

ISSN 0280-5316
ISRN LUTFD2/TFRT--5486--SE

Adaptiv Effektreglering av Kokarvattenreaktor

Mats Larsson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Oktober 1993

ISSN 0280-5316
ISRN LUTFD2/TFRT--5486--SE

Adaptiv Effektreglering av Kokarvattenreaktor

Mats Larsson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
Oktober 1993

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden	<i>Document name</i> Master thesis	
	<i>Date of issue</i> October 1993	
	<i>Document Number</i> ISRN LUTFD2/TFRT--5486--SE	
<i>Author(s)</i> Mats Larsson	<i>Supervisor</i> Bohao Liao - ABB, Tore Hägglund	
	<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Adaptiv effektreglering av kokarvattenreaktor (Adaptive control of boiling water reactors)		
<i>Abstract</i> <p>The purpose of the report is to investigate if the power control system in a boiling water reactor can be improved by means of adaptive control. The report shows that improved regulation is possible, in terms of better damping of the neutron flux variations as well as improved stability margins.</p> <p>An indirect self-tuning regulator using pole-placement for controller design and extended least-squares for parameter identification is designed. The robustness of the adaptive controller is investigated in simulations with SIMULINK and Macintosh computer. The results are verified in simulations with BISON, a process simulator developed at ABB Atom.</p> <p>The improved regulative performance is a result of the controller's ability to adapt to process variations. To aid parameter estimation an external perturbation signal is introduced. The neutron-flux variations due to the perturbation is negligible.</p>		
<i>Key words</i>		
<i>Classification system and/or index terms (if any)</i>		
<i>Supplementary bibliographical information</i>		
<i>ISSN and key title</i> 0280-5316		<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 45	<i>Recipient's notes</i>
<i>Security classification</i>		

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, Fax +46 46 110019, Telex: 33248 lubbis lund.

INNEHÅLL

1.	Inledning	3
2.	Problembeskrivning	4
	Kokarvattenreaktorn	4
	Effektreglersystemet	4
	Processmodell	5
	Målsättningar och krav	6
3.	Identifiering	7
3.1	Teori	7
	Minstakvadratmetoden	7
	ARX-modell	8
	ARMAX-modell	8
	Excitation	9
3.2	Simulering	10
	Val av modellordning	10
	Jämförelse av RLS och ELS	10
	Exponential forgetting	12
	Estimatoruppvridding	14
4.	Regulatordesign	15
4.1	Teori	15
4.2	Simulering	16
	Val av modellsystemets poler	18
	Val av observerarpolynom	18
	Känslighet för modellfel	19
	Skydd mot integratoruppvridding	21
	Jämförelse med PID-regulator	21
5.	Självinställande regulator	24
5.1	Simulering	24
	Följförmåga	27
	Jämförelse av adaptiv och fix regulator	28
	Verifiering	30
6.	Kommentarer	32
A.	Implementering i SIMULINK	34
B.	Implementation i FORTRAN	38
C.	Översiktsbild på BWR	44
D.	Referenser	45
E.	Tack	45

INLEDNING

Syftet med examensarbetet är att med hjälp av simuleringsundersökningar undersöka huruvida det är möjligt att förbättra effektregeringen i en kokarvattenreaktor (BWR) med hjälp av adaptiv reglering.

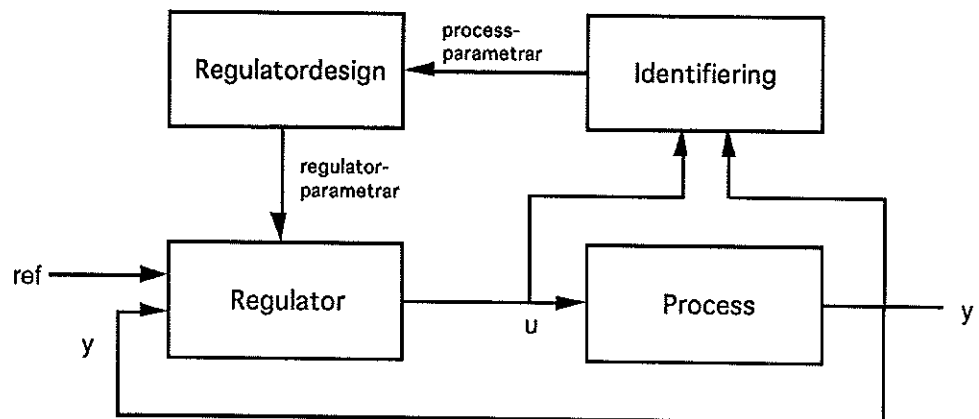
Idén bakom adaptiv reglering är att man låter regulatorn själv anpassa sig efter processen som regleras. Tekniken har sitt främsta användningsområde i samband med reglering av processer som varierar i tiden eller med arbetspunkten. Adaptiv reglering kan ses som en automatisering av identifierings- och designfaserna i ett reglerproblem.

Den typ av adaptiv regulator som jag valt att använda kallas för indirekt självinställande regulator (STR). Fördelen med en STR är att identifierings- och reglerproblemet separeras, vilket ger stor frihet i valet av identifierings- och regleralgoritm. En annan fördel är att man kan förutsäga självinställarens egenskaper genom att analysera reglering och identifiering separat.

En indirekt STRs arbetscykel kan delas upp i tre faser :

- processidentifiering - en förenklad processmodell identifieras ur in- och utsignalerna från processen.
- regulatordesign - den identifierade processmodellen används för att ställa in en regulator.
- reglering - regulatorn används för att reglera processen.

I figur 1.1 finns blockschema för den självinställande regulatorn.



Figur 1.1 : Blockschema för process och självinställande regulator.

Den stora nackdelen med verkliga adaptiva reglerloopar är att deras stabilitet ej kan bevisas matematiskt. Adaptiva regulatorer har däremot i praktiken visat sig ha mycket goda stabilitetsegenskaper.

Eftersom neutronflödesreglering i en BWR är en olinjär process som är beroende av en mängd faktorer som varierar med arbetspunkten bör en adaptiv regulator både kunna förbättra regleregenskaperna i neutronflödesloopen, såväl genom att minska variationerna i neutronflödet som genom att förbättra reglerloopens stabilitetsegenskaper.

2. PROBLEMBESKRIVNING

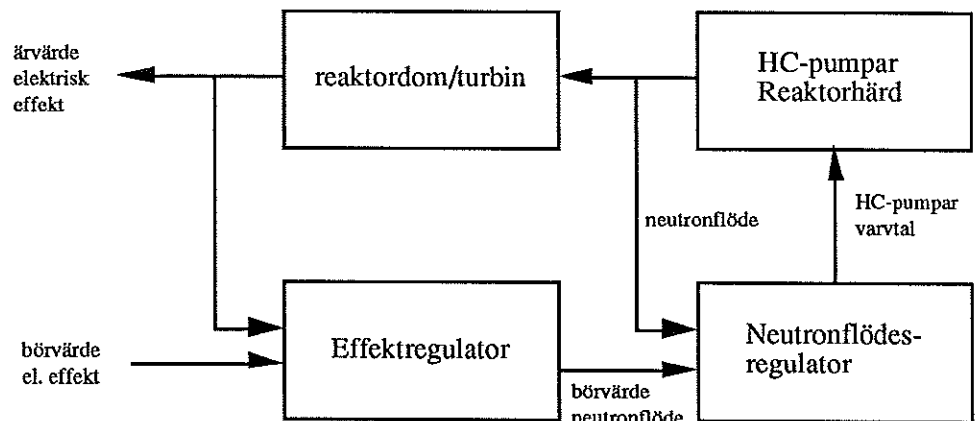
Kokarvattenreaktorn

Den effektutvecklande delen i en kokarvattenreaktor (BWR) är reaktorhärden där en mängd bränsleelement sitter monterade. Mellan bränsleelementen finns styrtavarna som kan användas för att grovreglera effektutvecklingen. Under drift pumpas vatten, hårdkylflödet, mellan bränslestavarna, varvid vattnet förångas. Ångan stiger sedan upp i toppen av reaktorn där den släpps genom turbinen. Efter passagen genom turbinen kondenseras ångan och vattnet återförs till reaktorn av matarvattensystemet som håller nivån i reaktortanken konstant genom att reglera flödet genom matarvattenledningen. I appendix C finns en översiktsbild av en BWR.

Den termiska effektutvecklingen i härden styrs först och främst av styrtavornas lägen och huvudcirkulationsflödet (=HC-flöde). Även reaktortrycket och matarvattentemperaturen har stor betydelse för effektutvecklingen. Dessa hålls någorlunda konstanta av matarvatten- och turbinreglersystemen.

I lågeffektsområdet styrs effektutvecklingen av styrtavarna varvid HC-flödet hålls konstant lågt. I högeffektområdet (>70 %) regleras effektutvecklingen företrädesvis genom små variationer i HC-flödet och styrtavornas läge justeras endast för att få en jämn fördelning av effektutvecklingen i härden. Bästa driftsekonomi fås vid hög effekt och lågt HC-flöde, men dessvärre närmar sig processen instabilitet vid låga HC-flöden. Efterhand som bränsleelement bränns ut sjunker deras effektutveckling. Detta kompenseras så långt det är möjligt genom styrtavsutdragnings och först därefter ökas HC-flödet.

Effektreglersystemet



Figur 2.2 : Översiktsbild över effektreglersystemet i en BWR.

I figur 2.2 syns en schematisk bild av effektreglersystemet i en BWR. Effektreglersystemet består i princip av två kaskadkopplade reglerkretsar, elektriska effektkretsen, och neutronflödeskretsen. Dessutom finns en reglerloop i pumparna, vars dynamik försummas.

Neutronflödeskretsen är markant snabbare än elektriska effektkretsen och kan därför utifrån ses som en ren förstärkning av effektregulatorn. Elektriska effektbörvärdet ställs in av operatören och ärvärdet mäts vid generatorerna. Styrtavornas läge styrs ej av effektreglersystemet.

I Oskarshamn 2 används idag en PI-regulator som effektregulator och som neutronflödesregulator en PID-regulator kompletterad med ett bandstoppfilter för att se till att regulatorn inte exciterar processen kring resonansfrekvensen. Jag kommer att inrikta mig på att förbättra neutronflödesregulatorn med hjälp av adaptiv reglering.

Processmodell

För att modellera reaktorn vid simuleringarna har en processmodell på formen

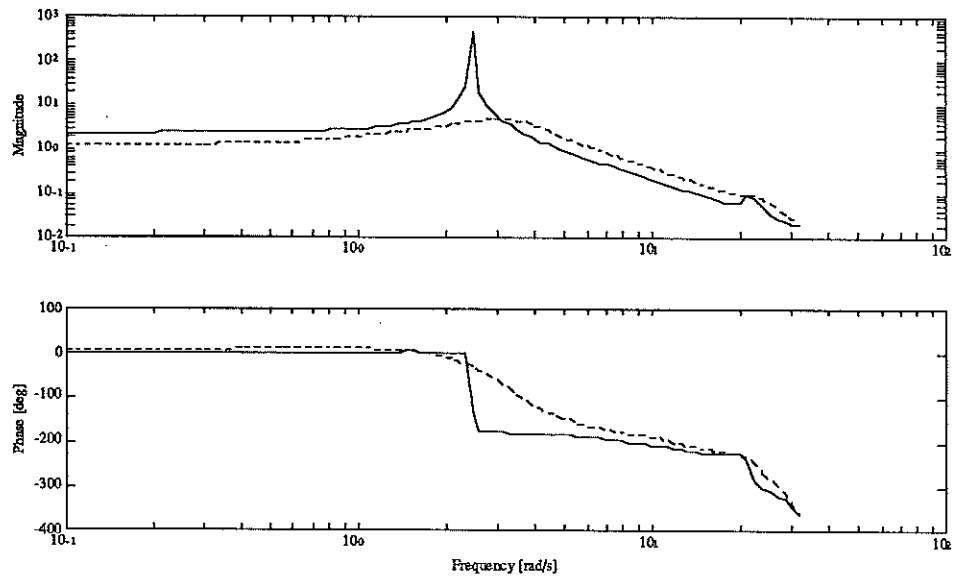
$$y(k) = H_u(q) u(k) + H_e(q) e(k) = \frac{B(q)}{A(q)} u(k) + \frac{D(q)}{C(q)} e(k) \quad (2.1)$$

där $e(k)$ är gaussiskt vitt brus i diskret tid, identifierats fram med hjälp av Matlabs system identification toolbox.

Störningstermen innehåller förutom direkta mätfel även variationer i bla. matarvattenflöde och reaktortryck. Utsignalen, den termiska effektutvecklingen, och styrsignalen, korrektionen av varvtalet anges i procent av full reaktoreffekt respektive nominellt pumpvartal.

Mätdata för processtermen ($H_u(q) u(k)$) togs från simuleringsprogrammet BISON och för störningstermen ($H_e(q) e(k)$) från mätningar i Oskarshamn 2. Då mätningarna gjordes användes bränslet Svea 64. Mätvärdena var samplade med samplingstiden 0.1 s. Överföringsfunktionerna är av sjätte respektive femte ordningen. Eftersom processmodellens två olika delar identifierats ur mätdata med olika ursprung korrigerades störningsdelen så att dess dominerande poler sammanföll med processdelens.

Processen beror av arbetspunkten och flera modeller måste därför tas fram. A,B,C och D-polynomen identifierades vid 6 olika arbetspunkter vid konstant effekt ($\approx 70\%$) och varierande härdkylflöde (2400-5000 kg/s). Förflyttning av arbetspunkten längs denna linje svarar mot en ändring av styrstavsläget. Mellan de 6 arbetspunkterna görs linjär interpolation av varje koefficient i de olika polynomen. Processens stationära förstärkning sjunker i detta område linjärt med ökande HC-flöde. I figur 2.3 syns bodediagram för den identifierade processmodellen vid härdkylflödena 2400 kg/s och 5000 kg/s.



Figur 2.3 : Bodediagram för överföringsfunktionen $H(q) = \frac{B(q)}{A(q)}$, vid HC-flöde på 2400 kg/s (heldragna) och 5000 kg/s (prickade).

För låga HC-flöden har processen en mycket illa dämpad resonanstopp vid ca 0.4 Hz. Vid härdkylflöden lägre än ca 2500 kg/s är processen instabil vid denna effektnivå. Denna resonans beror på en komplicerad växelverkan mellan tryck, temperatur, flöde genom reaktorhärden och ånghalten i kylvattnet. Dess frekvensläge beror till största delen av härdens fysiska dimensioner och är därför någorlunda konstant.

Anledningen till att just detta effekt-/HC-flödesområde valdes är att processen är som mest svårhanterlig i detta område. Övergången mellan stabil och instabil process är speciellt intressant att studera eftersom den är svår att klara med konventionella regulatorer.

Målsättningar och krav

Målsättningen med regleringen är att hålla variationerna i neutronflödet så små som möjligt. Det viktigaste är att få en regulator som tillförlitligt kan hantera stora processvariationer. Regulatorn skall kunna kompensera processens instabilitet och ge goda stabilitetsmarginaler.

För häftiga skiftningar i styrsignalen kan orsaka resonanssvängningar i härdkylflödet och därför begränsas styrsignalens derivata till $\pm 5\%/s$. Variationer i tryck eller ånguttag påverkar systemet i form av laststörningar och regulatorn skall därför innehålla en integrator. För att kunna kommunicera synkront med andra reglersystem skall regulatorn innehålla en tidsfördröjning. Samplingstiden är fastställd till 0.1 s.

3. IDENTIFIERING

Estimatorns uppgift är att skapa en så tillförlitlig bild av processen som möjligt. Vid adaptiv reglering måste identifieringen ske on-line och därför måste någon rekursiv identifieringsalgoritm användas. Jag har implementerat och jämfört rekursiva minstakvadratmetoden (RLS) och utökade rekursiva minstakvadratmetoden (ELS).

3.1 TEORI

Minstakvadratmetoden

Minstakvadratmetoden är en metod för att anpassa parametrar i en förutbestämd modellstruktur till en mängd mätvärden. Metoden förutsätter att modellen kan parametreras enligt

$$y(t) = \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2 + \dots + \varphi_n(t)\theta_n = \varphi^T(t)\theta(t) + e(t) \quad (3.1)$$

där θ är parametrar, $\varphi^T(t)$ är en uppsättning av kända funktioner och $e(t)$ är vitt brus. Minstakvadratproblemet är att hitta den uppsättning parametrar som minimerar summan av de kvadratiska prediktionsfelen dvs. minimerar förlustfunktionen

$$V(\theta, t) = \frac{1}{2} \sum_{i=1}^N \varepsilon(i)^2 = \frac{1}{2} \sum_{i=1}^N (y(i) - \varphi^T(i)\theta(i))^2 \quad (3.2)$$

$V(\theta, N)$ minimeras om man uppdaterar parametrarna enligt:

Recursive Least-Squares (RLS) [2,3]:

$$\varepsilon(t) = y(t) - \varphi^T(t)\hat{\theta}(t-1)$$

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

$$K(t) = P(t-1)\varphi(t) / (1 + \varphi^T(t)P(t-1)\varphi(t))$$

$$P(t) = (I - K(t)\varphi^T(t))P(t-1) \quad (3.3)$$

$P(t)$ är en matris som kan sägas vara ett mått på osäkerheten i skattningarna. En växande P -matris innebär att osäkerheten i parametrarna ökar och skattningarna kommer att vara mer benägna att ändras.

En brist i RLS-algoritmen är att den ej tar hänsyn till att parametrarna kan variera i tiden. Ett sätt att ta hänsyn till detta är att modifiera förlustfunktionen och i stället minimera

$$V(\theta, t) = \frac{1}{2} \sum_{i=1}^t \lambda^{t-i} \varepsilon(i)^2 = \frac{1}{2} \sum_{i=1}^t \lambda^{t-i} (y(i) - \varphi^T(i)\theta(i))^2 \quad 0 < \lambda < 1 \quad (3.4)$$

Äldre mätvärden viktas alltså bort till förmån för nyare. En algoritm som minimerar förlustfunktionen (3.4) är :

RLS with exponential forgetting [2,3]:

$$\begin{aligned}\varepsilon(t) &= y(t) - \varphi^T(t)\hat{\theta}(t-1) \\ \hat{\theta}(t) &= \hat{\theta}(t-1) + K(t)\varepsilon(t) \\ K(t) &= P(t-1)\varphi(t) / (\lambda + \varphi^T(t)P(t-1)\varphi(t)) \\ P(t) &= (I - K(t)\varphi^T(t))P(t-1)/\lambda\end{aligned}\tag{3.5}$$

Konstanten λ kallas glömskefaktor och har stor betydelse för uppförandet hos parameterskattningarna. Denna parameter styr hur många värden som skall tas med i beräkningen av parameterskattningarna. Väljs λ för stort hinner estimatorn inte med att följa snabba processvariationer och väljs det för litet blir estimatorn brus känslig eftersom ett enstaka stort mätvärde påverkar estimaten kraftigt (om än under kort tid). Normalt väljer man λ något mindre än 1. En nackdel med alla algoritmer som tar hänsyn till variationer i processen är att de i teorin aldrig konvergerar fullständigt, utan parameterskattningarna kommer att oscillera. Turligtvis kan λ väljas så att oscillationens amplitud blir liten.

ARX-modell

Minstakvadratmetoden kan direkt användas för att skatta parametrar i nedanstående modell :

$$\begin{aligned}A(q^{-1})y(k) &= B(q^{-1})u(k) + e(k) \\ \text{där} \\ A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \\ B(q^{-1}) &= b_1q^{-1} + \dots + b_{nb}q^{-nb}\end{aligned}\tag{3.6}$$

där $e(k)$ är Gaussiskt vitt brus i diskret tid. Detta är ekvivalent med regressionsmodellen :

$$\begin{aligned}y(t) &= \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2 + \dots + \varphi_{na+nb}(t)\theta_{na+nb} = \varphi^T(t)\theta(t) \\ \text{där} \\ \varphi^T(t) &= (-y(t-1) \dots -y(t-na) \ u(t-1) \dots u(t-nb)) \\ \theta &= (a_1 \dots a_{na} \ b_1 \dots b_{nb})\end{aligned}\tag{3.7}$$

En nackdel med denna modellstruktur är att brustermerna, $e(k)$, antas vara okorrelerade. Om brustermerna verkligen är korrelerade kommer estimatorn ej att ge konsistenta parameterskattningar. Antagandet om okorrelerade brustermer är sällan giltigt, speciellt inte då identifieringen görs med återkopplat system.

ARMAX-modell

Ett sätt att modellera en process som störs av korrelerat brus är :

$$\begin{aligned}A(q^{-1})y(k) &= B(q^{-1})u(k) + C(q^{-1})e(k) \\ \text{där} \\ A(q^{-1}) &= 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \\ B(q^{-1}) &= b_1q^{-1} + \dots + b_{nb}q^{-nb} \\ C(q^{-1}) &= 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc}\end{aligned}\tag{3.8}$$

där $e(k)$ är Gaussiskt vitt brus i diskret tid. (ARMAX). Polynomet $C(q^{-1})$ 'färgar' bruset. Denna struktur kan ej direkt skrivas om på regressionsform och användas vid identifiering eftersom det vita bruset $e(k)$ ej är mätbart. Det enklaste sättet att komma runt detta är att approximera $e(k)$ med prediktionsfelet $\varepsilon(t)$ i (3.3) eller (3.5). Görs detta kan modellen skrivas på regressionsform enligt :

$$y(t) = \varphi_1(t)\theta_1 + \varphi_2(t)\theta_2 + \dots + \varphi_{na+nb+nc}(t)\theta_{na+nb+nc} = \varphi^T(t)\theta(t)$$

där

$$\varphi^T(t) = (-y(t-1) \dots -y(t-na) \quad u(t-1) \dots u(t-nb) \quad \hat{\varepsilon}(t-1) \dots \hat{\varepsilon}(t-nc)) \quad (3.9)$$

$$\theta = (a_1 \dots a_{na} \quad b_1 \dots b_{nb} \quad c_1 \dots c_{nc})$$

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

Parametrarna θ ovan kan nu skattas med hjälp av samma algoritm som i ARX-fallet. Denna identifieringsmetod brukar kallas utökade rekursiva minstakvadratmetoden. (ELS).

Excitation

Ett problem vid identifieringen är att insignalerna till estimatorn måste vara tillräckligt exciterande, dvs. måste ha tillräckligt effektinnehåll, för att den identifierade modellen skall vara tillförlitlig. Den identifierade modellen kommer att ha bäst anpassning i de frekvensband där mätdata har störst effektinnehåll. För dålig excitation kan även leda till problem i form av estimatorupprivning, ett problem som behandlas i simuleringssnittet.

Processens naturliga brusnivå ger tillräcklig excitation för att ge trovärdiga skattningar av A- och C-polynomen i ARMAX-modellen (3.8) men innehåller väldigt lite information om B-polynomet. För att skatta B-parametrarna krävs börvärdesändringar eller liknande, som emellertid är mycket sällsynta i detta fall. För att ge en tillförlitlig modell måste estimatorn därför tillföra en störsignal till processen. Om störsignalens amplitud väljs rätt kan vi få en tillförlitligare modell av processen utan att försämra reglerprestandan nämnvärt.

3.2 SIMULERING

Val av modellordning

För att kunna få en bra reglering krävs att processmodellen som ligger till grund för regulatordesignen beskriver processen väl. Om en felaktig modellstruktur ansätts, eller dess gradtal sätts för lågt kan mätvärdena inte anpassas på ett bra sätt. Sätts gradtalen onödigt högt får vi onödigt tunga beräkningar i identifierings- och designsteget. I samband med adaptiv reglering väljer man oftast lägsta möjliga modellordning som ger någorlunda bra resultat.

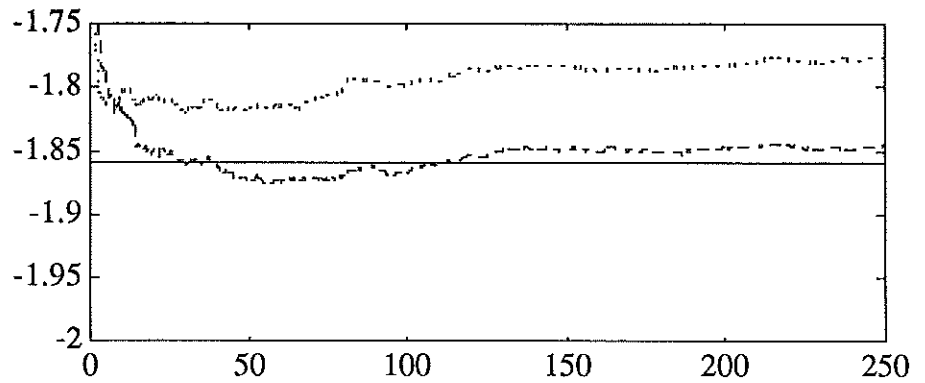
För att välja lämpligt ordningstal gjorde jag en serie identifieringar av koefficienter i en ARMAX-modell av varierande ordning. Identifieringar gjordes med ELS på öppet system vid arbetspunkterna 3300 kg/s och 5000 kg/s. För att driva systemet användes en PRBS med amplitud $\pm 0.3\%$. Variansen av prediktionsfelet sjunker efterhand som modellenpassningen blir bättre.

<i>gradtal</i>	<i>V[ε]</i>	<i>V[ε]</i>
<i>na=nb=nc</i>	3300 kg/s	5000 kg/s
1	0.3232	0.1051
2	0.0560	0.0576
3	0.0557	0.0563
4	0.0555	0.0563

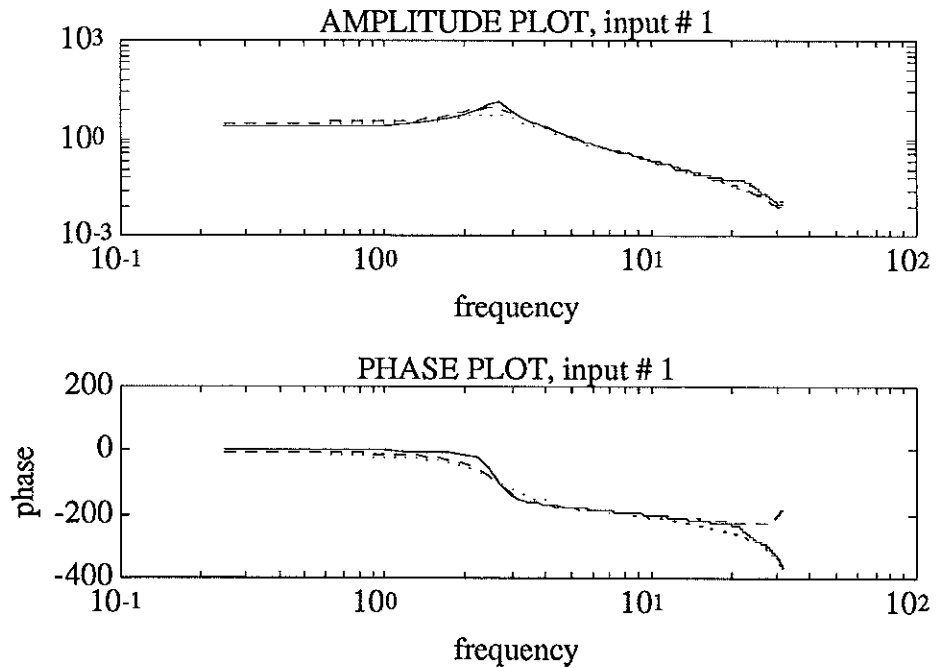
Ur mätvärdena i tabellen ovan ser vi att ökningen av modellordningen från ett till två gav en kraftig vinst, medan vidare ökning knappast lönar sig alls. För att hålla nere beräkningsbördan i designfasen valde jag approximera processen med en andra ordningens modell.

Jämförelse av RLS och ELS

Olika identifieringsmetoder kan ge olika värden på skattningarna. I vår processmodell (2.1) är brusets som påverkar systemet i själva verket inte okorrelerat och RLS-algoritmen konvergerar därför inte mot de sanna parametervärdena. ELS-algoritmen skattar och kompenserar för brusets korrelation och ger därför bättre skattningar. Figur 3.1 visar skattningarna av parametern a_1 för RLS och ELS-metoden. Vid skattningen är vi endast intresserade av processens dominerande dynamik och det 'sanna' värde som refereras till i figurtexten är första koefficienten i $A_u(q)$ i processmodellen (A.3). De övriga parametrarna uppvisar likartat beteende. Experimentet gjordes med slutet system vid arbetspunkten 3300 kg/s. För att excitera systemet användes en PRBS med amplitud $\pm 1\%$.



Figur 3.1 - Skattningarna av parametern a_1 vid användning av ELS (streckad) och RLS (prickad). Den heldragna linjen innebär markererar parameterns 'sanna' värde (se text).

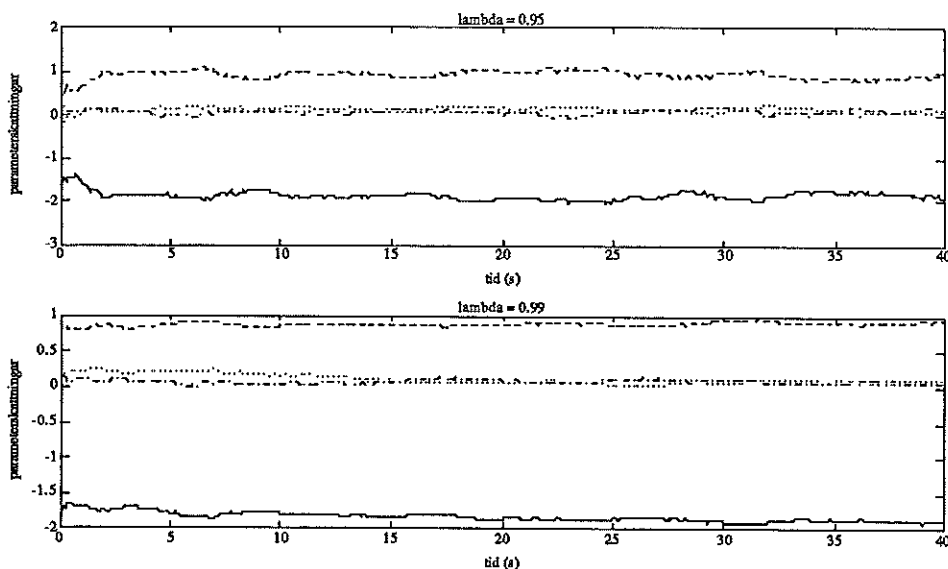


Figur 3.2 - Bodediagram för verkliga systemet(heldragen), ELS-skattningen(streckad) och RLS-skattningen(prickad).

Även av bodediagrammet i figur 3.2 framgår att ELS ger bättre skattningar än RLS. ELS valdes därför för parameterskattningarna.

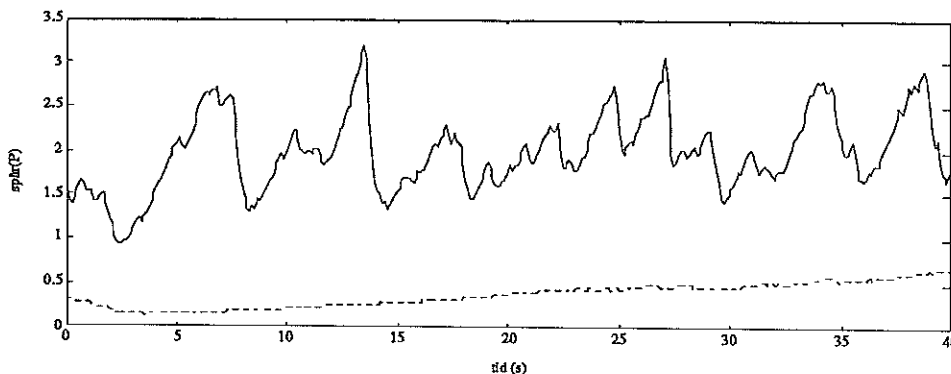
Exponential forgetting

För att kunna följa förändringar i processen valde jag att använda RLS with exponential forgetting (3.6). Som tidigare nämnts bestämmer λ vikterna mellan nya och äldre värden vid skattningen av parametrarna. Då λ ändras påverkas först och främst parameterskattningarnas varians och förmågan att följa snabba processförändringar. Om λ väljs för stort kommer estimatorn inte att hinna upptäcka snabba processvariationer tillräckligt snabbt. Om λ väljs litet tas få mätningar med i skattningarna och parametervariansen blir stor. Om excitationen är dålig kan dessutom P-matrisen växa okontrollerat. För att närmare undersöka inverkan av λ simulerade jag öppna systemet för några olika värden. Simuleringen gjordes med öppet system vid arbetspunkten 3300 kg/s och som insignal användes en PRBS med amplitud 0.3 %. I figur 3.3 finns parameterskattningarna plottade för $\lambda = 0.95$ och $\lambda = 0.99$.



Figur 3.3 - Skattningar av parametrarna i A- och B-polynomen av en ARMAX-modell för $\lambda = 0.95$ och $\lambda = 0.99$.

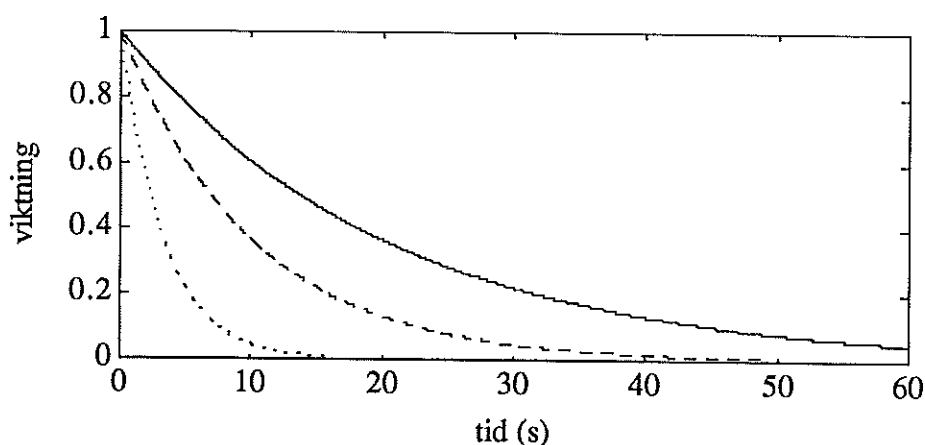
Man ser att parametrarnas varians minskar då λ närmar sig 1, men för att kunna följa processförändringar krävs en glömskefaktor skiljd från 1.



Figur 3.4 - Variationerna av P-matrissens spår för $\lambda = 0.95$ (heldragen) och $\lambda = 0.99$ (streckad). Spåret kan ses som den totala osäkerheten i parametrarna.

Vid varje uppdatering av P-matrisen enligt (3.5) skalas P-matrisen upp en faktor $1/\lambda$. Samtidigt skalas P-matrisen ner beroende på excitationen i systemet. Om excitationen är för dålig kommer P-matrisen inte att konvergera. Exempel på detta ser vi i figur 3.4 där P-matrissens spår under simuleringen visas.

För att kunna ställa in λ krävs att man tar hänsyn till hur snabbt processförändringar måste upptäckas och därefter ser till att excitationen är tillräcklig för att hålla P-matrisen under kontroll. Det är därför önskvärt att välja λ så stort som möjligt, utan att för den skull göra det omöjligt för skattaren att upptäcka processförändringar i tid.



Figur 3.5 - Viktningen av mätvärden för $\lambda = 0.995$ (heldragen), 0.99 (streckad) och 0.95 (prickad) vid samplingstid 0.1 s.

Viktningen av mätvärden åskådliggörs av figur 3.5. Man kan se att för $\lambda = 0.95$ har ett 10 sekunder gammalt mätvärde nästan ingen inverkan, men för $\lambda = 0.99$ har mätvärdet nästan hälften så stor betydelse som de allra nyaste. Antar vi att effektnivån i insignalerna är konstant svarar arean under kurvorna fram till en viss tidpunkt mot betydelsen av mätvärdena fram till denna tidpunkt.

Ett rimligt krav är att processförändringar skall upptäckas inom en halv minut. Kravet att skattningarna till 90 % skall bestämmas av sista 30 sekundernas mätvärden kan alltså, då samplingstiden är 0.1 s skrivas

$$\frac{\sum_{k=0}^{300} \lambda^k}{\sum_{k=0}^{\infty} \lambda^k} = 0.90 \Leftrightarrow \lambda = \sqrt[601]{1-0.95} \approx 0.9901 \quad (3.10)$$

alltså väljs $\lambda = 0.99$

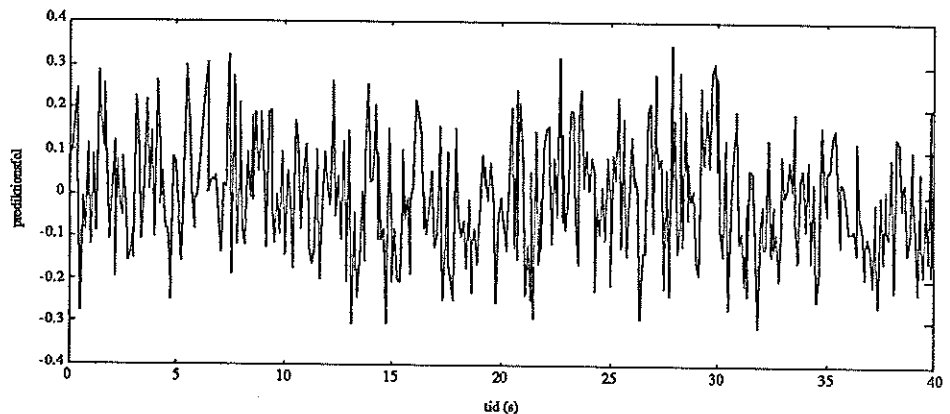
Estimatoruppvridning

Vid användning av glömskefaktor och excitationen i systemet är dålig kommer estimatorn långsamt att glömma sina parametervärden sedan de en gång konvergerat. Orsaken till detta är att sedan parametrarna konvergerat förbättras regleringen och variationerna i utsignalen dämpas av regulatorn. Estimatorn 'glömmer' mätdata i konstant takt vare sig systemet är tillräckligt exciterat eller ej och därför växer P-matrisen och parametrarna börjar svänga kring sina sanna värden. Fenomenet brukar kallas för estimatoruppvridning.

Det finns olika sätt att hantera detta men de flesta metoder bygger antingen på att stänga av estimatorn då excitationen är dålig eller att på något vis reglera P-matrisens storlek och därmed variansen i parametrarna.

Directional forgetting[2] är en metod som bygger på glömma data bara i den takt ny kommer in. Tyvärr tvingas P-matrisen till en förutbestämd storlek vare sig det finns excitation eller ej och parametervariansen regleras men man kan inte vara säker på att parametrarna oscillerar kring sina sanna värden.

Att stänga av adaptationen (dödzon) är ej lämpligt i vårt fall eftersom en snabb sänkning av arbetspunkten kan medföra instabilitet om parametrarna inte korrigeras i tid. Det är det svårt att momentant bedöma om excitationen är tillräcklig eftersom den ökning av prediktionsfelet som sker på grund av modellfel nästan helt dränks av processens brusnivå. Figur 3.6 visar prediktionsfelet under simulering med en kraftig ökning av prediktionsfelet.



Figur 3.6 : Prediktionsfel, vid tiden 20 s ändras arbetspunkten från 2400 till 5000 kg/s.

Ett annat sätt att reglera variansen i parametrarna är att tillföra en extern störning till systemet. Detta exciterar systemet och estimatorn ger säkrare parameterskattningar som också oscillerar kring sina rätta värden. En extern störning av systemet kan accepteras om den inte medför alltför kraftig ökning av utsignalsvariationerna. Störningen minskar variansen i parametrarna och ökar därför säkerheten i form av en mer korrekt modell. Om störningens amplitud regleras på ett bra sätt bör säkerhetsvinsten uppväga förlusten i form av variationer i utsignalen.

4. REGULATORDESIGN

Jag har valt att använda polplaceringsmetoden[1] för regulatordesign. Principen bakom metoden är att bestämma det slutna systemets egenskaper genom att flytta dess poler. Metoden är egentligen bäst lämpad för servoproblem, men lägger man lite tid på inställningen av regulatorn kan man tillämpa metoden även på reglerproblem. Polplaceringsmetoden är lätt att förstå och man kan enkelt införa tidsfördröjning och integrator. Polplaceringsmetoden är också väl lämpad för implementering i datorprogram.

Eftersom parameterskattaren reagerar långsamt på processförändringar kommer regulatordesignen periodvis att bygga på en felaktig processmodell. Därför är det viktigt att designmetoden ger en regulator som är robust mot modellfel.

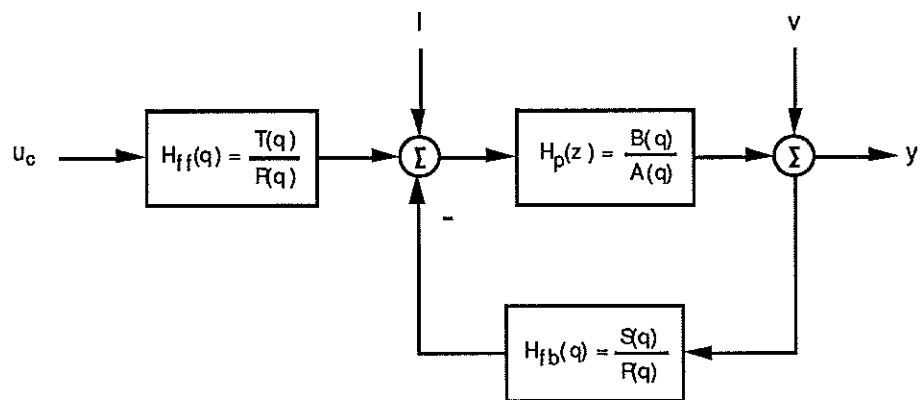
4.1 TEORI

Designparametrar i polplaceringsalgoritmen är modellsystemets karakteristiska polynom $A_m(q)$ och observerarpolynomet $A_o(q)$.

Regulatorn är på formen

$$R(q) u(k) = -S(q) y(k) + T(q) u_c(k) \tag{4.1}$$

där R och S ges av (4.4)



Figur 4.1: Blockschema för slutna systemet.

Utsignalen hos det slutna systemet ges då av

$$y = H_{cl}(q) u_c + H_l(q) l + H_v(q) v = \frac{BT}{AR+BS} u_c + \frac{BR}{AR+BS} l + \frac{AR}{AR+BS} v \tag{4.3}$$

där v betecknar brus och l en laststörning. Processen har i vissa områden ett instabilt nollställe varför förkortning av nollstället i regulatorn ej tillåts. För att kunna kompensera laststörningar och modellfel i lågfrekvensområdet vill vi att regulatorn skall innehålla en integrator. Detta fås om $R(q)$ innehåller faktorn $(q-1)$. Regulatorn skall innehålla en tidsfördröjning och R-polynomet måste därför vara av högre ordning än såväl S- som T-polynomet. Detta åstadkommes genom att införa en extra pol i vårt modellpolynom.

Eftersom återkoppling endast sker från utsignalen måste övriga tillstånd skattas. Observerarens karakteristiska ekvation specificeras i observerarpolynomet A_o . Normalt väljs observerarens poler något snabbare än systemet.

En algoritm för att generera polplaceringsregulatorn enligt ovan ges nedan :

$$\begin{aligned} \text{Välj designparametrar : } A_o \text{ och } A_m, \text{ sätt } t_0 &= \frac{A_m(0)}{B(0)} \\ \text{Lös ekvationen : } A(q-1)R_1 + BS &= A_o A_m \\ \text{Bildra : } T = t_0 A_o \text{ och } R &= (q-1)R_1 \end{aligned} \quad (4.4)$$

Och styrlagen ges alltså av :

$$R(q) u(k) = -S(q) y(k) + T(q) u_c(k) \quad (4.5)$$

Kretsförstärkningsfunktionen

Kretsförstärkningsfunktionen definieras som :

$$H_{ig}(q) = H_{fb}(q)H_p(q) = \frac{B(q) S(q)}{A(q) R(q)} \quad (4.6)$$

Denna kan beräknas om systemet antas vara fullständigt givet av processmodellen (2.1). Detta antagande är befogat eftersom vi bara är intresserade av polplaceringsmetodens allmänna egenskaper och inga absoluta siffervärden.

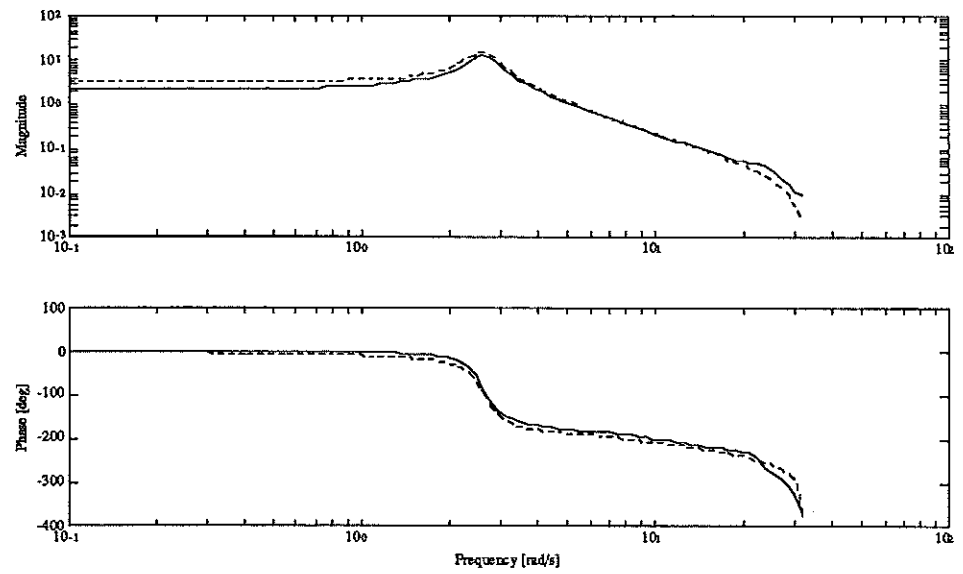
Kretsförstärkningsfunktionen ger god insikt om slutna systemets stabilitetsegenskaper. Bland annat kan fas- och amplitudmarginal bestämmas.

4.2 SIMULERING

För att få en uppfattning om regulatorns prestanda under ideala förhållanden har en serie simuleringar utförts med ett antagande om en konstant, fullständigt känd process. Med stöd av dessa valdes sedan designparametrarna som måste specificeras vid regulatordesignen. För att inte utsätta regulatorn för omodellerad dynamik modellerades processen vid dessa simuleringar med en reducerad processmodell av andra ordningen :

$$y(k) = \frac{0.6870q + 0.9731}{q^2 - 1.8684q + 0.9333} u(k) + \frac{q^2 - 0.0011q - 0.3510}{q^2 - 1.8684q + 0.9333} e(k) \quad (4.7)$$

där $e(k)$ är vitt gaussiskt brus. Överensstämmelsen mellan den reducerade och ordinarie processmodellen åskådliggörs i figur 4.2.



Figur 4.2 : Överensstämmelsen mellan den reducerade (streckad) och ordinarie (heldragen) processmodellen.

De resultat som uppnås här kommer troligen inte att kunna uppnås i det adaptiva fallet, eftersom vi använder oss av en förenklad modell, men resultaten kan användas som måttstock vid utvärdering av den adaptiva regulatorn.

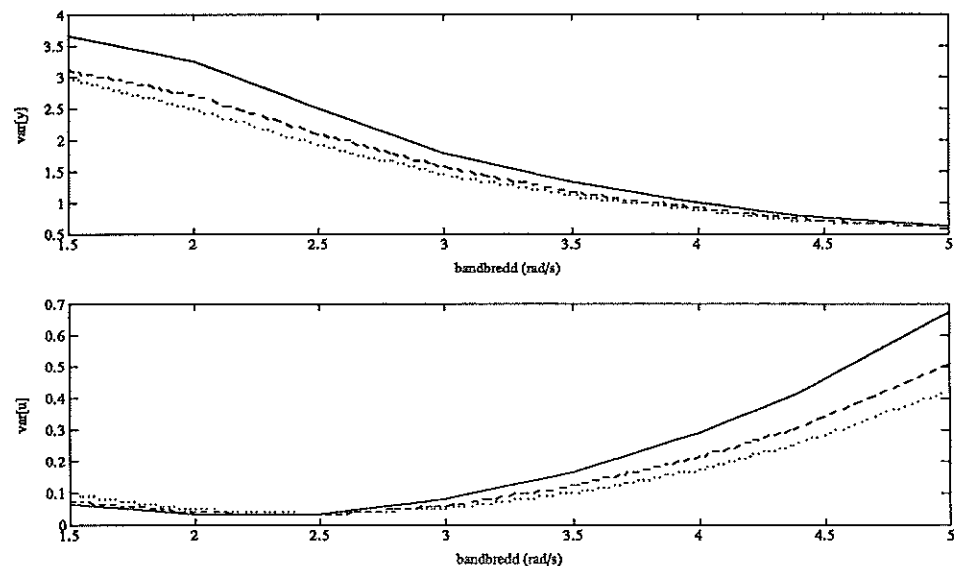
Val av modellsystemets poler

Modellsystemets poler väljs som

$$A_m(q) = \text{ZOH}^1 \left\{ (s^2 + 2\omega_0\zeta s + \omega_0^2)(s + \omega_0) \right\} \quad (4.8)$$

där ω_0 är den önskade bandbredden och ζ den önskade dämpningen hos slutna systemet (butterworth). För att bestämma lämpliga värden på ω_0 och ζ simulerades systemet för några olika värden.

Valet av ω_0 styr hur häftigt regulatorn reagerar på börvärdesändringar och reglerfel. Väljs ω_0 stort fås ett snabbt system som kräver stora styrsignaler och en exakt modell i ett brett frekvensintervall. För att bestämma lämpliga värden på ω_0 och ζ simulerade jag systemet (4.7) reglerat med styrlagen enligt (4.5). Börvärdet hölls konstant.



Figur 4.2 : Variansen i utsignalen (y) och styrsignalen (u) som funktion av ω_0 , för olika $\zeta = 0.5$ (heldragen), 0.7 (streckad) och 0.9 (prickad).

I figur 4.2 visas hur variansen i styrsignal och utsignal påverkas av olika ω_0 och ζ . Man kan se att då ω_0 ökas minskas utsignalvariansen men styrsignalens varians ökar. En rimlig avvägning fås om man väljer $\omega_0 = 3.0$. Detta ger $\omega_0 h = 0.30$ vilket passar bra in i tumregeln enligt [1] för valet av samplingstid. Valet av ζ påverkar inte varianserna nämnvärt och sätts därför något godtyckligt till 0.7 .

Val av observerarpolynom

Enligt gradtalsreglerna behöver vi en andra ordningens observerare. Observerarens poler placeras på samma sätt som modellsystemets men en faktor α snabbare. Observerarens karakteristiska polynom ges då av

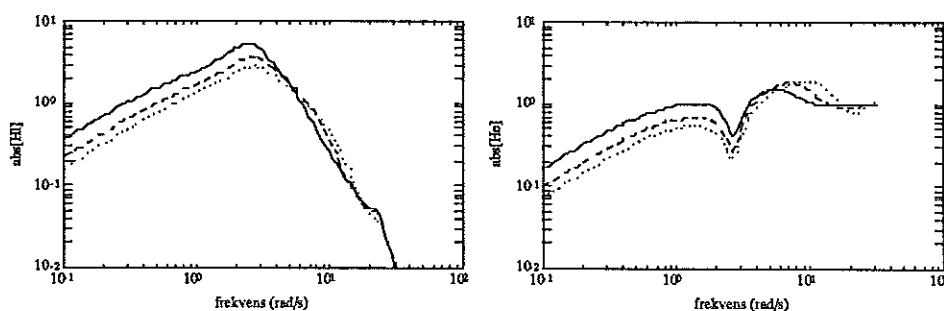
$$A_o(q) = \text{ZOH} \left\{ (s^2 + 2\alpha\omega_0\zeta s + (\alpha\omega_0)^2) \right\} \quad (4.9)$$

¹ ZOH - nollte ordningens sampling, de kontinuerliga polerna $s=p$ avbildas på de tidsdiskreta $z = e^{-hp}$.

Observeraren påverkar hur störningar fortplantar sig genom systemet. Väljer man observeraren snabb dämpas lågfrekventa störningar ut snabbt men regulatorn blir samtidigt bruskänslig. Slutna systemets förstärkning av last(l)- respektive mätstörningar(v) ges av

$$|H_l(q)| = \left| \frac{H_p(q)}{1 + H_{lg}(q)} \right| \text{ och } |H_v(q)| = \left| \frac{1}{1 + H_{lg}(q)} \right| \quad (4.10-11)$$

Förstärkningarna vid 3300 kg/s visas i figur 4.3.



Figur 4.3 : Förstärkning av laststörningar (vänster) och mätstörningar (höger) för $\alpha=1$ (heldragen), $\alpha=2$ (streckad) och $\alpha=4$ (prickad). ($\omega_0 = 3.0$)

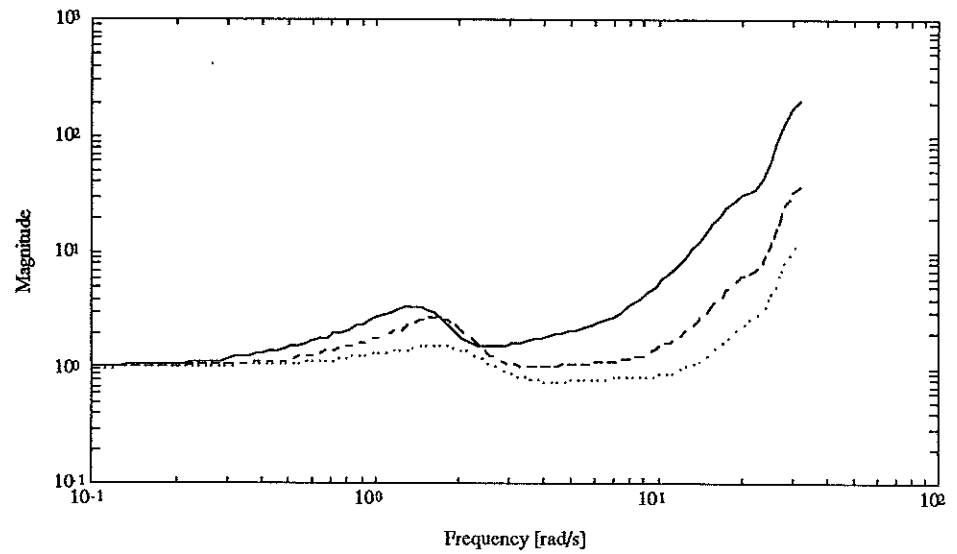
Man kan se att en ökning av α kan ge bättre dämpning av laststörningar och mätstörningar upp till ca 7 rad/s till priset av högre högfrekvensförstärkning. Valet $\alpha = 4$ ger en bra avvägning.

Känslighet för modellfel

Ett tillräckligt villkor för slutna systemets stabilitet är

$$\left| \frac{\Delta H(q)}{H_p(q)} \right| < \left| \frac{1 + H_{lg}(q)}{H_{lg}(q)} \right|, |q|=1 \quad (4.12)$$

Antar vi även här att processen beskrivs fullständigt av processmodellen (2.1) kan toleransen mot modellfel bestämmas. Figuren nedan visar toleransen för modellfel i några olika fall.



Figur 4.4 : Tolerans mot modellfel för $\omega_0 = 2$ (heldragen), 3(streckad) och 4 (punktad). ($\alpha = 4$)

Modellen måste alltså vara mera tillförlitlig i ett bredare frekvensband om den önskade bandbredden ökas. Man kan se att toleransen är som lägst vid frekvensen ω_0 . För $\omega_0 = 3$ behövs en bra modell upp till ca 20 rad/s.

Skydd mot integratoruppvridding

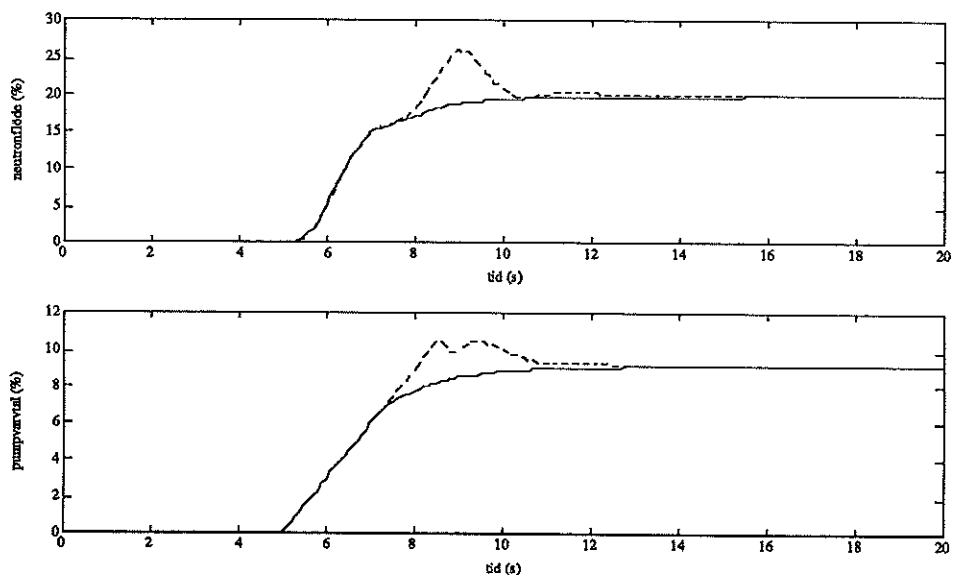
Eftersom styrsignalen begränsas bör skydd mot integratoruppvridding i regulatoren byggas in. Regulatorns integrator integrerar om vi får ett kvarstående reglerfel som följd av att styrsignalen mättat kommer integratorn att 'ladda' upp sig till allt ökande värden. Skyddet kan åstadkommas genom att modifiera styrlagen enligt

$$\begin{cases} A_f v(k) = T u_c(k) - S y(k) + (A_f - R) u(k) \\ u(k) = \text{sat } v(k) \end{cases} \quad (4.13)$$

Polynomet A_f skall innehålla observerarpolynomets dynamik och dessutom krävs, för att styrlagen skall vara realiserbar, att A_f skall vara av samma eller högre grad än R . A_f väljs därför som

$$A_f(q) = \text{ZOH} \{ (s^2 + 2\alpha\omega_0\zeta s + (\alpha\omega_0)^2)(s + \alpha\omega_0) \} \quad (4.14)$$

Nyttan av skyddet ses tydligt i figuren nedan. För att öka tydligheten har störningstermen satts till noll.



Figur 4.5 : Stegsvär med (heldragen) och utan(streckad linje) skydd mot integratoruppvridding.

Överslängningen i det oskyddade fallet beror på att det tar tid att 'tömma' integratorn sedan styrsignalen mättats.

Jämförelse med PID-regulator

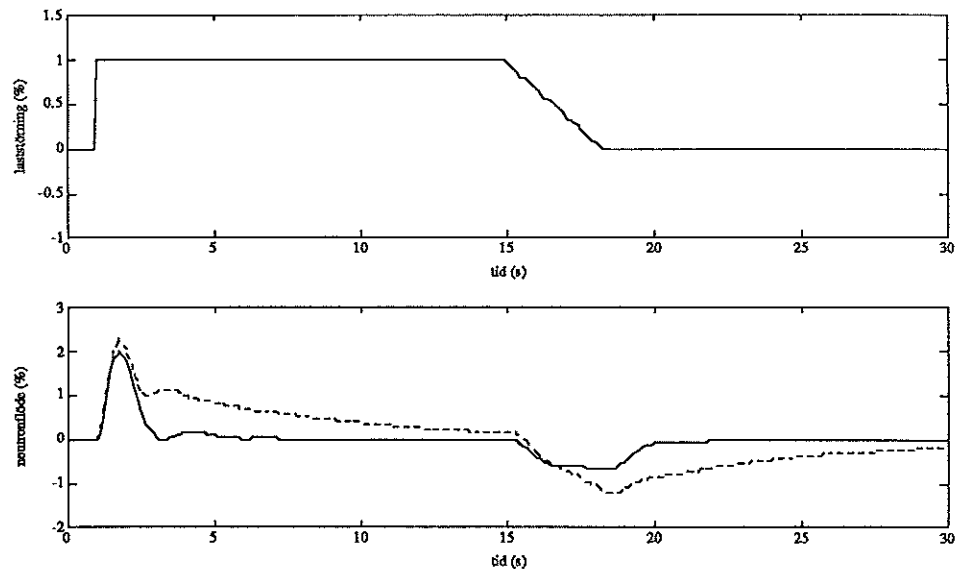
Med stöd av simuleringarna valdes en polplacerings regulator med $\omega_0 = 3$, $\zeta = 0.7$, $\alpha = 4$. Som jämförelse användes en PID-regulator med parametrarna $K = 0.2$, $T_i = 2$ och $T_d = 0.6$. För att få en rättvis jämförelse användes här den fullständiga processmodellen (2.1). PID-parametrarna har valts för att ge en dämpning av neutronflödesvariationerna som är ungefär likvärdig med polplaceringsregulatorn. Detta bör ej ses som en optimal inställning av PIDn och jämförelsen är kanske ändå inte helt rättvis. För att kontrollera toleransen mot modellfel ställdes regulatorerna in vid 3300 kg/s men provades också vid andra arbetspunkter.

Tabellen nedan visar varianserna i ut- och insignal vid stationär drift.

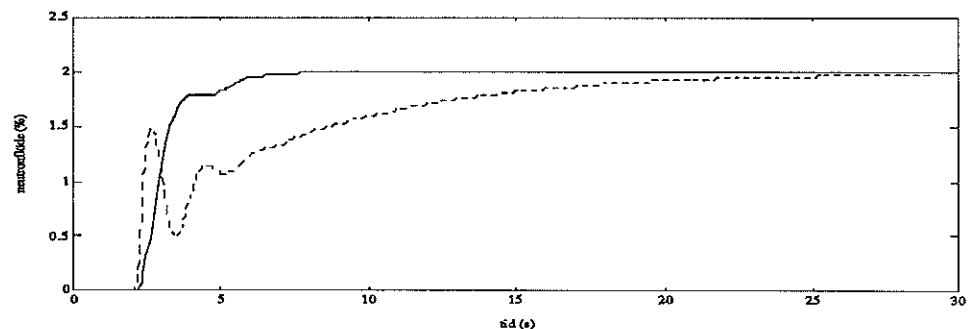
Polplacering :			PID :		
HC-flöde	V[u]	V[y]	HC-flöde	V[u]	V[y]
2400	0.3055	1.6826	2400	0.2729	1.5648
3300	0.2567	0.9177	3300	0.1944	0.9095
5000	0.2437	0.2906	5000	0.1224	0.3311

Tabellen visar att PID-regulatorn är något bättre än polplaceringsregulatorn på att minska variationerna i utsignalen. PID-regulatorn ger också lägre varians i styrsignalen.

Även förmågan att kompensera bort laststörningar är viktig. Denna förmåga behövs t.ex. då arbetspunkten för HC-pumparna är felaktigt inställd av processimulatorens. Som figuren nedan visar hanterar polplaceringsregulatorn laststörningar bäst.



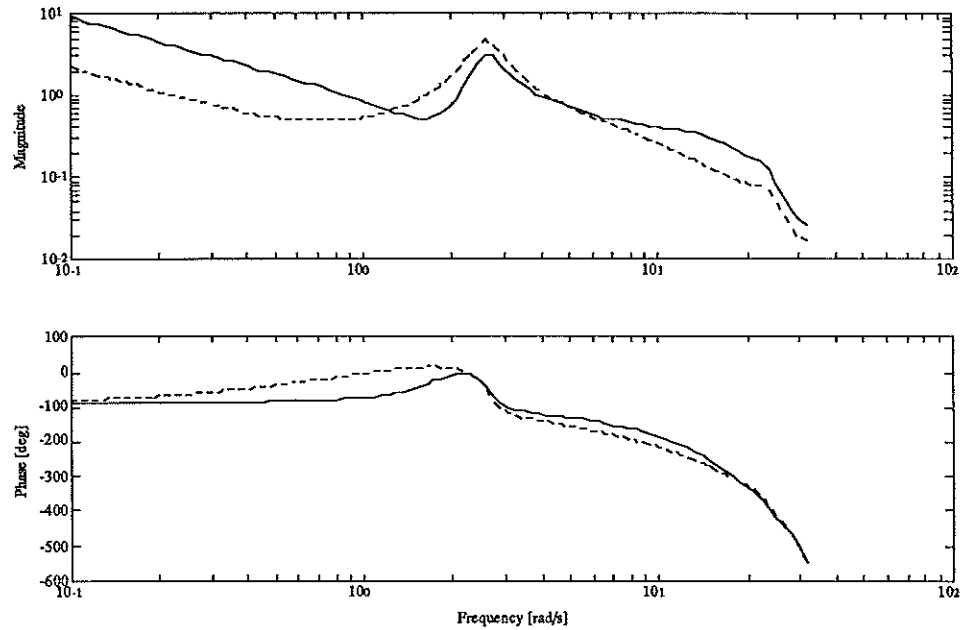
Figur 4.6 : Kompensering av laststörning vid 3300 kg/s. Helderagen - polplacering, streckad - PID.



Figur 4.7 : Svar på stegändring i börvärdet vid 3300 kg/s. Helderagen - polplacering, streckad - PID.

Figur 4.7 visar svaren på en stegändring i börvärdet för de bägge regulatorerna. PID-ns stegsvar är något snabbare men också sämre dämpat.

För att bedöma stabilitetsmarginalerna användes krets-förstärkningsfunktionen :



Figur 4.8 : Krets förstärkningsfunktionen för polplacering (heldragen) och PID (streckad).

<i>Polplacering :</i>			<i>PID :</i>		
<i>HC-flöde</i>	<i>amplitud- marginal</i>	<i>fas- marginal</i>	<i>HC-flöde</i>	<i>amplitud- marginal</i>	<i>fas- marginal</i>
2400	2.5809	47.9897	2400	2.3220	26.9487
3300	2.4090	61.0613	3300	2.3343	36.8106
5000	1.5035	57.2484	5000	1.6778	37.8508

Polplaceringsregulatorn ger alltså ca 20° bättre fasmarginal än PID-regulatorn. Detta beror på att den lyfter faskurvan kraftigt för höga frekvenser. PIDn är av lägre ordning och förmår inte lyfta fasen lika mycket. Detta är nödvändigt för att få ett robust reglering av neutronflödet och någon form av kompenseringslänk bör användas vid PID-reglering.

Polplaceringsmetoden kan alltså ge bättre stabilitetsmarginaler med bibehållen dämpning av neutronflödesvariationerna. Designberäkningarna är enkla att implementera i datorprogram vilket gör att metoden lämpar sig väl för adaptiv reglering.

SJÄLVINSTÄLLANDE REGULATOR

De olika delarna kopplades ihop till en självinställande regulator enligt figur 1.1. Designparametrarna i estimator- och regulator valdes enligt kapitel 3 och 4 dvs :

Regulatordesign : $\omega_0 = 3$, $\zeta = 0.7$, $\alpha = 4$

Estimator : $n_a=n_b=n_c=2$, $\lambda = 0.99$

5.1 SIMULERING

För att förhindra integratoruppvridning och därmed hålla osäkerheten i estimatorns skattningar under kontroll använder jag en störsignal. Den ökade säkerheten i parameterskattningarna ger en säkrare reglering.

För att undersöka effekten av en extern störning utfördes en serie simuleringar där styrsignalen stördes av en PRBS. Störsignalens amplitud sattes till

$$A_{\text{stör}} = \min(3, k \sum_{i=1}^{n_a+n_b} P(i,i)) \quad (3.11)$$

Detta kan ses som en proportionell reglering av variansen.

Ytterligare ett sätt att göra det adaptiva systemet säkrare är filtrera insignalerna till estimatorn. Processen har en mindre resonanstopp vid ca 25 rad/s som kan vålla problem i identifieringen. Endast signaler i det frekvensband som är intressant vid regulatordesignen skall släppas igenom. Vidare bör insignalernas medelvärden filtreras bort. Regulatorn behöver en bra modell upp till ca 20 rad/s.

Genom simulering kom jag fram till att ett butterworth bandpassfilter med passband [0.15, 20] rad/s fungerade bra. Även störsignalen som tillförs processen filtreras genom bandpassfiltret. För att undersöka inverkan av bandpassfiltret samt ställa in parametern k gjordes en simulering :

Utan bandpassfilter :

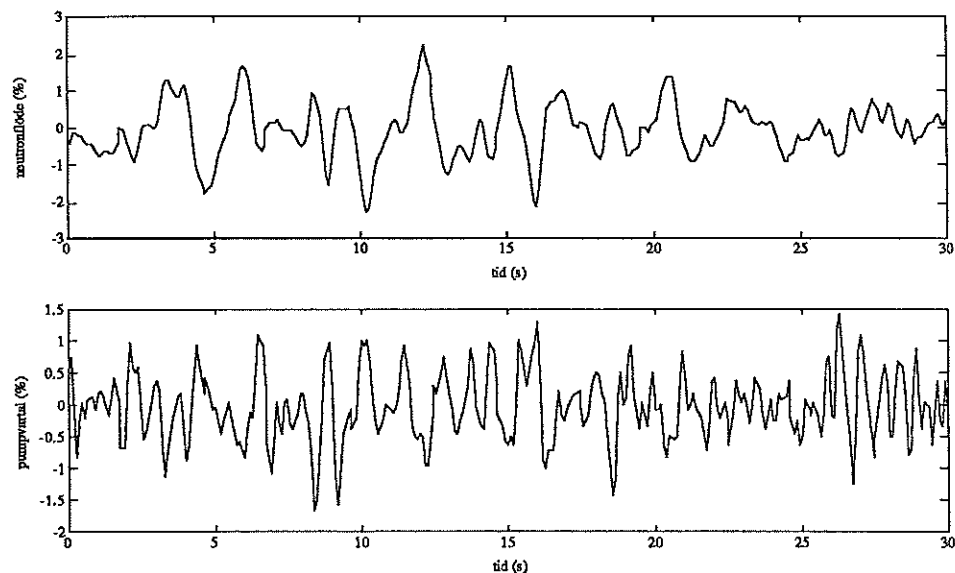
k	$V[u]$	$V[y]$	$V[a1]$	$V[a2]$	$V[b1]$	$V[b2]$	$V[c1]$	$V[c2]$
0.00	0.2999	0.9397	0.0051	0.0039	0.0303	0.0072	0.0088	0.0173
0.10	0.4385	0.7300	0.0094	0.0104	0.0219	0.0103	0.0174	0.0770
0.20	0.4582	0.7707	0.0060	0.0066	0.0121	0.0057	0.0119	0.0442
0.30	0.4623	0.8317	0.0052	0.0056	0.0089	0.0044	0.0103	0.0329
0.50	0.4871	0.9139	0.0044	0.0048	0.0063	0.0034	0.0088	0.0239
1.00	0.5609	1.1212	0.0032	0.0036	0.0038	0.0024	0.0073	0.0152

Med bandpassfilter :

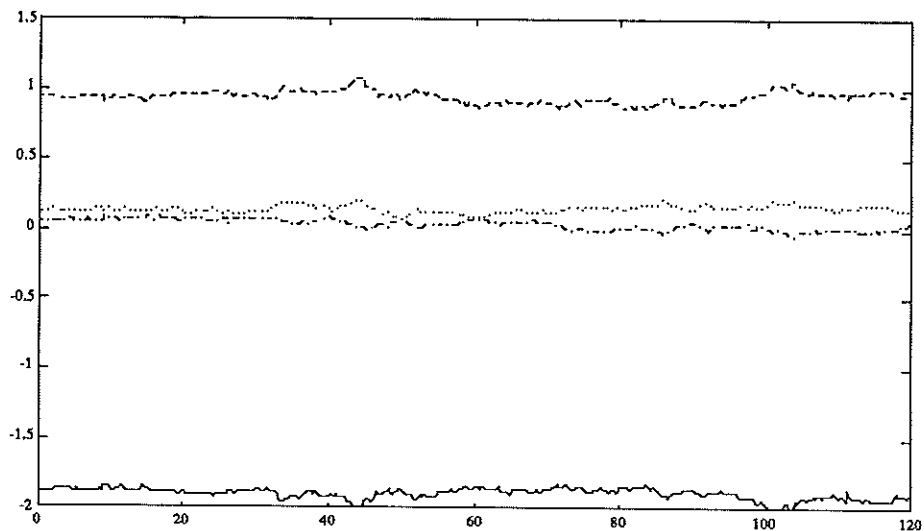
k	$V[u]$	$V[y]$	$V[a1]$	$V[a2]$	$V[b1]$	$V[b2]$	$V[c1]$	$V[c2]$
0.00	0.3888	0.6292	0.0062	0.0055	0.0021	0.0005	0.0105	0.0123
0.10	0.3944	0.6722	0.0032	0.0033	0.0018	0.0008	0.0112	0.0122
0.20	0.4136	0.7189	0.0022	0.0024	0.0017	0.0009	0.0119	0.0128
0.30	0.4362	0.7668	0.0018	0.0019	0.0015	0.0009	0.0119	0.0126
0.50	0.4791	0.8631	0.0014	0.0015	0.0011	0.0008	0.0111	0.0115
1.00	0.5764	1.0920	0.0010	0.0010	0.0008	0.0007	0.0106	0.0124

Bandpassfiltret minskar variansen i parametrarna och vi får därmed en bättre reglering. Väljs $k = 0.2$ ökar inte utsignalvariansen nämnvärt men parametervariansen minskar något.

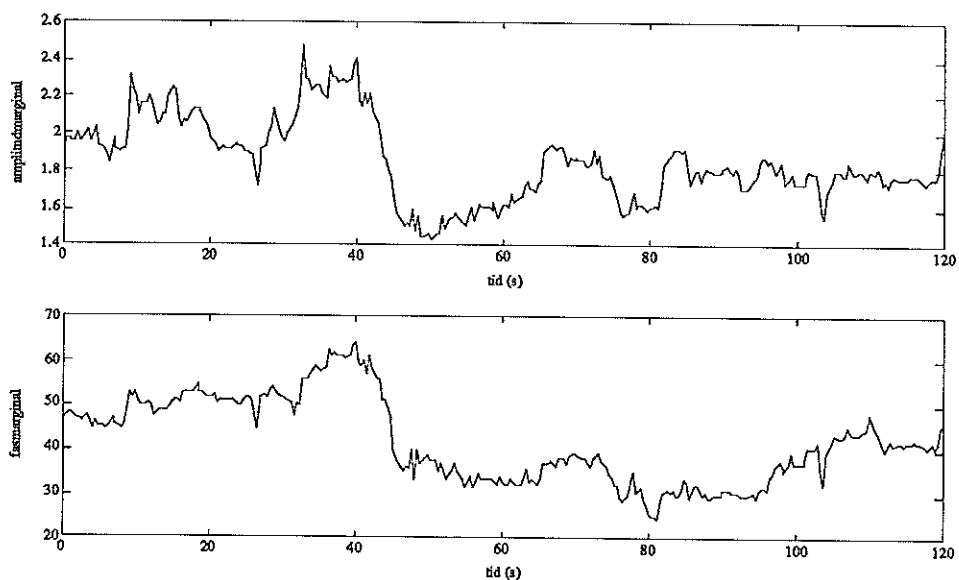
Figurerna 5.1-5.3 visar resultaten av en simulering med bandpassfilter enligt ovan och $k=0.2$. Beräkningen av stabilitetsmarginalerna är baserad på processmodellen (2.1).



Figur 5.1: Variationerna i ut- och styrsignal under drift.



Figur 5.2: Parameterskattningar a1-a2 (överst) och b1-b2 (underst).



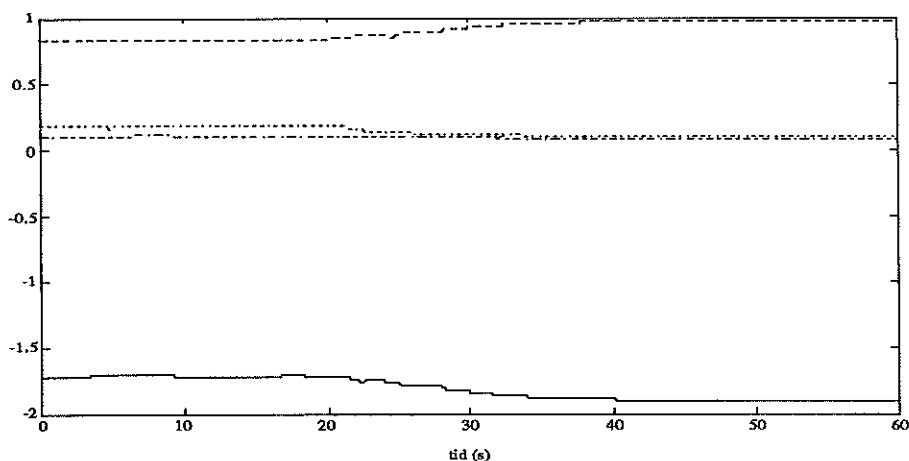
Figur 5.3: Stabilitetsmarginaler vid 3300 kg/s.

Ur figur 5.1 ser vi att styrsignalen är relativt brusig vilket framför allt beror på att regulatorn valts snabb, vilket är nödvändigt för att dämpa svängningarna i neutronflödet. Parameterskattningarna varierar alltså ganska kraftigt, men trots detta är stabilitetsmarginalerna är goda - i snitt ca 1.8 respektive 42°.

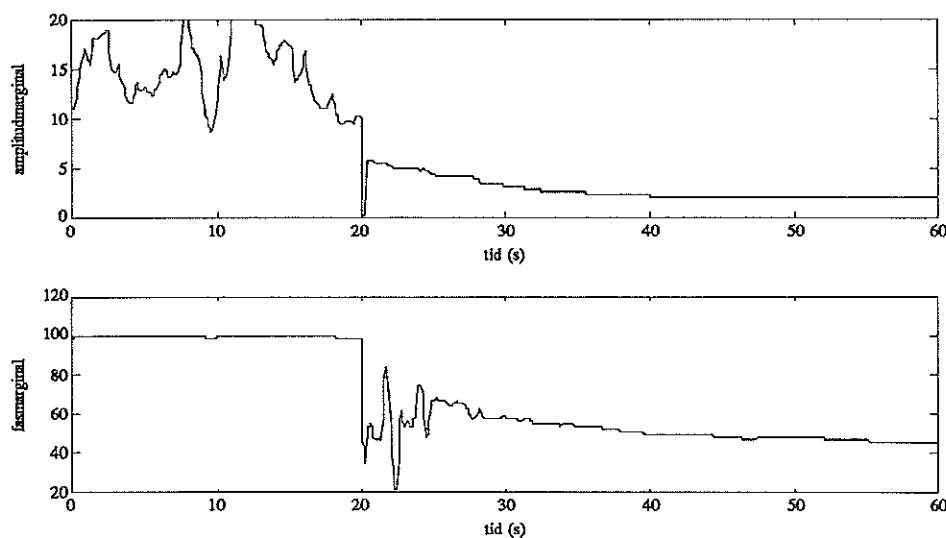
Följförmåga

Då HC-flödet minskas under ca 2500 kg/s blir processen instabil. För att kunna stabilisera systemet måste den adaptiva regulatorn snabbt kunna ställa in sig vid den nya arbetspunkten. Simuleringen startar med HC-flödet vid 5000 kg/s för att efter 20 sekunder momentant minskas till 2400 kg/s.

För att öka tydligheten stängdes processmodellens brusterm av i denna simulering. Brustermen påverkar ej förmågan att följa processförändringar.



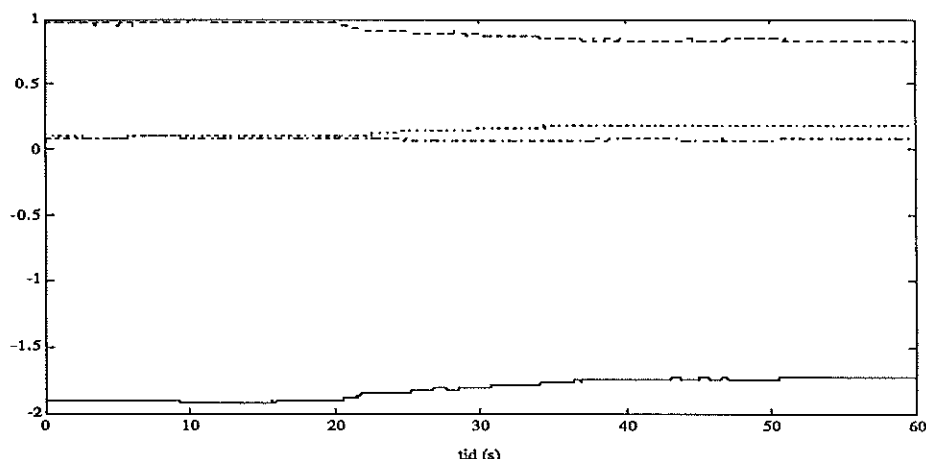
Figur 5.4 : Parameterskattningar vid snabb minskning av HC-flödet.



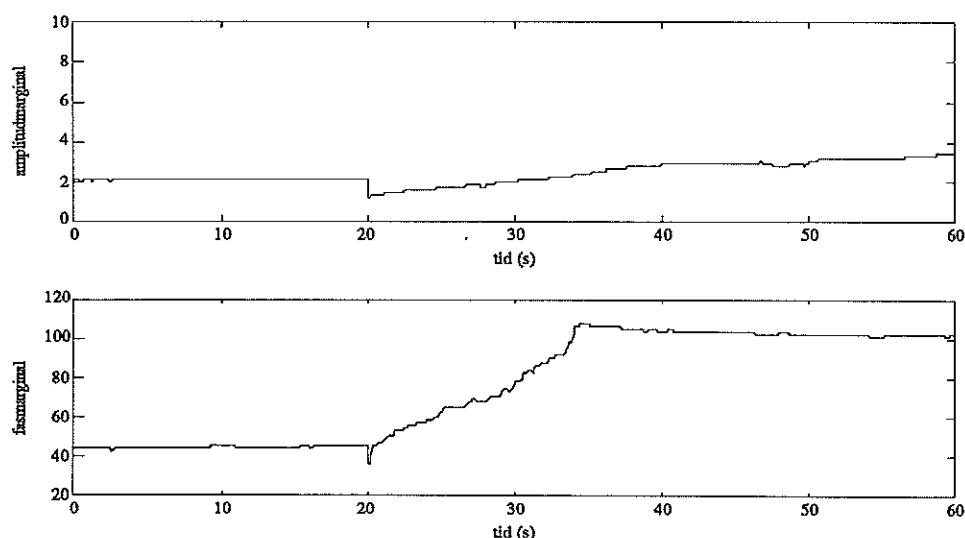
Figur 5.5 : Stabilitetsmarginaler vid snabb minskning av HC-flödet.

Den hastiga minskningen av HC-flödet gör alltså slutna systemet instabilt för ett ögonblick. Detta kompenseras dock mycket snabbt eftersom instabiliteten skapar excitation och parametrarna korrigeras så att instabiliteten försvinner.

Simuleringen startar med HC-flödet 2400 kg/s för att efter 20 s ökas momentant upp till 5000 kg/s.



Figur 5.6 : Parameterskattningar vid snabb ökning av HC-flödet.



Figur 5.7 : Stabilitetsmarginaler vid snabb ökning av HC-flödet.

De två simuleringarna visar att regulatorn ställer om sig på ca 20 s. Processvariationer som leder till instabilitet kommer att upptäckas och kompenseras för snabbare.

Stabilitetsmarginalerna är tillräckliga för att ge en säker reglering.

Jämförelse av adaptiv och fix regulator

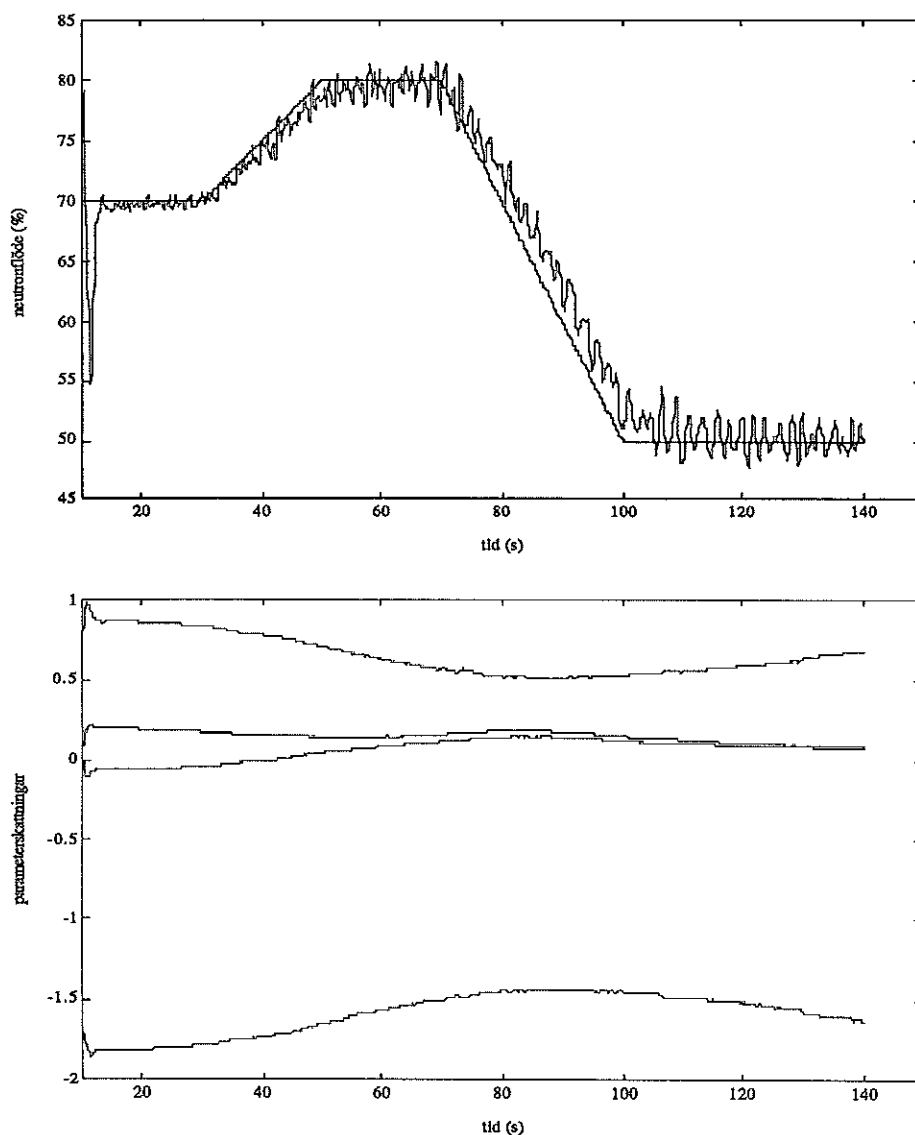
För att utvärdera den adaptiva regulatorn gjorde jag en jämförelse mellan den, en fix RST-regulator och en fix PID-regulator. De fasta regulatorerna är de som användes i kapitel 4. Regulatorernas dämpning av processens brusnivå finns tabellerad nedan. Test 1 & 2 mäts varianserna vid sänkning respektive ökning av HC-flödet föregående avsnitt. I test 3 hålls HC-flödet konstant som 3300 kg/s.

regulatortyp	Test 1		Test 2		Test 3	
	V[u]	V[y]	V[u]	V[y]	V[u]	V[y]
fix polplacering	0.2012	3.0303	0.1145	0.7246	0.1407	1.4325
PID	0.2093	1.6820	0.1452	1.0089	0.1548	1.1591
adaptiv polplac.	0.3517	0.9441	0.3042	0.6766	0.3968	0.6108

Den adaptiva regulatorn reglerar alltså bättre i samtliga test. Detta är underligt eftersom den fixa regulatorn borde vara ideal vid 3300 kg/s (Test 3). Troligen beror resultatet på att den adaptiva regulatorn hanterar den omodellerade dynamiken bättre än den fixa.

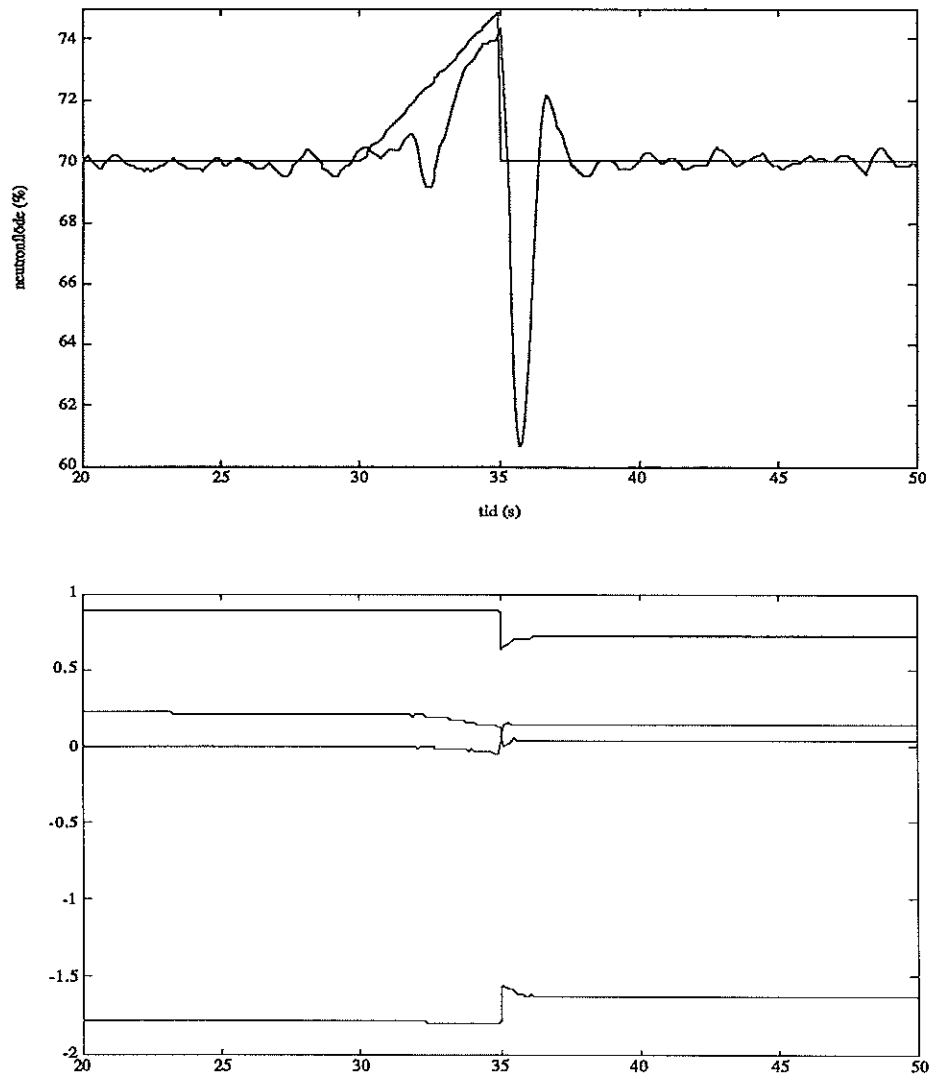
Den största fördelen med adaptiv reglering är ändå att regulatorn ställer om sig då processen ändras. Eftersom jag i mina simuleringar endast undersökt reglering vid HC-flöden 2400-5000 kg/s kommer vinsterna med den adaptiva regulatorn troligen att öka ytterligare då processen varierar mer.

Verifiering



Figur 5.8 : Simulering av variationer i effektläget.

Figur 5.8 visar resultatet av en simulering med BISON. Ändringen av effektläget gör att processen ändras. Resultaten är likartade med de som erhöles i SIMULINK.



Figur 5.9 : Störning i effektläget med konstant ånguttag.

Figur 5.9 visar resultat av ytterligare en simulering. Reaktorn körs med tryckregulatorn urkopplad och därmed konstant ånguttag. Då effektläget ändras långsamt fås en lugn uppgång men den snabba nedgången medför en häftig transient. Regulatorn stabiliserar dock processen. Denna simulering visar alltså att den adaptiva regulatorn kan användas för att öka säkerheten vid felfunktion hos andra reglersystem.

Verifieringen visar att den adaptiva regulatorn förmår att ställa in sig även till processmodellen i BISON. Slutligen skall sägas att BISON har mycket god överensstämmelse med verkligheten och regulatorn skulle troligen fungera även i en verklig reaktor.

6. KOMMENTARER

Säkerhet

Lämpligheten i att låta regulatorn helt fritt få ställa sina parametrar kan diskuteras och för att ytterligare öka säkerheten kan en rad modifieringar av den adaptiva regulatorn göras.

Exempelvis kan man utföra identifieringen kontinuerligt men kräva att operatören ger tillstånd för att korrigera regulatorparametrarna. En nackdel med denna lösning är att den adaptiva systemet ej längre självt kan stabilisera processen vid stora förändringar.

En annan lösning är att använda den adaptiva regulatorn för att under kontrollerade former skapa ett regulatorbibliotek genom att göra identifiering vid olika driftförhållanden. Under regleringen kan sedan rätt regulatorparametrar laddas in i regulatorn beroende på driftförhållandena. Denna teknik kallas Gain-Scheduling och används ofta som alternativ till adaptiv reglering. Tyvärr fungerar denna metod inte i praktiken för detta problem, eftersom processen beror av alltför många faktorer. Förutom arbetspunkten påverkar också bla. matarvattentemperaturen och bränslets typ och kondition. Eftersom många faktorer påverkar blir modellbiblioteket stort och framtagningen av detta mycket tidsödande. Dessutom krävs trovärdiga mätningar av de storheter som påverkar processen.

Jag förespråkar den lösning som beskrivs i rapporten. Den ger god säkerhet eftersom :

- Identifieringen ger väntevärdesriktiga skattningar varför begynnande instabilitet som skapar excitation som gör att parameterskattningarna snabbt korrigeras och ger regulatorn goda möjligheter att stabilisera processen.
- Kopplingen av en störsignals amplitud till P-matrisen gör att variansen i parametrarna begränsas.
- Regulatorn som ligger till grund för det adaptiva systemet är robust och kan hantera dåliga skattningar av processen.

Att snabbt kunna stabilisera processen vid snabba processändringar är värdefullt eftersom det ger operatören tid att besluta om snabbstopp eller nedstyrning.

Implementation

Vid mina simuleringar har jag försummat HC-pumparnas dynamik. I äldre reaktorer är HC-pumparna mycket långsamma och kan ej användas för att kompensera resonanssvängningarna i härden. Dock kan ett system där den önskade bandbredden satts lågt användas för att se till att regulatören i varje fall inte försämrar stabiliteten. I vissa nyare reaktorer är pumparna av en snabbare typ vars dynamik bör kunna försummas, och är snabba nog för att dämpa resonanssvängningarna.

Vid en implementation av den adaptiva regulatören bör vissa modifieringar till parameterskattaren och regulatordesignen göras :

Identifieringsproblem är så gott som alltid illa konditionerade. Med SIMULINK uppträdde inga numeriska problem vid identifieringen, men dessa kan gör sig troligen påminna vid en implementation i enklare datorsystem. Genom att uppdatera kvadratrotten av P-matrisen kan algoritmen göras mera numeriskt robust. Kvadratrotsalgoritmen beskrivs t.ex. i [3]. Häftiga transienter kan tvinga ner P-matrisens element till noll och därmed låsa parameterskattningarna för alltid. Därför bör en undre gräns sättas på P-matrisens diagonalelement.

Med samplings tiden 0.1 s är beräkningskraften som behövs för den adaptiva regulatören måttlig, och en implementation bör vara möjlig på ABB Masters pascalkort. I appendix D finns ett FORTRAN-program som implementerar den adaptiva regulatören enligt kapitel 5.

A. IMPLEMENTERING I SIMULINK

Simuleringarna utfördes med hjälp av SIMULINK. I detta appendix beskrivs de simuleringsmodeller som tagits fram. För att minska beräkningbördan vid simuleringarna implementerades processmodellen i diskret tid.

Processmodell

Processmodellen implementerades med hjälp av standardblocken i SIMULINK.

Formen

$$y(k) = H_u(q) u(k) + H_e(q) e(k) = \frac{B(q)}{A(q)} u(k) + \frac{D(q)}{C(q)} e(k) \quad (\text{A.1})$$

är ej lämpad för direkt implementering. Om de instabila polerna till $H_u(q)$ och $H_e(q)$ inte sammanfaller exakt kommer vi att få svängningar i utsignalen som ej går att styra med regulator. Visserligen har brustermens instabila poler anpassats till processtermens men pga. av numeriska fel vid simuleringen kommer en direkt implementation av (A.1) inte att fungera. Modellen (A.1) kan dock skrivas om enligt :

$$y(k) = \frac{B(q)}{A_1(q)A_u(q)} u(k) + \frac{D(q)}{C_1(q)A_u(q)} e(k) \quad (\text{A.2})$$

där $A(q) = A_1(q)A_u(q)$ och $C(q) = C_1(q)A_u(q)$ där $A_u(q)$ innehåller den (nästan) instabila dynamiken i $A(q)$ och $C(q)$. (A.2) kan implementeras som

$$y(k) = \left(\frac{B(q)}{q^2 A_1(q)} u(k) + \frac{D(q)}{q^2 C_1(q)} e(k) \right) \frac{q^2}{A_u(q)} \quad (\text{A.3})$$

som har bättre numeriska egenskaper.

Följande matlabprogram definierar och skriver om de identifierade processmodellerna på formen (A.3)

```

sampletime = 0.1;

%Models sample interval 0.1 second
%P701 1205MW 2400kg/s
B701=[0 5.9655 12.1740 0.4230 -2.7622 -5.4797]*1.3;
A701=[1.0000 -1.5916 0.4773 -0.4994 1.2193 -0.5491];%P702
1205MW 2600kg/s
B702=[0 5.9635 13.1743 2.3299 -2.2350 -5.1708];
A702=[1.0000 -1.5578 0.4341 -0.4951 1.2172 -0.5389];
%P703 1205MW 2800kg/s
B703=[0 6.8365 14.7475 0.9698 -3.6934 -6.7033];
A703=[1.0000 -1.5769 0.4176 -0.4088 1.1797 -0.5593];
%P704 1205MW 3300kg/s
B704=[0 10.1445 14.4286 -0.7131 -6.5611 -5.6063];
A704=[1.0000 -1.5499 0.2521 -0.0106 0.7981 -0.4361];
%P7x1 1205MW 4000kg/s
B7x1=[0 11.0864 18.1505 -3.7691 -10.7035 -7.0464];
A7x1=[1.0000 -1.5242 0.1367 0.1539 0.7405 -0.4626];
%P7x2 1205MW 5000kg/s
B7x2=[0 14.3449 20.5024 -7.7805 -13.0921 -8.2647];
A7x2=[1.0000 -1.5142 0.1886 0.1755 0.5757 -0.3789];

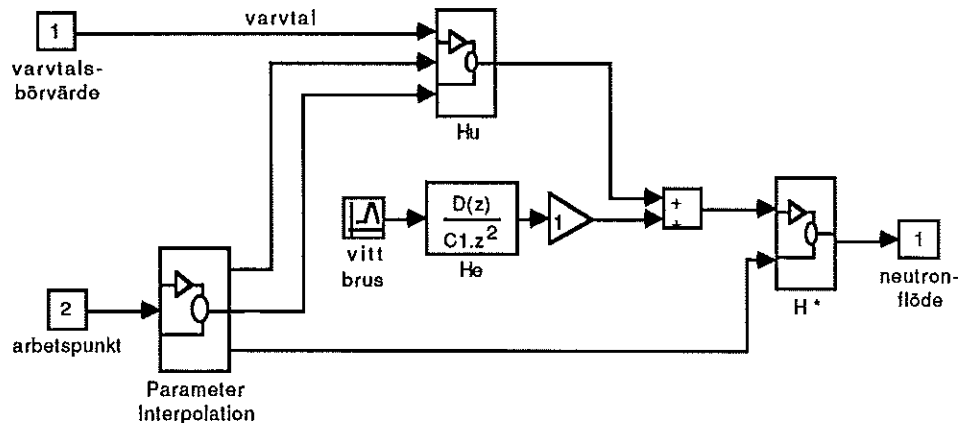
%Noise model sample frequency=12.5 Hz
Cs=[1.0000 -1.6788 1.1386 -0.7469 0.3788];
Ds=[1.0000 -0.3660 0.2893 0.0963]*0.1689;
% resample noisemodel with sampling time 0.1 s
[Dsc,Csc] = D2CM(Ds,Cs,1/12.5,'zoh');
[D,C] = sample(Dsc,Csc,0.1);

Coeffd = [2400 B701 A701; 2600 B702 A702; 2800 B703 A703;
3300 B704 A704; 4000 B7x1 A7x1; 5000 B7x2 A7x2];
% normalize models (input,output in percent)
Coeffd(:,2:7) = Coeffd(:,2:7)/100;

% modify noisemodel to fit process modell
% and rewrite according to (A.3)
noisepoles = roots(C)';
C1 = real(poly(noisepoles(3:4))); % C1
for i=1:6
    poles = roots(Coeffd(i,8:13))';
    Aunstab = poly(poles(3:4));
    A1 = poly([poles(1:2) poles(5)]);
    models(i,1) = Coeffd(i,1);
    models(i,2:4) = A1(2:4); % A1
    models(i,5:9) = Coeffd(i,3:7); % B
    models(i,10:11) = Aunstab(2:3); % A*
end

save reactorparameters C1 D models sampletime

```



Figur A.1 : Blockschema för processmodellen.

För att kunna använda processmodellen vid godtyckliga arbetspunkter interpoleras varje koefficient i polynomen A_1 , B och A_u . C_1 och D -polynomen är desamma för alla arbetspunkter. SIMULINKs **Lookup-table** block används för interpolationen.

Estimator

Estimatoren enligt (3.5) implementerades med hjälp av funktionsblocket **MATLAB-function**. Modellen som ansätts är en ARMAX-modell enligt (3.8, 3.9). Estimatorns tillstånd, P , ϕ , och θ samt glömskefaktorn λ och ordningstalen n_a , n_b och n_c deklarerades som globala variabler. Matlab funktionen nedan implementerar estimatormen :

```
function [out]=ERLS (in) ;

% estimates the parameters in the ARMAX model
%
% A(q) y(k) = B(q) u(k) + C(q) e(k)
%
% where degA = narls, degB = nbrls-1, degC = ncrls
% using extended recursive least squares with exponential
% forgetting
% To be used with SIMULINK's MATLAB-function block.
% in(1) = process input, in(2) = process output

phitPphi = phirls'*Prls*phirls;
y = in(2); u = in(1);

e = y - phirls'*thetarls;
K = Prls * phirls/(lambdarls + phitPphi);
Prls = (Irls - K*phirls')*Prls/lambdarls;
thetarls = thetarls + K*e;

ep = y - phirls'*thetarls;
phirls=[-y; phirls(1:narls-1);u;phirls(narls+1:narls+nbrls-1); ep; phirls(narls+nbrls+1:narls+nbrls+ncrls-1)];
out=[thetarls' Prls(1,1)+Prls(2,2)+Prls(3,3)+Prls(4,4) e];
```

Regulatordesign

Även regulatordesignen implementerades med hjälp av funktionsblocket **MATLAB-function**. A_o och A_m deklareraras globalt i MATLAB. Designekvationen (4.4) har lösts symboliskt och programmet sätter in siffervärden i lösningen.

```

function [out] = RSTDesign3(parameters);

a1 = parameters(1); a2 = parameters(2);
b1 = parameters(3); b2 = parameters(4);
ao1 = Ao(2); ao2 = Ao(3);
am1 = Am(2); am2 = Am(3); am3 = Am(4);

r1 = am1 + ao1 + 1 - a1;
t1 = ao2 + am1*ao1 + am2 + a1 - a2 - r1*(a1-1);
t2 = am1*ao2 + am2*ao1 + am3 + a2 + r1*(a1-a2);
t3 = am2*ao2 + am3*ao1 + r1*a2;
t4 = am3*ao2;
t5 = b1*a1 - b1*a2 - b1*b1*a2/b2;
t6 = t3 + t1*(a1-a2) - b1*(t4+a2*t1)/b2;
t7 = b2 + b1 - b1*a1;
t8 = t2 + t1*(1-a1);

s0 = (b2*t8 - b1*t6)/(b2*t7 - b1*t5);
s1 = (t6 - s0*t5)/b2;
s2 = (t4 + a2*t1 - s0*b1*a2)/b2;
r2 = t1 - b1*s0;

R = [1 r1-1 r2-r1 -r2];
S = [s0 s1 s2];
T = Ao*sum(Am)/(b1+b2);

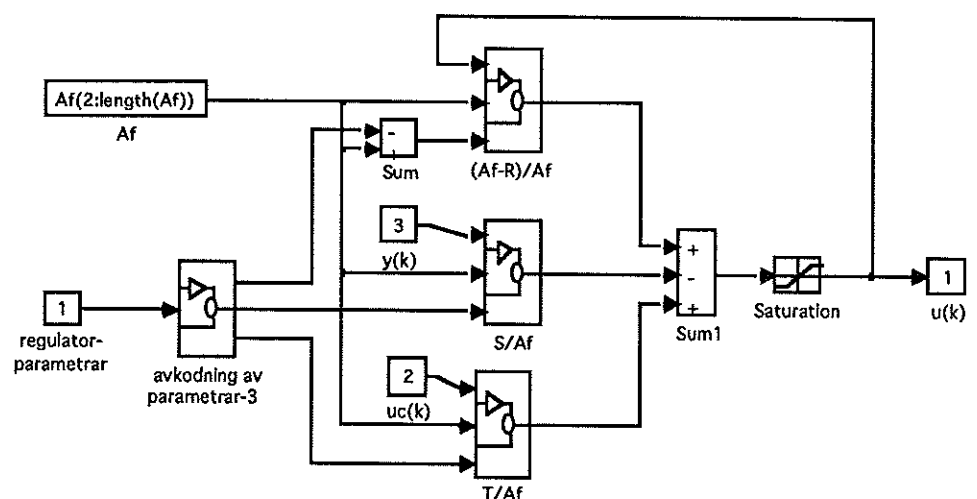
out = [R(2:4) 0 S 0 T]';

```

Vid en verklig implementering bör dessa beräkningar utökas med en kontroll av gemensamma faktorer i A & B-polynomen.

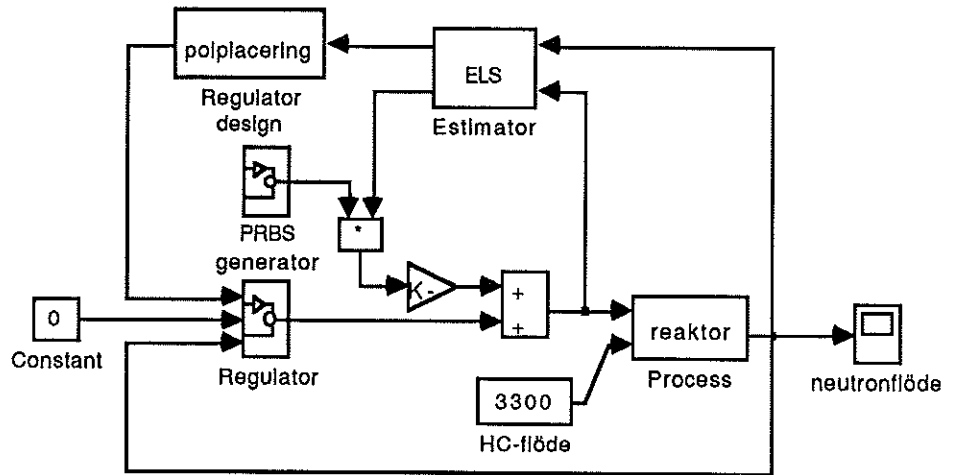
Regulator

Regulatorn realiserar styrlagen (4.13). Dess parametrar kommer in som insignaler från estimatorn. Blockschemata för regulatorn ses i figuren nedan :



Figur A.2 : Blockschemata för regulatorn.

Sammankoppling



Figur A.3: Blockschema för slutna systemet.

B. IMPLEMENTATION I FORTRAN

För att verifiera mina resultat i implementerades den adaptiva regulatorn i ett FORTRAN-program för simulering med BISON. BISON är en simulator som bygger på fysikaliska modeller av kraftverket. I princip all dynamik simuleras - totalt flera hundra differentialekvationer.

Tyvärr fick jag problem med kommunikationen med BISON och någon verifiering kunde inte göras inom tiden för exjobbet. Programmet är dock testat med en enklare reaktormodell och bör vara korrekt skrivet.

```

FUNCTION ScalarP (V1,V2,n)
C multiplication of vector V1 and V2 ant returns the
C scalar V1'*V2
DIMENSION V1 (n) ,V2 (n)
INTEGER i
  r = 0.
  DO i = 1, n
    r = r + V1(i)*V2(i)
  ENDDO
  ScalarP = r
  RETURN
END

SUBROUTINE MatVecP (R,DM,V,n)
C multiplication of matrix M(n,n) and the vector V(n,1)
C the result is stored in the vector R(n,1)
DIMENSION R(n) ,DM(n,n) ,V(n)
INTEGER i,j
  DO i = 1, n
    R(i) = 0.
    DO j = 1, n
      R(i) = R(i) + V(j)*DM(i,j)
    ENDDO
  ENDDO
END

SUBROUTINE VecVecP (R,V1,V2,n)
C multiplication of (n,1) vector and the (1,n) vector V2.
C The result is stored in the (n,n) matrix R = V1*V2
DIMENSION R(n,n) ,V1 (n) ,V2 (n)
INTEGER i,j
  DO i = 1, n

```

```

        DO j = 1, n
          R(i,j) = V1(i)*V2(j)
        ENDDO
      ENDDO
    END

      SUBROUTINE ELS(u,y,theta,P,time)
C
C   estimates the parameters in the ARMAX model
C
C    $A(q) y(k) = B(q) u(k) + C(q) e(k)$ 
C
C   where degA = na, degB = nb-1, degC = n
C   using extended recursive least squares with
exponential forgetting

      PARAMETER na = 2, nb = 2, nc = 0, n = na+nb+nc
      PARAMETER lambda = 0.995
      DIMENSION P(n,n), theta(n)
      REAL du,dy,yold,uold
      REAL K(n), phi(n), phitPphi, w(n), wwt(n,n)
      SAVE phi,yold,uold

      du=u
      dy = y

      IF (time.GT.15.) THEN
        du = u -uold
        dy = y - yold
      ENDIF
      e = dy - ScalarP(phi,theta,n)
      CALL MatVecP(w,P,phi,n)
      den = lambda + ScalarP(phi,w,n)
      DO i = 1, n
        theta(i) = theta(i) + w(i)/den*e
      ENDDO
      CALL VecVecP(wwt,w,w,n)
      DO i = 1, n
        DO j = 1, n
          P(i,j) = (P(i,j) - wwt(i,j)/den)/lambda
        ENDDO
      ENDDO

C   update phi-vector
      i = na-1
1     IF (i.GE.1) THEN
        phi(i+1) = phi(i)
        i = i - 1
        GOTO 1
      ENDIF
      phi(1) = -dy
      i = na+nb-1
2     IF (i.GE.na+1) THEN
        phi(i+1) = phi(i)
        i = i - 1
        GOTO 2
      ENDIF
      phi(na+1) = du
      yold = y
      uold = u
    END

      SUBROUTINE RSTDesign3(parameters, regpar)

```

```

DIMENSION parameters(4), regpar(12)
REAL a1,a2,b1,b2,ao1,ao2,am1,am2,am3
REAL t1,t2,t3,t4,t5,t6,t7,t8,s0,s1,t0,r2

a1 = parameters(1)
a2 = parameters(2)
b1 = parameters(3)
b2 = parameters(4)

C data for omega = 3, zeta = 0.7, alpha = 4
DATA ao1,ao2,am1,am2,am3/-0.5653,0.1864,-
2.3249,1.8306,-0.4868/

r1 = am1 + ao1 + 1 - a1
t1 = ao2 + am1*ao1 + am2 + a1 - a2 - r1*(a1-1)
t2 = am1*ao2 + am2*ao1 + am3 + a2 + r1*(a1-a2)
t3 = am2*ao2 + am3*ao1 + r1*a2
t4 = am3*ao2
t5 = b1*a1 - b1*a2 - b1*b1*a2/b2
t6 = t3 + t1*(a1-a2) - b1*(t4+a2*t1)/b2
t7 = b2 + b1 - b1*a1
t8 = t2 + t1*(1-a1)

s0 = (b2*t8 - b1*t6)/(b2*t7 - b1*t5)
s1 = (t6 - s0*t5)/b2
s2 = (t4 + a2*t1 - s0*b1*a2)/b2
r2 = t1 - b1*s0
t0 = (1+am1+am2+am3)/(b1+b2)

regpar(1) = 1
regpar(2) = r1-1
regpar(3) = r2-r1
regpar(4) = -r2
regpar(5) = 0
regpar(6) = s0
regpar(7) = s1
regpar(8) = s2
regpar(9) = 0
regpar(10) = t0
regpar(11) = t0*ao1
regpar(12) = t0*ao2

END

SUBROUTINE RST(u,y,uc,regpar,states)
C Implements the control law :
C Ru= -Sy + Tuc according to CCS, chapter 10
C
C u = control signal
C y = process output
C uc = reference value
C ul,uh = limits of control signal
C regpar = regulator parameters([1 r1..rdeg+1 s0..
sdeg t0..tdeg])
C deg = order of regulator
C states = regulator states as :
C ([-u(k)..-u(k-deg) -y(k)..-y(k-
deg) uc(k)..uc(k-deg)])

C parameters
PARAMETER ul = -100.0, uh = 100.0
PARAMETER deg = 3

```

```

PARAMETER tstep = 0.1

C Arguments ...
C REAL u, y, uc
C DIMENSION regpar((deg+1)*3),
states((deg+1)*3)

C local variables
REAL initialpars((deg+1)*3)
REAL rsum, ssum, tsum
INTEGER i, soff, toff

C update statevector first ...
i = deg
1 IF (i.GE.1) THEN
states(i+1) = states(i)
states(i+deg+2) = states(i+deg+1)
states(i+deg+deg+3) = states(i+deg+deg+2)
i = i - 1
GOTO 1
ENDIF
states(deg+2) = -y
states(deg+deg+3) = uc

C calculate control signal
u = 0.
DO i = 2, (deg+1)*3
u = u + regpar(i) * states(i)
ENDDO
states(1) = -u
C saturation of control signal
IF (u.GT.uh) THEN
u = uh
ELSEIF (u.LT.ul) THEN
u = ul
ENDIF
END

SUBROUTINE PRBS(u)
PARAMETER limit = 0.5
LOGICAL x, q
SAVE q
x = (rand()).GE.limit)
q = ((q.AND..NOT.x).OR.(x.AND..NOT.q))
IF (q.EQ..TRUE.) THEN
u = 1.
ELSE
u = -1.
ENDIF
END

SUBROUTINE STR(ACTION, LOG, time, u, y, uc)
C ACTION = type of call, 'REGULATE' or 'STEADY STATE'
C time = time of call
C u = control signal
C y = process output
C uc = reference value

C initialize process parameters for 3300 kg/s, 70 %
CHARACTER*(*) ACTION, LOG
C REAL time, u, y, uc
REAL states(12), regpar(12), theta(4), theta0(4)

```

```

REAL P(4,4),PRBSamp,partialtrace
C *****Parameters for a PID*****
REAL RI,RD

SAVE RI,RD,Ryold

PARAMETER RK = 0.8, RTi = 20.0, RTd = 0.6, RN = 20
C *****
SAVE nextupdate,theta,P,states,regpar
DATA theta0/-1.8815,0.9485,0.1031,0.1121/
C DATA theta0/-1.3,0.49,-0.05,0.1,0.0263,-0.2177/
PARAMETER tstep = 0.1
PARAMETER P0 = 0.3, k = 9.0, degc = 3
PARAMETER storfile = 50, sigfile = 51, thetafile = 52
REAL rsum,ssum,tsum
C *****
C bi = RK*tstep/RTi
C bd = RK*RN*RTd/(RTd+RN*tstep)
C ad = RTd/(RTd + RN*tstep)
C *****
C uc=0.7

C SUBROUTINE STR(...)
uc = uc * 100
y = y * 100
u = u * 100

C write(*,*) ACTION
C IF (ACTION.EQ.'REGULATE') THEN
C write(*,*) time,nextupdate*tstep
C IF (time.GE.(nextupdate*tstep)) THEN
C*****
C ddy=y-uc
CALL RST(u,ddy,0,regpar,states)
C CALL RST(u,y,uc,regpar,states)
C *****
C RD = ad*RD - bd*(y-yold)
C RP = RK*(uc-y)
C u = RP + RI + RD
C RI = RI + bi*(uc-y)
C yold = y
C *****
C partialtrace =
abs(P(1,1)+P(2,2)+P(3,3)+P(4,4))
PRBSamp = min(1.0,partialtrace*k)
CALL PRBS(ud)
u = u + ud*PRBSamp
write(*,*) u
CALL ELS(u,ddy,theta,P,time)
IF (time.GT.10.0) THEN
CALL RSTDesign3(theta, regpar)
ELSE
CALL RSTDesign3(theta0, regpar)
ENDIF
nextupdate = nextupdate + 1
C log regulator signals
C*****
u=u+71.0
IF (time.GT.10.0) THEN
u=max(u,-0.0)
u=min(u,100.0)
ELSE
u=max(u,50)

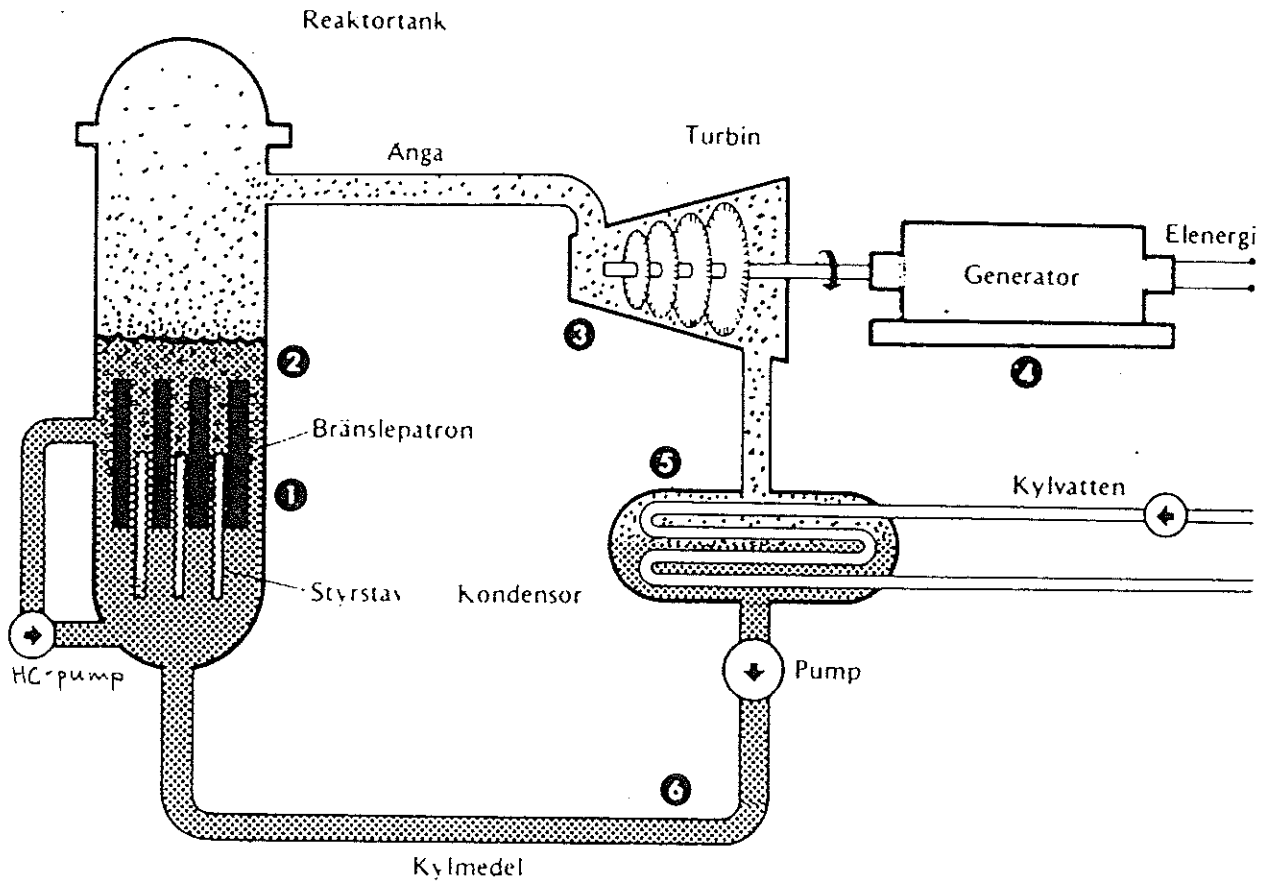
```

```

        u=min(u,80)
        ENDIF
C*****
        IF (LOG.EQ.'LOGGER ON') THEN
            WRITE(sigfile,*)(time,u, y, uc)
            WRITE(thetofile,*)(theta(i), i=1,4)
            WRITE(storfile,*)(u, PRBSamp)
        ENDIF
        ENDIF
        ELSEIF (ACTION.EQ.'STEADY STATE') THEN
C
            initialize regulator
            nextupdate = (time/tstep) + 1
            DO i = 1,6
                theta(i) = theta0(i)
            ENDDO
            CALL RSTDesign3(theta0, regpar)
            rsum = 0.0
            ssum = 0.0
            tsum = 0.0
            DO i = 1, degc+1
                rsum = regpar(i) + rsum
                ssum = regpar(i+degc+1) + ssum
                tsum = regpar(i+degc+degc+2) + tsum
            ENDDO
            y = rsum*u/(tsum-ssum)
            y=70.0
C
            uc=y
            uc =70.0
            DO i = 1, degc+1
C*****
                states(i) = -0
                states(i+degc) = -0
                states(i+degc+degc) = 0
C*****
            ENDDO
            DO i = 1,4
                DO j = 1,4
                    IF (i.EQ.j) THEN
                        P(i,j) = P0
                    ELSE
                        P(i,j) = 0
                    ENDIF
                ENDDO
            ENDDO
            IF (LOG.EQ.'LOGGER ON') THEN
                WRITE(sigfile,*)(time,u, y, uc)
C*****
C
                yold = uc
C
                DI = u
C
                y = uc
C*****
                WRITE(thetofile,*)(theta(i), i=1,4)
                WRITE(storfile,*)(u, PRBSamp)
            ENDIF
        ELSE
            WRITE(*,*)'STR called with illegal action :'
            WRITE(*,*)'ACTION = ',ACTION
        ENDIF
        uc = uc / 100
        y = y / 100
        u = u / 100
    END

```

C. ÖVERSIKTSBILD PÅ BWR



Figur C.1 : Översiktsbild på en BWR.

D. REFERENSER

- [1] K.J. Åström och B. Wittenmark - Computer Controlled Systems, Prentice-Hall (1990)
- [2] K.J. Åström och B. Wittenmark - Adaptive Control, Addison-Wesley (1989)
- [3] L. Ljung - System Identification, theory for the user - Prentice-Hall (1987)

E. TACK

Under arbetets gång har jag fått värdefull hjälp från många håll. Speciellt vill jag tacka mina mina handledare Bohao Liao vid ABB Atom och Tore Hägglund vid Reglerteknik, LTH. Bohao Liao har t.ex. gjort merparten av arbetet med verifiering i BISON.