

CODEN: LUTFD2/(TFRT-5358)/1-108/(1986)

Pulsbreddsmodulering av styrsignal till DC-motor

Jonas Andersson

Institutionen för Reglerteknik
Lunds Tekniska Högskola
November 1986

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> November 1986	
		<i>Document Number</i> CODEN: LUTFD2/(TFRT-5358)/1-108/(1986)	
<i>Author(s)</i> Jonas Andersson		<i>Supervisor</i> Rolf Johansson, Stellan Stenmark	
		<i>Sponsoring organisation</i>	
<i>Title and subtitle</i> Pulsbreddsmodulering av styrsignal till DC-motor). (Pulse width modulation of control input to DC motor).			
<i>Abstract</i> <p>The purpose of this work was to investigate if the choice of frequency, on a pulse width modulated control signal to a DC motor, might influence the behaviour of the system, as well as to see how the system reacts in connection with the controlling by means of such a signal.</p> <p>I came to the conclusion that the frequency was not critical seen from the system's point of view, but should be chosen so that the control algorithm just can manage to be run through in one period of the pulse width modulated signal. This would enable the control of each pulse. The power-transistor card did not perfectly transmit the pulse width modulated signal to the motor, which might be the reason why the number of revolutions did not increase linearly with the duty cycle of the out-signal.</p> <p>The system functioned in a good way when I P-controlled the motor speed. It was not notable that the control signal is pulse width modulated instead of analogous. The Motor Control Unit developed by AB Sandvik Electronics was very well prepared for the pulse width modulation.</p>			
<i>Key words</i> Pulse width modulation, DC motor.			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i>			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 108	<i>Recipient's notes</i>	
<i>Security classification</i>			

PULSBREDDSMODULERING AV STYRSIGNAL

TILL DC-MOTOR.

av

Jonas Andersson

Handledare: Stellan Stenmark, AB Sandvik Electronics
Rolf Johansson, Institutionen för
Reglerteknik vid Lunds
Tekniska Högskola

Examensarbete

SUMMERING

Examensarbetets uppgift var att undersöka om valet av frekvensen, på en pulsbreddsmodulerad styrsignal till en DC-motor, påverkade systemets uppförande samt hur systemet uppför sig vid reglering med en sådan styrsignal.

Jag fann att frekvensen inte var kritisk ur systemets synpunkt, utan bör väljas så att regleralgoritmen precis hinner genomlöpas under en period av den pulsbreddsmodulerade signalen. Detta för att reglering av varje puls skall kunna ske. Kraftkortet överförde inte den pulsbreddsmodulerade signalen perfekt till motorn, vilket kan vara anledningen till att varvtalet inte ökade linjärt med duty cyclen på utsignalen.

Systemet uppförde sig mycket bra då jag P-reglerade hastigheten. Att styrsignalen är pulsbreddsmodulerad istället för analog märktes inte. Motorstyrkortet, som Sandvik har tagit fram, var mycket bra förberett för pulsbreddsmoduleringen.

INNEHÅLLSFÖRTECKNING

<u>Kap</u>		<u>Sid</u>
	Summering.....	1
	Innehållsförteckning.....	2
	Inledning.....	4
1	Litteraturöversikt.....	6
2	Utrustning.....	10
3	Frekvensmätningar med pulsgenerator.....	13
3.1	Allmänt.....	13
3.2	Ström, spänning och duty cycle relaterat till varvtal.....	14
3.3	Övriga registreringar.....	16
3.4	Tabeller.....	18
3.5	Diagram.....	23
3.6	Slutsatser av frekvensmätningar.....	36
4	Tester med regulatorer.....	37
4.1	Allmänt.....	37
4.2	Öppen loop körning.....	38
4.2.1	Allmänt.....	38
4.2.2	Mätningar.....	39
4.2.3	Teoretiska beräkningar.....	41
4.2.4	Slutsatser.....	42
4.3	P-reglering av hastighet.....	43
4.3.1	Allmänt.....	43
4.3.2	System med förstärkning i återkopplingen.....	44
4.3.2.1	Mätningar.....	44
4.3.2.2	Teoretiska beräkningar.....	46
4.3.3	Förstärkningen efter återkopplingen.....	47

	<u>Sid</u>
4.3.3.1	Mätningar.....47
4.3.3.2	Teoretiska beräkningar.....48
4.3.4	Slutsatser.....50
4.4	Lägesåterkoppling.....50
4.4.1	Allmänt.....50
4.4.2	P- och PI-reglering.....51
4.4.3	Sammanfattning.....53
4.5	Börvärdesstyrning med P-reglerad hastighet.....54
4.5.1	Allmänt.....54
4.5.2	Tester för lämplig förstärkning.....55
4.5.3	Bestämmande av maximal acceleration.....57
4.5.4	Lägsta hastighet vid reglering och reglerfelet.60
4.5.5	Test av olika frekvenser.....66
4.5.6	Slutsatser.....68
4.6	Slutsatser av regleringarna i kapitel 4.....68
5	Mätningar på switchdel.....70
5.1	Allmänt.....70
5.2	Mätningar.....70
5.3	Slutsatser.....75
6	Sammanfattning.....77
	Referenser.....81
Appendix	
A. Flödesschema för regulatorprogram	
B. Taco-generatorn och störningar	
C. MCU-kort	
D. Kraftkort	
E. Mätinstrument och mätmetoder	
F. Data på DC-motor, taco- och pulsgivare	

INLEDNING

Under sommaren 1986 har jag utfört mitt examensarbete vid AB Sandvik Electronics. Uppgiften var att ta reda på om frekvensen hos en pulsbreddsmodulerad styrsignal till en DC-motor påverkar egenskaperna i systemet. I resten av rapporten betecknar jag pulsbreddsmodulering med PWM, efter engelskans Puls Width Modulation. En andra uppgift var att undersöka om PWM inverkar på resultatet vid reglering.

Arbetet började med en litteratursökning, vars resultat finns i kapitel 1 och i referenserna. I rapporten framgår att jobbet mycket är av experimentell natur. För att få ett svar på första uppgiften ovan har jag gjort frekvensmätningar där PWM-signalen kom från en pulsgenerator. Dessa finns redovisade i kapitel 3. Samtidigt undersökte jag också switchdelen för att kunna ge en bedömning av denna. Undersökningarna återfinns i kapitel 5.

Vid de fortsatta mätningarna använde jag AB Sandvik Electronics eget framtagna motorstyrkort. Med detta utförde jag några olika digitala regleringar. Vad regleringarna gav återfinns i kapitel 4. Jag har där även gjort några teoretiska beräkningar för att se om mätningarna överensstämmer med en enkel teoretisk modell. Varje kapitel börjar med en allmän genomgång av mätningarna, och avslutas med slutsatser. Jag har dessutom gjort en sammanfattning av alla mätningar och resultat i kapitel 6.

Hur den totala utrustningen ser ut finns i kapitel 2. Beskrivningen är där schematiskt. Noggrannare beskrivningar återfinns i appendix längst

bak i rapporten. Här har jag även lagt programmen som användes vid regleringen. Dessa är i rapporten utförda som flödesschema.

1 LITTERATURÖVERSIKT

För att sätta mig in i vad PWM är och vad som tidigare gjorts inom området har jag gjort en litteratursökning. Denna gjorde jag med hjälp av Sandvik AB:s bibliotek. Vi sökte i en databas vid namn INSPEC. Som huvudsökord använde jag "DC motors". Som undersökord användes "Pulse Width Modulation", "Pulse Time Modulation" och "Pulse Frequency Modulation". De artiklar som framkom har jag tagit med bland referenserna.

I (1) görs en modell över en motor med en överföringsfunktion i diskret tid. För att modellen skall stämma antas att pulsperioden för PWM-signalen är mycket mindre än motorns tidskonstant. Regleringen är implementerad med en microprocessor och en programmerbar timer. För att få fram tidskonstanter tas ett stegsvar. Man simulerar stegsvaret med en första och en andra ordningens modell och kvadrerar avvikelserna mot stegsvaret man har mätt upp.

Det andra ordningens system gav en så liten förbättring att de inte tyckte att det försvarade den ökade komplexiteten som hade uppstått vid regleringen. Man kommer dessutom fram till en undre gräns för samplingsperioden genom att regleralgoritmen måste hinna genomlöpas mellan varje sampling.

Även referens (3) tar upp teori. Här genomgås ett generellt n :te ordningens system. Här räknas dock i kontinuerlig tid.

På elektriska rullstolar används PWM för att batteridriften skall ge så lång transportsträcka som möjligt mellan laddningarna. I (2) jämförs lågfrek-

vent (400 Hz) PWM, med och utan filter, med högfrekvent (20 kHz). Den högfrekventa visade sig vara effektivare och gav dessutom mindre oljud. Undersökningarna kompletteras med teoretiska beräkningar, där strömmen serieutvecklas för att bestämma virvelströms- och hysteresefförluster.

En p-reglerad seriemotor undersöks i (4). Man kommer fram till att i PWM med naturlig sampling av seriemotorer uppstår ofta fenomen av typ subharmonisk och pseudo-random oscillation.

(5) är min gamla lärobok i reglerteknik, som jag har haft som stöd vid utvecklingen av de olika regulatorerna jag använt.

De fem första referenser jag här har tagit upp är de som satte mig in i PWM och de som jag använt under arbetets gång.

Litteratursökningen gav dock ett flertal ytterligare referenser. Jag tar med även dessa bland referenserna för att underlätta ett fortsatt arbete med PWM.

Några av dessa referenser tar upp olika regleralgoritmer. (6) analyserar yttre och inre reglerloopar av olika komplexitet. En positionsregleralgoritm som är baserad på tidsoptimal reglering med användande av Pontryagins minimiprincip går att läsa om i (7). Man kommer här fram till att hastighetsmätningen kan göras känsligare om den externa pulsfrekvensen är hastighetsberoende.

I (8) används en PI-regulator som kör med max ut-signal och integralreset då börvärdet överstiger ett maxvärde. Adaptiv reglering genomgås i (9).

Lindningsisolationens livslängd och statorns kopparförluster får en teoretisk genomgång i (10). Enligt författarna är kopparförlusterna bl a beroende av lindningsdesignen.

Bak inducerade emk:n används i (11) - i stället för avkänning av positionen - för regleringen. I (14) används också en annorlunda metod att reglera, nämligen PLL realiserad med en micro-processor.

Den enda artikeln med pulsfrekvensmodulering är (13).

Många artiklar beskriver kretsar och kopplingar som är gjorda för PWM. (12) använder återkopplade inverterare för att generera pulsfrekvens. Styrningen sker dock här så att motorn polas antingen positivt eller negativt. Det innebär att vid stillastående polas den lika mycket åt båda hållen, vilket medför att den har den största energiförbrukningen då. I (17) genomgås en krets som fungerar enligt 2-kvadrantprincipen. (18) går igenom olika halvledarkretsar för pulsstyrning, vilket även (19) gör.

IBM beskriver en ny krets i (20) och ett kontrollsystem i (21).

(15) tar upp styrning enligt 4-kvadrantprincipen. Man använder här också regenerativ bromsning.

Allmänt om servos kan man läsa i (16), och allmänt om PWM återfinns i (22).

Som ni kanske har märkt har genomgången av de sista referenserna varit mycket kursiv, och detta beror på den ringa betydelse dessa har haft för mitt arbete. Jag har ändå velat ta med dessa för att orientera läsaren om vad min litteratursökning har givit för resultat.

För att kunna mäta hur en DC-motor uppför sig vid pulsstyrning hade jag en uppställning enligt nedan, Figur 2.1. Om data och detaljer i utrustningen kan man läsa i appendix.

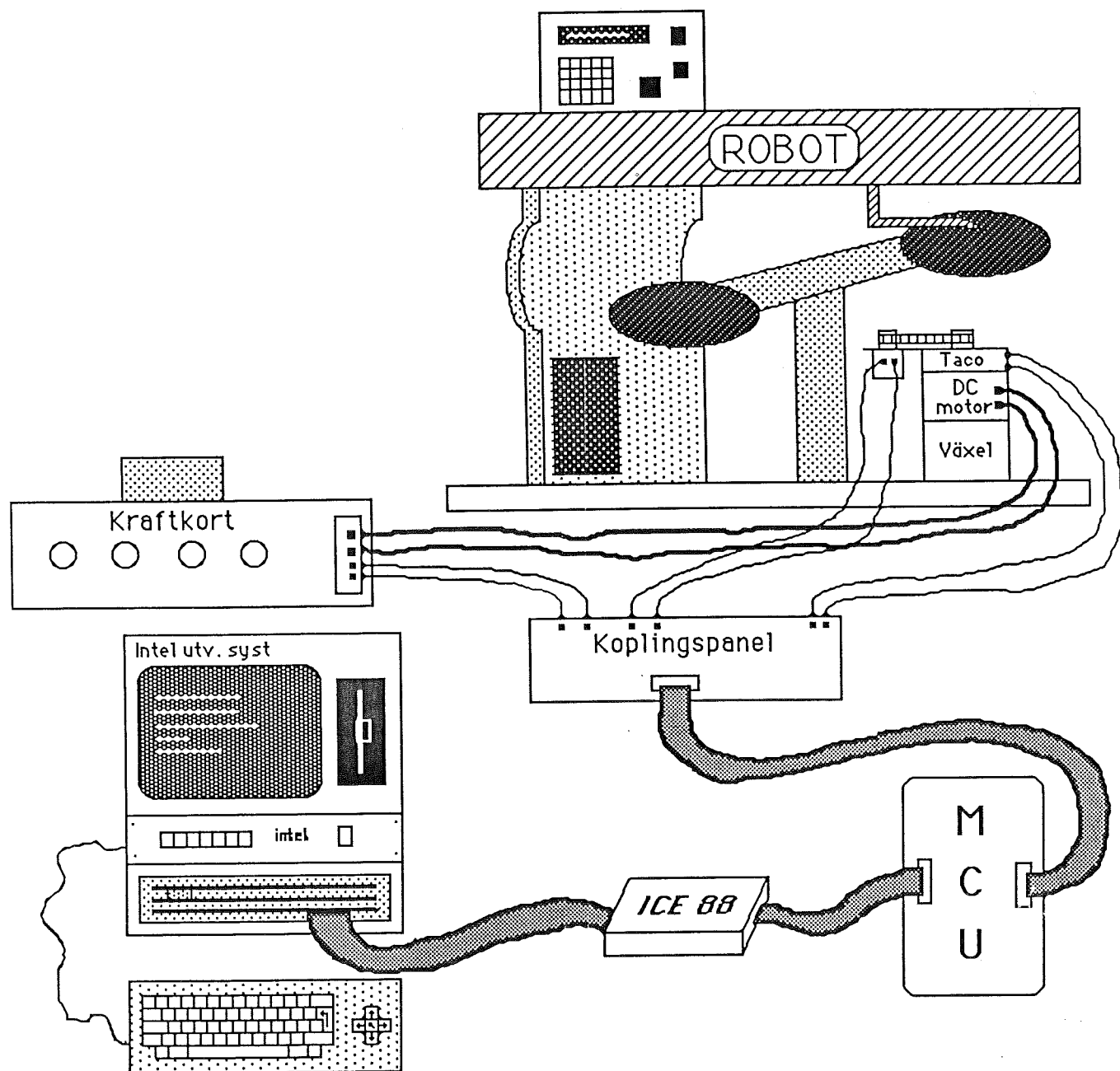


Fig. 2.1. Schematisk bild över den totala utrustningen där mätningarna gjordes.

För att få en verklig situation använde jag en DC-motor som var monterad i en robot. Detta medför att motorn utsätts för en friktions- och tröghetsmomentsbelastning. Motorn var så monterad att den via växlar och kuggremmar drev två plan att cirkulera för att distribuera föremål på roboten. Direkt på motorn var en tacogenerator monterad. Denna använde jag för hastighetsmätning av motorns rotation. Dessutom var en pulsgivare kopplad via kuggrem till motorn för att mäta läge.

Det jag kallar för kraftkort innehåller transistor-switcharna som driver motorn. Där återfinns också elektroniken som översätter MCU-kortets TTL-nivåer till styrsignaler för transistorerna. I stället för MCU-kort kan här anslutas en pulsgenerator direkt, vilket jag gjort i kapitel 3 för att enkla- re variera olika parametrar vid öppen loop-styrning.

MCU-kortet (Motor Control Unit, utvecklat vid Sandvik Electronics), som jag använde vid styrning med hjälp av regulatorer, kopplas via en kopplingspanel till kraftkortet. Genom denna panel var även taco och pulsgivare kopplade till MCU:n. Här fanns möjlighet att använda optokopplare. MCU:ns huvudkomponenter, så som jag använde den, var processorn och en intervalltimer som genererade PWM-signalen.

I vanliga fall styrs MCU-kortet via en huvuddator, men för att förenkla handhavandet använde jag en ICE-88 (In-Circuit Emulator) som ersatte mikroprocessorn på kortet. Via Intels utvecklingssystem genom ICE:n gjorde jag programmering och all körning.

I utrustningen ingick dessutom spänningsaggregat.
Mätinstrumenten är inte heller utritade, utan
dessa och mätmetoderna har sitt eget appendix.

3 FREKVENSMÄTNINGAR

3.1 Allmänt

I arbetet ingår bl a att hitta en lämplig puls-frekvens. Litteratursökningen gav ingen formel för detta, men ref. (2, Inigo) gav mig en viss ledning i vilket område jag bör leta i. Detta var från några hundra Hz till några tio-tal kHz. Ett krav från Sandvik var dock att man ville reglera minst en gång var millisekund. Därför kunde mätningarna begränsas neråt till 1 kHz. Vid dessa mätningar använder jag en pulsgenerator för att generera pulserna. Mer om detta och hur mätinstrumenten kopplades återfinns i Appendix E. Pulsgeneratoren kopplades direkt till kraftkortet (beskrivning på kraftkortet finns i Appendix D), dvs ingen återkoppling användes. När man sen använder MCU-kortet vid styrning, krävs en viss tid av kortet för att genomlöpa en algoritm. Denna begränsar frekvenserna uppåt. Jag mätte på frekvenser upp till 40 kHz.

Jag har mätt duty cyclen på signalen till kraftkortet, strömmen genom motorn och spänningen över den. Dessa är relaterade till varvtalet på motorn. Utredning om dessa mätningar finns i kap 3.2. Mät-datan är tabellerad i kap 3.4. I kap 3.5 har jag fört in dessa i diagram för att lättare kunna utläsa resultaten. Dessutom har en del andra mätningar och registreringar utförts, vilka är kommenterade i kap 3.3. För jämförande studier har motsvarande mätningar också gjorts vid analog körning. Vid dessa mätningar kopplade jag ett likspänningsaggregat direkt över motorns poler.

3.2 Ström, spänning och duty cycle relaterat till varvtal

Varvtal

Några av uppgifterna med detta arbete har varit att se vilken dynamik systemet får med pulsbreddsmodulering, dvs ta reda på största och minsta hastighet och vari begränsningen ligger. Största hastighet låg vid 3500 rpm. Pulsstyrningen begränsade inte utan det var amplituden på pulserna som gjorde det, dvs den likspänning som kunde fås över motorns poler med maximal duty cycle. Den maximala hastigheten påverkades inte av pulsfrekvensen. Eftersom jag körde med öppen styrning gick motorn ojämnt vid de lägsta hastigheterna, vilket den också gjorde vid motsvarande analoga körning. För att kunna bestämma en lägsta hastighet satte jag upp ett kriterium. Lägsta hastighet definierades som den där pendlingen i hastigheten, topp till topp, var mindre än 10 % av medelvärdet. Detta gav en hastighet runt 400 rpm. Inte heller lägsta hastigheten påverkades av pulsfrekvensen.

En annan viktig detalj att undersöka var ojämn gång. Att ojämn gång skulle ha uppstått pga pulsstyrningen kan inte påstås eftersom motorn uppförde sig likadant vid alla frekvenser som vid analog körning. Vid 40 kHz gick dock motorn något ojämnt ända upp till 1000 rpm, men klarade 10 % kriteriet ändå vid 400 rpm precis som de andra frekvenserna. Vid 5, 10, 20 och 31.25 kHz började dock hastigheten pendla när jag lade på 40 % duty cycle. Ripplet, som var 2-3 % av medelvärdet på hastigheten, låg på ca 5 Hz. Någon anledning till dessa har jag ej kunnat finna. Där uppstod inga problem när jag styrde med regulatorer, se kap 4.

Ström genom motorn

Eftersom jag mäter den ström som spänningsaggregatet levererar så kommer även strömmen som kraftkortet förbrukar med. Då jag lade på 0 % duty cycle var viloströmmen dock endast 76 mA.

Strömmätningarna är inritade i diagram i kap 3.5, fig 3.9-16. Man kan se att en viss ström, från 0.8 till 1.4 A, behövs för att motorn skall börja rotera. Detta pga friktionen hos motor och kraftöverföring. Det var den högsta frekvensen som krävde den lägsta startströmmen, medan den analoga körningen krävde den största. Strömmen ökar sen linjärt från start, pga friktionen. Strömmen ökar minst vid analog körning och är vid 3000 rpm 2.1 A där. Strömmen vid 3000 rpm ökar med ökande puls-frekvens till 2.4 A för 40 kHz. Strömmens linjäritet från start till 3000 rpm minskar med ökande frekvens.

Spänning över motorn och duty cyclen

Ur diagrammen i kap 3.5, fig 3.1-8, kan vi se att en viss spänning behövs för att motorn skall börja röra sig. Detta är precis som för strömmen beroende på friktionen i systemet. För övrigt ökar spänningen helt linjärt med ökande varvtal. Detta är precis som det bör vara med en permanentmagnetmotor. Spänning ligger dock något lägre relativt varvtalet vid pulsstyrning än vid analog styrning. Vid 40 kHz ca 0.5 V och vid 1 kHz ca 1 V över hela varvtalsområdet.

Duty cyclen, se fig. 3.17-3.23 i kap 3.5, var inte linjär mot spänningen och därmed ej heller mot varvtalet. En möjlig anledning till detta var kraftkortet. Just detta utreds i kap 5. Olinjäri-

teten är mest utmärkande vid låga frekvenser och minskar för ökande frekvens. Den kraftiga olinjäriteten vid de låga frekvenserna kan dock vara till en fördel vid viss reglering. Vill man ha små exakta rörelser omväxlande med snabba förflyttningar ger olinjäriteten en högre förstärkning vid de små sakta rörelserna och lägre vid förflyttningarna. Och vid förflyttningar är noggrann reglering oftast inte så viktig varför den lägre förstärkningen där inte är ett bekymmer. Allt detta är naturligtvis under förutsättning att regleralgoritmen har konstant förstärkning. Något mer man kan lägga märke till är att det krävs högre duty cycle för att motorn skall rotera om man använder högre frekvens. Även detta går jag igenom i kap 5.

3.3 Övriga registreringar

Förlustvärme

Förutom de mätningar jag redovisade i kap 3.2 har jag gjort vissa andra som kan vara av betydelse vid införandet av PWM-styrning. I det förra kapitlet gick jag igenom ström och spänning, men det skulle vara intressantare att se hur dessa samverkar, dvs effekten. Pga att jag inte hade några instrument att mäta vare sig ineffekt eller effekt ut på axel använde jag min hand för att mäta temperaturen på motorn. Eftersom jag inte bestämde absolut temperatur på motorn utan jämförde vid PWM- och analogstyrning minskas felet i mätningen. Någon märkbar uppvärmning pga PWM-styrningen förekom ej, utan motorn var precis som för analog körning ljummen hela tiden.

Oljud

Vid start och låga varvtal förekom ett pip när frekvenserna 1, 2, 5 och 10 kHz användes. Vid de högre frekvenserna kunde man naturligtvis inte höra något. Men om man bestämmer sig för en hög frekvens bör ljudnivåmätning göras pga att icke hörbara ljud kan orsaka hörselnedsättningar. De ljud som hördes, alltså vid de lägre frekvenserna, dränktes av ljudet från kraftöverföringen när varvtalet ökades. Därför kan man påstå att pulsstyrningen inte har tillfört oljud som är högre än systemet hade innan. Används dock systemet mest vid låga hastigheter inverkar dock att PWM-styrsignalen tillför en renton som är mer irriterande än oljudet från kraftöverföringen.

Taco

PWM-styrningen medförde en del störningar på taco-signalen. Dessa gav dock inte upphov till några problem vid reglermätningarna. Mer om detta finns i appendix B.

3.4 Tabeller

Här följer mätresultaten. Tabell 3.1 kommer från den analoga körningen och sen följer mätningarna efter ökande frekvens. I är strömmen genom motorn och U är spänningen över den. Pulslängd är den tid spänningen är till var period. Detta ger duty cyclen i procent som $100 \cdot (\text{pulslängd}/\text{periodtid})$. n_u är den spänning som tacon genererar och n är översättningen till varv per minut. I kap 3.2 redogör jag för att mätningar har gjorts för att bestämma en lägsta hastighet på systemet med ett 10 % kriterium. De lägsta hastigheter som klarade detta är inramade med streckade linjer.

Puls- längd/ ms	Duty cycle/ %	I/ A	U/ V	n_u / V	$n=n_u \frac{1000}{6}$ rpm
—	—	1,49	4,00	1,20	200
—	—	1,50	5,00	2,18	363
—	—	1,53	6,00	3,02	503
—	—	1,57	7,00	3,95	658
—	—	1,62	8,00	4,87	812
—	—	1,65	9,00	5,80	967
—	—	1,68	10,00	6,70	1117
—	—	1,74	11,00	7,64	1273
—	—	1,77	12,00	8,64	1440
—	—	1,81	13,00	9,60	1600
—	—	1,84	14,00	10,44	1740
—	—	1,85	15,00	11,50	1917
—	—	1,89	16,00	12,40	2067
—	—	1,92	17,00	13,30	2217
—	—	1,97	18,00	14,20	2367
—	—	2,01	19,00	15,18	2530
—	—	2,03	20,00	16,00	2667
—	—	2,08	21,00	16,95	2825
—	—	2,13	22,00	17,90	2983
—	—	2,15	23,00	18,90	3150
—	—	2,20	24,00	19,75	3292

Tab. 3.1. Mätdata från analog körning.

Puls- längd/ ms	Duty cycle/ %	I/ A	U/ V	n_u / V	$n=n_u \frac{1000}{6}$ rpm
0,10	10	1,21	2,30	1,0	167
0,12	12	1,36	3,95	2,4	400
0,15	15	1,62	9,61	7,7	1283
0,20	20	1,91	14,47	12,4	2067
0,30	30	2,08	17,24	14,8	2467
0,40	40	2,19	19,32	16,5	2750
0,50	50	2,27	21,19	18,2	3033
0,60	60	2,32	22,36	19,1	3183
0,70	70	2,36	23,30	20,0	3333
0,80	80	2,40	24,02	20,5	3417
0,90	90	2,42	24,66	21,0	3500
0,98	98	2,44	25,07	21,1	3517

Tab. 3.2. Mätdata från körning med PWM-signal.
Pulsfrekvens = 1kHz.

Puls- längd/ μs	Duty cycle/ %	I/ A	U/ V	n_u / V	$n = n_u \frac{1000}{6}$ rpm
62	12,4	1,14	2,85	1,2	200
68	13,6	1,24	4,60	2,5	417
100	20	1,65	11,45	9,0	1500
150	30	1,99	16,90	14,1	2350
200	40	2,11	18,54	15,3	2550
250	50	2,24	20,45	17,5	2917
300	60	2,30	21,86	18,7	3117
350	70	2,36	22,90	19,5	3250
400	80	2,40	23,73	20,1	3350
450	90	2,44	24,42	20,7	3450
490	98	2,45	24,72	20,9	3483

Tab. 3.3. Mätdata från körning med PWM-signal.
Pulsfrekvens = 2kHz.

Puls- längd/ μs	Duty cycle/ %	I/ A	U/ V	n_u / V	$n = n_u \frac{1000}{6}$ rpm
34	17	1,10	4,80	2,5	417
40	20	1,29	7,00	5,0	833
60	30	1,72	14,59	12,3	2050
80	40	1,98	16,80	14,4	2400
100	50	2,07	18,71	15,7	2617
120	60	2,20	20,71	17,7	2950
140	70	2,25	22,00	18,7	3117
160	80	2,32	23,09	19,6	3267
180	90	2,35	24,03	20,4	3400

Tab. 3.4. Mätdata från körning med PWM-signal.
Pulsfrekvens = 5kHz.

Puls- längd/ μs	Duty cycle/ %	I/ A	U/ V	n_u / V	$n=n_u \frac{1000}{6}$ rpm
20	20	1,01	3,50	1,7	283
20,5	20,5	1,03	4,25	2,4	400
30	30	1,46	11,19	9,0	1500
40	40	1,70	15,20	12,6	2100
50	50	1,92	16,34	13,6	2267
60	60	2,08	18,82	15,7	2617
70	70	2,19	20,66	17,4	2900
80	80	2,26	22,16	18,8	3133
90	90	2,31	22,89	19,2	3200

Tab. 3.5. Mätdata från körning med PWM-signal.
Pulsfrekvens = 10 kHz.

Puls- längd/ μs	Duty cycle/ %	I/ A	U/ V	n_u / V	$n=n_u \frac{1000}{6}$ rpm
11,5	23	0,95	3,63	1,6	267
12	24	0,97	4,44	2,5	417
15	30	1,21	8,25	6,0	1000
20	40	1,50	12,20	9,7	1617
25	50	1,76	13,72	11,0	1833
30	60	1,95	16,69	13,6	2267
35	70	2,10	18,97	15,6	2600
40	80	2,22	20,92	17,4	2900
45	90	2,36	22,88	19,1	3183
49	98	2,46	24,67	20,6	3433

Tab. 3.6. Mätdata från körning med PWM-signal.
Pulsfrekvens = 20 kHz.

Puls- längd/ μs	Duty cycle/%	I/ A	U/ V	n_u / V	$n=n_u \frac{1000}{6}$ rpm
8,0	25	0,92	4,47	2,4	400
9,6	30	1,10	6,82	4,5	750
12,8	40	1,40	9,75	7,3	1217
16,0	50	1,67	11,67	8,9	1483
19,2	60	1,95	15,22	12,2	2033
22,4	70	2,10	17,60	14,4	2400
25,6	80	2,27	20,06	16,5	2750
28,8	90	2,43	22,65	18,8	3133
31,0	96,9	2,55	24,85	20,9	3483

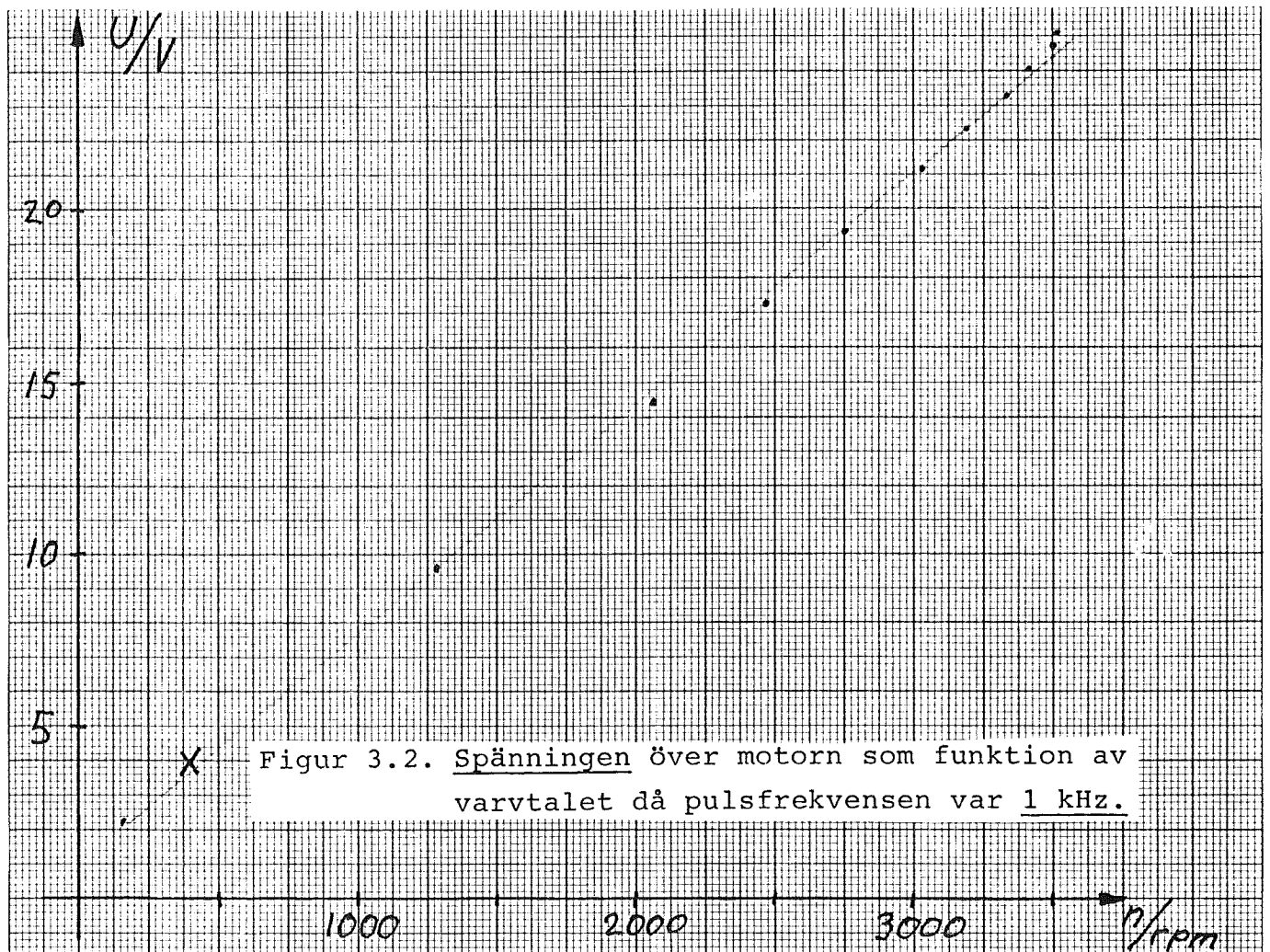
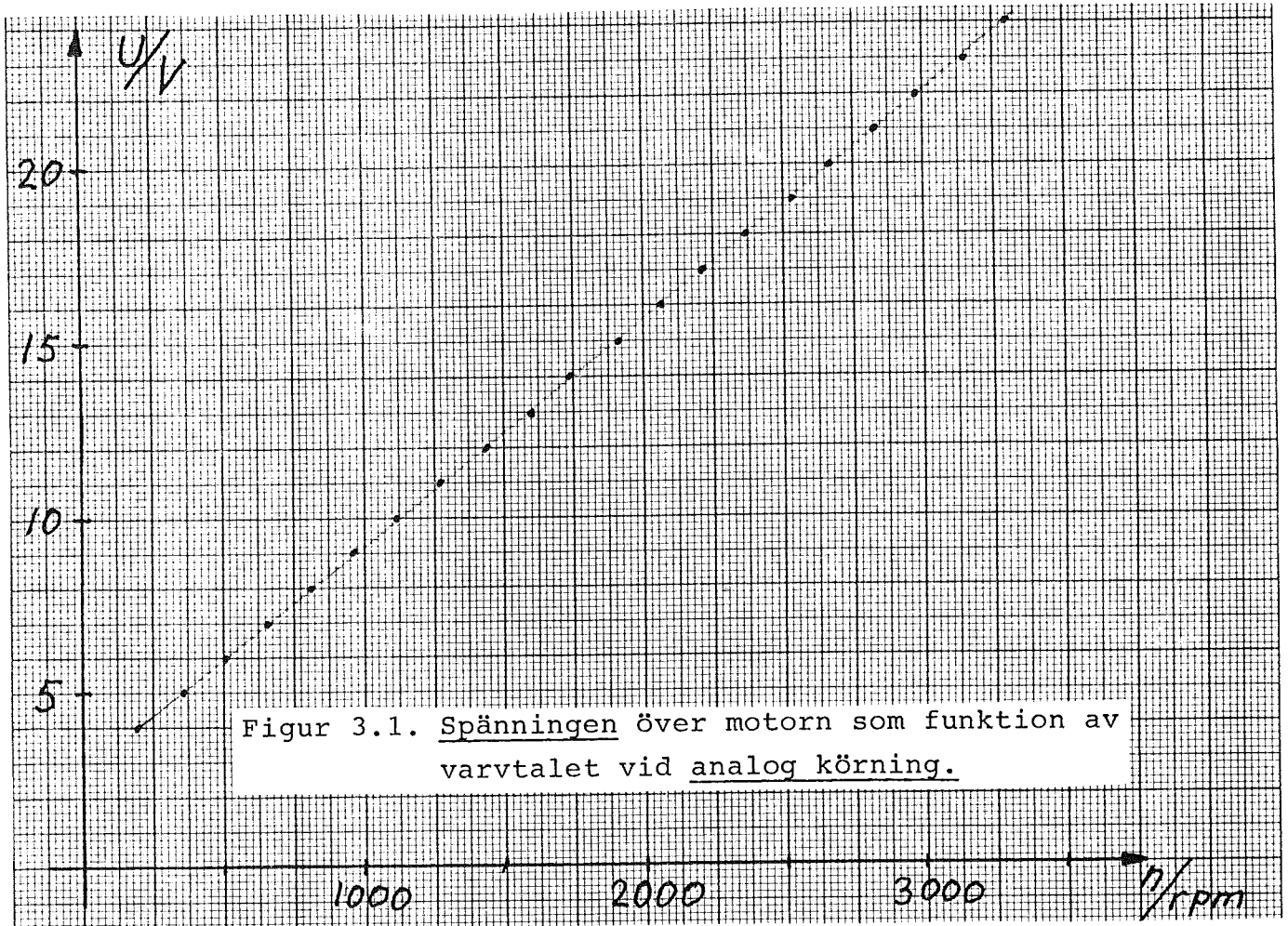
Tab. 3.7. Mätdata från körning med PWM-signal.
Pulsfrekvens = 31,25 kHz.

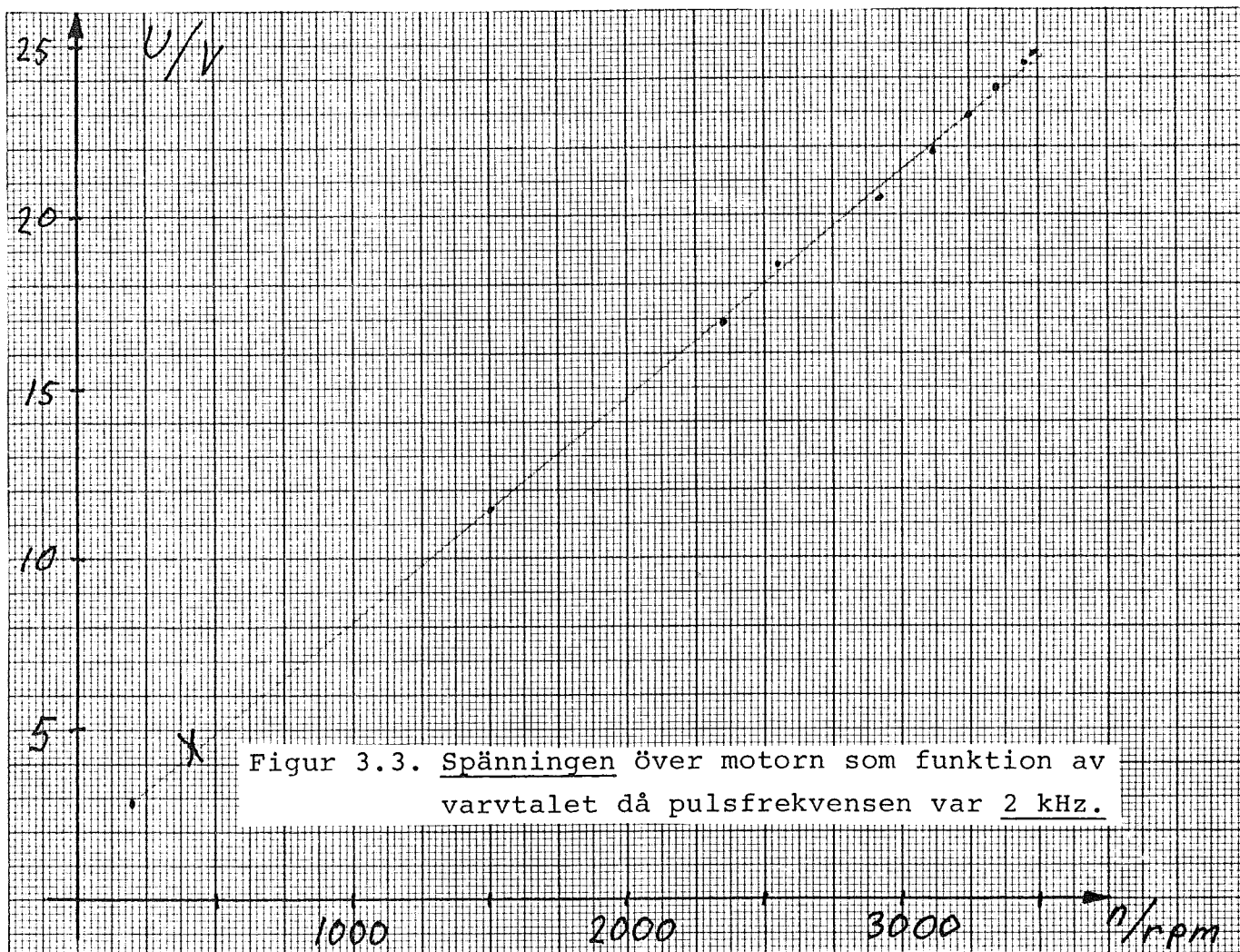
Puls- längd/ μs	Duty cycle/%	I/ A	U/ V	n_u / V	$n=n_u \frac{1000}{6}$ rpm
6,0	24	0,94	3,42	1,1	183
6,2	24,8	0,93	4,55	2,4	400
7,5	30	1,10	6,36	3,8	633
10,0	40	1,37	8,74	6,0	1000
12,5	50	1,64	10,79	7,9	1317
15,0	60	1,93	14,20	11,2	1867
17,5	70	2,13	16,98	13,8	2300
20,0	80	2,28	19,68	16,1	2683
22,5	90	2,44	22,84	19,0	3167
24,0	96	2,54	24,81	20,9	3483

Tab. 3.8. Mätdata från körning med PWM-signal.
Pulsfrekvens = 40 kHz.

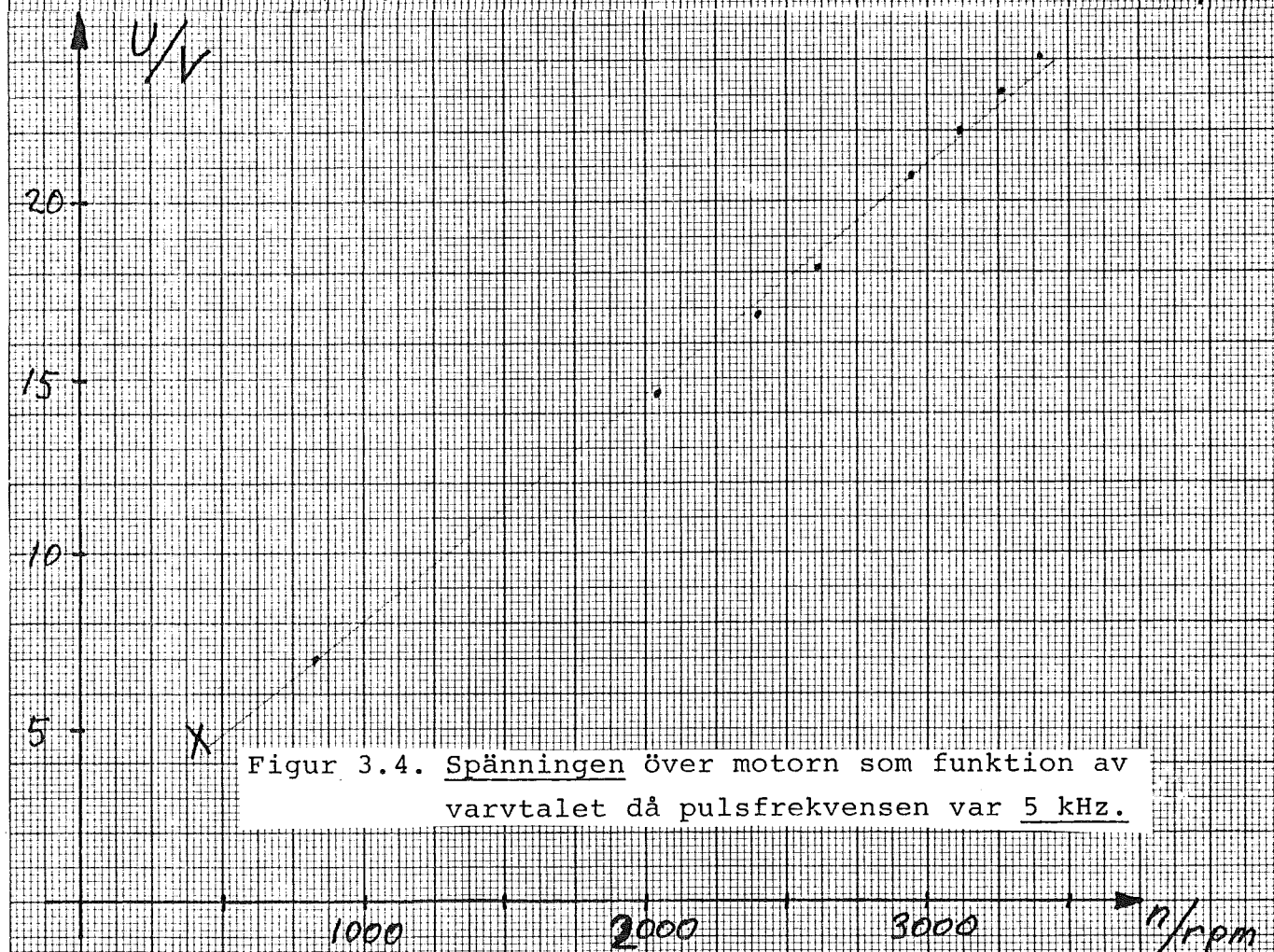
3.5 Diagram

Mätningarna som är tabellerade i kap 3.4 har jag här fört in i diagram. Fig. 3.1-3.8 innehåller spänningen relaterat till varvtalet. Fig 3.9-3.16 innehåller strömmen relaterat mot varvtalet och fig 3.17-3.23 duty cyclen mot varvtalet. Ström-, spännings- och duty cycle diagrammen börjar med de lägsta frekvenserna. För ström och spänning är den allra första figuren från analoga körningen. X markerar mätpunkt vid lägsta hastighet med 10 % kriterium enligt kap 3.2.

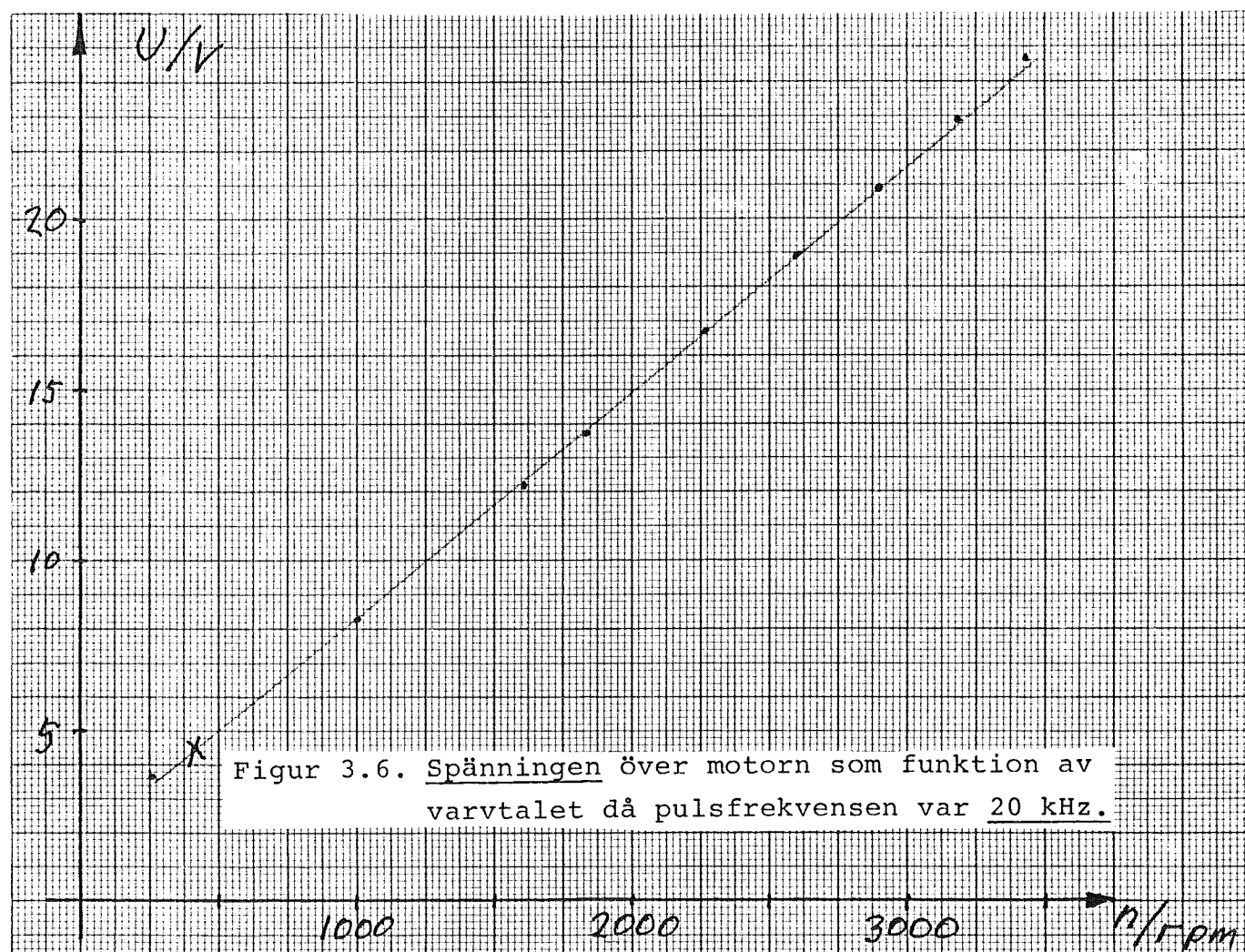
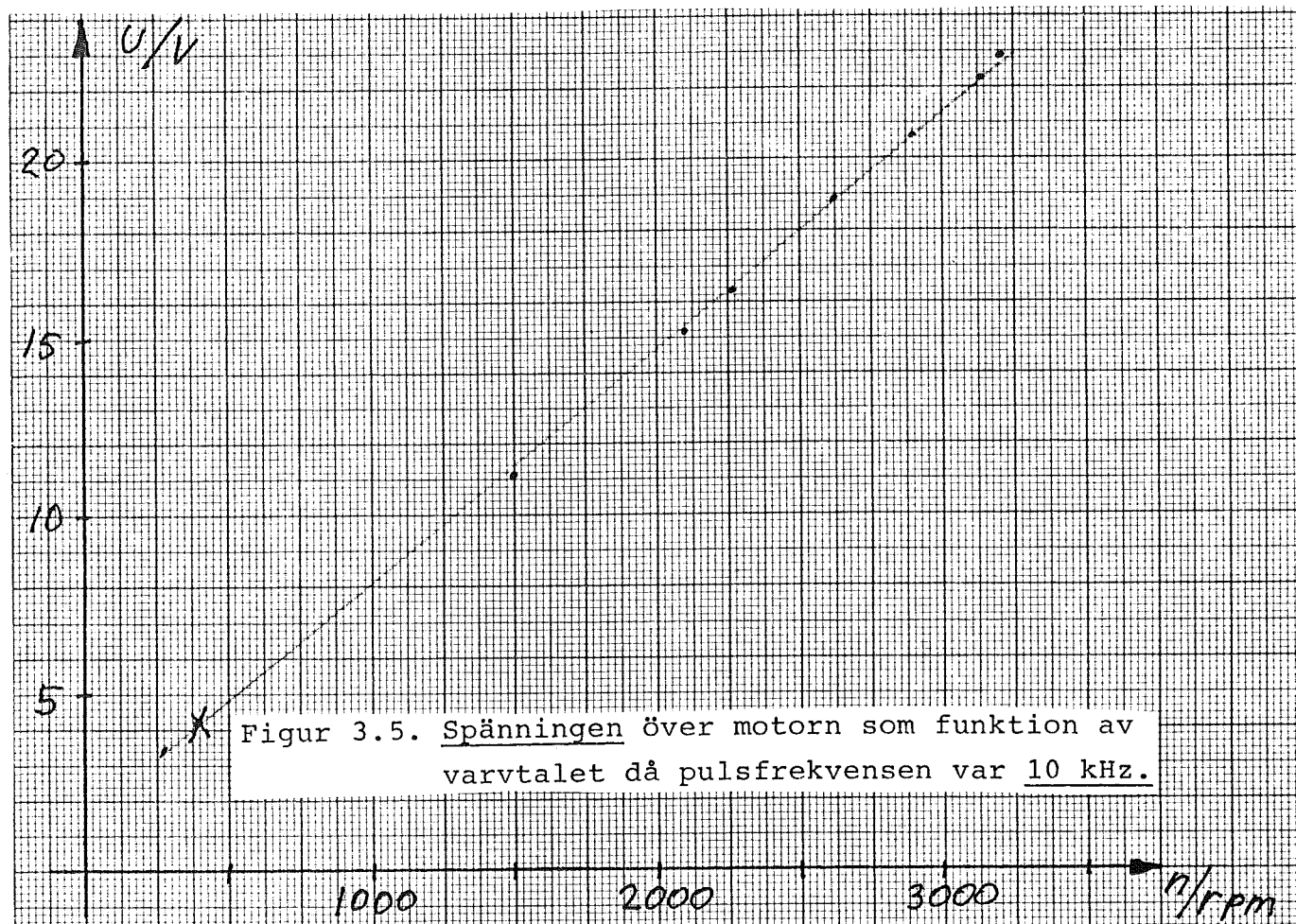


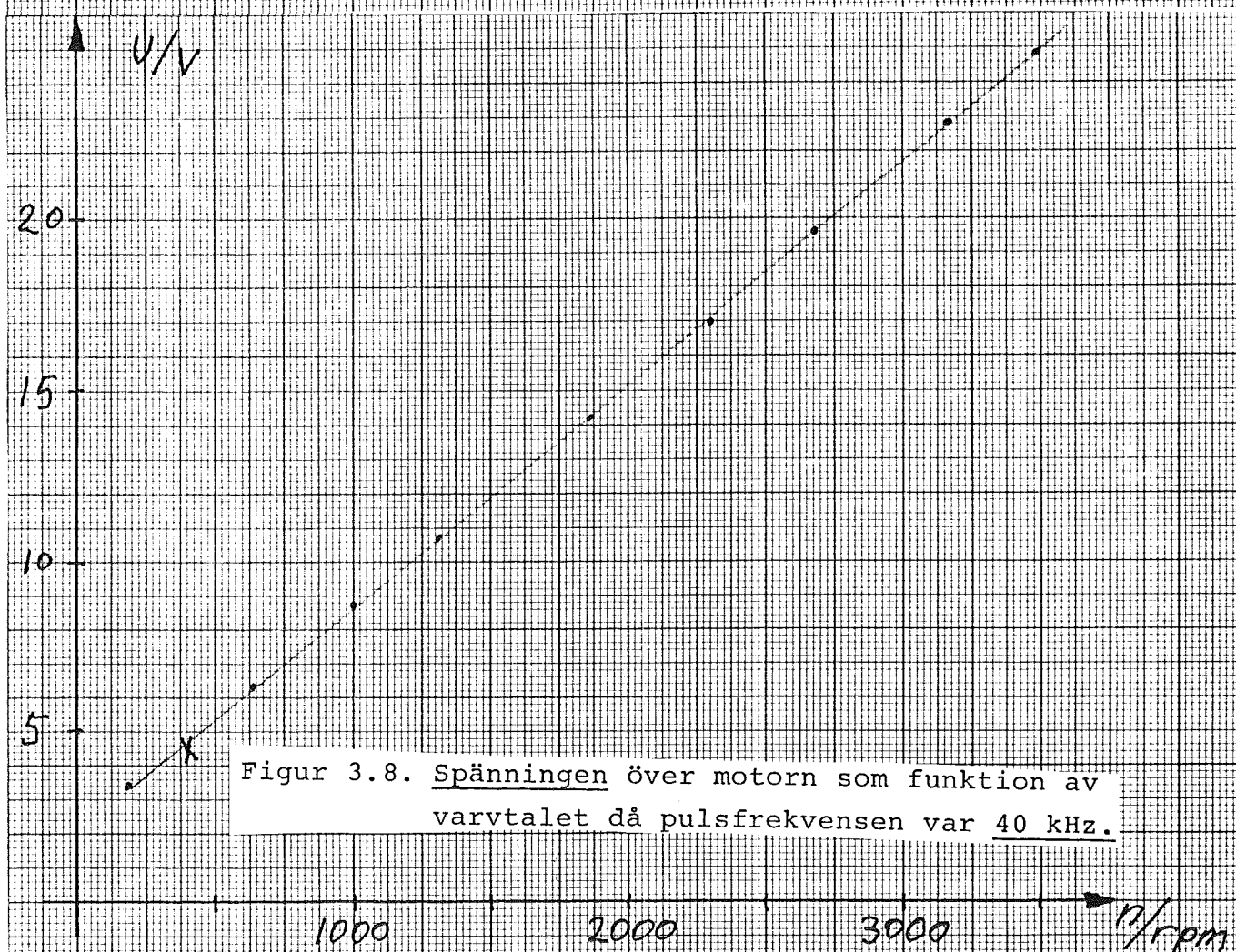
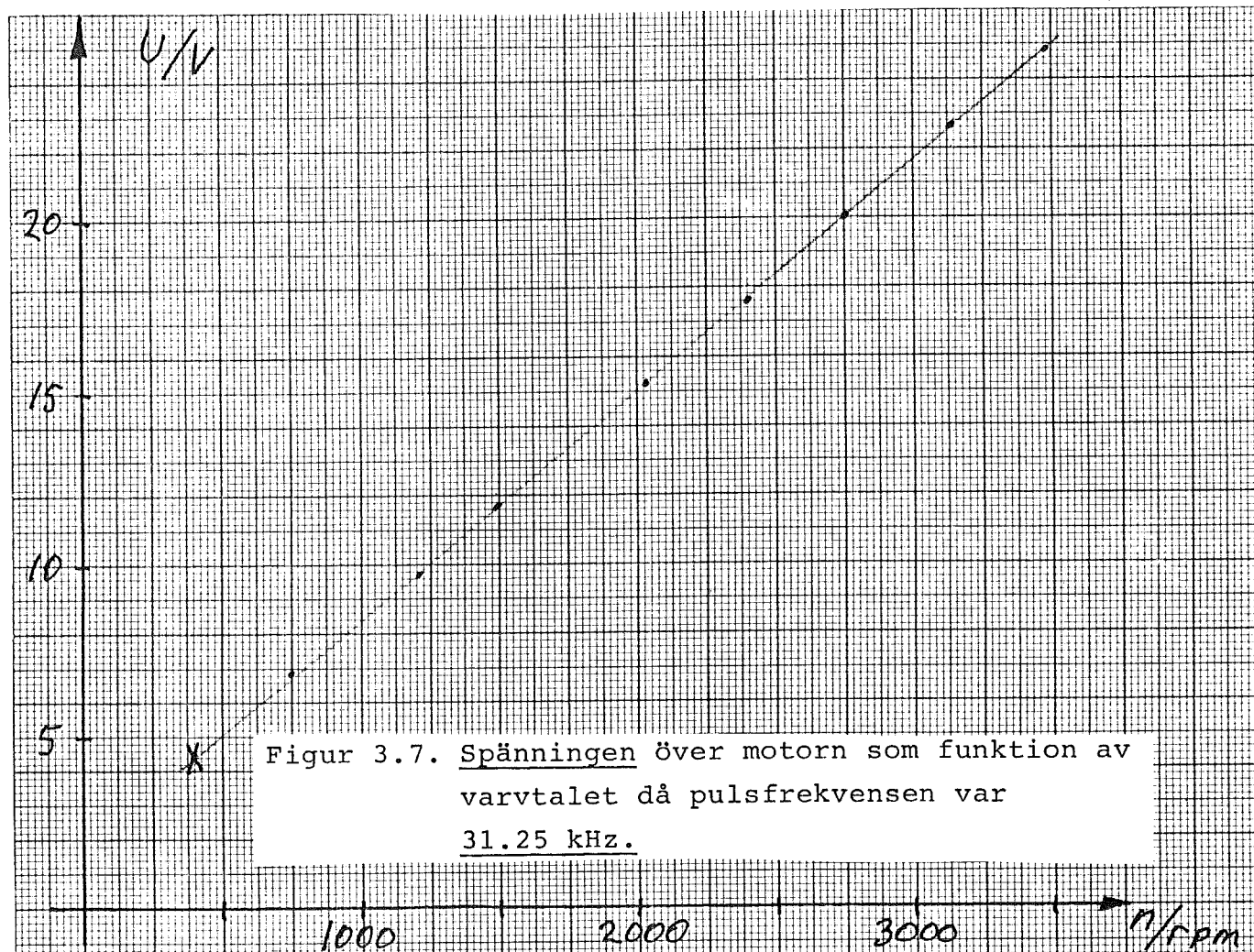


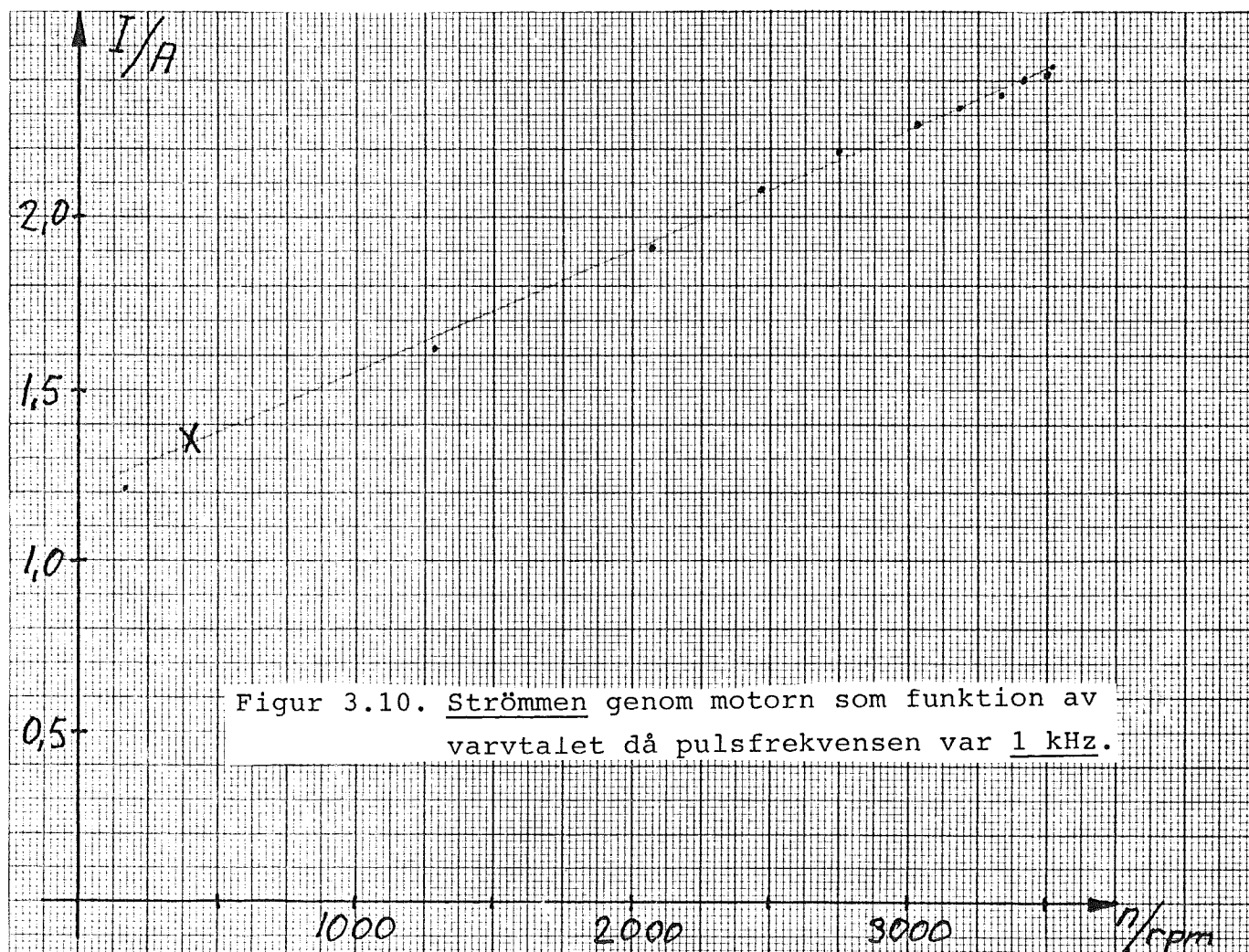
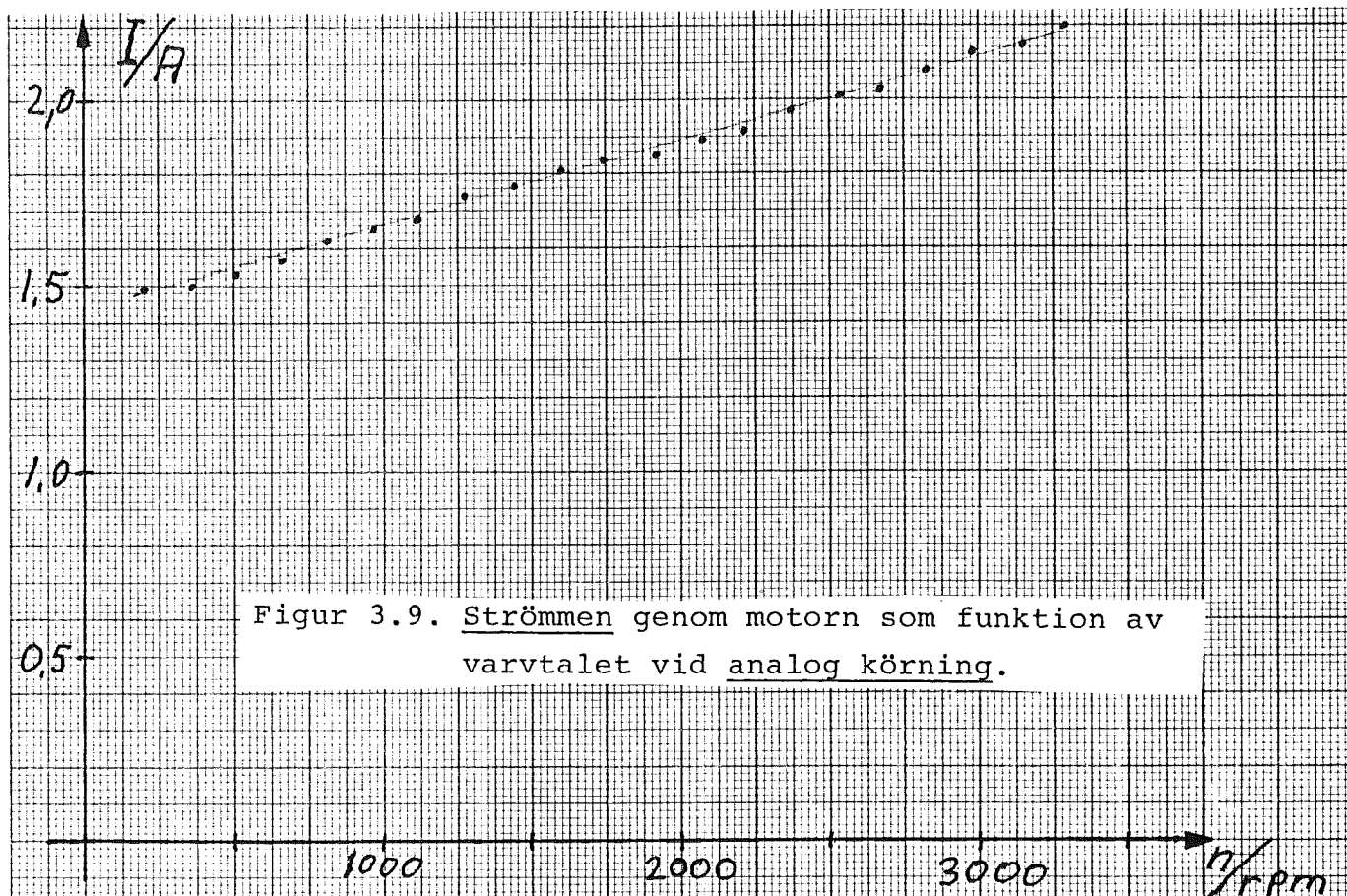
Figur 3.3. Spänningen över motorn som funktion av varvtalet då pulsfrekvensen var 2 kHz.

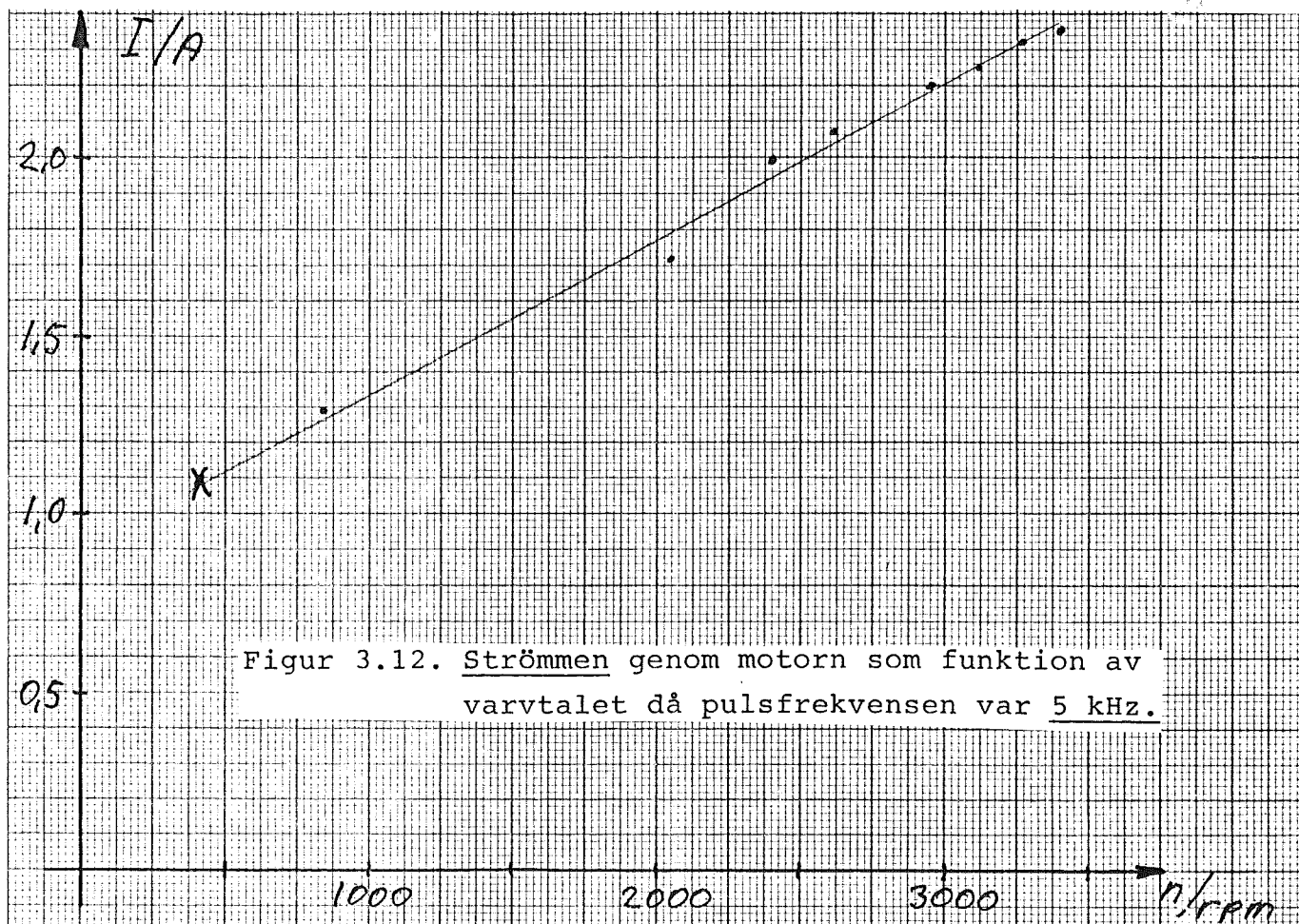
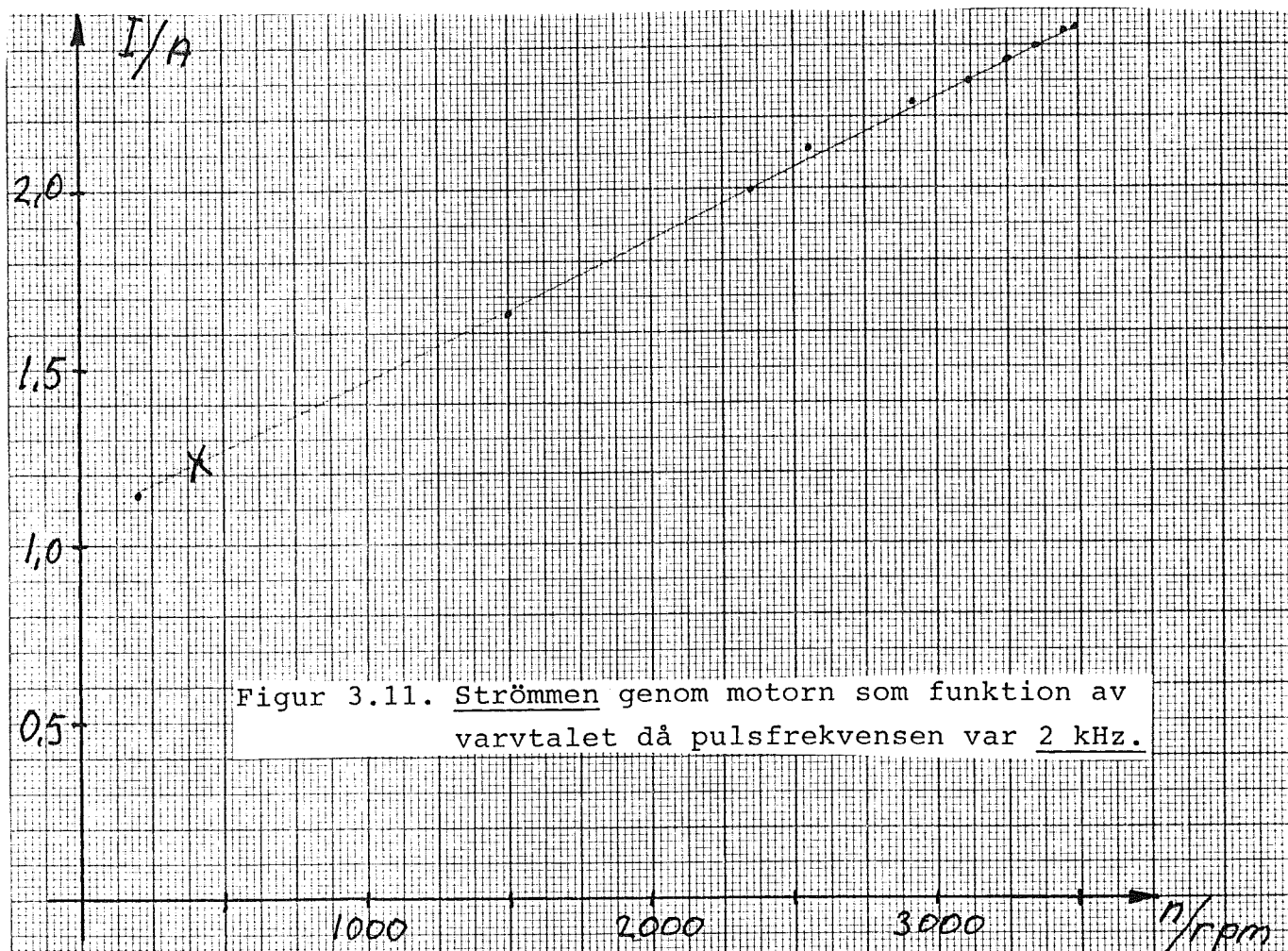


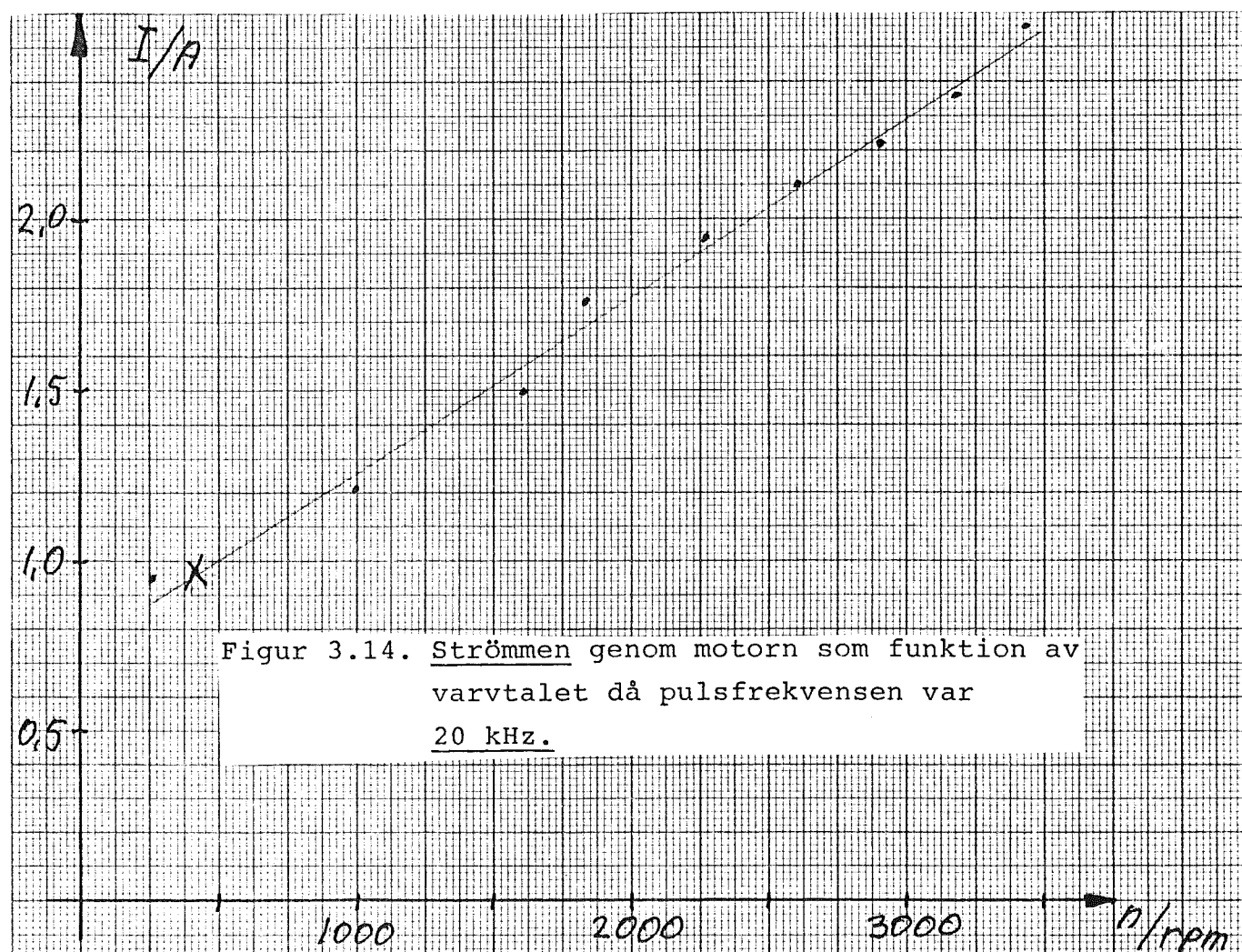
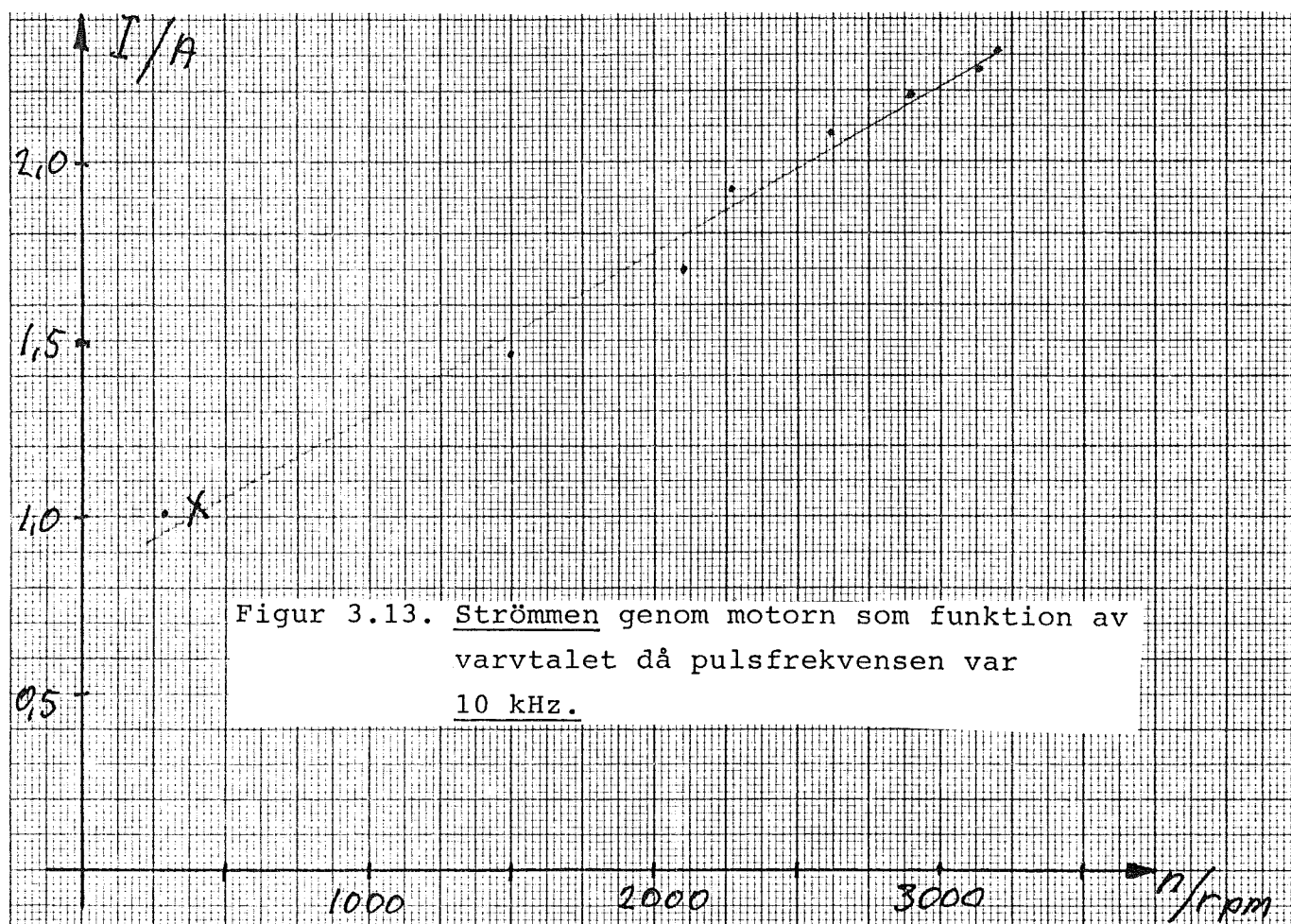
Figur 3.4. Spänningen över motorn som funktion av varvtalet då pulsfrekvensen var 5 kHz.

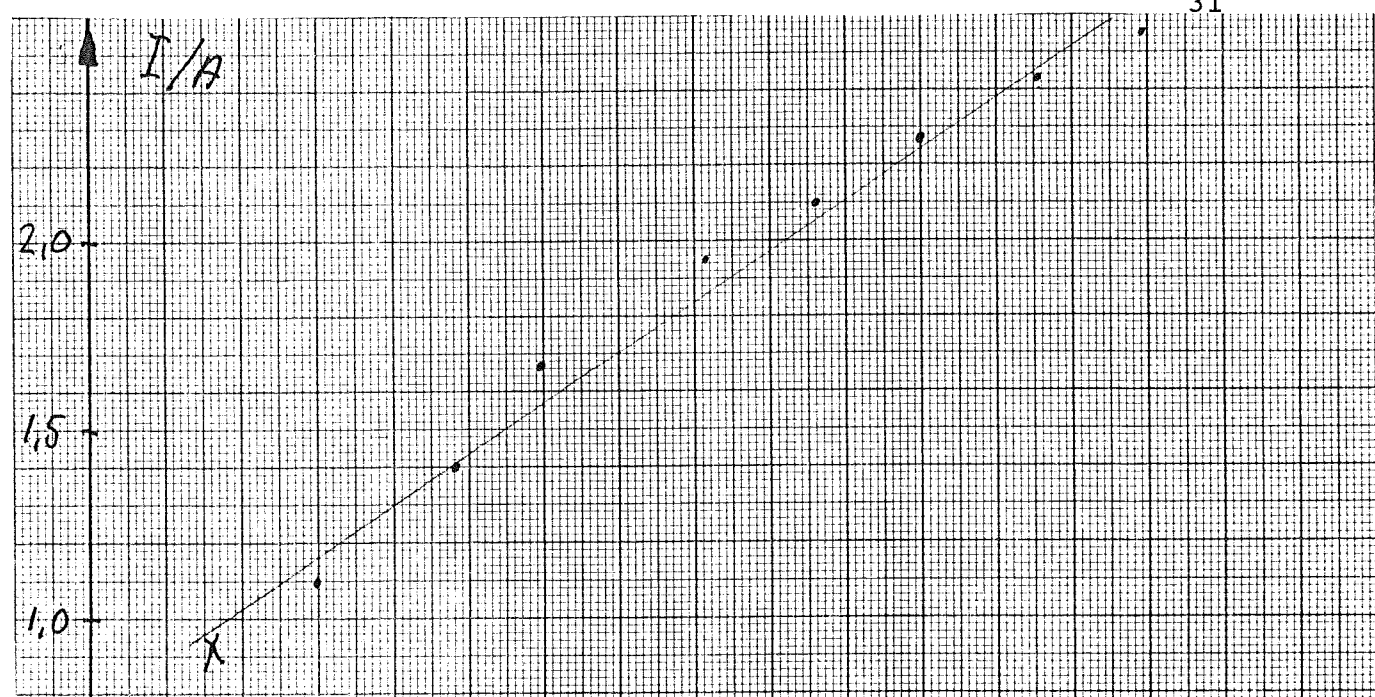




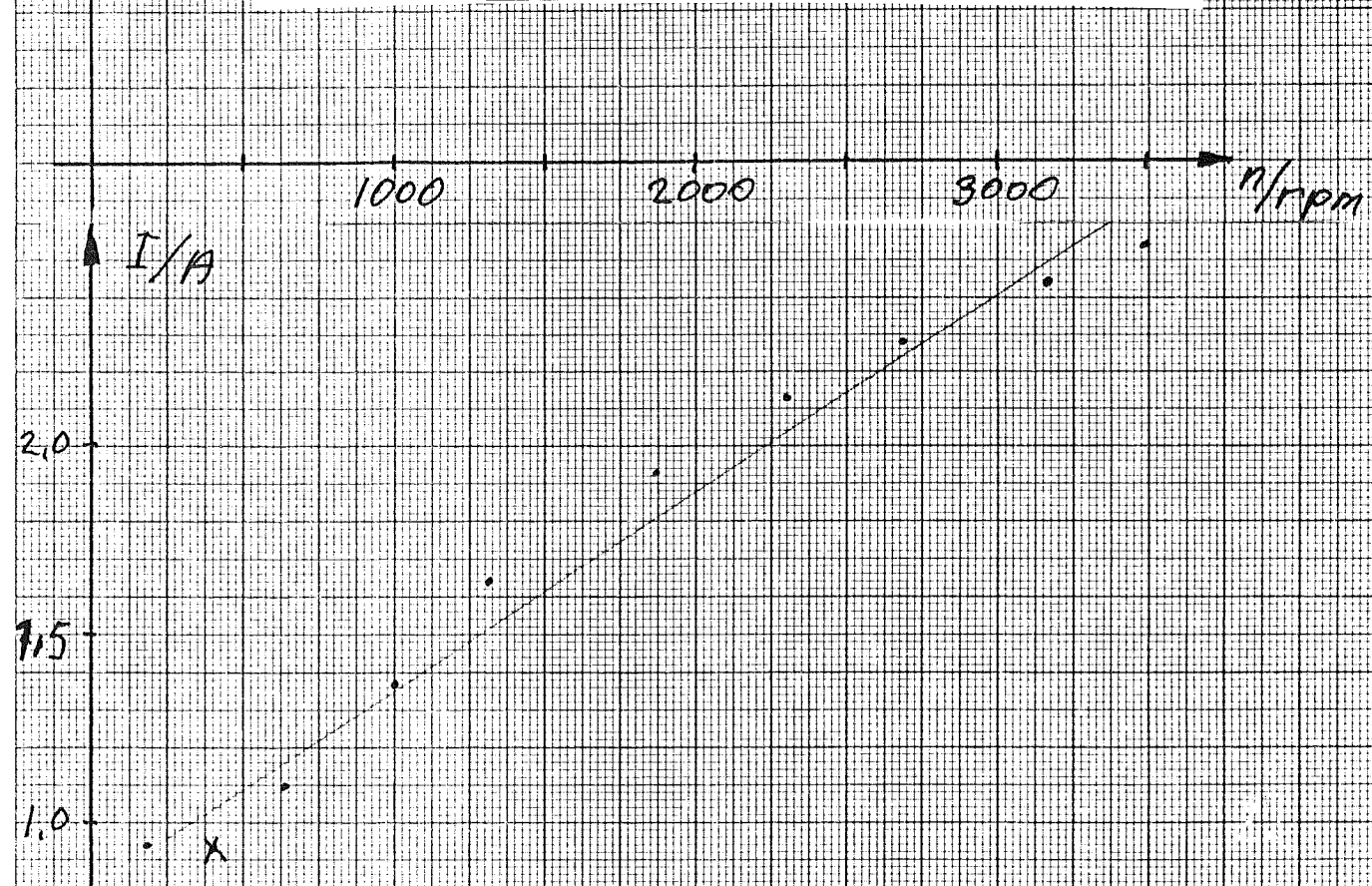




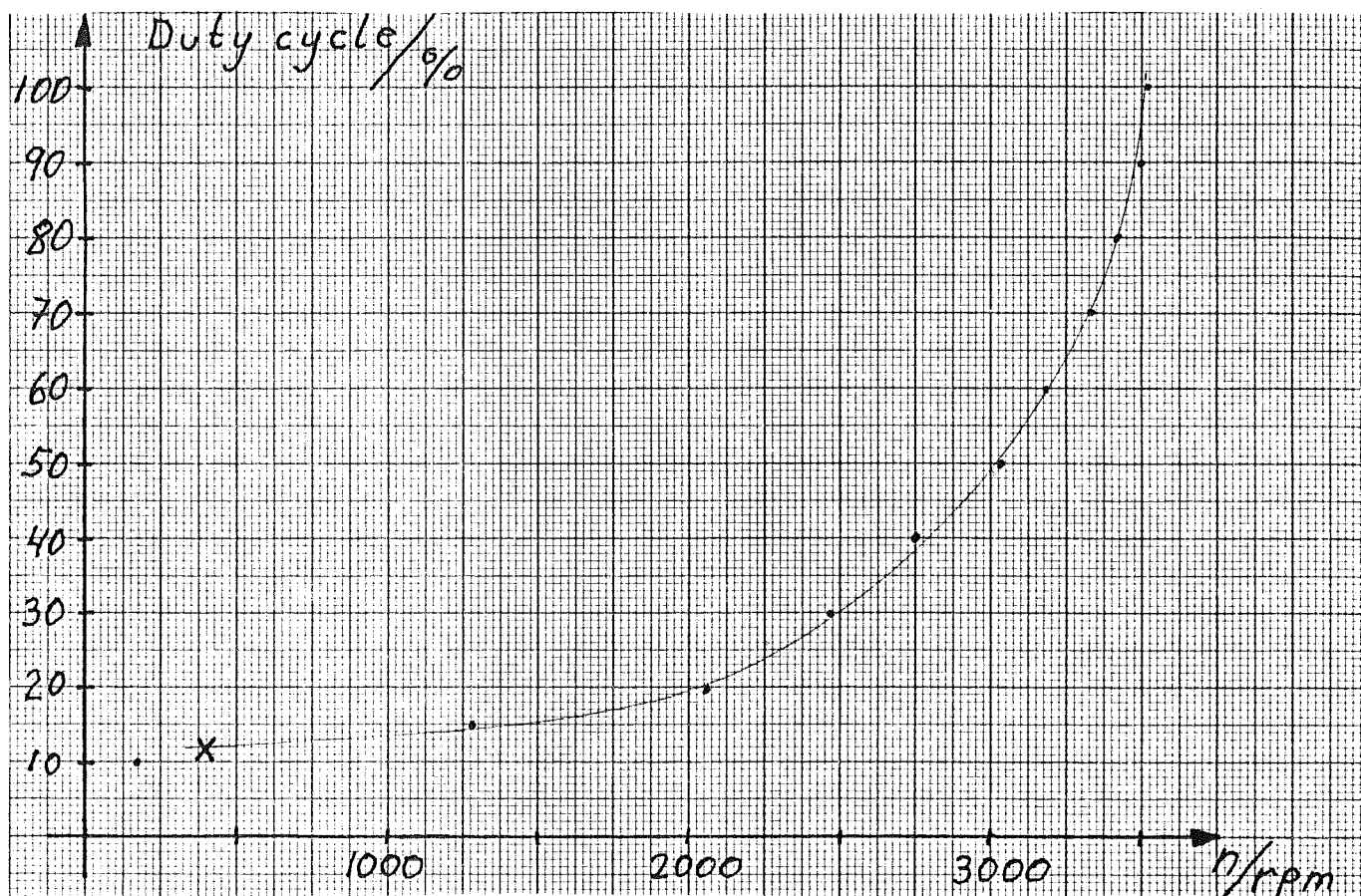




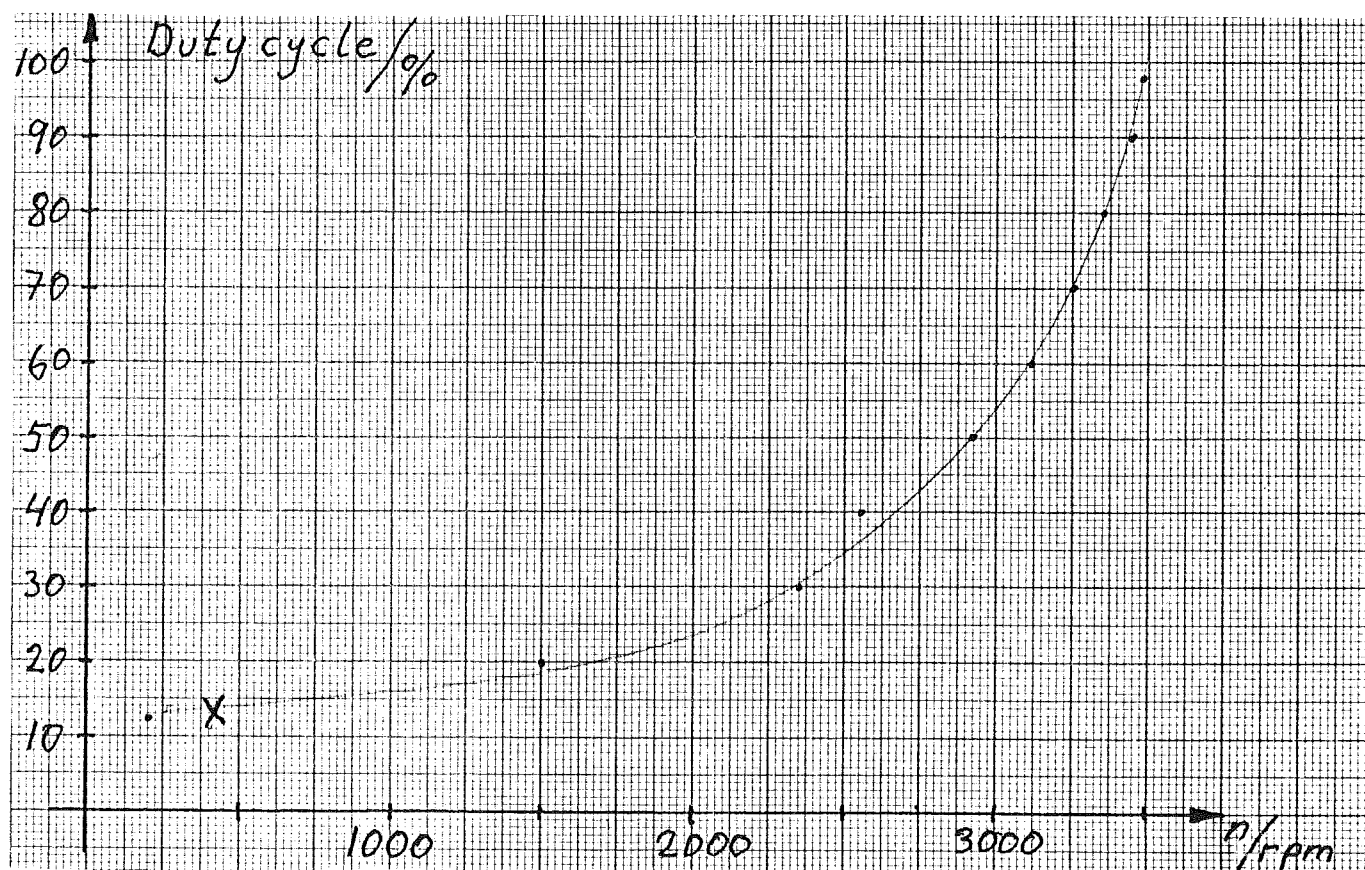
Figur 3.15. Strömmen genom motorn som funktion av varvtalet då pulsfrekvensen var 31.25 kHz.



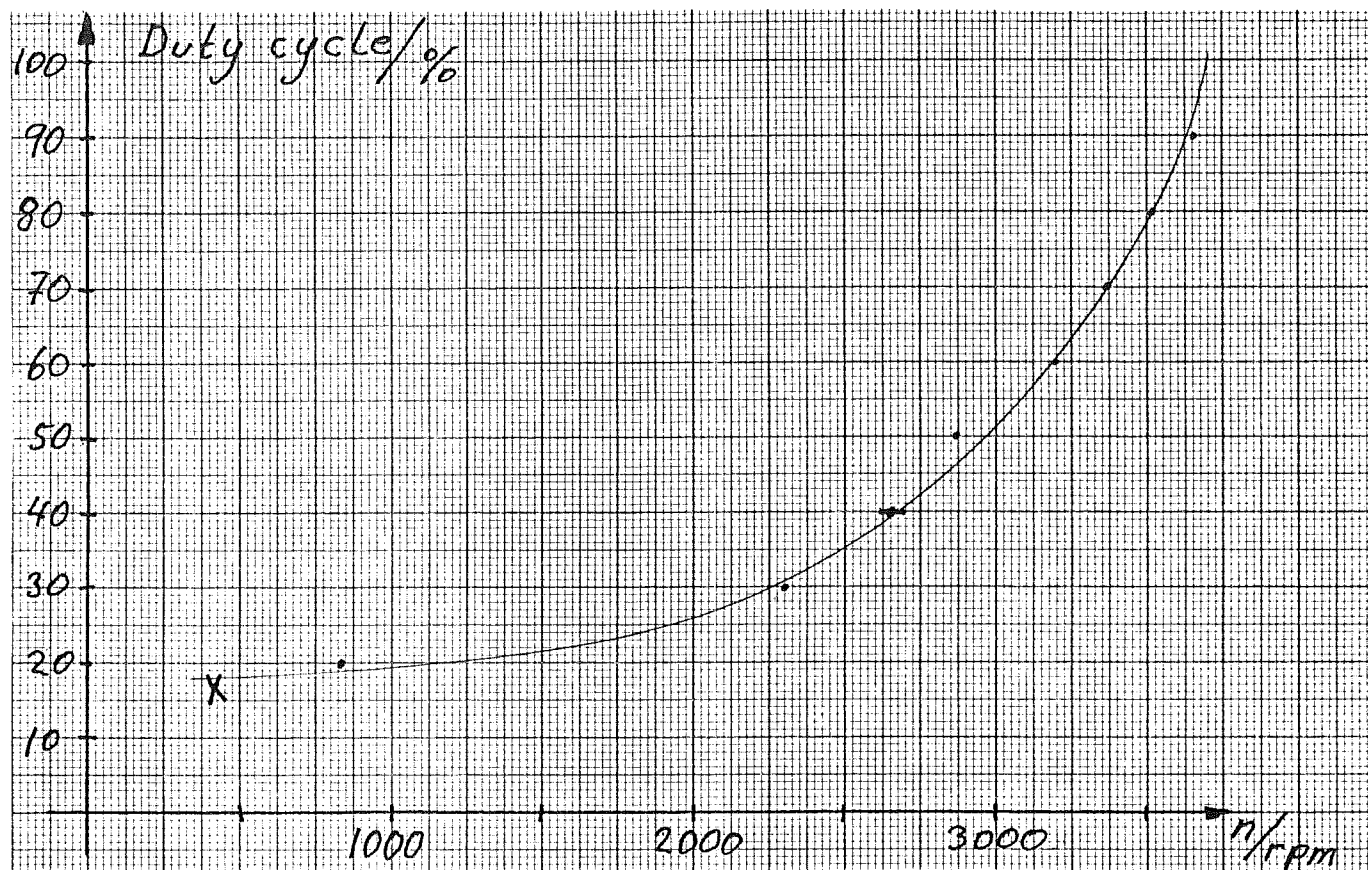
Figur 3.16. Strömmen genom motorn som funktion av varvtalet då pulsfrekvensen var 40 kHz.



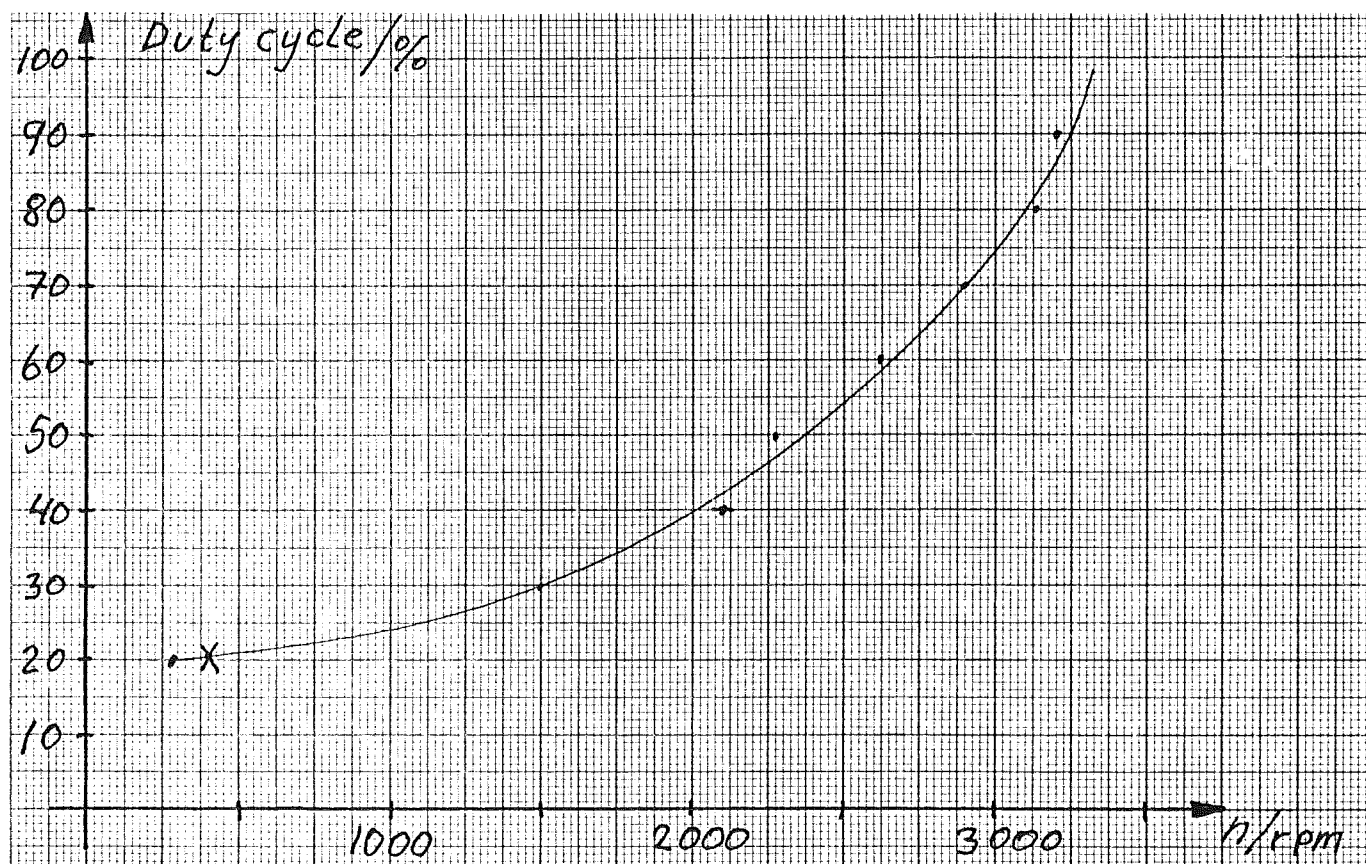
Figur 3.17. Styrsignalens duty cycle som funktion av varvtalet då pulsfrekvensen var 1 kHz.



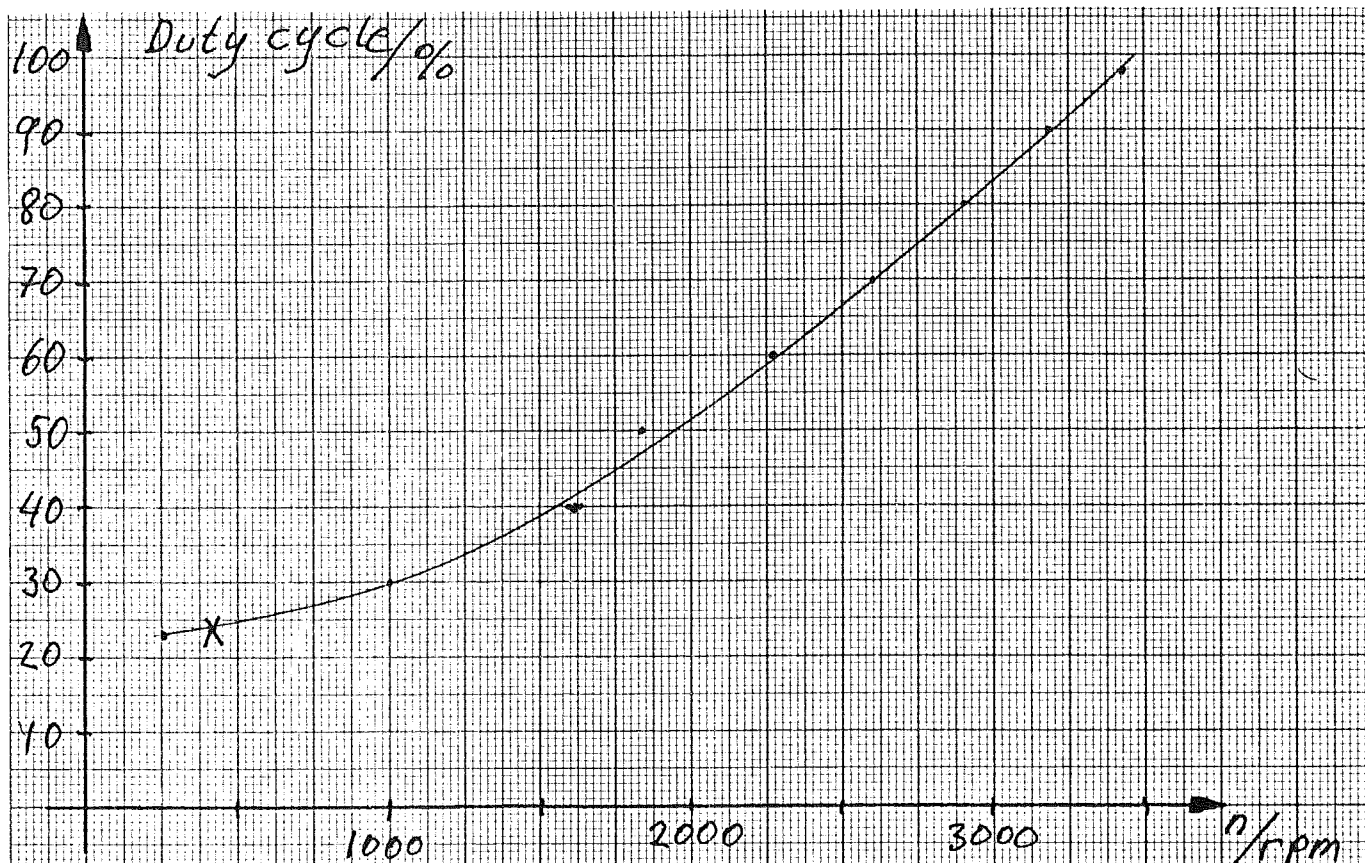
Figur 3.18. Styrsignalens duty cycle som funktion av varvtalet då pulsfrekvensen var 2 kHz.



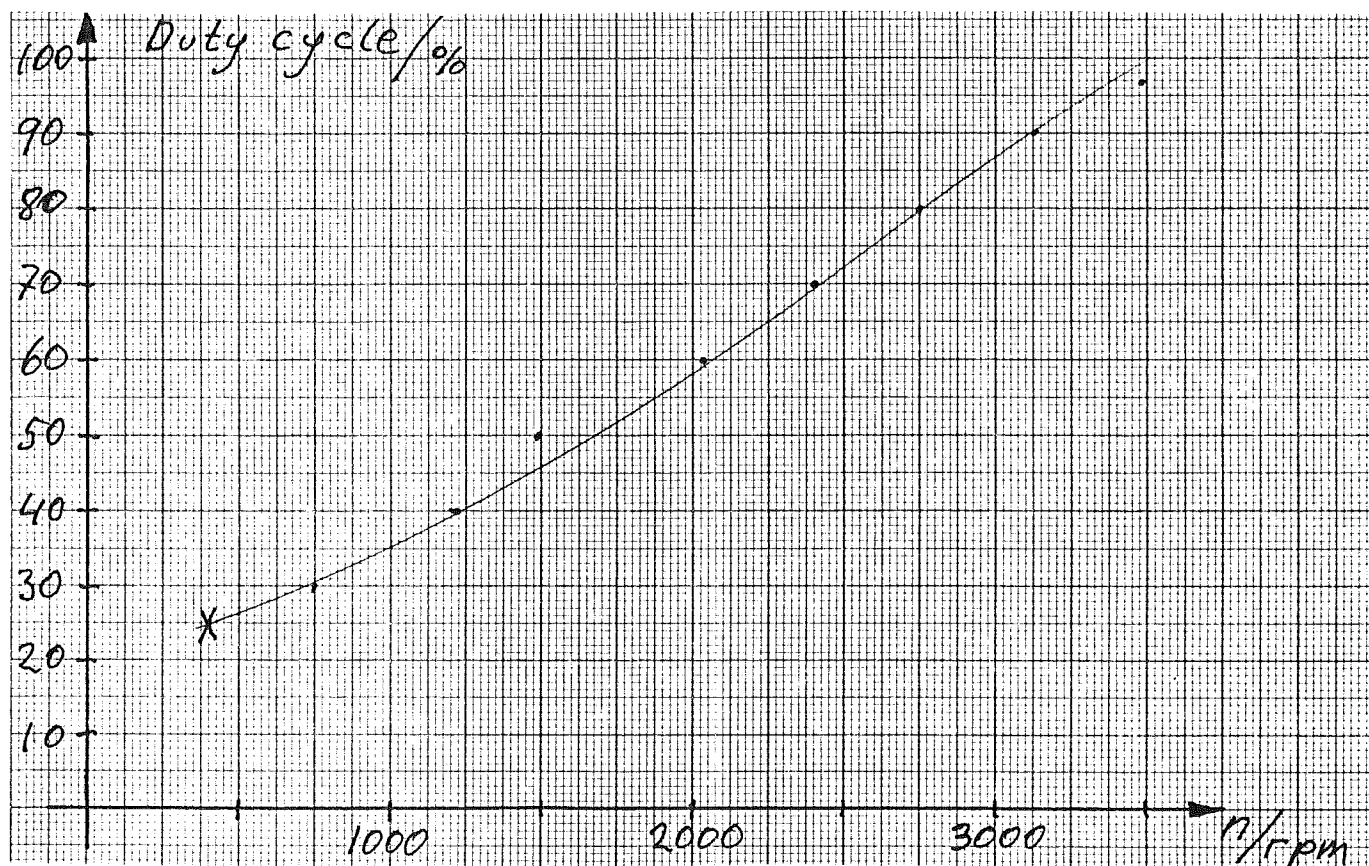
Figur 3.19. Styrsignalens duty cycle som funktion av varvtalet då pulsfrekvensen var 5 kHz.



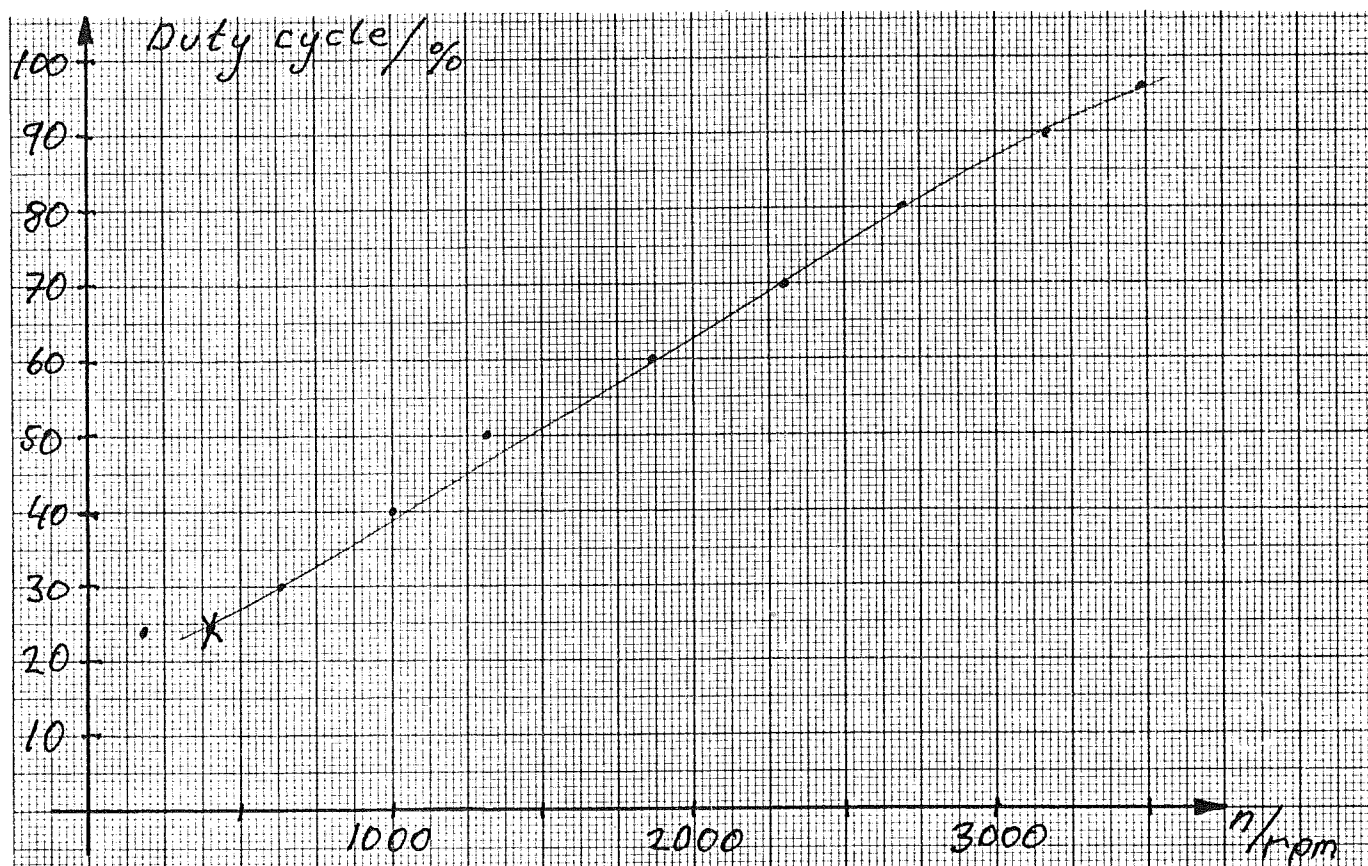
Figur 3.20. Styrsignalens duty cycle som funktion av varvtalet då pulsfrekvensen var 10 kHz.



Figur 3.21. Styrsignalens duty cycle som funktion av varvtalet då pulsfrekvensen var 20 kHz.



Figur 3.22. Styrsignalens duty cycle som funktion av varvtalet då pulsfrekvensen var 31.25 kHz.



Figur 3.23. Styrsignalens duty cycle som funktion av varvtalet då pulsfrekvensen var 40 kHz.

3.6 Slutsatser av frekvensmätningar

Som man kan se av mätningarna är inte valet av pulsfrekvens kritisk. Med utgångspunkt från dessa mätningar bestämde jag mig för att använda 2 kHz i mina regulatormätningar i nästa kapitel. Detta därför att 1 kHz inte ger noggrannare reglering pga ADC:ns upplösning och därför att 2 kHz medför ett något mindre märkbart oljud vid de låga hastigheterna. Att gå mycket högre upp i frekvens begränsades av min algoritm, som måste hinna genomlöpas.

Sammanfattningsvis kan man säga att vid framtida användning av PWM-styrning kan pulsfrekvensen ställas så högt det går, så att algoritmen och eventuell kommunikation med en huvuddator precis hinner med att genomlöpas under en period hos PWM-signalen.

4 TESTER MED REGULATORER

4.1 Allmänt om regleringen

Frekvensmätningarna i kapitel 3 syftade till att få fram en lämplig arbetsfrekvens för systemet. Detta kapitel är till för att utröna hur systemet uppför sig vid reglering, då styrsignalen är en PWM-signal. Vid dessa tester använder jag hela tiden samma frekvens, 2 kHz. Jag valde denna eftersom programmet måste hinna med att genomlöpas och den uppförde sig bra vid mätningarna i kap 3. Att jag inte valde lägre berodde på att jag ville ha så hög frekvens som möjligt, för att oljudet skulle vara så litet som möjligt. Dessa och fler slutsatser finns i kap 3.6.

Algoritmen som jag använder är uppbyggd enligt fig. 4.1.

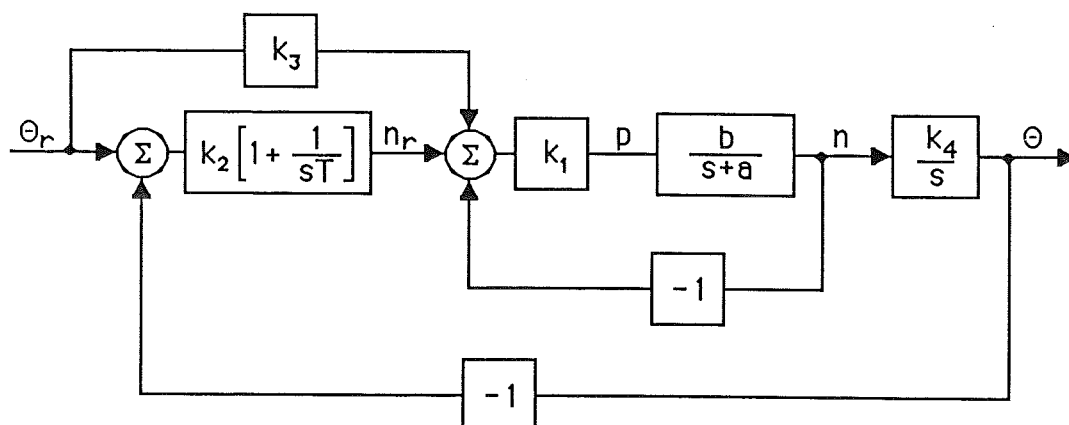


Fig. 4.1. Totala algoritmen som går att använda med program beskrivet i appendix A.

Programmet som realiserar denna algoritm är närmare beskrivet i appendix A. Vid testerna använder jag dock ej hela algoritmen. Jag börjar med öppen loop styrning för att få motorparametrarna, sen

fortsätter jag med att sätta dit återkopplingarna en efter en och mäter fram de förstärkningar som ger bäst system. I kap 4.5 börvärdesstyr jag och använder där endast hastighetsåterkopplingen i fig. 4.1 ovan. Hur mycket jag använder står beskrivet vid de olika testerna i kapitlena 4.2 till 4.5. Så som regleringen är uppbyggd ges en signal till motorn som en pulsvidd p . Denna blir med de förstärkningar som jag använt uttryckt i hundra-dels procent. Om $t_{ex} = 1500$ och $n_r = 2000$ vid ett jämviktsläge och $k_1 = 4$ skulle vi få $p = 2000$. Jag förutsätter nu att $k_3 = 0$. Med $p = 2000$ har vi alltså en duty cycle på 20 %. Eftersom jag använder en duty cycle som utsignal till motorn blir naturligtvis motorparametrarna a och b beroende av pulsamplituden. Vid alla dessa mätningar använder jag dock samma amplitud som vid frekvensmätningarna i kap 3. Detta blir ca 27 V pga att kraftkortet tar en del av matningsspänningens 30 V. Mer om detta finns i appendix D. Eftersom pulsgivaren ger 200 pulser per varv får vi förstärkningen $k_4 = 200$. Data på mätutrustningen finns i appendix E.

4.2 Öppen loop körning

4.2.1 Allmänt

Öppen loop mätningarna har jag gjort för att ta reda på motorns parametrar a och b . För att kunna köra i öppen loop har jag i den totala algoritmen (se fig. 4.1) tagit bort återkopplingarna och satt $k_2 = 0$. Dessutom har jag satt $k_1 = k_3 = 1$ för att få in ett börvärde. Som utsignal har jag varvtalet n . Detta ger ett system enl. fig. 4.2.

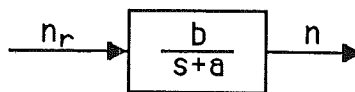


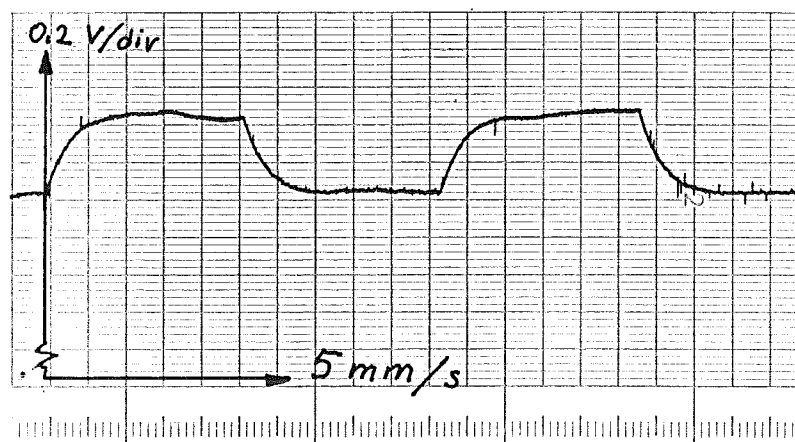
Fig. 4.2. System vid öppen loop körning.

För att få reda på parametrarna i min modell mäter jag stegsvaret. Hur detta läggs på åskådliggörs i appendix A, fig. A6.

4.2.2 Mätningarna

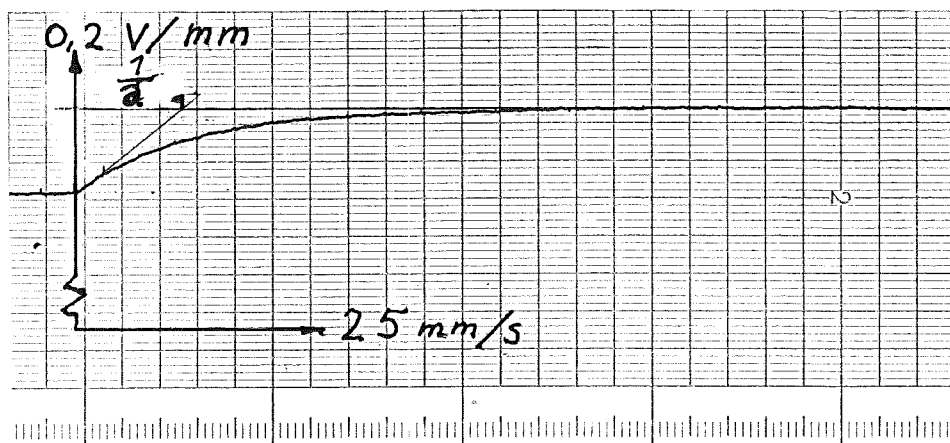
När jag lägger på steget i börvärdet står inte motorn stilla utan ligger i ett stationärt tillstånd. För att hinna med och mäta lägger jag på börvärdet som en fyrkantsvåg där det undre börvärdet är n_{r1} och det övre är n_{r2} .

Jag börjar med att ha $n_{r1} = 2000$ och $n_{r2} = 2300$. Det ger ett steg på 300. Detta gav ett svar enligt fig. 4.3. Signalen är en direkt mätning av tacon. Denna ger ju 6 V/1000 rpm och graderingen 0.2 V/mm motsvarar alltså 33.3 rpm/mm.



Figur 4.3. Stegsvaret då steget in var 300.

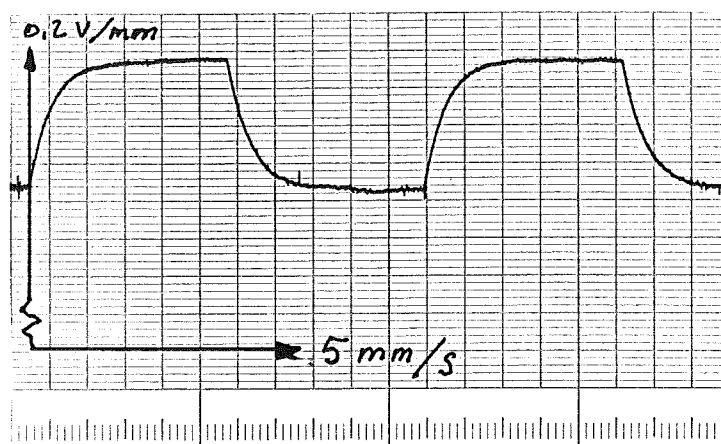
För att kunna mäta noggrannare har jag i fig. 4.4 dragit ut tidsaxeln.



Figur 4.4. Stegsvaret förstorat. Steget in fortfarande 300.

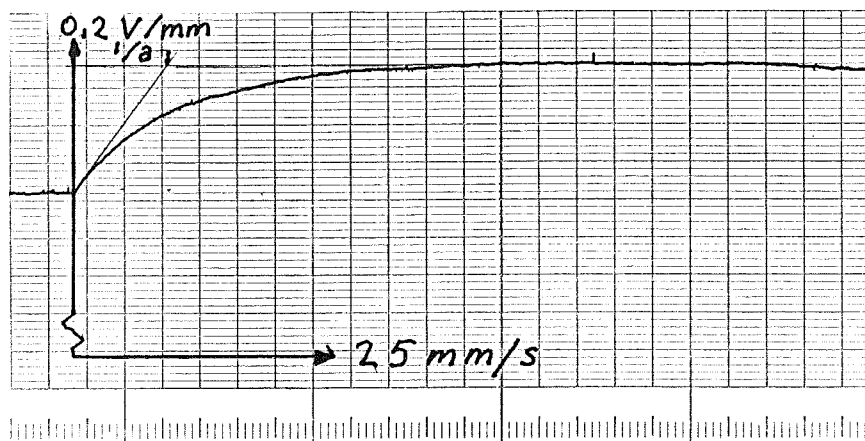
Ur figuren kan man snabbt få ett ungefärligt värde på a (se fig. 4.4). Denna ger $1/a = 14 \text{ (mm)}/25 \text{ (mm/s)} = 0.56 \text{ s}$. Alltså a ungefär lika med 1.79 s^{-1} . Noggrannare uträkningar görs i kap 4.2.3.

Jag gjorde en mätning till för att se hur systemet uppför sig om man lägger på ett något större steg. Här är $n_{r1} = 2000$ och $n_{r2} = 2500$ vilket ger ett steg på 500. Stegsvaret blev då som i fig. 4.5.



Figur 4.5. Stegsvår då steget in var 500.

Precis som vid föregående mätning har jag förstorat tidsaxeln för att få noggrannare mätning.



Figur 4.6. Stegsvaret förstorat. Steg in 500.

Mätningen av $1/a$ blev här 0.5 s, dvs $a = 2.00 \text{ s}^{-1}$. Nu följer de teoretiska beräkningarna i nästa kapitel.

4.2.3 Teoretiska beräkningar

För att få ut värde på parametrarna sätter jag nedan upp överföringsfunktionen och identifierar den med svaren i fig. 4.4 och 4.6.

Steget in är

$$n_r = n_{r2} - n_{r1} \quad (4.2.a)$$

Detta ger utsignalen n som

$$n = \frac{b}{s+a} \cdot \frac{n_r}{s} \quad (4.2.b)$$

Med invers laplace-transform fås stegsvaret som

$$n = n_r \cdot \frac{b}{a} (1 - e^{-at}) \quad (4.2.c)$$

Ur kurvan får jag sen bl a

$$n(\infty) = n_r \cdot \frac{b}{a} \quad (4.2.d)$$

Därefter tar jag det a jag fick ur figuren och får ett b. Efter lite mer anpassning av kurvan fick jag värdena

$$a = 1.65 \text{ s}^{-1}$$

$$b = 2.02 \text{ s}^{-1}$$

då jag satte in 300 som steg.

Likadana räkningar men med 500 som steg gav

$$a = 1.92 \text{ s}^{-1}$$

$$b = 2.18 \text{ s}^{-1}$$

4.2.4 Slutsatser

Efter att ha anpassat parametrarna ovan till respektive kurva (fig. 4.4 och 4.6) kan jag konstatera att värdena

$$a = 1.65 \text{ s}^{-1}$$

$$b = 2.02 \text{ s}^{-1}$$

insatt i (4.2.c) stämde något bättre överens med kurvan i fig. 4.4 än vad de andra värdena stämde med fig. 4.6. Detta kan bero på att då jag lägger

på ett större steg så gör olinjäriteten i duty cyclen, se fig. 3.18 i kapitel 3, sig mer gällande.

Med de första parametrarna fick jag ett stegsvar som maximalt avvek 3 % från kurvans värde. Avvikelsen låg dock under största delen av accelerationen på ca 1 %. Med de andra parametrarna fick jag värden som i början av accelerationen var 3 % mer och i slutet 3 % mindre än värdena på kurvan.

4.3 P-reglering av hastighet

4.3.1 Allmänt

Med dessa mätningar vill jag försöka få fram en förstärkning där systemet uppför sig bra. Det gäller alltså att få fram en så stor förstärkning som möjligt, för att få ett snabbt system. Förstärkningen skall dock inte vara så stor att signalen till motorn går i mättningsområde, dvs ger 100 % duty cycle ut, därför att systemet då blir väldigt stirrigt och skiftar mellan att ge full fram till att ge full broms.

Vid dessa mätningar testar jag två olika modeller. Den ena har förstärkningen i återkopplingen, se fig. 4.7, och den andre förstärker först efter återkopplingen, se fig. 4.8.

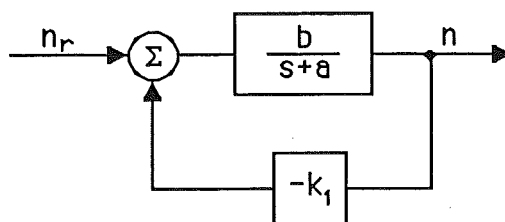


Fig.4.7. System med förstärkning i återkopplingen.

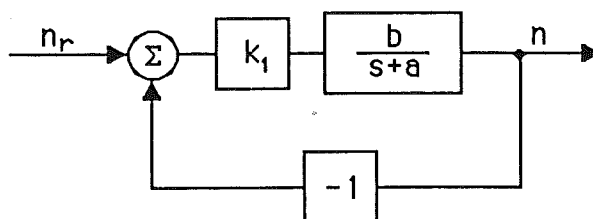


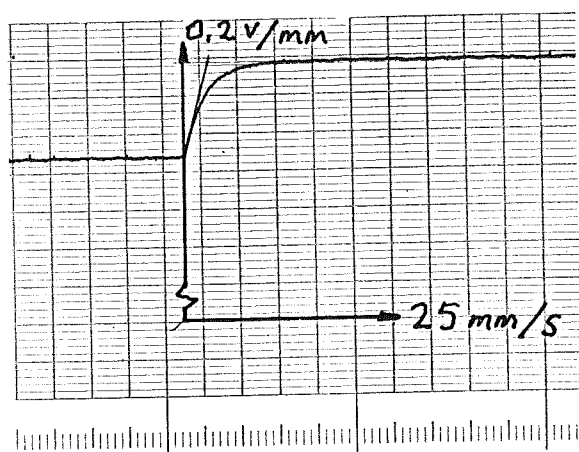
Fig. 4.8. System med förstärkningen efter återkopplingen.

Systemet i fig. 4.7 har en lägre förstärkning i det totala systemet än vad systemet i fig. 4.8 har. Vad detta innebär återkommer jag till i respektive underkapitel för de båda mätningarna och i slutsatserna i kap 4.3.4.

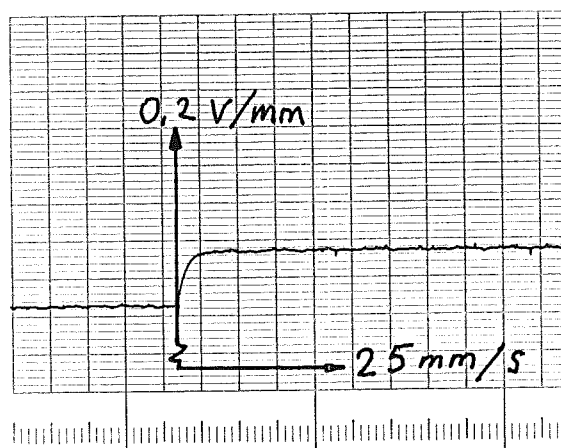
4.3.2 System med förstärkning i återkopplingen

4.3.2.1 Mätningar

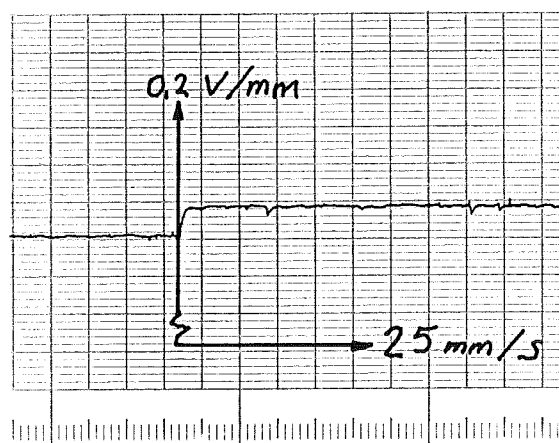
För att få fram en lämplig förstärkning till systemet i fig. 4.7 började jag med att sätta $k_1 = 1$. Därefter dubblade jag förstärkningen vid varje mätning. Vid dessa mätningar lade jag på steg precis som vid mätningarna i kap 4.3. För att motorn skall rotera behövs dock här en större insignal pga den lägre förstärkningen i systemet. Här låter jag $n_{r1} = 8000$ och $n_{r2} = 10000$. Det ger ett steg som är $n_r = 2000$. De stegsvar jag fick finns i fig. 4.9 ($k_1=4$), 4.10 ($k_1=8$), 4.11 ($k_1=16$).



Figur 4.9. Stegsvvar då förstärkningen k_1 var 4.



Figur 4.10. Stegsvvar då förstärkningen k_1 var 8.



Figur 4.11. Stegsvvar då förstärkningen k_1 var 16.

Den förstärkning som gav det bästa systemet var $k_1=4$. Alltså det svar som syns i fig. 4.9. I figurerna 4.10 och 4.11 syns att systemet blev stressigare vid högre förstärkning. Vid öppen loop mätningarna approximerade jag systemet med en första ordningens överföringsfunktion. Med denna återkoppling blir systemet fortfarande av första ordningen. Därför kan jag ta tidskonstanten för detta system precis som vid öppen loop. Detta gav $1/a_1 = 0.128 \text{ s}$, dvs $a_1 = 7.81 \text{ s}^{-1}$, vid $k_1=4$. Systemet har alltså blivit snabbare.

4.3.2.2 Teoretiska beräkningar

Med steget n_r in så får vi utsignalen som

$$n = \frac{b}{s+(a+k_1 \cdot b)} \cdot \frac{n_r}{s} \quad (4.3.a)$$

Den tidskonstant som jag utläste ur fig. 4.9 är alltså

$$a_1 = a + k_1 \cdot b \quad (4.3.b)$$

Inverse Laplace transform av (4.3.a) ger tidsfunktionen

$$n = n_r \cdot \frac{b}{a+k_1 \cdot b} (1 - e^{-(a+k_1 \cdot b)t}) \quad (4.3.c)$$

Med parametrarna a och b från öppen loop mätningarna och $k_1=4$ så jämförde jag överföringsfunktionen (4.3.c) med svaret i fig. 4.9. Avvikelsen blev som störst 1.8 % och vid det stationära värdet skilde det endast 0.5 %.

Ur figurerna 4.9 till 4.11 ser man också ett resultat av att ha förstärkningen i återkopplingen. Amplituden blir mindre ju högre förstärkning man har. Nedanstående resonemang ger en förklaring till detta.

Ur formel (4.3.c) fås vid stationaritet

$$n(\infty) = n_r \frac{b}{a + k_1 \cdot b} \quad (4.3.d)$$

Då förstärkningen är stor är $k_1 \cdot b > a$ och då kan vi för överblickens skull approximera (4.3.d) med

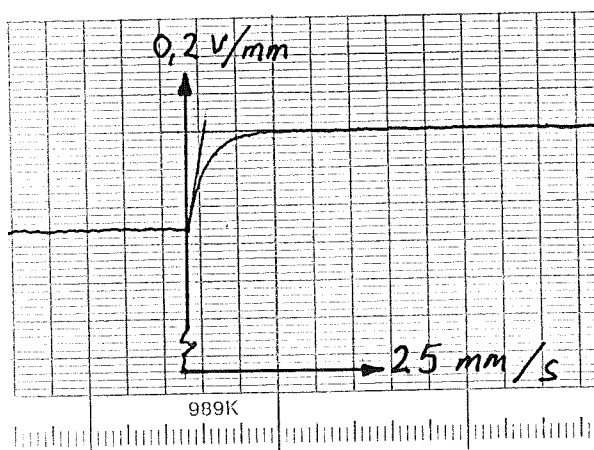
$$n \approx n_r \cdot \frac{1}{k_1} \quad (4.3.e)$$

För att öka förstärkningen i det totala systemet kan man sätta förstärkningen efter återkopplingen istället. Detta gör jag i nästa kapitel.

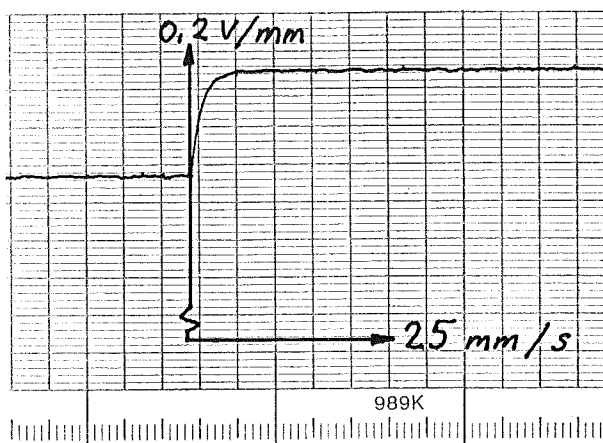
4.3.3 Förstärkningen efter återkopplingen

4.3.3.1 Mätningar

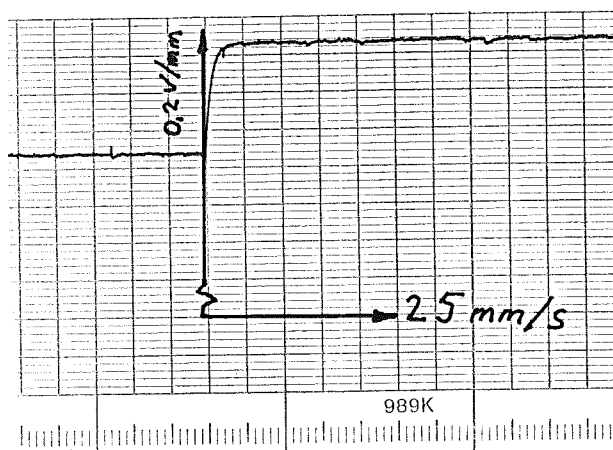
Liksom för det förra systemet har jag med detta systemet, se fig. 4.8, testat olika förstärkningar. Eftersom förstärkningen i det totala systemet är högre här än vad återkopplingen enl. fig. 4.7 gav, använder jag något lägre börvärden. Dessa är $n_{r1} = 1500$ och $n_{r2} = 2000$. Detta ger ett steg $n_r = 500$. Resultaten syns i fig. 4.12 ($k_1 = 4$), 4.13 ($k_1 = 8$) och 4.14 ($k_1 = 16$).



Figur 4.12. Stegsvär då $k_1 = 4$.



Figur 4.13. Stegsvär då $k_1=8$.



Figur 4.14. Stegsvär då $k_1=16$.

Precis som i föregående kapitel blev förstärkning-
 en $k_1=4$ den bästa. Liksom tidigare mätningar kan
 man här ta tidskonstanten ur fig. 4.12. Denna gav
 $1/a_1 = 0.088$, dvs $a_1 = 11.4 \text{ s}^{-1}$. Systemet
 har alltså blivit ännu snabbare men reglerar för
 den skull inte stressigare.

4.3.3.2 Teoretiska beräkningar

Stegsvaret blir med nuvarande återkoppling

$$n = \frac{k_1 b}{s + (a + k_1 b)} \cdot \frac{n_r}{s} \quad (4.3.f)$$

Tidskonstanten är alltså fortfarande densamma

$$a_1 = a + k_1 b \quad (4.3.g)$$

Ur figuren fick jag ju en lägre tidskonstant än tidigare fastän sambandet är lika. Detta kan bero på att parametrarna a och b inte längre stämmer pga att när steget läggs på blir styrsignalen mycket större, tack vare regleringen, än den var vid öppen loop mätningen. I och med detta kommer man längre ut i det olinjära området hos duty cyclen, se fig. 3.18 i kap 3. Jag testar därför tidsfunktionen

$$n = n_r \frac{k_1 b}{a + k_1 b} (1 - e^{-(a+k_1 b)t}) \quad (4.3.h)$$

mot fig. 4.12.

De teoretiska värdena som (4.3.h) gav ligger hela tiden ca 5 % under värdena i figur 4.12. Jag testade med att anpassa parametrarna i (4.3.h) efter figur 4.12 och detta gav

$$a = 1.36 \text{ s}^{-1}$$

$$b = 2.20 \text{ s}^{-1}$$

De har alltså förändrats en del. Nu fås endast en avvikelse på ca max 0.8 % och ingen i det stationära tillståndet därför att jag anpassade värdena.

Anledningen med denna återkoppling som jag har här, se fig. 4.8, var att få bort den stationära hastighetens beroende av förstärkningen. Detta syns tydligt i fig. 4.12-4.14. Den enda skillnaden efter jämvikt beror nu bara på att reglerfelet minskar vid ökande förstärkning. Vid stationaritet får vi (4.3.h) till

$$n(\infty) = n_r \cdot \frac{k_1 b}{a + k_1 b} \quad (4.3.i)$$

och eftersom $b \cdot k_1 > a$ kan vi göra en approximation av (4.3.i) till följande

$$n \approx n_r \quad (4.3.j)$$

4.3.4 Slutsatser

Ur mätningarna med P-reglerad hastighet kan man se att systemet är bestämdare, variationen i stationär hastighet mycket liten och snabbare, tidskonstanten mindre än vid öppen loop styrning. Det system som är att rekommendera är där förstärkningen sker efter återkopplingen, se fig. 4.8. Hur stor förstärkningen skall vara beror på tillämpningen, men vid stegsvarsmätningen ovan var $k_1=4$ den bästa förstärkningen för ett långt men snabbt system.

4.4 Lägesåterkoppling

4.4.1 Allmänt

Nu är det dags att lägga på lägesåterkopplingen. Detta innebar dock problem. Programmet hann inte genomlöpas inom en sampelperiod. Så som programmet är uppbyggt medförde det att samplingstakten minskades. Men programmet är beroende av korrekt samplingsperiod för att den skall hinna med att läsa av lägesgivaren. Jag går först igenom vad mätningarna gav, nedan i kap 4.4.2, och försöker sen ge en förklaring i kap 4.4.3.

4.4.2 P- och PI-reglering av läget

Vid P-regleringen använde jag en algoritm enligt fig. 4.15.

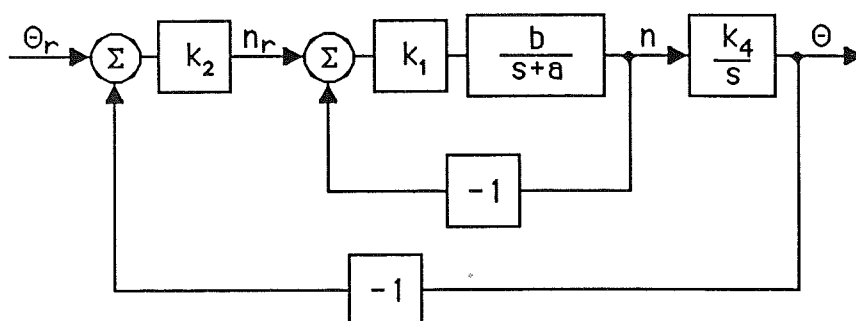


Fig. 4.15. Algoritm vid P-reglering av läget.

Vid PI-regleringen enligt fig. 4.16.

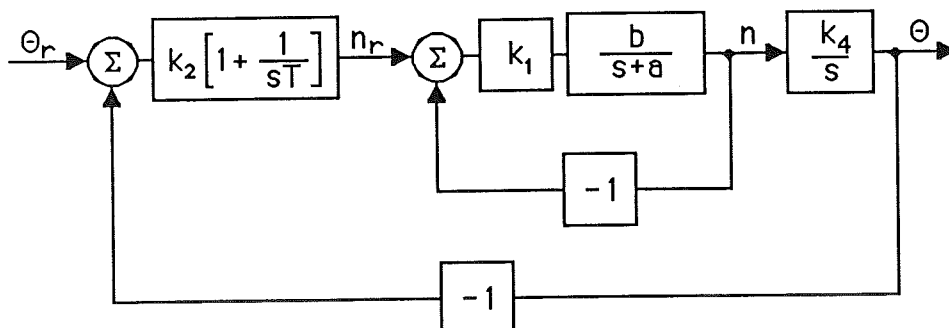


Fig. 4.16. Algoritm vid PI-reglering av läget.

Vid P-regleringen får man en överföringsfunktion enligt följande

$$\Theta = \frac{k_1 \cdot k_2 \cdot k_4 \cdot b}{s^2 + (a + k_1 \cdot b)s + k_1 \cdot k_2 \cdot k_4 \cdot b} \cdot \Theta_r \quad (4.4.a)$$

Identifierar vi nämnaren med

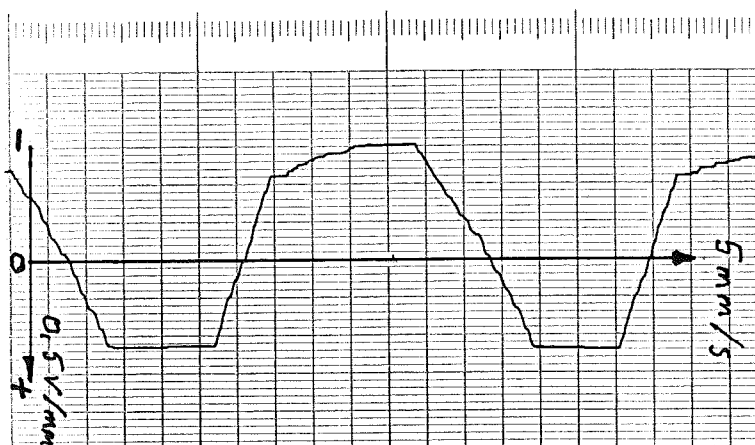
$$s^2 + 2\omega_o \xi s + \omega_o^2 \quad (4.4.b)$$

så får vi

$$k_2 = \frac{\left(\frac{a+k_1 \cdot b}{2\xi}\right)^2}{k_1 \cdot k_4 \cdot b} \quad (4.4.c)$$

I ett reproducerande system skall $\xi = 0.7$. Detta ger $k_2 = 0.030$. ($k_1=4$ som i kap 4.3 och $k_4=200$).

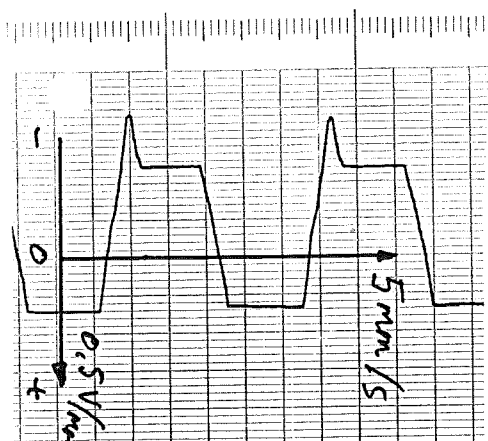
Stegsvarsmätning gav resultat enligt fig 4.16. Jag har tagit läget i programmet och kört ut det via DAC:en för att kunna registrera det på remsan nedan.



Figur 4.16. Stegsvaret då lägesförstärkningen $k_2 = 0.030$.

Jag testade vid en del andra förstärkningar också, men som man ser i fig. 4.16 ovan fungerar inte regleringen tillfredsställande. Börvärdena var lika stora i både + och - riktning. (30000 och ett varv på motorn motsvarar 200). I verkligheten snurrar plattan mer åt det positiva hållet än åt det andra. Fig. 4.16 ger dock sken av att det roterar längst mot det negativa hållet. Tydligt klarar inte datorn av att hålla reda på läget vid samplingen på 2 kHz.

Jag gjorde liknande mätningar med integraldelen inkopplad. Först gjorde jag beräkningar som vid P-regulatorn för att få lämpliga parametrar och testa sen stegsvaret. Resultatet visas i fig. 4.17.



Figur 4.17. Stegsvaret då integraldelen kopplades in.

Här spårade regleringen fullständigt ut. I fig. 4.17 ges också sken av att motorn snurrade längre åt det negativa hållet och dessutom med en stor översläng åt det hållet. Men överslängen var liten och motorn snurrade längst åt det positiva hållet. I det följande kapitlet skall jag försöka ge en förklaring av skeendet.

4.4.3 Sammanfattning

Som föregående kapitel tydligt visar i figurerna så fungerar regleringen inte alls. Anledningen till detta ligger nog till stor del i hur man håller reda på läget. I appendix A figurerna A7 och A8 beskriver jag hur jag har löst problemet med att räknaren endast har 8 bitar. Detta sätt att realisera det på kräver att räknaren inte räknar för långt mellan varje sampling. Absolut inte mer än ett varv (256 pulser). Om programmet hade

hunnit med samplingstakten (2 kHz) skulle detta inte vara något problem, men den effektiva samplingshastigheten låg nu omkring 500 Hz. Eftersom ovanstående stora reglering inte fungerade gick jag över till att testa med börvärdesstyrning av den enkla modell jag hade i kap 4.3, se fig. 4.8. Dessa tester har jag kommenterat i nästa kapitel. Ovanstående mätningar visar att genomlöpningstiden hos en regleralgoritm är av högsta betydelse för en bra reglering.

4.5 Börvärdesstyrning med P-reglerad hastighet

4.5.1 Allmänt

Precis som i kap 4.3 använder jag här en modell av systemet enligt figur 4.18.

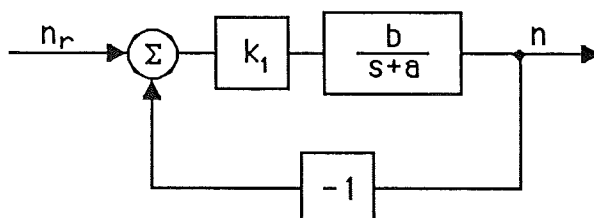


Fig. 4.18. System med P-reglerad hastighet.

Men i kap 4.3 testade jag systemet genom att lägga på steg. Här har jag bl a förenklat programmet för att snabba upp det lite. Det ser nu ut som i figur A2 i appendix A. Dessutom har jag förändrat möjligheterna vid utläggandet av börvärdet. Jag kan nu lägga ut det enligt figur A5 i appendix A. Hur stor förstärkning jag kan lägga på testar jag i kap 4.5.2.

Rampen som läggs ut repeteras, se figur A5, så att jag kan mäta hur fort systemet kan hänga med vid

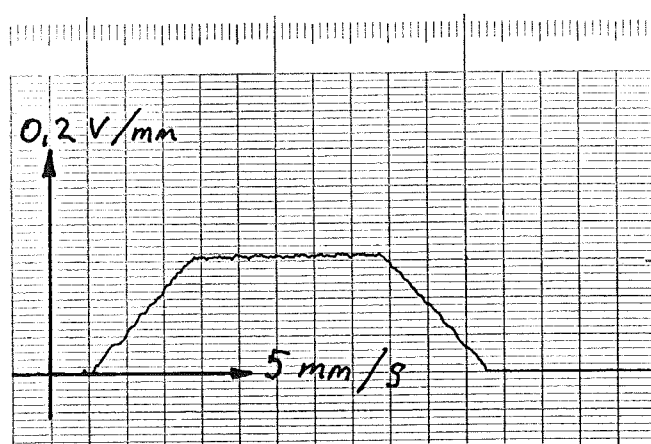
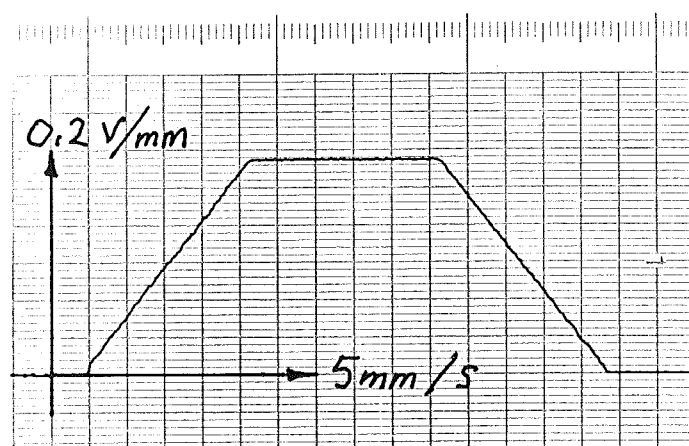
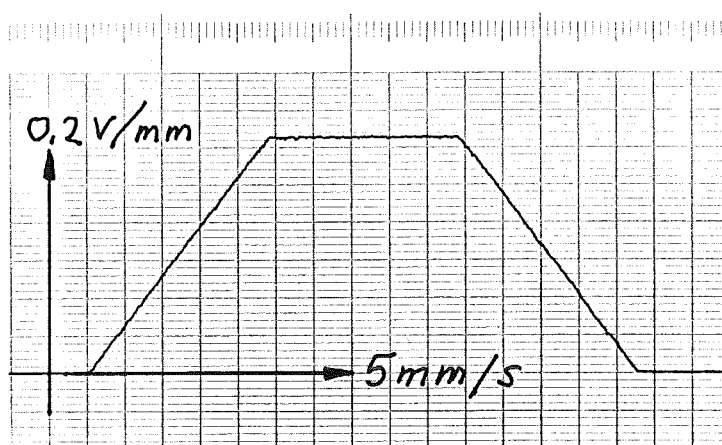
acceleration och retardation. Dessa mätningar finns i kap 4.5.3. Vid frekvensmätningarna i kap 3 bestämde jag en lägsta hastighet vid öppen loop styrning, men i kap 4.5.4 testar jag hur sakta man kan köra vid reglering. Jag testar dessutom hur stort reglerfelet är.

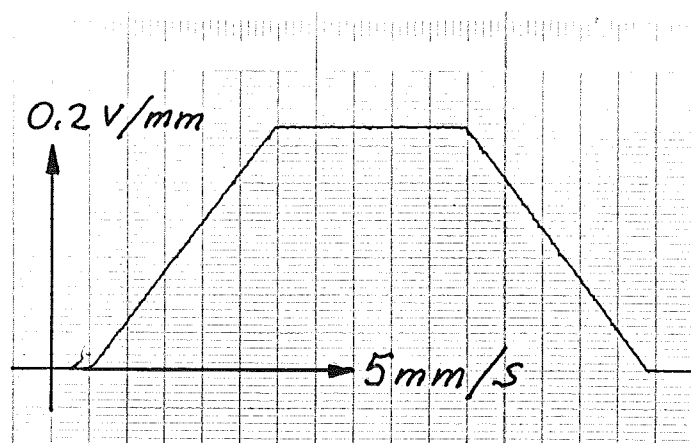
Till slut testar jag i kap 4.5.5 om valet av frekvens inverkar på regleringen. I kap 4.5.6 sammanfattar jag vad mätningarna har gett i kap 4.5. Vid alla mätningar i följande underkapitel är det taco-signalen som jag har mätt och redovisat på skrivarpapperet.

4.5.2 Tester för lämplig förstärkning

Eftersom det är en bra förstärkning som jag vill åt så har jag satt $acc = ret$ låga, i figur A5. Accelerationen testar jag ju i nästa kapitel. Acc lät jag vara 0.1 rpm/0.5 ms. Det innebär att jag la till 0.1 rpm på hastighetsbörvärdet vid varje sampling. Samplingsperioden liksom perioden för PWM-signalen är här 0.5 ms, dvs 2 kHz frekvens. Jag lät motorn börja från stillastående och lade på en ramp på börvärdet från 0 till 1000 rpm. Efter en tids reglering vid konstant hastighet lade jag på en omvänd ramp för retardation ner till 0 igen.

Mätningarnas resultat finns i fig 4.19-4.22 nedan. Det enda som skiljer mätningarna åt är förstärkningen och den är $k_1=2.5$ i fig 4.19, $k_1=10$ i fig 4.20, $k_1=20$ i fig 4.21 och $k_1=30$ i fig 4.22.

Figur 4.19. Taco-signal då $k_1=2.5$.Figur 4.20. Taco-signal då $k_1=10$.Figur 4.21. Taco-signal då $k_1=20$.



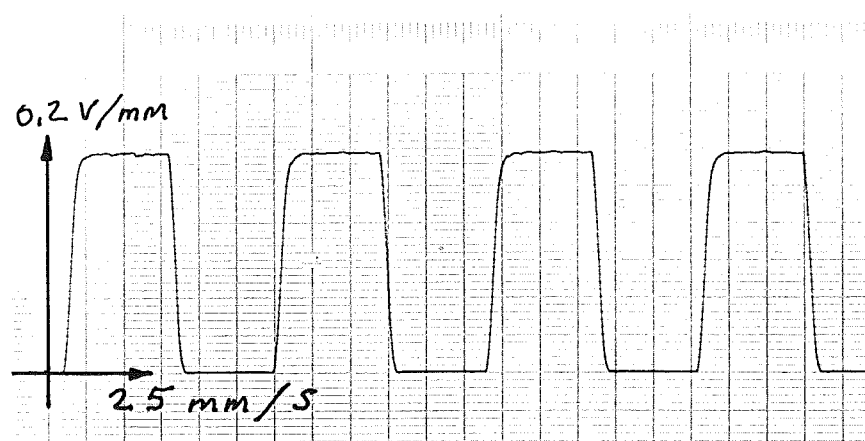
Figur 4.22. Taco-signal då $k_1=30$.

Man ser i fig 4.19 att $k_1=2.5$ är en för liten förstärkning för att nogga kunna följa accelerationen. Med $k_1=10$ blir det bättre men den hänger inte riktigt med i starten. Det är först vid $k_1=20$ och $k_1=30$ som motorn följer börvärdet bra. Då följer motorn nogga även vid start och slut av rampen. Vid $k_1=30$ är dock PWM-signalen ut till motorn något stirrig. Den slår ibland back och ibland fullt fram för att kunna följa nogga. Det är alltså $k_1=20$ som ger den bästa regleringen men ändå med en lugn utsignal. Denna ($k_1=20$) förstärkning använder jag i fortsättningen av testerna i de följande underkapitlen, såvida jag inte anger något annat.

4.5.3 Bestämmande av maximal acceleration

I detta kapitel testar jag inte bara accelerationen, överskriften till trots, utan mäter också vilken maximal retardation systemet klarar. Med maximal menar jag att motorn skall med regleringens hjälp följa med börvärdet. Den skall inte hänga efter som vid ett stegsvar, dvs följa någon exponentialkurva. Precis som vid förstärkningsmätningen pendlar börvärdet mellan 0 (stillastående) och 1000 rpm. Pulsfrekvensen är fortfarande 2 kHz och förstärkningen $k_1=20$.

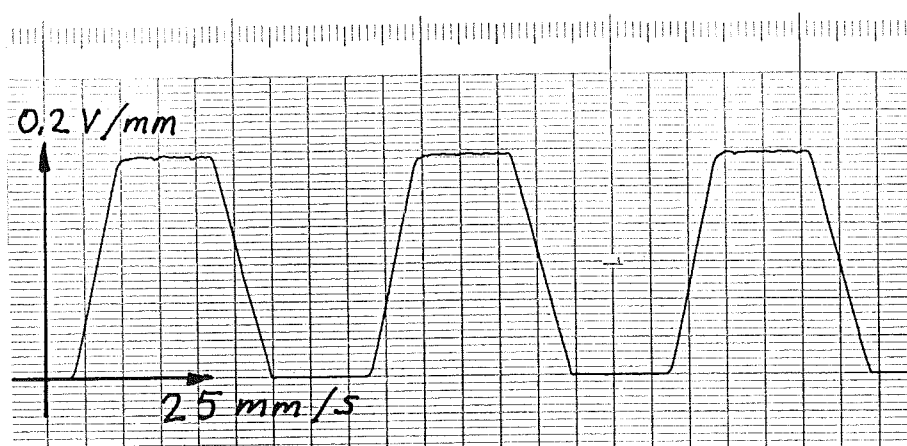
Hur motorn reagerade för de olika accelerationerna (retardationerna lika med respektive acceleration) ser vi på den registrerade taco-signalen i fig 4.23-4.26. Accelerationen var 10 rpm/0.5 ms i fig 4.23, 5 rpm/0.5 ms i fig 4.24, 2 rpm/0.5 ms i fig 4.25 och 1 rpm/0.5 ms i fig 4.26.



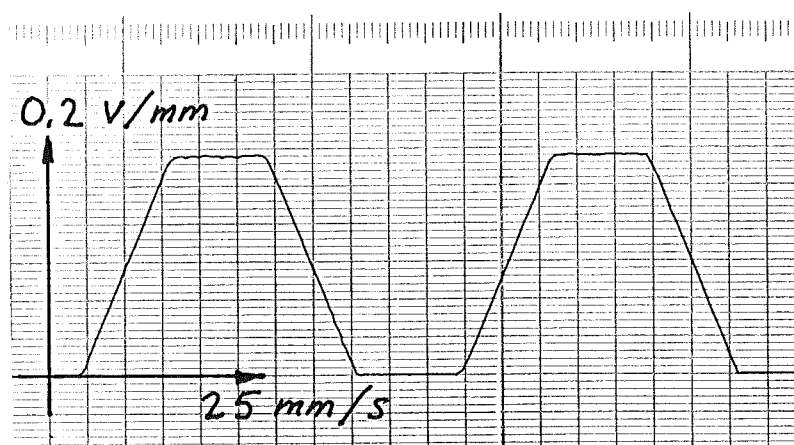
Figur 4.23. Taco-signal då accelerationen var 10 rpm/0.5 ms.



Figur 4.24. Taco-signal då accelerationen var 5 rpm/0.5 ms.



Figur 4.25. Taco-signal då accelerationen var
2 rpm/0.5 ms.



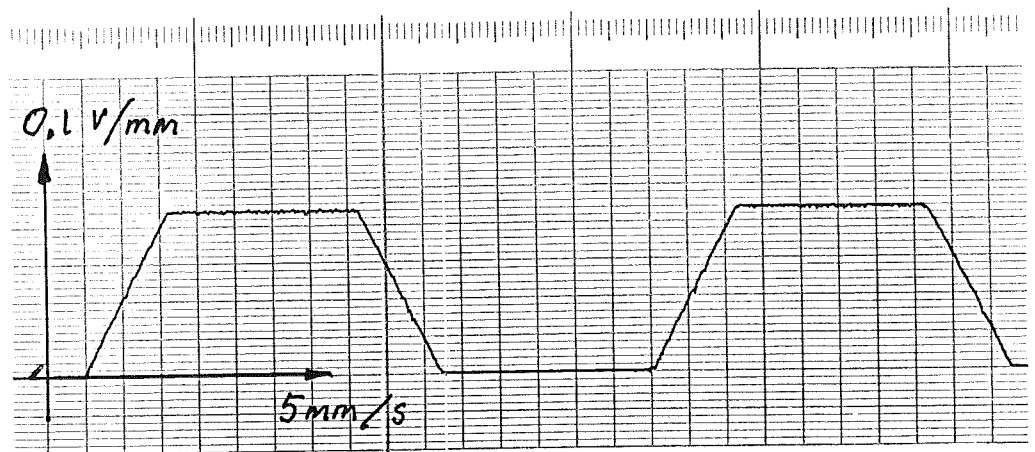
Figur 4.26. Taco-signal då accelerationen var
1 rpm/0.5 ms.

Man kan se i figurerna 4.23 och 4.24 att motorn inte klarar kriteriet med att följa börvärdet. I fig 4.25 klarar motorn av att följa med vid retardationen och först i fig 4.26 klarar den av accelerationen. Maximal retardation är alltså 2 rpm/0.5 ms och uttryckt i SI-enheter blir det 66.7 varv/s^2 . Den maximala accelerationen blev 1 rpm/0.5 ms och i SI-enheter får vi 33.3 varv/s^2 .

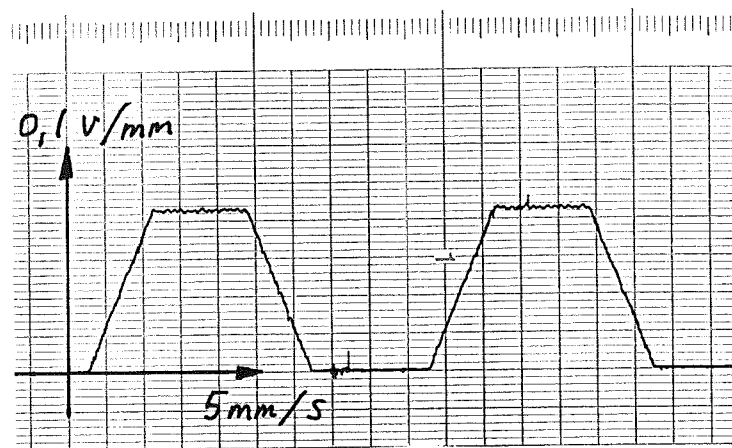
4.5.4 Lägsta hastighet vid reglering och reglerfelet

I kapitel 3 testade jag den lägsta hastigheten vid öppen loop körning. Här ser jag hur sakta man kan köra då hastigheten P-regleras. Jag lägger på börvärdet likadant som jag har gjort ovan, se figur A5 appendix A. Motorn börjar vid stillastående och styrs av en ramp upp till det maximala börvärdet. Det är detta maximala värdet som anges vid figurerna nedan. Det är vid detta värde jag ser hur jämnt motorn går. Förstärkningen k_1 (se fig 4.18) är hela tiden 20, den bästa förstärkningen i kap 4.5.2, utom i fig 4.34 där den är 40. Jag har höjt den här för att se hur mycket regleringen förbättras vid låga hastigheter.

I de fem följande figurerna har jag samma gradering på tidsaxeln. Jag börjar med börvärdet 500 och testar vid varje lägre hundratal. Man ser också på figurerna att efterhand som jag har mindre dämpning i y-led (dvs mindre dämpning på skrivarrens ingång), blir störningarna mer synliga. Dessa ses lättast där motorn har stannat, det lägre värdet i figurerna. Här får vi utslag fastän motorn står stilla. I fig 4.31 syns detta tydligt. Detta inverkar också då motorn roterar vid sitt högsta värde. Motorn går alltså inte riktigt så ojämnt som det ser ut i figurerna fr o m fig 4.31. När man ser i figurerna bör man alltså hela tiden jämföra det lägre med det högre värdet för att kunna uppskatta störningen.



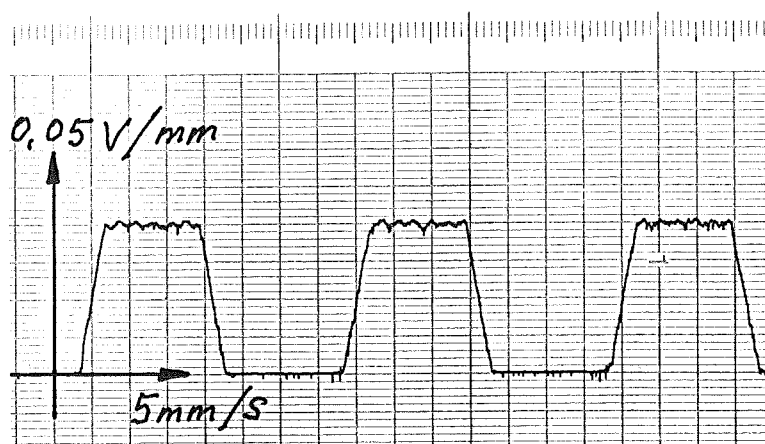
Figur 4.27. Taco-signal då börvärdet in är 500 rpm ($k_1=20$).



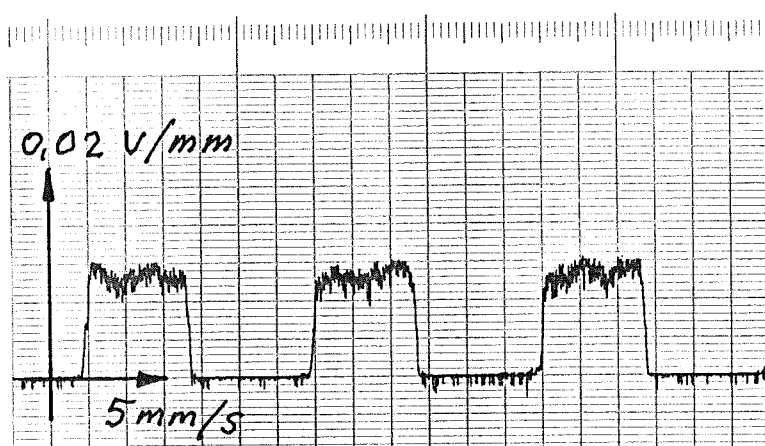
Figur 4.28. Taco-signal då börvärdet in är 400 rpm ($k_1=20$).



Figur 4.29. Taco-signal då börvärdet in är 300 rpm ($K_1=20$).



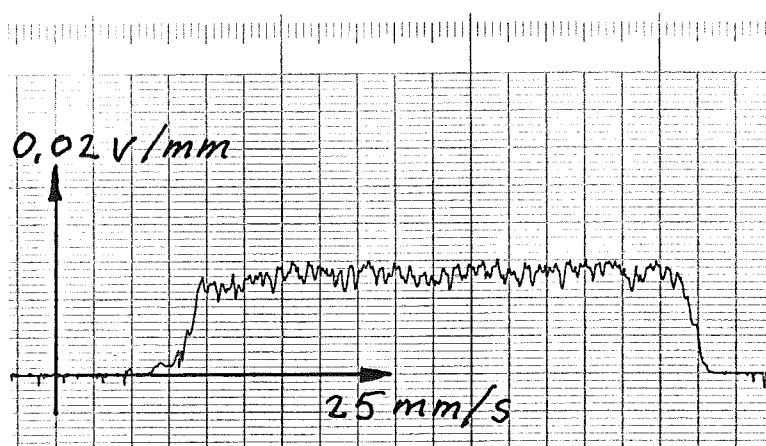
Figur 4.30. Taco-signal då börvärdet in är 200 rpm ($k_1=20$).



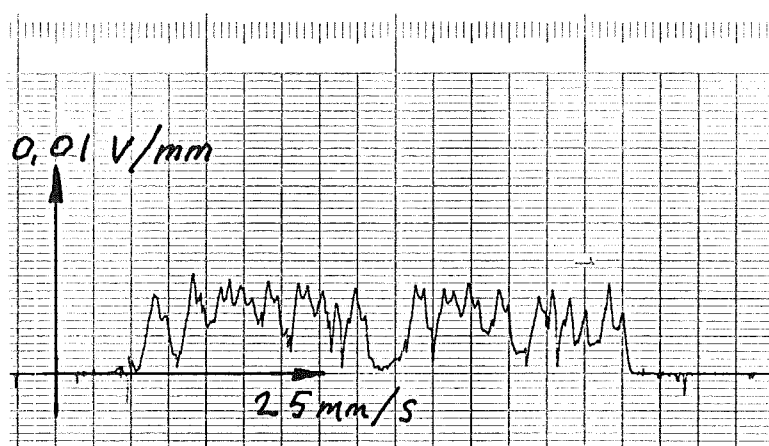
Figur 4.31. Taco-signal då börvärdet in är 100 rpm ($k_1=20$).

Ur figurerna ovan kan man se att 10 % kriteriet (dvs variationen av hastigheten topp till topp skall vara mindre än 10 % av medelvärdet) uppfylls enkelt i fig 4.27-4.30. Vid börvärdet 200 rpm (fig 4.30) ser man dock att det börjar bli svårare att reglera (något ojämn gång). Men det är först vid börvärdet 100 rpm (fig 4.31) som det börjar bli problem med att uppfylla 10 % kriteriet.

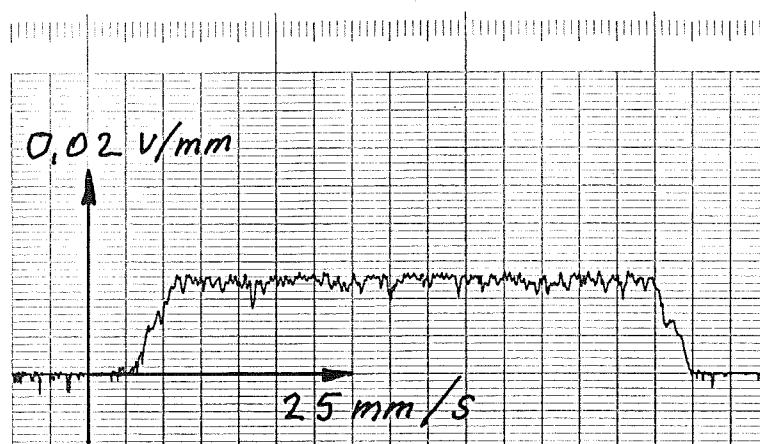
För att lättare kunna urskilja vad som händer i figuren så har jag nedan ytterligare fyra figurer. Här har jag förstorat tidsaxeln och testat en del lägre börvärden (100, 75 och 50 rpm).



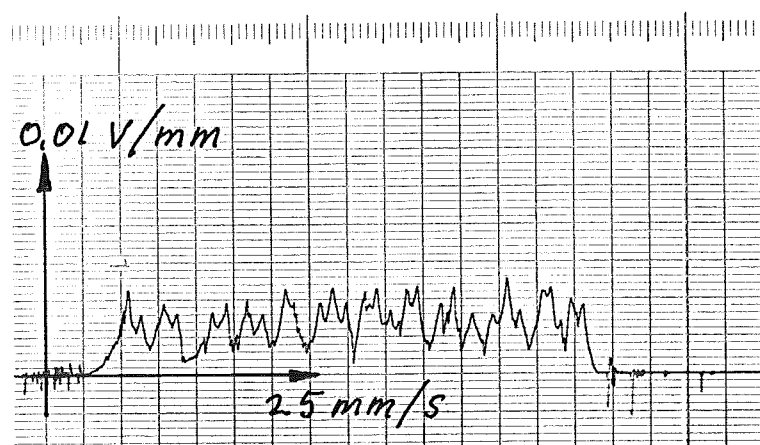
Figur 4.32. Taco-signal då börvärdet in är 100 rpm ($k_1=20$).



Figur 4.33. Taco-signal då börvärdet in är 75 rpm ($k_1=20$).



Figur 4.34. Taco-signal då börvärdet in är 75 rpm
(OBS! $k_1=40$).



Figur 4.35. Taco-signal då börvärdet in är 50 rpm
($k_1=20$).

Man kan nu se i fig 4.32 att motorn inte längre klarar 10 % kriteriet, men det blir ännu värre i fig 4.33 och 4.35 (börvärdena 75 och 50). Jag testade därför med att dubbla förstärkningen i fig 4.34. Här kan man se att motorn nu går lika fort med börvärdet 75 som med börvärdet 100 i fig 4.32, där förstärkningen endast var 20.

Om inte en konstant hastighet är det absolut viktigaste utan det primära ligger i att nå en viss position. Då kan man naturligtvis ta sig dit med den hastighet man har i fig 4.35 vid finjusteringen.

Ovan har jag bara angett vilket börvärde som behöves för att klara av 10 % kriteriet. Ur figuren kan man också utläsa vad det innebar i aktuell hastighet, dvs ärvärdet. Som lägsta varvtal att klara kriteriet har vi börvärdet 200 rpm i fig 4.30. Figuren ger oss att motorn vid maximal rotation har ett varvtal på 167 rpm. Tänjer vi något på kriteriet och fördubblar förstärkningen så får vi att vid börvärdet 75, i fig 4.34, roterar motorn med varvtalet 42 rpm, vilket är avsevärt lägre än vid den öppna styrningen i kapitel 3.

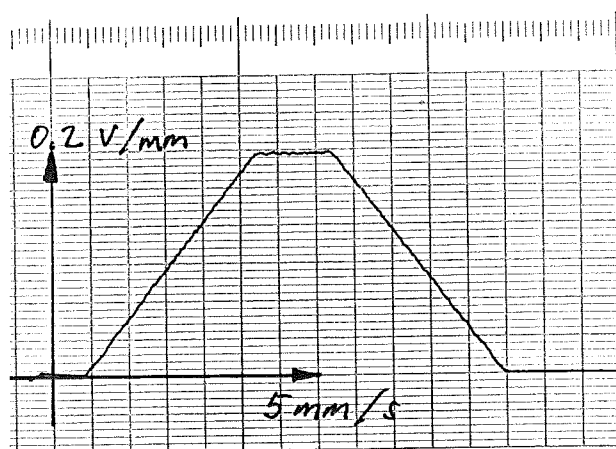
Nu tänkte jag gå över till att se hur stort reglerfelet var. Detta därför att man ur figurerna 4.27 till 4.31 kan utläsa hur stor den aktuella hastigheten är.

Vid mätningarna i de tidigare kapitlerna använde jag ett börvärde på 1000 rpm. Det visade sig att där var reglerfelet minst, 8 %, och det ökade sedan successivt för minskande börvärde. Reglerfelet ligger mellan 12 % och 18 % i figurerna 4.27 till 4.30. Vid de lägre hastigheterna ökar reglerfelet dramatiskt. Detta beror bl a på att friktionen gör sig mer påmind där. I fig 4.31 (100 rpm) har vi ett reglerfel på 57 % och i fig 4.33 är det inte längre lönt att tala om reglerfelet då hastigheten pendlar så kraftigt. När jag ökade förstärkningen, se fig 4.34, erhöll jag ett medelreglerfel på 44 %.

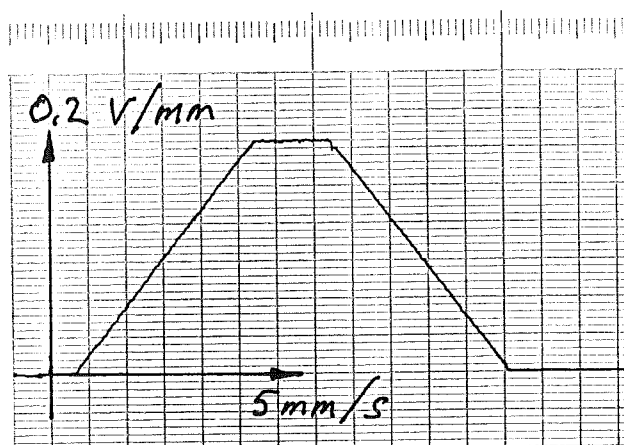
4.5.5 Test av olika frekvenser

Till sist gjorde jag också några mätningar för att se om valet av frekvensen på PWM-signalen påverkade hur systemet uppförde sig vid reglering. Jag testade vid frekvenserna 1, 2 och 4 kHz.

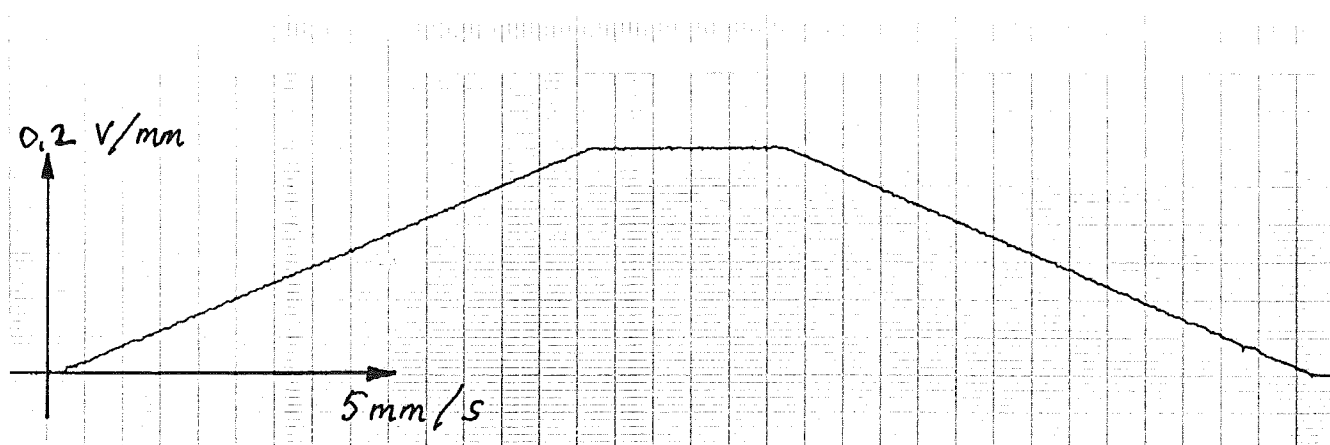
Dessa mätningar återfinns nedan.



Figur 4.36. Taco-signal då frekvensen på PWM-signalen var 1 kHz.



Figur 4.37. Taco-signal då frekvensen på PWM-signalen var 2 kHz.



Figur 4.38. Taco-signal då frekvensen på PWM-signalen var 4 kHz.

I figurerna ovan finns ingen märkbar skillnad i uppförandet. Valet av frekvens på PWM-signalen är alltså inte avgörande för systemets uppförande. Vid mätningen i fig 4.38 hände dock något egendomligt. Börvärdet var vid denna mätning likadant som vid de två föregående mätningarna, förutom att jag pga heltalsräkningarna i detta förenklade program var tvungen att ha dubbelt så stor acceleration på rampen. I fig 4.38 kan man dock se att accelerationen i själva verket blev lägre än innan. Detta berodde på att programmet inte hann med vid 4 kHz, därför att jag lät sampelhastigheten, dvs realklockans frekvens, vara lika med PWM-signalens frekvens. Därmed kördes min interrupt-rutin flera gånger innan programmet hade genomlöpts för att lägga ut ny utsignal. Eftersom min interruptrutin, pga att den innehöll omvandling av tacons samplade värde till varv per minut (många multiplikationer), tog lång tid förlängdes den effektiva sampelperioden. PWM-signalen, som genererades av egen klocka, fortsatte dock att vara 4 kHz. Kontentan blir här liksom i kapitel 3 att valet av frekvens på

PWM-signalen inte är avgörande ur motorns synpunkt, utan bör väljas så att programmet precis hinner att genomlöpas. Vill man kan man ju låta PWM-signalen och realtidsklockan ha olika frekvens. Mer om detta står i appendix C.

4.5.6 Slutsatser

Ur testerna med börvärdesstyrning kan man se att systemet uppför sig mycket bra. $k_1=20$ var en förstärkning som jag kom fram till var lämplig vid mätningarna i kap 4.5.2. I kap 4.5.4 kunde man dock se att vid mycket låga hastigheter krävdes en högre förstärkning för att variationen inte skulle bli för stor. Men skall man bara reglera sig fram till ett läge kan man göra detta mycket noga även om hastigheten varierar.

Mätningarna i kapitel 4.5.3 visar att mycket stora accelerationer kan läggas på börvärdet utan att systemet släpar efter som vid ett stegsvar. I kap 4.5.5 kom jag fram till samma resultat som i kap 3. Valet av frekvens på PWM-signalen och realtidsklockan bör väljas efter hur snabbt regleralgoritmen hinner med.

4.6 Slutsatser av regleringarna i kapitel 4

I kap 4.2 började jag med att köra systemet i öppen loop. Dessa mätningar gav att en approximation av systemet med en första ordningens överföringsfunktion stämde bra. När jag kopplade på P-regulatorn i kap 4.3 blev systemet snabbare och bestämdare. Efter anpassning av parametrar kan jag bara konstatera att systemet uppför sig som teorin säger. Att motorn styrs av en PWM-signal har alltså ingen negativ inverkan. I kap 4.4 försökte jag lägesåterkoppla. Tyvärr var programmet för

stort och klarade inte av att genomlöpas mellan samplingarna. För att få ut lite mer om systemet då det regleras och styrsignalen är pulsbreddsmodulerad testar jag med börvärdesstyrning i kap 4.5. Jag låter P-regleringen från kap 4.3 vara kvar.

Jag lägger här på en något större förstärkning eftersom utsignalen inte bottenar lika lätt, tack vare att inga steg läggs på. Vid accelerationsmätningarna märker man att systemet klarar av att reglera vid höga accelerationer och inte bara lägga på full fram för att hänga med. Systemet klarar en högre retardation än acceleration, och detta beror på att vid inbromsning tar jag inte bara bort spänningen utan kan lägga på full spänning fast motpolad. Naturligtvis klarar systemet, här vid P-reglering, en mycket lägre lägsta hastighet än i kap 3, där mätningarna gjordes vid öppen loop. Tolererar man en ojämn gång vid de låga hastigheterna kommer man ner i mycket små varvtal och kan reglera sig noga i läge.

Jag testade reglerfelet samtidigt som jag gjorde mätningarna om vilken lägsta hastighet systemet klarar. Ur dessa tester kan man se att när man sänker hastigheten på motorn börjar det procentuella reglerfelet öka för de lägsta hastigheterna. Till slut gjorde jag försök med att reglera vid olika frekvenser på PWM-signalen. Någon skillnad i uppträdandet har inte kunnat upptäckas.

5 MÄTNINGAR PÅ SWITCHDEL

5.1 Allmänt

På kraftkortet, beskrivet i appendix D, har jag gjort en del mätningar för att se hur det förmedlar MCU-kortets ideala PWM-signal till motorn. I fig 5.1 nedan har jag gjort en skiss över kraftkortets huvudtransistorer för att beskriva i vilka punkter jag mätte.

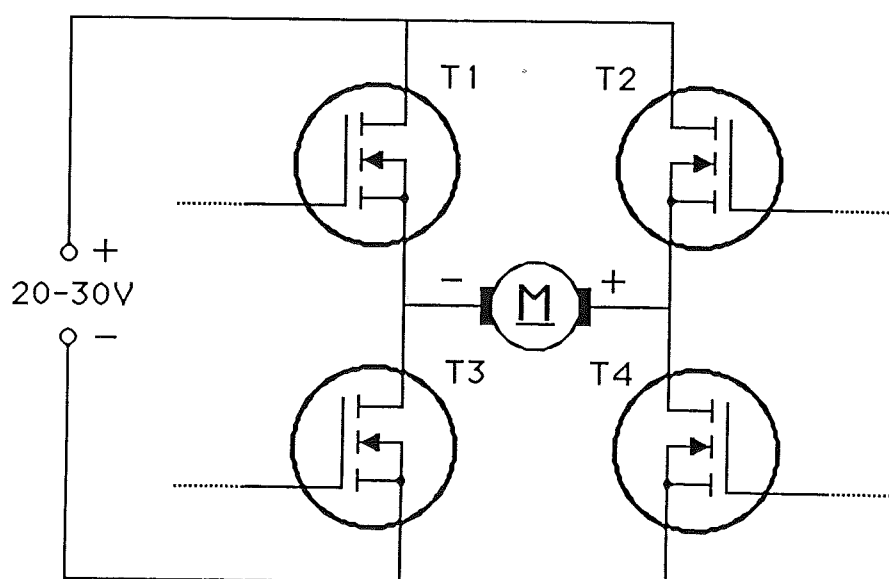
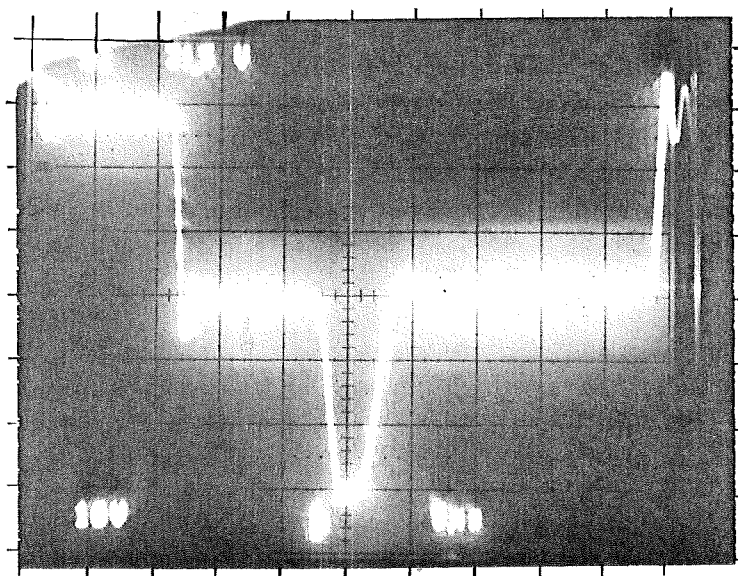


Fig. 5.1. Skiss av kraftkortet där endast krafttransistorerna är utritade.

Den fullständiga ritningen finns i figur D1 i appendix D. Med mätningarna i kap 5.2 lägger jag fram en möjlig förklaring till olinjäriteten i duty cyclen relaterad till varvtalet vid frekvensmätningarna i kap 3. De enda tillförlitlighetsproblemen jag upptäckte i systemet fanns hos kraftkortet. Detta och resultaten från mätningarna går jag igenom i kap 5.3.

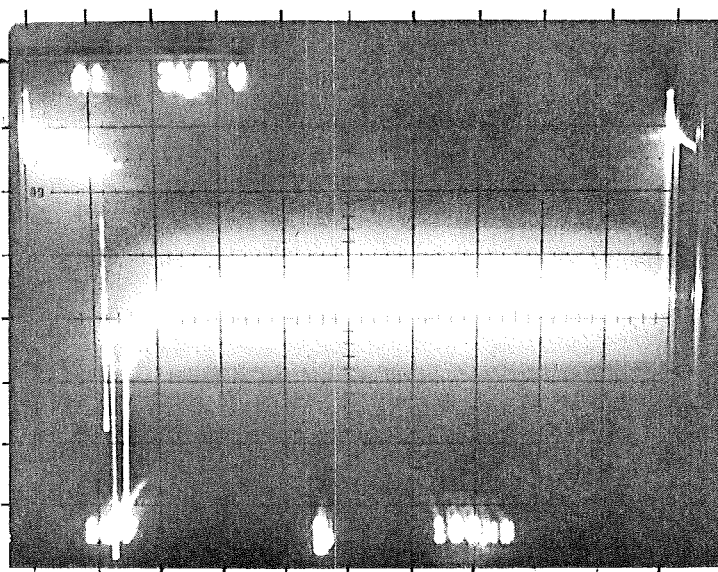
5.2 Mätningar

När jag mätte spänningen över motorns poler vid PWM-styrning fick jag ett resultat som i fig 5.2. Här är pulsfrekvensen 20 kHz.



Figur 5.2. Spänning över motornspoler vid en pulsfrekvens på 20 kHz.

Vid en mätning då pulsfrekvensen var 1 kHz blev resultatet som i fig 5.3.

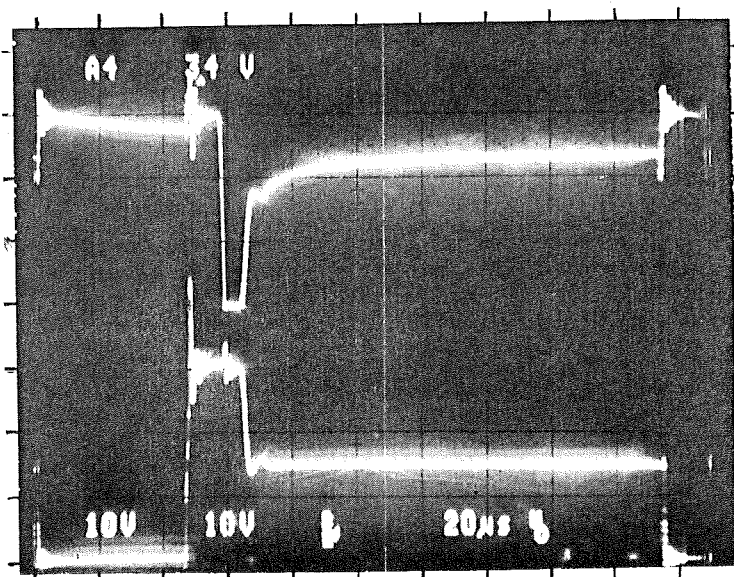


Figur 5.3. Spänningen över motorns poler då pulsfrekvensen var 1 kHz.

Det man hade önskat sig att finna var en fyrkants-signal med en viss förutbestämd duty cycle. Som man ser i figurerna ovan fanns dessutom en negativ puls med. Denna uppträder alltid en och samma tidpunkt efter den negativa flanken hos PWM-signalen. I figurerna kan man se att det är ca

10 μ s. Anledningen står att finna i en fördröjning i stängningen hos transistorerna.

Nu körs motorn så att T1 och T4, se fig 5.1, hela tiden är stängda medan T2 och T3 står för pulsningen av motorn. I fig 5.4 nedan har jag dels mätt spänningen hos motorns positiva pol relativt jord (den övre strålen i figuren) och den negativa relativt jord (den undre strålen).

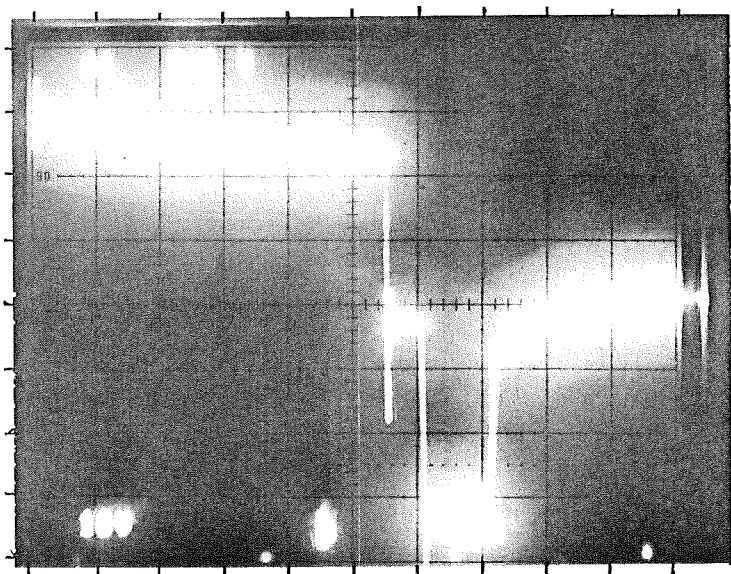


Figur 5.4. Spänningen hos motorns positiva pol (övre strålen) och den negativa (undre strålen) relativt jord vid 5 kHz puls-frekvens.

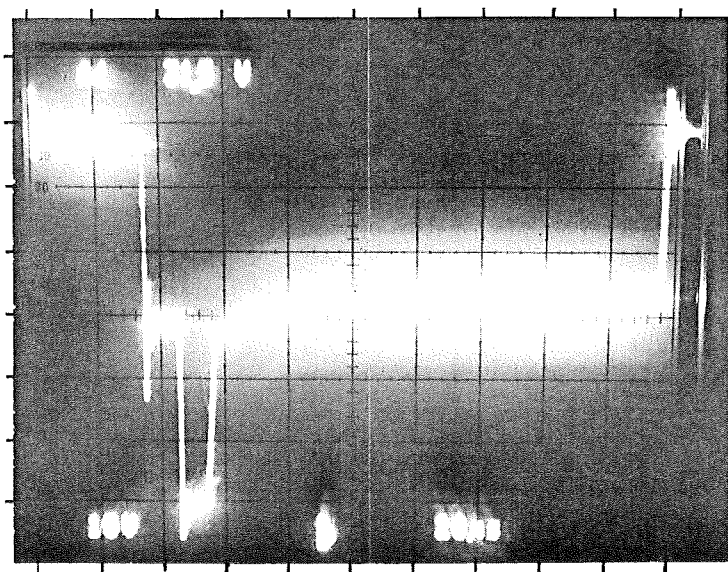
Här kan vi se att transistorerna T2 och T3 inte bryter spänningen samtidigt. Transistor T3 bryter först. Detta gör att motorns negativa pol går upp på samma potential (30 V) som den positiva, eftersom strömtillförseln har stängts. I fig 5.2 ser man detta i att spänningen över motorns poler går från 30 V ner till 0 V. När sen T2 stänger, 10 μ s senare än T3, sjunker spänningen över den positiva polen hos motorn, se övre strålen i fig 5.4. Av någon anledning går nu potentialen där ner till 0 V samtidigt som potentialen vid motorns negativa pol fortfarande är 30 V. Detta innebär att motorn får en negativ spänning på 30 V över sig. Detta är den negativa pulsen i fig 5.2. Sedan

inträder ett jämviktstillstånd där T2 och T3 delar på matningsspänningen så att motorn återigen får 0 V över sig.

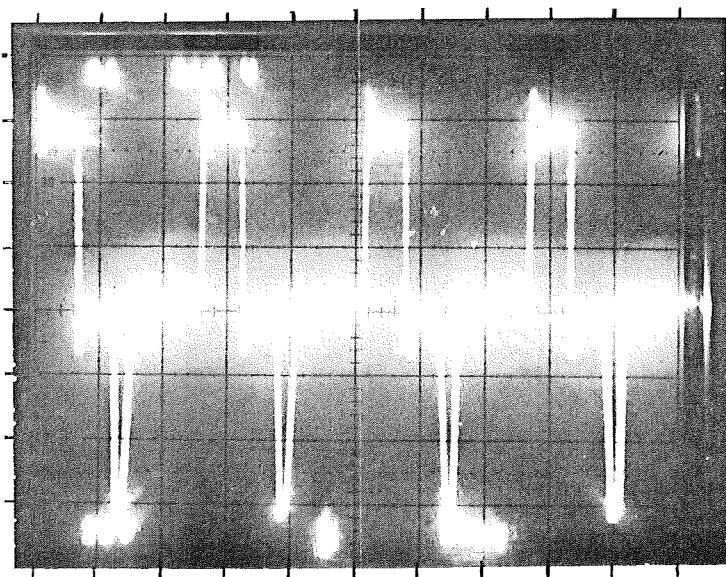
För att se om pulsens utsträckning påverkades av pulsfrekvensen gjorde jag 3 mätningar vid olika frekvenser och hade tidbasen konstant på 20 μ s/ruta.



Figur 5.5. Spänningen över motorns poler då pulsfrekvensen var 1 kHz.



Figur 5.6. Spänningen över motorns poler då pulsfrekvensen var 5 kHz.



Figur 5.7. Spänningen över motorns poler då puls-frekvensen var 20 kHz.

I figurerna ovan ser vi att den negativa pulsen är längre vid 1 kHz än 5 kHz och att den är längre vid 5 kHz än 20 kHz. Men det är inte frekvensen i sig själv som bestämmer pulsens längd, utan det är den tid som transistorerna T2 och T3 har varit till (dvs då det ligger 30 V över motorn) som bestämmer längden. Vid 1 kHz måste transistorerna vara till längre än vid 20 kHz för att uppnå en ekvivalent duty cycle. Om vi t ex vill ha en duty cycle på 10 % måste transistorerna vara till 0.1 ms vid 1 kHz medan de bara är till 5 μ s vid 20 kHz. Alltså ju längre tid transistorerna T2 och T3 är till ju längre negativ puls. När den negativa pulsen sen har uppnått en viss längd blir den inte längre. När sen duty cyclen går upp mot 100 % minskar den sen igen, för att helt försvinna då duty cyclen är 100 %.

Vid frekvensmätningarna i kapitel 3 upptäckte jag också att för högre frekvens behövdes en större duty cycle för att motorn skulle börja rotera. Resultaten ovan ger en direkt förklaring till detta. När man ökar duty cyclen successivt från noll upp till det att motorn startar ökar också

den negativa pulsen upp till en viss gräns. På detta sätt motverkas den positiva pulsen. Man kan säga att den positiva pulsen motverkas av en bestämd längd då den negativa pulsen har uppnått sitt maximum. Men en bestämd pulslängd innebär en större duty cycle vid 20 kHz än vid 5 kHz och därför krävs en större duty cycle för start vid 20 kHz. Vid 1 kHz är den negativa pulsens längd relativt periodlängden för PWM-signalen så liten att den har en mycket liten inverkan vid start. Anledningen till att motorn ändå inte startar förrän vid ca 10 % duty cycle har sin förklaring i att friktionen måste övervinnas.

5.3. Slutsatser

Vid mätningarna ovan kan vi se att en oönskad negativ puls på PWM-signalen uppträder. Denna härrör sig från kraftkortet. Om det är denna puls som orsakar olinjäriteten i duty cyclen relaterat mot varvtalet i mätningarna i kapitel 3, är svårt att säga med enbart de mätningar jag gjort i kap 5.2. Eftersom den uppstår då T2, se fig 5.1, stänger skulle man kunna låta kraftkortet förmedla PWM-signalen på ett annorlunda vis. I stället för att både T2 och T3 stänger och öppnar i takt med PWM-signalen, skulle man kunna låta T2 vara öppen hela tiden och T3 förmedla PWM-signalen. När man sedan backar skulle T1 hela tiden vara öppen och T4 generera PWM-signalen. På detta sätt kanske man skulle kunna få bort den negativa pulsen.

I kapitel 3 kunde man se att för högre pulsfrekvens krävdes det högre duty cycle för att motorn skulle börja rotera. I mätningarna i kap 5.2 kom jag fram till att detta härrör sig från den ovan beskrivna negativa pulsen.

I inledningen av kapitel 5 nämnde jag också att de enda tillförlitlighetsproblemen jag upptäckte fanns hos kraftkortet. Det var inverterarna

(4049:orna, se figur D1 i appendix D) och krafttransistorerna (BUZ 25, dock inritade som BUZ 15 i fig D1) som gick sönder. Som vi kan se på ritningen i fig D1 har 3 stycken inverterare parallellkopplats för att orka driva, men dessa blev ändå efter en tids körning mycket varma och slutade att fungera. På ritningen kan vi se att kraftkortet är konstruerat redan 1981 och man hade inte kraftigare inverterare då. I dag finns det inverterare som klarar att ge betydligt större ström och som skulle kunna ersätta 4049:orna.

Krafttransistorerna T1 och T2 gick sönder då jag hade en kraftig förstärkning i återkopplingen, som orsakade att regleringen gav mycket stora ut signaler. Dels fram och dels broms, dvs back, för att reglera noggrant. Detta orsakade stora strömmar, som i sin tur gav stora effektförluster i T1 och T2 p g a att dessa har en spänning på mellan 3 och 4 V över sig vid tillslag. Anledningen till denna spänning har jag beskrivit i appendix D.

Som jag beskrev ovan finns det nya inverterare, som klarar mer ström, vilka skulle kunna ersätta de befintliga. Men eftersom kraftkorts konstruktionen är så gammal bör man kanske först undersöka om det inte har kommit ut helt nya och bättre konstruktioner som tål mer.

SAMMANFATTNING

Mitt arbete med att undersöka ett system med pulsbreddsmodulering av styrsignalen till en DC-motor kan delas in i två delar. Den första var att testa systemet med olika frekvenser på PWM-signalen, och i den andra undersökte jag hur systemet uppförde sig vid reglering.

Jag började arbetet med en litteratursökning. Denna gav mest artiklar av orienterad art och gav inga direkta ledtrådar till de ovan beskrivna delarna. Därför är arbetet experimentellt inriktat. Litteratursökningens resultat är sammanfattat i kapitel 1.

Testningen av de olika frekvenserna på PWM-signalen är redovisade i kapitel 3. Här testade jag frekvenser från 1 kHz till 40 kHz, var och en med olika duty cykler. Dess jämförde jag med en analog körning. Jag mätte duty cycle på PWM-signal, ström genom motorn och spänning över den vid några olika varvtal. Spänningen ökade linjärt mot varvtalet som sig bör hos en permanent magnet DC-motor. Strömmen ökade också linjärt. Den ström som krävdes för att motorn skulle starta minskade något för ökande frekvens och var störst vid analog körning. Detta kan, bero på den gynnsamma effekt som pulsbreddsmodulering har på friktionen i ett system. Vid höga varvtal var situationen den omvända. Där hade den analoga körningen minst strömförbrukning medan PWM-körningarna hade ungefär samma.

Duty cyclen var inte linjär mot varvtalet. Anledningen till detta kan vara att kraftkortet inte överför MCU-kortets PWM-signal idealt till motorn. Kraftkortets utsignal på motorn har jag gjort mätningar på som är redovisade i kapitel 5. Olinjäriteten minskar för ökande frekvens på PWM-signalen. Vid reglering med en P-regulator, enligt den

modell jag har i kapitel 4, medför olinjäriteten att förstärkningen i systemet minskar för ökande hastighet på motorn. Den minsta duty cycle som krävdes för att motorn skulle börja rotera steg för ökande frekvens. Detta berodde på att kraftkortet inte förmedlade pulsbreddsmoduleringen perfekt till motorn.

Den lägsta hastigheten som motorn kunde ha vid PWM-körning påverkades inte av pulsfrekvensen och någon ytterligare uppvärmning av motorn på PWM kunde inte märkas. Pulsningen medförde att en pipande signal kunde höras vid låga varvtal, den var dock inte högre än att den dränktes av ljudet från motor och utväxlingen vid högre varvtal.

Slutsatserna man kan dra av frekvensmätningarna är att valet av pulsfrekvens inte är kritiskt ur systemets synpunkt. Frekvensen bör alltså väljas så att en regleralgoritm precis hinner att genomlöpas under en pulsperiod, för att reglering av varje puls skall kunna ske.

I den andra delen av mitt arbete där jag undersökte om PWM inverkar på regleringen använder jag hela tiden samma pulsfrekvens, 2 kHz. Jag börjar med att köra i öppen loop för att få fram parametrarna i min modell av systemet. Modellen är av första ordningen. Därefter P-reglerar jag hastigheten och tar fram en så hög förstärkning som möjligt utan att systemet blir stirrigt och utsignalen bottenar. Med ovanstående paramterar och förstärkningen jag fick vid P-regleringen räknar jag fram lämpliga värde på förstärkning och tidkonstant för lägesåterkoppling. Vid P- och PI-reglering av läget uppstod dock problem. Programmet hann inte längre med att genomlöpas under en period av PWM-signalen. Samplingstakten sjönk och därmed hann inte programmet med att sampla lägesgivarens 8-bitars räknare tillräckligt ofta. Detta medförde att regleringen av läget ej blev korrekt.

Eftersom regleralgoritmen blev för lång vid lägesåterkopplingen testade jag istället med börvärdesstyrning av hastigheten, som fortfarande var P-reglerad. Vid föregående regleringar mätte jag systemets stegsvar. Här med börvärdesstyrningen la jag istället på en ramp upp till en viss hastighet och lät sen motorn gå vid det varvtalet. Därefter retarderade jag med en ramp till en lägre hastighet där motorn fick rotera en tid för att sen accelerera igen osv. Eftersom inget steg lades på kunde förstärkningen ökas utan att utsignalen bottnade.

Här testade jag bl a hur stor acceleration och retardation systemet klarar, med förutsättning att hastigheten reglerades hela tiden och inte släpa efter som vid ett stegsvar. Systemet klarade mycket höga accelerationer och retardationer. Retardationerna kunde göras något större än accelerationerna med ovanstående kriterium. Detta beror på att jag inte bara kan stänga av strömmen till motorn utan även pola om den för att snabbare driva ner hastigheten. Den lägsta hastigheten vid reglering var mycket lägre än vid frekvensmätningen, då systemet var i öppen loop. Hastigheten var något ojämn vid de allra lägsta hastigheterna, men gick att få jämnare om förstärkningen höjdes. Även om hastigheten är något ojämn kan man ju reglera noga i läge vid de låga hastigheterna.

Till sist testade jag dessutom några olika frekvenser på pulserna för att se om dessa hade någon inverkan vid regleringen. Här kom jag fram till samma slutsats som vid frekvensmätningarna i öppen loop. Pulsfrekvensen bör väljas så hög att regleralgoritmen precis hinner att genomlöpas. PWM-regleringen är jämförbar med en analog reglering.

För att utföra regleringen använde jag Sandviks MCU-kort, som finns närmare beskrivet i Appendix C. Detta är noga och bra förberett för pulsbreddsmodulering.

REFERENSER

- (1) Patrick F Muir and Charles P Neuman:
Pulsewidth modulation control of brushless DC motors for robotic applications. IEEE Transactions on industrial electronics (USA), IE-32 (1985) 3, 222-229.
- (2) R M Inigo: Electric wheelchair propulsion systems: A comparative efficiency study. IEE Sotheastcon, Raleigh, 1985, 308-311.
- (3) E A Lodochnikov and YE G Orlov: On pulse control system design for electric machines with non-salient pulse width modulator. Electric technology USSR, 3 (1972), 75-88.
- (4) Celso P Bottura and Yaro Burian Jr: Stability of periodic oscillations in proportional chopper of a series motor. 1977 Joint automatic control conference, San Francisco, 1977, 964-969.
- (5) Karl Johan Åström: Reglerteori. Almqvist & Wiksell Förlag AB, Stockholm, 1968. ISBN 91-20-04438-0.
- (6) P D Krut'ko and N A Lakota: Automatic control Algorithms for drive systems. Tekhnicheskaya Kibernetika, No 3, 1983, 151-161.
- (7) Adnan H Al-Anbuky & Kasim M Al-Aubaidy:
Microprocessor based controller for the control of position & speed of D C servo-motor. IECON'84. Control and instrumentation, Tokyo 1984. 595-599, Vol 1.

- (8) Il Hong Suh, Seung Ho Hwang and Zeungnam Bien: Proportional-integral plus bang-bang control of DC servo motors with PWM drives. Bridges between control science and technology, 9th triennial world congress of IFAC, Budapest, 1984. 2809-2813, vol 5.
- (9) G De Maria & P Marino: An adaptive pulse width modulator for control of DC drives. IEEE-IAS 1983, 18th annual meeting, Mexico City 1983, 749-754.
- (10) Fernand G G De Buck: Losses and parasitic torques in electric motors subjected to PWM waveforms. IEEE Transactions Industry applications, USA, IA-15 (1979) 1, 47-53.
- (11) M Paraskeva & M J Werson: Microprocessor control of a brushless DC motor. Conference on drives, motors, controls 84, Brighton Oct 1984, 80-82.
- (12) A Pshaenich: Use pulse width modulation to control dc motors. With this approach, both speed and direction can be controlled by a single potentiometer. Electronic design (USA), 22 (1974) 3, 68-70.
- (13) S V Lifanov and Yu. Ya. Morgovskiy: Thyristor pulse-frequency control systems for electrical machines. Soviet automatic control, 7 (1974) 5, 62-66.
- (14) H A Raphael: Motor control by PLL can be achieved with a microprocessor. Software replaces the phase detector, and the internal clock provides the circuit timing. Electronics design (USA), 23 (1975) 9, 54-57.
- (15) Balarama V Murty: Fast response reversible brushless DC drive with regenerative braking. IEEE-IAS 19th annual meeting, Chicago 1984, 445-450.

- (16) George A Kaufman: High performance servos for machine tools. 2nd biennial int machine tool techn conf, Chicago 1984, 8/117-126, vol 3.
- (17) Jim Williams and Stan Dendinger: Simplify feedback controllers with a 2-quadrant PWM IC. EDN (USA), 28 (1983) 11, 211-222.
- (18) S J Bailey: Semicon servos 1982; Source of motion control logic and power. Control Engineering, November 1982, 71-75.
- (19) S J Bailey: New motor drives can be tailored to loads. Control Engineering, August 1984, 84-88.
- (20) R A Schulz: Improved pulse-width modulated motor control technique. IBM technical disclosure bulletin, 27 (1984) 4B, 2518-2520.
- (21) J V Gac and J Mako: Digital motor speed control. IBM technical disclosure bulletin, 27 (1984) 1B, 464-465.
- (22) C P Russel: A transistor pulse-width modulated controller and permanent-magnet motor. Electronics & Power (GB), 25 (1979) 10, 707-709.

FLÖDESSCHEMA FÖR REGULATORPROGRAM

För att slippa en massa listning av flödesscheman inne i rapporten tar jag upp dem här. Detta appendix är i huvudsak en referens till kapitel 4 (Tester med regulatorer).

Jag har två olika program för dessa tester. I det fall där jag endast hastighetsåterkopplar använder jag ett enklare program för att minska exekveringstiden. Det bör dock nämnas att programmet ej är optimalt, men jag har valt att ha det så här för att öka läsbarheten i programmet och för att underlätta förändringar i programmet under testernas gång.

Båda programmen är uppbyggda på liknande sätt. Se Figur A1.

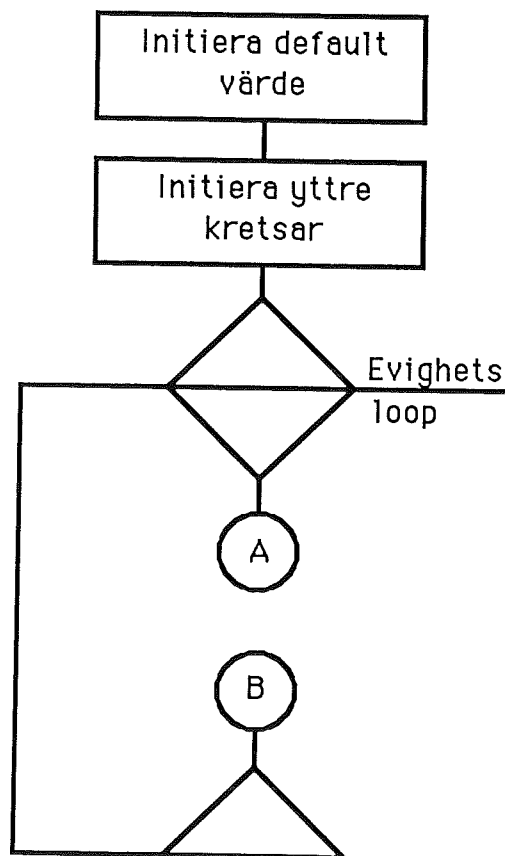


Fig. A1. Huvudprogrammet.

De börjar med att initiera defaultvärde. Här bestäms puls-frekvensen. Realtidsklockan har samma frekvens som puls-frekvensen. Dessutom sätter jag upp börvärde, förstärkning-ar och tidskonstanter för regleringen.

Därefter följer initieringen av de yttre kretsarna. Dessa är en timer, som lägger ut frekvenser, ett interface, som ger värde till DAC och styr lysdioder, och en interrupt-controller, som tar hand om realtidsklockans interrupt. För närmare beskrivning av dessa hänvisas till appendix C.

Sedan kommer själva körningen med motorn. De båda olika programmen börjar i A och slutar i B, (se fig A1) där de går in i evighetsloopen. Jag börjar med att beskriva det program som endast har hastighetsåterkoppling. Se Figur A2 nedan.

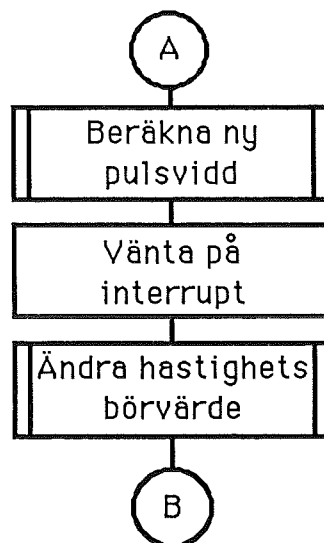


Fig. A2. Program med p-regulator i hastighetsloopen.

De olika subrutinerna beskrivs nedan i Figur A3 och A4. Som man ser i figur A3 nedan, används endast en P-regulator.

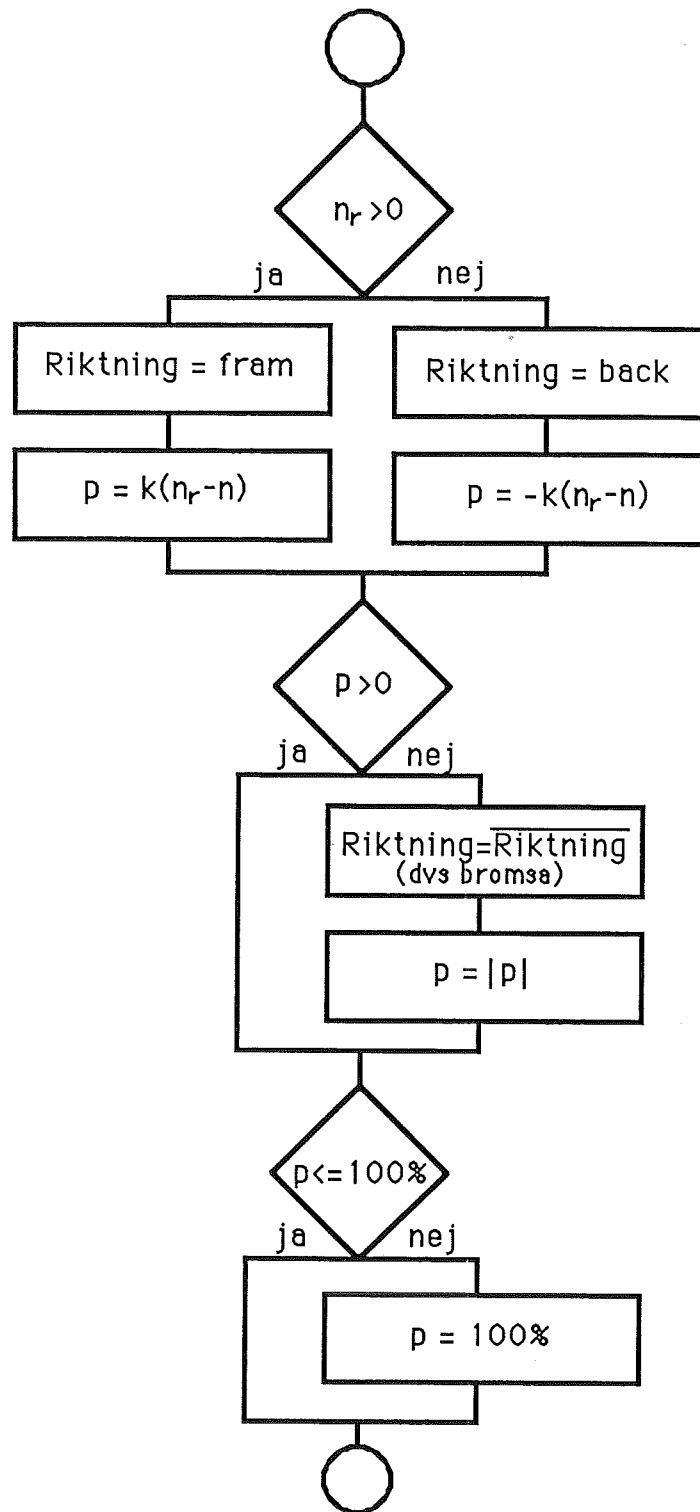


Fig. A3. Subrutin "Beräkna ny pulsvidd" från fig. A2.

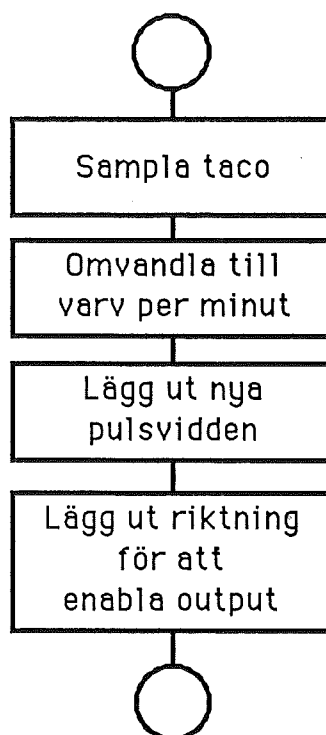


Fig. A4. Interrupt rutin.

I Figur A4 omvandlar jag taco-värdet, d v s det som ADC:n ger, till varv per minut för att börvärdet skall kunna läggas in som varv per minut och för att förstärkningen k skall få sin rätta betydelse i P-regulatorn. Här finns dock mycket exekveringstid att spara där inte läsbarheten har högsta prioritet utan där snabbheten hos programmet har största betydelsen. Här lägger jag också ut riktning, och detta är till för att tala om hur motorn skall polas, men framförallt för att pulserna skall komma ut till motorn. I appendix om MCU-kort beskriver jag varför det är så.

Vidare kan vi se i Figur A2 att det finns ett block som ändrar börvärdet för hastigheten. Denna ändrar börvärde enligt Figur A5.

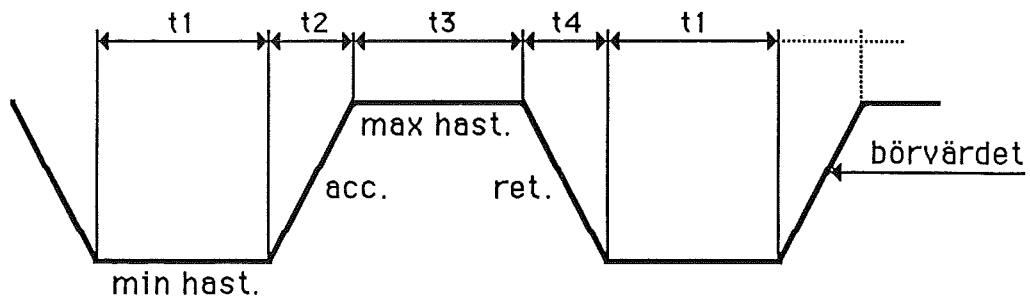


Fig. A5. Börvärdet, så som det läggs ut av subrutin "Ändra hastighets börvärde" i fig. A2.

Man kan här ställa in olika min- och max-hastigheter, accelerationer och retardationer. Dessutom kan t_1 och t_3 ställas in. Acc och ret bestämmer hur stora t_2 och t_4 blir.

Nu går jag över till det andra programmet, som förutom hastighetsåterkoppling dessutom har positionsåterkoppling. Huvuddragen ses nedan i Figur A6.

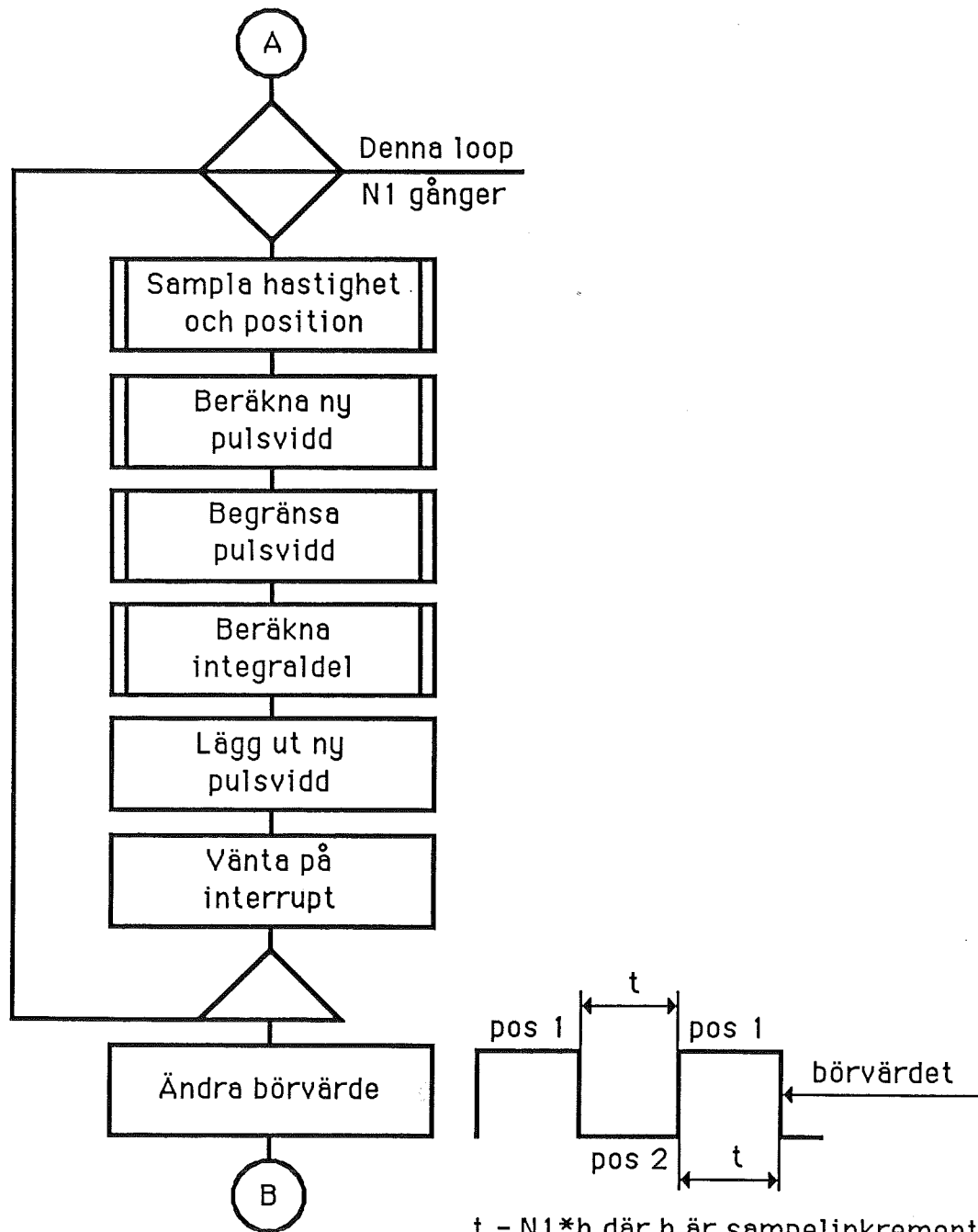


Fig. A6. Program med p-regulator i hastighetsloopen och pi-regulator i lägesloopen.

Detta, liksom det andra programmet, börjar med att sampla av givarna, som vi ser i Figur A7.

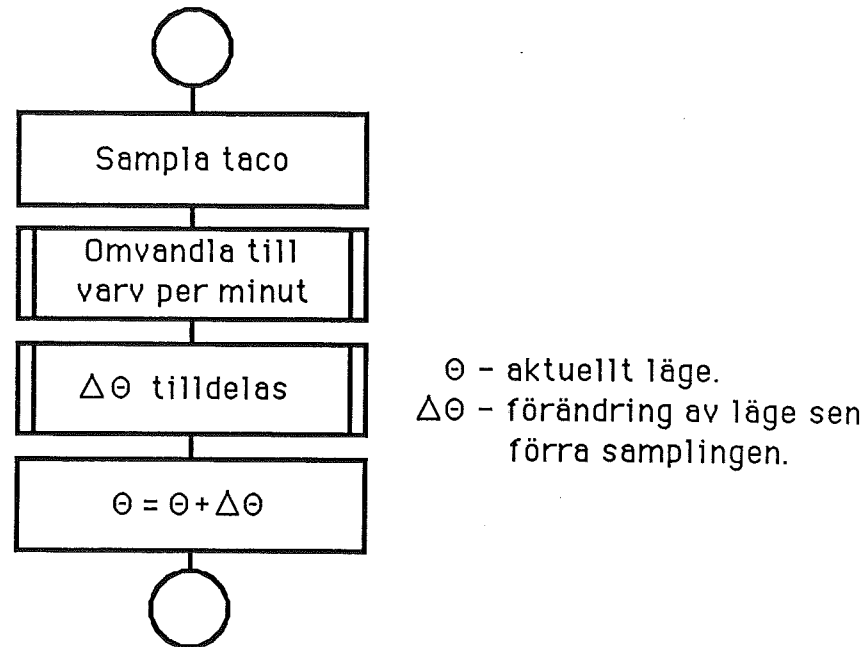


Fig. A7. Subrutin "Sampla hastighet och position" i fig. A6.

I Figur A8 finns en utförligare beskrivning av hur positionsändringen erhålles. Eftersom det endast är en 8-bitarsräknare som håller reda på positionen måste jag lägga till 256, om räknaren har gått från 256 till 0, och dra ifrån 256 om den har gått från 0 till 256.

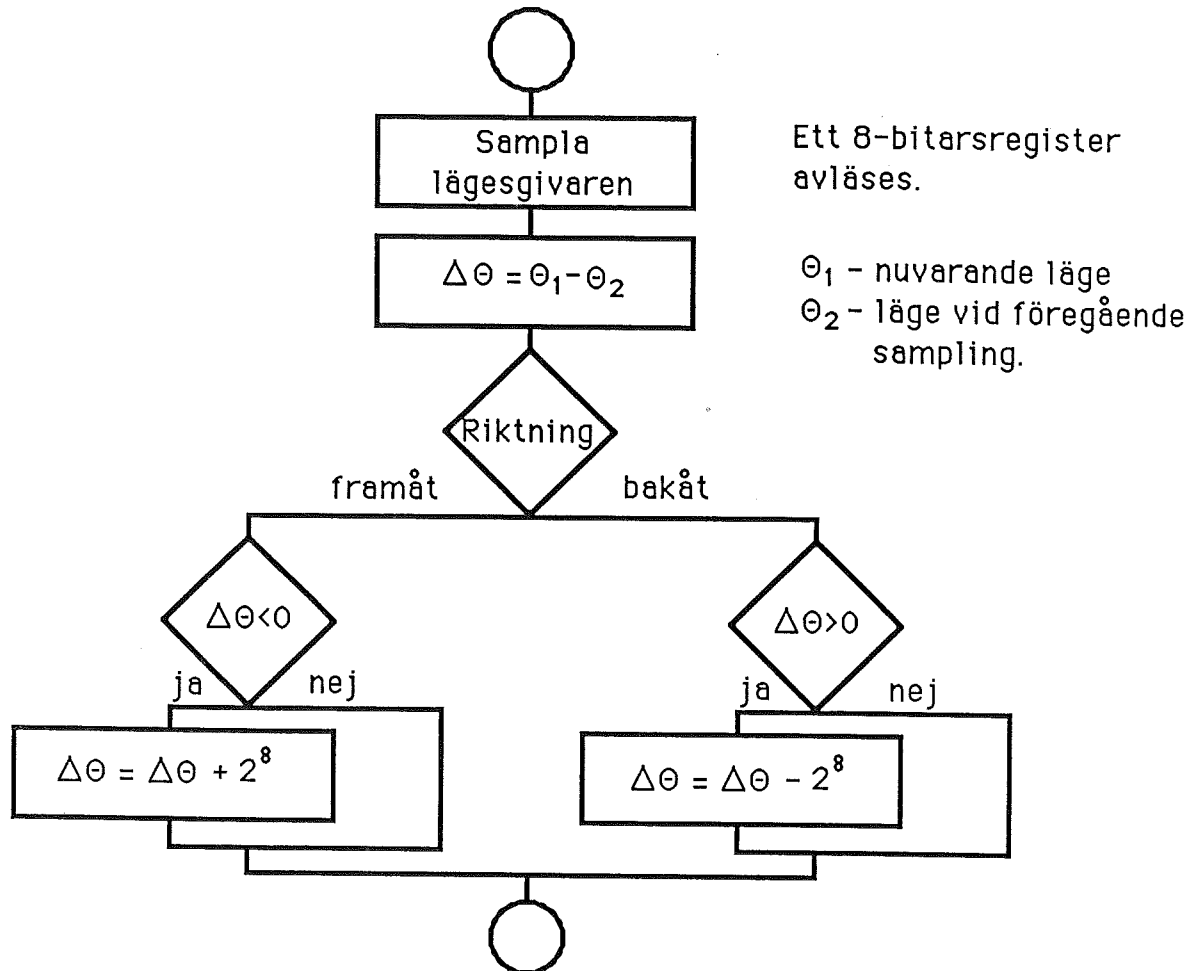


Fig. A8. Subrutin "Δθ tilldelas" i fig. A7.

Därefter är det dags att räkna ut pulsvidden. I Figur A9 kan vi se formeln för regulatorn. Precis som innan är det en P-regulator på hastighetsåterkopplingen, men dessutom finns här en PI-regulator på positionsåterkopplingen. Förutom detta finns även en framkoppling av positionsreferensen. Om man inte vill ha en integrationsterm går den att disabla, precis som framkopplingen går att ta bort genom att sätta $k_3 = 0$.

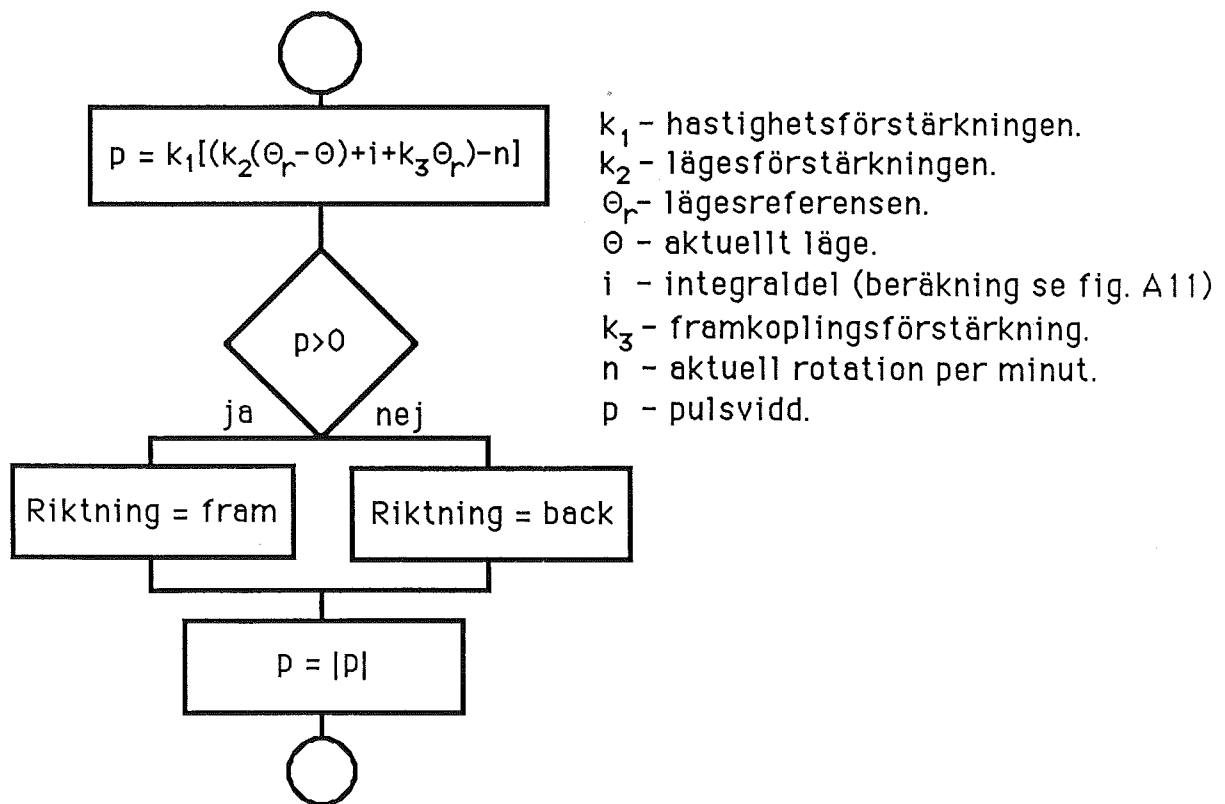
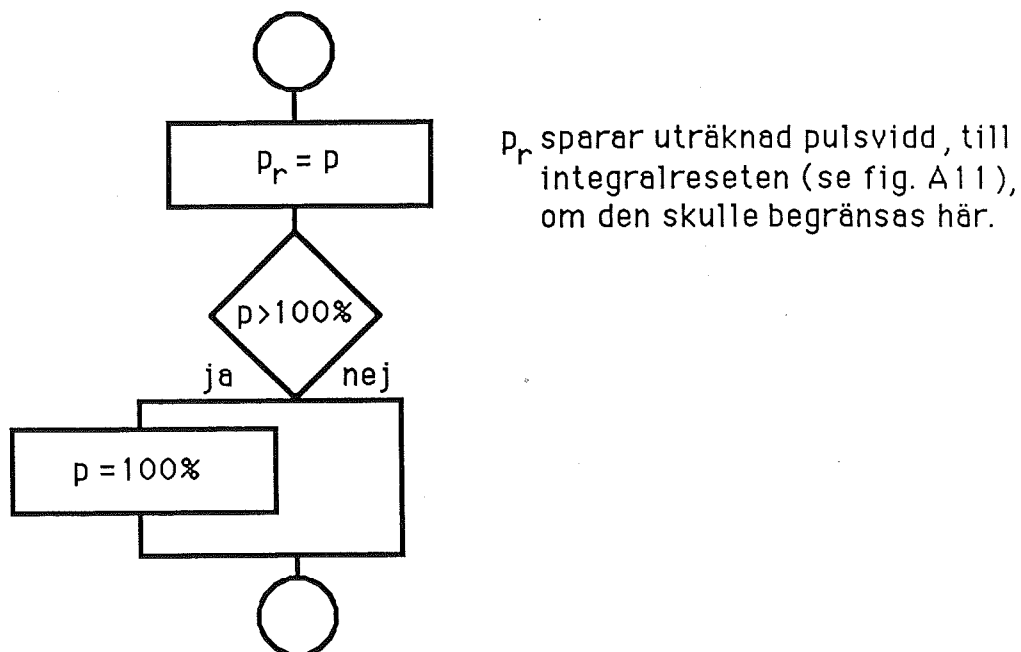


Fig. A9. Subrutin "Beräkna ny pulsvidd" i fig. A6.

Eftersom man inte kan lägga på mer än 100% pulsvidd begränsas pulsvidden. Se Figur A10.

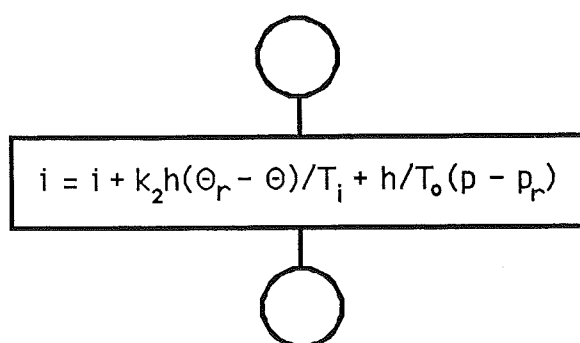


p_r sparar uträknad pulsvidd, till integralreseten (se fig. A11), om den skulle begränsas här.

Fig. A10. Subrutin "Begränsa pulsvidd" i fig. A6

Om nu pulsvidden skulle ha varit större än 100% spar jag det gamla värdet för att användas till integralreseten.

Integraluträkningen ses i Figur A11. Den sista termen är just integralreseten.



i - integraldel.
 k_2 - lägesförstärkningen.
 n - sampelinkrementet.
 Θ_r - lägesreferensen.
 Θ - aktuellt läge.
 T_i - tidskonstant för integreringen.
 T_o - konstant för integralreset, oftast är $T_o = h$.
 p - den pulsvidd som läggs ut.
 p_r - den pulsvidd som räknades i fig. A9.

Fig. A11. Subrutin "Beräkna integraldel" i fig. A6.

Det som nu återstår (se Figur A6) är att lägga ut ny pulsvidd och vänta på interrupt (för att hålla konstant sampel-frekvens). Till sist i Figur A6 finns ett block som ändrar börvärdet för att jag skulle kunna mäta upp stegsvaret. Detta sker utanför iterationen, som görs N1 gånger. N1 väljs så att stegsvaret hinner klinga ut innan ett nytt steg läggs på.

I de båda ovan beskrivna program har jag räknat med heltal för att hålla exekveringstiden nere. Detta innebär bl a att inga decimaler existerar. För att hålla noggrannheten uppe har jag därför låtit förstärkningar och andra konstanter vara uttryckta i tusendelar och till slut dividerat med tusen. Eftersom det är en 16-bitars processor begränsades talen till 65536. Detta räckte inte till i det andra programmet, utan jag fick använda double precision. Det innebär en negativ effekt på exekveringstiden som gav problem. Mer om detta finns i kapitel 4.

TACO-GENERATORN OCH STÖRNINGAR

Taco-generatorn ger 6 V/1000 rpm. Data på den finns i appendix F.

Anledningen till att jag skrivit detta kapitel är att jag vill belysa hur pulsningen kan påverka utrustningen. Bilderna B1 och B2 nedan är foton på taco-signalen.

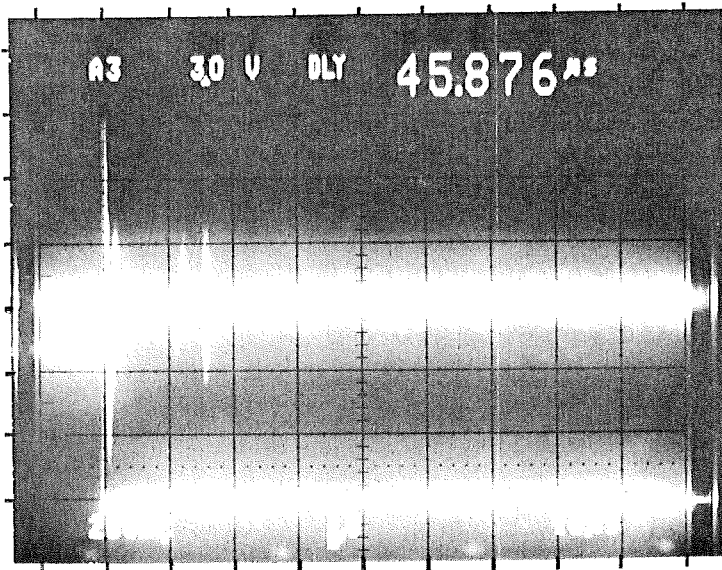


Bild B1.
Taco-signal

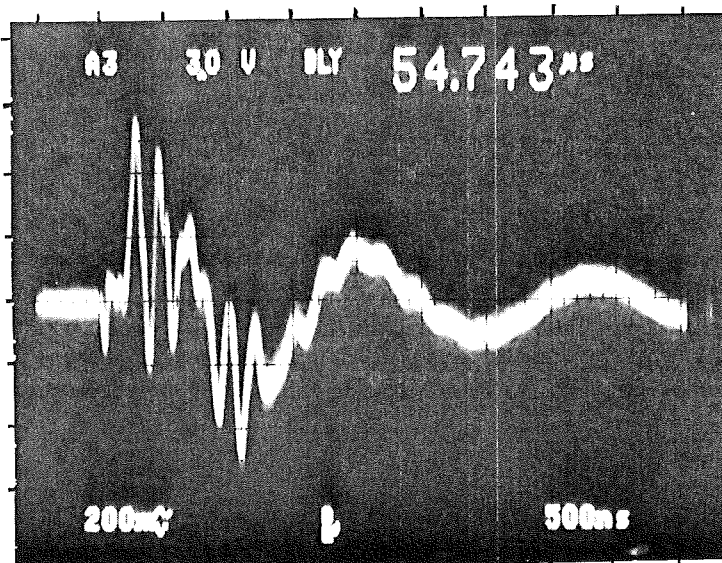


Bild B2.
Taco-signal in-
zoomad precis vid
störningen

Mätningen är gjord på Tektronix oscilloskop 2465 CTS. Motorn hade ett varvtal på ca 1900 rpm. Tacon gav alltså ca 11.4V. Vid mätningen var oscilloskopet AC-kopplat för att endast ringningarna skulle kunna ses. Ringningarna startar precis när PWM-signalen (5 kHz) har sin negativa flank och det är alltså denna som induceras över till tacon. På bild B1 är den undre strålen PWM-signalen. Likadant sker vid positiv flank. På bild B2 har svepet ändrats från B1:s 10 μ s/div till 0.5 μ s/div. Spänningsskalan är 0.2 V/div.

Dessa ringningar gav mig inte några problem med regleringen. Man bör dock vara uppmärksam på att sådana här störningar finns, därför att de eventuellt kan ställa till problem någon annanstans i utrustningen. Liknande fenomen uppträdde när jag i början av mina mätningar hade ett switchat likströmsaggregat.

MCU-KORT

Figur C1 är en schematisk ritning av hur MCU-kortet är uppbyggt.

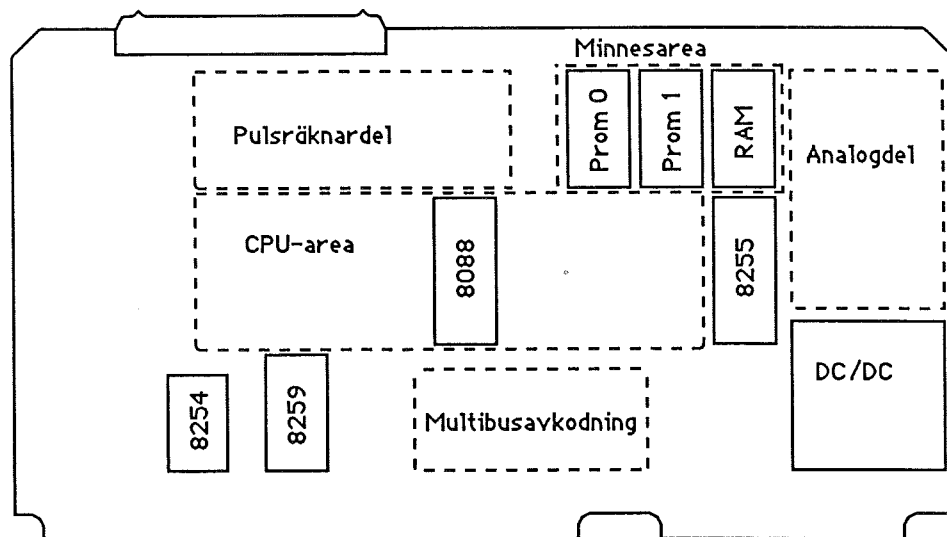


Fig. C1. MCU-kortet.

Som mikroprocessor används Intels 8088. I CPU-arean finns dessutom en klockkrets och avkodningar för bl a chip select. I minnesarean finns tre minnen. När jag körde var PROM 0 utbytt mot ett RAM för att möjliggöra programmering från ICE:n. Multibussavkodningen är till för att möjliggöra styrning av kortet med en huvuddator. DC/DC-omvandlaren gav referensspänningar till DAC:en, och den i sin tur fick sina inställningar och värden av CPU:n via parallellporten (8255:an). 8259:an är en programmerbar avbrottshanterare som vid min tillämpning endast tog emot realtidsavbrotten. I analogdelen sitter bl a ADC:n och DAC:en. Intervalltimern 8254:an var den som gav mig PWM-signalen och realtidsklockan.

I Figur C2 har jag ritat ett blockschema över de kretsar och byggblock som är de viktigaste vid mina mätningar.

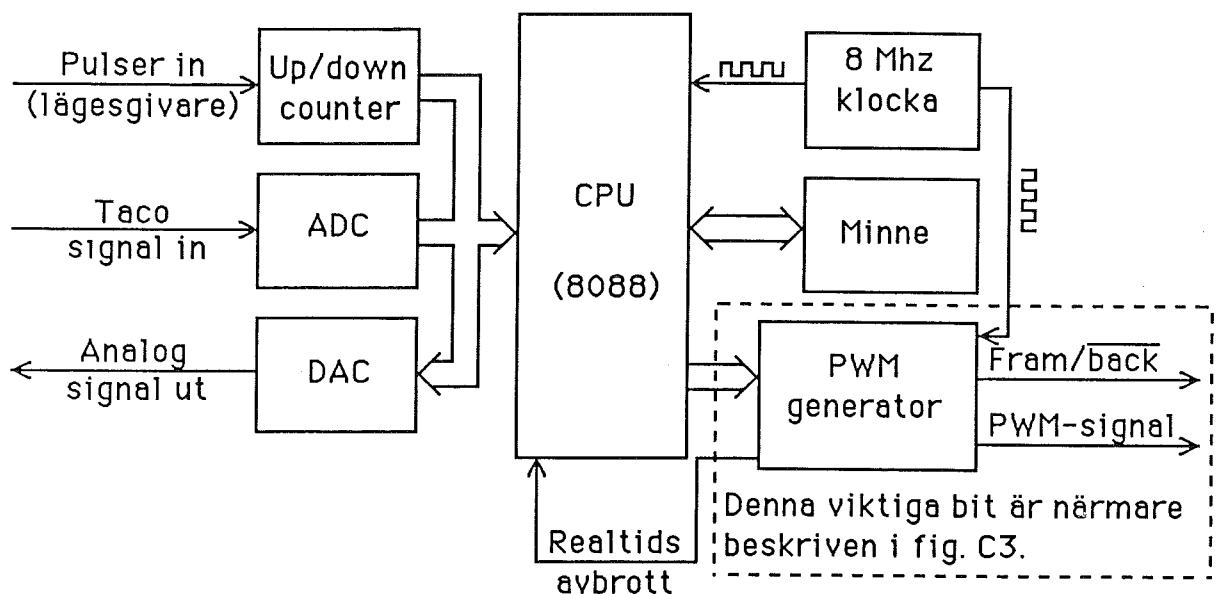


Fig. C2. Blockschema över de viktigaste delarna ur PWM synpunkt.

8-MHz-klockan klockar både processorn och intervalltimern. UP-/DOWN-countern är en 8-bitars räknare som håller reda på läget. Denna kan processorn läsa av. Pulsgivaren genererar två fyrkantssignaler som sinsemellan är fasförskjutna 90 grader. Detta gör att räknaren kan avgöra om motorn roterar framåt eller bakåt.

Tacon går in via ett resistansnät och en MUX till ADC:n. Detta gör att man direkt från mjukvara kan ställa ADC:ns arbetsområde. Den är på 12 bitar. DAC:en används i vanliga fall, när ingen PWM används, till att ge börvärde till ett DC-servo. Jag använde den vid mätning på regulator med lägesåterkoppling, till att generera en spänning som motsvarar läget. På detta sätt kunde jag registrera läget på en skrivare.

Eftersom intervalltimern med tillhörande kretsar är huvudbyggblocket för mig har jag en mer detaljerad beskrivning nedan i Figur C3.

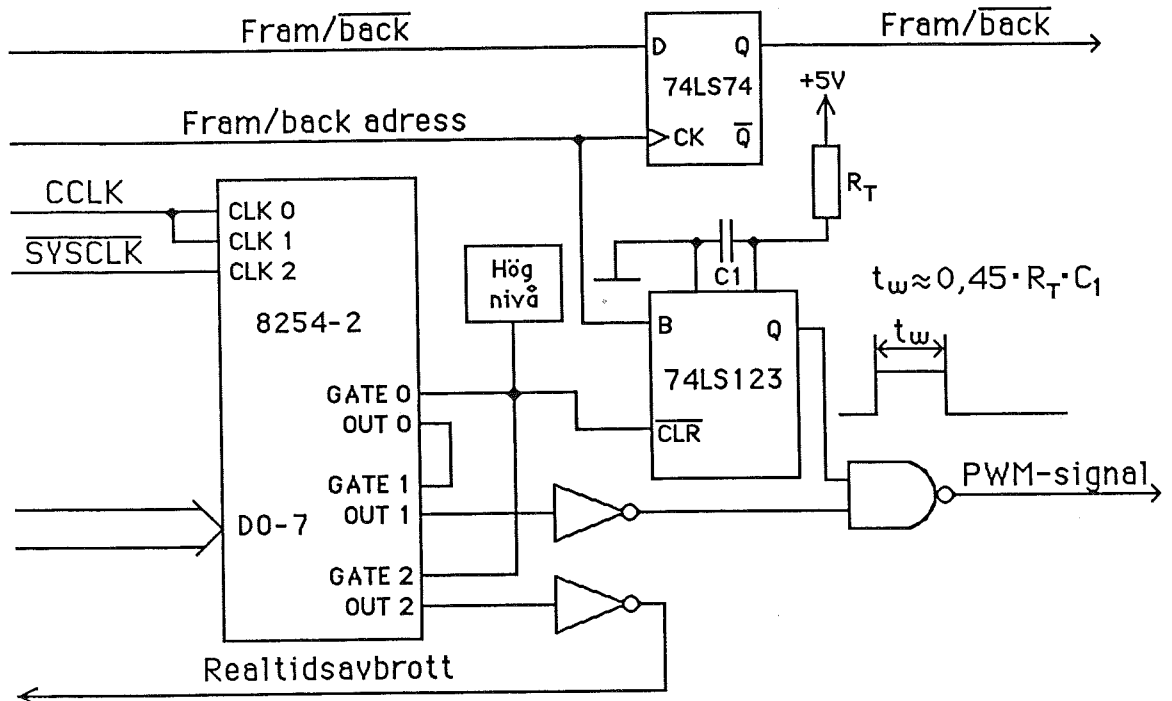


Fig. C3. PWM genereraren med säkerhetsbrytaren och fram/back signalen.

I Figur C3 är det 8254:an som är basen. Clk1 är till för att bestämma pulslängden. Denna är ställd i mode 1 (one-shot). Det innebär att varje gång den triggas stannar den låg, så länge jag har sagt till den via CPU:n. Clk1 triggas av Clk0 som är i mode 3. På detta sätt genereras pulsfrekvensen. I denna mode genererar den en fyrkantsvåg, som är en neddelning av systemklockan (8 MHz). Jag delar ner den antingen 4000 eller 8000 gånger vid mina regulatortester för att få 2 eller 1 kHz. Clk2, som är realtidsklockan, lät jag alltid gå exakt som Clk1, därför att jag reglerade varje puls.

Clk0 och Clk1 går att klocka separat. Detta gör att upplösningen hos regleringen kan göras större, vilket beror på

neddelningen av klockan in i timern. Kör jag med 2 kHz pulsfrekvens och 8 MHz klocka in så måste jag som jag sade ovan dela ner 4000 gånger. Det innebär att Clk1 kan välja en pulsbredd med en upplösning av en klockperiod för klockan in, d v s $1/4000$. Digitalt motsvarar detta ca 12 bitar. Fördubblar jag pulsfrekvensen så är upplösningen bara $1/2000$, ca 11 bitar. Ökar jag nu klockan in till Clk0 och 1 kan upplösningen ökas för en bestämd pulsfrekvens. Detta är en fördel om man vill ha hög pulsfrekvens men samtidigt kunna reglera lika noggrant som ADC:n samplar tacogeneratorn.

Vill man ha en hög pulsfrekvens men inte reglera med samma frekvens går det bra, därför att Clk1 inte behöver laddas med nytt värde var gång den triggas. Den håller kvar det gamla värdet om inget nytt fås. På så vis kan man ha en pulsfrekvens på t ex 20 kHz men endast reglera med, låt oss säga, 1 kHz. PWM-frekvensen måste vara lika eller högre än realtidsfrekvensen, eftersom man bara kan reglera en gång per period i PWM-signalen.

Krets 74LS74 (i Figur C3) är en vanlig d-vippa, som genererar fram- eller backsignalen. När man adresserar vippan avkodas adressen till klockingången på vippan. På så sätt klockas den. Denna klocksignal går också till den monostabila vippan. Detta är en "säkerhetsbrytare". Denna vippa genererar en puls som är ca 2.8 ms lång. Vippans utgång måste vara hög för att pulserna till motorn skall komma igenom NAND-grinden. Därför måste den monostabila vippan triggas med ca 360 Hz för att motorn skall kunna gå. Lämpligtvis triggar man den var gång man reglerar pulsbredden. Skulle nu programmet falla ifrån så fortsätter intervalltimern att generera pulser, men den monostabila vippan ser till att de inte kommer fram till motorn. På detta sätt snurrar motorn inte okontrollerat utan stannar bara vid fel.

Det bör nämnas att på MCU-korten i originalskick var R_T vid den monostabila vippan endast 33 kohm. Detta gav dock bara en puls på 0.9 ms, varför jag ändrade till 100 kohm. Annars hade inte den monostabila vippan triggats tillräckligt ofta för att möjliggöra en så låg pulsfrekvens som 1kHz. Den lägsta pulsfrekvens jag använde.

KRAFTKORT

För att driva motorn med en PWM-signal har en kretslösning från Siemens använts. En ritning på denna finns i Figur D1. Krafttransistorerna är Siemens SIPMOS. De i Figur D1 angivna BUZ 15 har dock ersatts med BUZ 25. De mätningar jag har gjort på denna del återfinns i kapitel 5. För att lättare kunna använda kraftdelen har extra inverterare placerats före de befintliga på ingångarna B och D. Dessa är inritade i Figur D1.

Vid mina testkörningar använde jag en matning på 30V. Konstruktionen gör dock att endast ca 27 av dessa ligger över motorn, resterande ligger över antingen T1 eller T2 (beroende på rotationsriktning för motorn). Spänningen över T3 resp T4 är försumbar vid drivning.

Plus och minus vid motorn i Figur D1 är till för att man skall veta hur motorn skall installeras (för önskad polning) på kopplingsplint (kopplingsplint, Figur D4).

För att kunna driva detta med bara PWM- och FRAM-/BACK-signalerna från MCU-kort har jag gjort vissa kompletteringar. Dessa återfinns i figurerna D2 och D3. Figur D3 innehåller kretslayouten och återfinns på nästa sida, medan Figur D2 nedan är en funktionsbeskrivning.

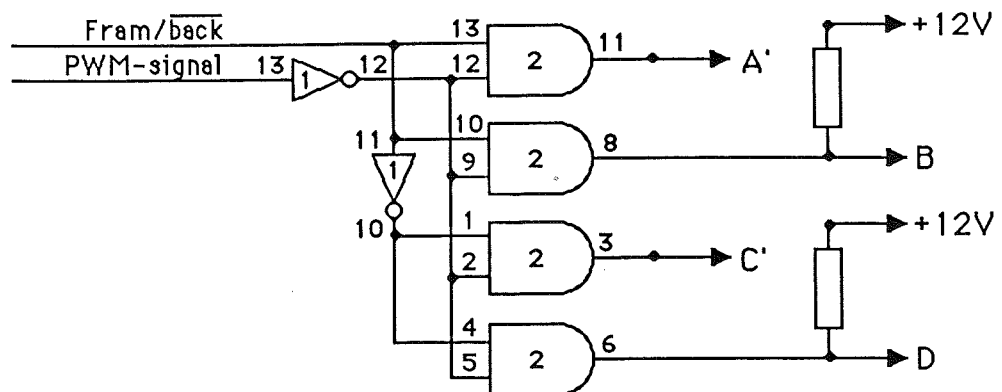


Fig. D2. Funktionsbeskriven av kompletteringskort till kraftkort.

Kretsarna i Figur D2 är TTL-kretsar för att passa MCU-kortets nivåer. Siemens kort i Figur D1 vill dock ha 12V vid "etta" på ingångarna A till D. För att lösa detta har jag använt 74LS09 med sina open-collector-utgångar. Till B och D har jag anslutit utgångarna direkt med hjälp av pull-up-motstånden R1 och R2 till 12V. För A och C har jag utnyttjat en del befintligt. A och C har anslutits direkt till 12V och open-collector-utgångarna har anslutits till A' och C'. På detta sätt används 6.8 k och 3.3 k i Figur D1 som pull-up-motstånd.

För att kunna köra motorn utan styrning från processorkort har jag tillverkat en manuell FRAM-/BACK-signal. Man låter FRAM-/BACK-ingången på plinten (se Figur D4) vara öppen och byglar b1 till b2 och b3 till b4 (se Figur D3).

Byglas sedan b5 till b6 fås BACK-signal, och när byggligen tas bort fås FRAM-signal. Sedan ansluts en pulsgenerator till PWM på plinten med en amplitud på 5V.

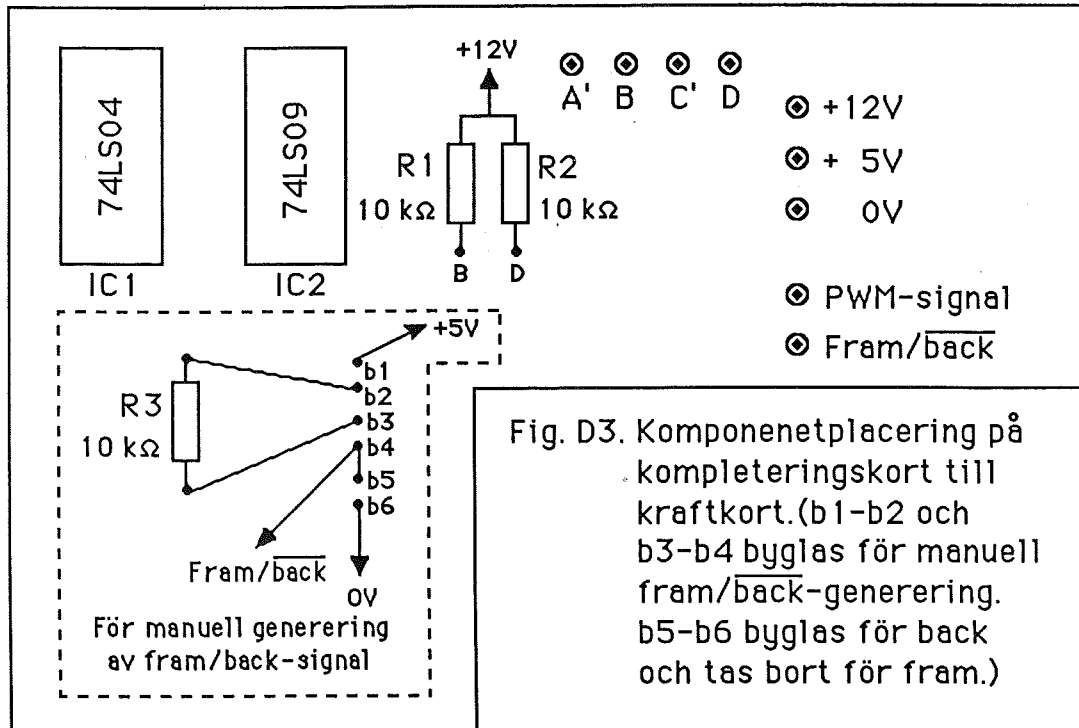


Fig. D3. Komponentplacering på kompletteringskort till kraftkort. (b1-b2 och b3-b4 bygglas för manuell fram/back-generering. b5-b6 bygglas för back och tas bort för fram.)

⊙ 10	—
⊙ 9	—
⊙ 8	Fram/back (fram 5V, back 0V)
⊙ 7	PWM-signal (hög 5V, låg 0V)
⊙ 6	+ 5 V
⊙ 5	Pluspol till motor
⊙ 4	Minuspol till motor
⊙ 3	+ 12 V
⊙ 2	0 V
⊙ 1	+ 20-30 V

Fig. D4. Plintinkoppling på kraftkort.

MÄTINSTRUMENT OCH MÄTMETODER

Mätningarna i kapitel 3 utfördes enligt följande:

Motorns hastighet mättes på taco-generatorn med Philips PM3234 oscilloskop. Den andra kanalen anslöt jag till puls-generatorn för att bestämma duty-cyclen. Puls-generatorn var av märket Hewlett-Packard, typ hp 8011A Pulse Generator.

DC-nivån på motorn mättes med Fluke 8000A True RMS Multi-meter, inställd på DC-volt-området. Denna kopplades direkt över motorn på + och -, se Figur D1. För att mäta strömmen genom motorn använde jag CTE 7908 Multimeter, inställd på DC-ström. Amperemetern kopplades i serie med 30-volts matningsspänningen, se Figur D1.

Alla mätningar som utförts enligt ovanstående metod finns i tabeller och diagram i kapitel 3.

I kapitel 4 mätte jag endast hastighet med taco och läge via DAC:n från MCU-kortet. Dessa registrerade jag med en skrivare av märket Hewlett Packard, modell hp 7402A Recorder.

I kapitel 5 gjorde jag flera mätningar på kraftkortet. Hur dessa utfördes finns angivet i det kapitlet. Mätningarna är redovisade med foto och är utförda på Tektronix oscilloskop 2465 CTS.

DATA PÅ DC-MOTOR, TACO- OCH PULSGIVARE

DC-motor: Fabrikat CEM
Code: FD 57 449
Type: F12M4R No 364171
Max.värde: U 37V, I 8 A
 P 210 W, n 4800 rpm

Taco-generator: Fabrikat CEM
Nr FD FC12T R0024
6 V/1000 rpm

Pulsgivare: Fabrikat DATE TeCHnology, inc.
Nr RS23TM Encoder
Model -200-40/5-L

DC-motorn ovan driver två plan att rotera via en växel och kuggremmar. Växeln är av märket Setan Nr 2512, med utväxling 1/50. Kuggremmarna gör slutligen så att utväxling mellan plan och motor är 1/180.

Taco-generatorn sitter direkt monterad på DC-motorn, och pulsgivaren är kopplad via kuggrem, med utväxling 1/1, till motorn.