1,2 \qquad (12)$$

$$\beta_i = K_{ht}A_i^2/K_{htv} \qquad i=1,2 \qquad (13)$$

vilket kräver tre till fyra iterationer. Beräkningen av ånguttaget från de båda ledningarna göres i systemet VALVE. Insignalerna till de båda ångledningsmodellerna blir då det uttagna ångflödet och domtrycket, som kommer från systemet BOIL, och utsignalerna det till turbinerna levererade ångflödet. Experimentellt har man funnit brusvariationer hos de mätta signalerna t.ex. i tryck och flödesmätningarna. Även varvtalet hos huvudcirkulationspumparna har en brusartad varvtalsstörning. För att simulera detta har vissa brussignaler lagts in. Detta göres i systemet PUMP, samt vid mätningar i PCON. I samtliga fall erhålles lämpligt brusutseende genom filtrering av vitt brus. I systemen PCON och MEASURE filtreras även mätsignalerna för att simulera sensorernas begränsade bandbredd.

För att underlätta hanteringen vid simulering av hela systemet har en mängd Simnonkommandon för kompilering, initialvärdessättning, lagring av variabler etc sammanförts till ett macro, PCD, som vid anrop utför alla underkommandon.

3 IDENTIFIERINGS OCH BRUSANALYS METODER

3.1 Inledning

Brusanalys av data från ett system kan utföras antingen on-line med någon rekursiv algoritm eller på data som samlats in under en viss tidsrymd. Vid on-line behandlingen får man kontinuerligt ny information om systemet, vid den andra metoden får man sin gamla information uppdaterad med något tidsintervall.

Det insamlade datamaterialet kan behandlas på många olika sätt. De mest uppenbara storheterna man vill beräkna är de 'vanliga' statistiska måtten medelvärde, varians, standardavvikelse etc. För att få mer detaljerad information kan man beräkna signalernas spektra och korrelationsfunktioner. Har man tillgång till både in och utsignal från något intressant system kan en modell av systemet identifieras.

De statistiska storheter, spektra och modeller som räknats fram skall sedan jämföras med lagrade referensvärden. Ur de detekterade avvikelsena skall sedan expertsystemet avgöra om dessa beror på någon felfunktion i systemet och i så fall ge en tänkbar diagnos. Reglerna som fordras för detta kan erhållas genom fysikaliska resonemang, simuleringar eller empiriskt.

Vid mina undersökningar har programpaketet Idpac använts. Idpac har kommandon för identifiering, statistiska beräkningar och frekvensanalys av insamlat datamaterial. Speciellt trevligt är att datafiler som genererats vid simuleringar i Simnon direkt kan användas för Idpacanalyser. I Appendix 1 finns en kort beskrivning av Idpac. Önskas mera detaljerade upplysningar se [6].

3.2 Metodbeskrivning

I 3.2.1-3 presenteras olika brusbehandlingsmetoder, vilka exemplifieras i kapitel 4 med studium av EHW-servot.

3.2.1 Statistiska mått

Beräkning av de statistiska måtten medelvärde, standardavvikelse och varians göres med formlerna,

$$m_y = \frac{1}{N} \Sigma y_i \tag{1}$$

$$\sigma_y^2 = \frac{1}{N} (\Sigma y_i^2 - N m_y) \tag{2}$$

Formlerna kan lätt anpassas för att användas rekursivt. Dessa mått ger naturligtvis en mycket grov bild av processen, men kan ofta ge värdefull information då de kombineras med andra analyser av processen.

3.2.2 Auto- och korskovarians

För att undersöka hur mycket två signaler är kopplade till varann kan deras korskovarians, R_{xy}, beräknas. I Idpac göres detta med formeln

$$R_{xy}(\tau) = \frac{1}{N} \sum_{i=\max(1,1-\tau)}^{\max(N-\tau,N)} (x_i - m_x)(y_{i+\tau} - m_y) \quad , \tau = -NOL,\ldots,NOL \tag{3}$$

korskorrelationen fås sedan som

$$r_{xy}(\tau) = \sqrt{\frac{R_{xy}(\tau)}{R_{xx}(0)\, R_{yy}(0)}} \tag{4}$$

Sättes $x(t)=y(t)$ i formlerna ovan får man R_{yy}, autokovariansen för y och r_{yy} autokorrelationen för y. Ett högt väde på r_{xy} indikerar ett starkt samband mellan x och y på tidsavståndet τ. Vid analysen kan man söka efter speciella kännetecken hos r_{xy} t.ex. τ för kraftigt markerade koorrelationsmaxima, storleken hos dessa maxima etc.

3.2.3 Spektrum

En mer detaljerad bild av signalerna fås genom att studera deras autospektra eller autokorrelationsfunktion (ett Fouriertransformpar). Vi har framför allt valt att studera signalernas spektra, vilket beräknas med formeln

$$\Phi(\omega) = \frac{T}{\pi} \left(R_{yy}(0) + 2\cdot\sum_{\tau=1}^{\infty} R_{yy}(\tau)\cdot\cos(\omega\tau T)\right) \quad (5)$$

där R_{yy} är autokovariansen för y(t). I Idpac är beräkningarna implementerade på ett något annorlunda sätt eftersom y(t) där är en ändlig mängd sampel. För att kunna jämföra två spektra behöver man kunna ange ett spektrums karakteristiska utseende. Exempel på möjliga jämförelsetal som också har fysikaliska tolkningar är

1. Lågfrekvens amplitud
2. Eventuella resonansers storlek och läge
3. Energiinnehåll i vissa frekvensintervall, beräknat som $\int\Phi^2 d\omega$, där Φ är spektraltätheten.

Algoritmer för detektering och klassificering av resonansers storlek finns föreslagna bl.a. i [4].

3.2.4 Identifiering

Då två signaler i ett system är kända kam en modell som releterar utsignalen till insignalen + vitt brus beräknas. Ur denna beräkning får man mycket information om systemet, men priset man får betala är ganska omfattande räkningar.

För våra identifieringar har maximum likelihood metoden anvvänts. Ur de givna signalerna u(t) och y(t) beräknas modellen

$$A(q^{-1})y(t) = B(q^{-1})u(t) + \lambda C(q^{-1})e(t) \quad (6)$$

där A,B och C är polynom i skiftoperatorn q ($q^{-1}y(1)=y(t-1)$) och λ är en konstant. Den erhållna modellen kan användas till mycket. Några exempel:

1.Frekvenssvaret för den diskreta överföringsfunktionen mellan u och y

$$H(\omega) = \frac{B(e^{i\omega T})}{A(e^{i\omega T})} \quad (7)$$

beräknas och studeras med avseende på resonanstoppar, faskurvor o.s.v. (jfr 3.2.1).

2. Residualerna

$$\varepsilon = \frac{A}{C} y - \frac{B}{C} u \tag{8}$$

kan beräknas och ur varians och spektrum av dessa kan modellens riktighet avgöras. Residualerna skall, om identifieringen lyckats, vara små. En dålig identifiering kan bero på att man valt fel modellordning eller att det finns någon olinjäritet i systemet. Är modellordningen känd kan en ökad residualvarians således bero på att det uppstått någon olinjäritet i systemet, t.ex. ett glapp.

3. Om referensmodellens skiftoperatorspolynom är

$$A_r(q^{-1}) = \Sigma a_{ri} q^{-1} \quad \text{(analogt för } B_r \text{ och } C_r\text{)} \tag{9}$$

och den identifierade modellens

$$A(q^{-1}) = \Sigma a_i q^{-1} \quad \text{(analogt för B och C)} \tag{10}$$

kan man beräkna parameteravståndet

$$d^2 = \sum_{i=1}^{n} \frac{(a_{ri} - a_i)^2}{\sigma_{ai}} + \frac{(b_{ri} - b_i)^2}{\sigma_{bi}} + \frac{(\lambda_r c_{ri} - \lambda c_i)^2}{\sigma_{ci}} \tag{11}$$

där σ_{ai}, σ_{bi} och σ_{ci} är standardavvikelserna för variationerna i parametrarna a_i, b_i och c_i. Om man antager att a_i och b_i är helt oberoende väljer man $\sigma_{ai}, \sigma_{bi}, \sigma_{ci} = 1$. Detta antagande har gjorts vid beräknande av parameteravstånd i kapitel 4. Parameteravståndet kan användas som ett mått på hur mycket modellerna avviker från referensmodellen.

3.3 Exempel

I detta kapitel kommer ett exempel med identifiering av ett andra ordningens system att demonsteras. Det kontinuerliga system som undersöktes var

$$G(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{14}$$

Systemet matades med en PRBS-signal vid fem olika driftspunkter, d.v.s. fem (ω_0, ζ) punkter. Till utsignalen adderades en brussignal $\in N(0,0.16)$, och ur insignalen och denna störda utsignal identifierades ett diskret system $H(q^{-1})$. Där

$$H(q^{-1}) = \frac{b_1 q^{-1} + b_2 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2}} \tag{15}$$

Ur koefficienterna i det diskreta systemet kan sedan ω_0 och ζ för det kontinuerliga systemet skattas med formlerna

$$\omega_0^* = \frac{1}{T_0}\left[\arccos^2\left(-\frac{a_1}{2\sqrt{a_2}}\right) + \frac{\ln^2(a_2)}{4}\right] \tag{16}$$

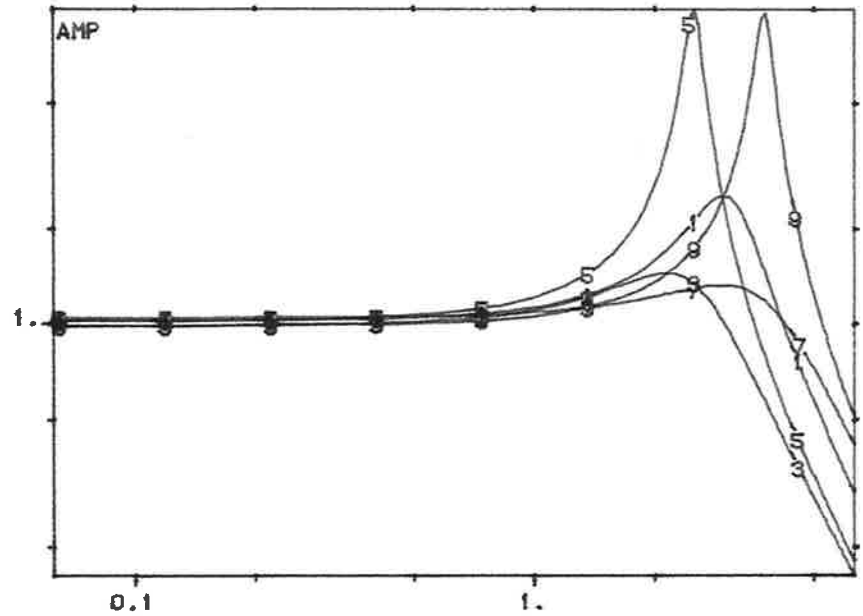
$$\zeta^* = -\frac{\ln(a_2)}{2\omega_0^* T_0} \tag{17}$$

Resultaten presenteras i tabell 3.1

Simulerat system		Skattat system	
ω_0	ζ	ω_0^*	ζ^*
3.140	0.200	3.105	0.203
2.510	0.400	2.502	0.363
2.510	0.050	2.513	0.052
3.770	0.400	3.751	0.429
3.770	0.050	3.771	0.051

Tabell 1.

Den diskreta överföringsfunktionens frekvenssvar i de olika driftspunkterna presenteras i figur 1



Figur 3.1 Överföringsfunktionens frekvenssvar i de fem simulerade driftspunkterna

Om systemet representerade en fysikalisk process kunde en felfunktion bestå i att parametrarna ω_0 och ζ ändrades.

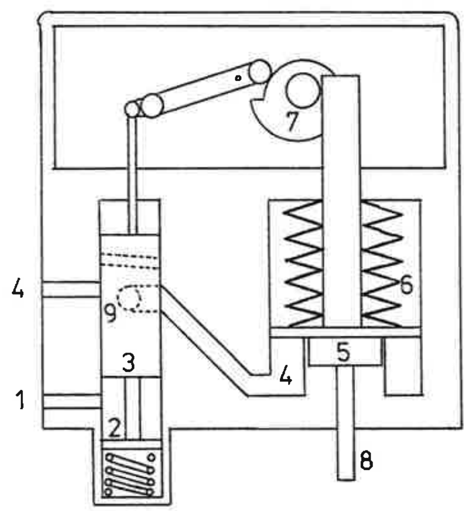
Detta kunde tänkas bero på ändrat tryck i hydraul eller pneumatikcylindrar, ändrad friktion e.dyl. Ur beräkningar liknande de här gjorda skulle man kunna ställa upp regler för vilka värden på ω_0 och ζ som kan accepteras utan att man behöver ge larm för felfunktion. Kunde man ej utföra transformationen $(a_1, a_2) \rightarrow (\omega_0, \zeta)$, skulle man kunna få en regel genom att t.ex. specificera tillåtna lägen för resonanstoppen i figur 1. För högre ordningars system är det ej helt lätt att direkt tolka koefficienterna i skiftoperatorpolynomen som fysikaliska egenskaper, som vi kan göra här med ω_0^* och ζ^*. I dessa fall kan det vara lättare att studera överföringsfunktionen m.a.p. amplitud och faskurvor, resonanser etc.

4 STUDIER AV EHW-SERVOT

4.1 Beskrivning av servot

Servot i fråga är ett elektrohydrauliskt servo av fabrikat BBC. Servot styr ventilen som reglerar ånguttaget till HT-turbinen. Som insignal har servot PI-regulatorns utsignal. Eftersom ånguttaget ej är direkt proportionellt mot ventilläget har en olinjär kamskiva lagts så att ånguttaget blir en linjär funktion av PI-signalen. D.v.s man önskar att förstärkningen för delsystemet servo-ventil skall bli oberoende av ångflödet. En mycket förenklad bild av servot visas i figur 4.1.

PI-regulatorns utsignal påverkar ett elektriskt system som omvandlar regulatorns utspänning till styroljetryck. Vid ökande styroljetryck går styroljan in i servots styringång (1), och påverkar styrkolven (2) som i sin tur påverkar styrsliden (3), vilken öppnar ingången till kraftcylindern (4). Genom denna ingång kommer kraftolja och pressar upp kolven (5) mot tallriksfjädrarna (6). Kraftkolven påverkar en kamskiva (7) via en kuggkrans, och kolvens läge återkopplas till till styrsliden via kamskivan, en kamföljare och ett par hävarmar. Styrslidens läge ändras nu så att kraftoljeflödet minskar tills jämvikt inträffar. Vid minskat styroljetryck pressas kolven (5) tillbaka av tallriksfjädrarna (6) och kraftoljan dräneras via ett avlopp (9). Stången (8) överför kraftcylinderns läge till HT-ventilen och styr därmed ångflödet.



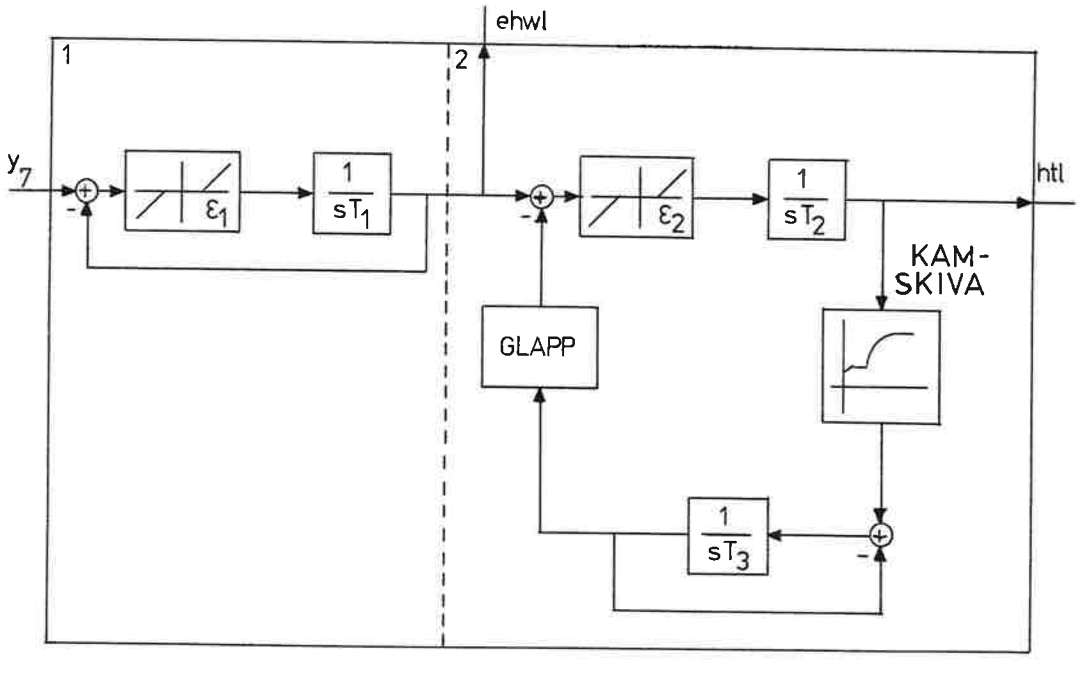
Figur 4.1 Principskiss av servot. Underlag för modellbeskrivning.

4.2 Modellering av servot

Experimentellt har man man konstaterat att servot uppför sig som ett andra ordningens system med en resonans i $\omega = 0.8$ Hz. Insignalen är då styroljetrycket och utsignalen kraftkolvens läge.

För att ställa upp en differentialekvationsmodell för systemet gör vi följande överläggning. Styroljan påverkar styrkolven mycket snabbt (tidskonstanten ≈ 10 ms), och vi bortser därför från dess dynamik. Kraftkolven däremot är betydligt större och det tar därför mycket längre tid för denna att reagera för ändringar i oljetrycket. Kolvens tidskonstant kan uppskattas till 0.1 s. Servot beskrives därför av en via kamskivan återkopplad integrator. Hela återkopplingen utgöres av systemet kamskiva- kamföljare-styrslid, och denna återkoppling antages kunna beskrivas av ett första ordningens system och ett glapp. Tidskonstanten T_3 i återkopplongsdynamiken avpassas så att hela systemet får en resonans i 0.8 Hz. T_3 blir då 0.14 s. Eventuell torrfriktion i kraftcylindern modelleras med dödzonen eps_2.

Glappet i kamföljaren och hävstängernas leder beskrivs av systemet GLAPP. Denna dynamiska modell av servot visas i figur 4.2.



Figur 4.2 Servots överföringsfunktion

Bortser man från glappet och dödzonerna kan överföringsfunktionen för delsystem 2 skrivas som

$$G(s) = \frac{T_3 s + 1}{T_2 T_3 s^2 + T_2 s + k} \tag{1}$$

$$k = \frac{\partial F(htl_0)}{\partial y} \tag{2}$$

Där $F(y)$ är kamskivefunktionen och htl_0 är htl för aktuell driftspunkt. Det elektriska system som som omvandlar PI-regulatorns utspänning till styroljetryck har modellerats som ett första ordningens system med tidskonstanten $T_1 = 0.05$ s.

4.3 Simuleringsförsök

För att närmare undersöka servot och några anslysmetoder har en del olika driftsfall och tänkbara felfunktioner simulerats. Servot har matats med en PRBS signal och in och utsignalen har lagrats. Alla tre signalerna y7, ehwl och htl är mätbara i det verkliga systemet, och att identifiera delsystem 1 ur y7 och ehwl erbjuder inga principiella svårigheter. Därför har vi valt att låta insignalen och ehwl signalen vara lika, d.v.s. att att koppla insignalen förbi delsystem 1, detta för att få en PRBS signal som insignal. Därmed kommer vi enbart att studera delsystem 2. I verkligheten har man emellertid ej en PRBS signal som insignal och därmed fungerar identifieringen ganska dåligt, ty en förutsättning vid identifiering är att insignalen är konstant under samplingsintervallet. Antar man att signalen ändras linjärt mellan samplingstidpunkterna kan man,genom att derivera in och utsignalerna (multiplicera med $1-q^{-1}$), få en styckvis konstant signal som kan användas vid identifieringen. Faktorn $1-q^{-1}$ förekommer i både höger och vänsterled, och kommer ej att påverka resultatet. Detta trick behöver vi nu ej använda.

Vid undersökningen har först systemet simulerats i Simnon varvid ehwl och htl signalerna har lagrats. Insignalen till servot har varit en konstant signal med medelvärde i intervallet [0,1] överlagrad med en PRBS signal med amplituden 0.01. Simuleringstiden har varit 200 s och samplingsfrekvensen 5 Hz. Detta har gett 1001 sampel.Dessa filer som genererats i Simnon har sedan analyserats i Idpac. Vid Idpac analysen har variablerna först skalats om så att de fått tidsmedelvärdet noll. Detta göres i Idpac med kommandot TREND. De formler som Idpac använder för beräkning av statistiska storheter och spektra har presenterats i kapitel 3. Från varje simulering har en modell identifierats. Genomgående har ml-metoden använts och modellerna har antagits vara av andra ordningen. Alla analysresultat har jämförts med en referensmodell med parameteruppsättningen

$$T_2 = 0.10 \quad T_3 = 0.14$$

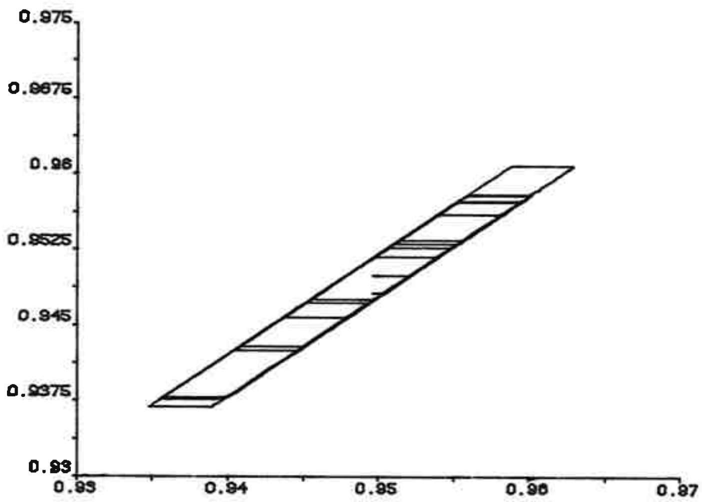
$$e_1 = 0.00 \quad e_2 = 0.00$$

$$eps_1 = 0.00 \; eps_2 = 0.00$$

I 4.3.1 undersökes ändring av e_1 och e_2 d.v.s. glapp. I 4.3.2 ändras eps_1 och eps_2 vilket innebär ändrad friktion. I 4.3.3 ändras T_2 och T_3 som svarar mot ändring i dynamiken av någon annan anledning t.ex. ändring av hydraultrycket. I 4.3.4 visas effekten av olika nivå på insignalen.

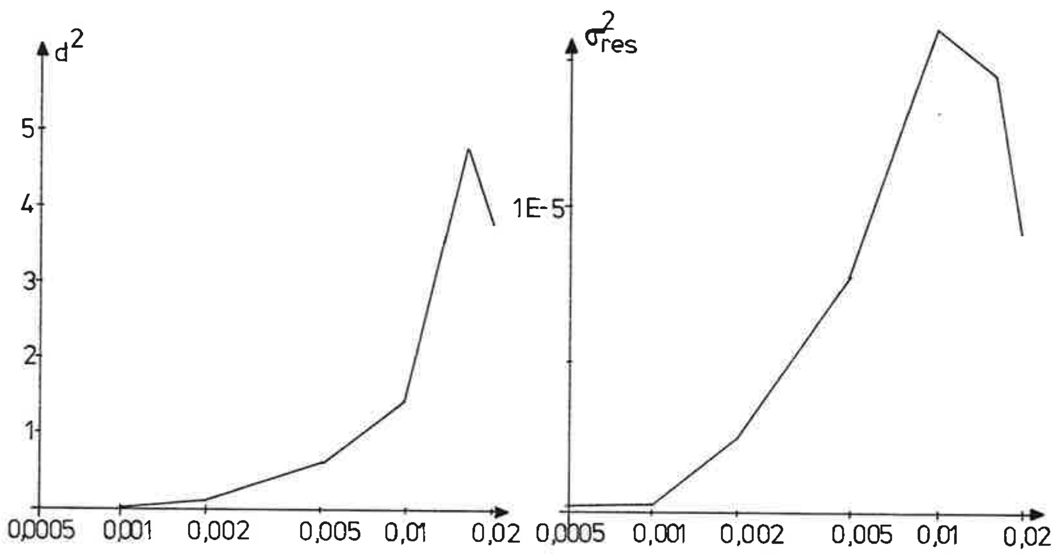
4.3.1 Glapp

Den första olinjäritet som studerades var glapp i servot. Glappet implementerades i koden för ehw-servot genom att integrera $\dot{y}_3$ eller 0, beroende på tecknet hos y_3 och var i (y_3,x) planet man befinner sig. En kurva i detta plan då man har ett glapp visas i figur 4.3.

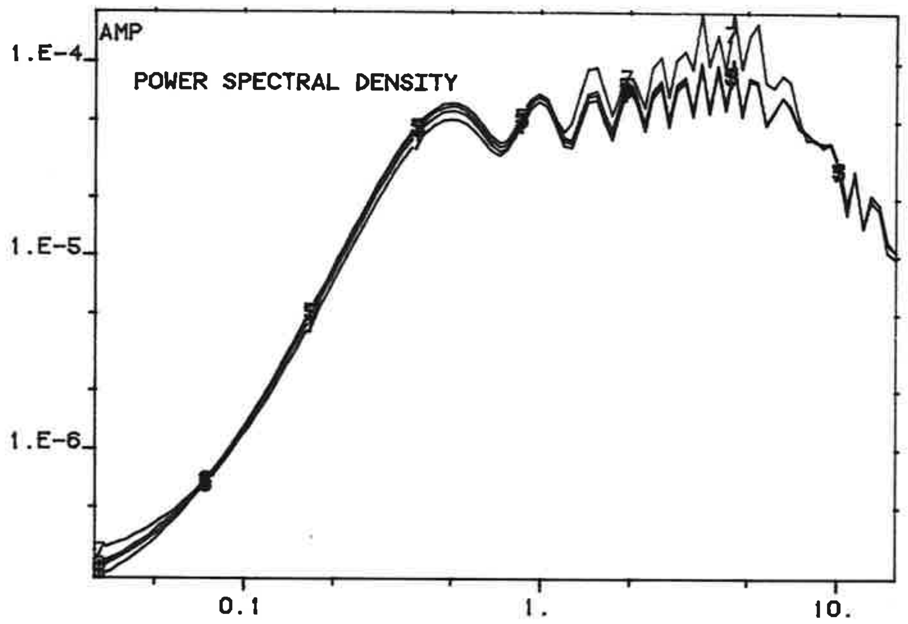


Figur 4.3 Glappets utseende i (y_3,x) planet.

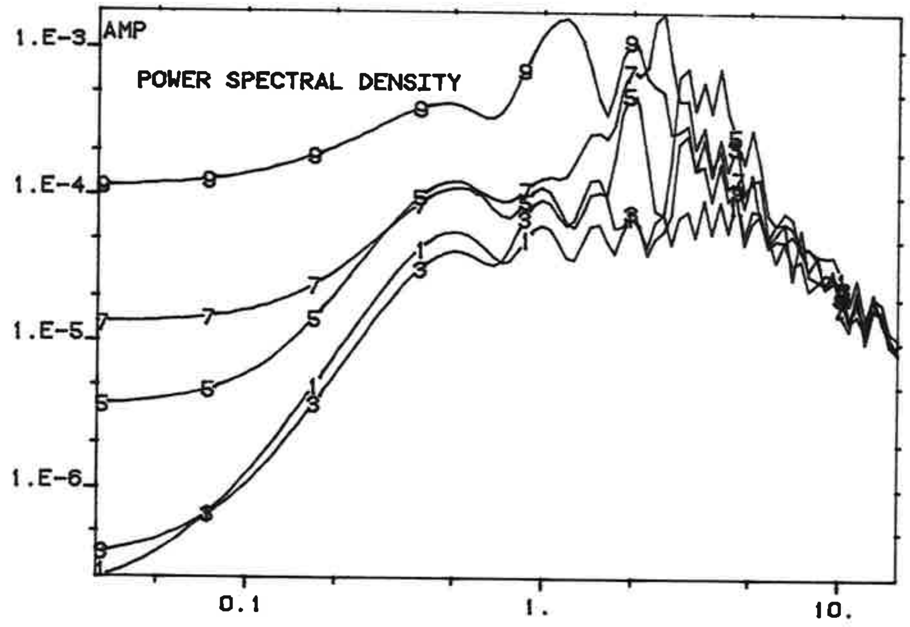
Parametern e_2 anger glappets storlek, medan e_1 bara är en hjälpstorhet som behövs av numeriska skäl. Glapp har simulerats e_2 = 0, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.015, 0.02. De stora galppen ger tydliga utslag vid analysen, men är fysikaliskt orimliga. Htl-läget är 80 mm vid fullt utslag. Ett glapp på 0.02 skulle i praktiken då motsvara 1.6 mm, vilket man knappast kan förvänta av en Schweizisk precisions produkt. För övrigt ger redan e_2=0.002 stora ändringar i domtrycket då BWR modellen körs (se kapitel 5). De stora glappen här är således mest att se som en test av analysmetoderna. Resultaten av simuleringarna och analyserna presenteras nedan. I Figur 4.4 visas parameteravståndets och residualvariansens beroende av e_2. Utsignalens spektra visas i figur 4.5 och 4.6. I Figur 4.7 visas amlpitudkuvurna för de identifierade modellerna. Figur 4.8 och figur 4.9 visar slutligen korskorrelation mellan in och utsignal.



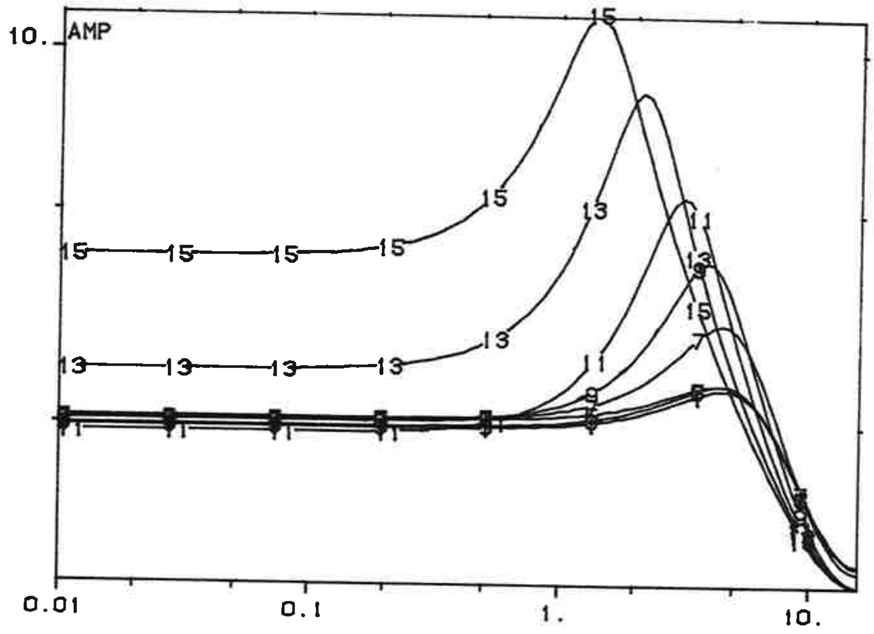
Figur 4.4 Parameteravstånd och residualvarians som funktion av glappstorleken.



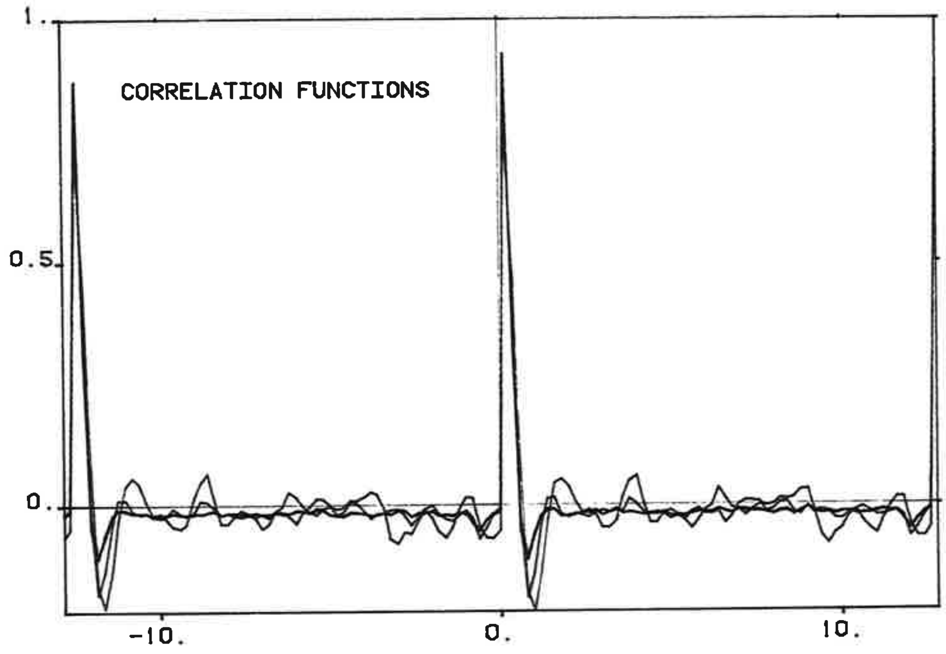
Figur 4.5 Utsignalspektrum för e_2=0, 0.0005, 0.001, 0.002.



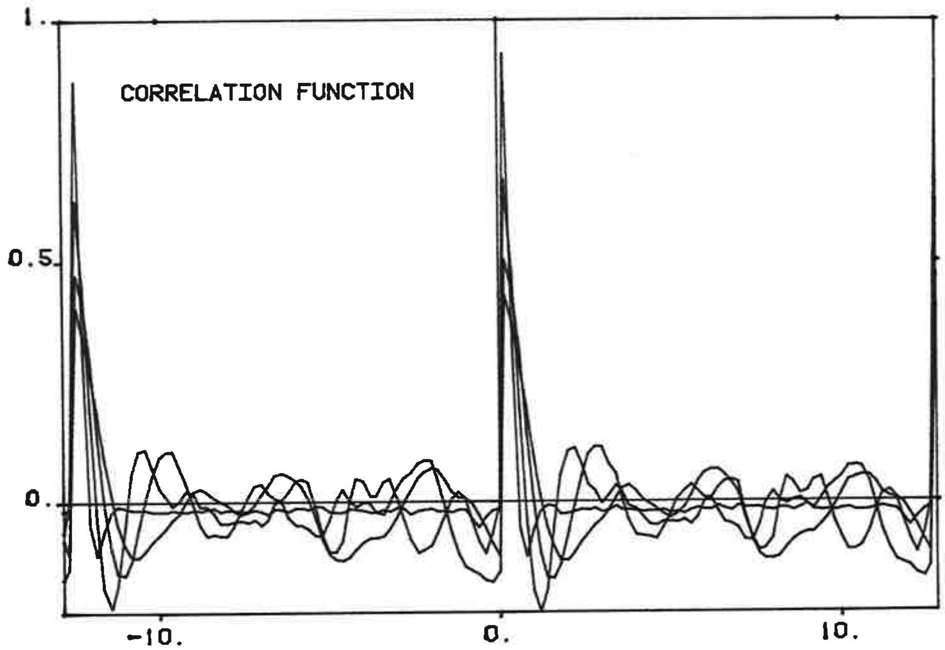
Figur 4.6 Utsignalspektrum för e_2=0, 0.005, 0.01,0.015, 0.02.



Figur 4.7 Amplitudkurvor för de identifierade modellerna. e_2= 0, 0.0005, 0.001, 0.002, 0.005, 0.01, 0.015, 0.02.



Figur 4.8 Korskorrelation mellan in och utsignal för e_2=0, 0.0005, 0.001, 0.002, 0.005.



Figur 4.9 Korskorrelation mellan in och utsignal för e_2= 0, 0.01, 0.015, 0.02.

En hel del intressanta iakttagelser kan göras från dessa figurer.

Parameteravståndet och residualvariansen ökar kraftigt för $e_2 > 0.002$. Residualvariansens ökning tyder på svårigheter att identifiera modellen p.g.a. en kraftig olinjäritet. Det är också i detta område man i BWR modellen börjar märka glappet (se kapitel 5.2).

Utsignalens spektrum visar en karakteristisk resonans vid ≈ 3 rad/s. Servots lågfrekvensförstärkning stiger med ökat glapp, medan högfrekvensförstärkningen är praktiskt taget oberoende av glappets storlek.

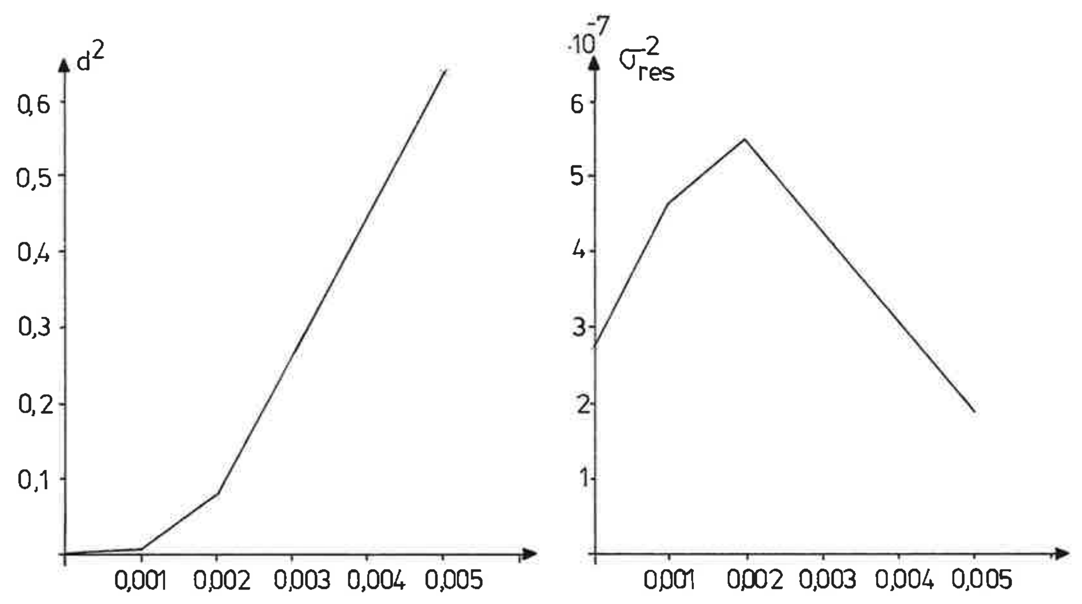
För glapp>0.002 är resonanstoppens läge hos amplitudkurvorna för de identifierade systemen ett direkt mått på e_2

Läget och storlek för första minimum i korrelations funktionen är också ett en bra indikation på att ett glapp uppstått.

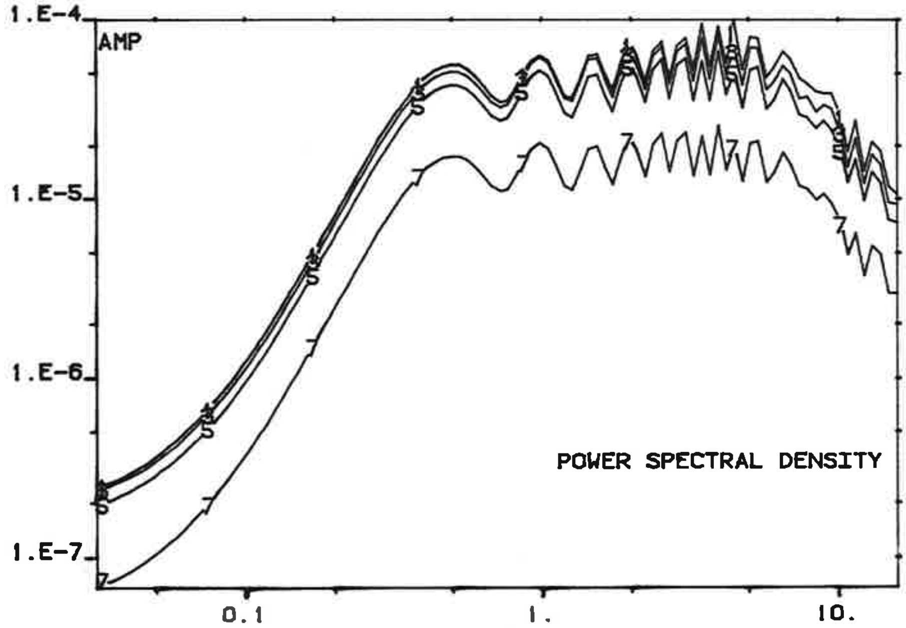
Den enklaste detekteringsmetoden torde dock vara att i utsignalens spektrum söka efter den karakteristiska resonansen vid 3 rad/s, detta under förutsättning att insignalens spektrum ej ändrats. Har så skett måste t.ex. identifiering tillgripas, för att mera direkt detektera servots förändring.

4.3.2 Dödzon

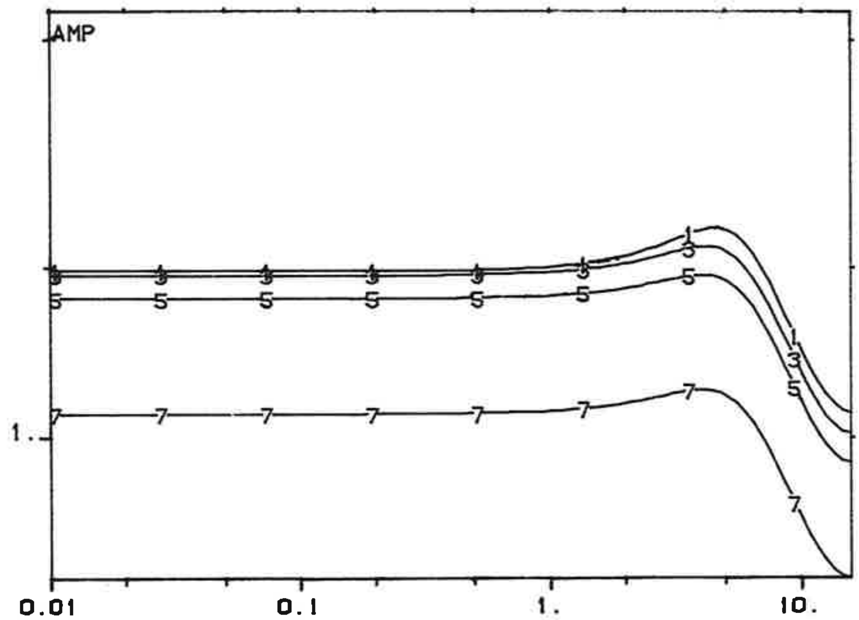
Dödzonen kan betraktas som en form av torrfriktion; en kolv rör sig ej förrän en kraft överstiger ett visst värde. Dödzonens storlek anges av konstanten eps_2. För att överhuvudtaget få någon utsignal fordras att man väljer eps_2 ganska litet. Vi har här valt att undersöka $eps_2 = 0.001$, 0.002, 0.005.



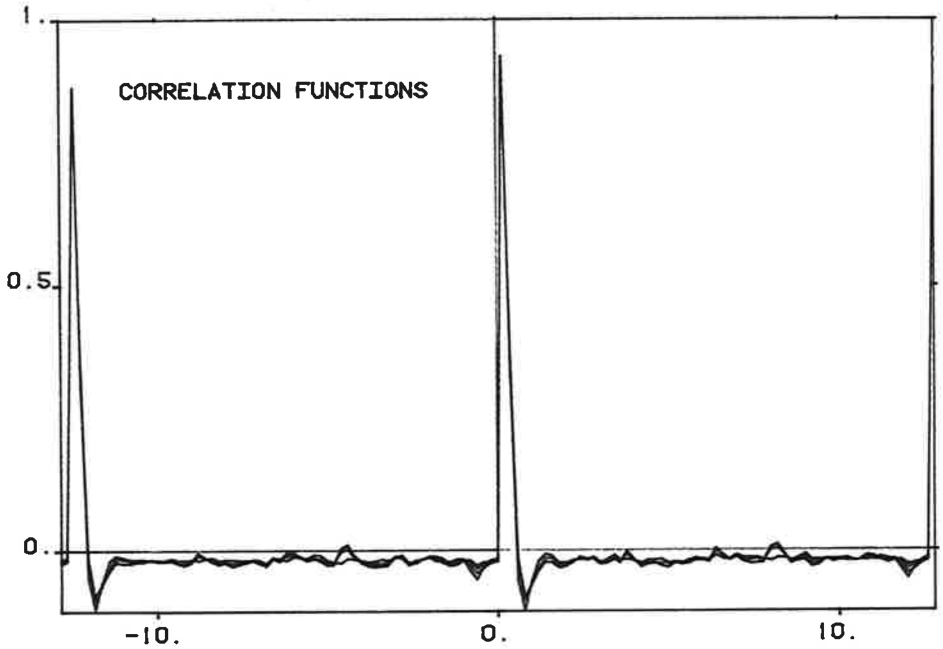
Figur 4.10 Parameteravstånd och residualvarians som funktion av dödzonens storlek.



Figur 4.11 Utsignalens spektrum. eps_2 = 0, 0.001, 0.002, 0.005.



Figur 4.12 Identifierade modellens amplitudkurva eps_2=0, 0.001, 0.002, 0.005.



Figur 4.13 Korskorrelation mellan in och utsignal. $eps_2 = 0, 0.001, 0.002, 0.005$

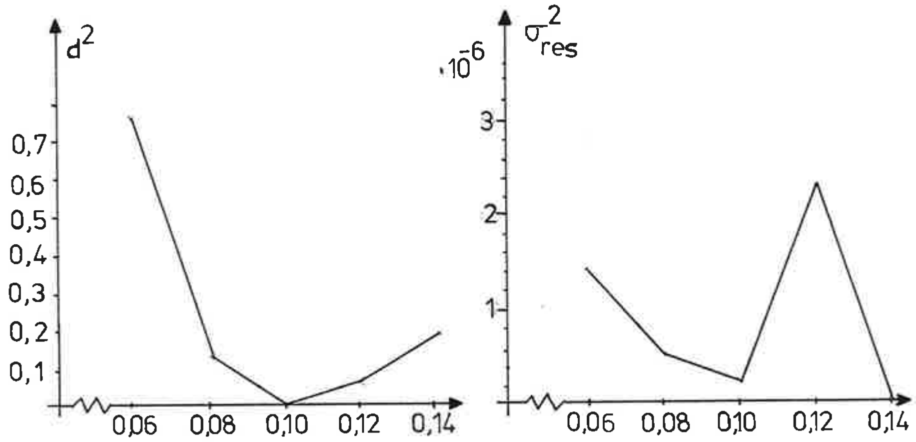
Det mest karakteristiska vid en ökning av dödzonen är att den identifierade modellens amplitudkurva sjunker likformigt i bodedigrammet, över alla frekvenser.

4.3.3 Ändring av tidskonstanterna

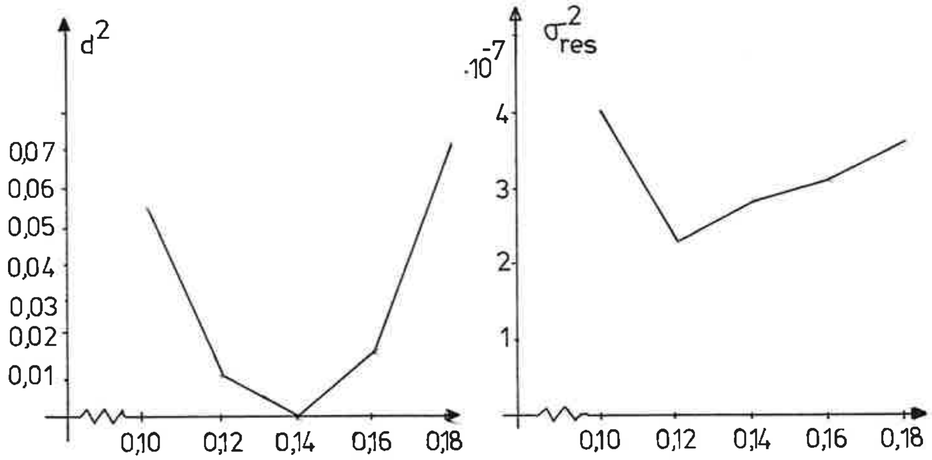
En ändring av T_2 eller T_3 svarar mot en ändring av differentialekvationernas dynamik. Detta yttrar sig praktiskt som ändrad gångtid hos servot. En anledning till detta kan vara ändrat tryck i hydrauloljan. I normalt driftsfall är T_2=0.10 och T_3=0.14. Vi har valt att undersöka driftsfallen

T_2	0.06	0.08	0.12	0.14	0.10	0.10	0.10	0.10
T_3	0.14	0.14	0.14	0.14	0.10	0.12	0.16	0.18

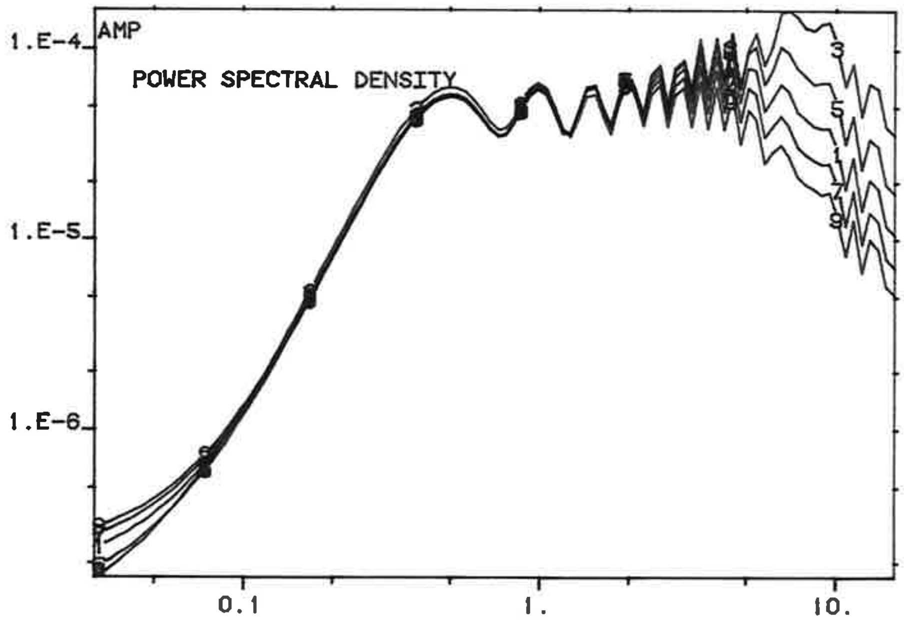
a)



b)

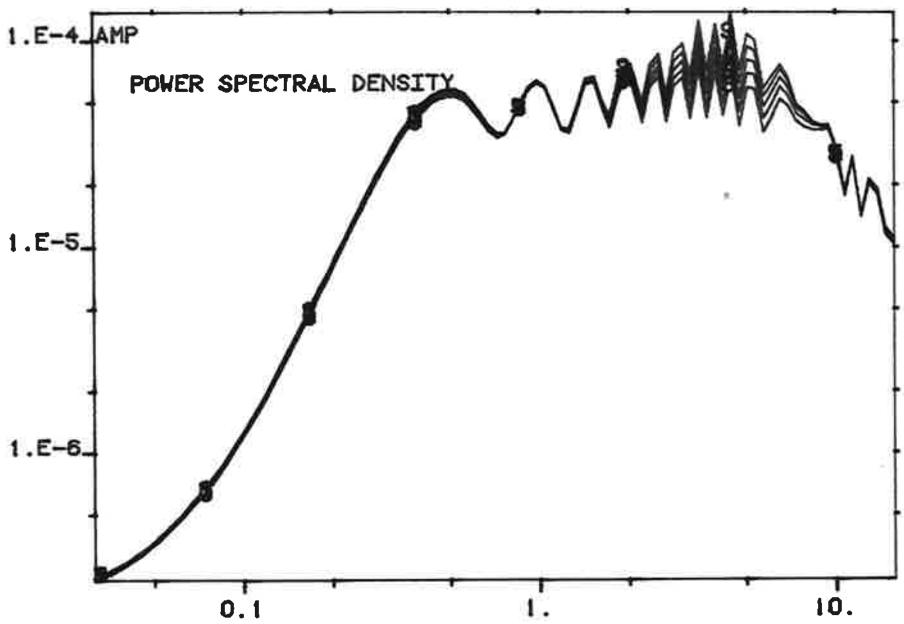


Figur 4.14 Parameteravstånd och residualvarians för a) avvikelser i T_2 b) avvikelser i T_3.



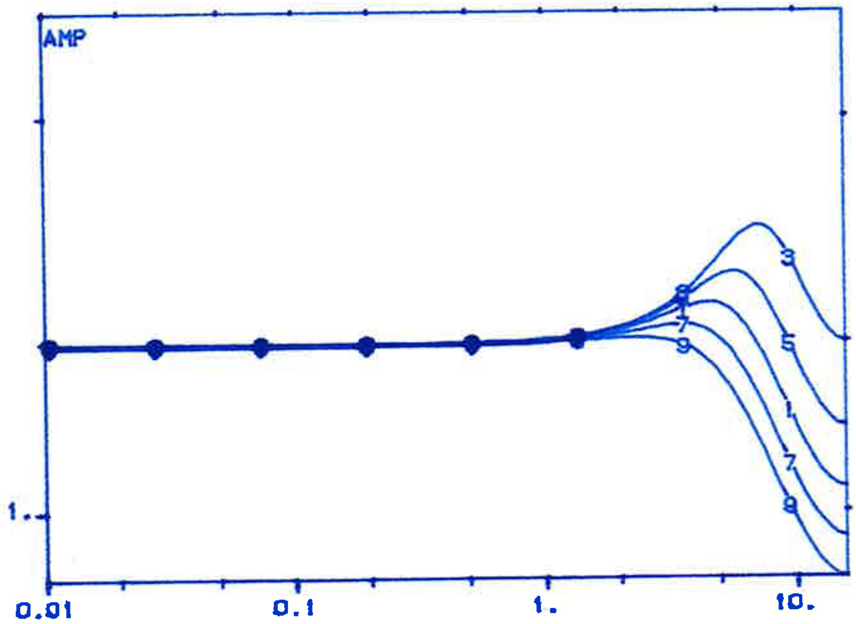
Figur 4.15 Utsignalens spektrum för avvikelser i T_2.

Referenskurva: 1.

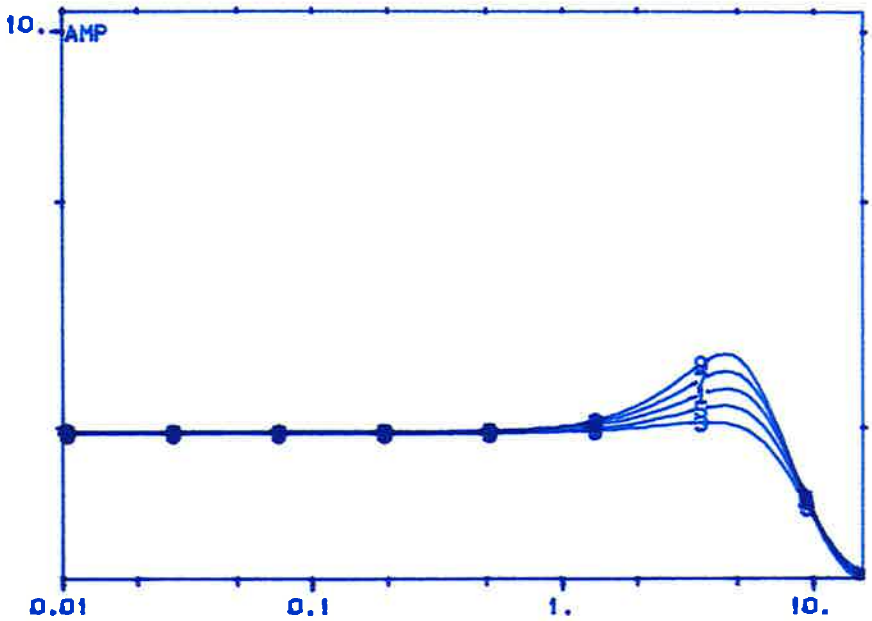


Figur 4.16 Utsignalens spektrum för avvikelser i T_3.

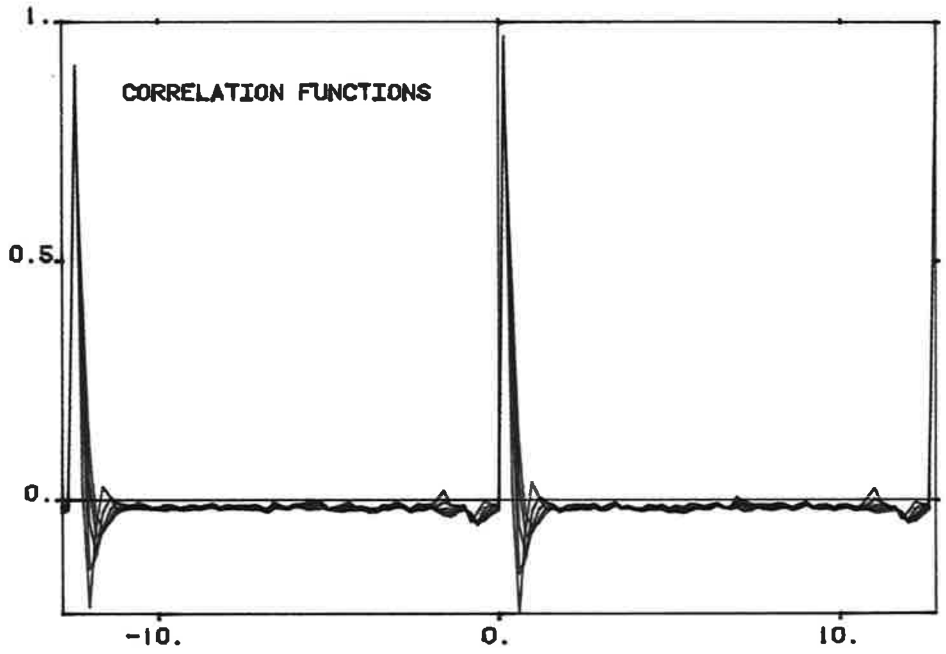
Referenskurva: 1.



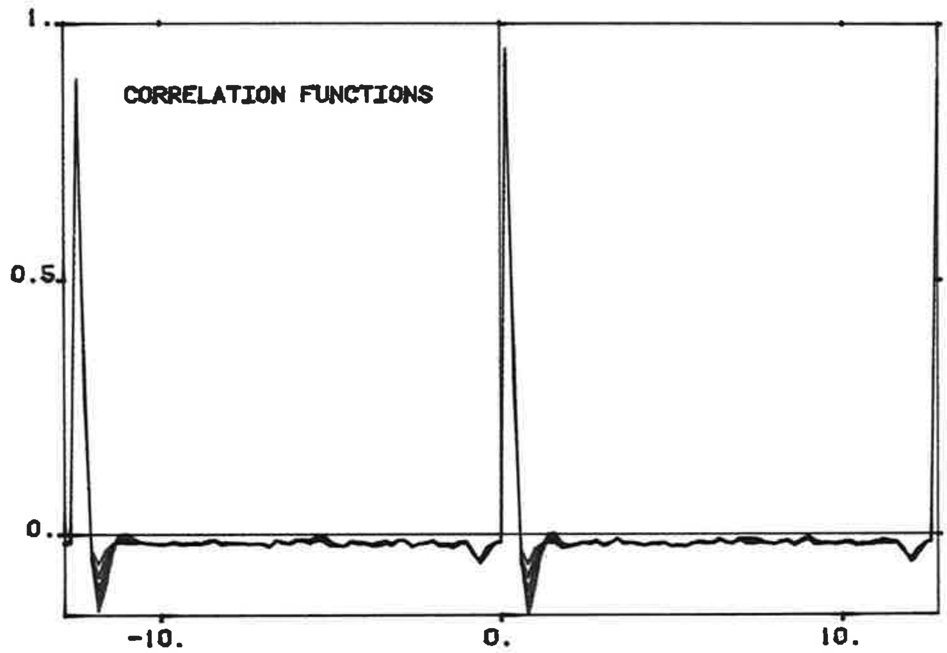
Figur 4.17 De identifierade modellernas amplitudkurvor för avvikelser i T_2. Referenskurva: 1.



Figur 4.18 De identifierade modellernas amplitudkurvor för avvikelser i T_3. Referenskurva: 1.



Figur 4.19 Korskorrelation mellan in och utsignal för avvikelser i T_2. Referenskurva: 1.

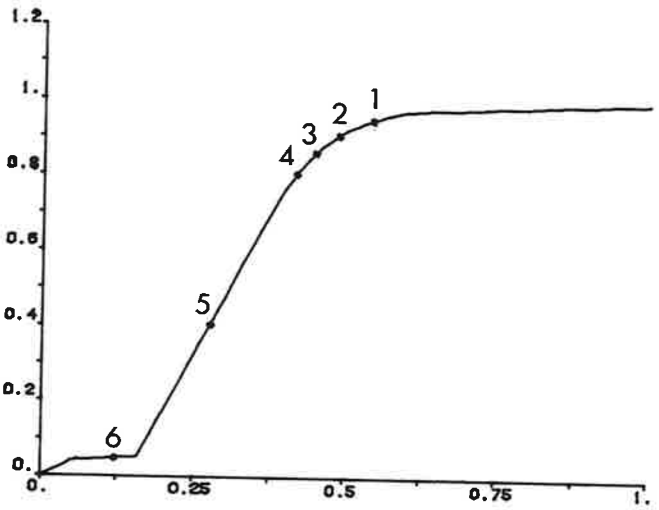


Figur 4.20 Korskorrelation mellan in och utsignal för avvikelser i T_3. Referenskurva: 1.

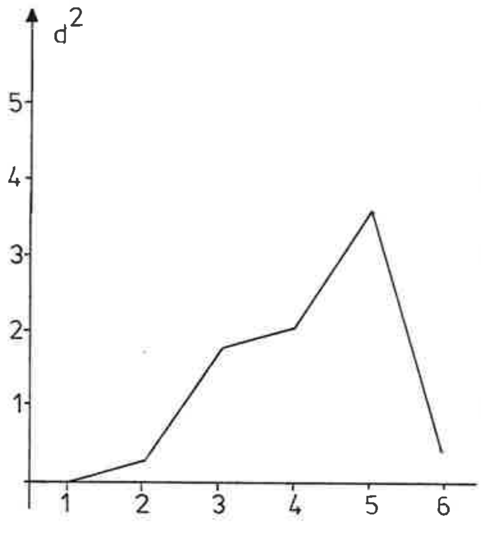
En ändring av T_2 är mycket lätt att detektera ur dessa kurvor, helt enkelt genom att titta på högfrekvensnivån i utsignalspektrat. Nivån, eller effektintegralen kan här direkt kalibreras mot T_2. En ändring av T_3 är svårare att upptäcka, vilket framgår av figurerna. Möjligen kan man för $T_3 > 0.14$ s använda resonanstoppen vid 5 rad/s i de identifierade modellernas amplitudkurvor för att detektera avvikelsen, samtidigt som man observerar att lågfrekvensförstärkningen är oförändrad.

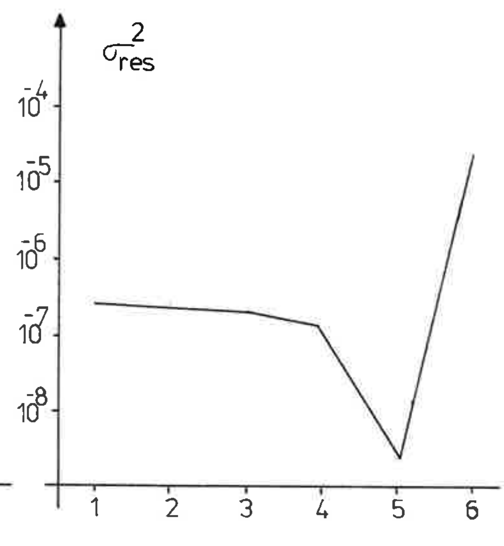
4.3.4 Ändring av driftspunkten

Då man gör uppstarter av reaktorn kan det vara av intresse att veta något om servots uppförande vid några olika driftslägen för insignalen. Dessa driftspunkter presenteras i kamskivekurvan (figur 4.14), där htl-läget avsätts mot ehw-läget.

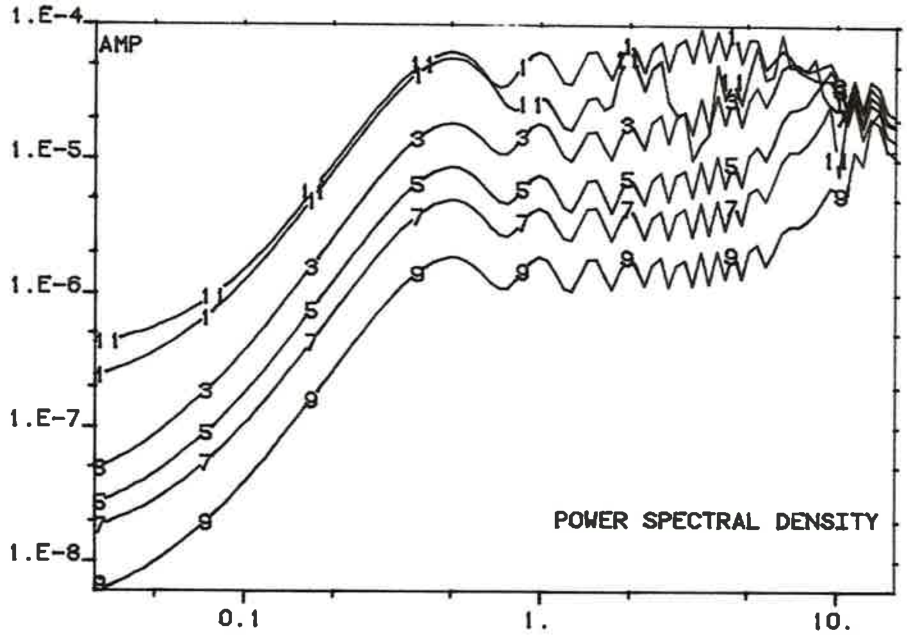


Figur 4.21 Driftspunkterna 1 till 6 avsatta i kamskivekurvan.

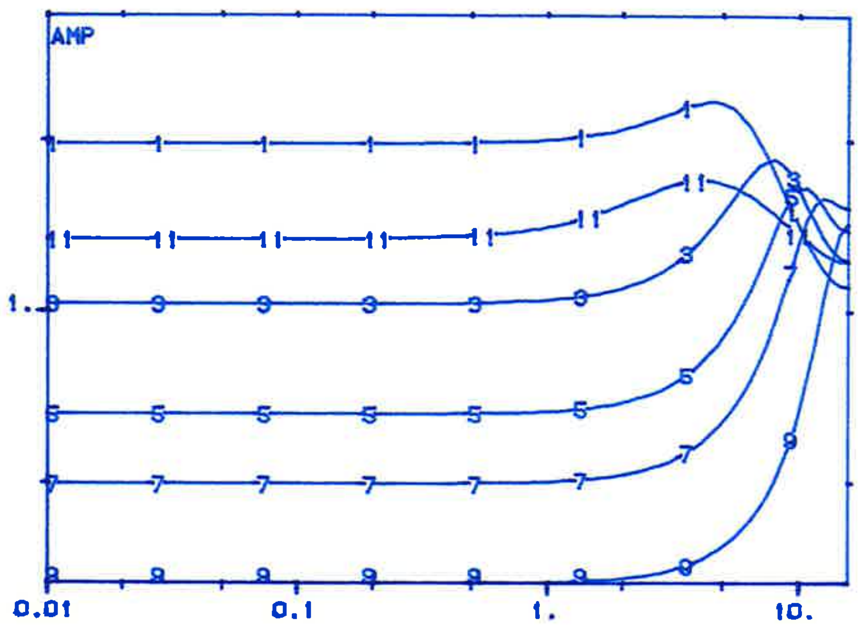




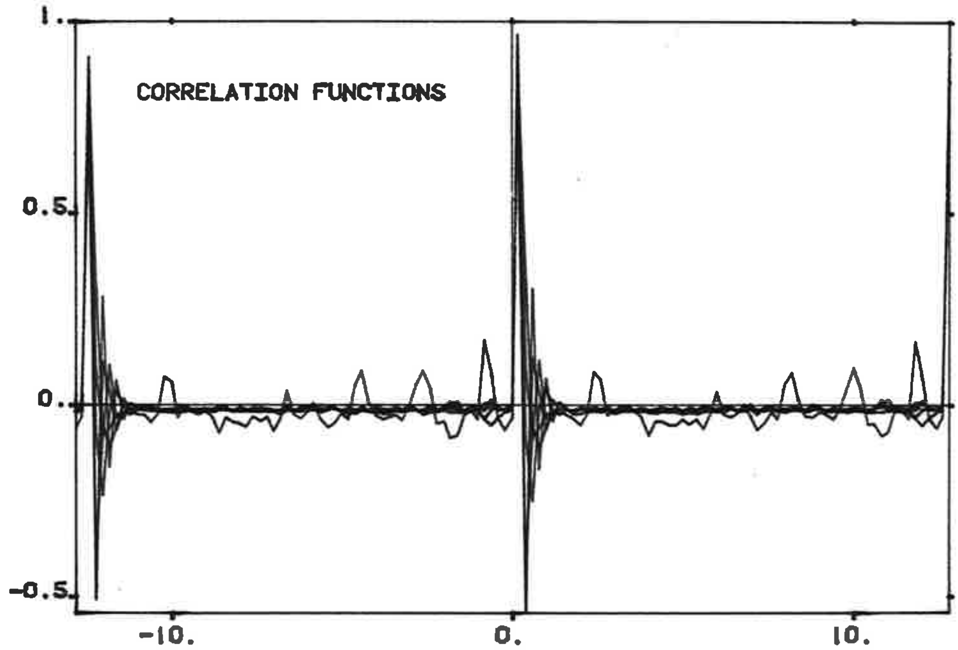
Figur 4.22 Parameteravstånd och residualvarians som funktion av driftspunkt angiven i htl-läge.



Figur 4.23 Utsignalens spektrum för driftpunkterna i ordning enligt figur 4.21



Figur 4.24 De identifierade modellernas amplitudkurvor.



Figur 4.25 Korskorrelationen mellan in och utsignal vid driftpunkterna enligt figur 4.14.

Man ser ur de identifierade modellerna tydligt hur kamskivekurvans lutning resulterar i förändring av lågfrekvensförstärkningen. (Jfr (2)).

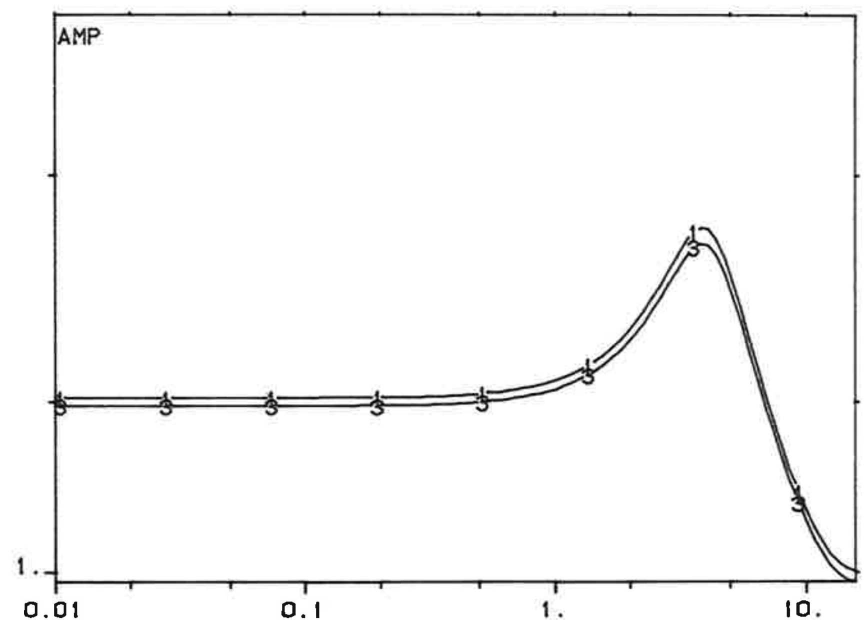
4.3.5 Brusstörning vid mätning av ehw-läget.

Jag ger här exempel på ett två av de ovanstående analyserna men här är ehw-lägesmätningen störd av en brussignal. Det störande bruset antages vara vitt, och har en amplitud på 10% av PRBS-signalens amplitud. Observera att bruset endast påverkar mätningen av ehw-läget och ej utsignalen. Därför räcker det att studera hur identifierings och korskorrelations beräkningen störs. De exempel jag valt är

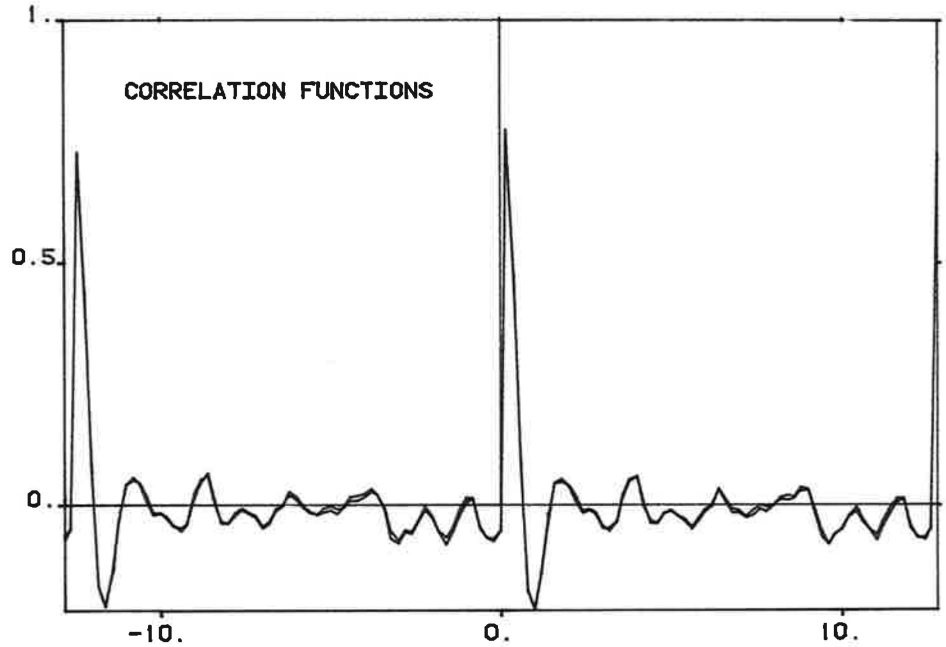
1. Servo med 0.005 glapp
2. Tidskonstant T_2 ändrad till 0.14

1.Parameteravståndet mellan det störda systemets identifierade modell och referensmodellen var 0.524,mot 0.607 mellan det ostörda systemets modell och referensmodellen. Residualernas varians var 2.38E-5 mot 8.52E-6 hos referenssystemet.

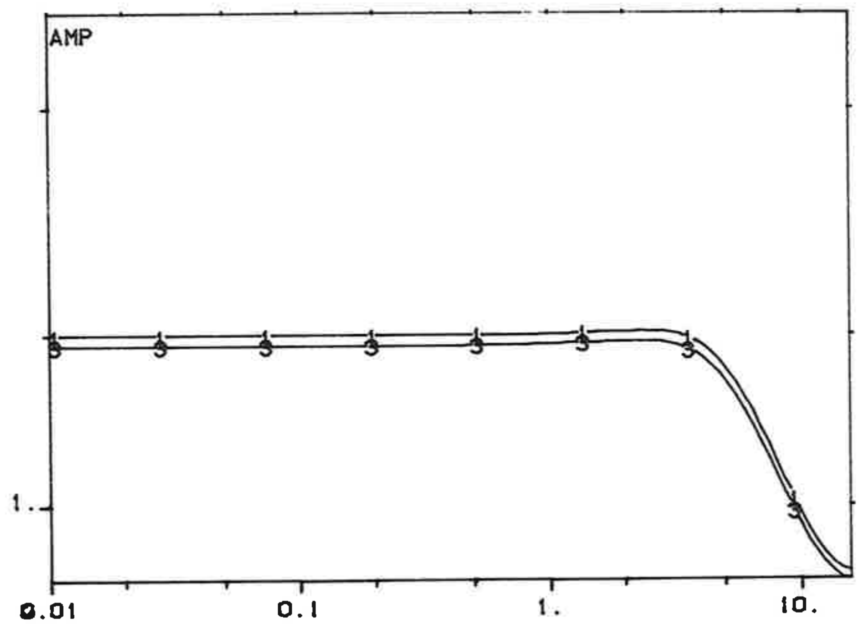
2.Parameteravståndet mellan det störda systemets identifierade modell och referensmodellen var 0.242,mot 0.194 mellan det ostörda systemets modell och referensmodellen. Residualernas varians var 6.29E-6 mot 5.41E-5 hos referenssystemet.



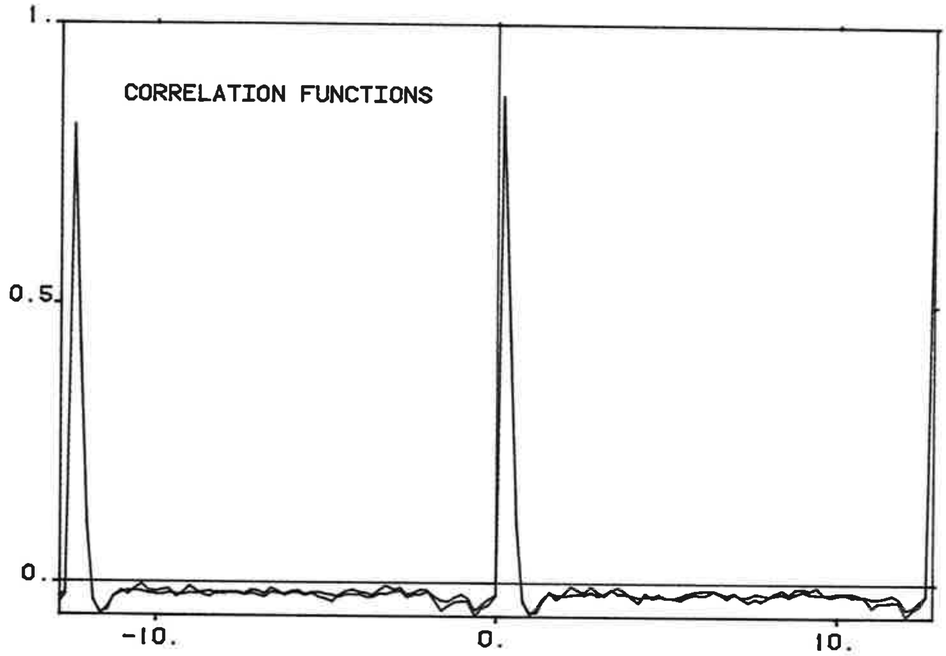
Figur 4.26 Exempel 1. Identifierade modellens amplitudkurva. 1:stört system, 2:ostört system.



Figur 4.27 Exempel 1. Korskorrelation mellan in och utsignal. 1:stört system, 2:ostört system.



Figur 4.28 Exempel 2. Identifierade modellens amplitudkurva. 1:stört system, 2:ostört system.



Figur 4.29 Exempel 2. Korskorrelation mellan in och utsignal. 1:stört system, 2:ostört system.

Som synes blir avvikelserna från referenskurvorna i bodediagrammen ej speciellt stora. Men det faktum att avvikelser uppstår genom mätbrus måste man ta hänsyn till vid feldetekteringen, t.ex genom att öka toleransen för vad som skall vara en avvikelse. I detta enkla fall med enbart mätbrus ser man att parameteravståndet kan variera ±0.1 jämfört med en ostörd modell. Parameteravstånd bör man, då man vet att man har en brusstörd signal, behandla med försiktighet. En annan metod är att minska brusets inverkan genom att utföra analyserna på ett större statistiskt material.

4.4 Detekteringsregler

Jag ger här förslag på några regler, med hjälp av vilka man kan detektera följande felfunktioner

1. Glapp >0.002
2. Dödzon >0.001
3. Ändring av T_2 och T_3 i intervallen $T_2 \in [0.06, 0.14]$ $T_3 \in [0.14, 0.18]$

Reglerna är bara testade för att detektera en felfunktion åt gången, och gäller under förutsättning att insignalen är en PRBS-signal med medelvärde 0.94 och amplitud 0.01. Mätningarna förutsättes vara ostörda. Dessa regler bygger i hög grad på visuella intryck av kurvorna som pressenterats tidigare, och behöver för att kunna användas för datorberäkningar skärpas matematiskt. Följande beteckningar användes vid regeldefinitionen:

Φ_y	uppmätt autospektrum
$\Phi_{y,ref}$	autospektrum för referenssignalen
H_{ref}	referensmodellens frekvenssvar
H	identifierade modellens frekvenssvar

R0: R1 - R2 är tillämpliga om insignalspektrat ej har ändrats i frekvensintervallet 4-12 rad/s.

R1: Om den partiella variansen $\int_{\omega_1}^{\omega_2} \Phi_y^2 d\omega$ avviker från $\int_{\omega_2}^{\omega_2} \Phi_{y,ref}^2 d\omega$ $>\varepsilon_1$ har T_2 ändrats, förutsatt att lågfrekvensdelarna av Φ_y och $\Phi_{y,ref}$ är skilda med $<\varepsilon_2$. Avvikelsen är ett direkt mått på hur mycket T_2 avviker från 0.10 s. $\omega_1 \approx 7$ rad/s, $\omega_2 \approx 12$ rad/s.

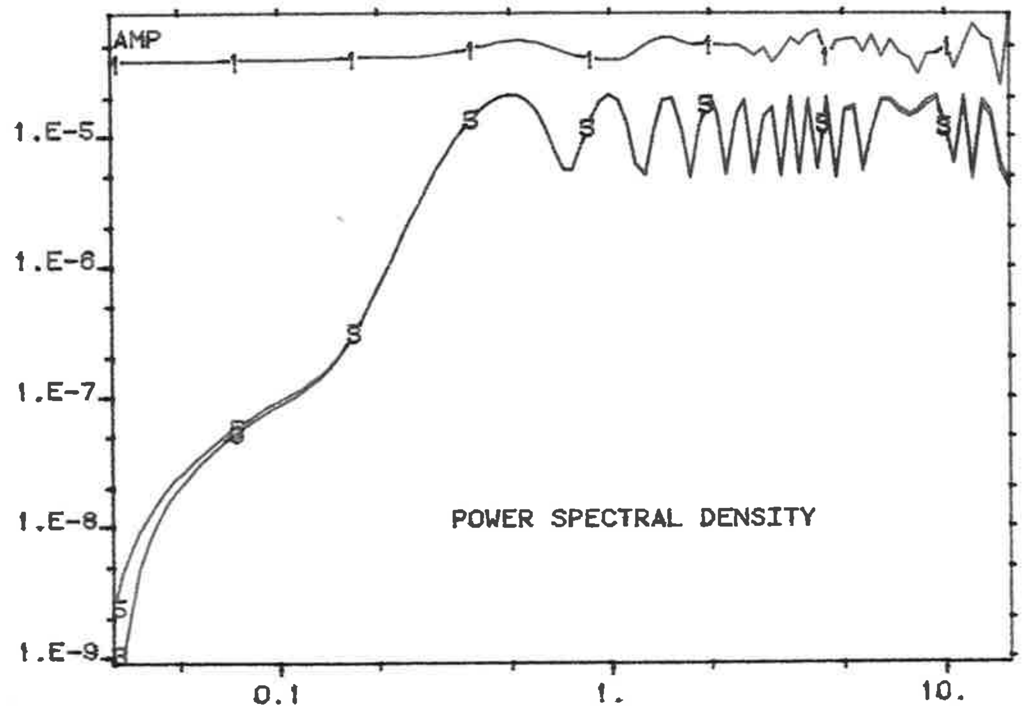
R2: Om det i utsignalspektrat finns en resonanstopp >0.0001 i intervallet 2 - 5 rad/s och parameteravståndet >0.01 så finns det ett glapp >0.002 i annat fall är glappet <0.002.

R3: Är H av samma form i bodediagrammet som H_{ref} men förskjuten i $\log|H|$-led för alla frekvenser och förskjutningen $>\varepsilon_3$ så har vi en dödzon >0.002.

R4: Om parameteravståndet >0.02 och vi har en resonanstopp i ≈ 5 rad/s så har T_3 ett värde i $[0.14, 0.18]$.

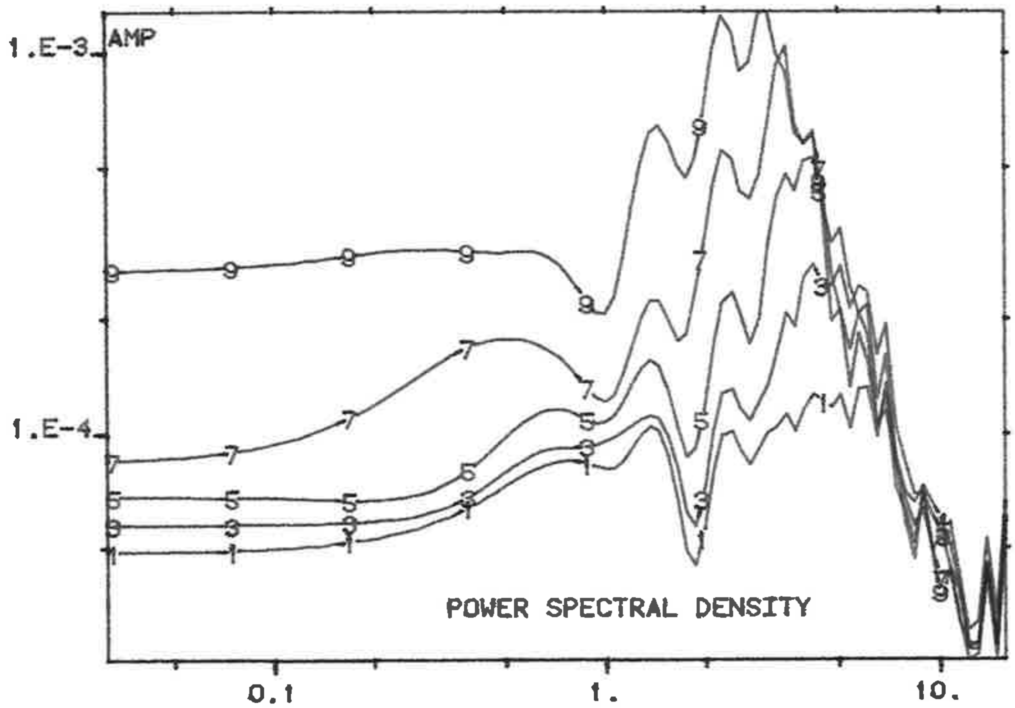
Vissa undersökningar har gjorts med andra insignaler än en ren PRBS-signal t.ex. vitt brus och PRBS-signal som passerat ett lågpassfilter med T=0.05s.

I figur 4.30 visas spektrum av de olika signaler som har undersökts.

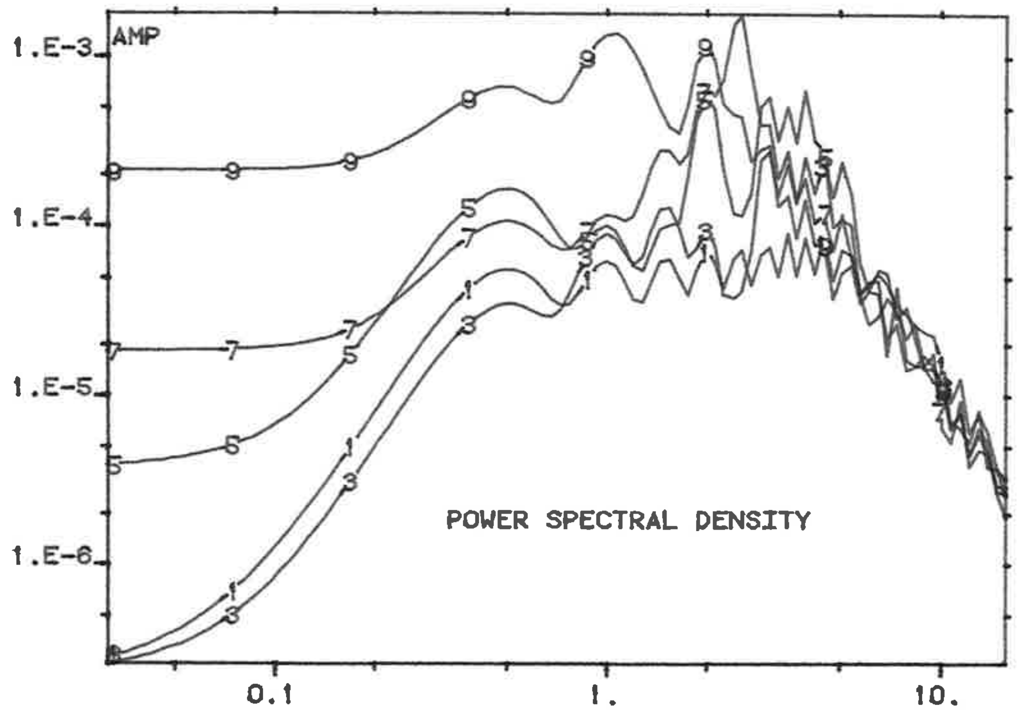


Figur 4.30 Kurva 1: spektrum av vitt brus, kurva 3: spektrum av störd PRBS-signal, kurva 5: spektrum av ostörd PRBS-signal.

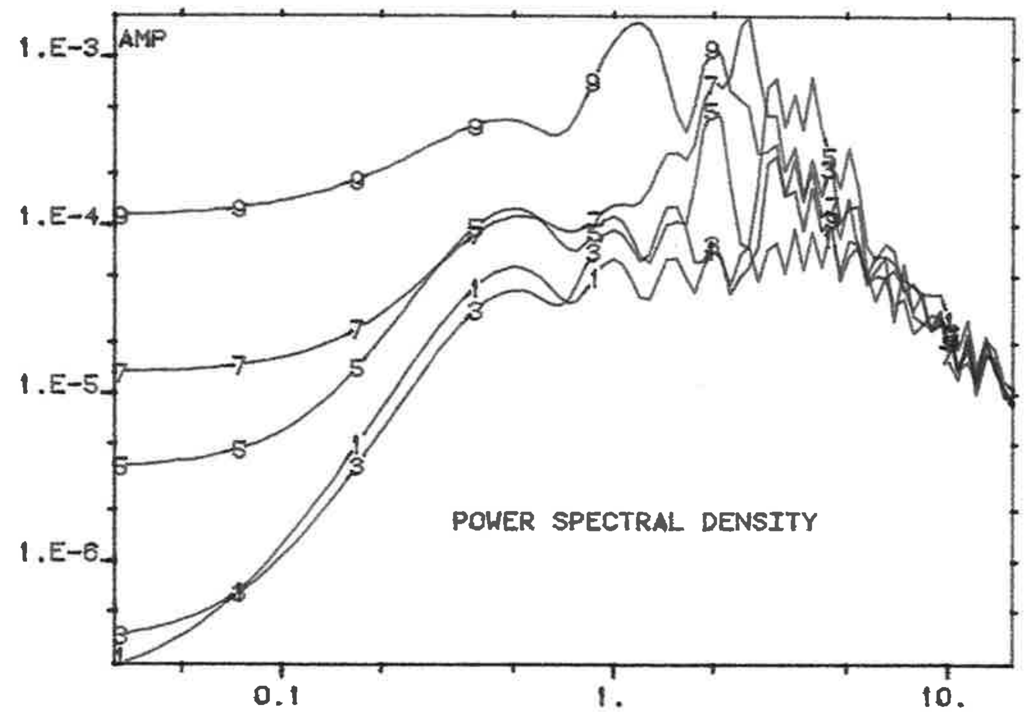
Figurerna 4.31-33 visar autopektra av utsignalerna då servot matas med vitt brus, störd respektive ostörd PRBS-signal. Observera att resonanserna vid 1-4 rad/s blir oförändrade i de tre fallen. Den största förändringen blir i fallet med vitt brus där utsignalens effekt för låga frekvenser drastiskt har höjts på grund av det ändrade insignalspektrat.



Figur 4.31 Utsignalens spektrum för e_2=0, 0.005, 0.01, 0.015, 0.02 då insignalen är vitt brus.

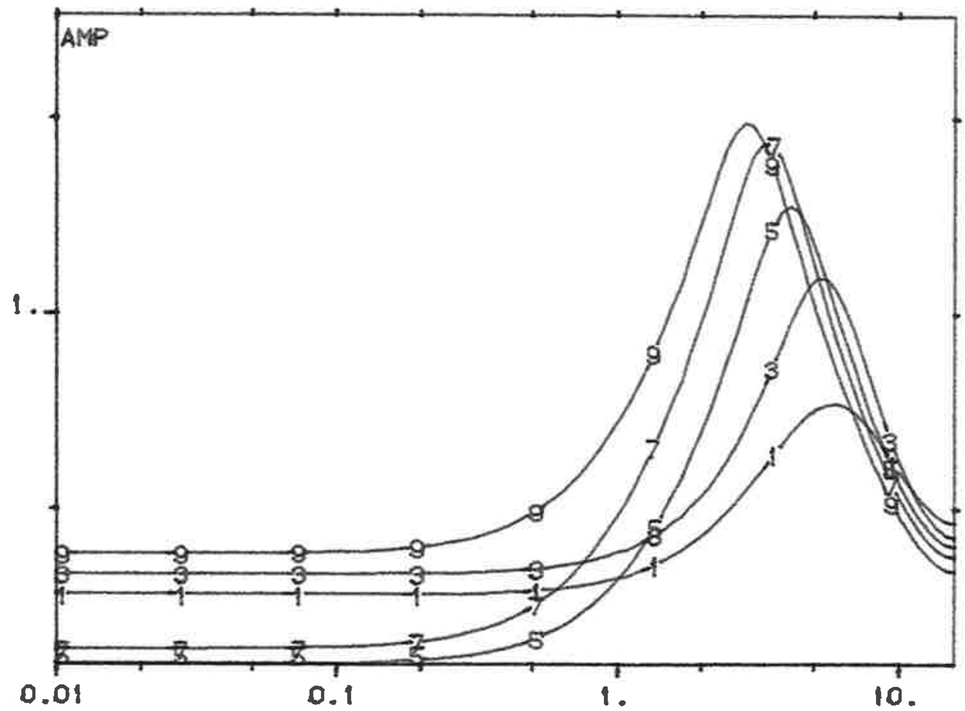


Figur 4.32 Utsignalens spektrum för e_2=0, 0.005, 0.01, 0.015, 0.02 då insignalen är en störd PRBS-signal.

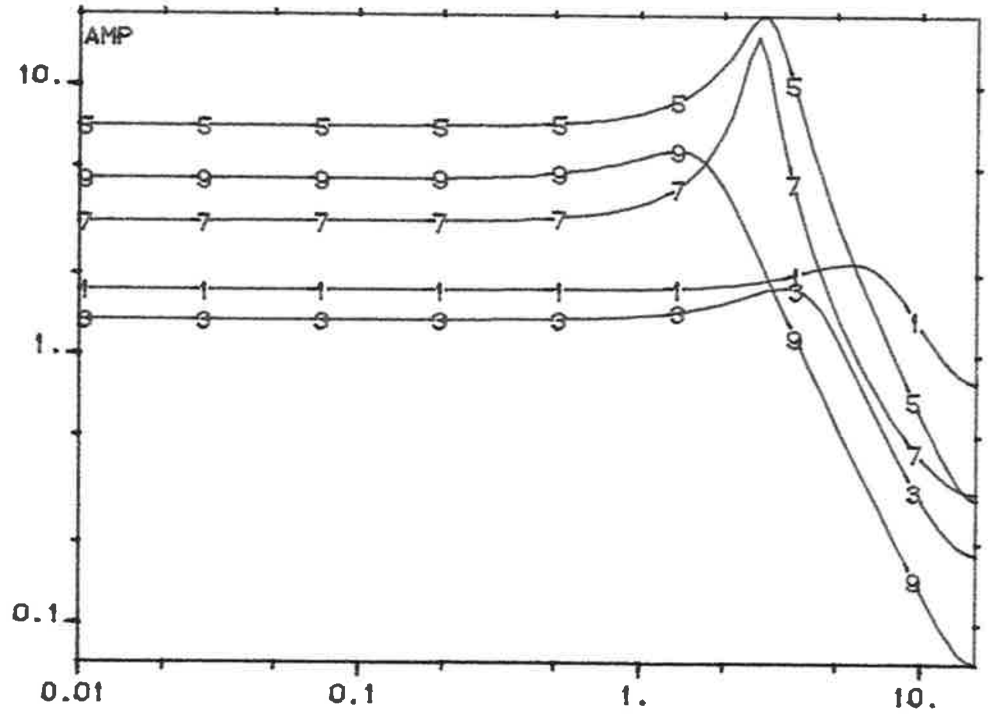


Figur 4.33 Utsignalens spektrum för e_2=0, 0.005, 0.01, 0.015, 0.02 då insignalen är en PRBS-signal.

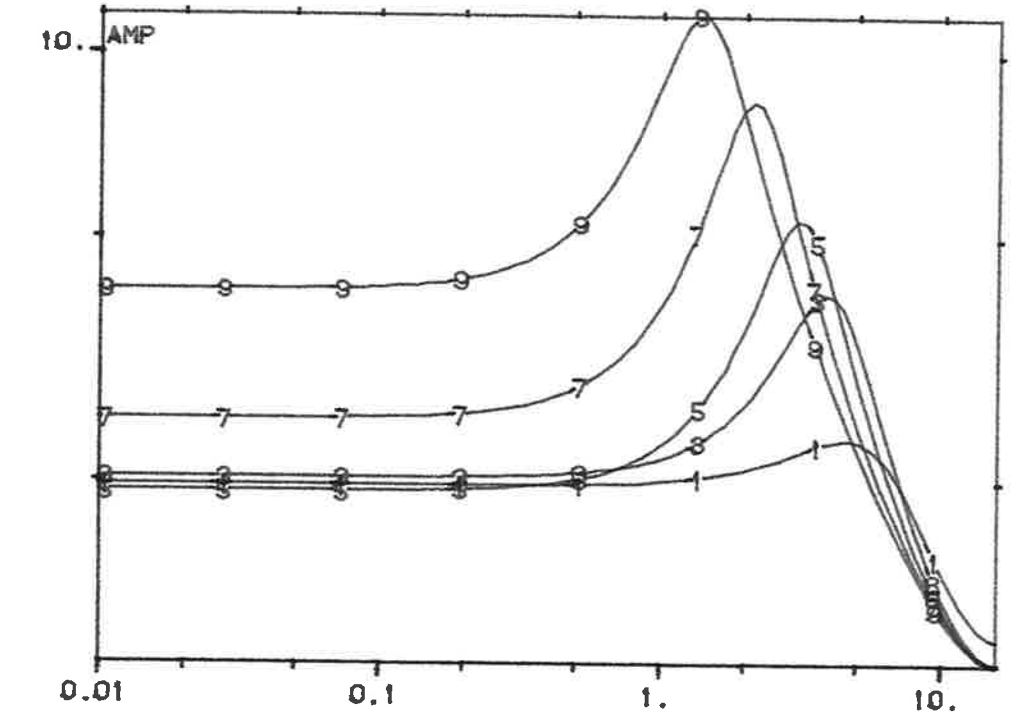
Identifiering har utförts i dessa tre fall. I figurerna 4.34-36 visas modellernas frekvenssvar.



Figur 4.34 Identifierade modellernas amplitudkurva för e_2=0, 0.005, 0.01, 0.015, 0.02 då insignalen är vitt brus.



Figur 4.35 Identifierade modellernas amplitudkurva för e_2=0, 0.005, 0.01, 0.015, 0.02 då insignalen är en störd PRBS-signal.



Figur 4.36 Identifierade modellernas amplitudkurva för e_2=0, 0.005, 0.01, 0.015, 0.02 då insignalen är en PRBS-signal.

För att de regler som använder autospektrat skall vara användbara fordras att insignalspektrat i referensfallet och det undersökta fallet skall vara lika i ≈ 1-12 rad/s, och detta kunde anses gälla för alla undersökta signaler.

Att använda en PRBS eller en störd PRBS-signal ger vid identifieringen liknande resultat, men vitt brus ger ,som väntat, ett avvikande resultat. Regler som använder identifierade modeller fordrar att man har god kontroll över insignalerna. Storheten 'parameteravstånd' skall, som påpekats i 4.3.5, behandlas med stor försiktighet.

5 SIMULERINGAR MED BWR MODELLEN

5.1 Deterministiska störningar

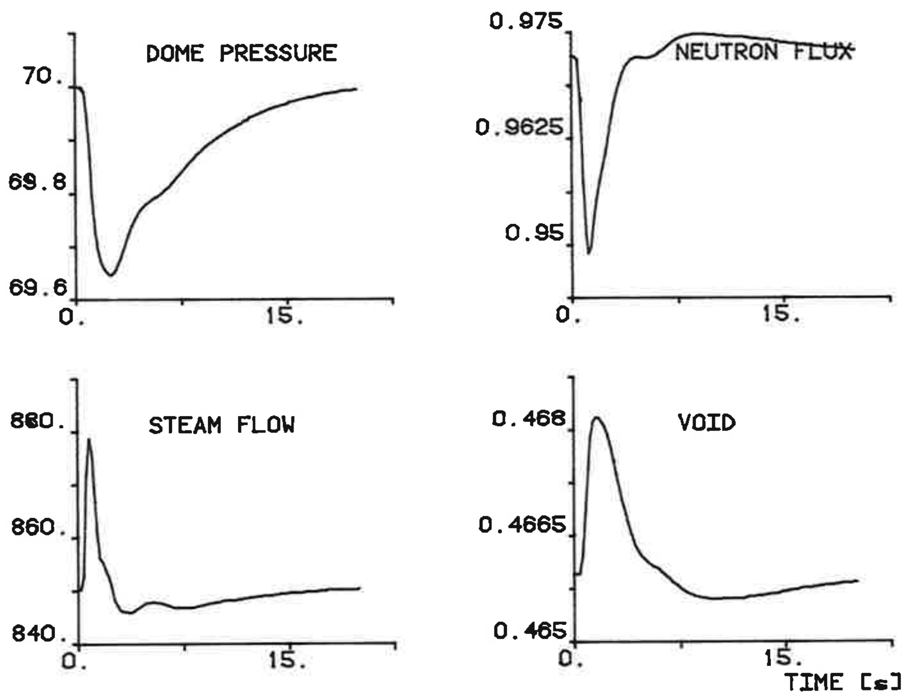
För att visa modellens uppförande ges här exempel på modellens svar vid ändringar av olika parametrar. För att lättare förstå signalernas ändringar presenteras i tabell 5.1 stationära värden vid 100% effekt för modellens viktigare signaler.

Storhet	100% -värde
Domtryck	70 bar
Ångflöde	850 kg/s
Effekt	1600 MW
Flux	1

Tabell 5.1

5.1.1 Införande av stegstörning på PI-regulatorns utgång

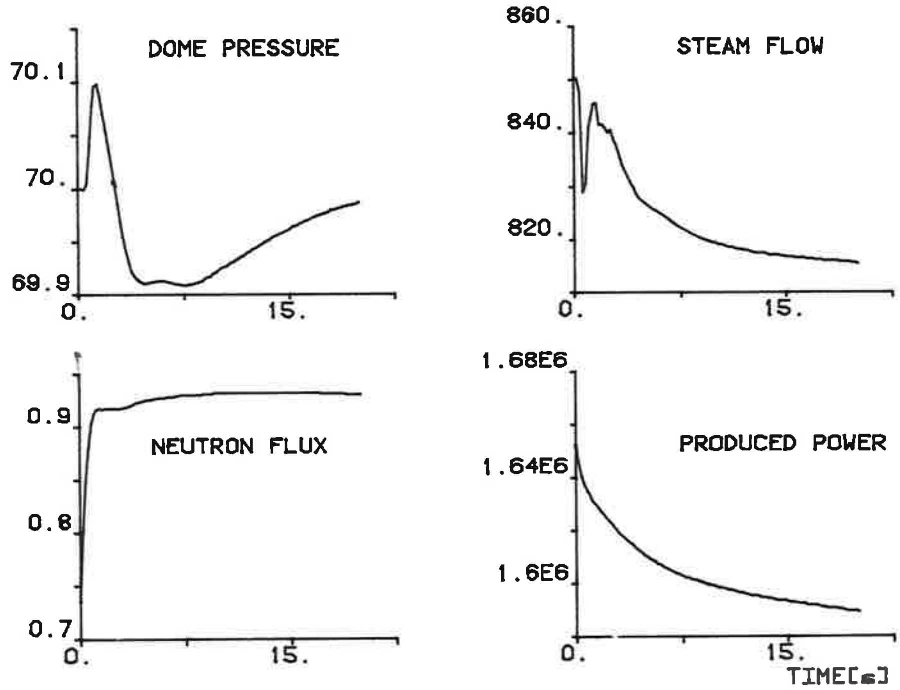
Införandet av denna störning innebär att addera en spänning till PI-regulatorns utgång. Är spänningen positiv öppnas HT-ventilen, domtrycket sjunker och ångflödet ökar, tills PI-regulatorn åter styr processen till normal arbetspunkt.



Figur 5.1 Domtryck, neutronflux och ångflöde efter en stegstörning på ehw-servots ingång.

5.1.2 Ändring av styrstavsreaktivitesten

Styrstavsreaktiviteten ges i Simnonmodellen av parametern Ross. Ross representerar mängden styrstavar i härden. Mängden styrstavar påverkar den i härden producerade effekten. Vi har här valt att skjuta in styrstavarna så att effekten sjunker från 100% till 95%.



Figur 5.2 Domtryck, neutronflux, ångflöde och producerad effekt vid införande av styrstavarna.

5.2 Stokastiska störningar

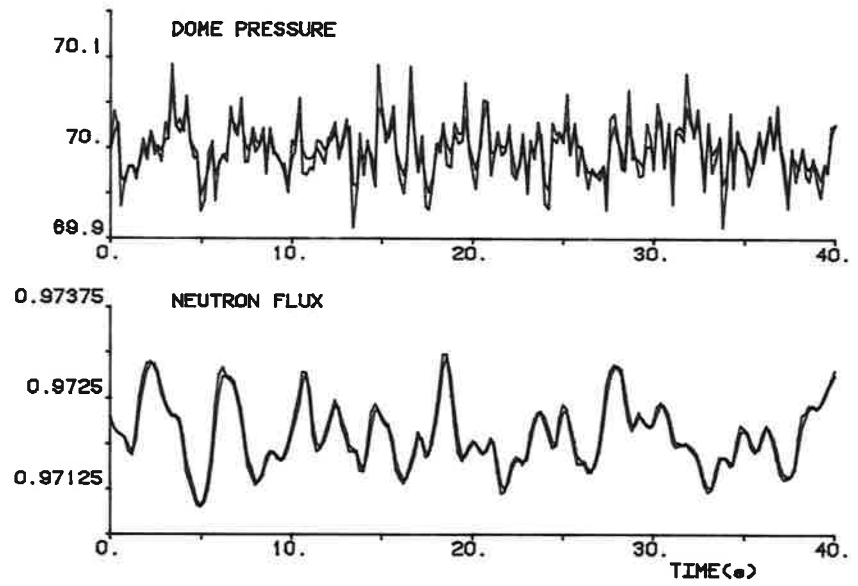
Vi demonsterar här en del brusstörningars inverkan på systemet. Signalerna presenteras dels som tidsfunktioner och dels som spektra. De brussignaler som använts här är

1. Tryckgivarbrus
2. Neutronflödesgivarbrus
3. Brus i HC pumpen

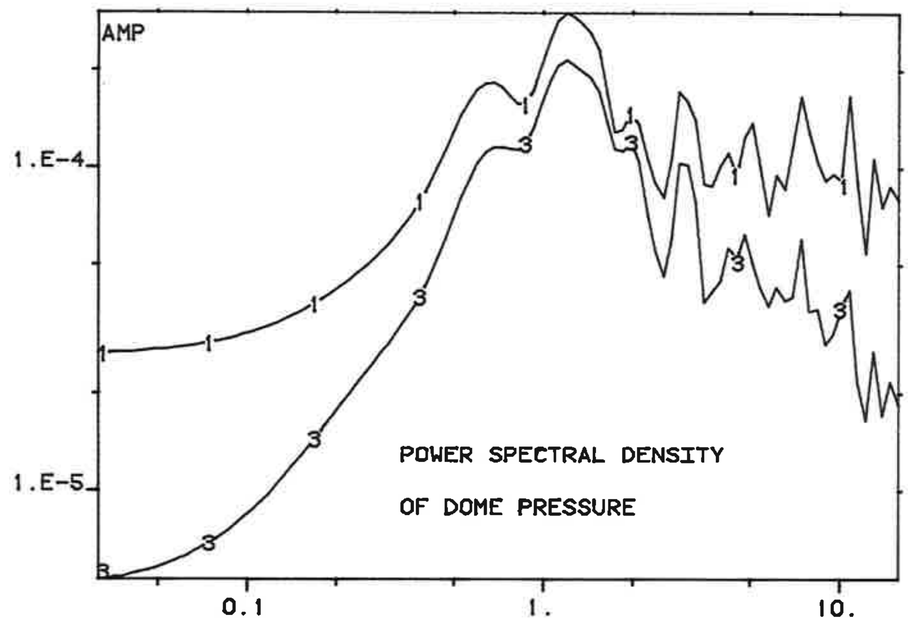
De signaler som främst kommer att analyseras vid körning med expertsystemet är neutronflödet och domtrycket. Vi demonstrerar därför brusets inverkan på just dessa signaler.

5.2.1 Tryckgivarbrus

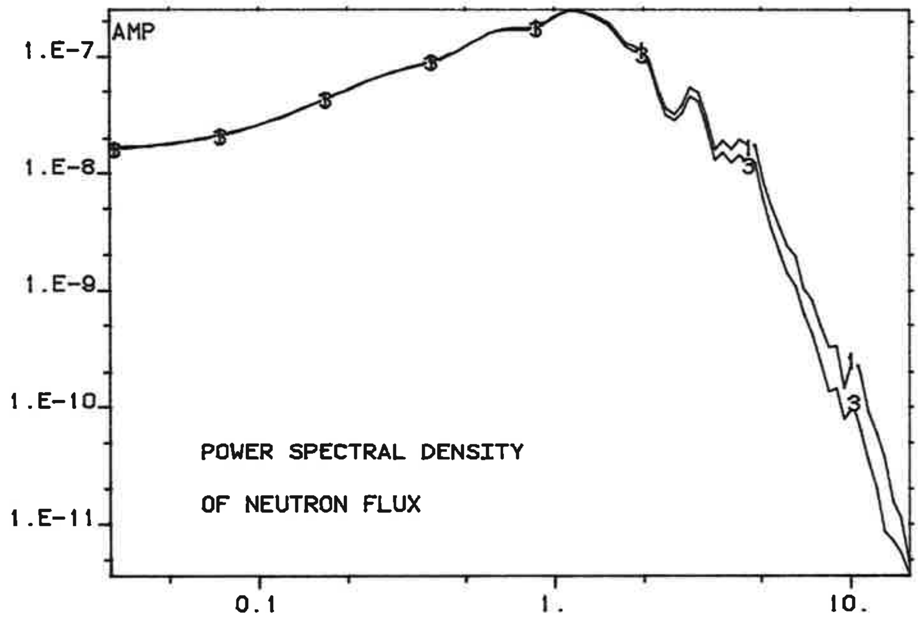
Bruset som adderades till trycksignalen från BOIL-systemet, är vitt brus som filtrerats genom ett första ordningens filter med tidskonstanten τ_{pr}. Konstanten G_p som finns i modellen anger brusets amplitud.



Figur 5.3 Uppmätt domtryck och flux vid införande av brus i en tryckgivare. 1) $\tau_{pr} = 0.05$ 2) $\tau_{pr} = 0.15$.



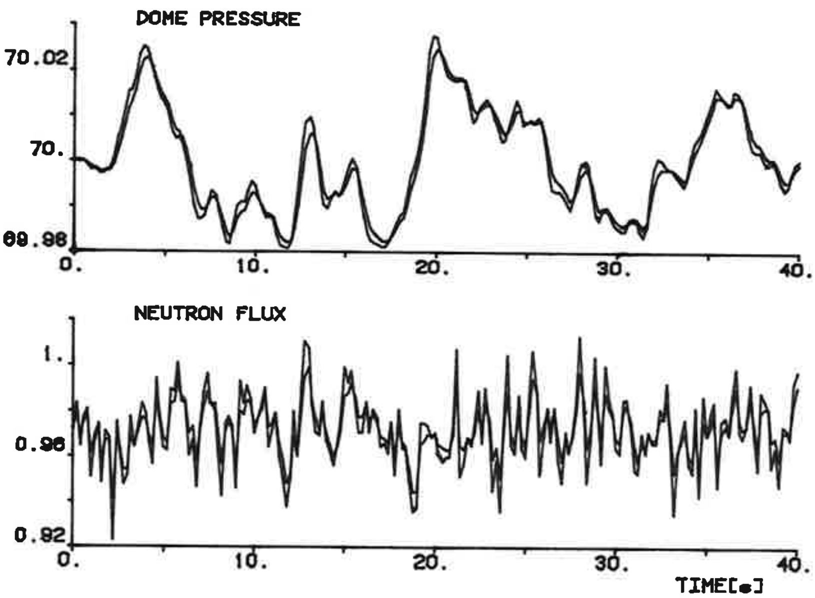
Figur 5.4 Spektrum av trycksignalerna i figur 5.3.



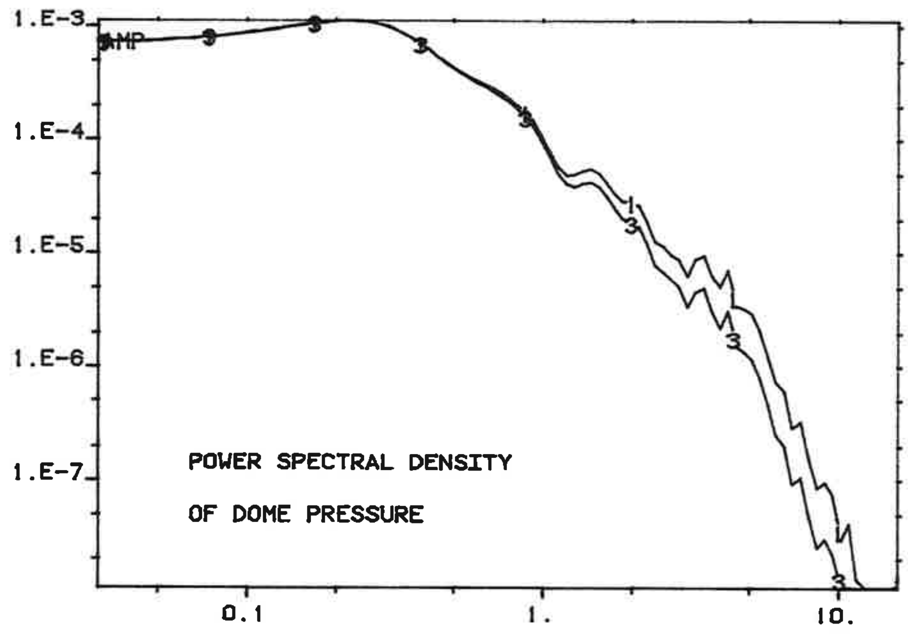
Figur 5.5 Spektrum av fluxsignalerna i figur 5.3

5.2.2 Neutronflödesgivarbrus

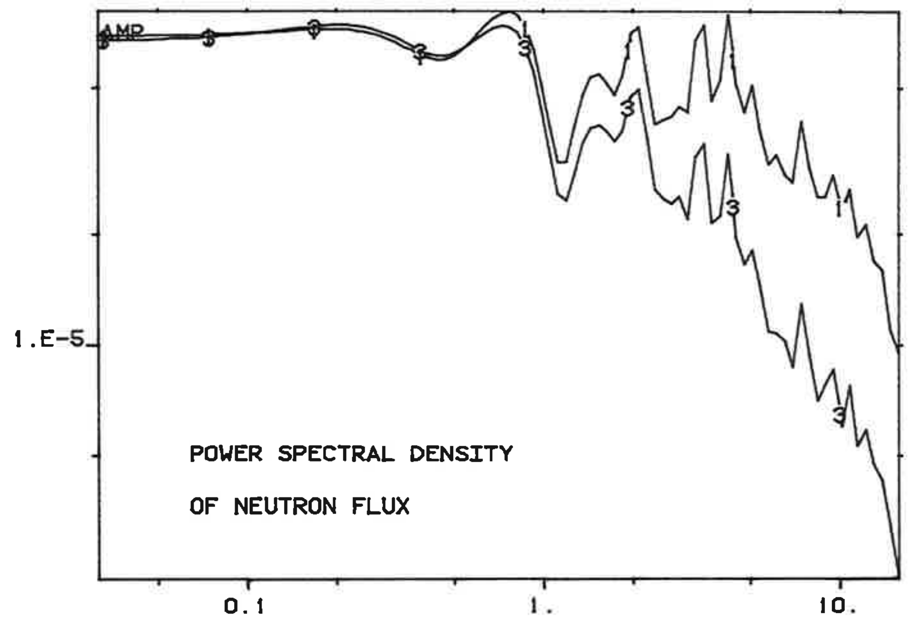
Bruset som adderades till flödessignalen från BOIL-systemet är vitt brus som filtrerats genom ett första ordningens filter med tidskonstanten τ_{fx}.Konstanten G_f anger det adderade brusets amplitud.



Figur 5.6 Uppmätt domtryck och flux vid införande av brus i en neutronflödesgivare. 1)τ_{fx} = 0.20 s 2)τ_{fx} = 0.40 s.



Figur 5.7 Spektrum av trycksignalerna i figur 5.5.



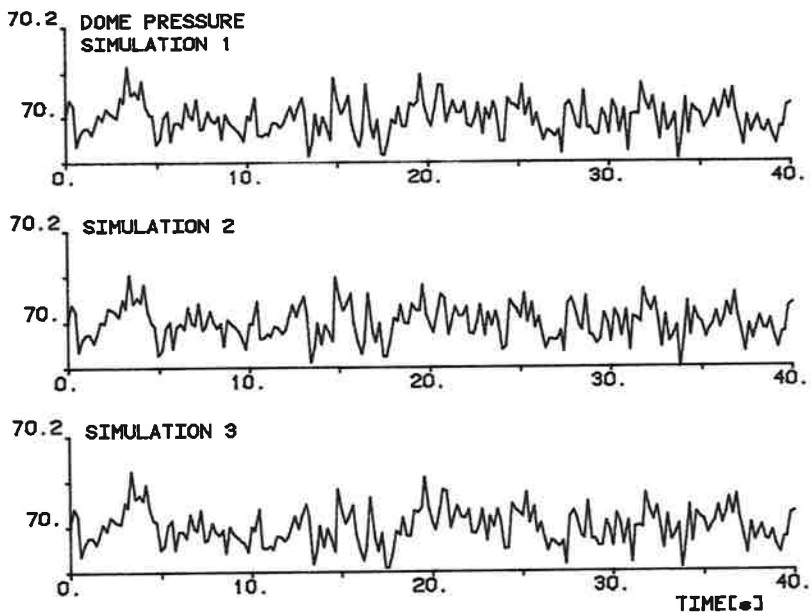
Figur 5.8 Spektrum av fluxsignalerna i figur 5.5.

5.2.3 HC-brus tryck och neutronflödesgivarbrus och glapp.

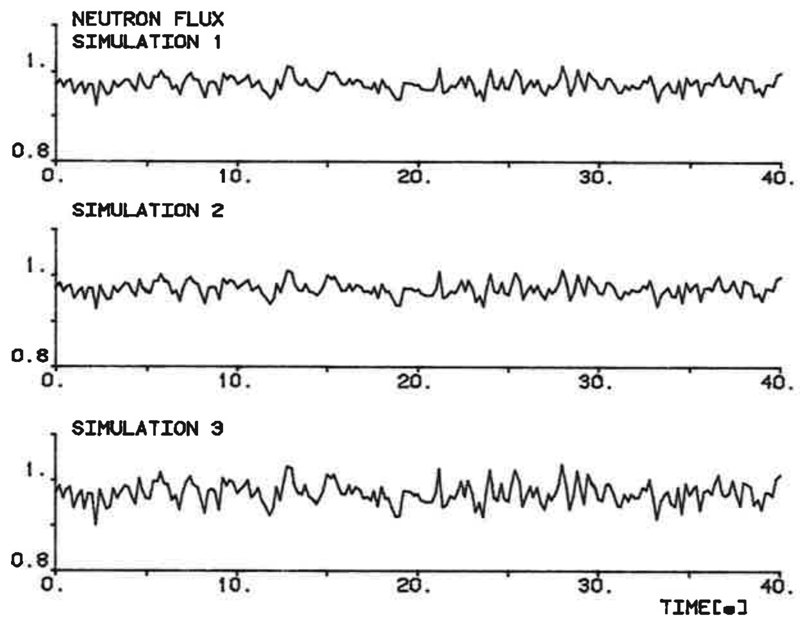
Här har utförts några simuleringar av BWR modellen med tryckgivarbrus, neutronflödesgivarbrus, brus från varvtalsvariationer i HC-pumparna samt ett glapp i ehw-servot. Aktuella parametrar i de utförda simuleringarna presenteras i tabell 5.2. Simulering 1 svarar mot de brusstörningar man kan vänta sig i ett normalt driftsfall. Simulering 2 visar effekten av ett litet glapp i ehw-servot, och i simulering 3 demonstreras effekten av förhöjt neutronflödesgivarbrus.

Simulering	τ_{pr}	τ_{fx}	G_p	G_f	Glapp
1	0.05	0.20	0.10	0.10	0
2	0.05	0.20	0.10	0.10	0.002
3	0.05	0.20	0.10	0.15	0

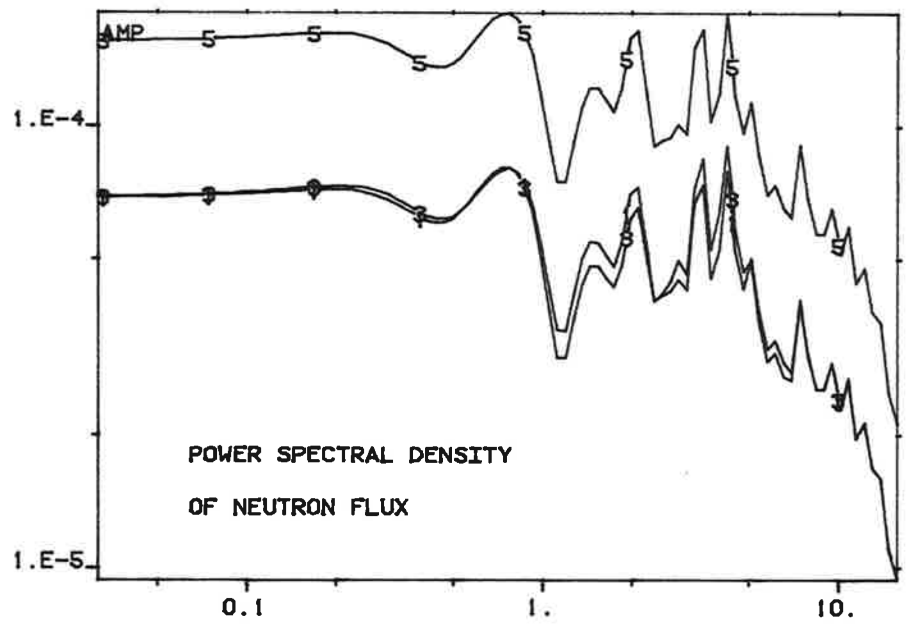
Tabell 5.1



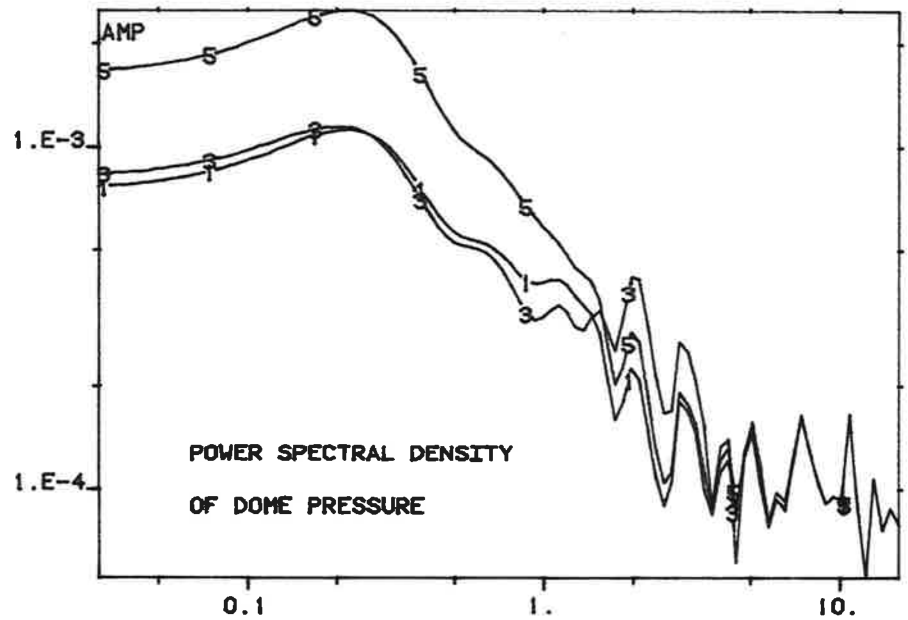
Figur 5.9 Uppmätt domtryck enligt simuleringarna 1 2 och 3.



Figur 5.10 Uppmätt flux enligt simuleringarna 1, 2 och 3.



Figur 5.11 Spektra av domtryck enligt simuleringarna 1, 2 och 3.



Figur 5.12 Spektra av neutronflux enligt simuleringarna 1, 2 och 3.

5.3 Kommentarer

Simuleringar liknande dessa kan göras med med olika felfunktioner för att testa ut feldetektreingsregler. De felfunktioner jag har testat här, glapp i servot och förhöjt givarbrus , ger sig till känna i tryck och flödessignalernas spektra. Spektra för simuleringarna med glapp uppvisar en karakreristisk resonans vid ≈2 rad/s. (Jämför kap 4!) Förhöjt givarbrus märkes genom att flödessignalens spektraltäthet ökar, medan trycksignalen påverkas ganska lite, flödessignalens varians kan också vara lämplig att studera här.

6 ON-LINE IDENTIFIERING

I kapitel 3 har olika metoder för feldetektering diskuterats. Ett sätt att behandla inkommande data är att ur dessa försöka identifiera en en modell. Skall denna identifiering utföras on line är en rekursiv algoritm att föredra. Dels får man kontinuerligt sin modell uppdaterad med ny information och dels sparas minnesutrymme.

6.1 Algoritmbeskrivning

Signalerna $y(t),u(t)$ kommer som insignaler till identifieringsalgoritmen. Från dessa önskar vi identifiera modellen

$$(1+A)\ y(t) = \left\{\frac{B}{1+F}\right\} u(t-T_d) + \left\{\frac{1+C}{1+D}\right\} e(t) \qquad (1)$$

där $e(t)$ är vitt brus, och A,B,C,D är polynom i skiftoperatorn q^{-1} ($q^{-1}y(t)=y(t-1)$). Genom att sätta $F=D=0$ får man den modell som erhålles vid maximum likelihood identifiering, nämligen

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

där

$$A(q^{-1}) = a_1 q^{-1} + \dots + a_n q^{-n} \qquad (3)$$

$$B(q^{-1}) = b_1 q^{-1} + \dots + b_m q^{-m} \qquad (4)$$

$$C(q^{-1}) = c_1 q^{-1} + \dots + c_k q^{-k} \qquad (5)$$

För identifieringen inför vi nu parametervektorn $\theta(t)$ och mätvärdesvektorn $\varphi(t)$ enligt

$$\varphi(t)=\begin{bmatrix} -y(t-1) \\ \vdots \\ -y(t-n_a) \\ -z(t-1) \\ \vdots \\ -z(t-n_f) \\ u(t-T_d-1) \\ \vdots \\ u(t-T_d-n_b) \\ -v(t-1) \\ \vdots \\ -v(t-n_d) \\ e(t-1) \\ \vdots \\ e(t-n_c) \end{bmatrix} \quad (6) \qquad \theta(t)=\begin{bmatrix} a_1 \\ \vdots \\ a_{na} \\ f_1 \\ \vdots \\ f_{nf} \\ b_1 \\ \vdots \\ b_{nb} \\ d_1 \\ \vdots \\ d_{nd} \\ c_1 \\ \vdots \\ c_{nc} \end{bmatrix} \quad (7)$$

där F och D definieras ur

$$(1 + F)\ z = B\ u \tag{8}$$

$$(1 + D)\ v = (1 + C)\ e \tag{9}$$

Systemet skrives nu

$$y(t) = \theta(t)^T \varphi(t) + e(t) \tag{10}$$

Och identifieringen utföres med algoritmen

$$\lambda(t) = \lambda_o \lambda(t-1) + (1 - \lambda_o) \tag{11}$$

$$P(t) = \left[P(t-1) - \frac{P(t-1)\varphi(t-1)\varphi^T(t-1)P(t-1)}{\lambda(t) + \varphi^T(t)\ P(t-1)\ \varphi(t)} \right] / \lambda(t) + \beta(t-1) I \tag{12}$$

$$e(t) = y(t) - \theta^T(t-1)\varphi(t) \tag{13}$$

$$\theta(t) = \theta(t-1) + P(t)\varphi(t)e(t) \tag{14}$$

där $\beta(t)$ väljes enligt

$$w(t) = \gamma_1 w(t-1) + \Delta\hat{\theta}(t) \tag{15}$$

$$\Delta\theta(t) = \theta(t) - \theta(t-1) \tag{16}$$

$$s(t) = \mathrm{sign}\,(\Delta\theta(t)^T v(t-1)) \tag{17}$$

$$r(t) = \gamma_2 r(t-1) + (1 - \gamma_2)s(t) \tag{18}$$

<u>Om</u> $r(t)) \delta$ <u>så</u> $\beta(t+1) := \dfrac{1}{\varphi(t)^T \varphi(t)} (v_0(t) - v(t))$

<u>annars</u> $\beta(t+1) := 0$ (19)

där

$$v_0(t) = \frac{\lambda(t)}{\lambda(t) + \varphi(t)^T P(t-1)\varphi(t)} \tag{20}$$

För närmare beskrivning av denna algoritm, där även förslag till val av $v(t)$ göres se [10].

6.2 Program

Ovanstående formler har implementerats i ett Pascalprogram, vilket finns i Appendix 3. Formlerna ovan beskrives i programmet av proceduren recid. Hela programmet test gör det möjligt att testa proceduren för olika insignalsekvenser. För detta skrevs ett program som genererade en datafil med in och utsignaler från ett diskret system. Identifieringsprogrammet läste sedan data från denna fil och försökte identifiera modellen som genererat dessa data. Ett program som utför denna datagenetrering också i Appendix 3. All nödvändig information om programmen finns som kommentarer i programkoden.

Meningen är att proceduen recid skall kunna användas i andra program. Det är då noga att alla globala variabeldefinitioner utföres och att parametrarna får rätta värden innan recid anropas. Varje gång recid anropas, med $u(t)$ och $y(t)$ som inparametrar, kommer mätvektorn φ att uppdateras och en ny parametervektor θ att beräknas.

7 SLUTSATSER

Jag har testat och kompletterat en given simnonmodell av en BWR. Denna fungerar nu väl vid 100% effekt. En av kompletteringarna var en modell av ett elektrohydrauliskt servo. I detta servo infördes diverse felfunktioner och åtskilliga simuleringar har gjorts för att få underlag för att föreslå regler för detektering av dessa felfunktioner. Är man medveten om vilka felfunktioner som kan uppstå så är det möjligt att med ett omfattande simuleringsarbete (och fysikalisk intuition) ställa upp liknande regler för hela BWR modellen. I kapitel 5 antyds vissa möjligheter. Emellertid återstår mycket arbete med att införa nya felfunktioner i modellen och analysera simuleringsresultat. Det i kapitel 6 beskrivna pascalprogrammet fungerar väl och kan användas i expertsystemet efter anpassning till APPLE-systemet.

8 REFERENSER

[1] Bergman S;"Development of a reactor noise monitor based on artificial intelligence and recursive parameter estimation." ;Paper presented at 15:th Informal Meeting on Reactor Noise, Petten, Holland May, 1982.

[2] Bergman S, Åström K J;" Fault detection in Boiling water reactors by noise analysis." CODEN:LUTFD2/(TFRT-7250)/1-021/(1983)

[3] Bergman S;"Reactor noise surveillance by parameter estimation and pattern recognition methods" CODEN:LUTFD2/(TFRT-7252)/0-034/(1983)

[4] Blomberg P E, Espefält R, Lorenzen J, Åkerhielm F; "Brusanalys för systemövervakning." STUDSVIK/RR-78/15

[5] Wittenmark B, Åström K J;"Computer control theory."

[6] Wieslander J;"Idpac commands - Users Guide." CODEN:LUTFD2/(TFRT-3157)/1-108/(1980)

[7] Åström K J;"A Simnon tutorial." CODEN:LUTFD2/(TFRT-3168)/1-052/(1982)

[8] Bergman S; Reaktorreglering med Instrumentering Chalmers Tekniska Högskola, Göteborg.

[9] Rasmusson U;"Modellering och simulering av dynamik hos en kokarreaktor." CODEN:LUTFD2/(TFRT-5295)/ /(1983)

[10] Hägglund T;"Adaptive control with fault detection." CODEN:LUTFD2/(TFRT-7242)/1-034/(1982)

APPENDIX 1 SIMNON OCH IDPAC

Simnon och Idpac är två programpaket som är utvecklade vid Institutionen för Reglerteknik vid LTH. Programpaketen är avsedda att användas interaktivt för simulering, identifiering och analys av data.

<u>Simnon</u> De system man vill simulera i Simnon beskrives antingen i kontinuerlig form som ett system av ordinära differentialekvationer eller i diskret form som ett system av differensekvationer. Dessa system kallas i Simnon COTINUOUS SYSTEM och DISCRETE SYSTEM. En modell kan bestå av av flera undersystem, där det är tillåtet att använda både kontinuerliga och diskreta system samtidigt. Delsystemen knyts samman i ett CONNECTING SYSTEM. I varje system definieras tillstånden, deras derivator (om det är ett kontinuerligt system) eller "nästa värde" (om det är ett diskret system) med STATE, DER eller NEW. En variabel som svarar mot tiden kan definieras med TIME. I systemen kan sedan ekvationerna skrivas med vanliga beteckningar. Diverse standardfunktioner och olinjäriteter är också tillåtna. Variabler åtföljda av ": tal" kallas parametrar och kan ändras med ett kommando vid körningen av systemet. En mängd kommandon finns för hantering av filerna, simulering och presentation a resultaten. Jag beskriver här de viktigaste.

ASHOW
Detta kommando har samma funktion som SHOW, men skalar axlarna automatiskt.

AXES [specifikation av axlarna]
Ritar upp axlar enligt specifikationerna.

INIT <tillstånd>:{tal}
Sätter initialvärdet hos ett tillstånd till det angivna värdet.

PAR <parameter>:{tal}
Tilldelar parametern det angivna värdet.

SHOW {variabel}/{filnamn}
Ritar lagrade variabler från en fil. Variablerna har lagrats med STORE.

SIMU <start> <stopp>/{filnamn}
Simulerar det kompilerade systemet från t=<start> till t=,stopp> och lagrar värdena på filen.

STORE {variabel}
Lagrar variablerna på filer för att man sedan skall kunna visa dessa med SHOW eller ASHOW.

SYST {filnamn}
Definierar systemet och kompilerar filerna i argumentet.

För mer detaljerad information om Simnon kommandon se [5] eller [7]. Flera kommandon kan sättas samman till ett MACRO. Alla dessa kommandon utföres då när macrot anropas. Man är då säker på att man gör samma kommandosekvens vid varje simulering.

Idpac Idpac är liksom Simnon ett interaktivt kommandostyrt programpaket. Idpac ger användaren möjlighet att utifrån indata identifiera modeller, utföra statistiska analyser, beräkna signalers spektrum etc. Till Idpac kommer data i form av datafiler, som kan vara genererade i Simnon med kommandot STORE, dessa data kan sedan behandlas med olika kommandon. Som resultat av behandling i Idpac får man datafiler eller modeller i form av textfiler. Datafilerna kan sedan plottas i olika format i Idpac. De kommandon vi använt oss av är främst

ASPEC

Som indata till detta kommando ger man en tidsserie, Idpac beräknar seriens autospektrum med en formel liknande (3.5), och levererar resultatet i form av en frekvensfil.

BODE

Detta kommando plottar en frekvensfil i form av ett bodediagram.

CCOF

Detta kommando beräknar korskorrelationen mellan två infiler.

ML

Detta kommando utför identifiering enligt maximum likelihood metoden med en insignalfil och en utsignalfil vilka ges som argument. Resultatet blir modellen i form av en textfil.

RESID

Givet utsignalfil, insignalfil och en modell beräknar Idpac residualerna enligt (3.8) och utför diverse statistisk analys på dem.

SPTRF

Kommandot beräknar frekvenssvaret enligt (3.7) för en modell som ges som argument.

TREND

Kommandot tar bort en trend från en infil. Trenden kan vara ett polynom av grad 0, 1, 2 eller 3.

Vidare har en del filhanteringskommandon använts. Liksom i Simnon kan flera kommandon sättas samman till ett macro med repetitionssatser, villkorssatser etc. Speciellt användbart i Idpacmacron är möjligheten att numrera filer i en repetitionsats. En närmare beskrivning av Idpac finns i [6].

APPENDIX 2 SIMNONMODELL AV EN BWR

```
CONTINUOUS SYSTEM BOIL

"File:BOILBOX.T

"Author:STEN BERGMAN
"       SYDKRAFT AB
"       FACK
"       S-217 01 MALMOE
"       SWEDEN


"Description:
"            Models the termohydraulics of a boiling
"            water reactor. The model is a global simplified
"            point model containing modes for:
"            Steam quality dynamics
"            Reactor dome pressure dynamics

"Assumptions:
"            1. A linear steam-quality distribution along the
"               boiling section
"            2. A linear void-steam/quality function
"            3. A perfect feedwater control
"

INPUT Qheat Qstm Nox Nhcp

OUTPUT Void Qp Z1

STATE Xr Pr Nof Qhc  Tdc

DER Dxr Dpr Dnof Dqhc Dtdc

"Inputs:     Qheat   Heat flow from fuel               [KW]
"            Qstm    Steam outlet flow                 [kg/s]
"            Nox     Boiling boundary noise input      [M]

"States:     Xr      Steam quality                     [PU]
"            Pr      Reactor dome pressure             [BAR]
"            Nof     Delayed Nox                       [M]

Dnof=(Nox*Gb/3-Nof)/Tau

Ts=104.409*Pr^0.237
Xdc=((Qhc-Qstm)*Ts+Qstm*Tfn)/Qhc
Zx=Qhc*Cp*Gamma*(Ts-Tdc)/Qheat
A1=(Xr*(Row-Ros)+Ros)/Ros
Z1=MIN(MAX(Zx,0),1)
A2=Ros*LN(A1)/(Xr*(Row-Ros))
A3=Row*(1.-A2)/(Row-Ros)
Void=A3*(1.-Z1)

Dtdc=(Xdc-Tdc)/Tpc
```

```
Dpr=(Qhc*Xr*Gamma-Qstm+Nof)/(Vs*Drdp)

Dadx=A0*EXP(B0*Xr)
Z1=Qheat-Hc*Xr*Qhc*Gamma-Qhc*Cp*(Ts-Tdc)
Z2=(Hc*(1.-Xr)+Ros*Dhsdr)*Void*Vr*Drdp*Dpr
T1=Hc*((1.-Xr)*Ros+Xr*Row)*Vr*Dadx
Zxr=(Z1-Z2)/T1

Dxr= IF Xr ( Eps THEN( IF Zxr(0 THEN 0. ELSE Zxr) ELSE Zxr

"Mass-flow calculations
Qp=Qhc*Gamma*Xr
Dph=W0*Nhcp*Nhcp+W1*Nhcp*Qhc/4+W2*Qhc*Abs(Qhc)/16
Dpb=Void*G*L*C*Row
Dqhc=(Dph+Dpb-(k1+k2+k3*(1+30*Xr))*Qhc*Qhc)/Tauh

"Parameters:

Gb:0            "Noise factor
Gamma:0.9       "Box flow factor
Cp:4.22         "Specific entalpy of water
Row:740.03      "Density of water
Ros:36.53       "Density of steam
Hs:2773.        "Entalpy of steam
Hw:1276.        "Entalpy of water
Hc:1497         "Condensation entalpy
L:3.71          "Core flow lenght          [M]
Vr:16.87        "Core volume               [M3]
Vs:105.02       "Steam dome volume         [M3]
G:9.815         "Gravitation constant      [M/S2]
Eps:1.E-8       "Test quantity
Drdp:.59        "Steam density der. resp. to pressure
Dhsdr:-1.3      "Entalpy der. resp. to density
A0:6.9191       "
B0:-9.2875      "
Lambda:0.       "Noise factor              [M]
Tfn:180.        "Feedwater temperature     [deg C]
Tau:0.5         "                          [s]
Tpc:10          "
C:1.01325E-5    "1 Pascal/1 Bar
W0:1.966E-6     "Pumpconstant
W1:1.647E-6     " --"--
W2:-1.535E-6    " --"--
Tauh:0.01       "
k1:1E-8         "
k2:1E-8         "
k3:0.455E-8     "

END
```

```
CONTINUOUS SYSTEM CORE

"File:CORE.T

"Version 2:82-10-21

"AUTHOR:Sten Bergman


"Description:Models the neutron kinetics of a boiling water
"             reactor as a point model with one meangroup
"             of precursors.
"             The reactivity feedback is calculated from
"             fuel-temperature void- and xenon poisioins.
"             Input parameters rod-rectivity.

INPUT Void Tmod
OUTPUT Ron Flux

STATE  C Qheat Tf
DER    Dc Dqhe Dtf

"Input:     Void    Mean void content in boiler           [pu]
"           Tmod    Moderator temperature                 [deg C]

"States:    C       Concentration of precursors           [pu]
"           Qheat   Heat transfer power to coolant        [kW]
"           Tf      Fuel mean temperature                 [deg C]

Ron=Rof-Roxe-Ross+gamma1*Void*100.+gamma2*Tmod+gamma3*Tf
Ro=Ron*K

Dc1=lam*C*Ro/(beta-Ro)
Dc=if c<0 then (if Dc1<0 then 0 else Dc1) else Dc1
Flux1=Lambda*lam*C/(beta-Ro)
Flux=min(max(Flux1,0.02),2)
Dqhe=(Flux*Kh-Qheat)/Tauh
Dtf=(Flux*Kb+Tf0-Tf)/Tauf


"Parameters:

gamma1:-180.0        "Void coefficient                     [pcm/%]
gamma2:-20.          "Moderator temperature coeff          [pcm/deg C
gamma3:-2.5          "Doppler coefficient                  [pcm/deg C
beta:7.5e-3          "Delayed neutron fraction             [pu]
Lambda:1.0e-3        "Generation time for prompt neutrons  [sec]
lam:7.9648e-2        "Mean group decay constant            [sec]
Roxe:2570.           "Xenon poisoning                      [pcm]
Rof:1.33e5           "Fuel reactivity                      [pcm]
K:1.0e-5             "Scale constant
Tauf:1.0             "Fuel time constant                   [sec]
Tauh:10.0            "Heat transfer time constant          [sec]
Kb:800.              "Fuel temp constant                   [deg C]
```

```
Kh:1.7e6           "Heat flow constant                  [kW]
Tf0:300.0          "Zero power temperature              [des C]
Ross:113960        "Control rod reactivity              [pcm]
Nc:0.0             "Noise constant

END
```

```
CONTINUOUS SYSTEM EHW

"File:EHW4.T

"Authors: Sten Bergman
"
"         Per Persson

"Description: Models an electrohydaulic servo with nonlinear
"             feedback.The model also includes a play which
"             size can be changed with the parameter e2.

"Input:       y7: The signal from the regulator
"        ehwnois: white noise to disturb measurements


INPUT y7 ehwnois

OUTPUT htl ehwl

STATE y1  y2  y3  x

DER  dy1 dy2 dy3 dx

"Function nr 1  to simulate friction
xh=y7-y1
F1=if xh<-eps1 then xh+eps1 else if xh>eps1 then xh-eps1 else 0

"Function nr 2  to simulate friction
xg=y1-x
F2=if xg<-eps2 then xg+eps2 else if xg>eps2 then xg-eps2 else 0

"Function nr 3
G1=k1*y2+l1
G2=k2*y2+l2
G3=k3*y2+l3
G4=a0*y2*y2*y2+a1*y2*y2+a2*y2+a3
G7=k7*y2+l7
h1=if (y2<I1) then G1 else G2
h2=if (y2>I1) then G2 else h1
h3=if (y2>I2) then G3 else h2
h4=if (y2>I3) then G4 else h3
h7=if (y2>I6) then G7 else h4
F3=h7*Q-N

"Introduction of a nonlinear function:

up=dy3

a=y3+e2
b=y3-e2
c=a+e1
d=b-e1
p1=if  x<a and  x>b  then 1 else 0
p2=if  x>d and  x<a and up<0 then 1 else 0
```

```
p3=if  x<c and  x>b and up>0 then 1 else 0

dx=if p1 then 0 else if p2 then 0 else if p3 then 0 else up


dy1=F1/T1

dy2=F2/T2

dy3=(F3-y3)/T3

ehwl=y1+ehwnoise

htl=y2

"Parameters:

a0:11.240143369   "Coefficients for the nonlinear function:
a1:-20.120430107
a2:12.24265239
a3:-1.547849462

k1:0.51612854
k2:0.07373701
k3:1.93157327
k7:0.04435495

l1:0.35483899
l2:0.37764943
l3:0.08111986
l7:0.95564502

I1:0.05156163
I2:0.15907615
I3:0.37500000
I6:0.60000000

Q:1.55
N:.55
T1:0.05           "Time constants
T2:0.1
T3:0.14

eps1:0.0          "Friction constants
eps2:0.0

e1:0              "Constants for nonlinear functions
e2:0

END
```

```
CONTINUOUS SYSTEM MEASURE

"File: MEASURE.T

"Author: Per Persson

"Descripion:   This  system simulates the measurements of
"              the steamflow and the circulationpumpflow.
"              The  limited   bandwidth  of  the  sensors
"              is  implemented   with  lowpass   filters.

INPUT  Qh  Qst

OUTPUT Qhc Qstmeas

STATE Y Y1 Z Z1

DER  DY DY1 DZ DZ1

DY=omega*omega*Qh-2*ksi*omega*Y-omega*omega*Y1
DY1=Y
Qhc=Y1

DZ=c1*Qst-c2*Z-c1*Z1
DZ1=Z
Qstmeas=Z1

"Parameters:
omega:1.93          "Resonance frequency [rad/s]
ksi:0.3             "Relative damping [pu]

c1:24.8756
c2:18.1592

END
```

```
MACRO PC100  a b e d

"File:PC100.T

"a=boilbox
"b=core
"e=logger
"d=noise1

let n.noise1=4
let nodd.noise1=177653

syst pcon3 ehw4 st1 st2 a b e d valve measure  pump  pct

par dt[logger]:0.2

par dt[noise1]:0.1
par same[noise1]:1.0

init  pr[boil]:70.
init  Xr[boil]:0.17165
init  Qhc[boil]:5503.7
init  Tdc[boil]:269.44
init  Nof[boil]:0.0

init  y1[steam1]:425.13
init  y3[steam1]:425.13
init  y5[steam1]:425.13
init  y7[steam1]:425.13
init  y9[steam1]:425.13

init  y2[steam1]:68.999
init  y4[steam1]:67.999
init  y6[steam1]:66.998
init  y8[steam1]:65.998
init  y10[steam1]:64.997

init  y1[steam2]:425.13
init  y3[steam2]:425.13
init  y5[steam2]:425.13
init  y7[steam2]:425.13
init  y9[steam2]:425.13

init  y2[steam2]:68.999
init  y4[steam2]:67.999
init  y6[steam2]:66.998
init  y8[steam2]:65.998
init  y10[steam2]:64.997

init  pm[pcon3]:70.
init  y1[pcon3]:-0.0513452
init  y2[pcon3]:-0.0924216
init  y3[pcon3]:-0.0513452
init  y4[pcon3]: 0.8747000
init  y5[pcon3]:0.0
```

```
init  y6[pcon3]:0.0

init  y1[ehw]:0.92054
init  y2[ehw]:0.50064
init  y3[ehw]:0.92054
init  x[ehw]:0.92054

init  C[core]:91.517
init  Qheat[core]:1.65221E6
init  Tf[core]:1077.5

init  z[measure]:0
init  z1[measure]:850.26
init  y[measure]:0
init  y1[measure]:5503.7

init  y[pump]:0
init  y1[pump]:0

algor rk

error 0.0001

store  pr[boil] void[boil] Qstm[boil]
store Qstmeas[measure] Qhc[measure] -add
store y7[pcon3] Ffx[pcon3] Flx[pcon3] prmea[pcon3]-add
store htl1[valve] htl2[valve] ehwl-add

par amp:0.0

END
```

```
MACRO PC20 a b e d

"File:PC20.T

"a=boilbox
"b=core
"e=logger
"d=noise1
" System test of pressure control system
let n.noise1=4
let nodd.noise1=177653

syst pcon3 ehw4 st1 st2 a b e d valve measure  pump  pct

par dt[logger]:0.2

par dt[noise1]:0.1

init  pr[boil]:70.
init  Xr[boil]:0.0314722
init Qhc[boil]:5994.7
init Tdc[boil]:282.78
init Nof[boil]:0.0

init y1[steam1]:84.898
init y3[steam1]:84.898
init y5[steam1]:84.898
init y7[steam1]:84.898
init y9[steam1]:84.898

init y2[steam1]:69.960
init y4[steam1]:69.920
init y6[steam1]:69.880
init y8[steam1]:69.840
init y10[steam1]:69.800

init y1[steam2]:84.898
init y3[steam2]:84.898
init y5[steam2]:84.898
init y7[steam2]:84.898
init y9[steam2]:84.898

init y2[steam2]:69.960
init y4[steam2]:69.920
init y6[steam2]:69.880
init y8[steam2]:69.840
init y10[steam2]:69.800

init pm[pcon3]:70.
init y1[pcon3]:0.21822
init y2[pcon3]:0.39279
init y3[pcon3]:0.21822
init y4[pcon3]:0.17470
init y5[pcon3]:0.0
init y6[pcon3]:0.0
```

```
init y1[ehw]:0.41233
init y2[ehw]:0.17147
init y3[ehw]:0.41233
init x[ehw]: 0.41233

init     C[core]:18.279
init Qheat[core]:3.30E5
init    Tf[core]:455.29

init  z[measure]:0
init z1[measure]:169.79
init  y[measure]:0
init y1[measure]:5994.5

init  y[pump]:0
init y1[pump]:0

algor rk
error 0.0001

store  pr[boil] void[boil] Qstm[boil]
store Qstmeas[measure] Qhc[measure] -add
store y7[pcon3] Ffx[pcon3] Flx[pcon3] prmea[pcon3]-add
store htl1[valve] htl2[valve] ehwl-add

par amp:0
par ross:120358

END
```

```
MACRO PCD100

"Calls the Macro PC100  a b c d  with the correct parameters.
"PCDNEW assumes a preassuredrop of 1 bar/node.The two pipes are
"interacting.The new ehw-model is used.

PC100  boilbox core noise1 logger

END
```

```
MACRO PCD20

"Calls the Macro PC20  a b c d  with the correct parameters.
"PCDNEW assumes a preassuredrop of 1 bar/node.The two pipes are
"interacting.The new ehw-model is used.

PC20  boilbox core noise1 logger

END
```

```
CONTINUOUS SYSTEM PCON3

"File: PCON3.T

"Author:  S.Bergman
"         SYDKRAFT AB
"         217 01 MALMOE, SWEDEN


"Description:   A revised model of the pressure control
"               system in BVT

"Variables:     Prmea:measured preassure
"               Piout:output from PI-regulator
"               Ffx:measured fluxlevel
"               y7:output from regulator

INPUT P1 Flx Pnois Flnois

OUTPUT Piut Paf Out Y7 Ffx Prmea

STATE Pm Y1 Y2 Y3 Y4 Y5 Y6

DER Dpm Dy1 Dy2 Dy3 Dy4 Dy5 Dy6

"Inputs:   P1     Dome pressure [bar]
"          Flx    Neutron flux  [pu]
"          Pnois  Pressure sensor noise  [bar]
"          Flnois Neutron flux noise [%]


Prmea=Pm+Y6
x1=(Pm+Y6-Pref)/10.
Piut=k1*x1+Y1
Paf=Y3
Ffx=Flx+Y5
Out=Ffx-(alfa*Ffx-Y4)
xz=Out+Paf
Y7=min(max(Xz,0.),1.)

Dpm=(P1-Pm)/Tr

Dy1=k1*x1/Ti

Dy2=Piut*b2-a2*Y3
Dy3=Y2+b1*Piut-a1*Y3

Dy4=(alfa*Ffx-Y4)/Tb

Dy5=(Flnois*Gf/3-Y5)/Taufx

Dy6=(Pnois*Gp/3-Y6)/Taupr

"Parameters:
a1:2.80
```

```
a2:4.00
b1:1.00
b2:4.00
k1:0.85
Ti:5.0
alfa:0.9
Pref:70.0
Tr:0.2
Tb:10.0
Gp:0.0      "Pressuresensornoiseconstants:
Taupr:0.05
Gf:0.0      "Fluxsensornoiseconstants:
Taufx:0.2

END
```

```
CONNECTING SYSTEM PCT

"File:PCT3STM.T

"Version: 1982-09-08

"Description:  Connecting of two steamlines, dome
"              pressure control, ehw servo and a
"              thermodynamic model of the boilingprocess.


TIME t

Y7[ehw]=Y7[pcon3]+Dist*cos(2*pi*f*t)
ehwnois[ehw]=Gehw*E4[noise1]/3

Htl1[valve]=Htl[ehw]
Htl2[valve]=Htl[ehw]
Y101[valve]=Y10[steam1]
Y102[valve]=Y10[steam2]

Po[steam1]=Pr[boil]
Fi1[steam1]=Fi1[valve]

Po[steam2]=Pr[boil]
Fi2[steam2]=Fi2[valve]

P1[pcon3]=Pr[boil]
Flx[pcon3]=Flux[core]
Pnois[pcon3]=E1[noise1]
Flnois[pcon3]=E2[noise1]

Qstm[boil]=Fiout[steam1]+Fiout[steam2]
Nhcp[boil]=Nhcp[pump]
Qheat[boil]=Qheat[core]
Nox[boil]=0

Void[core]=Void[boil]
Tmod[core]=Tdc[boil]

Qst[measure]=Qstm[boil]
Qh[measure]=Qhc[boil]

Pumpnois[pump]=E3[noise1]

"Parameters:

f:0.0           "Frequency of disturbance
Dist:0.0        "Amplitude of distubance
Pi:3.141592
Gehw:0          "Noiseconstant

END
```

```
CONTINUOUS SYSTEM PUMP

"File:PUMP.T

"Author:Per Persson

"Description: Models the moise of the circulationpumpspeed

INPUT Pumpnois

OUTPUT Nhcp

STATE Y Y1

DER DY DY1

DY=c1*c1*Pumpnois-2*c1*c2*Y-c1*c1*Y1
DY1=Y

Nhcp=Nx*(1+amp*Y1)

"Parameters:

c1:1         "Resonance frequency                    [rad/s]
c2:0.02      "Damping
amp:0.01     "relative noise amplitude
Nx:1000      "Pump speed                              [rpm]

END
```

```
CONTINUOUS SYSTEM STEAM1

"File:ST1.T

"Version 2:82-11-23

"Author:STEN BERGMAN
"       SYDKRAFT AB


"Description:
"           A simplified nonlinear
"           5-node steam-line model

INPUT Po  Fi1
OUTPUT  Fiout

STATE Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10
DER D1 D2 D3 D4 D5 D6 D7 D8 D9 D10

"Inputs: Po    Pressure in reactor dome        [BAR]
"        Fi1   steamoutlet from the pipe       [KG/S]
"Outputs:Fiout Steamoutlet from reactor        [KG/S]

"States: Y1 Y3 Y5 Y7 Y9   Intermediate steam flow  [KG/S]
"        Y2 Y4 Y6 Y8 Y10  Node pressure            [BAR]


Fiout=Y1


D1=(Po-Y2-My*Y1*ABS(Y1))/Tau1
D3=(Y2-Y4-My*Y3*ABS(Y3))/Tau1
D5=(Y4-Y6-My*Y5*ABS(Y5))/Tau1
D7=(Y6-Y8-My*Y7*ABS(Y7))/Tau1
D9=(Y8-Y10-My*Y9*ABS(Y9))/Tau1

D2=(Y1-Y3)/Tau2
D4=(Y3-Y5)/Tau2
D6=(Y5-Y7)/Tau2
D8=(Y7-Y9)/Tau2
D10=(Y9-Fi1)/Tau2


"Parameters:
Tau1:2.84e-3        "Time constant             [SEC]
Tau2:1.875          "Time constant             [SEC]
My:55.363E-7        "Friction constant

END
```

```
CONTINUOUS SYSTEM STEAM2

"File:ST2.T

"Version 2:82-06-09

"Author:STEN BERGMAN
"       SYDKRAFT AB


"Description:
"            A simplified nonlinear
"            5-node steam-line model

INPUT Po Fi2
OUTPUT Fiout

STATE Y1 Y2 Y3 Y4 Y5 Y6 Y7 Y8 Y9 Y10
DER D1 D2 D3 D4 D5 D6 D7 D8 D9 D10

"Inputs: Po    Pressure in reactor dome        [BAR]
"        fi2   Steamoutlet from the pipe       [KG/S]
"Outputs:Fiout Steamoutlet from reactor        [KG/S]

"States: Y1 Y3 Y5 Y7 Y9   Intermediate steam flow  [KG/S]
"        Y2 Y4 Y6 Y8 Y10  Node pressure            [BAR]

Fiout=Y1


D1=(Po-Y2-Y1*ABS(Y1)*My)/Tau1
D3=(Y2-Y4-Y3*ABS(Y3)*My)/Tau1
D5=(Y4-Y6-Y5*ABS(Y5)*My)/Tau1
D7=(Y6-Y8-Y7*ABS(Y7)*My)/Tau1
D9=(Y8-Y10-Y9*ABS(Y9)*My)/Tau1

D2=(Y1-Y3)/Tau2
D4=(Y3-Y5)/Tau2
D6=(Y5-Y7)/Tau2
D8=(Y7-Y9)/Tau2
D10=(Y9-Fi2)/Tau2


"Parameters:
Tau1:2.84e-3       "Time constant             [SEC]
Tau2:1.875         "Time constant             [SEC]
My:55.363E-7       "Friction constant

END
```

```
CONTINUOUS SYSTEM VALVE

"File:VALVE.T

"Author: Per Persson

"Description: This system describes the interaction between
"             the two steampipes which are described by the
"             systems STEAM1 and STEAM2.

INPUT ht11 ht12 Y101 Y102

OUTPUT Fi1 Fi2

x1=ht11-oht1
ahl1=if x1<0  then 0 else ght*x1
ahk1=if x1<0  then 0 else ght*x1*x1*0.52/((.52-oht1)*(0.52-oht1))
G11=ahk1
G12=k2*ht11+12
G13=ahl1
h11=if (ht11<.3) then G11 else G12
h12=if (ht11>.3) then G12 else h11
h13=if (ht11>.5) then G13 else h12
Ah1=h13

x2=ht12-oht1
ahl2=if x2<0  then 0 else ght*x2
ahk2=if x2<0  then 0 else ght*x2*x2*0.52/((.52-oht1)*(0.52-oht1))
G21=ahk2
G22=k2*ht12+12
G23=ahl2
h21=if (ht12<.3) then G21 else G22
h22=if (ht12>.3) then G22 else h21
h23=if (ht12>.5) then G23 else h22
Ah2=h23

A1=MIN(MAX(Ah1,0),1)
A2=MIN(MAX(Ah2,0),1)

alfa1=(Y101-Pho)*A1*A1/Khtv
alfa2=(Y102-Pho)*A2*A2/Khtv

beta1=Kht*A1*A1/Khtv
beta2=Kht*A2*A2/Khtv

A=2*beta1
B=-2*alfa1+beta1*beta1-beta1*beta2
C=-2*alfa1*beta1
D=beta1*beta2*alfa1-beta1*beta1*alfa2+alfa1*alfa1

F0=850.00*(A1+A2)/2
H0=C+F0*(2*B+F0*(3*A+4*F0))                    "Iteration #1
F1=F0-(D+F0*(C+F0*(B+F0*(A+F0))))/H0
H1=C+F1*(2*B+F1*(3*A+4*F1))                    "Iteration #2
F2=F1-(D+F1*(C+F1*(B+F1*(A+F1))))/H1
```

```
H2=C+F2*(2*B+F2*(3*A+4*F2))                        "Iteration #3
F3=F2-(D+F2*(C+F2*(B+F2*(A+F2))))/H2
H3=C+F3*(2*B+F3*(3*A+4*F3))                        "Iteration #4
F4=F3-(D+F3*(C+F3*(B+F3*(A+F3))))/H3

Fi1=F4
Fi2=SQRT((beta2*Fi1*Fi1-alfa1*beta2+alfa2*beta1)/beta1)

"Parameters:

Ght:1.0                  "Valve constants
Oht1:0.05
Khtv:9.002E-6
Kht:6.02E-2
Pho:5.8
k2:1.54267
l2:-.321333

END
```

APPENDIX 3 PROGRAM FÖR REKURSIV IDENTIFIERING

```
program test(input,output,data);

{ Author : Per Persson
           1983 - 03 - 30
This program reads data from the file 'DATA'
generated by datagen2 and identifies the model
which generated the signal y(t) from u(t)}

type    fitype = array [1..50] of real;
       mattype = array [1..50,1..50] of real;
       rectype = record
                  u : real;
                  y : real;
                 end;
       filtype = file of rectype;

var   fi,theta : fitype;  {variables necessary for recid}
lambda,lambda0 : real;
 gamma1,gamma2 : real;
          beta : real;
         death : real;
             r : real;
             s : real;
             w : fitype;
          p1,p : mattype;
            Td : integer;
            ua : fitype;
     y,y1,u,u1 : real;
    z1,v1,eps1 : real;
         ljung : integer;
   yl,yh,ul,uh : integer;
   el,eh,zl,zh : integer;
         vl,vh : integer;

             t : integer;   {variables for the testprogram}
            pe : boolean;
          data : filtype;
           a,b : array [1..50] of real;
          nlev : real;
         print : integer;
      mo,np,j : integer;
            dt : rectype;
            ch : char;
na,nb,nc,nd,nf : integer;
{-------------------------------------------------------------}
procedure initall;

{Initializes all parameters necessary to carry out an
identification}

var i,j : integer;
     ph : real;
      h : rectype;
```

```
begin
 for i:=1 to 50 do begin fi[i]:=0; w[i]:=0; theta[i]:=0 end;
 for i:=1 to 50 do begin ua[i]:=0 end;
 reset(data);
 read(data,h);
 np:=trunc(h.u+0.5);
 writeln(np,' SAMPEL');
 nlev:=h.y;
 writeln('noiselevel:  ',nlev:7:4);
 read(data,h);
 mo:=trunc(h.u+0.5);
 writeln('true modelorder: ',mo:1);
 write('ID modelorder=    '); readln(na);
 nb:=na;
 nc:=na;
 nd:=na;
 nf:=na;
 yl:=1;
 yh:=na;
 ul:=yh+1;
 uh:=na+nb;
 el:=uh+1;
 eh:=na+nb+nc;
 zl:=eh+1;
 zh:=na+nb+nc+nf;
 vl:=zh+1;
 vh:=na+nb+nc+nf+nd;
 for j:=1 to mo do
 begin
  read(data,h);
  a[j]:=h.y;
  b[j]:=h.u;
 end;
 for j:=1 to mo do write('  a',j:1,':',a[j]:7:4); writeln;
 for j:=1 to mo do write('  b',j:1,':',b[j]:7:4); writeln;
 for j:=1 to na do
 begin
  write('a',j:1,'=     '); readln(theta[yl+j-1]);
 end;
 for j:=1 to na do
 begin
  write('b',j:1,'=     '); readln(theta[ul+j-1]);
 end;
 beta:=0;
 t:=0;
 r:=0;
 s:=0;
 y:=0;
 y1:=0;
 u:=0;
 u1:=0;
 eps1:=0;
 z1:=0;
 v1:=0;
 write('gamma1=    '); readln(gamma1);
```

```
 write('gamma2=    '); readln(gamma2);
 write('lambda0=   '); readln(lambda0);
 write('lambda=    '); readln(lambda);
 write('death=     '); readln(death);
 write('delay Td= '); readln(Td);
 write('ljung=     '); readln(ljung);
 write('print=     '); readln(print);
 for i:=yl to vh do for j:=yl to vh do p[i,j]:=0;
 writeln('p for y');
 write('py=        '); readln(ph);
 for i:=yl to yh do p[i,i]:=ph;
 writeln('p for u');
 write('pu=        '); readln(ph);
 for i:=ul to uh do p[i,i]:=ph;
 writeln('p for eps');
 write('peps       '); readln(ph);
 if ph=0 then pe:=false else pe:=true;
 for i:=el to eh do p[i,i]:=ph;
 if ljung>0 then
 begin
  writeln('p for v and z');
  write('pzv        '); readln(ph);
  for i:=zl to vh do p[i,i]:=ph;
 end;
end;
{-------------------------------------------------------------}
procedure showcoeff;
var i,j :integer;
begin
 writeln('TIME ',t+1:4,' SAMPEL');
 write('A:');
 for i:=yl to yh do write(theta[i]:9:4);
 write('   true A:');
 for i:=1 to mo do write(a[i]:9:4);
 writeln;
 write('B:');
 for i:=ul to uh do write(theta[i]:9:4);
 write('   true B:');
 for i:=1 to mo do write(b[i]:9:4);
 writeln;
 write('C:');
 for i:=el to eh do write(theta[i]:9:4);
 write('   nlev:',nlev:9:4); writeln;
 writeln('s:',s:9:4);
 writeln('r:',r:9:4);
 writeln('beta:',beta:9:4);
 writeln('P matrix');
 for i:=yl to eh do
 begin
  for j:=yl to eh do write(p[i,j]:6:2);
  writeln;
 end;
 write('print:');readln(print);
 writeln;
 writeln;
```

```
end;
{-----------------------------------------------------------------}
procedure recid(ut,yt : real);
var  thetain : fitype;
      ny,ny0 : real;

{ This procedure implements  an identification
algoritm for identifying A,B,C,D and F in

             B u(t-Td)      Ce(t)
   Ay(t) = ----------- + ---------
              1 + F         1 + D

where y(t) and u(t) are given as inputs. If the constant
'ljung' is set = 0 then the polynomials F and D are
set = 0.

THESE VARIABLES MUST BE GLOBALLY DEFINED:

type    fitype = array [1..50] of real;
       mattype = array [1..50,1..50] of real;
       rectype = record
                  u : real;
                  y : real;
                 end;
       filtype = file of rectype;


var   fi,theta : fitype;  measurement and parametervector
lambda,lambda0 : real;    forgettingfactor
 gamma1,gamma2 : real;    variables to compute beta
          beta : real;
         death : real;
             r : real;
             s : real;
             w : fitype;
          p1,p : mattype; p-matrix in identification algorithm
            Td : integer; delay time
            ua : fitype;  delay vector
     y,y1,u,u1 : real;    help variables
    z1,v1,eps1 : real;
         ljung : integer;
   yl,yh,ul,uh : integer; upper and lower bounds for the
   el,eh,zl,zh : integer; y,u,e,z and v parts of fi and theta
         vl,vh : integer;}

{-----------------------------------------------------------------}
  procedure multvm(min: mattype;vin:fitype;var vut:fitype);

  {Multiplies a vector and a matrix}

  var i,j : integer;
       vi : real;
  begin
   if ljung>0 then
```

```
   for i:=yl to vh do
   begin
    vi:=0;
    for j:=yl to vh do vi:=min[i,j]*vin[j]+vi;
    vut[i]:=vi;
   end
  else
   for i:=yl to eh do
   begin
    vi:=0;
    for j:=yl to eh do vi:=min[i,j]*vin[j]+vi;
    vut[i]:=vi;
   end;
 end;
{---------------------------------------------------------------}
 procedure multmm(min1,min2:mattype;var mut:mattype);

 {Multiplies two matrixes}

 var i,j,k : integer;
       mij : real;
 begin
  if ljung>0 then
   for i:=yl to vh do
   for j:=yl to vh do
   begin
    mij:=0;
    for k:=yl to vh do mij:=mij+min1[i,k]*min2[k,j];
    mut[i,j]:=mij;
   end
  else
   for i:=yl to eh do
   for j:=yl to eh do
   begin
    mij:=0;
    for k:=yl to eh do mij:=mij+min1[i,k]*min2[k,j];
    mut[i,j]:=mij;
   end;
 end;
{------------------------------------------------------------}
 procedure newdata;

 {Accepts new data from the program and if Td>0 delays u}

 var  i : integer;
 uh,uh1 : real;
 begin
  y1:=y;
  y:=yt;
  if Td = 0 then begin u1:=u; u:=ut end
  else
   begin
    u1:=ua[Td];
    for i:=Td downto 2 do ua[i]:=ua[i-1];
    ua[1]:=ut;
```

```
    end;
  end;
{-------------------------------------------------------------}
  procedure updatefi;
  var i : integer;
   h,h1 : real;
  begin
   if na>=1 then
    begin
     for i:=yh downto yl+1 do  fi[i]:=fi[i-1];
     fi[yl]:=-y1;
    end;
   if nb>=1 then
    begin
     for i:=uh downto ul+1 do fi[i]:=fi[i-1];
     fi[ul]:=u1;
    end;
  if nc>=1 then
    begin
     for i:=eh downto el+1 do fi[i]:=fi[i-1];
     fi[el]:=eps1;
    end;
   if ljung>0 then
    begin
     for i:=vh downto vl+1 do fi[i]:=fi[i-1];
     fi[vl]:=-v1;
     for i:=zh downto zl+1 do fi[i]:=fi[i-1];
     fi[zl]:=-z1;
    end;
 end;
{----------------------------------------------------------------}
  procedure computez;
  var i : integer;
  begin
   z1:=0;
   for i:=zl to zh do z1:=z1-theta[i]*fi[i];
   for i:=ul to uh do z1:=z1+theta[i]*fi[i];
  end;
{--------------------------------------------------------------}
  procedure computev;
  var i : integer;
  begin
   v1:=0;
   for i:=vl to vh do v1:=v1-theta[i]*fi[i];
   for i:=el to eh do v1:=v1+theta[i]*fi[i];
   v1:=v1+eps1;
  end;
{--------------------------------------------------------------}
  procedure computeeps;
  var i : integer;
  begin
   eps1:=y;
   for i:=yl to eh do eps1:=eps1-theta[i]*fi[i];
   if ljung>0 then for i:=zl to vh do eps1:=eps1-theta[i]*fi[i];
  end;
```

```
{-------------------------------------------------------------}
  procedure computep;
  var fih,fih1 : mattype;
           vec : fitype;
             h : real;
           i,j : integer;
  begin
   p1:=p;
   for i:=yl to vh do
   for j:= yl to vh do fih[i,j]:=fi[i]*fi[j];
   multmm(fih,p,fih1);
   multmm(p,fih1,fih);
   multvm(p,fi,vec);
   h:=0;
   for i:=yl to vh do h:=h+vec[i]*fi[i];
   for i:=yl to vh do
   for j:=yl to vh do
    p[i,j]:=(p[i,j]- (fih[i,j]/(lambda+h)))/lambda;
   if ljung>0 then
    for i:=yl to vh do p[i,i]:=p[i,i]+beta
   else
    for i:=yl to uh do p[i,i]:=p[i,i]+beta;
    if pe then for i:= el to eh do p[i,i]:=p[i,i]+beta;
  end;
{--------------------------------------------------------------}
  procedure computetheta;
  var i : integer;
  begin
   multvm(p,fi,thetain);
   for i:=yl to eh do thetain[i]:=thetain[i]*eps1;
   if ljung>0 then
    for i:=zl to vh do thetain[i]:=thetain[i]*eps1;
   for i:=yl to eh do theta[i]:=theta[i]+thetain[i];
   if ljung>0 then
    for i:=zl to vh do theta[i]:=theta[i]+thetain[i];
  end;
{--------------------------------------------------------------}
  procedure computelambda;
  begin
   lambda:=lambda0*lambda + (1-lambda0);
  end;
{--------------------------------------------------------------}
  procedure computebeta;
  var      i : integer;
   fi2,h1,h : real;
        vut : fitype;
  begin
   s:=0;
   for i:=yl to eh do s:=s+thetain[i]*w[i];
   for i:=yl to eh do w[i]:=gamma1*w[i]+thetain[i];
   if ljung >0 then
   begin
    for i:=zl to vh do s:=s+thetain[i]*w[i];
    for i:=zl to vh do w[i]:=gamma1*w[i]+thetain[i];
   end;
```

```
   if s>0 then s:=1 else s:=-1;
   r:=r*gamma2+(1-gamma2)*s;
   fi2:=0;
   for i:=yl to eh do fi2:=fi2+fi[i]*fi[i];
   if ljung >0 then
    for i:=zl to vh do fi2:=fi2+fi[i]*fi[i];
   multvm(p1,fi,vut);
   h1:=0;
   for i:=yl to eh do h1:=h1+fi[i]*vut[i];
   if ljung >0 then
    for i:=zl to vh do h1:=h1+fi[i]*vut[i];
   ny0:=lambda/(lambda + h1);
   if r<0.9 then ny:=1 else ny:=8*(-r+1);
   if r>1 then ny:=0;
   if fi2<>0 then  h:=(ny0-ny)/fi2 else h:=1000;
   if r>death then beta:=h else beta:=0;
  end;
{------RECID-----------------------------------------------------}
begin
 newdata;
 updatefi;
 if ljung>0 then computez;
 computeeps;
 if ljung>0 then computev;
 computelambda;
 computep;
 computetheta;
 computebeta;
end;

{------MAIN------------------------------------------------------}

begin
 ch:='y';
 while ch='y' do
 begin
  initall;
  for j:=1 to np do
  begin
   read(data,dt);
   recid(dt.u,dt.y);
   if print<>0 then if (t mod print)=0 then showcoeff;
   if t=np-1 then showcoeff;
   t:=t+1;
  end;
  write('Another identification? '); readln(ch);
 end;{while}
end.
```

```
program datagen(input,output,data);

{ Author : Per Persson
           1983 - 03 -30

This program generates data from a discrete system
       Ay = Bu + e
where A and B are polynomials in the shiftoperator.
e is noise and u is the insignal generated in a
shift register. The  file 'DATA' is filled with
records containing two real numbers. The file
contains
record 1       : number of sampels = n , noiselevel
record 2       : modelorder = mo
record 3       : b[1] , a[1]
.
.
record 2+mo    : b[mo] , a[mo]
record 2+mo+1  : u[1] , y[1]
.
.
record 3+mo+n : u[n] , y[n]}
var        y : array [0..50] of real;
           u : array [0..50] of real;
           a : array [1..50] of real;
           b : array [1..50] of real;
         niv : real;
          mo : integer;
    noiselev : real;
        seed : integer;
    p1,p2,p3 : integer;
    p4,p5,p6 : integer;
    p7,p8,p9 : integer;
     p10,p11 : integer;
         p12 : integer;
       j,n,i : integer;
        data : file of
                record
                 u : real;
                 y : real;
                end;
         obs : record
                u : real;
                y : real;
               end;
{-----------------------------------------------------------}
function mth$random(var seed:integer):real;extern;
{-----------------------------------------------------------}
procedure prbs;
{Generates the u(t) sequence recursively}
var ph : integer;
begin
 u[0]:=p12;
 if (p11*p12=0) and (p11+p12=1) then ph:=1 else ph:=0;
 p12:=p11;
```

```
 p11:=p10;
 p10:=p9;
 p9:=p8;
 p8:=p7;
 p7:=p6;
 p6:=p5;
 p5:=p4;
 p4:=p3;
 p3:=p2;
 p2:=p1;
 p1:=ph;
end;
{----------------------------------------------------------}
procedure start;
var i : integer;
begin
 write('modelorder='); readln(mo);
 for i:=1 to mo do
 begin
  write('a',i:1,'=    '); readln(a[i]);
 end;
 for i:=1 to mo do
 begin
  write('b',i:1,'=    '); readln(b[i]);
 end;
 write('noiselevel=  '); readln(noiselev);
 write('antal sampel='); readln(n);
 p1:=1;p2:=0;p3:=1;p4:=0;p5:=1;p6:=1;
 p7:=1;p8:=0;p9:=0;p10:=1;p11:=1;p12:=1;
 for i:=1 to 50 do
 begin
  y[i]:=0;
  u[i]:=0;
 end;
 seed:=1;
 rewrite(data);
 obs.u:=n;
 obs.y:=noiselev;
 write(data,obs);
 obs.u:=mo;
 write(data,obs);
 for i:=1 to mo do
 begin
  obs.u:=b[i];
  obs.y:=a[i];
  write(data,obs);
 end;
 niv:=0.5
end;
{----------------------------------------------------------}
begin
 start;
 for i:=1 to n do
 begin
  prbs;
```

```
  y[0]:=0;
  for j:=1 to mo do y[0]:=y[0]-a[j]*y[j]+b[j]*u[j];
  obs.u:=u[0];
  y[0]:=y[0]+noiselev*(mth$random(seed)-niv);
  obs.y:=y[0];
  write(data,obs);
  for j:=mo downto 1 do
  begin
   y[j]:=y[j-1];
   u[j]:=u[j-1];
  end;
 end;
end.
```

Abstract: This master thesis describes a Simnon model of a boiling water reactor to be used in simulating faults and disturbances. These faults and disturbanses will be detected by noise analysis. Some methods in identification and noise analysis are also described and are applied on some malfunctions of a servo. A Pascal program for recursive parameter identification was also written and tested. This program is to be used in an expert system for noise analysis on the nuclear power plant Barsebäck.