

CODEN: LUTFD2/(TFRT-5446)/1-62/(1991)

Digital reglering av en AC/DC omriktare

Istephan Tamras

Institutionen för Reglerteknik
Lunds Tekniska Högskola
December 1991

Department of Automatic Control Lund Institute of Technology P.O. Box 118 S-221 00 Lund Sweden		Document name Master Thesis	
		Date of issue December 1991	
		Document Number CODEN: LUTFD2/(TFRT-5446)/1-62/(1991)	
Author(s) Istephan Tamras		Supervisor Nils Bäckman and Rolf Johansson	
		Sponsoring organisation	
Title and subtitle Digital reglering av en AC/DC-omriktare. (Digital Control of an AC/DC-converter).			
Abstract <p>A microprocessor-based digital control equipment for a power supply is considered in this master thesis. The goal is achieved by means of system identification and digital control design.</p>			
Key words			
Classification system and/or index terms (if any)			
Supplementary bibliographical information			
ISSN and key title			ISBN
Language Swedish	Number of pages 62	Recipient's notes	
Security classification			

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.

Innehållsförteckning

Förord	2
1. Inledning	3
1.1 Bakgrund	3
1.2 Rapportens uppläggning	3
2. Processbeskrivning	4
2.1 Allmänt om processen	4
2.2 Buck- boost (Flyback)	4
3. Processidentifiering	7
3.1 Processdelning	7
3.2 Experiment	10
3.3 Identifiering av likriktarblocket	11
3.4 Den slutliga modellen	16
3.5 Modellvalidering	19
4. Reglering	20
4.1 Kravspecifikation	20
4.2 Linjär tidsdiskret regulator	20
4.3 Tidsdiskret PID-regulator	27
4.4 Reglering av likriktaren	35
5. Sammanfattning	37
Referenser	38
Bilagor	39

Förord

Denna rapport avser ett examensarbete som gjordes på Ericsson Components AB, Power Division, i Stockholm i samarbete med institutionen för reglerteknik vid Lunds tekniska högskola sommaren 1991. Arbetet har stor del inneburit identifiering och digital reglering av en AC/DC omvandlare som användes för att strömförsörja ett radiostasstativ.

Jag vill här passa på att framföra ett tack till mina handledare, Nils Bäckman på Ericsson och högskolelektor Rolf Johansson på reglerteknik för deras aktiva medverkan i arbetet.

Jag vill också tacka Hans-Eric Aaltonen för hans hjälp angående assembler och regulatorimplementering. Vidare vill jag tacka Ulf Tollaksen, Rolf Jonsson och andra personer på Ericsson för deras hjälp.

1. Inledning

1.1 Bakgrund

I denna rapport beskrivs identifiering och digital reglering av en AC/DC omvandlare som användes för att strömförsörja ett radiostasstativ. Identifiering innebär att man sätter en matematisk modell som beskriver processens beteende. Den matematiska modellen kan sedan användas för att dimensionera en regulator som reglerar processen. I denna rapport har polplaceringsmetoden som bygger på kännedom av processmodellen behandlats. Dessvärre kunde vi inte implementera den, eftersom den krävde fler variabler och det fanns begränsad tillgång av minnet i processorn (microcontroller 80C196 Intel) som samtidigt sköter andra uppgifter. Dessutom hade vi ont om tid. Däremot kunde vi realisera en digital PID-regulator, den klarade mycket bra kravspecifikationerna.

1.2 Rapportens uppläggning

I kapitel 2 ska vi kortfattat beskriva processen och den grundkoppling som likriktarens funktion bygger på.

I kapitel 3 ska vi identifiera processen, dvs finna en matematisk modell till processen. I 3.1 ska vi dela processen i ett antal block. Överföringsfunktionen till enkla block beräknas analytiskt. I 3.2 beskriver vi experimentet med vars hjälp vi kan beräkna överföringsfunktionen till huvudblocket (likriktaren). I 3.4 gör vi en visuell identifiering av likriktarblocket samt en identifiering enligt minsta kvadratmetoden. I 3.4 sätts alla block ihop för att få den slutliga modellen. I 3.5 gör vi en modellvalidering.

I kapitel 4 beskriver vi kravspecifikationerna i 4.1. Därefter i 4.2 beskriver vi teorin för polplacering samt dimensionerar en tidsdiskret linjär regulator enligt den metoden. I 4.3 ger vi en teoretisk bakgrund för PID-regulatorer. I 4.4 realiserar vi en digital PID-regulator för likriktaren.

2. Processbeskrivning

I detta kapitel ska vi kortfattat beskriva processen. I 2.1 ska vi allmänt beskriva processen. Därefter i 2.2 ska vi beskriva den grundkoppling som likriktarens funktion bygger på.

2.1 Allmänt om processen

Likriktaren användes för att strömförsörja ett radiostasativ. Den omvandlar inkommande nätspänning, 230 VAC, till en reglerad +24 VDC. Likriktaren kan kopplas parallellt i ett obegränsat antal och dela lasten. Likriktarens funktion bygger på en grundkoppling som kallas för Buck-boost. Ibland kallad för flyback.

2.2 Buck- boost (Flyback)

Buck boost är en DC/DC omvandlare. Kopplingsschema över den visas i figur 2.1 .

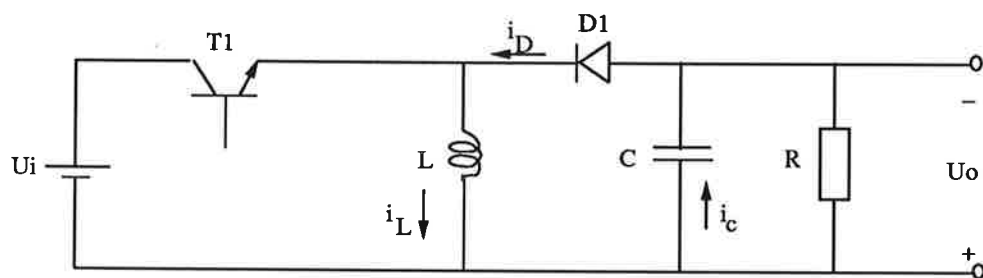


Figure 2.1 Grundkoppling Buck-boost (Flyback) DC/DC omvandlare.

Transistorn T1 fungerar som switch. Transistorn styrs med ett pulståg med frekvens $f=1/T_s$ och ledintervall D. Flyback har alltså två tillstånd. Det första är när transistorn T1 leder och det andra är när den icke leder. Dessa två tillstånd visas i figur 2.2.

När T1 sluts börjar strömmen genom induktansen att stiga med lutning

$$\frac{di}{dt} = \frac{U_i}{L}$$

och när T1 öppnas sjunker strömmen med lutning

$$\frac{di}{dt} = \frac{-U_o}{L}$$

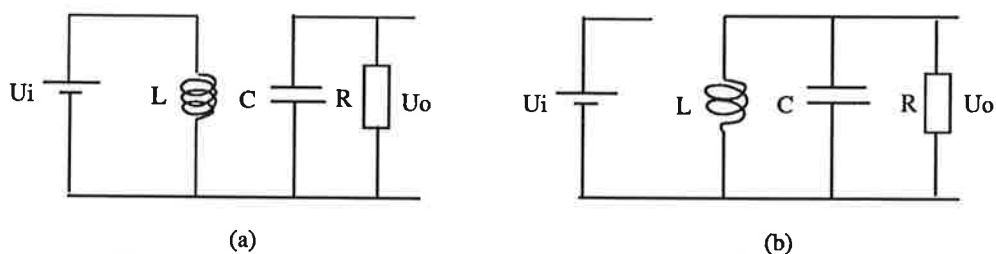


Figure 2.2 Flybacks två tillstånd.(a) Sluten switch.(b) Öppen switch.

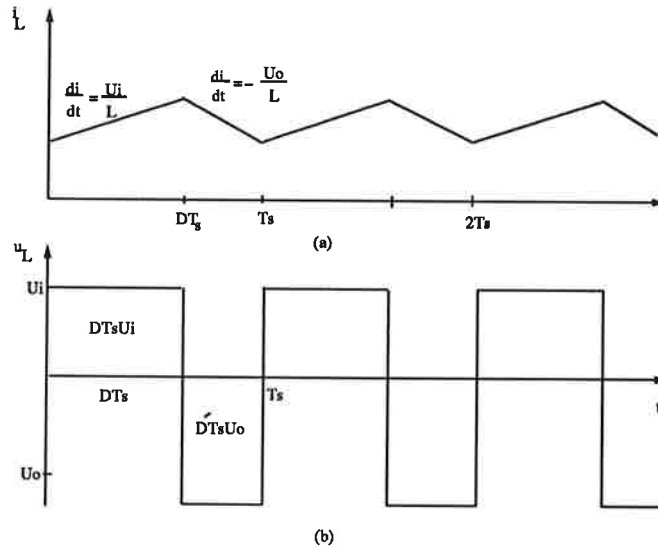


Figure 2.3 (a) Strömmen genom induktansen. (b) Spänningen över induktansen.

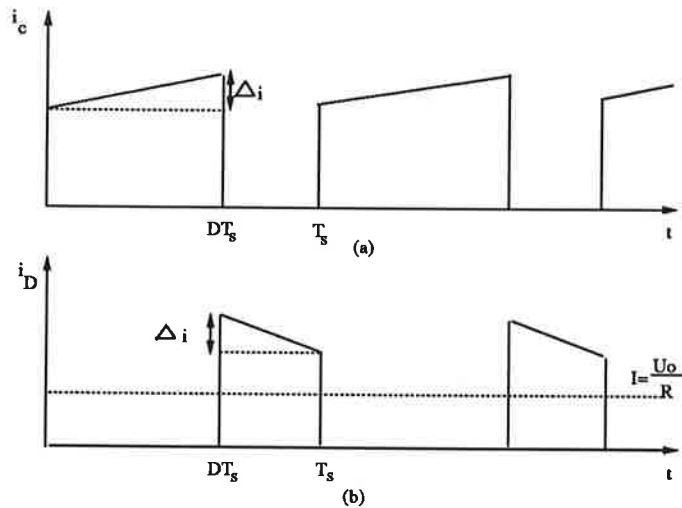


Figure 2.4 (a) Strömmen genom kondensatorn. (b) Strömmen genom dioden.

som framgår av figur 2.3 .

I figur 2.3(b) är den positiva spänningstidytan $DT_s U_i$ lika med den negativa spänningstidytan $(1 - D)T_s U_o$

$$DT_s U_i = -(1 - D)T_s U_o$$

$$U_o = -\frac{D}{1 - D} U_i = -\frac{D}{D'} U_i$$

$$\Delta i = \frac{U_i}{L} DT_s = \frac{U_o}{L} D' T_s$$

Utspänningen bestäms alltså av pulsbredden D . Som vi kan se i figur 2.4 så finns det stort rippel på både in- och utströmmar. Utspänningen kan styras från 0 till $-\infty$. Utspänningen är negativ på grund av att drosseln är fixerad mot jord.

Flyback har verkningsgrad

$$\eta = \frac{1}{1 + \frac{R}{R_i} \frac{1}{D'}}$$

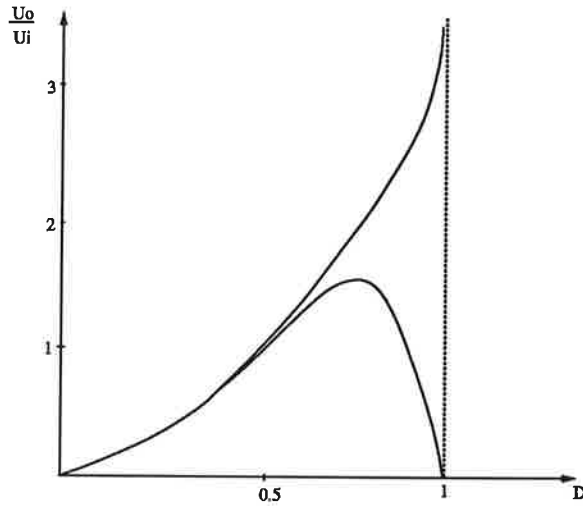


Figure 2.5 Flybackspänningsförstärkning.

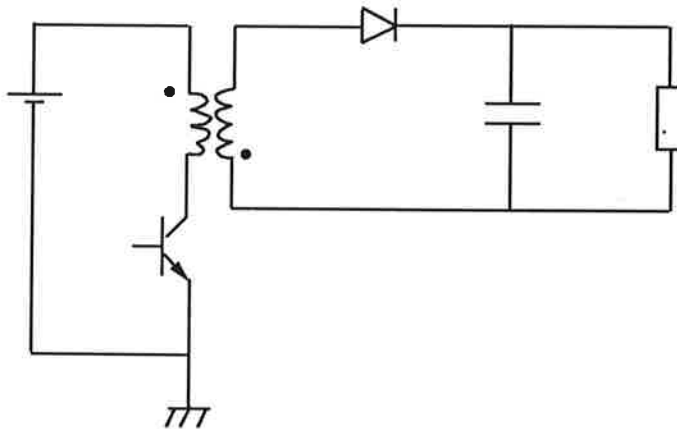


Figure 2.6 Isolerad variant av flyback.

Flybackspänningsförstärkning blir

$$\frac{U_o}{U_i} = \frac{D}{D'\eta}$$

I den likriktare som vi ska behandla användes en isolerad variant av flyback, se figur 2.6. Spänningsförstärkning kan teoretiskt gå mot oändligheten, men begränsas av R_l som är drosselns inre resistans. Flyback är mycket vanlig och det är billigare att tillverka en isolerad variant. Så gott som alla omvandlare under 100 watt använder flyback.

3. Processidentifiering

Processidentifiering betyder att man finner en matematisk modell, som beskriver processens beteende så bra som möjligt. Med matematisk modell menar vi här en överföringsfunktion som beskriver sambandet mellan insignalen och utsignalen till processen. Det finns två olika tillvägagångssätt att göra identifieringen. Man kan antingen göra mätningar och analys i frekvensplanet eller i tidsplanet. Här har vi valt att göra det i frekvensplanet. Eftersom denna process är ganska stor, ska vi dela den i ett antal block och identifiera dessa var för sig och detta gör vi i 3.1. Därefter i 3.2 ska vi beskriva hur själva experimentet och mätningarna gick till. I 3.3 beskriver vi identifiering av likriktarblocket. Först gör vi en visuell identifiering. Därefter gör vi identifieringen enligt minsta kvadrat metoden. I 3.4 sätter vi ihop alla block för att få den slutliga modellen.

3.1 Processdelning

Eftersom den här processen är ganska komplicerad, är det lämpligt att dela den i olika block. Överföringsfunktionen till varje block beräknas antingen analytiskt eller experimentellt. Sedan sätts de ihop för att få hela processens överföringsfunktion. Denna process kan delas i fyra block som framgår av figur 3.1. Nedan följer en beskrivning av de olika blocken.

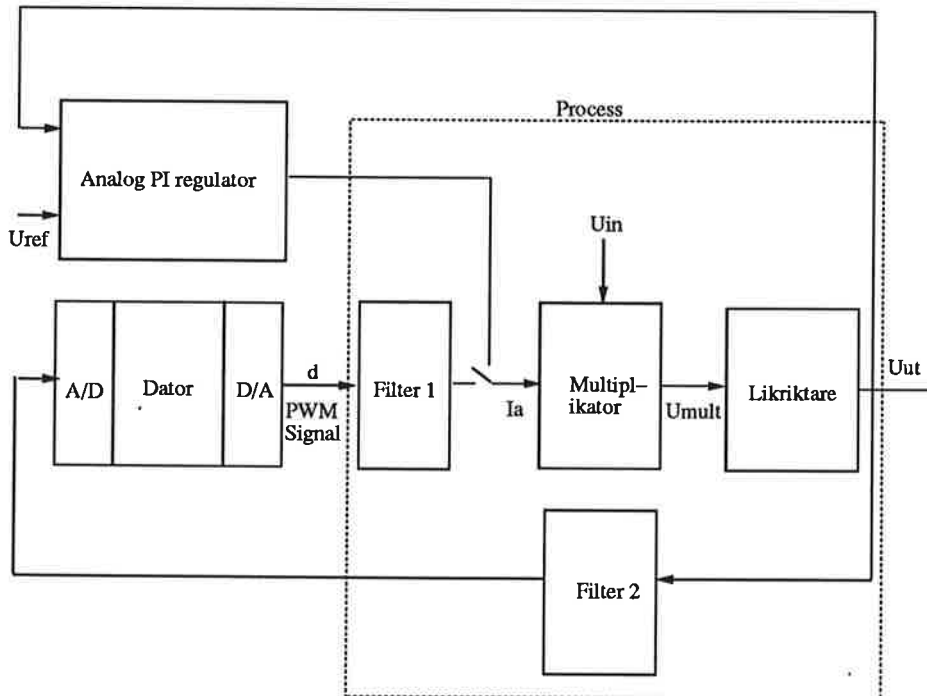


Figure 3.1 Blockschemat över processen.

Sätts de ihop så får man hela filtrets överföringsfunktion

$$\frac{U_o(s)}{U_i(s)} = \frac{R_2}{(1 + R_1CS)^2(R_1 + R_2 + R_1R_2CS)}$$

Där $R_1 = 30K\Omega$, $R_2 = 2M\Omega$, $C=10$ nF. Filtret har alltså tre poler. En dubbel pol i

$$s = \frac{-1}{R_1C} = -3333$$

och en pol i

$$s = \frac{-(R_1 + R_2)}{R_1R_2C} = -3383$$

Filtret åstadkommer alltså en medelvärdebildning av PWM signalen. Utifrån filtret får vi en likspänning. Dess nivå är beroende av ledintervallet d . Denna nivå omvandlas sedan till en ström I_a

$$I_a = \frac{d \cdot 12 - 1.1}{95.3 \cdot 10^3} \text{ A} \quad (3.1)$$

3.1.2 Multiplikatorn

Multiplikatorn har två insignaler U_{in} som är en helriktad nätspänning och varierar mellan 0 – 4.5 V och strömmen I_a från filter 1. Multiplikatorns utsignal är U_{mult} . Multiplikatorns funktion i processen är

$$U_{mult} = R \frac{I_a \cdot I_b}{I_c} + 1.1$$

där

$$I_a = \frac{U_{in} - 1.1}{68.1 \cdot 10^3} + \frac{5 - 1.1}{241.5 \cdot 10^3}$$

$$I_c = \frac{U_{in} - 1.1}{44.2 \cdot 10^3}$$

När $U_{in} = 4.5V$ (maxvärdet) får följande förhållande mellan d och U_{mult}

$$U_{mult} = 2.9d + 0.833$$

Multiplikatorn ger alltså en förstärkning på 2.9 om $U_{in}=4.5$ V.

3.1.3 Filter 2

Detta filter är en RC länk ($R = 1K\Omega$, $C = 2.2\mu F$), som filtrerar utsignalen från höga frekvenser speciellt 75 KHz (huvud FET-transistorns switch-frekvens) innan utsignalens värde skickas till A/D omvandlaren. Detta filter har en tidskonstant på 2,2 ms och brytfrekvens vid 455 Hz .

3.1.4 Likriktarblocket

Detta block har U_{mult} som insignal och U_{ut} som utsignal och består av tre likriktarmoduler som framgår av figur 3.4. Dessa tre moduler är parallellkopplade och levererar tillsammans en uteffekt på 1200 watt.

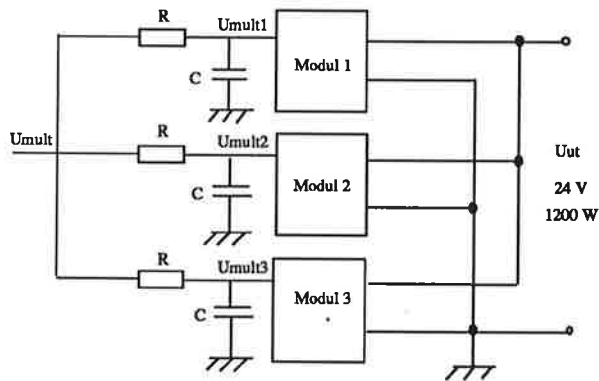


Figure 3.4 likriktarblocket.

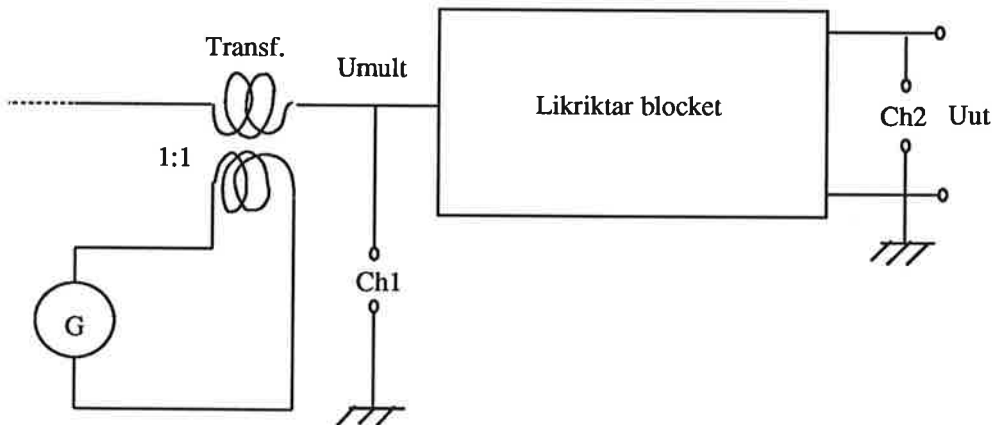


Figure 3.5 Koppling av frekvensanalysatorns tre kanaler till processen.

3.2 Experiment

För likriktarblocket är det ganska svårt att analytiskt räkna fram en matematisk modell. Därför ska vi försöka göra en frekvensanalys m.h.a frekvensanalysatorn Schumberger Solartron. Eftersom likriktaren inte fungerar utan regulatören, var vi tvungna att göra mätningen i slutna slinga. Spänningen U_{mult} ser vid normaldrift ut som U_{in} plus en likspänningsnivå. Den har 100 Hz som grundton och andra övertoner. För att slippa dessa frekvenser matade vi kretsen med likspänning med hjälp av en vanlig likriktarbrygga vilket gjorde att vi fick U_{mult} till en helriktad likspänning. Frekvensanalysatorn har tre kanaler (kanal 1, kanal 2 och generator kanal G). För att injicera sinussignalerna i likriktaren i slutna slinga kopplade vi en transformator som hade omsättnings 1:1 i kretsen. Generatorkanalen kopplades till primärsidan, kanal 1 kopplades till U_{mult} och kanal 2 kopplades till utgången U_{ut} , se figur 3.5.

Eftersom multiplikatorn har två insignaler som påverkar U_{mult} tog vi Bodediagram vid olika spänningsnivåer på U_{in} genom att ändra matningsspänningen till likriktarbrygga. Vi svepte frekvensen upp till 10 KHz och fann mest intressant dynamik i intervallet 0-400 Hz. Vi tog Bodediagram vid följande matningsspänningar

- Vid 242 V, 10-400 Hz, se bilaga 1.1
- Vid 220 V, 10-400 Hz, se bilaga 1.2
- Vid 200 V, 10-400 Hz, se bilaga 1.3
- Vid 160 V, 10-400 Hz, se bilaga 1.4
- Vid 220 V, 10-1000 Hz, se bilaga 1.5

3.3 Identifiering av likriktarblocket

I detta avsnitt ska vi först göra en visuell identifiering av likriktarblocket från Bodediagrammen vi fick från experimentet. Sedan beskriver vi minsta kvadratmetoden och gör identifiering enligt denna metod.

3.3.1 Visuell identifiering

Genom att titta på alla dessa Bodediagram kan man lätt konstatera att alla ser likadana ut förutom lågfrekvensförstärkning som varierar med U_{in} . Man kan se att likriktaren har en integrator eftersom lågfrekvensförstärkning sjunker med 20 dB/dekad och faskurvan går mot 90° . I närheten av 60 Hz sjunker amplitudkurvan med en lutning på 40 dB/dekad vilket tyder på en dubbel pol. Vid 100 Hz finns det ett dubbelt nollställe eftersom det bryts upp d.v.s +40 dB/dekad samt faskurvan går tillbaka till -90° .

3.3.2 Minsta kvadrat anpassning

En överföringsfunktion sägs vara stabil om den har alla sina poler i vänstra halva planet (VHP). Den sägs vara rationell av ordning (m,n) om den kan skrivas på formen

$$G(s) = \frac{(b_1 s^m + b_2 s^{m-1} + \dots + b_m)}{(s^n + a_1 s^{n-1} + \dots + a_n)} \quad (3.2)$$

m brukar vara mindre än n . Låt $G(s)$ vara en stabil överföringsfunktion. Vi vill finna en bra approximation \hat{G} till $G(s)$. Det finns olika mått på hur bra en approximation är. En vanlig typ av distansfunktion är den metriska d

$$d(G, \hat{G}) = \|G - \hat{G}\| \quad (3.3)$$

Alla approximationer som uppfyller interpolationskrav

$$\hat{G}(i\omega_k) = G(i\omega_k), \quad k = 1, \dots, N \quad (3.4)$$

är de för vilka

$$\|G(i\omega_k) - \hat{G}(i\omega_k)\| = 0 \quad (3.5)$$

Att minimera felet $G - \hat{G}$ på detta sätt motsvarar minsta kvadrat anpassning av parametrarna θ

$$\hat{\theta} = (\hat{a}_1, \hat{a}_2, \dots, \hat{a}_n, \hat{b}_1, \hat{b}_2, \dots, \hat{b}_n)^T \quad (3.6)$$

i en rationell approximation

$$\hat{G}(s) = \frac{\hat{B}(s)}{\hat{A}(s)} = \frac{\hat{b}_1 s^{n-1} + \hat{b}_2 s^{n-2} + \dots + \hat{b}_n}{s^n + \hat{a}_1 s^{n-1} + \dots + \hat{a}_n} \quad (3.7)$$

En fördel med minsta kvadratanpassning är att räknearbetet är oberoende av vilken typ funktionen G är (rationell, icke rationell) eftersom bara funktionens värde vid N punkter behöver vara känt. Målet är att optimera felet

$$\min_{\theta} \sum_{k=1}^N |G(i\omega_k) - \hat{G}(i\omega_k)|^2 \quad (3.8)$$

Problemet med att optimera 3.8 är att de inblandade uttrycken inte är linjära i parametrar. Då kan den inte lösas med vanlig minsta kvadratmetoden. Levy metoden löste delvis denna svårighet genom att istället minimera felfunktionen

$$\min_{\theta} \sum_{k=1}^N |\hat{A}(i\omega_k)G(i\omega_k) - \hat{B}(i\omega_k)|^2 \quad (3.9)$$

Denna felfunktion kan formuleras som ett standard minsta kvadrat problem genom att definiera vektorn

$$Y = \begin{pmatrix} (i\omega_1)^n G(i\omega_1) \\ (i\omega_2)^n G(i\omega_2) \\ \vdots \\ (i\omega_N)^n G(i\omega_N) \end{pmatrix} \quad (3.10)$$

och regressionsvektorn

$$\Phi = \begin{pmatrix} -(i\omega_1)^{n-1} G(i\omega_1) & \dots & -G(i\omega_1)(i\omega_1)^{n-1} & \dots & i\omega_1 & 1 \\ -(i\omega_2)^{n-1} G(i\omega_2) & \dots & -G(i\omega_2)(i\omega_2)^{n-1} & \dots & i\omega_2 & 1 \\ \vdots & \dots & \vdots & \dots & \vdots & \vdots \\ -(i\omega_N)^{n-1} G(i\omega_N) & \dots & -G(i\omega_N)(i\omega_N)^{n-1} & \dots & i\omega_N & 1 \end{pmatrix} \quad (3.11)$$

och parametrarvektorn

$$\theta = \left(a_1 \quad \dots \quad a_n \quad b_1 \quad \dots \quad b_n \right)^T \quad (3.12)$$

Minsta kvadratlösningen blir

$$\hat{\theta} = (\Phi^* \Phi)^{-1} \Phi^* Y \quad (3.13)$$

där Φ^* är transponat och komplexkonjugat av Φ .

Vid experimentet gjorde vi flera mätningar vid olika arbetspunkter. Vi gör identifieringen vid arbetspunkten 220 V eftersom U_{in} har sitt maximum vid denna spänning samt att vi fick högre lågfrekvens förstärkning än vid mindre U_{in} . Programmet Matlab användes för att göra identifieringen med minsta kvadratanpassning. Från frekvensanalysatorn fick vi tre datakolonner i 66 punkter(frekvens i Hz, amplitud $|G(i\omega)|$ i dB och fas $|\Phi(i\omega)|$ i grader). För att kunna använda Matlabsrutiner krävs det att vi omvandlar de data vi fått från frekvensanalysatorn till en matris (Frekvensrespons FR) med två kolonner och ett antal(antal mätpunkter) rader. I den första kolonnen ska det stå frekvens i rad/s och i den andra överföringsfunktionens värde i komplex form $G(i\omega)$ vid ett antal frekvenser genom att använda följande form

$$G(i\omega) = |G(i\omega)| e^{i\Phi(i\omega)} \quad (3.14)$$

I Matlab boxerna finns det en rutin som heter *lsbac* som använder minsta kvadratanpassning. Inparametrar till den har vi FR,NB,NA,WEIGHTING. Utparametrar är B, A, THE ERROR. Där B och A är parametrarna i täljaren respektive nämnaren i 3.2 och THE-ERROR ger ett mått på hur bra anpassningen är. NB och NA är gradtalen på respektive polynom. Genom att ge olika gradtal på NB och NA och titta på error samt jämföra frekvensrespons

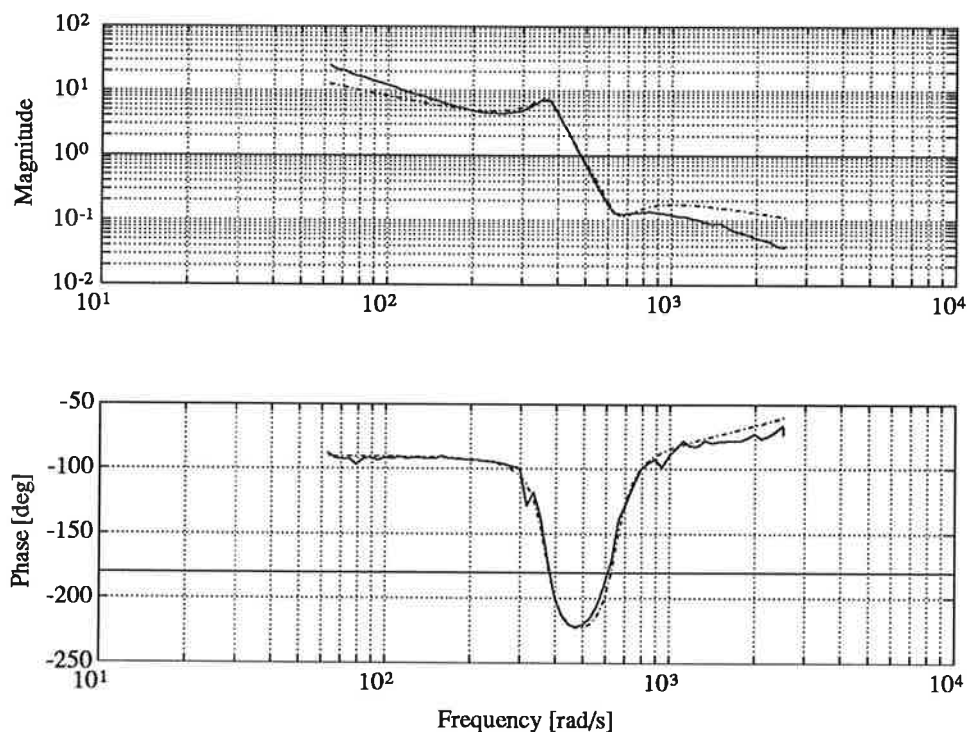


Figure 3.6 Bodediagram enligt experimentet (hel) och bodediagram till den anpassade modellen (streckad).

från den erhållna modellen (m.h.a *FRC* rutinen) fick vi bästa anpassning vid $NB=3$ och $NA=3$. Till att börja med viktade vi med defaultvärdet d.v.s vi viktade alla värden med ett och sedan itererade vi fram till en ännu bättre anpassning genom att ändra viktvektorn speciellt vid låga och höga frekvenser. Det är viktigt att ha bra anpassning där det finns poler och nollställen. Efter många iterationer fick vi bästa anpassning vid en viss viktning som det framgår av koden i bilaga 6.1.

Figur 3.6 och figur 3.7 visar Bodediagram respektive Nyquistdiagram till den skattade modellen (streckad) och den verkliga från experimentet (hel). Likriktsblocket har alltså tre poler och tre nollställen, se figur 3.8.

$$\begin{aligned}
 n_1 &= -4119.4 \\
 n_2 &= -68.9 + i646.9 \\
 n_3 &= -68.9 - i646.9 \\
 p_1 &\approx 0 \\
 p_2 &= -39.02 + i366.19 \\
 p_2 &= -39.02 - i366.19
 \end{aligned}$$

Eftersom denna modell har samma grad på täljaren och nämnaren, har den en direktterm dvs en direkt koppling mellan insignal och utsignal vilket gör att det blir svårt att dimensionera en regulator som kan reglera den. Försummar vi direkttermen, får vi följande poler och nollställen, se figur 3.9.

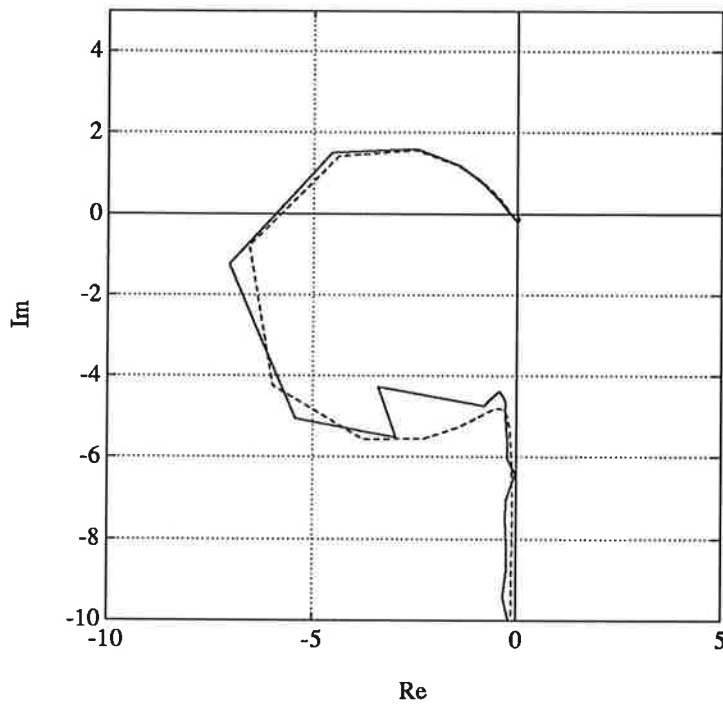


Figure 3.7 Nyquistdiagram enligt experimentet (hel) och Nyquistdiagram till den anpassade modellen (streckad).

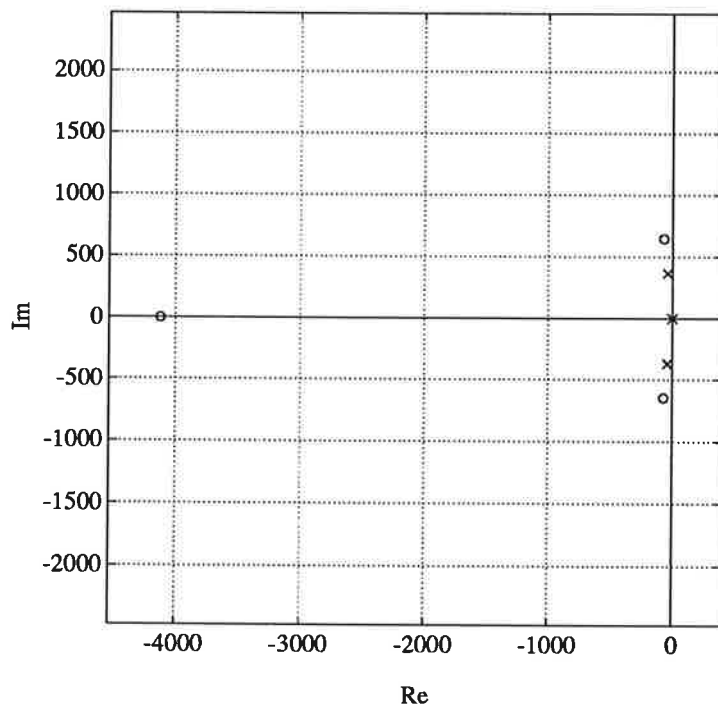


Figure 3.8 Tre poler och tre nollställen till den anpassade modellen i s-planet.

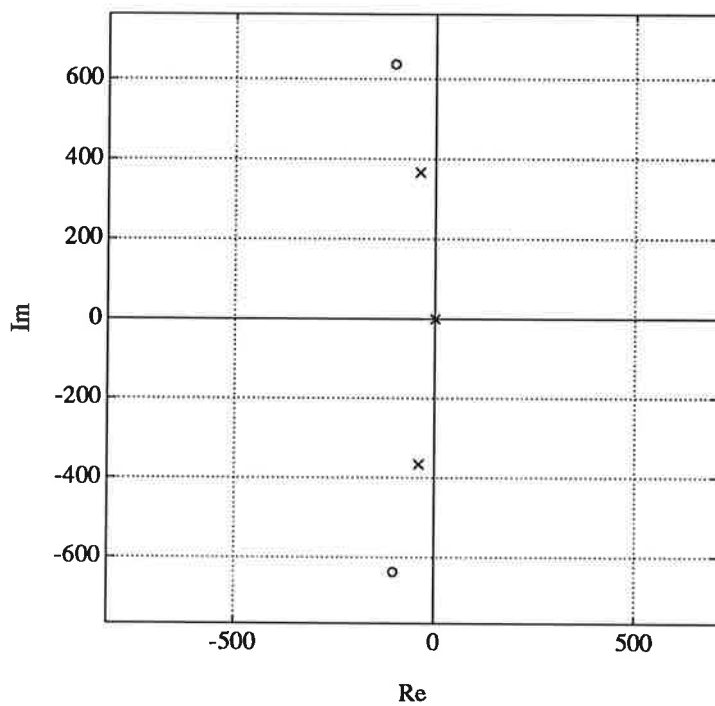


Figure 3.9 Tre poler och två nollställen till den anpassade modellen i s-planet.

$$n_1 = -102.28 + i637.69$$

$$n_2 = -102.28 - i637.69$$

$$p_1 \approx 0$$

$$p_2 = -39.02 + i366.19$$

$$p_3 = -39.02 - i366.19$$

I figur 3.10 kan vi se att om vi försummar direkt termen, påverkar det inte mycket dynamiken. Överföringsfunktionen till likriktarblocket blir

$$H_3(s) = \frac{(s - n_1)(s - n_2)}{s(s - p_1)(s - p_2)} \quad (3.15)$$

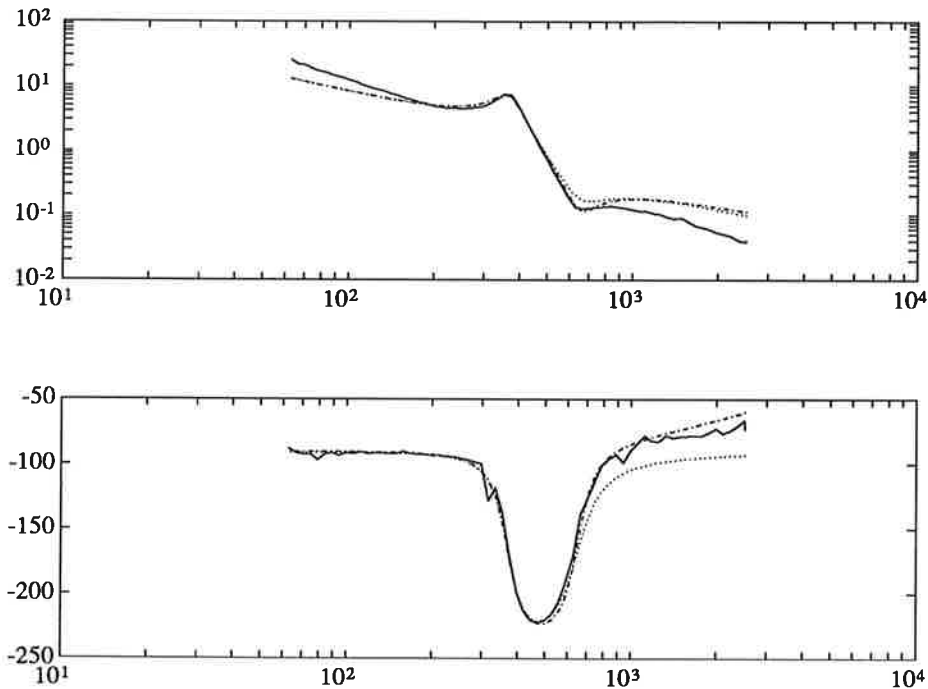


Figure 3.10 Bodediagram till processen (hel), modell med direktterm $NA=3$ och $NB=3$ (streckprickad), modell utan direktterm $NA=3$ och $NB=2$ (prickad).

3.4 Den slutliga modellen

Den slutliga modellen fås genom att sätta ihop alla block dvs likriktarblocket som har 2 nollställen och 3 poler, filter 1 som har 3 poler och filter 2 som har en pol. Totalt blir det 7 poler och 2 nollställen som framgår av figur 3.11, se också bilaga 6.2.

$$H_7(s) = \frac{(s - n_1)(s - n_2)}{s(s - p_1)(s - p_2)(0.0003s + 1)^2(0.0003s + 1.0150)} \quad (3.16)$$

Man eftersträvar modeller av låg ordning för att få enklare förståelse av processens dynamik samt göra enklare regulatorer. Som vi kan se så har filter 1 mycket snabbare poler än de andra. Om vi försummar dessa tre poler, kommer processen att vara av 4:e ordning, se figur 3.12, och Bodediagrammen till båda modellerna visas i figur 3.13.

$$H_4(s) = \frac{(s - n_1)(s - n_2)}{s(s - p_1)(s - p_2)(0.0003s + 1.0150)} \quad (3.17)$$

Vi ser att amplitudkurvan inte ändras medan faskurvan är lite mer fördelaktig i det intressanta området och avviker mycket vid högre frekvenser.

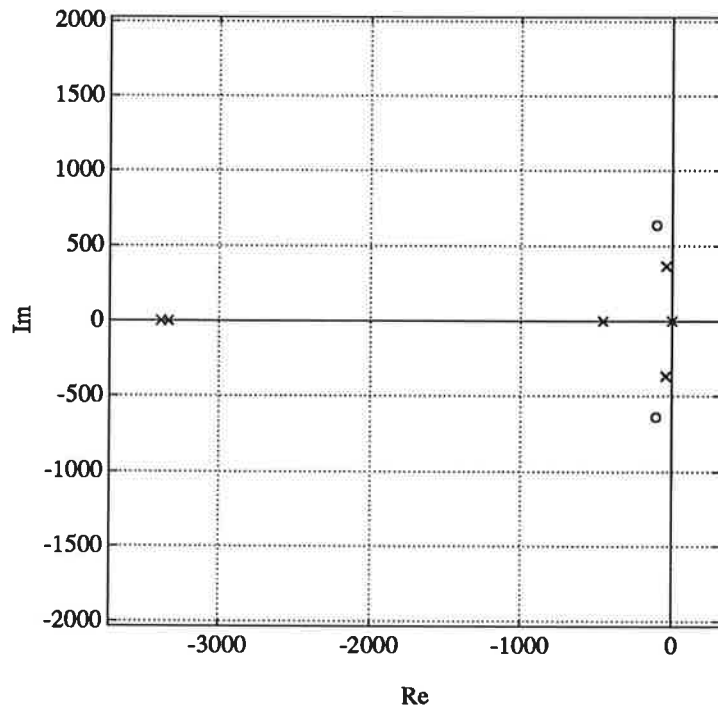


Figure 3.11 Poler och nollställen till hela processen.

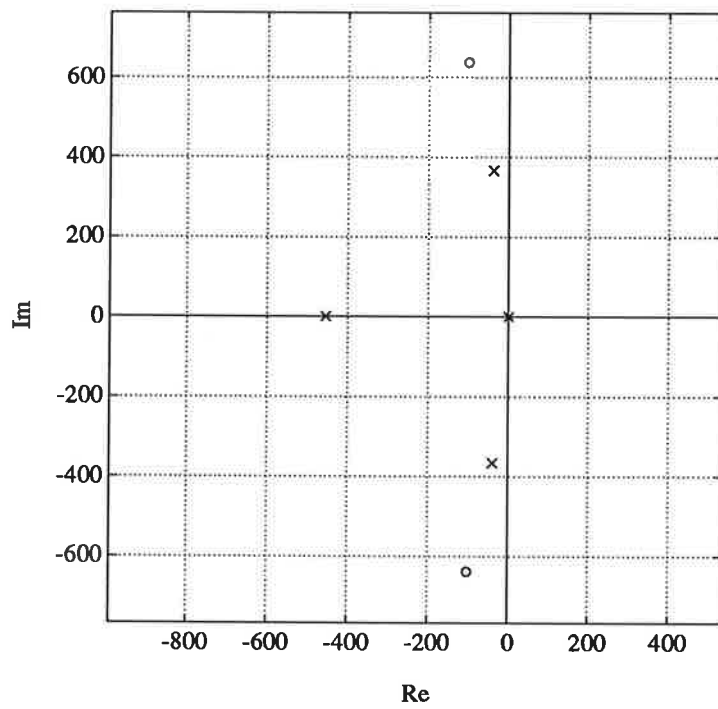


Figure 3.12 Reducerad modell till processen.

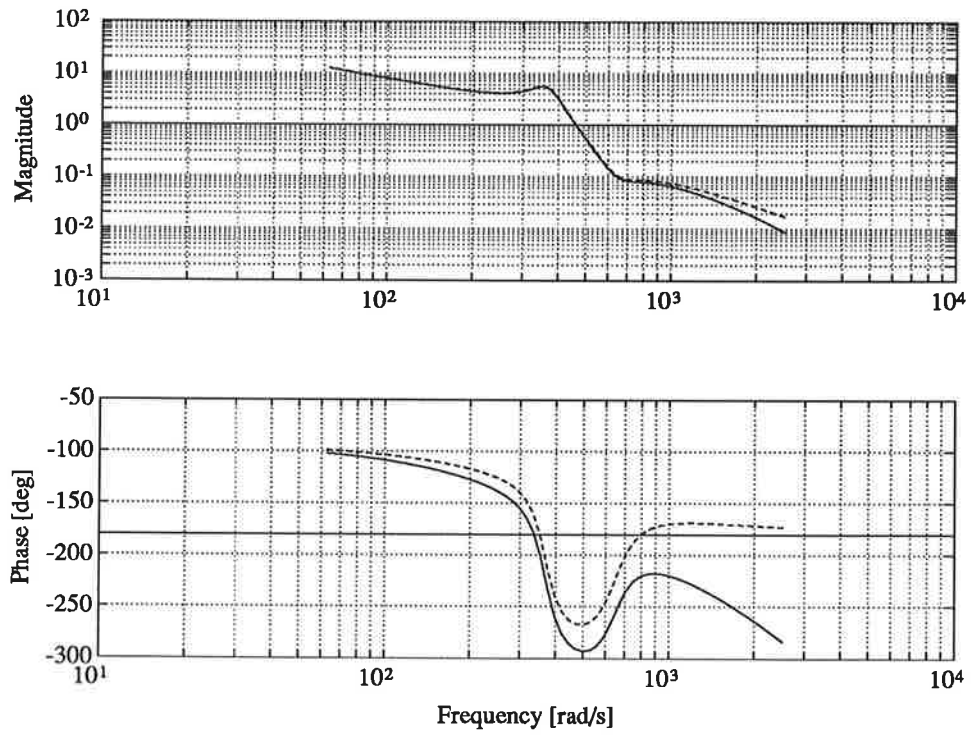


Figure 3.13 Bodediagram till processen, 7:e ordning (hel), 4:e ordning (streckad).

3.5 Modellvalidering

I figur 3.14 visas bodediagrammet till hela processen (2 nollställen och 7 poler).

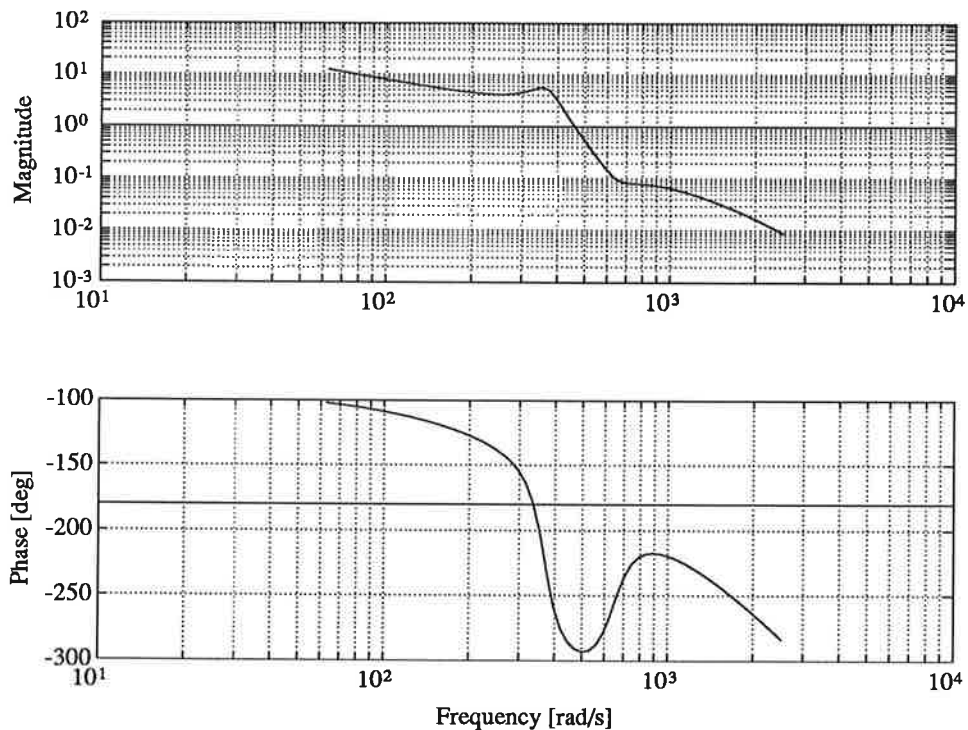


Figure 3.14 Bodediagram till hela processen.

Som vi kan se så skär faskurvan -180° axeln vid cirka 320 rad/sek vilket motsvarar cirka 50 Hz och att amplitudmarginalen är $1/5 = 0.182$. När vi gjorde Ziegler-Nichols test på processen, fick vi självsvängning vid 50 Hz och den kritiska förstärkningen var cirka 0.2 . Detta överensstämmer bra med vad modellen förutsäger.

4. Reglering

I det här kapitlet ska vi behandla regulatordesign. Det finns huvudsakligen två olika tillvägagångssätt att dimensionera en regulator. Det första sättet att göra dimensioneringen är att man utgår ifrån processmodellen d.v.s processen antas vara känd. Processmodellen kan i enkla fall räknas fram analytiskt eller tas fram genom att använda olika identifieringsmetoder så som vi har gjort. Det andra sättet att göra dimensioneringen är utan att ha någon kännedom av processmodellen.

4.1 Kravspecifikation

De krav specificationer som måste uppfyllas är följande

1. Utspänningen börjar stiga X ms efter tillslag av matningspänningen. Utspänningen stiger till 24 V, vid nominell belastning, inom ≤ 5000 ms. 10% – 90% av inställt värde ≤ 200 ms.
2. Regulatornoggrannhet statistiskt $0 - 100\%I_{ut} \leq 50$ mV och vid lastdelning ≤ 300 mV.
3. Regulatornoggrannhet dynamiskt vid 50% laständring (=25 A), inom lastområdet 10 – 90% utan batteri, ≤ 2 V. Insvängningstiden ≤ 100 ms.
4. Psfometriska (dvs hörbara) störningar ≤ 1 mV.

4.2 Linjär tidsdiskret regulator

I detta avsnitt ska vi behandla först sampling av kontinuerliga system kortfattat. Sedan ska vi behandla dimensionering av linjära tidsdiskreta regulatorer som ofta kallas för RST-regulatorer. För att dimensionera dessa regulatorer ska vi använda polplaceringsmetoden som baseras på insignal utsignalmodellen. För mer detaljer hänvisas till (Computer Controlled Systems, Åström och Wittenmark).

4.2.1 Sampling av kontinuerliga system

Ett fundamentalproblem är att beskriva ett kontinuerligt system kopplat till en dator via A/D och D/A omvandlare. Betrakta det system som visas i figur 4.1.

Signalerna i datorn är sekvenserna $\{u(t_k)\}$ och $\{y(t_k)\}$. Problemet är att finna en relation mellan dessa sekvenser. Att finna det diskreta systemet till det kontinuerliga systemet kallas för sampling av det kontinuerliga systemet. Den erhållna modellen kallas för (eng stroboscopic) modell eftersom den ger förhållandet mellan variablerna bara vid samplingsögonblicket. Antag att det kontinuerliga systemet är givet av följande tillståndsform

$$\begin{aligned}\dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t) \\ y(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t)\end{aligned}\tag{4.1}$$

Systemet har r insignaler, p utsignaler, och är av ordning n .

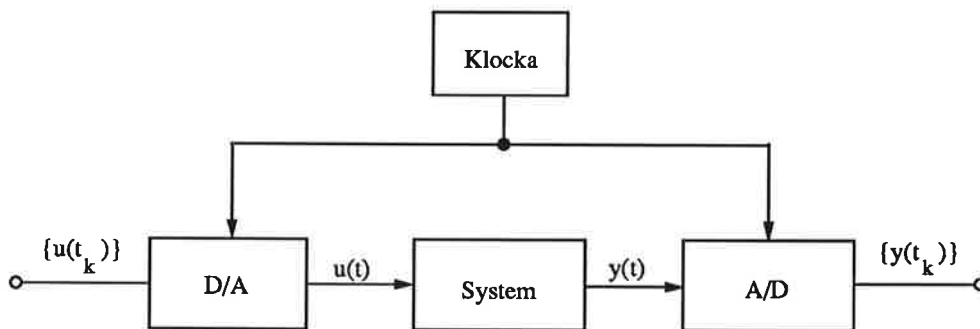


Figure 4.1 Blockschema över ett kontinuerligt system kopplat till A/D och D/A omvandlare.

där

$$\dot{x}(t) = \begin{pmatrix} \dot{x}_1(t) \\ \dot{x}_2(t) \\ \vdots \\ \dot{x}_n(t) \end{pmatrix}, A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix}, B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1r} \\ b_{21} & b_{22} & \dots & b_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nr} \end{pmatrix}$$

$$u(t) = \begin{pmatrix} u_1(t) \\ u_2(t) \\ \vdots \\ u_r(t) \end{pmatrix}, y(t) = \begin{pmatrix} y_1(t) \\ y_2(t) \\ \vdots \\ y_p(t) \end{pmatrix}, C = (c_1 \quad c_2 \quad \dots \quad c_n)$$

$$D = \begin{pmatrix} d_{11} & d_{12} & \dots & d_{1r} \\ d_{21} & d_{22} & \dots & d_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ d_{p1} & d_{p2} & \dots & d_{pr} \end{pmatrix}$$

Samplar man ovanstående kontinuerliga system med samplingsperioden h samt använder man D/A med hållkrets (eng zero order hold) dvs styckvis konstant insignal, får man följande diskreta tidsinvarianta system

$$\begin{aligned} x(kh + h) &= \Phi x(kh) + \Gamma u(k) \\ y(kh) &= Cx(kh) + Du(kh) \end{aligned} \quad (4.2)$$

där

$$\begin{aligned} \Phi &= e^{Ah} \\ \Gamma &= \int_0^h e^{As} ds \cdot B \end{aligned} \quad (4.3)$$

För att kunna sampla ett kontinuerligt system behövs det alltså att man räknar en matrisexponential och en integration av en matrisexponential. Det finns olika metoder att göra detta

- Serie utveckling av matrisexponentialen.
- Laplace transform, laplace transform av e^{At} är $(sI - A)^{-1}$.
- Cayley- Hamilton's teorem (se CCS Åström).
Ett sätt att förenkla räkningen är att beräkna

$$\Psi = \int_0^h e^{As} ds = Ih + \frac{Ah^2}{2!} + \frac{A^2h^3}{3!} + \dots + \frac{A^i h^{i+1}}{(i+1)!} + \dots$$

Matriserna ϕ och γ ges då av

$$\begin{cases} \Phi = I + A\Psi \\ \Gamma = \Psi B \end{cases} \quad (4.4)$$

Handräkning är möjligt för system av låg ordning, $n \leq 2$. För högre ordning behövs det dator hjälp. I Matlab utförs detta med hjälp av kommandot `c2d`.

Vi inför framåt skiftoperatör q som har egenskapen

$$qf(k) = f(k+1)$$

och bakåt skiftoperatör q^{-1} som har egenskapen

$$q^{-1}f(k) = f(k-1)$$

Skift operatör q används för att förenkla manipulationen av högre ordnings differensekvationer. Betrakta

$$y(k+na) + a_1y(k+na-1) + \dots + a_nay(k) = b_0u(k+nb) + \dots + b_nb u(k) \quad (4.5)$$

där $na \geq nb$.

Genom att använda skiftoperatör q får vi

$$(q^{na} + a_1q^{na-1} + \dots + a_n) y(k) = (b_0q^{nb} + \dots + b_nb) u(k)$$

I många läroböcker användes z som såväl skiftoperatör som z -transformen. Med introduktion av polynomen

$$A(z) = z^{na} + a_1z^{na-1} + \dots + a_n$$

och

$$B(z) = b_0z^{nb} + b_1z^{nb-1} + \dots + b_nb$$

kan differensekvationen 4.5 skrivas som

$$A(q)y(k) = B(q)u(k) \quad (4.6)$$

Med bakåt skiftoperatör blir (x)

$$A^*(q^{-1})y(k) = B^*(q^{-1})u(k-d) \quad (4.7)$$

där $d = na - nb$, $A^*(q^{-1}) = q^{-na}A(q)$ och $B^*(q^{-1}) = q^{-nb}B(q)$ Med hjälp av skiftoperatör q kan vi skriva insignal-utsignalrelationen i form av en rationell

funktion antingen i framåt- eller bakåtskiftoperatorn. För att få insignal utsignalrelation från 4.2 måste tillståndsvektorn elimineras

$$\mathbf{x}(k+1) = \mathbf{q}\mathbf{x}(k) = \Phi\mathbf{x}(k) + \Gamma u(k)$$

$$(qI - \Phi)\mathbf{x}(k) = \Gamma u(k)$$

Detta ger

$$y(k) = C\mathbf{x}(k) + Du(k) = [C(qI - \Phi)^{-1}\Gamma + D]u(k)$$

och överföringsfunktion i diskret form blir alltså

$$H(q) = C(qI - \Phi)^{-1}\Gamma + D = \frac{B(q)}{A(q)} \quad (4.8)$$

4.2.2 Regulatordimensionering

I detta avsnitt ska vi behandla dimensionering av linjära tidsdiskreta regulatorer (RST-regulatorer) enligt polplaceringsmetoden.

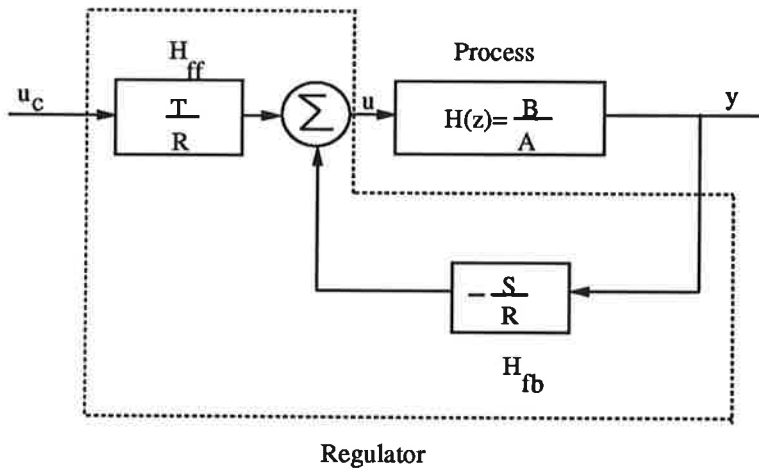
Antag att processen har en insignal u och en utsignal y . Förhållandet mellan u och y ges av överföringsfunktionen

$$H(z) = \frac{B(z)}{A(z)} \quad (4.9)$$

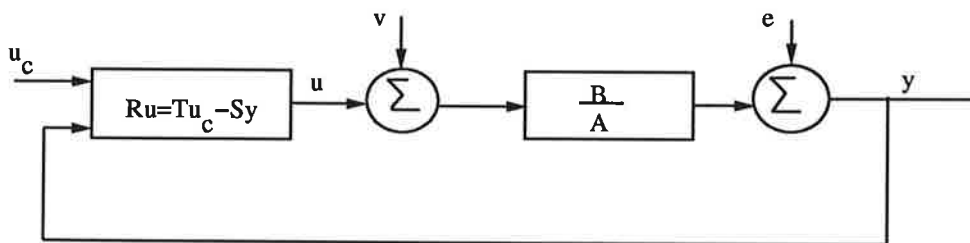
där $A(z)$ och $B(z)$ är polynom. $H(z)$ representerar processdynamiken inklusive hållkrets, sensorer, och antivikningsfilter. Polynomen A och B antas inte ha några gemensamma faktorer. Önskat slutet system är

$$H_m(z) = \frac{B_m(z)}{A_m(z)} \quad (4.10)$$

där A_m och B_m inte heller har några gemensamma faktorer.



(a)



(b)

Figure 4.2 (a) Blockdiagram över slutet system. (b) Blockdiagram över slutet system med laststörning v och mätbrus e .

Regulatorn har alltså en utsignal u och två insignaler, referenssignal u_c och processens utsignal y . Den kan skrivas som

$$u(k) = \frac{T(q)}{R(q)}u_c(k) - \frac{S(q)}{R(q)}y(k) \quad (4.11)$$

kraven

$$\deg R \geq \deg T \quad (4.12)$$

$$\deg R \geq \deg S \quad (4.13)$$

försäkrar att återkopplingsfunktionen H_{fb} och framkopplingsfunktionen H_{ff} blir kausala. Det slutna systemets överföringsfunktionen från referenssignal u_c , laststörning v , och mätbrus e till utsignalen y , se figur 4.2(b), ges av

$$y = \frac{BT}{AR + BS}u_c + \frac{BR}{AR + BS}v + \frac{AR}{AR + BS}e \quad (4.14)$$

Vi kräver att insignal utsignal förhållandet ska uppfylla

$$\frac{BT}{AR + BS} = \frac{B_m}{A_m} \quad (4.15)$$

Problemet är reducerat till att finna polynomen R , S , och T som uppfyller 4.15. Från 4.15 konstaterar vi att polerna till det slutna systemet är lösningen

till den karakteristiska ekvationen

$$AR + BS = 0 \quad (4.16)$$

Nollställena till det slutna systemet är nollställena till B och T . Det slutna systemets ordning brukar vara högre än modellens ordning. För att uppfylla 4.15 kan förkortning av poler och nollställena ske. Betrakta först det öppna systemets nollställena, dvs, nollställena till B . Om en faktor av B inte är en faktor B_m , då måste den vara en faktor av $AR + BS$. Den måste förkortas av en pol av det slutna systemet. Eftersom det slutna systemet måste vara stabilt, kan bara stabila nollställena förkortas. Faktoriserar B som

$$B = B^+ B^- \quad (4.17)$$

där B^- har alla sina nollställena utanför enhetscirkeln och B^+ har alla sina nollställena innanför enhetscirkeln. För att få en unik faktorisering sätts koefficienten till högsta exponenten i B^+ till ett. Polynomet B^+ kallas då att vara monisk. Eftersom B^- inte kan vara en faktor av $AR + BS$, måste den divideras med B_m , dvs,

$$B_m = B^- B'_m \quad (4.18)$$

Det betyder att instabila nollställena till processen ej kan ändras, utan de måste inkluderas i B_m . Eftersom B^+ är en faktor av $AR + BS$ måste den också vara en faktor av R .

$$R = B^+ R' \quad (4.19)$$

Ekvation 4.15 kan då skrivas som

$$\frac{B^+ B^- T}{B^+(AR' + B^- S)} = \frac{B^- B'_m}{A_m}$$

Förkortning av gemensamma faktorer ger

$$\frac{T}{AR' + B^- S} = \frac{B'_m}{A_m}$$

och att A_m är en faktor av $AR' + B^- S$. Observerar polynomet A_o förkortas i överföringsfunktionen från referenssignalen till utsignalen. Därför måste A_o vara en faktor av $AR + BS$. Vi får alltså

$$AR' + B^- S = A_o A_m \quad (4.20)$$

och

$$T = B'_m A_o$$

Det slutna systemets karakteristiska ekvation blir

$$AR + BS = B^+ A_o A_m \quad (4.21)$$

Polerna till det slutna systemet är alltså processens förkortade stabila nollställena, B^+ , modellens poler A_m och observerarpoler A_o . Det är bra att ha hög återkopplingsförstärkning vid låga frekvenser för att få ett system som inte är känsligt för fel i modellen vid låga frekvenser och lågfrekventlast störning. Detta kan åstadkommas genom att låta $(z - 1)^l$ vara en faktor i R . Inför man

$$R = (z - 1)^l R_1 \quad (4.22)$$

ger 4.20

$$A(z-1)^l R'_1 + B^- S = A_o A_m \quad (4.23)$$

Det existerar en kausal lösning till polplaceringen om följande krav är uppfyllda

$$\deg A_m - \deg B_m \geq \deg A - \deg B \quad (4.24)$$

och

$$\deg A_o \geq 2\deg A - \deg A_m - \deg B^+ + l - 1 \quad (4.25)$$

där A_o är observerarpolnomet, B^+ är den del av B som har alla sina nollställen inom stabilitets regionen i z -planet, l är antalet integratorer. Vi kan sammanfatta algoritmen

- *Data Process* modellen B/A , observerar polnomet A_o , önskat slutet system B_m/A_m , och stabilitets regionen .
- *Krav* Det förutsätts att *Data* uppfyller 4.18, 4.24 och 4.25.

1. Faktorisera B och B_m som

$$B = B^- B^+$$

$$B_m = B^- B'_m$$

där B^+ är monisk och har alla sina nollställen inom stabilitets regionen, och B^- har alla sina nollställen utanför.

2. Lös ekvationen

$$(z-1)^l A R'_1 + B^- S = A_o A_m$$

i förhållande till R'_1 och S . Välj en lösning så att

$$\deg S \leq l + \deg A$$

och

$$\deg R'_1 = \deg A_o + \deg A_m - \deg A - l$$

3. Regulatorn ges då av

$$Ru = Tu_c - Sy$$

där

$$R = B^+ R'_1$$

$$T = B'_m A_o$$

$$R'_1 = (z-1)^l R'_1$$

För att kunna välja var polerna av det slutna systemet ska placeras finns det två olika tillvägagångssätt att göra det.

- Dimensionera en regulator enligt linjärvadratisk lösning. En utförlig beskrivning av detta finns i (CCS, Åström och Wittenmark). I Matlab kan man beräkna en sådan regulator med hjälp av *lqr*. Tag sedan reda på var polerna av det slutna systemet ligger.

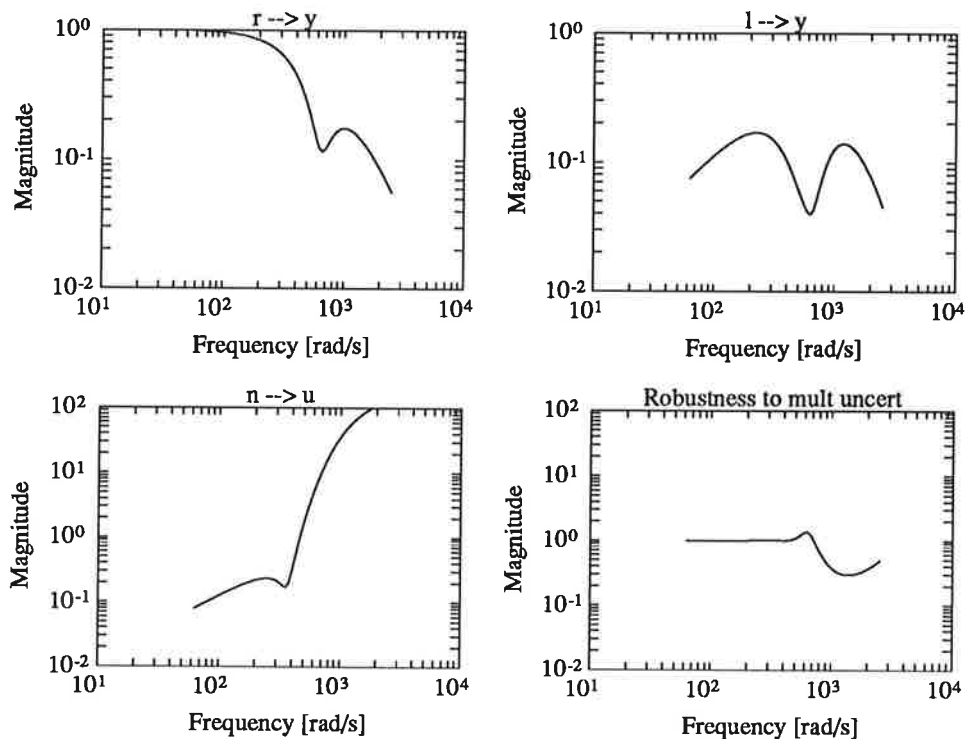


Figure 4.3 Simulering av överföringsfunktionen från referens r till utsignalen y , från last l till y , från mätbrus till styrsignalen u , samt osäkerhet i modellen (robusthet).

- Placera de enligt Butterworth-mönstret. I Matlab finns det en rutin som heter *polybutt* som placerar poler enligt Butterworth-mönstret.

Polybutt användes för att placera både A_m :s och A_o :s rötter, se bilaga 6.3. Regulatorn RST beräknades med hjälp av *rstd* och polynomen R , S , och T fick följande parametrar

$$R(q) = q^4 - 2.703q^3 + 2.015q^2 + 0.183q - 0.495.$$

$$S(q) = 62.950q^4 - 218.954q^3 + 286.572q^2 - 1671.96q + 36.670.$$

$$T(q) = 2.405q^4 - 6.261q^3 + 5.988q^2 - 2.545q + 0.405.$$

4.3 Tidsdiskret PID-regulator

De flesta reglerproblemen kan lösas med PID regulatorer. Det är ett standardverktyg för reglerproblem. Lärobokversion av PID-regulator brukar skrivas på följande form

$$u(t) = K[e(t) + \frac{1}{T_i} \int_{-\infty}^t e(s)ds + T_d \frac{de(t)}{dt}] = P + I + D \quad (4.26)$$

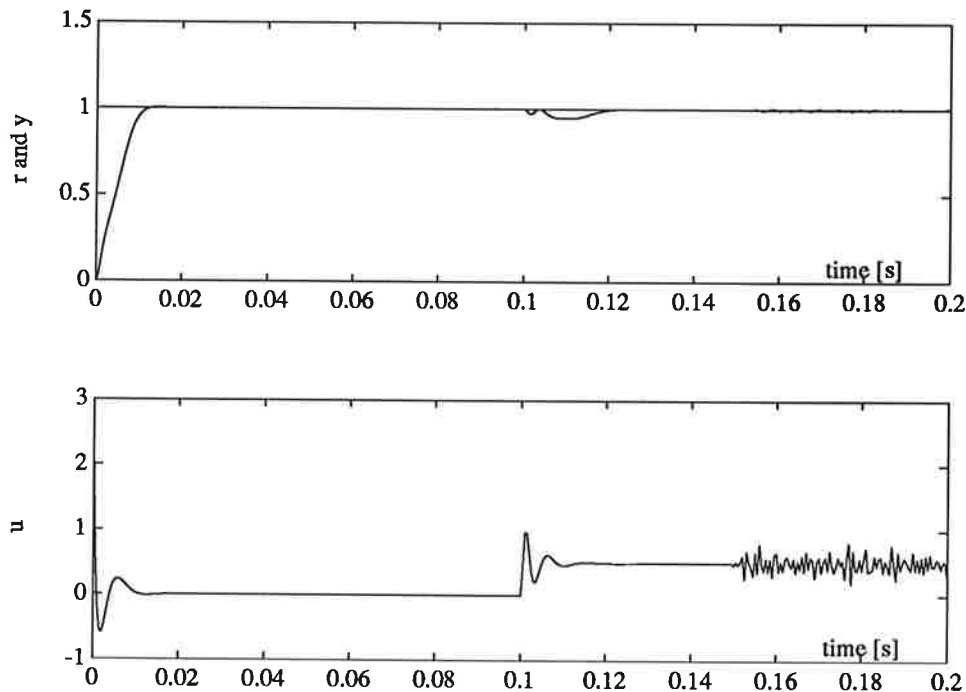


Figure 4.4 Simulering av slutna systemet. Steg vid $t=0$, laständring vid $t=0.1$, mätbrus vid $t=0.15$.

där

$$e(t) = u_c(t) - y(t).$$

$$T_i = \text{Integrationstid.}$$

$$T_d = \text{Derivatationstid.}$$

$$u(t) = \text{styrsignal.}$$

Felet e är skillnaden mellan referenssignal u_c (börvärdet) och processens utsignal y (ärvärdet). Styrsignal består alltså av tre termer, en term (P) som är proportionell mot felet e , en annan (I) som är proportionell mot integralen av felet e , och sista termen som är proportionell mot derivatan av felet e . Speciella fall kan erhållas genom att använda några av termerna d.v.s P, I, PI, PD-regulatorer. PI-regulatorer är vanligast. Det är också möjligt att ha mer komplicerade regulatorer t ex en extra derivataterm som ger en PIDD eller en DPID regulator. Namnet PID regulator brukar användas till alla dessa varianter. PID-regulatorer är mycket vanliga. De kan implementeras på många olika sätt. Ursprungligen implementerades PID-regulatorer med analogteknik. Idag implementeras nästan alla PID-regulatorer digitalt tack vare dagens teknik (mikroprocessorer). Först implementerade man PID-regulatorer genom en ren översättning av 4.26 vilket gjorde att man förlorade många av de goda egenskaperna som fanns hos den analoga implementationen.

4.3.1 Modifierad PID-regulator

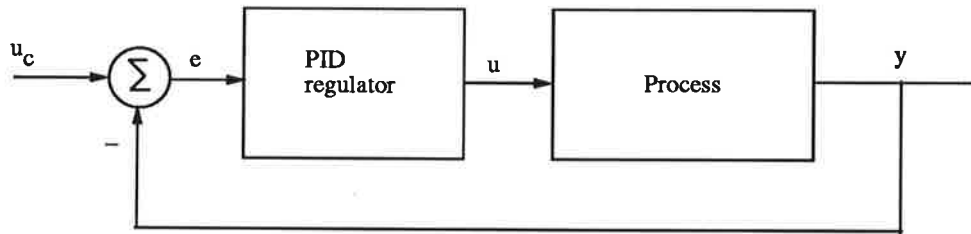


Figure 4.5 PID-reglering.

En ren derivata kan och bör inte implementeras eftersom den ger upphov till hög förstärkning av mätbrus. Förstärkning av derivata delen måste då vara begränsad. Detta kan åstadkommas genom att approximera sT_d som följande

$$sT_d \approx \frac{sT_d}{1 + sT_d/N} \quad (4.27)$$

Överföringsfunktionen till höger approximerar bra vid låga frekvenser men förstärkningen är begränsad till N vid höga frekvenser. Parametern N kallas därför högsta derivataförstärkning. Typiska värden för N är i intervallet 10 – 20. Approximationen i 4.27 ger 90° fas vridning för låga frekvenser. Vid $\omega = \sqrt{\frac{N}{T_d}}$ har fas vridningen sjunkit till 45° .

När man arbetade med analoga regulatorer fann man det fördelaktigt att inte låta derivatan verka på referenssignal eftersom derivivering av t ex steg ändringar kan ge mycket stora styrsignaler. Sedan fann man det också lämpligt att låta bara en del av referenssignal verka på proportionaldelen. PID-algoritmen blir

$$U(s) = K[bU_c(s) - Y(s) + \frac{1}{T_i}(U_c(s) - Y(s)) - \frac{sT_d}{1 + sT_d/N}Y(s)] \quad (4.28)$$

Där U , U_c och Y är Laplacetransform av u , u_c , och y . Det finns andra varianter av PID-algoritmen som används i kommersiella system.

4.3.2 Parameter inställning

På grund av PI- och PID-regulatorers allmänna förekomst har det funnits stort behov av enkla tumregler för inställning av reglerparametrar som ger acceptabla slutna regleregenskaper. Sådana ska möjliggöra en snabb trimning av parametrarna under drift, utan behov av komplicerade teoretiska analyser eller dyrbara specialinstrument. En tumregel som publicerades 1942 och fått stor användning är Ziegler-Nichols (svängnings) metod. Med denna metod kan man experimentellt finna överföringsfunktion $G(i\omega)$ vid en enda frekvens $i\omega_c$, där Nyquistkurva passerar den negativa reella axeln. Denna punkt kallas för den kritiska punkten, se figur 4.6.

Metoden är baserad på att man observerar när systemet blir instabilt med bara proportionell återkoppling, se figur 4.7. Antag att förstärkningen är tillräckligt stor så att processen är på stabilitetsgräns. Styrvariabeln u och utsignalen y sinusformade med fasförskjutning på -180° . De är dessutom relaterade med

$$u = -Ky$$

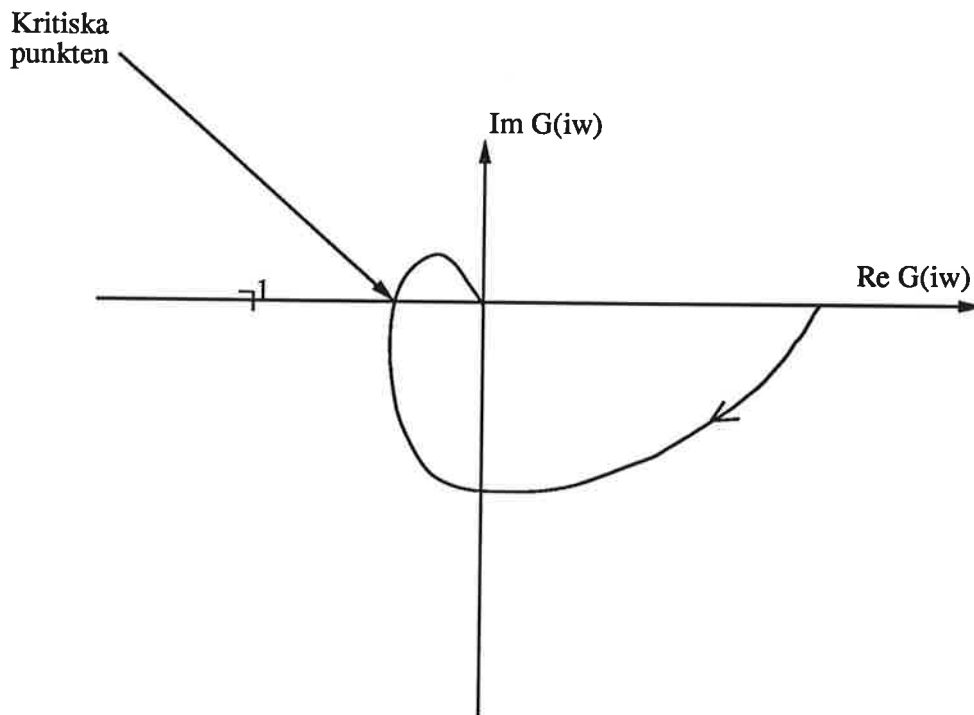


Figure 4.6 Nyquistkurva av ett system.

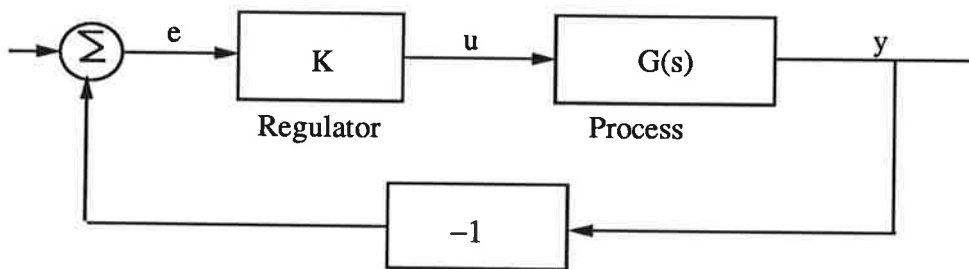


Figure 4.7 Blockdiagram av en sluten slinga, Systemet har proportional återkoppling.

på grund av proportionalåterkopplingen. Eftersom slingförstärkningen måste vara 1 för att behålla svängningen

$$K_c G(i\omega_c) = 1 \quad (4.29)$$

där förstärkningen som driver systemet till stabilitetsgräns, är kallad för den kritiska förstärkningen (K_c) och svängningsfrekvens $f_c = \frac{1}{T_c}$ för den kritiska frekvensen. Av 4.29 följer att

$$G(i\omega_c) = \frac{-1}{K_c} \quad (4.30)$$

Vi kan sammanfatta Ziegler-Nichols test med följande steg

1. Ställ in regulatorn så att den arbetar som ren P regulator, dvs sätt $T_i = \infty$ och $T_d = 0$.
2. Öka regulatorförstärkningen K successivt tills systemet når gränsen till instabilitet och självsvänger med konstant amplitud.

Regulator	Parametrar		
	K	T_i	T_d
P-regulator	$0.5 K_c$	-	-
PI-regulator	$0.45 K_c$	$0.8 T_c$	-
PID-regulator	$0.6 K_c$	$0.5 T_c$	$0.125 T_c$

Figure 4.8 Regulatorparametrar enligt Ziegler-Nichols

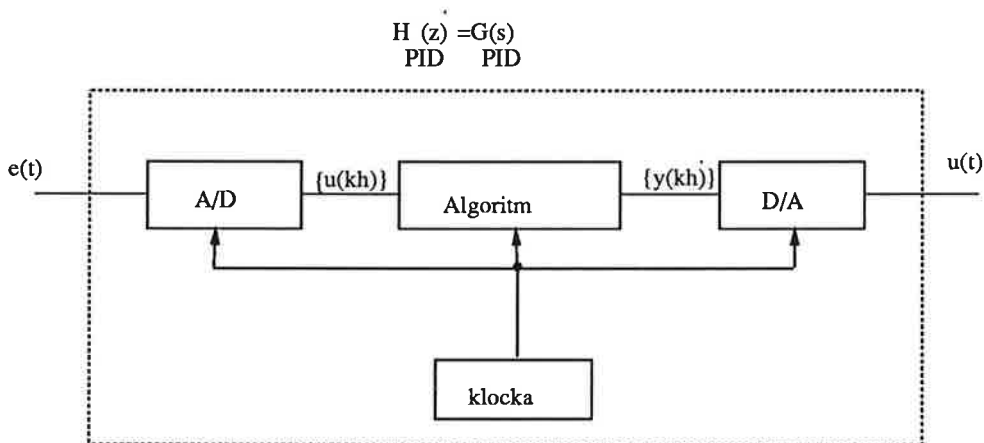


Figure 4.9 Approximering av regulatorns analoga överföringsfunktion $G(s)$ med dator

3. Notera det K-värdet ($= K_c$) då detta inträffar samt periodtiden ($= T_c$) hos självsvängningen. Den senare fås normalt från den skrivare som registrerar processutsignalen, alternativt en bildskärm eller plotter där denna presenteras.
4. Ändra regulatorparametrarna till de i figur 4.8 angivna värden.

4.3.3 Diskretisering av PID regulator

I detta avsnitt antar vi att den kontinuerliga PID-regulatorn $G_{PID}(s)$ är given. Vi önskar finna en algoritm till datorn, så att det digitala systemet approximerar överföringsfunktionen $G_{PID}(s)$, se figur 4.4.

Detta problem är mycket intressant för både analoga regulatorer och digitala filter. Digital implementering innebär också att signalerna måste rekonstrueras. Detta görs på många olika sätt t ex hållkrets som ger styckvis konstanta signaler. Några av de vanligaste approximationerna är differentiering och Tustin approximationer. En överföringsfunktion representerar en differentialekvation. Det är naturligt att erhålla en differensekvation genom

att approximera derivatan med framåt differens (Eulers metod)

$$p_x(t) = \frac{dx(t)}{dt} \approx \frac{x(t+h) - x(t)}{h} = \frac{q-1}{h} x(t)$$

eller bakåt differens

$$p_x(t) = \frac{dx(t)}{dt} \approx \frac{x(t) - x(t-h)}{h} = \frac{q-1}{qh} x(t)$$

Översatts detta till transformvariabeln betyder det att s ersätts med $\frac{z-1}{h}$ eller $\frac{z-1}{zh}$. Variablarna s och z är relaterade till varandra med $z = e^{sh}$. Differensapproximationerna motsvarar serieutvecklingar

- Eulers metod :

$$z = e^{sh} \approx 1 + sh \quad (4.31)$$

- Bakåt differens:

$$z = e^{sh} \approx \frac{1}{1 - sh} \quad (4.32)$$

En tredje approximation som motsvarar trapetsmetoden för numerisk integration är

- Trapetsmetoden:

$$z = e^{sh} \approx \frac{1 + \frac{sh}{2}}{1 - \frac{sh}{2}} \quad (4.33)$$

som i digital reglering kallas för Tustins approximation eller bilinjär transformation. Genom att använda dessa approximationer, kan överföringsfunktion $H(z)$ erhållas genom att man helt enkelt ersätter varje s i $G(s)$ med s' där

1. Framåt differens eller Eulers metod

$$s' = \frac{z-1}{h} \quad (4.34)$$

2. Bakåt differens

$$s' = \frac{z-1}{zh} \quad (4.35)$$

3. Tustins approximation

$$s' = \frac{2z-1}{hz+1} \quad (4.36)$$

som ger

$$H(z) = G(s')$$

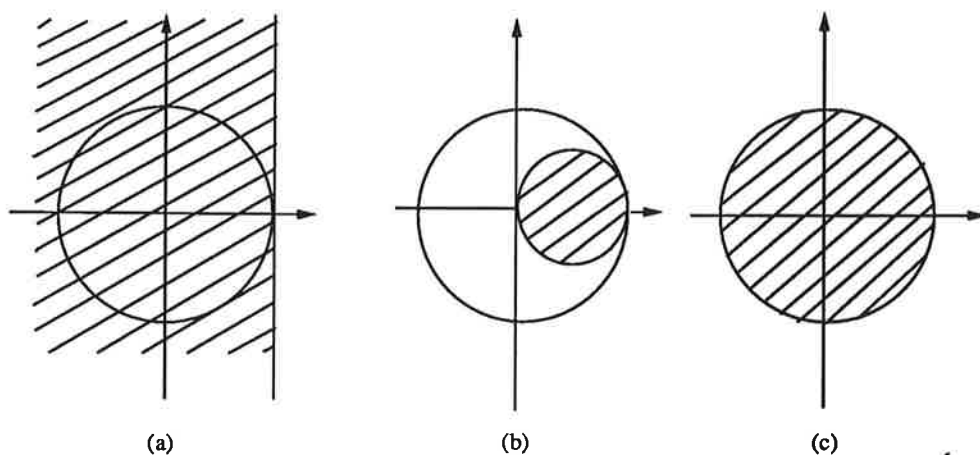


Figure 4.10 Avbildning av stabilitetsområde i s -planet på z -planet för approximationerna 4.34, 4.35 och 4.36.

Metoden är mycket enkel att tillämpa även för handräkning. Figur 4.10 visar hur stabilitetsområde $\text{Re}(s < 0)$ i s -planet avbildas på z -planet för 4.34, 4.35 och 4.36. Med framåt-differensapproximationen är det möjligt att ett stabilt kontinuerligt system kan avbildas till ett instabilt diskret system. När bakåt-differensapproximationen används, kommer ett stabilt kontinuerligt system alltid att ge ett stabilt diskret system. Det finns också instabila kontinuerliga system som transformeras till stabila diskreta system. Tustins approximationen har den fördelen att vänstra halvan av s -planet avbildas på enhetscirkeln i z -planet. Stabila kontinuerliga system kommer därför att transformeras till stabila diskreta system och instabila kontinuerliga system kommer att transformeras till instabila diskreta system. Eftersom PID-regulatorn som beskrivs av 4.28 är så enkel, finns det speciella metoder att diskritisera den. Följande är en mycket populär approximation som är mycket lätt att härleda. Proportionaldelen

$$P(t) = K(bu_c(t) - y(t)) \quad (4.37)$$

behöver ingen approximation eftersom den är en ren statisk del. Integrations-termen

$$I(t) = \frac{K}{T_i} \int e(s) ds \quad (4.38)$$

approximeras med en rektangulär approximation d.v.s

$$I(kh + h) = I(kh) + \frac{Kh}{T_i} e(kh) \quad (4.39)$$

Derivata delen ges av

$$\frac{T_d}{N} \frac{dD}{dt} + D = -KT_d \frac{dy}{dt} \quad (4.40)$$

som approximeras med bakåt-differens. Detta ger

$$D(kh) = \frac{T_d}{T_d + Nh} D(kh - h) - \frac{KT_d N}{T_d + Nh} (y(kh) - y(kh - h)) \quad (4.41)$$

Denna approximation har fördelen att den är alltid stabil och att den samplade polen går mot noll när T_d går mot noll. Tustins approximation ger en sådan

approximation att polen går mot $z = -1$ när T_d går mot noll. Styrssignalen ges av

$$u(kh) = P(kh) + I(kh) + D(kh) \quad (4.42)$$

Denna approximation har en pedagogisk fördel eftersom proportional-, integral-, och derivatatermerna är separerade. Andra approximationer ger liknande resultat. De kan allihop representeras som

$$R(q)u(kh) = T(q)u_c(kh) - S(q)y(kh) \quad (4.43)$$

där polynomen R, S, och T är av andra ordning.

$$R(q) = q^2 + r_1q + r_2 \quad (4.44)$$

$$S(q) = s_0q^2 + s_1q + s_2 \quad (4.45)$$

$$T(q) = t_0q^2 + t_1q + t_2 \quad (4.46)$$

Polynomet R är på följande form

$$R(q) = (q - 1)(q - a_d) = q^2 + r_1q + r_2$$

Där

$$s_0 = K(1 + b_d)$$

$$s_1 = -K(1 + a_d + 2b_d - b_i)$$

$$s_2 = K(a_d + b_d - b_i a_d)$$

$$t_0 = Kb$$

$$t_1 = -Kb$$

$$t_2 = Ka_d(b - b_i)$$

$$a_d = \frac{T_d}{Nh + T_d}$$

$$b_d = Na_d$$

$$b_i = \frac{h}{T_i}$$

4.3.4 Val av samplingsfrekvensen

I detta avsnitt ska vi ge några tumregler för val av samplingsperioden för digitala PID-regulatorer. Det finns signifikanta skillnader mellan PI-, och PID-regulatorer. För PI-regulatorer är samplingsperioden relaterad till integrationstiden. En tumregel är

$$\frac{h}{T_i} \approx 0.1 - 0.3$$

För PID-regulatorer måste samplingsperioden vara så kort att fasen inte påverkas av samplingen. Detta leder till att samplingsperioden måste väljas så att hN/T_d ska ligga i intervallet (0.2 - 0.6). Om $N=10$, ger

$$\frac{hN}{T_d} \approx 0.2 - 0.6$$

4.4 Reglering av likriktaren

Vi valde i första hand att implementera en PID-regulator eftersom den är enkel och klarar i de flesta fallen kravspecifikationerna. Den kräver färre variabler än polynommetoden då vi hade begränsningar i minnet eftersom mikroprocessorn sköter andra uppgifter vid sidan av regleringen.

4.4.1 Val av PID-algoritm

Vi försökte i början implementera 4.28, men vi fick inte den att fungera. Det berodde antagligen på fel i assemblekoden. Samtidigt försökte vi implementera följande algoritm vilket fungerade bra

$$u_{styr} = K[(b \cdot v_{ref} - y) + \frac{1}{sT_i}(v_{ref} - y) - sT_d y] \quad (4.47)$$

$$u_{styr} = [K(b \cdot v_{ref} - y) + \frac{K}{sT_i}(v_{ref} - y) - KT_d s y] \quad (4.48)$$

Med bakåtapproximation, dvs ersätta varje s i 4.48 med följande uttryck

$$s = \frac{z - 1}{zT_s} = \frac{1 - z^{-1}}{T_s}$$

får vi följande digitala PID regulator

$$U_{styr}(z) = [K(b \cdot V_{ref}(z) - Y(z)) + \frac{KT_s}{T_i} \frac{1}{(1 - z^{-1})} (V_{ref}(z) - Y(z)) - \frac{KT_d}{T_s} (1 - z^{-1}) Y(z)]$$

Översätts detta till tidsplanet får vi

$$u(k) = u(k-1) + Kb v_{ref}(k) - Ky(k) - (Kb v_{ref}(k) - Ky(k-1)) + \frac{KT_s}{T_i} (v_{ref}(k) - y(k)) - \frac{KT_d}{T_s} (y(k) - 2y(k-1) + y(k-2)) \quad (4.49)$$

Indexen k betyder det nuvarande samplet, $k-1$ ett sampl tidigare o.s.v. Värdet på fyra parametrar måste sättas in i 4.49. Det är K , T_i , T_d , T_s . Vi gjorde *Ziegler - Nichols* test på processen och fick följande värden på parametrarna

$$K_c = 0.23$$
$$T_c = \frac{1}{50} = 20 \text{ ms}$$

Vilket ger

$$K = 0.6 K_c = 0.138$$
$$T_i = 0.5 T_c = 10 \text{ ms}$$
$$T_d = 0.125 T_c = 2.5 \text{ ms}$$

Samplingsperioden T_s valdes till 1.26 ms

4.4.2 Regulatorprovning

Vi använde enchipdator (microcontroller 80C196 Intel) som är en 16-bits processor och drivs med 12 MHz klockfrekvens för att implementera 4.49. I bilagor 5.1, 5.2 och 5.3 visas koden i assembler för såväl PI- som PID-regulator. I bilaga 2 visas styrsignalen U_{mult} (gamla PID-regulatorn (överst), analog PI-regulator). Som vi kan se, hade den gamla PID-regulatorn en svängning vid 33 Hz som vi inte kunde förklara. Den analoga PI-regulatorn har en jämn och fin styrsignal, fast den är lastberoende.

Filter 2 hade från början en kapacitans på $2.2 \mu\text{F}$ som har ganska stor tidskonstant (2.2 ms). Först provade regulatorerna med denna kapacitans inkopplad. Sedan bytte vi den till 100 nF och körde samma mätningar en gång till. I bilagor 3.1-3.4 och 4.1-4.6 visas utsignalen (likspänning) för PI-respektive PID-regulatorer vid olika laständringar, steg i referenssignalen samt vid $2.2 \mu\text{F}$ och 100 nF. Som vi kan se, klarar båda PI- och PID-regulatorerna kravspecifikationerna bra och att det är bättre med $2.2 \mu\text{F}$ än med 100 nF.

Följande saker kontrollerades

1. Utspänningen steg till 24 V inom ≤ 200 ms.
2. Regulatornoggrannheten statistiskt 0 – 100% I_{ut} var ≤ 50 mV och vid lastdelning var den ≤ 300 mV.
3. Regulatornoggrannhet dynamiskt vid 50% laständring (25 A), inom lastområdet 10 – 90% var ≤ 2 V och insvängningstiden var ≤ 100 ms, se bilagor 3.1 – 3.2 för PI-regulator och bilagor 4.3 – 4.4 för PID-regulator.
4. Psfometriska störningar vid full last (50 A) var ≤ 1 mV.

5. Sammanfattning

Vi kan sammanfatta detta arbete med att vi har analyserat likriktaren och hittat en överföringsfunktion som beskriver dess beteende. Processen var av 7:e ordning. Vi försökte reducera ordningen genom att försumma de snabba polerna och vi fick en modell av 4:e ordningen som hade samma amplitudkurva men faskurvan var lite mer fördelaktig i det intressanta området. Det bör påpekas här att man måste vara försiktig, när man vill reducera ordningen på modellen. Om man vill förkorta några poler, måste man förflytta de kvarvarande polerna för att kompensera den förlorade dynamiken i de förkortade polerna.

Sedan vi har dimensionerat en RST-regulator (polplacering) genom att använda den reducerade modellen (4:e ordning). Dess värre kunde vi inte realisera den, eftersom den krävde fler variabler och det fanns inte tillräckligt med minne i processorn. Vi försökte sedan implementera en digital PID-regulator. Vi gjorde Ziegler-Nichols test för att inställa parametrarna i algoritmen. I början försökte vi realisera 4.28 dvs med filtrering av utsignalen. Vi fick inte den att fungera, tyvärr. Det berodde förmodligen på fel i assemblerkoden. Istället för att felsöka i koden försökte vi realisera 4.47. Vi skrev koden så att man kunde välja att antingen köra PI- eller PID-regulator. Båda fungerade bra och klarade kravspecifikationerna bra..

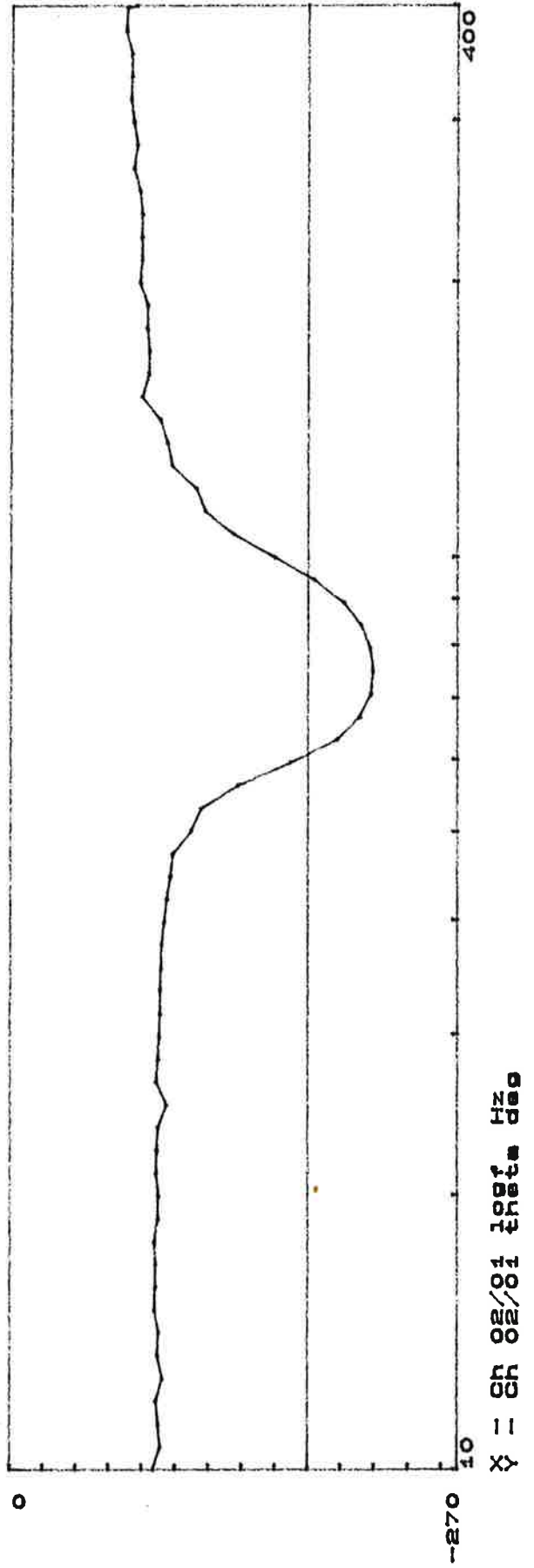
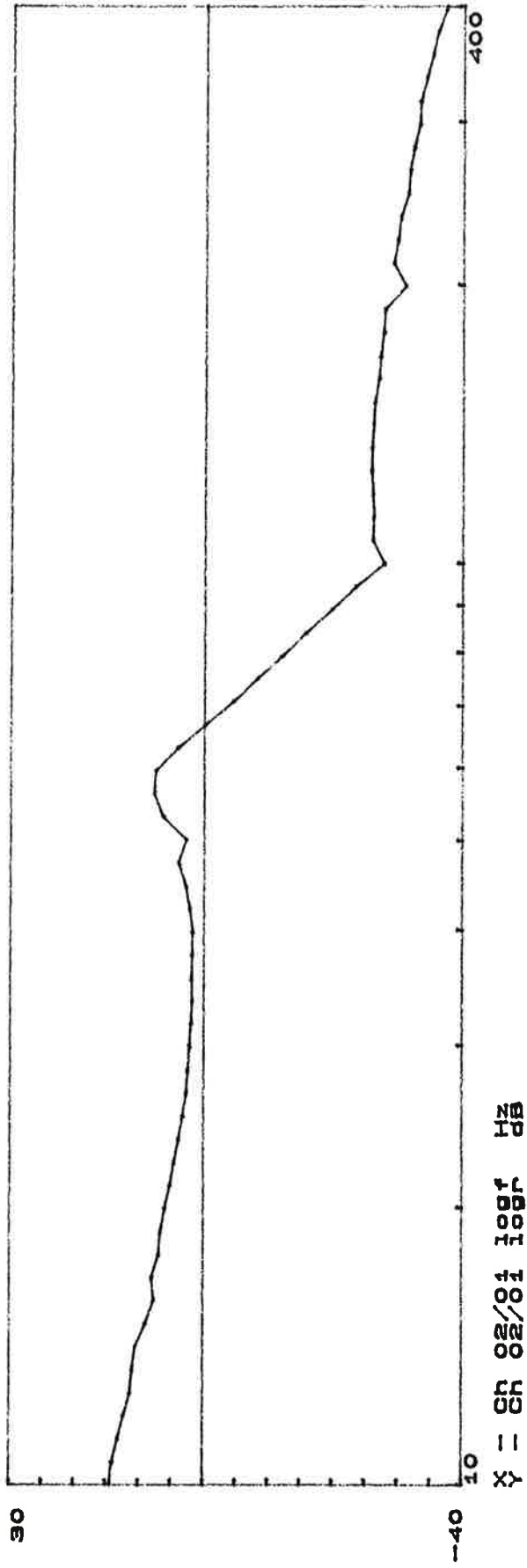
Det kan vara bra att iframtiden försöka realisera 4.28 eller göra en RST-regulator om man har tillgång till en annan processor. RST-regulatorer brukar vara bättre och klarar att reglera modeller med högre ordning än PID-regulatorer. PID- regulatorer är också bra och är mycket vanliga.

Referenser

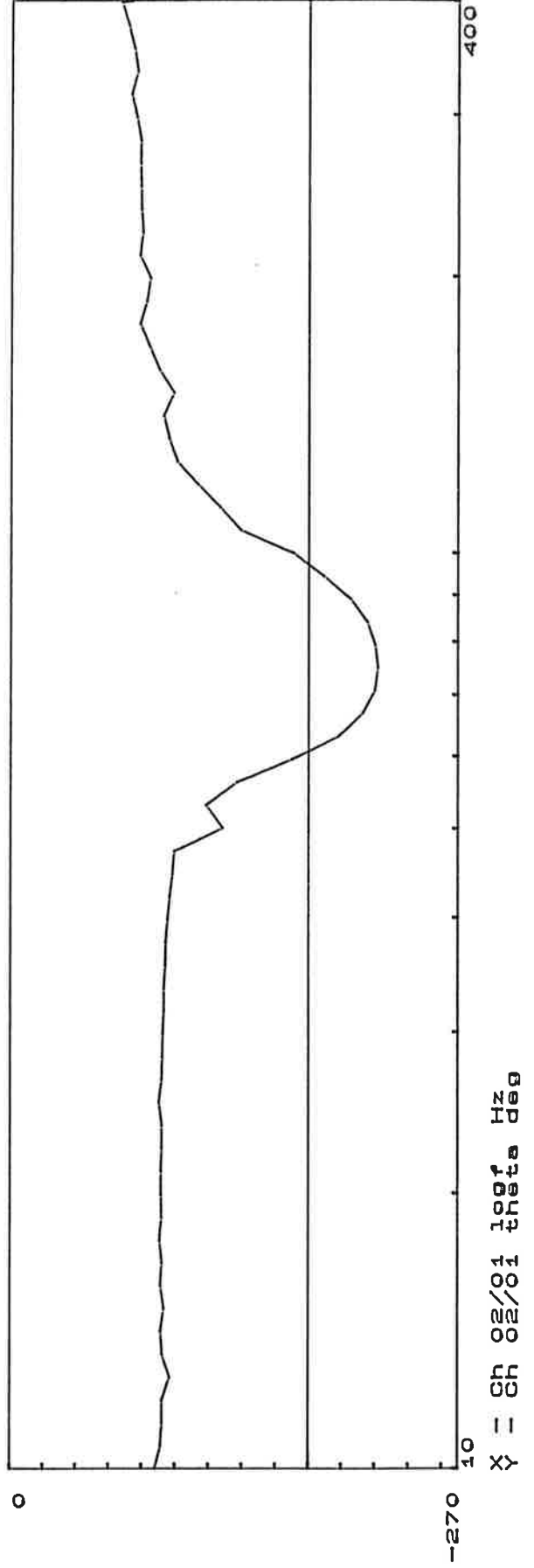
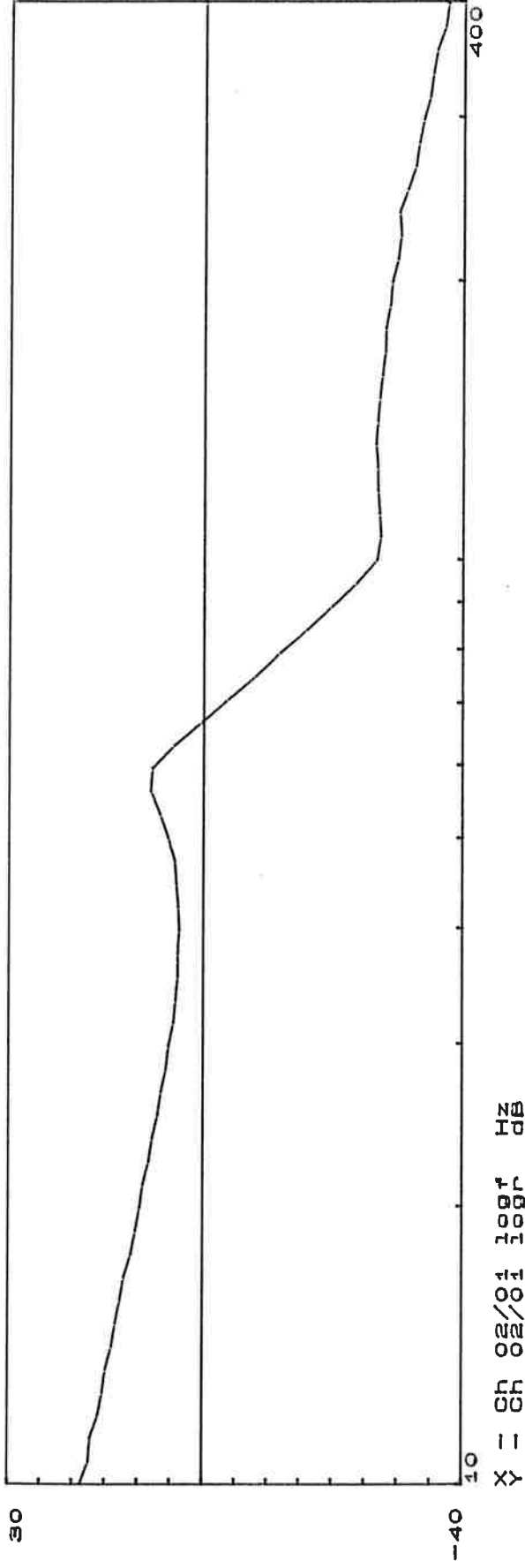
1. Åström and Wittenmark (1990): *Computer Controlled Systems*, Prentice-Hall.
2. Schmidtbauer B (1988): *Analog och digital reglerteknik*, Studentlitteratur.
3. Glad och Ljung (1981): *Reglerteknik*, Studentlitteratur.
4. Lilja M (1989): "Controller Design by Frequency Domain Approximations", Department of Automatic Control, Lund institute of Technology.
5. Åström and Hägglund (1988): *Automatic Tuning of PID Controllers*, Research Triangle Park: Instrument Society of America.
6. Johansson R (1991): *Process Identification*, Department of Automatic Control, Lund institute of Technology.
7. Bäckman N: *Stenciler*

Bilagor

Bilaga 1.1

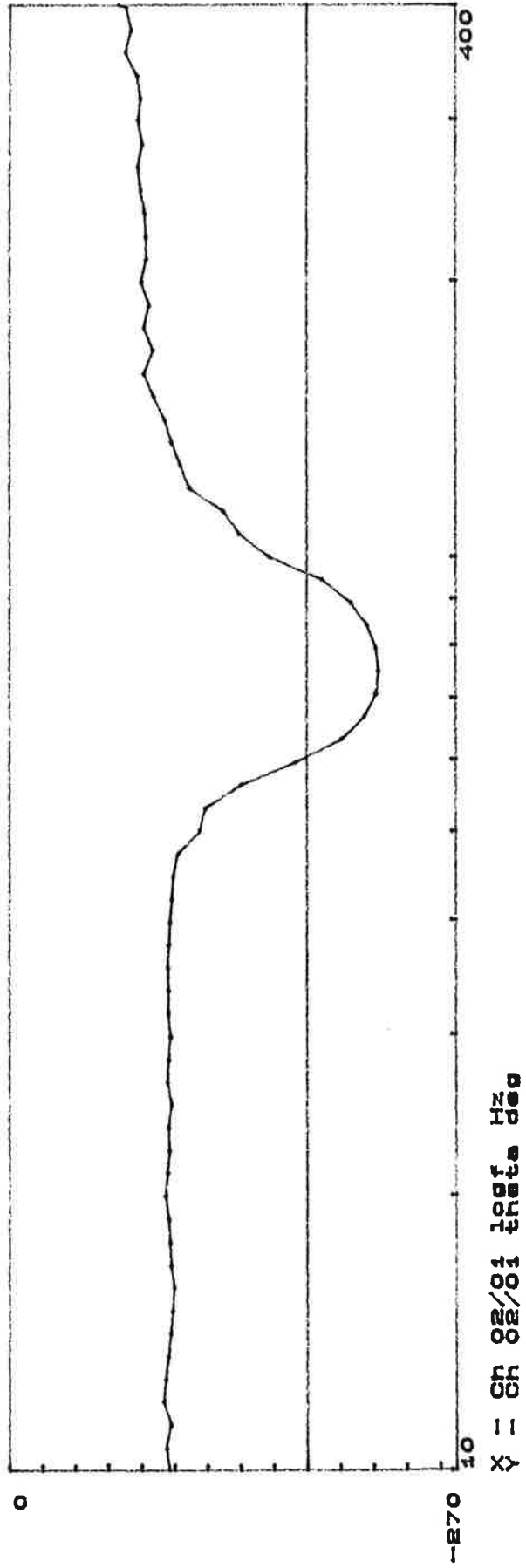
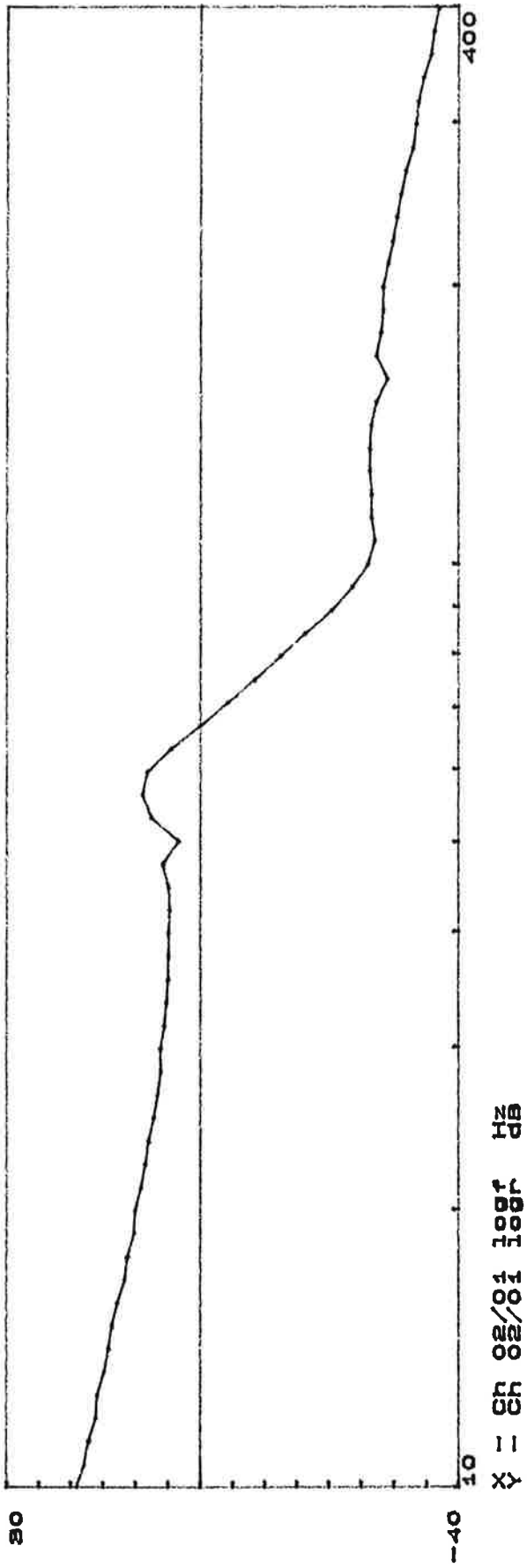


Bilaga 1.2



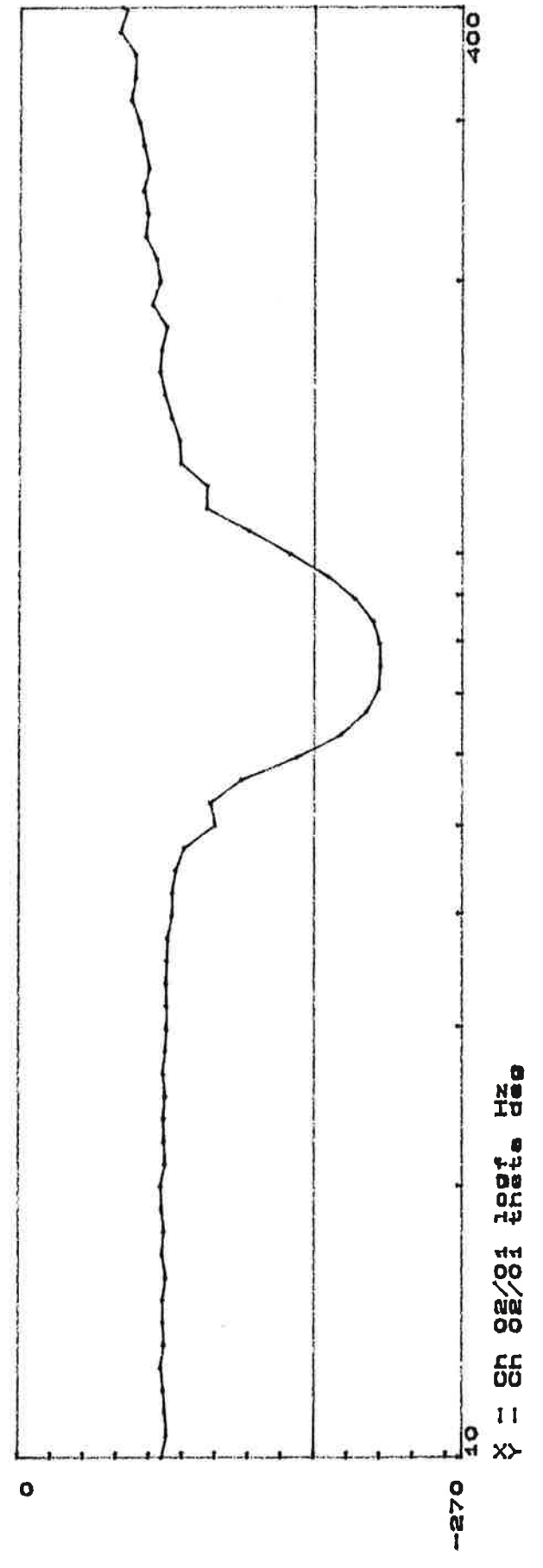
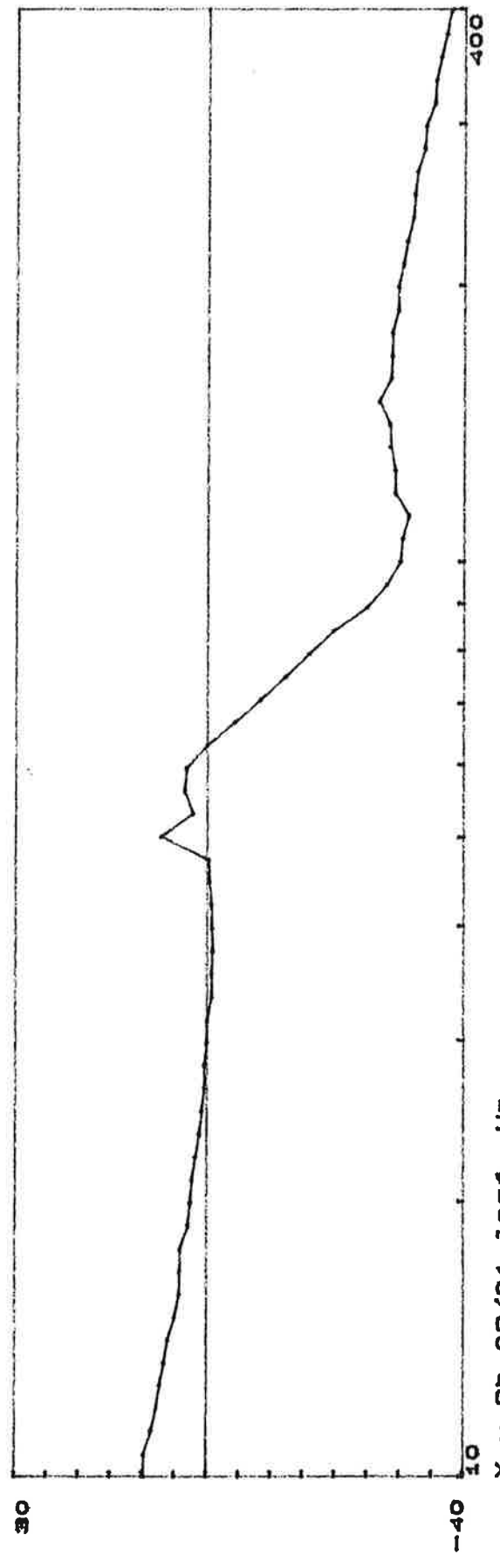
bilaga 1.3

2001



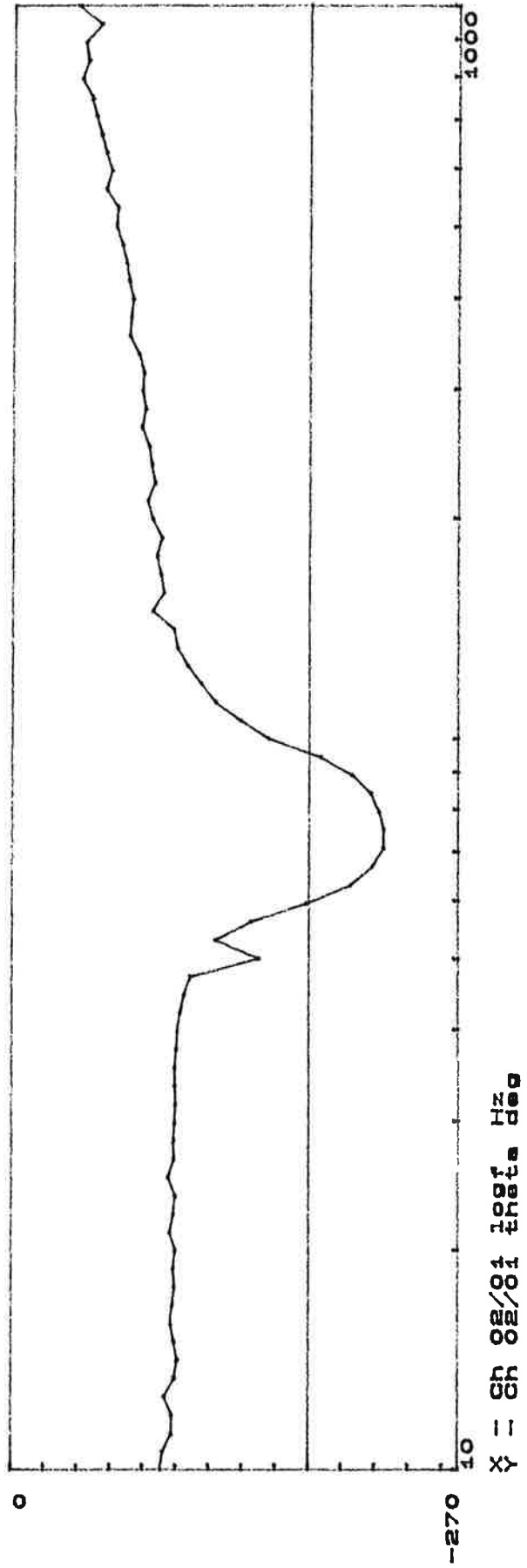
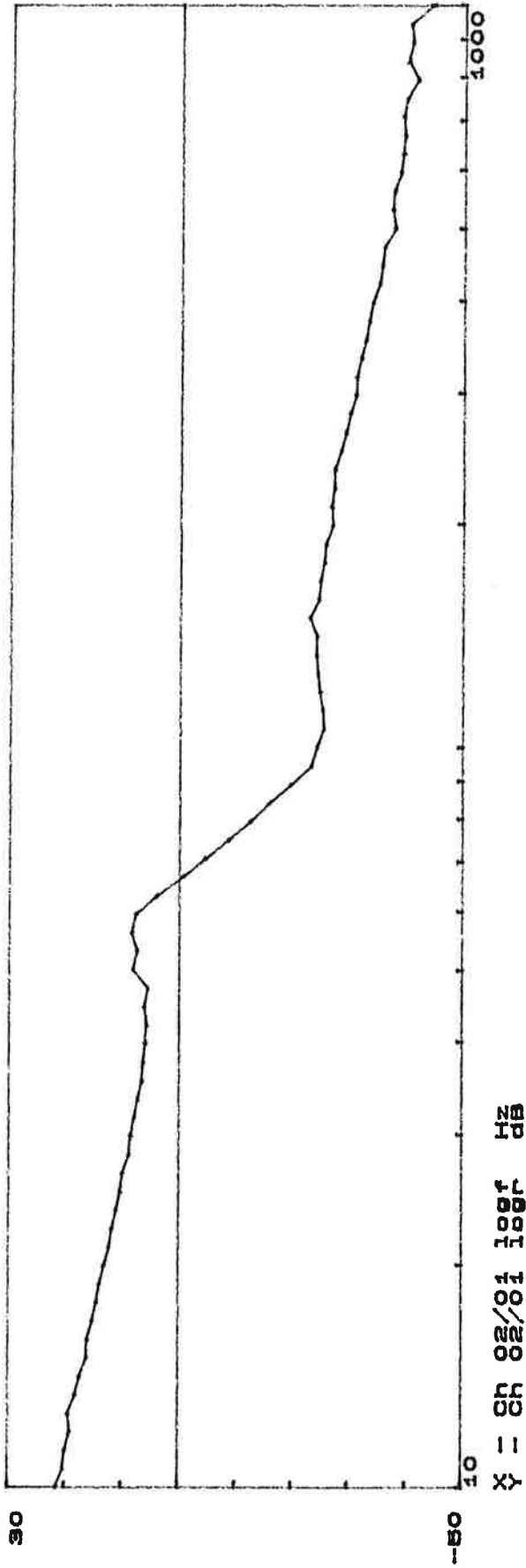
160 V

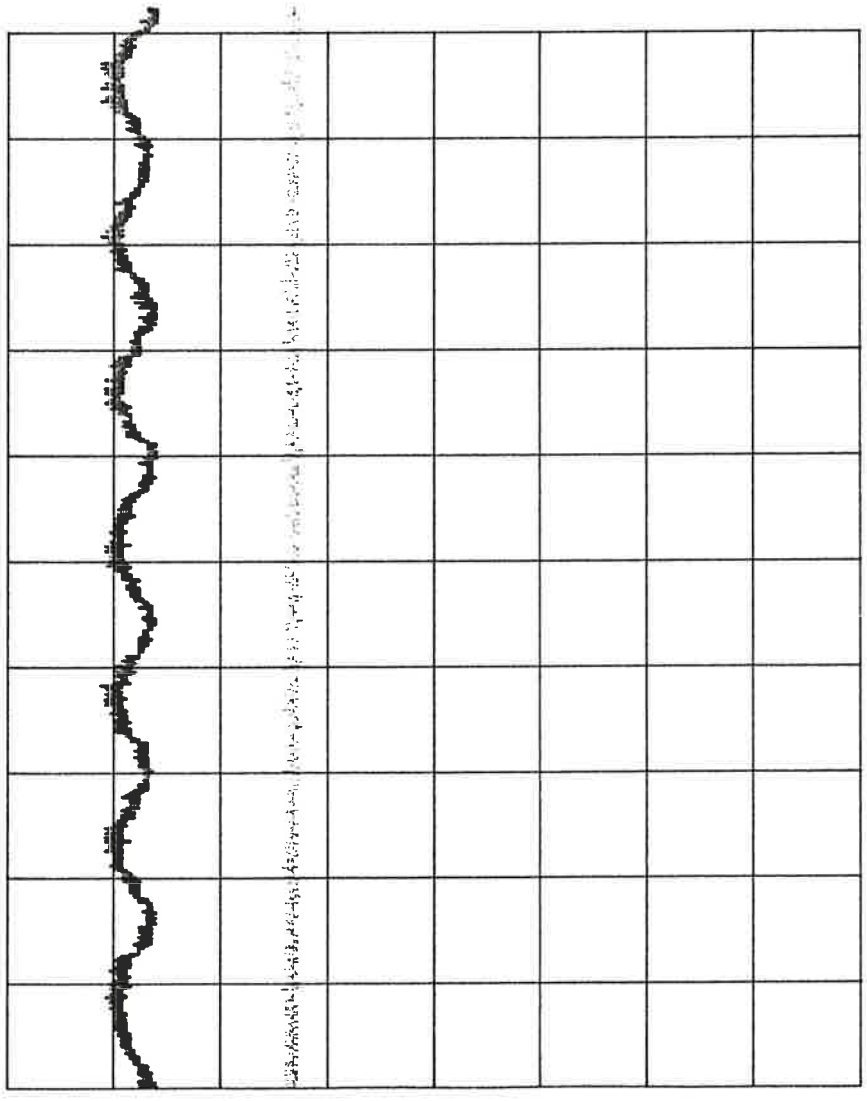
bilaga 1.4



bilaga 1.5

220





Bilag 2:

Styrsignalen Umult

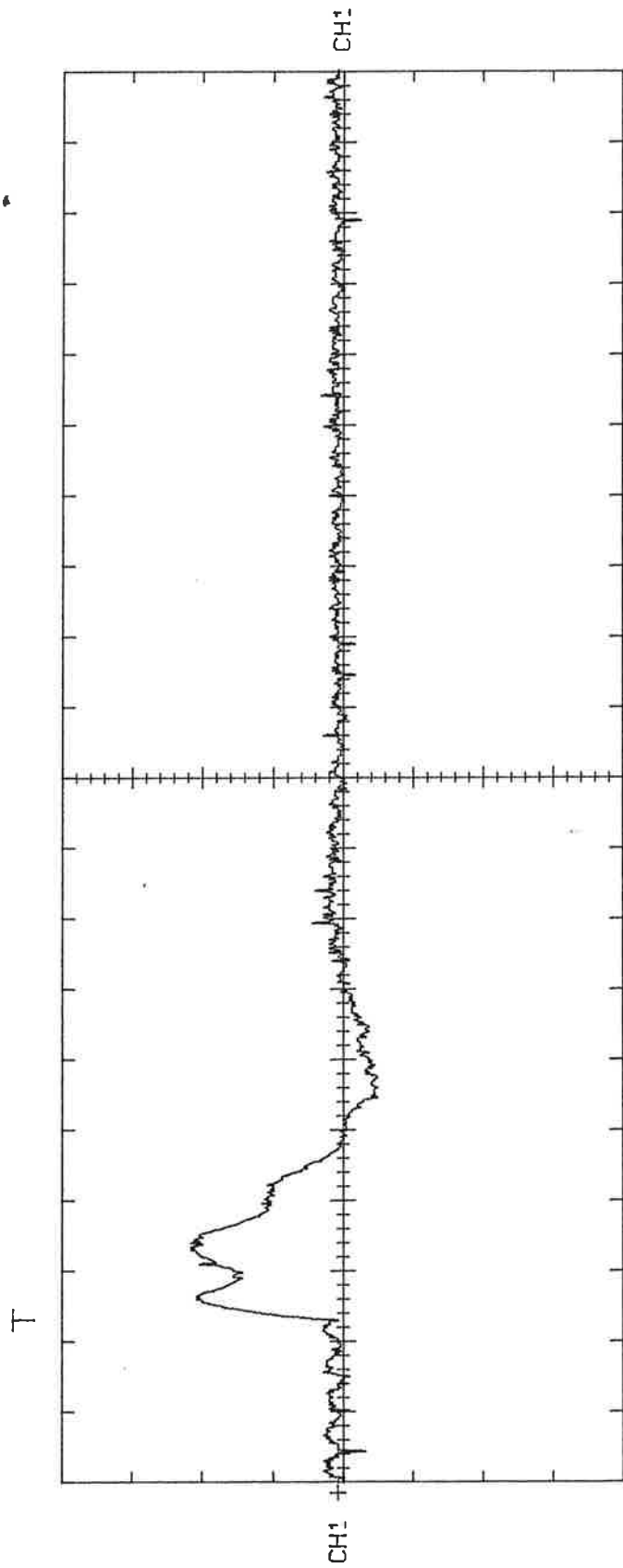
: Gammal PID-regulator

: Analog PI-regulator

CURSORS SETTINGS SHIFT TEXT_OFF

TEK/2430

CH1 AC 1 V /div NORMAL 20mSEC/div



Bilaga 3.1 :

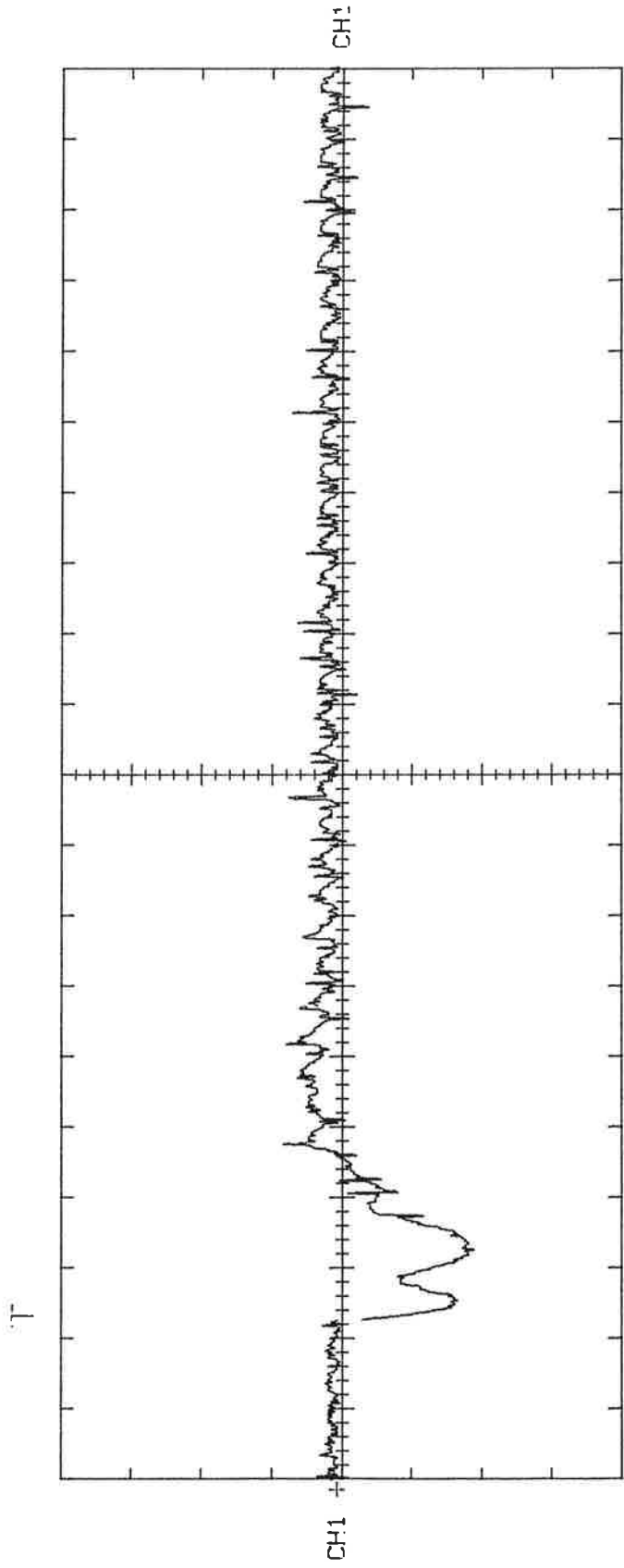
PI - regulator

Aulastning : 41,6A → 16A

C = 2,2 μF

TEK/2430

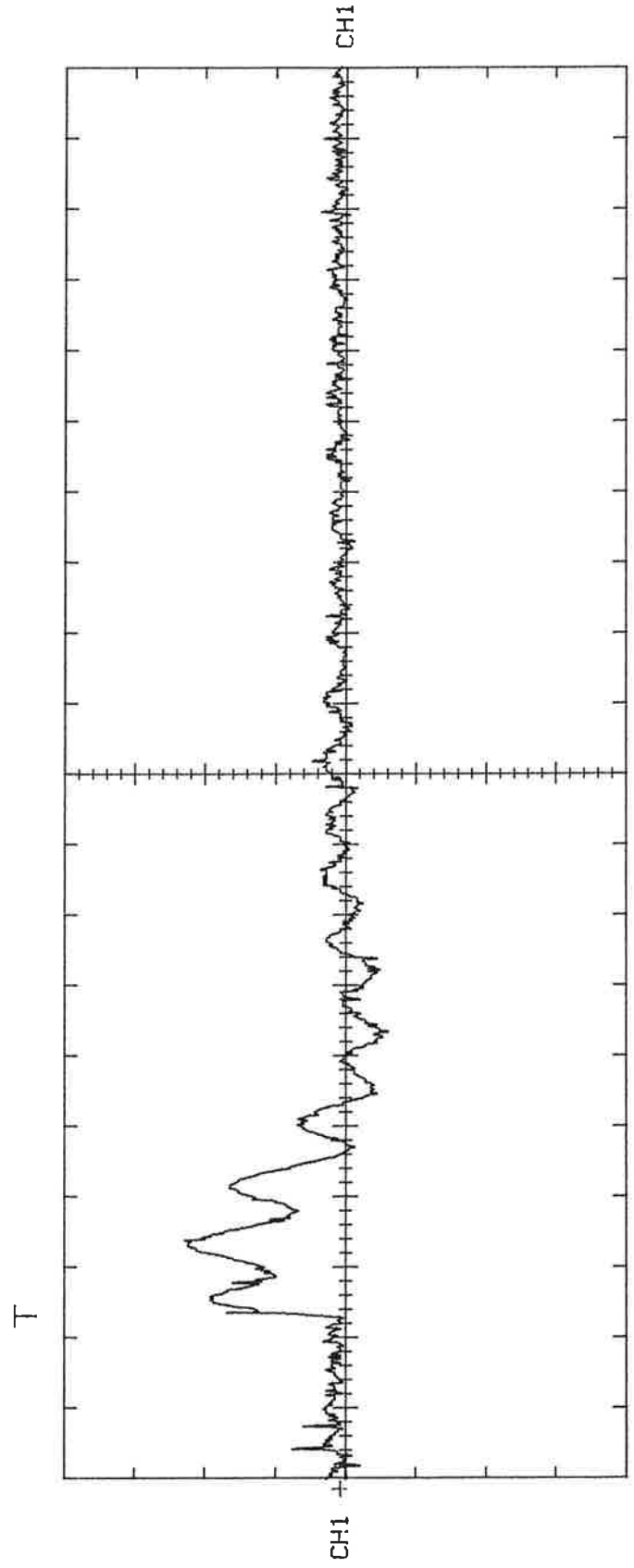
CH1 AC 1 V /div NORMAL 20mSEC/div



Bilaga 3.2 PI-regulator
Pålastning: 16A - 41,7A
C = 2,2 μ F

TEK/2430

CH1 AC 1 V /div NORMAL 20mSEC/div

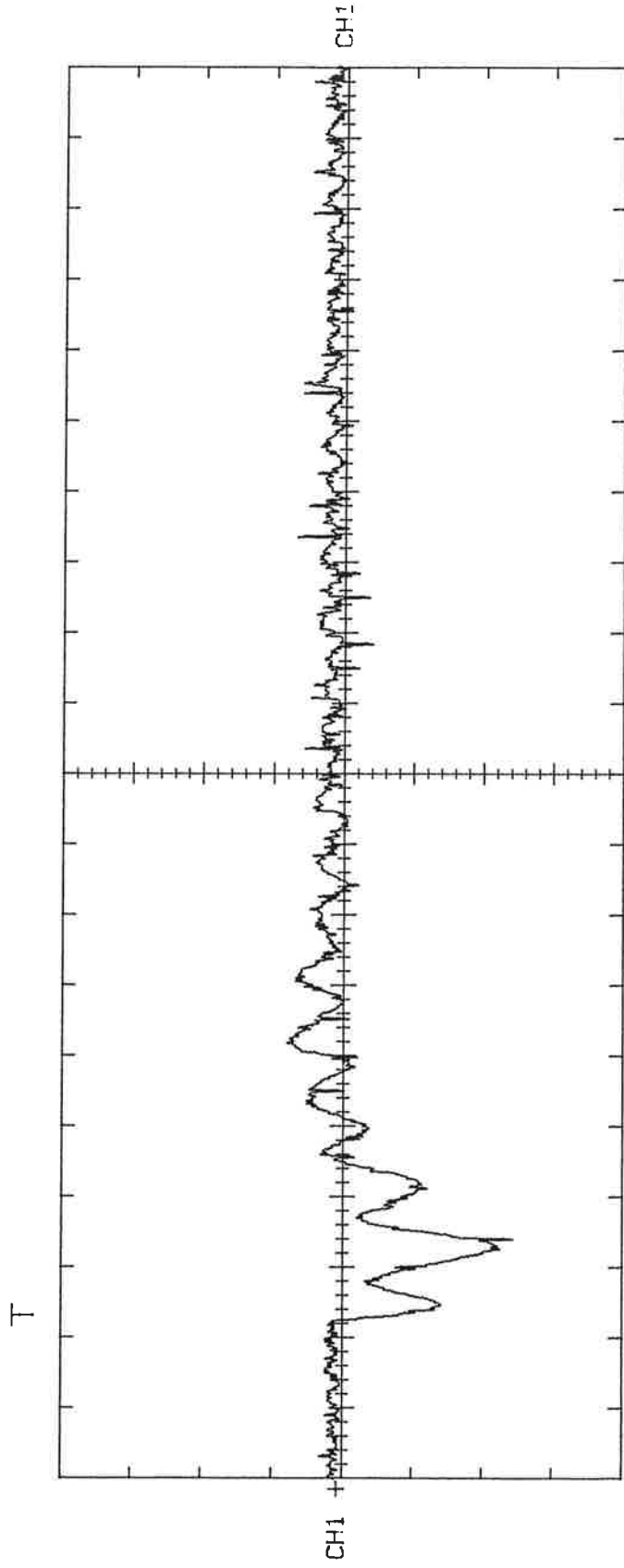


Bilaga 3.3:

PI-regulator
Aulastning: 41,6A → 16A
C = 100 nF

TEK/2430

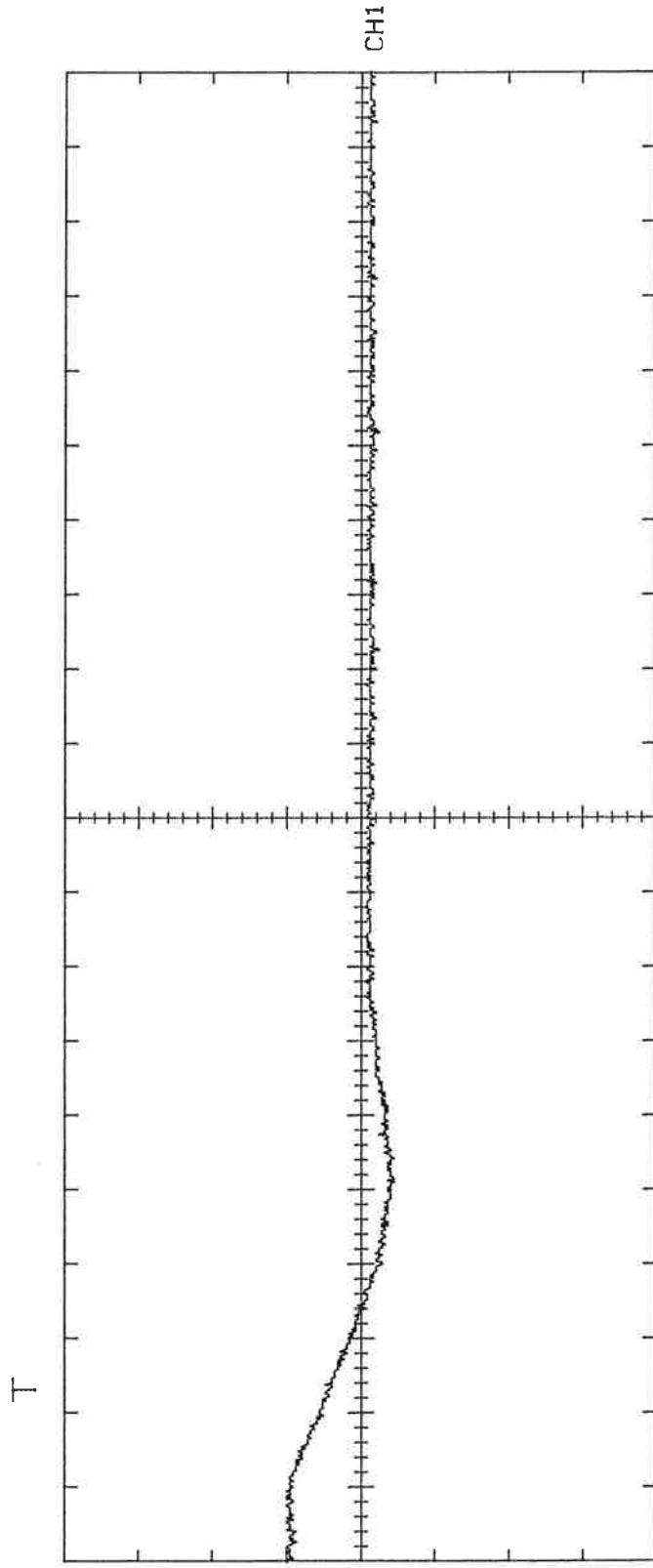
CH1 AC 1 V /div NORMAL 20mSEC/div



Bilag 3.4: PI-regulator
Pålastning: 16A → 41,6A
C = 100 nF

TEK/2430

CH1 DC 5 V /div 10mSEC/div NORMAL



CH1 +

Bilaga 4.1: PID-regulator

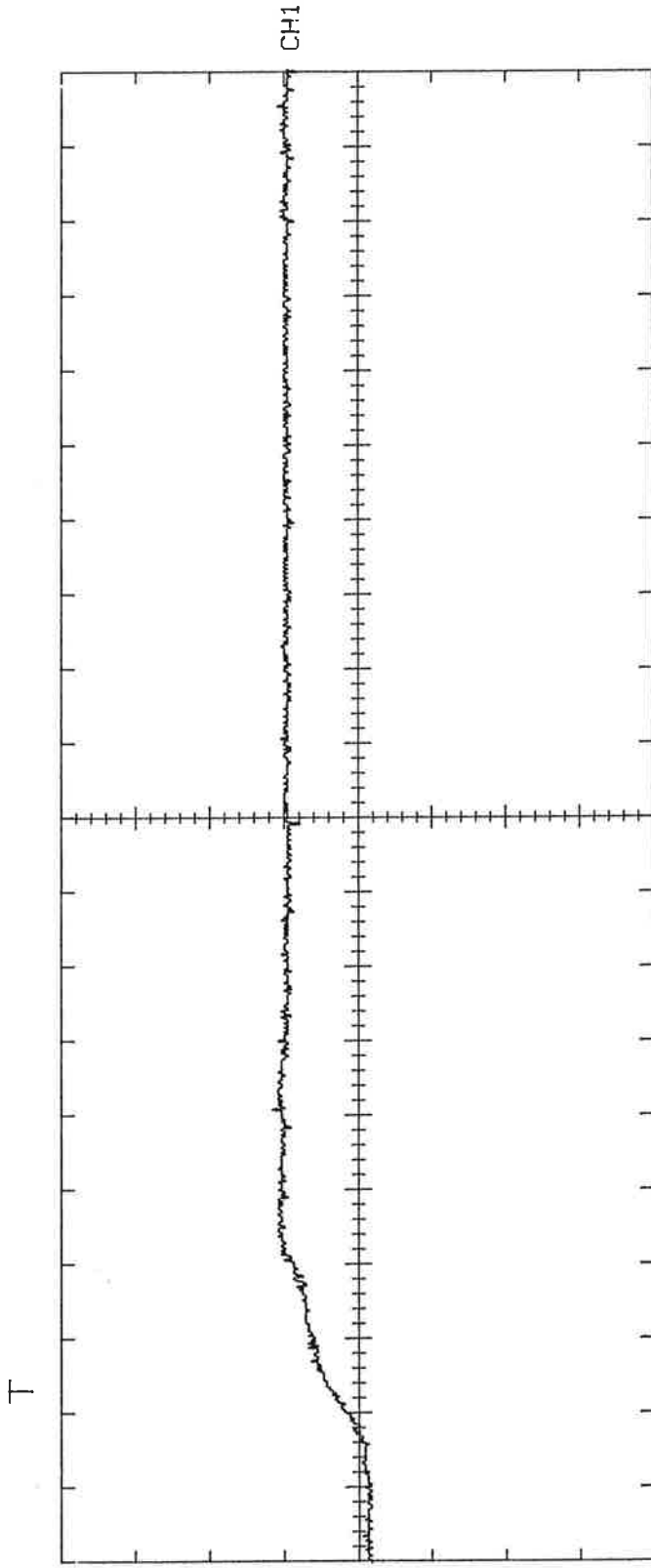
stegsvar

28,5 V → 23V

C = 2,2 μF

TEK/2430

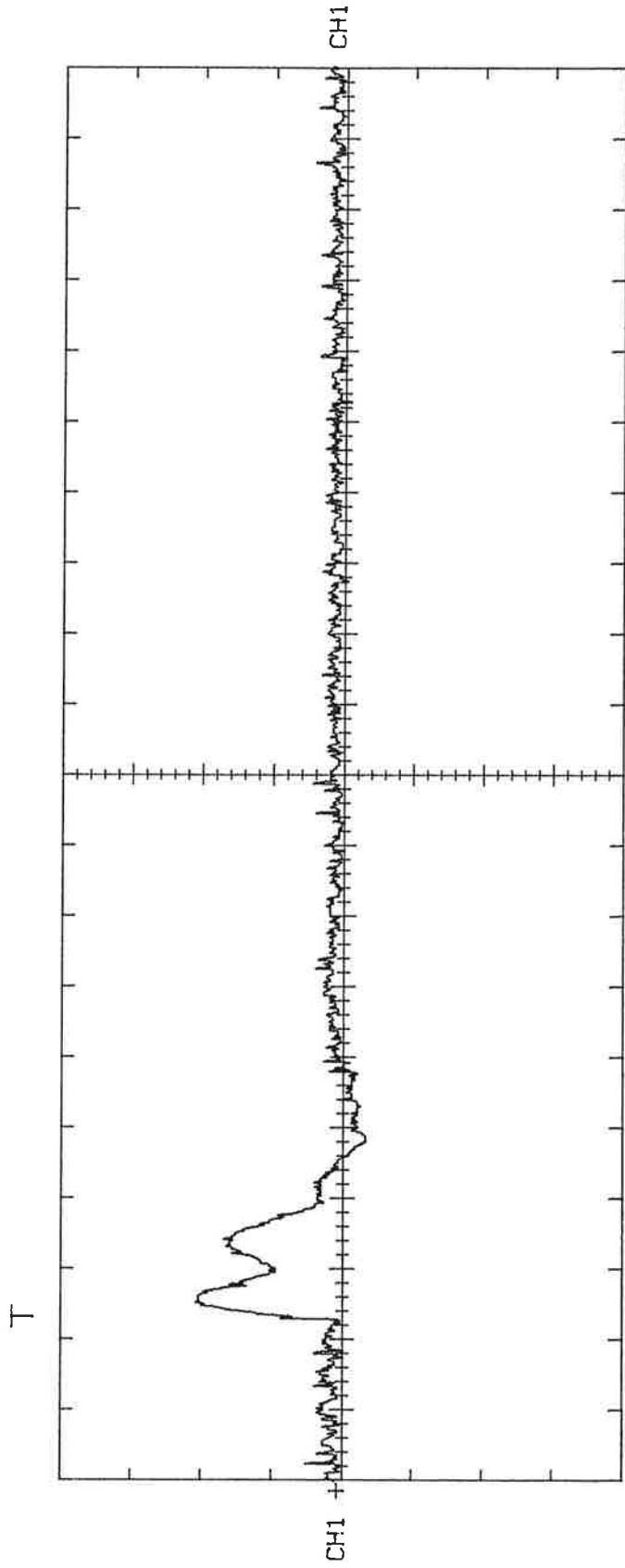
CH1 DC 5 V /div NORMAL 10mSEC/div



CH1 + Bilaga 4-2: PID-regulator 23 V → 28,5 V
Stegsvar C = 2,2 μF

TEK/2430

CH1 AC 1 V /div NORMAL 20mSEC/div



Bilaga 4.3:

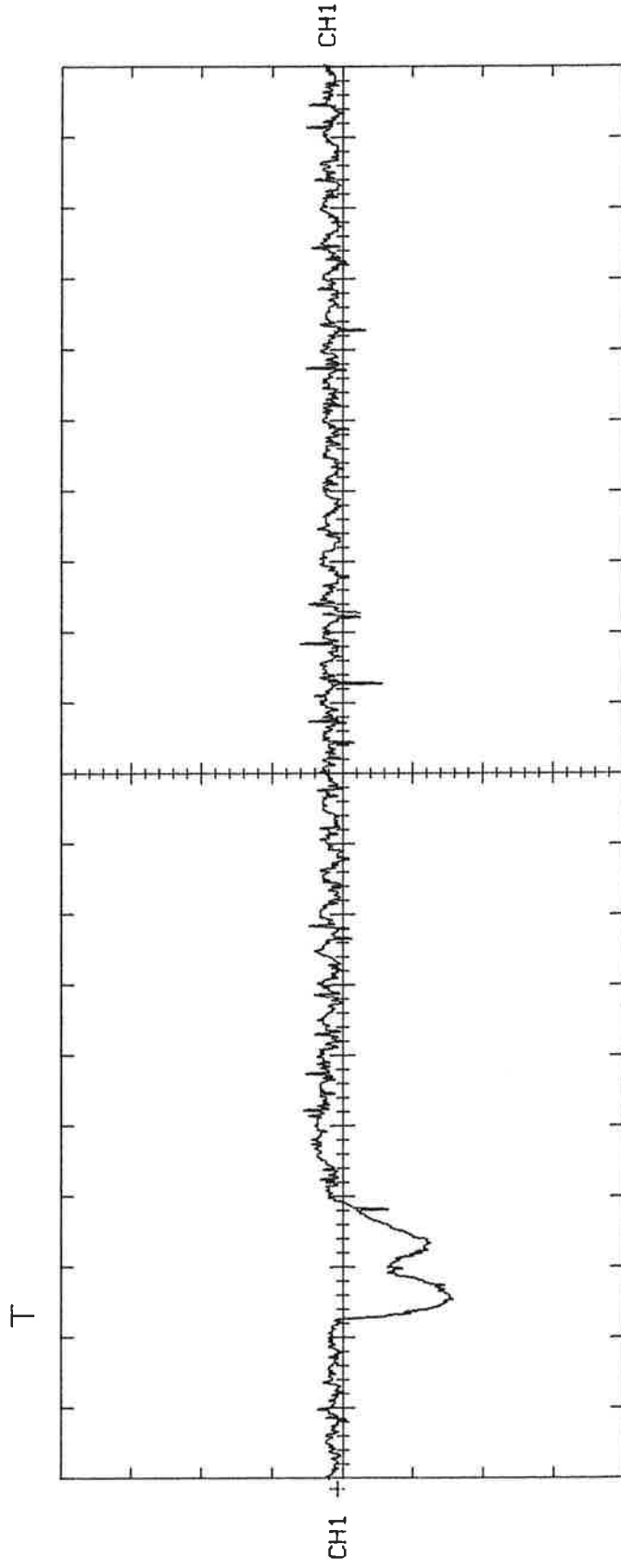
PID-regulator

Aulastning: 41,6 A → 16A

$C = 2,2 \mu\text{F}$

TEK/2430

CH1 AC 1 V /div NORMAL 20mSEC/div

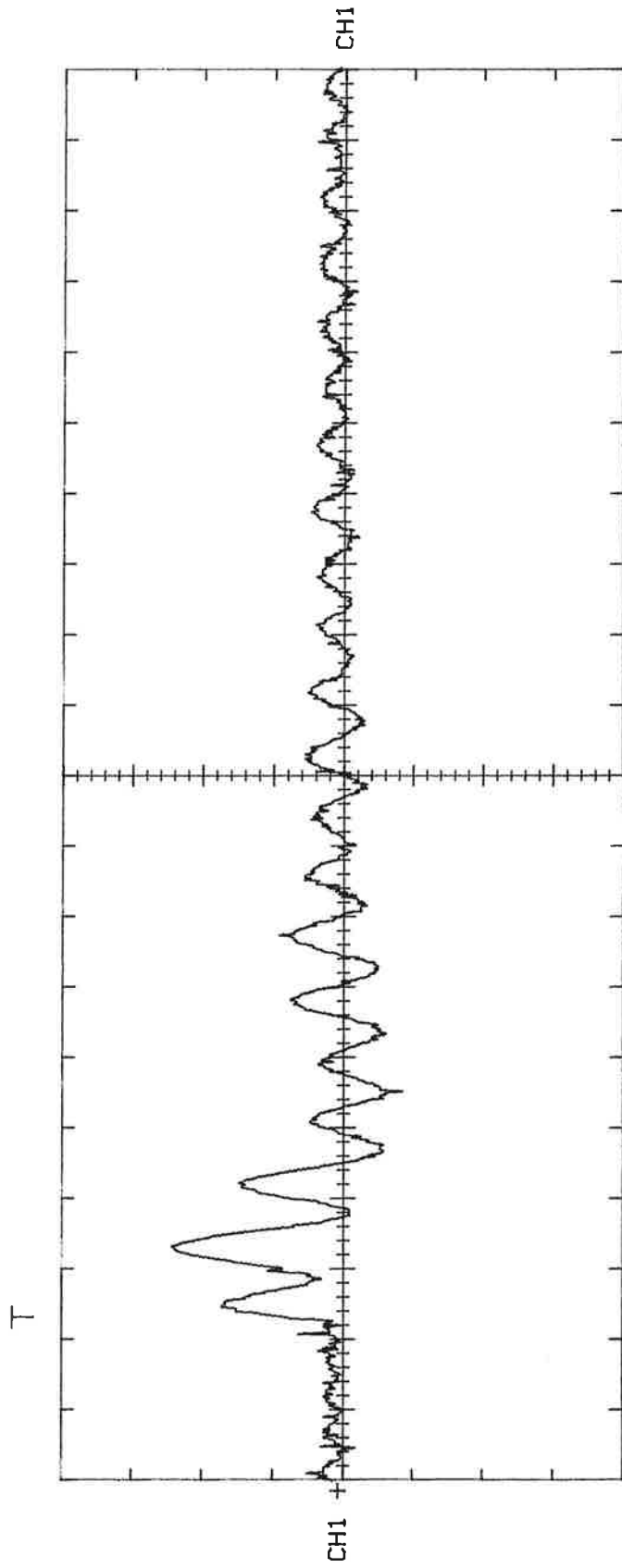


Bilaga 4.4:

PID-regulator
Pålastning: 16 A → 41,7 A
C = 2,2 nF

TEK/2430

CH1 AC 1 V /div NORMAL 20mSEC/div



Bilag 4.5:

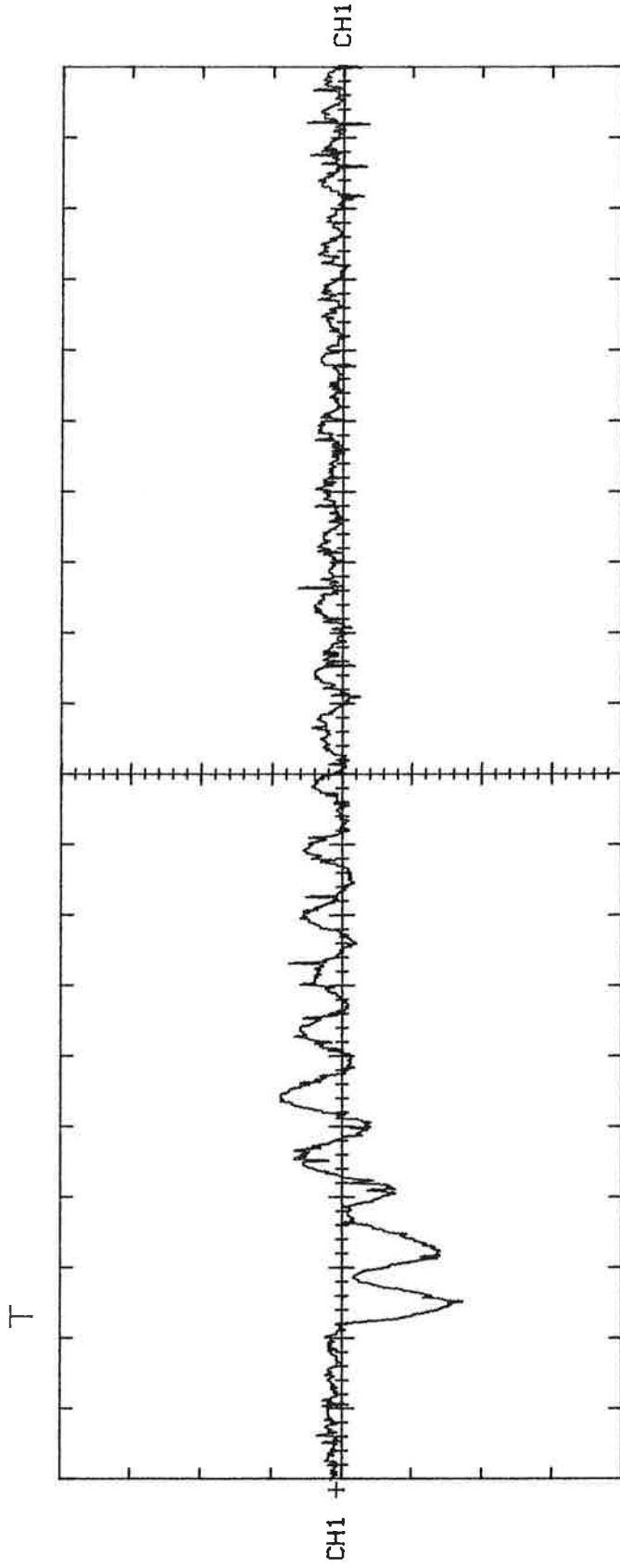
PID-regulator

Avlastning: 41,6 A → 16 A

C = 100 nF

TEK/2430

CH1 AC 1 V /div NORMAL 20mSEC/div



Bilaga 4.6:

PID-regulator
Pålastning: 16A → 41,6 A
 $C = 100 \text{ nF}$

```

; >>> Routine : Regul_V
; *****
; Regulate the rectifier output voltage
; *****
; P - REGULATOR
;
; PWM = (K1 * ERR) / 128
;
; K1 = K
; *****
; PI-regulator
;
; V_styr = O_V_styr + K5 * V_ref - K1 * U_meas - O_err + K2 * ERR
;
; PWM = 235 , V_styr/128 > 235
; V_styr/128 , 12 < V_styr/128 < 235
; 12 , V_styr/128 < 12
;
; K1 = K
; K2 = K * Ts/Ti
; K5 = K * b
;
; K = P-constant
; Ti = I-constant
; Ts = Sampleperiod
; b = V_ref-moderate-constant
;
; O_V_styr = Not trunked PWM-signal (prev iteration) (16-bit signed integer)
; PWM = PWM-signal (12 to 235)
; V_ref = Ref_voltage (0 to +1023)
; U_meas = Measured voltage (0 to +1023)
; ERR = V_ref - U_meas (-1023 to +1023)
; O_err = K5*V_ref - K1*U_meas (prev iteration) (32-bit signed integer)
; *****
; PID - REGULATOR
;
; V_styr = O_V_styr + K5 * V_ref - K1 * U_meas - O_err + K2 * ERR -
; - K4*(U_meas - 2*O_U_meas + OO_U_meas)
;
; PWM = 235 , V_styr/128 > 235
; V_styr/128 , 12 < V_styr/128 < 235
; 12 , V_styr/128 < 12
;
; K1 = K
; K2 = K * Ts/Ti
; K4 = K * Td/Ts
; K5 = K * b
;
; K = P-constant
; Ti = I-constant
; Td = D-constant
; Ts = Sampleperiod
; b = V_ref-moderate-constant
;
; PWM = PWM-signal (12 to 235)
; O_V_styr = Not trunked PWM-signal (prev iteration) (16-bit signed integer)
; V_ref = Ref_voltage (0 to +1023)

```

bilaga 5.2

```

; U_meas      = Measured Voltage                (0 to +1023)
; O_err       = K5*V_ref - K1*U_meas (prev iteration) (32-bit signed integer)
; ERR         = Ref_voltage - Measured Voltage   (-1023 to +1023)
; O_U_meas    = U_meas in previous iteration     ( 0 to +1023)
; OO_U_meas   = U_meas two iterations back      ( 0 to +1023)
;*****
Regul_V      JBS      PWM_flg,Stop_pos,Regul ; Branch if rect is not switched
             LDB      PWM_CONTROL,#NOT PWM_min ; Stop the rectifier
             CLR      O_U_err                ; Voltage diff in prev iteration
             CLR      O_err_1                ; Voltage diff in prev iteration
             CLR      O_err_2                ; Voltage diff in prev iteration
             CLR      O_V_styr               ; Voltage diff in prev iteration
             CLR      O_U_meas               ; Voltage in prev iteration
             CLR      OO_U_meas              ; Voltage two iterations back
             RET                               ; Exit
Regul        PUSH     W0                      ; Allocate temporary storage
             PUSH     W2                      ;
             PUSH     W3                      ;
             CMP      K2,#0000                ; If K2 is zero P-reg is selected
             JNE      Not_P_reg              ;
             LJMP     P_reg                  ;
Not_P_reg    PUSH     W0                      ; Store U_meas temporarily
             MUL      W0,K1                   ; Temp1 = U_meas * K1
             MUL      W2,V_ref,K5            ; Temp2 = V_ref * K5
             SUB      W2,W0                   ; Temp2 = Temp2 - Temp1
             SUBC     W3,W1                   ;
             LD       W0,O_err_1              ; Temp1 = O_err_1
             LD       W1,O_err_2              ; Temp1 = O_err_2
             LD       O_err_1,W2             ; O_err_1 prep for next iteration
             LD       O_err_2,W3             ; O_err_2 prep for next iteration
             SUB      W2,W0                   ; Temp2 = Temp2 - Temp1
             SUBC     W3,W1                   ;
             CLR      W1                      ;
             POP      W0                      ; Restore U_meas
             PUSH     W0                      ; Store U_meas
             SUB      W0,V_ref                ; Temp1 = U_meas - V_ref
             NEG      W0                      ; Temp1 = - Temp1
             MUL      W0,K2                   ; Temp1 = Temp1 * K2
             ADD      W0,W2                   ; Temp1 = Temp1 + Temp2
             ADDC     W1,W3                   ;
             POP      W2                      ;
             CMP      K4,#0000                ; If K4 is zero PI-reg is selected
             JE       Reg_com                 ;
             PUSH     O_U_meas                ;
             SHL      O_U_meas,#1            ; O_U_meas * 2
             SUB      OO_U_meas,O_U_meas     ; OO_U_meas - 2*O_U_meas
             LD       O_U_meas,W2            ; O_U_meas prepares for next iteration
             ADD      W2,OO_U_meas            ; Temp2 = U_meas + OO_U_meas - 2*O_U_meas
             MUL      W2,K4                   ; Temp1 = Temp1 * K4
             POP      OO_U_meas               ; OO_U_meas prep for next iteration
             SUB      W0,W2                   ; Temp1 = Temp1 - Temp2
             SUBC     W1,W3                   ;
Reg_com      ADD      W0,O_V_styr             ; Temp1 = Temp1 + O_PWM
             ADDC     W1,#0                   ; Temp1 = Temp1 + C
             CMP      W1,#0                   ;
             JLE      Not_big_1              ;
             LD       W0,#32767               ; Temp1 is set max value

```

bilaga 5.3

```

Not_big_1      SJMP   No_overfl_1      ;
               CMP    W1,#0           ;
               JGE    No_overfl_1      ;
               LD     W0,#-32768       ; Temp1 is set max negative value
No_overfl_1    CMP    W0,#30847       ; 30847 = PWM_max*128 + 127
               JLE    Not_big_2       ;
               LD     W0,#30847       ;
Not_big_2      CMP    W0,#2432        ; 2432 = PWM_min*128
               JGE    No_overfl_2      ;
               LD     W0,#2432        ;
No_overfl_2    LD     O_V_styr,W0     ; O_V_styr prep for next iteration
               SHRA   W0,#7           ; Temp1 = Temp1 / 128
               JNC    Round_down       ; Check if round down
               INC    W0              ; Round up positive W0
Round_down     SJMP   Set_PWM         ;
P_reg          SUB    W1,V_ref,W0     ;
               MUL   W0,W1,K1        ;
               SHRL  W0,#7           ;
               SJMP   Set_PWM         ;
Set_PWM        CMP    W0,#PWM_max     ; Check if result is over MAX
               JLE    Not_to_big      ; Branch if not
               LDB   W0,#PWM_max     ; Set to MAX
Not_to_big     SJMP   Not_to_small    ;
               CMP    W0,#PWM_min     ; Check if result is below MIN
               JGE    Not_to_small    ; Branch if not
               LDB   W0,#PWM_min     ; Set to MIN
Not_to_small   NOTB   W0             ; Invert all bits
               LDB   PWM_CONTROL,W0  ; Set PWM_control
Regul_exit    POP    W3              ;
               POP    W2              ;
               POP    W0              ;
               RET                    ; End of Regulator_V

```

```

% w is a vector (66x1) which contains frequency in rad/s.
% Gain from filter 1 to Umult is 2.9.
% G is a vector (66x1) which contains complex values of transfer function.

sl=[w 2.9*G];

% "sl" is now (66x2) vector and contains the experimental data.
% We use least square method to suit a model to sl by using lsbac.
% We try many combinations of nb, na, and weighting until the frequency response
% of the model is very close to that we got from the experiment.
% The frequency response "fresp" of the model can be calculated by using frc.
% To get more informations about any function in Matlab, use help command.

nb=3;
na=3;
weighting=[0.3 *ones(20,1);ones(26,1);0.003* ones(20,1)];
[bc,ac,er]=lsbac(sl,nb,na,weighting);
fresp=frc(bc,ac,0,w);

% We plot Bode diagram for both fresp and sl in one diagram.

bopl(sl);bosh(fresp,'-.');pause;

% We plot Nyquist diagram for both fresp and sl in one diagram.

nypl(sl,fresp,[],[-10 5 -10 5]);nygrid;

pause ; clg;

% We use tf2ss to convert transfer function to state space model.

[a,b,c,d]=tf2ss(bc,ac);

% We plot Bode diagram with direct term d and without it (d=0).

[mag1,phase1]=bode(a,b,c,d,1,w);
[mag2,phase2]=bode(a,b,c,0,1,w);

% Here we can see that we can neglect d.
% Now we convert the reduced model to transfer function form by using ss2tf.

[num,den]=ss2tf(a,b,c,0,1);

% calculate roots of bc, ac, num, den.

rootsbc=roots(bc)
rootsac=roots(ac)
rootsnum=roots(num)
rootsden=roots(den)

%Plot poles and zeros

clg; pzpl(bc,ac) ; pause;
clg; pzpl(num,den); pause;
clg

% We plot Bode diagram for experimental data, model with direct term and
% without it in one diagram.
subplot(211);
loglog(w,mag1,'-.',w,mag2,':',w,abs(sl(:,2)),'-')
subplot(212);
semilogx(w,phase1,'-.',w,phase2,':',w,arg(sl(:,2)),'-')

```

```
% Filter 1 has 3 poles (2 poles in pol and 1 in [600 2030000]).
% Filter 2 has 1 pole in [0.0022 1].
% The transfer function part from these filter is Bf/Af.

pol=[0.0003 1];
Af=conv(conv(conv(pol,pol),[600 2030000]),[0.0022 1]);
Bf=2e+6;

% Complete transfer function will be numfp/denfp, 2 zeros and 7 poles.

numfp=num*Bf;
denfp=conv(Af,den);

pzpl(numfp,denfp);pause;clg
% Gives figure 3.11

% If we neglect the fastest three poles, we gget numff/denff.

numff=num;
denff=conv([0.0022 1],den);

pzpl(numff,denff);pause;clg;
% gives figure 3.12

fr1=frc(numff,denff,0,w);
fr2=frc(numfp,denfp,0,w);
bopl(fr1,fr2),fr4,fr5);bogrid;
% Gives figure 3.13

bopl(fr2);bogrid;
% Gives figure 3.14
```

```
% h samplings period. Samplings frequency f=1/h.Choose f= 15 to 20 *fc,
% there fc is cross over frequency. fc=100 Hz.

h=1/2000;

% Sample

[bd,ad]=sample(numff,denff,h)

% ac is is the desierd denominator

ac =poly([exp(h*roots(polybutt(4,650,45)))')]);

% Observer polynom ao

ao=poly([exp(h*roots(polybutt(4,900,1)))')]);

% Regulator RST with one integrator

[r,s,t]=rstd(1,bd,ad,1,ac,ao,[1 -1]);

% Simulation

gp=frd(bd,ad,h,w);
gfb=frd(s,r,h,w);
gff=frd(t,r,h,w);
evpl(gp,gfb,gff,[1 4 -2 0;1 4 -2 0;1 4 -2 2;1 4 -2 2]);pause;clg
trldn = yusignal(0:h:0.2,1,0,-0.5,0.1,0,0.1,0.000001,0.15);
t1 = yusimd(bd,ad,r,s,t,trldn);
yupl(t1); pause ; clg;
```