

IMPLEMENTERING AV KVOTREGULATOR
PÅ MIKRODATOR

ANDERS NILSSON

RE-179 Juni 1976
Inst. för Reglerteknik
Lunds Tekniska Högskola

TILLHÖR REFERENSBIBLIOTEKET

UTLANAS EJ

IMPLEMENTERING AV KVOTREGULATOR PÅ MIKRODATOR

EXAMENSARBETE

UTFÖRT AV ANDERS NILSSON

UNDER VÅRTERMINEN 1976

VID INSTITUTIONEN FÖR REGLERTEKNIK

PÅ TEKNISKA HÖGSKOLAN I LUND

HANDLEDARE:

BJÖRN WITTENMARK

LEIF ANDERSSON

ROLF BRAUN

I N N E H Å L L S F Ö R T E C K N I N G

	Sammanfattning	2
	Abstract	3
1	Inledning	4
2	Beskrivning av kvotregulatorn	5
3	Hårdvara	8
	4040	9
	S/F-omvandlare	10
	Pulsräknare	10
	Pulslängdsgivare	11
	Termistorbrygga med probe	12
	Effektregulator	13
	Monovippa för avbrottssignal	13
	Programmeringshjälpmedel	13
4	Mjukvara	15
	KVOTR	16
	MULTH	18
	DIVIH	18
	ROTH	19
	DUB	19
	Modifikationer av KVOTR	20
5	Uppkopplingsinstruktion	21
6	Experiment	23
	System på analogmaskin	23
	Styrsignalen	23
	1:a och 2:a ordningens system	23
	Steg- och rampstörningar	25
	Variation av T_s	27
	Begränsning av insignalens ändringsmöjligheter	30
	Framkoppling	32
	Långsamt system	33
	Vattenvärmning	34
7	Referenser	35
	Appendix 1	
	Appendix 2	

S A M M A N F A T T N I N G

Kvotregulatorn, som definieras av sambandet $u(t) = y_r \cdot u(t-1)/y(t)$ (beteckningar: u insignal, y verklig utsignal och y_r önskad utsignal på det system som skall styras), har undersökts huvudsakligen ur praktisk synvinkel i detta examensarbete.

En mikrodator av typ Intel 4040 har använts för att utföra beräkningarna i algoritmen och för att administrera signalflödet.

Enheter har byggts upp för att kunna mäta på processer (spänningar eller temperaturer) och föra in dessa mätvärden i datorn, samt omvänt för att kunna ställa ut en styrsignal.

Ett basprogram som realiserar algoritmen och som upptar ca 325 minnesceller har skrivits i ett assemblerspråk för Intel 4040. Modifikationsmöjligheter har angivits i rapporten.

Experiment visar att kvotregulatorn fungerar på 1:a och 2:a ordningens system och eliminerar störningar tillfredsställande. Regulatorn är känslig för val av samplingstid, speciellt har den nackdelen att kräva mycket långa samplingsintervall vid långsamma system. De viktigaste fördelarna med kvotregulatorn är att inga parametrar i processen behöver vara kända och att stationärt fel i utsignalen saknas.

A B S T R A C T

The regulator defined by the expression $u(t) = y_r \cdot u(t-1) / y(t)$ (notations: u input signal, y measured output signal and y_r reference output signal of the process which is to be controlled) has been examined mainly from a practical point of view.

A microcomputer (Intel 4040) has been used to execute the calculations of the algorithm and to administrate the flow of signals.

Units have been built to be able to measure (voltages and temperatures) on processes and then put these measurements into the computer, and reversely to be able to put out a control signal.

A basic program extending over about 325 memory cells has been written in an assembly language for Intel 4040. Some modifications of the program have been mentioned in the report.

Experiments indicate that the regulator works on systems of first and second order and in a satisfactory way eliminates disturbances. The sampling time is a very critical parameter. Especially the regulator has the disadvantage to require very long sampling intervals when the process is slow. The most important advantages of the algorithm are the lack of a stationary error in the output signal and that you do not have to know any parameters of the process.

1.

I N L E D N I N G

Kvotregulatorn är en enkel parameterfri regulator, som kan reglera medelfelet till noll. Dess teoretiska uppbyggnad och funktion beskrives i kap. 2. Här anges förutom den enklaste varianten en del finesser, som kan förbättra regulatorns egenskaper.

Kvotregulatorn realiserats lättast som en digital regulator. En dator krävs alltså. Regulatorns enkelhet gör att endast små krav ställs på datorn varför en mikrodator är fullt tillräcklig. Den största svårigheten med att använda en sådan ligger i kommunikationen med omvärlden. Hur detta problem har lösts framgår av kap. 3. Här beskrives även mikrodatorn kortfattat och något om hur man programmerar den.

För att styra datorn till att uppfylla kvotregulatorns funktion krävs ett program avpassat för den använda mikrodatorn. Hur detta har konstruerats kan studeras i kap. 4.

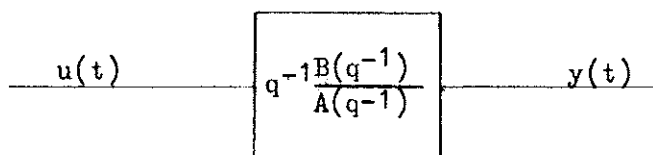
För den som önskar att använda uppställningen (mikrodatorn + yttre enheter) finns en uppkopplingsinstruktion angiven i kap. 5.

Till sist i kap. 6 prövas hur kvotregulatorn fungerar i praktiken. Olika system undersöks bl. a. uppvärmning av vatten på en kokplatta. Inverkan av störningar, samplingstidens betydelse och några modifikationer av basalgoritmen studeras.

2. BESKRIVNING AV KVOTREGULATORN

ALLMÄNT:

Antag att vi har ett samplat system med en insignal och en utsignal enligt figuren nedan.



Om vi bortser från dynamiken i systemet får vi kvar den statistiska delen $\frac{B(1)}{A(1)} \approx k$ dvs som approximation kan gälla $y(t) = k \cdot u(t-1)$. Vi gör följande uppskattning $\hat{k} = \frac{y(t)}{u(t-1)}$ vilket kan ses som ett specialfall av minsta kvadratuppskattningen

$$\hat{k}_{mk} = \frac{\frac{1}{N} \sum_i^N y(t)u(t-1)}{\frac{1}{N} \sum_i^N u(t-1)^2}$$

Regulatorn får då utseendet

$$u(t) = \frac{y_r}{\hat{k}} = \frac{y_r}{y(t)} u(t-1)$$

där y_r är referensvärdet för utsignalen.

Intuitivt verkar algoritmen vara riktig. Om en viss insignal ej har gett $y = y_r$ så ändras insignalen med faktorn y_r/y vid nästa sampling.

Ett nödvändigt krav för att kvotregulatorn skall fungera är att insignalen $u(t) > 0$ för alla t och att kvoten $y_r/y > 0$ för alla t .

Kvotregulatorn bör på grund av ovanstående approximation fungera bäst på system som ej innehåller alltför mycket dynamik.

Några fördelar hos regulatorn:

- o Inga parametrar hos systemet behöver vara kända
- o Ger inget stationärt fel, ty vid stationaritet gäller $u(t) = u(t-1) \Rightarrow y(t) = y_r$
- o Enkel ur matematisk synvinkel

STABILITET:

Kvotregulatorn är lokalt stabil för linjära dynamiska processer där utsignalen antar mer än hälften av sitt slutvärde i första mätpunkten efter ett steg i insignalen enligt referens 1. Tiden som förlöper till första mätpunkten bestäms av samplingstiden T_s . Vid tröga system krävs alltså en lång samplingstid. Överhuvudtaget ändras regulatorns inverkan på systemet kraftigt då T_s ändras. Vid användning av kvotregulatorn bör man alltså lägga ner en del arbete på att finna ett bra värde på T_s .

FINESSER:

1/ $u_{\min} > 1$

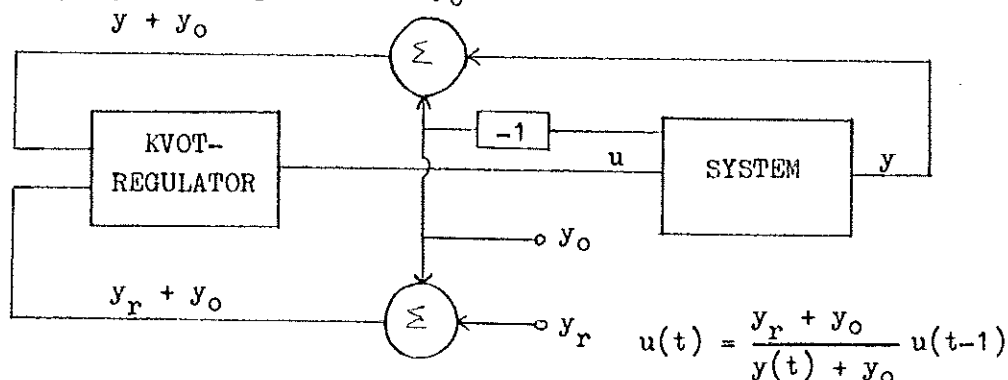
Vid en digital representation med n binära siffror av ingående variabler så representeras u i datorn med ett tal i intervallet $[1, 2^n - 1]$. u_{\min} motsvaras alltså av en etta. För att komma från detta värde på u krävs att $y_r/y(t) \geq 2$ om divisionen sker med avkortning. Om $u=1$ måste vi följdaktligen vänta tills $y(t) < y_r/2$ innan regulatorn kan börja öka på insignalen och därigenom även öka på y . Denna olägenhet kan reduceras genom att begränsa u så att exempelvis $u_{\min} = 10$. Observera dock att nu blir det svårare att nå låga y_r .

2/ begränsning av $|\Delta u|$

Då utsignalen svänger kraftigt svänger likaså insignalen kraftigt. Genom att begränsa $|\Delta u| = |u(t) - u(t-1)|$ får vi ett lugnare men samtidigt även en långsammare utsignal.

3/ framkoppling

Om systemet kan ge en negativ utsignal exempelvis vid en störning kan kvotregulatorn användas om vi har framkopplat systemet lämpligt. Se figur nedan. $y_0 > 0$



4/ medelvärdesbildning

Utgående från minstakvadratuppskattningen \hat{k}_{mk} kan vi uppskatta k genom att ta glidande medelvärden över de sista n värdena

$$\hat{k}_n = \frac{\frac{1}{n} \sum_{N-n}^N y(t)u(t-1)}{\frac{1}{n} \sum_{N-n}^N u(t-1)^2} \quad ; \quad u(t) = \frac{y_r}{\hat{k}_n} .$$

5/ skattning av systemets dynamik

Antag att systemet kan skrivas

$$y(t) = ay(t-1) + (1-a)ku(t-1)$$

där a kan gissas ganska väl. Sätt

$$\tilde{u}(t) = \hat{a}u(t-1) + (1-\hat{a})u(t-1) .$$

$\hat{a} = a$ ger $y(t) = k\tilde{u}(t)$.

Regulatorn kan nu skrivas som

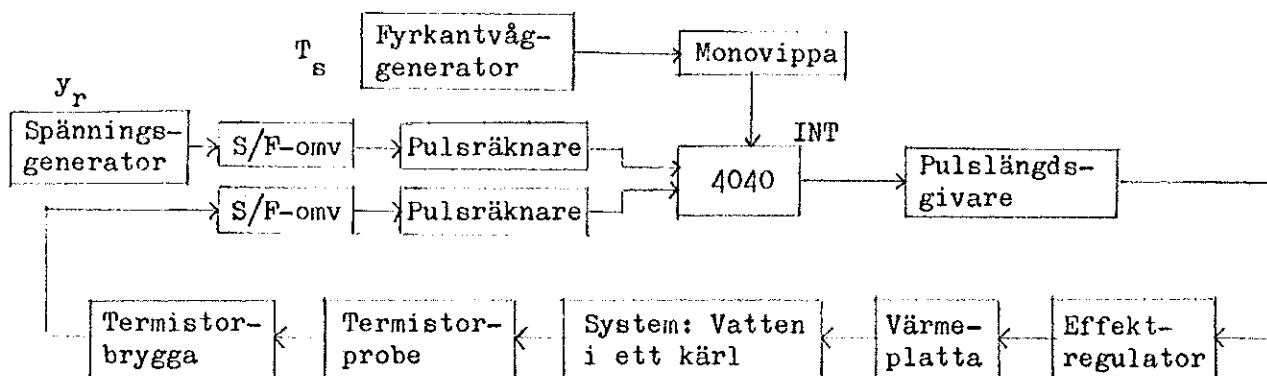
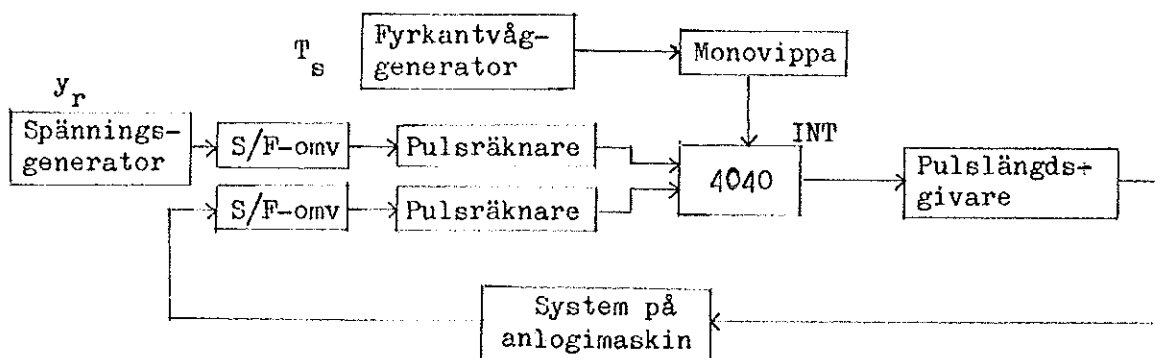
$$u(t) = \frac{y_r}{y(t)} \tilde{u}(t) .$$

$a = 0$ ger den vanliga kvotregulatorn.

3.

H Å R D V A R A

För att realisera en kvotregulator med utgångspunkt från en mikro dator krävs en mängd yttre enheter till datorn. Vad som krävs är beroende av det system man vill styra. Här nedan finns i form av blockscheman möjliga lösningar anpassade till två olika system. (S/F-omv betyder spänning till frekvens omvandlare)



Då det övre schemat ingår som en del av det undre ges här endast en förklaring av det senare. Utgående från systemet mätes temperaturen med en termistorprobe. Termistorns resistans omvandlas till en spänning proportionell mot temperaturen av termistorbryggan. S/F-omv överför spänningen till en frekvens som räknas av pulsräknaren och resultatet matas in i mikro datorn. Det beräknade u-värdet sändes till pulslängdsgivaren vars utsignal består av pulser som har en längd prop. mot u. Denna pulslängd avgör hur stor del av effekten från ett vägguttag som skall matas in på värmeplattan. y_r bestäms med en spänningsgenerator.

T_g bestäms av periodtiden hos fyrkantvågen vilken omvandlas av monovippan till en puls lämpad att sända som interruptsignal på 4040.

Nedan beskrives de olika enheterna var för sig. 4040:n, S/F-omv och programmeringshjälpmedeln fanns redan framtagna på institutionen varför de endast beskrives översiktligt. Konstruktionen av pulsräknare, pulslängdsgivare, termistorbrygga med probe, effektreulator och monovippa har däremot ingått som en del av examensarbetet.

4 0 4 0

Kopplingsschema för mikrodatorn finns på sidan 1 i appendix 1.

Några ur vår synpunkt viktiga data hos CPU:n:

- o 60 instruktioner
- o 10,8 μ s instruktionscykel
- o 4 bitars ordlängd
- o Avbrottsmöjligheter
- o 24 indexregister
- o 7 subrutinnivåer

En utförlig beskrivning av CPU:n återfinnes i referens 2.

Enligt kopplingsschemat finns 1 k PROM och 320 bit RAM tillgängligt som minnesutrymme.

På motsatta sidan från frontpanelen finns det på kretskortet fastsatt ett 64-pol kontaktdon. Genom detta kan 4040:n anslutas till andra enheter via en rack. En spänningsgenerator som ger drivspänning till kretsarna är ansluten på detta sätt. En förteckning över vad som är anslutet till de olika polerna finns på sidan 2 i appendix 1. Lagg speciellt märke till de 12 input lines (46-53, 56-59) och de 8 output lines (25-32) samt bussen (60-63), som kommer till användning vid kommunikationen med yttre enheter.

S/F-omvandlare

Funktion: En likspänning 0 - 10 V omvandlas till korta pulser med frekvensen 0 - 2550 Hz .

Kopplingsschema på sidan 3 i appendix 1.

Om en likspänning 0 - 10 V lägges på ingången fås dels pulser och dels en sågtandspänning ut båda med en frekvens som är proportionell mot inspänningen. Frekvensintervallet är 0 - 2550 Hz. Pulserna har amplituden 15 V och har längden 76 s. Linearitetsfelet är mindre än 0,05 %.

Kalibreringar:

Vid en låg spänning (exempelvis 10 mV) justeras potentiometern i trimhål 1 så att utsignalen får rätt frekvens (2,55 Hz).

Vid 10 V inspänning justeras potentiometern i trimhål 2 så att utsignalen får frekvensen 2550 Hz.

Pulsräknare

Funktion: Antalet pulser som kommer in på enheten under en given tid räknas och resultatet överföres till 4040:n.

Kopplingsschema på sidan 4 i appendix 1.

Kretslayout på sidan 5 i appendix 1.

Det finns två pulsräknare på kretskortet, vars ingångar är märkta IN 1 och IN 2. In på enheten sändes pulser med en frekvens på 0 - 2550 Hz. Ut från enheten fås då ett 8-bitars binärt tal proportionellt mot frekvensen in. Detta sändes i två omgångar på en buss till 4040, som alltså måste vara sammankopplad med pulsräknaren via racken.

Pulsräknaren arbetar enligt följande:

(jmf kopplingsschemat och för varje krets funktion se referens 3)
 Pulser räknas under den tid då SR-vippans (CD 4013) ϕ -utgång är satt.
 (om inte overflow inträffar). SR-vippan styrs av CPU:n. Räkningen startas då output line 4 exciteras och samtidigt nollställes räknaren (CD 4040). Mikrodatorn programmeras så att efter 0,1 s exciteras output line 5 varvid räkningen avslutas. Har frekvensen varit 2550 Hz räknas alltså 255 pulser vilket motsvarar det binära talet 1111 1111. Även om vi sänder in en högre frekvens fås samma resultat (overflow)

tack vare användningen av utgång 12 på räknaren. Det binära talet överföres till ingångarna på två multiplexerkretsar (CD 4016). På kommando från CPU:n (input lines 3,4,5 och 6) sänds de fyra mest signifikanta bitarna respektive de fyra minst signifikanta bitarna ut på bussen för att tas hand av 4040:n.

Pulslängdsgivare

Funktion: Ett binärt tal hämtas från 4040:n varefter pulser ställs ut vars längd är proportionell mot detta tal.

Kopplingsschema på sidan 6 i appendix 1.

Kretslayout på sidan 7 i appendix 1.

In på enheten kommer ett binärt tal på åtta bitar från 4040:n. Detta omvandlas till en mot talet proportionell pulslängd 0 - 13 ms alternativt 0 - 2,56 s. Kretsen måste sammankopplas med 4040:n via racken, samt med en krets som ger en fyrkantvåg med frekvenserna 19,2 kHz på pol 21 och 300 Hz på pol 22 (dessa frekvenser finns på en krets kallad Asynchronous Data Interface (UART) som är framtagen på institutionen).

Pulslängdsgivaren arbetar enligt följande: (jmf kopplingsschemat) De fyra mest signifikanta bitarna överföres till den latchkrets (CD 4042), som är märkt med ett C, från bussen då output line 2 aktiveras. Då output line 3 aktiveras överföres dessa till kretsen G samtidigt som de fyra minst signifikanta bitarna överföres till K. Det binära talet jämföres med det tal som bildas av räknarens (CD 4040) utgångar. Då dessa är lika fås en puls efter kretsen A. Denna puls nollställer SR-vippan (CD 4013) vilken sättes då räknaren nollställdes. ϕ -pulsen är proportionell mot det binära talet eftersom räknehastigheten hos räknaren är konstant. När räknaren har kommit upp till 256 nollställes denna och förloppet repeteras. Frekvensen hos pulserna bestäms alltså av frekvensen in på räknaren med en faktor $1/256$.
 Alt 1/ 19,2 kHz in på räknaren ger en maximal pulslängd på 13,3 ms.
 Alt 2/ 300 Hz in på krets B delas ner med en faktor 3 = 100 Hz in på räknaren vilket ger en maximal pulslängd på 2,56 s. Pulslängden är $k \cdot 10$ ms där k är ett heltal $0 \leq k \leq 255$ vilket är lämpligt för att effektreglera 50-periodisk växelström.

Termistorbrygga med probe

Funktion: Temperaturen vid probespetsen omvandlas till en likspänning 0 - 10 V .

Kopplingsschema på sidan 8 i appendix 1.

Kretskortet och panelen är tillverkade så att enheten kan innehålla två bryggor. För det här examensarbetet har endast en brygga behövts och därför saknas komponenter på den undre halvan av kretskortet.

En temperatur 0 - 100 °C på probespetsen omvandlas av enheten till en mot temperaturen proportionell spänning 0 - 10 V på utgången. Själva termistorn finns alltså i spetsen av proben omgiven av Silicone Heat Sink Compound för att ge god termisk kontakt med metallen som bildar ytterhölje till proben. Proben anslutes med en DIN-plugg till frontpanelen till termistorbryggan.

Effektregulator

Funktion: Med hjälp av en styrsignal avgöres hur mycket effekt som skall tas ut från en växelströmskälla (vanligt vägguttag).

Kopplingsschema på sidan 9 i appendix 1.

Om styrsignalen är 15 V så fås som utsignal 220 VAC och lampan tändes. Vid styrsignalen 0 V ligger även utgången på 0 V och lampan är släckt.

Monovippa för avbrottssignal

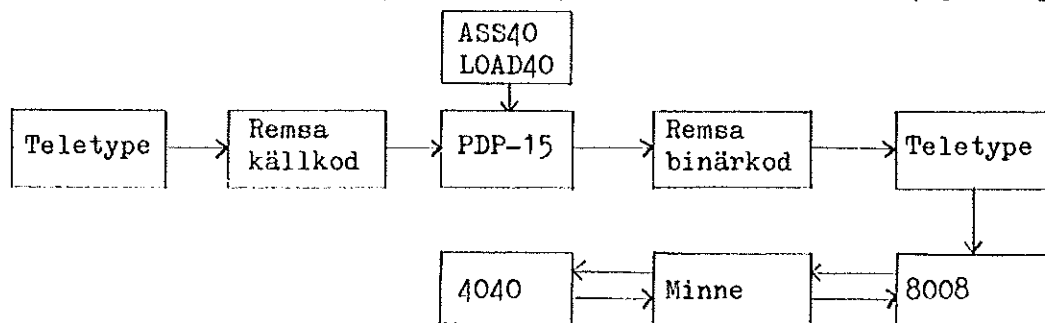
Funktion: Omvandlar en fyrkantvåg till pulser med samma frekvens.

Kopplingsschema på sidan 10 i appendix 1.

Som insignal användes en fyrkantvåg med amplituden 10 V. Monovippan triggar på dess negativa flank och ger som utsignal en puls med en pulslängd på ca 20 μ s, vilket är lagom som avbrottssignal på 4040:n.

Programmeringshjälpmedel

Vid programutvecklingen har följande teknik använts: (jmf figuren)



Ett första utkast till program har skrivits in på en teletype i ett assemblerspråk för 4040 (se referens 2). Remsan från denna har matats in i en dator (PDP-15) där programmet kan editeras. Man kan här lagra programmet på bandminne. På PDP-15 utföres även assemblering och laddning med programmen ASS40 och LOAD40. Assemblern översätter till maskinkod och laddaren länkar subrutiner, ser till att programmet läggs ut i rätt minnesceller och trycker ut resultatet på hålremsa. Denna remsa har matats in i en teletype och innehållet har lagts ut i ett RAM-minne via en mikrodator (Intel 8008). Med hjälp av 8008:n kan man undersöka, stega fram, ladda in och ändra i RAM-minnet. Till sist har minnet använts som programminne åt 4040:n.

Efter uttestning av programmen har dessa lagts in i PROM-minnen vilka appliceras på 4040:s kretskort.

4.

M J U K V A R A

Ett huvudprogram KVOTR och fyra subrutiner MULTH, DIVIH, ROTH och DUB användes för att styra mikrodatorn så att kvotregulatorns funktion uppfylles. Programmen är skrivna i ett assemblerspråk för Intel 4040 och för instruktionernas betydelse se appendix 2 sidan 1 samt referens 2 och 4. Talrepresentationen är 8 bitars binära heltal där 0 motsvarar minimal signal och 1111 1111 motsvarar maximal signal.

Programmets storlek framgår av nedanstående tabell där dels det minnesutrymme som upptas av programmet och dels den tid det tar för datorn att genomlöpa programmet anges. Den senare kan endast anges ungefärligt eftersom den är beroende av vilka tal som ingår i beräkningarna.

Program	Minnesutrymme (antal minnesceller)	Exekveringstid (ms)
KVOTR	109	210 (per loop)
MULTH	110	5,3
DIVIH	89	3,6
ROTH	17	0,2
DUB	60	0,5
	Totalt: 385	

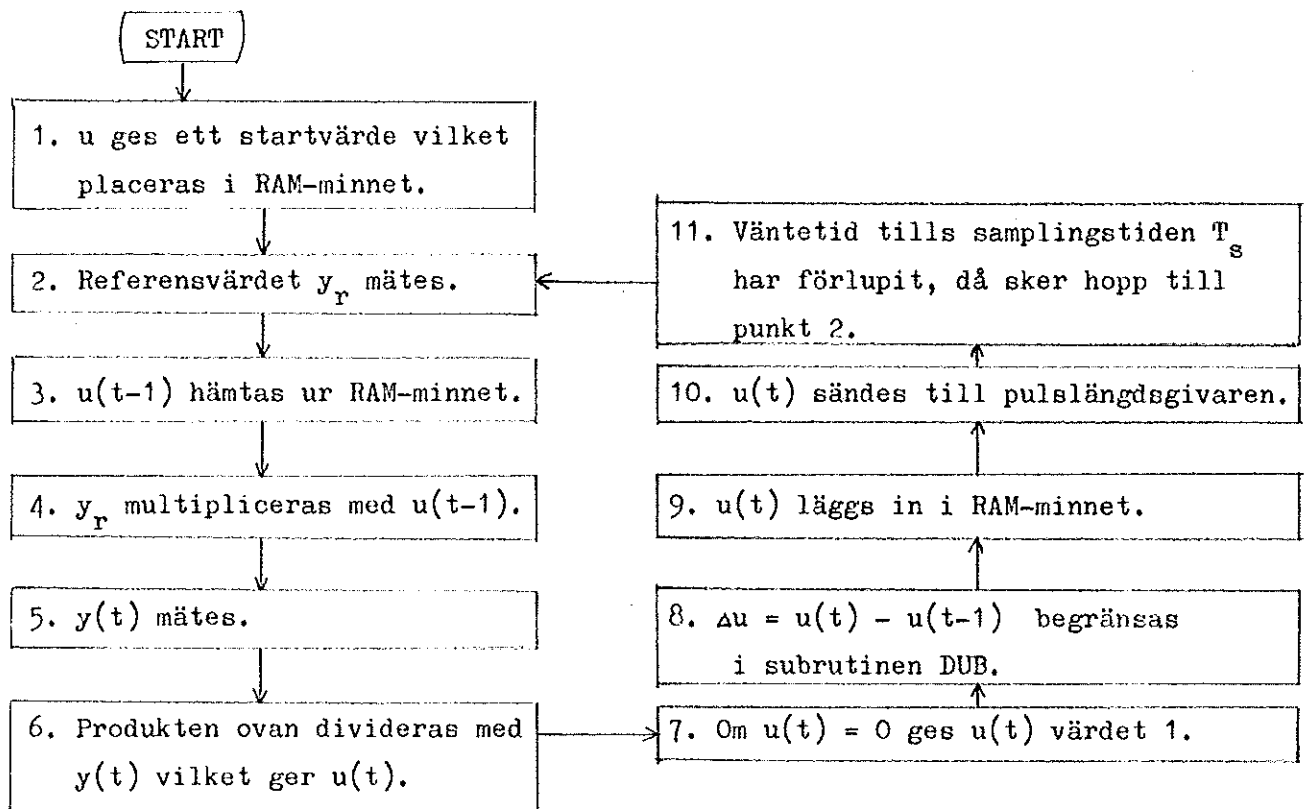
Programmen finns listade i appendix 2. I listningen har modifikationen med rutinen DUB upptagits. För att få basprogrammet strykes bara instruktionen JMS DUB .

Nedan beskrivs programmen var för sig.

KVOTR

Formel: $u(t) = y_r \cdot u(t-1)/y(t)$

Programmets huvuddrag angivet i ett flödesschema:



Kommentarer till ovanstående punkter:

1. Startvärdet på u har valts till $u_{\max}/2$ vilket ger en flexibel regulator. I speciella fall kan dock andra startvärden vara bättre. Värdet lagras i RAM-minnet, chip 0, register 0, status character 2 och 3 (jmf referens 2) och kommande styrsignaler läggs efterhand in i samma celler.
2. Referensvärdet y_r ligger som en spänning mellan 0 V och 10 V på S/F-omv vars utgång har kopplats till ingången IN1 på pulsräknaren. Räknaren startas genom en aktivering av output line 4 (WRR-instruktionen). Rutinen ETS åstadkommer en fördröjning avpassad så att

räknetiden blir 0,1 s. Räknaren stoppas genom en aktivering av output line 5. Aktivering av input line 5 (RDR-instruktionen) slussar in de fyra mest signifikanta bitarna till CPU:N. Aktivering av input line 6 slussar in övriga bitar.

4. Subrutinen MULTH användes. För detaljer se under rubriken MULTH nedan. Rutinen ger som resultat ett 16 bitars tal som får ligga kvar i indexregistren R4 - R7.
5. Samma teknik som under punkt 2 ovan. Observera dock att pulsräknarens ingång IN2 skall användas.
6. Subrutinen DIVIH användes. För detaljer se under rubriken DIVIH nedan. Resultatet $u(t)$ ligger i indexregistren R6 Och R7.
7. Nödvändigt eftersom $u(t) = 0$ förstör algoritmens fortsatta användande.
8. Se under rubriken DUB nedan. Om $|\Delta u| > a$ ändras alltså $u(t)$.
9. Jämför punkt 1 ovan.
10. Det sker huvudsakligen genom en aktivering av output lines 2 och 3.
11. Denna punkt har realiserats med följande teknik. Först möjliggöres avbrott genom satsen enable interrupt (EIN). Därefter snurrar programmet i en miniloop eftersom innehållet i ackumulatorn är nollställt. Då interruptsignal kommer sker automatiskt ett hopp till minnescell 3 i programminnet där en avbrottsrutin finns. I denna laddas ackumulatorn med en etta varefter hopp sker tillbaka till huvudprogrammet. Miniloopen lämnas. Disable interrupt (DIN) ser till att programmet ej avbrytes någon annanstans än i miniloopen. Till sist sker hopp till LOOP dvs till punkt 2 ovan. Samplingstiden är alltså lika med perioden hos interruptsignalen om programmet hinner genomlöpas mellan två signaler.

MULTH

I referens 4 finns en rutin MULT beskriven. Den multiplicerar två 8-bitars tal givna i 2-komplementform med varandra. Faktorerna skall ligga i indexregistren R0,R1 och R2,R3 , medan resultatet fås med 16 bitar i registren R4 - R7. Rutinen utnyttjar följande teknik. Talen delas upp i 4-bitars tal och multipliceras ihop enligt:

$$(A \cdot 2^4 + B) \cdot (C \cdot 2^4 + D) = A \cdot C \cdot 2^8 + (A \cdot D + B \cdot C) \cdot 2^4 + B \cdot D$$

där A och C är de fyra första bitarna av talen och B och D är de fyra sista bitarna

MULT har omarbetats så att den räknar på heltal och kallas då för MULTH. Följande ändringar har gjorts:

- o Nolltesten av faktorerna (instruktionerna 4 - 21 i MULT) har tagits bort eftersom KVOTR ser till att ingen av faktorerna kan vara noll.
- o ABSOL-rutinen och anropen av densamma (instruktionerna 22 - 33 i MULT) har plockats bort då endast positiva tal behandlas.
- o Då heltal multipliceras hamnar resultatet direkt i rätt position varför rutinen ROT ej behöver anropas (instruktion 130 i MULT).
- o En addition har glömts bort i MULT och därför har instruktionerna CLC och ADD R4 lagts till efter insrtuktion 117 i MULT.

DIVIH

I referens 4 finns en rutin DIV beskriven. Den dividerar ett 16-bitars tal i indexregistren R4 - R7 med ett 8-bitars tal i R0,R1. Det senare givet i 2-komplementform. Resultatet läggs i R6,R7. Rutinen använder sig av Burks algoritm för division utan återställning som beskrivs noga i referens 5.

DIV har omarbetats så att den räknar på heltal och kallas då för DIVIH. Följande ändringar har gjorts:

- o ABSOL-rutinen och anropet av densamma (instruktion 3 i DIV) har tagits bort då endast ett positivt tal är aktuellt från början.

- o Den del av programmet DIV som kommer under labels ROTR och TWOSC (instruktionerna 50 - 107 i DIV) har tagits bort eftersom de överför resultatet till tvåkomplementform vilket ej är önskvärt.
- o För att kunna använda tekniken i DIV för heltal har både dividend och divisor högerskiftats ett steg med nollutfyllnad från vänster. Divisorns sista bit har lagts i R2:s första position och dividendens sista bit har placerats i R9:s sista position. I de additioner, subtraktioner och högerskift som utföres i rutinen har hänsyn tagits till ettstegsförskjutningen.
- o Om divisorn är noll blir resultatet av divisionen det största möjliga representerbara talet dvs 1111 1111.

ROTH

I referens 4 finns rutinen ROT beskriven. Den vänsterskiftar indexregistren R4 - R7 ett steg med nollutfyllnad. För att passa till DIVIH har vänsterskiftet utökats med en bit nämligen sista biten i R9 som placerats längst till höger i registerkedjan.

DUB

Denna rutin begränsar $\Delta u = u(t) - u(t-1)$ på så sätt att $|\Delta u| \leq a$ där a väljes vid programmeringen. Om ovanstående relation ej är uppfylld av det framräknade $u(t)$ ersättes detta med $u(t-1) \pm a$ beroende på tecknet på Δu .

Beräkningsgång i rutinen:

1. $u(t) - u(t-1) \rightarrow \Delta u$
2. Om $\Delta u < 0$ så $-\Delta u \rightarrow |\Delta u|$ och $1 \rightarrow R0$ annars $\Delta u \rightarrow |\Delta u|$ och $0 \rightarrow R0$.
3. Om $|\Delta u| - a > 0$ så, om $R0 = 0$ så $u(t) = u(t-1) + a$ annars $u(t) = u(t-1) - a$, annars ändras ej $u(t)$.

Modifikationer av KVOTR

Några möjliga ändringar av algoritmen:

1. Insignalens förändring kan begränsas genom användandet av subrutinen DUB. Parametern a bestäms i rutinens andra instruktion. Satsen JMS DUB skjutes in efter sats 107 i KVOTR.
2. u_{\min} kan ökas från värdet 1. Som exempel väljes $u_{\min} = 16$. Då skall sats 104 JPZ B ändras till XCH R7 och sats 107 XCH R7 skall ändras till XCH R6.
3. Om mätningen av y_r lägges utanför huvudloopen i KVOTR kortas exekveringstiden med ca 0,1 s vilket möjliggör mindre värden på samplingstiden. Nackdelen med detta är att y_r ej kan ändras medan 4040:n exekverar KVOTR utan programmet måste omstartas för att få in det nya värdet. Praktiskt sker modifikationen genom att flytta labeln LOOP till sats 41 SRC R0.
4. Värdet på $u(0)$ kan ändras med satserna 11 och 13 i KVOTR.

5. U P P K O P P L I N G S I N S T R U K T I O N

I System på analogimaskin (jmf systemskiss på sidan 8)

- o Placera PROM-minnen innehållande lämpligt program på 4040:s kretskort.
- o Placera 4040:n, pulsräknaren, pulslängdsgivaren och den kalibrerade (se sidan 10) S/F-omv i racken.
- o Koppla upp ett system på analogimaskinen. Anslut ingången och utgången till tie-points.
- o Systemets utsignal kopplas till en av S/F-omv gröna ingångar. Motsvarande utgång på S/F-omv kopplas till pulsräknarens ingång IN 2.
- o Pulslängdsgivarens utsignal dämpas med en faktor $2/3$ med en potentiometer på analogimaskinen (för att få ner maxspänningen från 15 V till 10 V) och kopplas sedan som insignal till systemet. Observera att pulslängdsgivaren måste få en lämplig frekvens via kontaktdonet på baksidan av kortet. Denna fås enklast genom att placera enheten Asynchronous Data Interface (UART) i racken så att kontakt erhålles enheterna emellan.
- o Referensspänningen y_r (0 - 10 V) hämtas från analogimaskinen och kopplas in på den icke använda ingången på S/F-omv, vars utgång ansluts till pulsräknarens ingång IN 1.
- o Fyrkantvågen med full amplitud från analogimaskinens signalgenerator användes som insignal till monovippan. Den senares utsignal kopplas till det vita interruptuttaget på 4040:n. Periodtiden hos vågen är lika med samplingstiden T_s .
- o Spänningsgeneratoren i racken och analogimaskinen slås på. Omkopplaren på 4040:n slås över i läge RUN varvid regleringen kan börja om analogimaskinen arbetar i OPERATE.

II Vattenvärmning (jmf systeskiss på sidan 8)

- o Samma kopplingar som ovan om ordet system tolkas som effektregulator, värmeplatta, vatten i ett kärl och termistorbrygga med probe tillsammans.
- o Pulslängdsgivarens utsignal är då systemets insignal och kopplas till styringången på effektregulatorn.
- o I det jordade uttaget på den senare kopplas värmeplattan.
- o Ett kokkärl fylles med vatten och ställes på värmeplattan.
- o Termistorprobens spets sänks ner i vattnet och proben kopplas till termistorbryggan.
- o Utgången från termistorbryggan är i detta fallet systemets utsignal och kopplas alltså till en S/F-omv.

6.

E X P E R I M E N T

Kvotregulatorn har testats mot två olika typer av system.

- I Linjära system uppkopplade på analogmaskin där in- och utsignaler fysikaliskt är elektriska spänningar.
- II Ett system som består av en mängd vatten som värms upp av en kokplatta. Insignalen till systemet är den effekt som tillföres kokplattan och utsignalen är temperaturen hos vattnet.

I System på analogmaskin

Styrsignalen

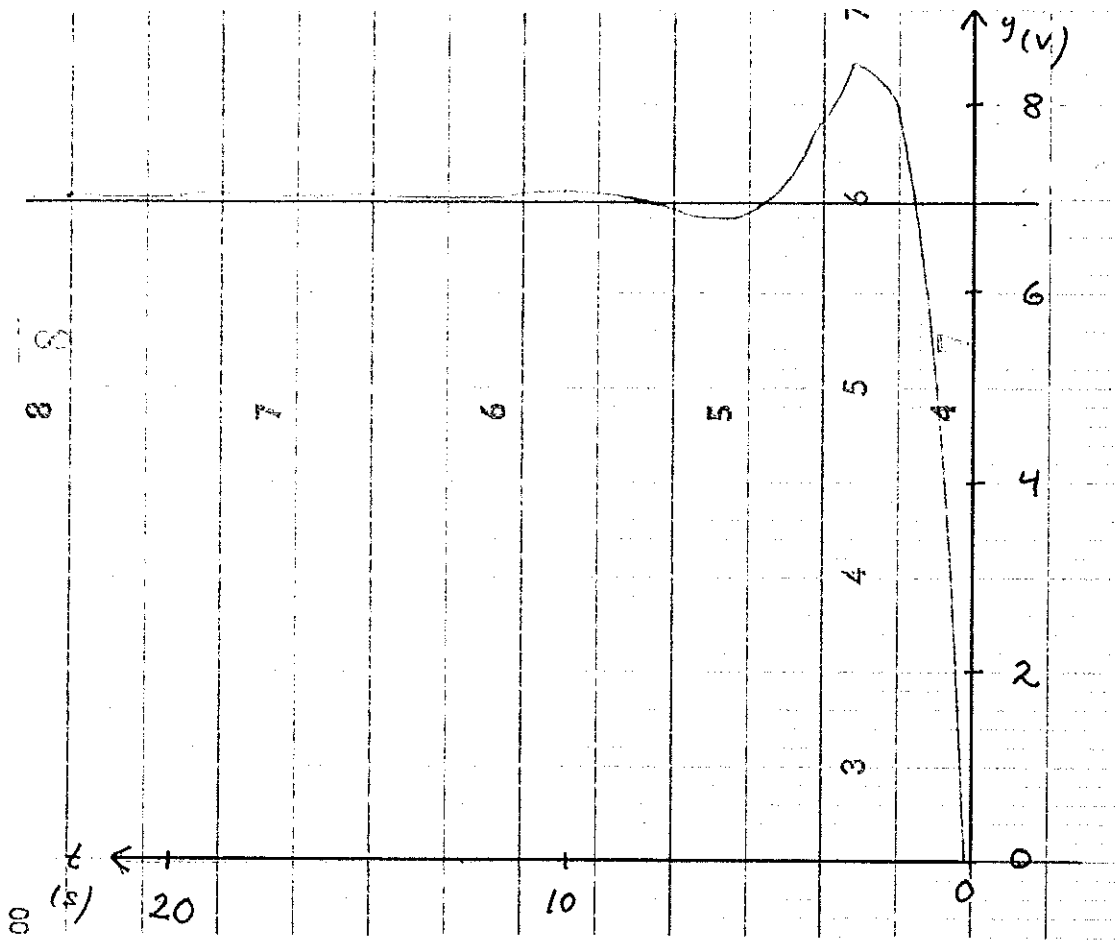
Styrsignalen skall enligt teorin vara konstant under samplingsintervallen. Detta kan uppnås genom att medelvärdesbilda utsignalen från pulslängdsgivaren med något enkelt filter. Problem uppstår emellertid gärna med dynamiken hos filtret. Problemet kringgås i vårt fall genom att de undersökta systemen är långsamma i förhållande till periodtiden (13 ms) hos den pulsmodulerade styrsignalen. Denna signal kan då användas direkt som insignal till systemet.

1:a och 2:a ordningens system

Basprogrammet KVOTR har körts mot systemet $G_1 = \frac{1}{s+1}$ med $y_r = 7$ V, $T_s = 1$ s och begynnelsevärdet $y(0) = 0$ V. Resultatet finns i figur 1. Skalan på tidsaxeln är $\frac{1}{320\text{mm}/\text{min}} = \frac{1}{5,3\text{mm}/\text{s}} = 1,9$ s/cm . Utsignalen y svänger alltså in sig mot y_r efter en relativt stor översläng.

För motsvarande försök där endast systemet har ändrats till $G_2 = \frac{1/8}{(s+1/4)(s+1/2)}$ återges utsignalen i figur 2. Här har tidsaxeln krympts ihop med en faktor 4. Vi får här ett långsammare gensvar från systemet vilket är att vänta eftersom det har större tidskonstanter än G_1 . Dessutom blir överslängen här ännu större än i förra fallet.

Av dessa försök framgår att regulatorn fungerar för dessa system och svänger in sig snabbt och bra mot y_r efter en första översläng som dock är stor.



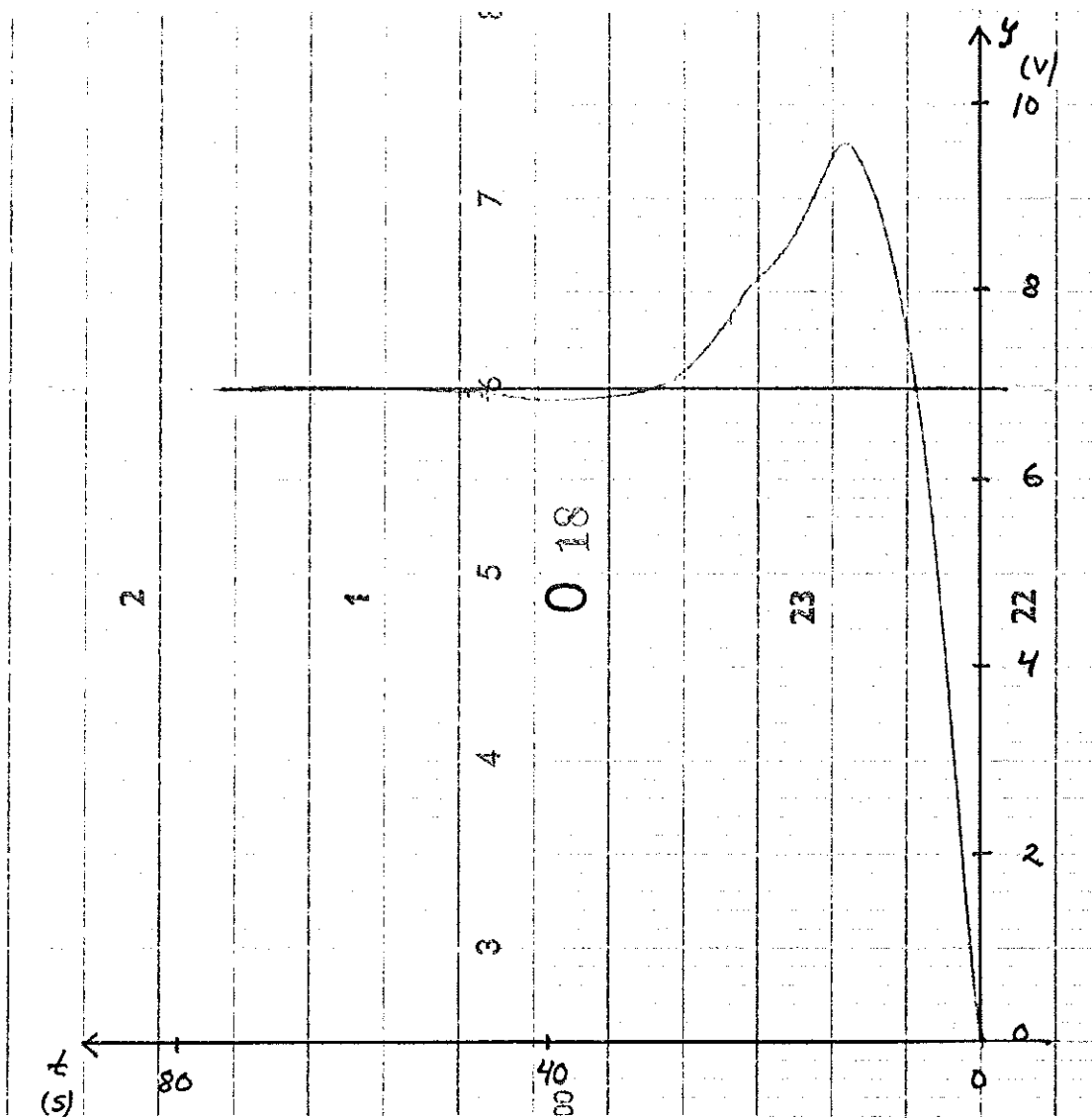
Figur 1

System: G_1

$T_s = 1$ s

$y_r = 7$ V

$y(0) = 0$ V



Figur 2

System: G_2

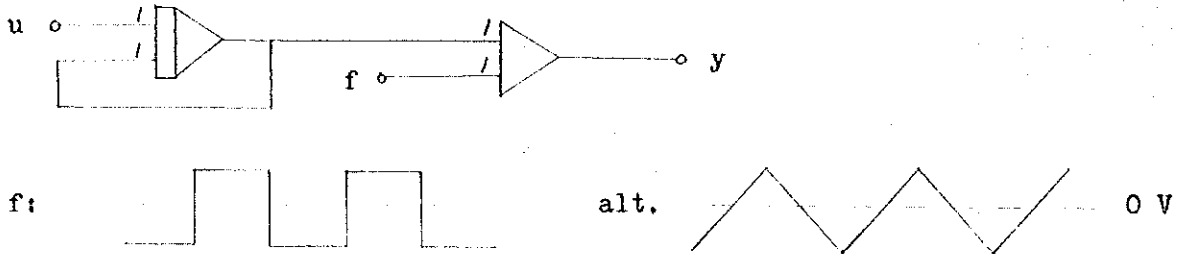
$T_s = 10$ s

$y_r = 7$ V

$y(0) = 0$ V

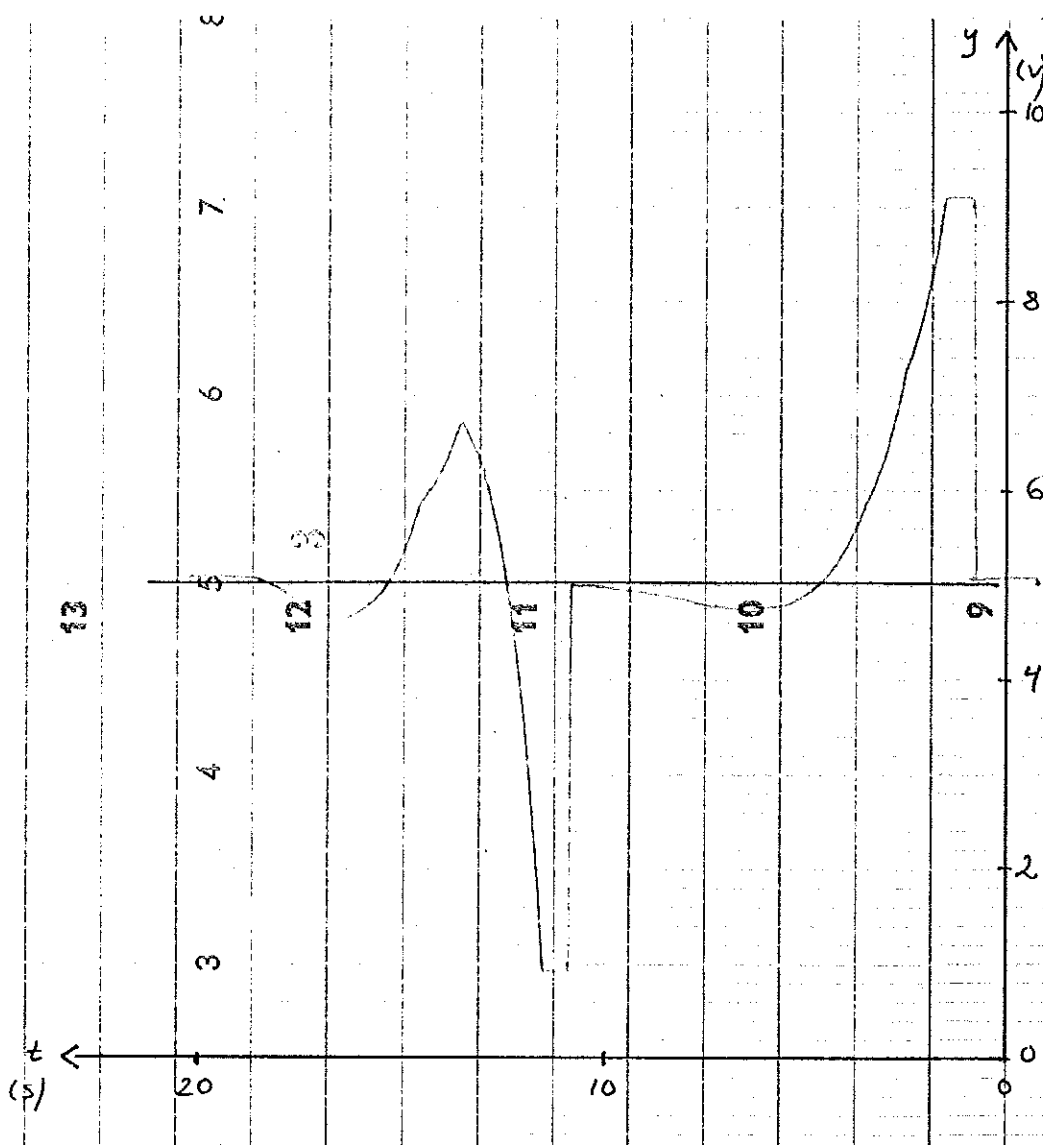
Steg- och rampstörningar

Störningarna har lagts in på följande sätt:



I figur 3 och 4 visas vad som händer då stegstörningar med amplituden 4 V respektive 2 V appliceras på systemet G_1 ($y_r = 5$ V, $T_s = 1$ s).

En positiv störning dämpas ut relativt långsamt medan en negativ kompenseras snabbt med en stark insignal vilket ger en stor översläng.



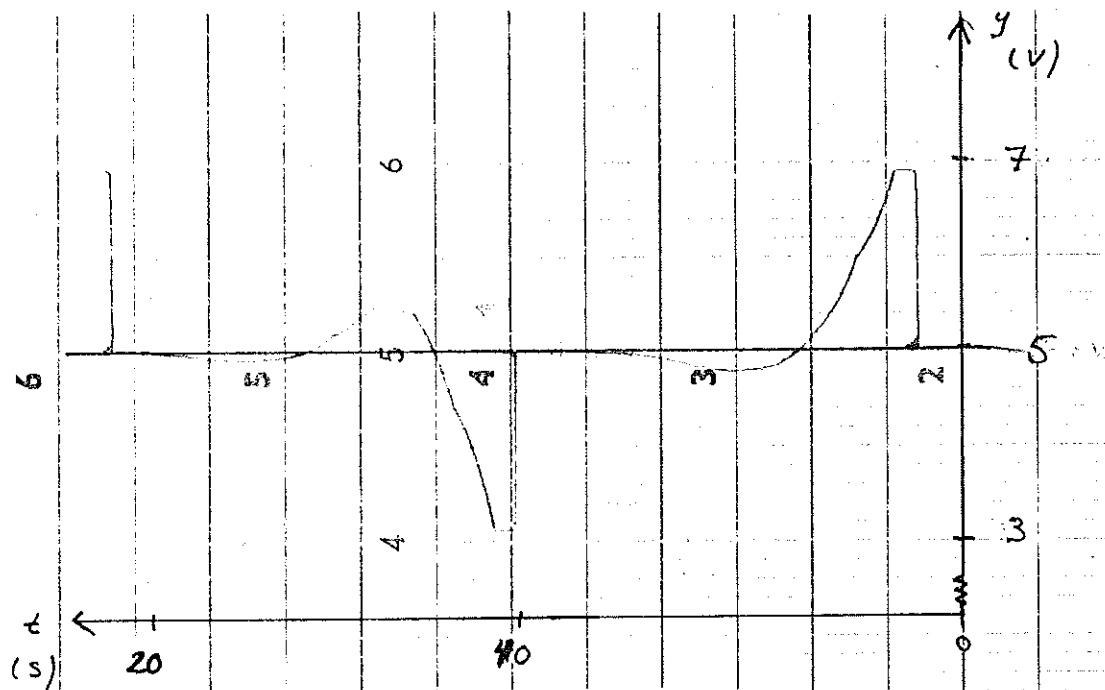
Figur 3

System: G_1

$y_r = 5$ V

Störning: ± 4 V

$T_s = 1$ s

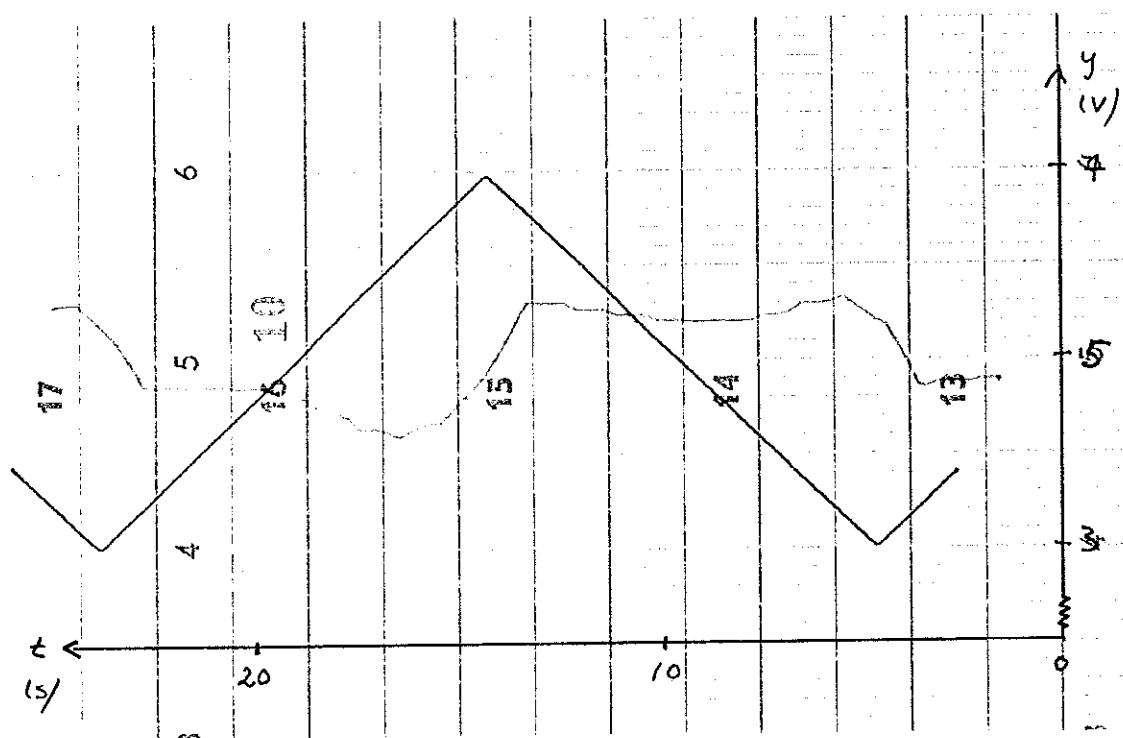


Figur 4

System: G_1 $T_s = 1$ s $y_r = 5$ VStörning: ± 2 V

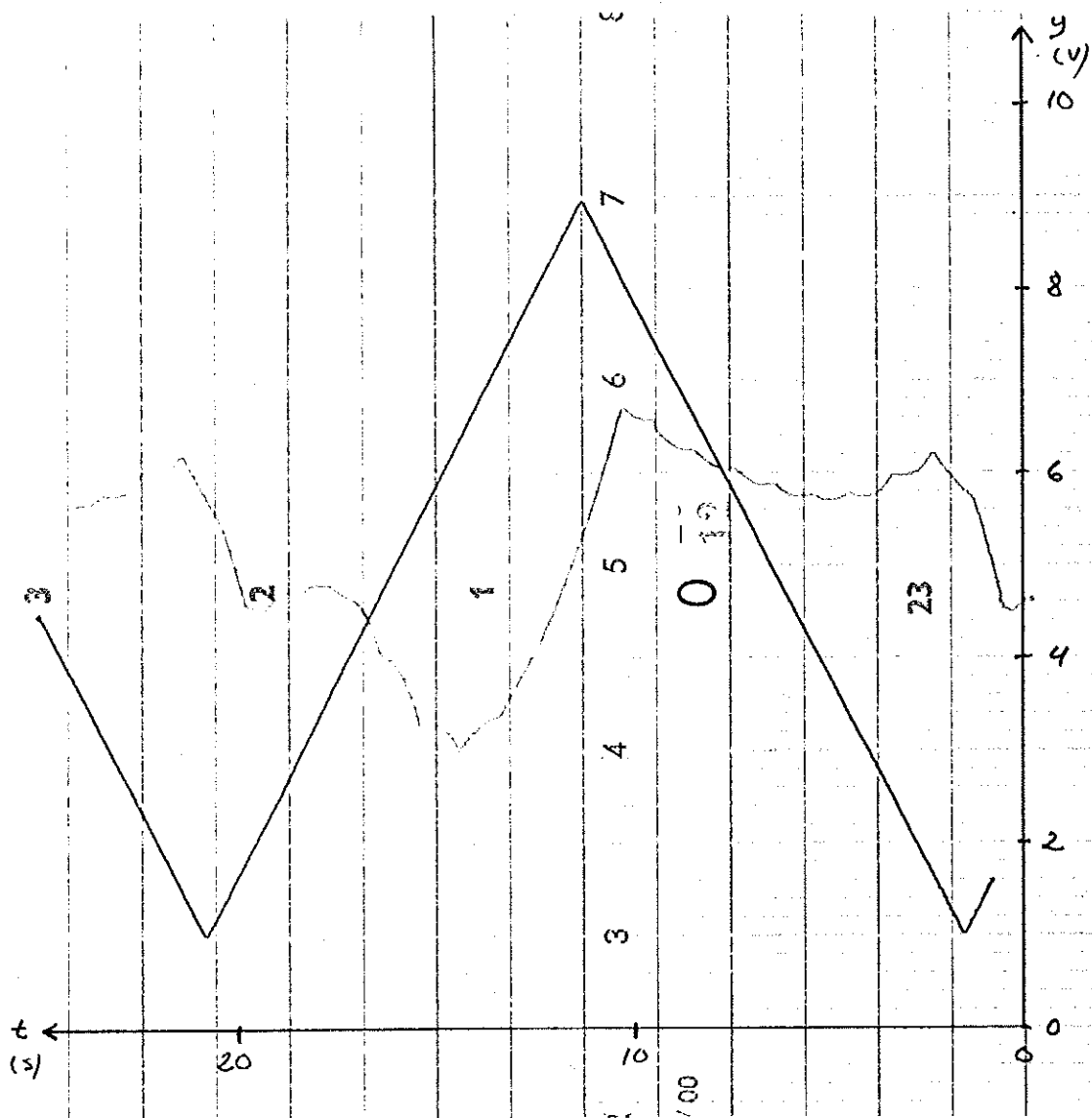
I figur 5 och 6 visas utsignalen då systemet störs av rampstörningar $0,4$ V/s respektive $0,8$ V/s. Störningens tidsmässiga inplacering och amplitud åskådliggöres av den blåa linjen.

I båda fallen fås ett stationärt fel. Regulatorn klarar alltså ej av att reglera ut störningarna helt. Vid den mindre störningen kompenserar regulatorn denna så att maximala avvikelser från y_r är ca $0,8$ V. Vid den större störningen blir motsvarande maximala avvikelse ca 2 V. Här framgår även att kvotregulatorn lättare klarar av en avtagande ramp än en växande.



Figur 5

System: G_1 $T_s = 1$ s $y_r = 5$ VStörning: $\pm 0,4$ V/s



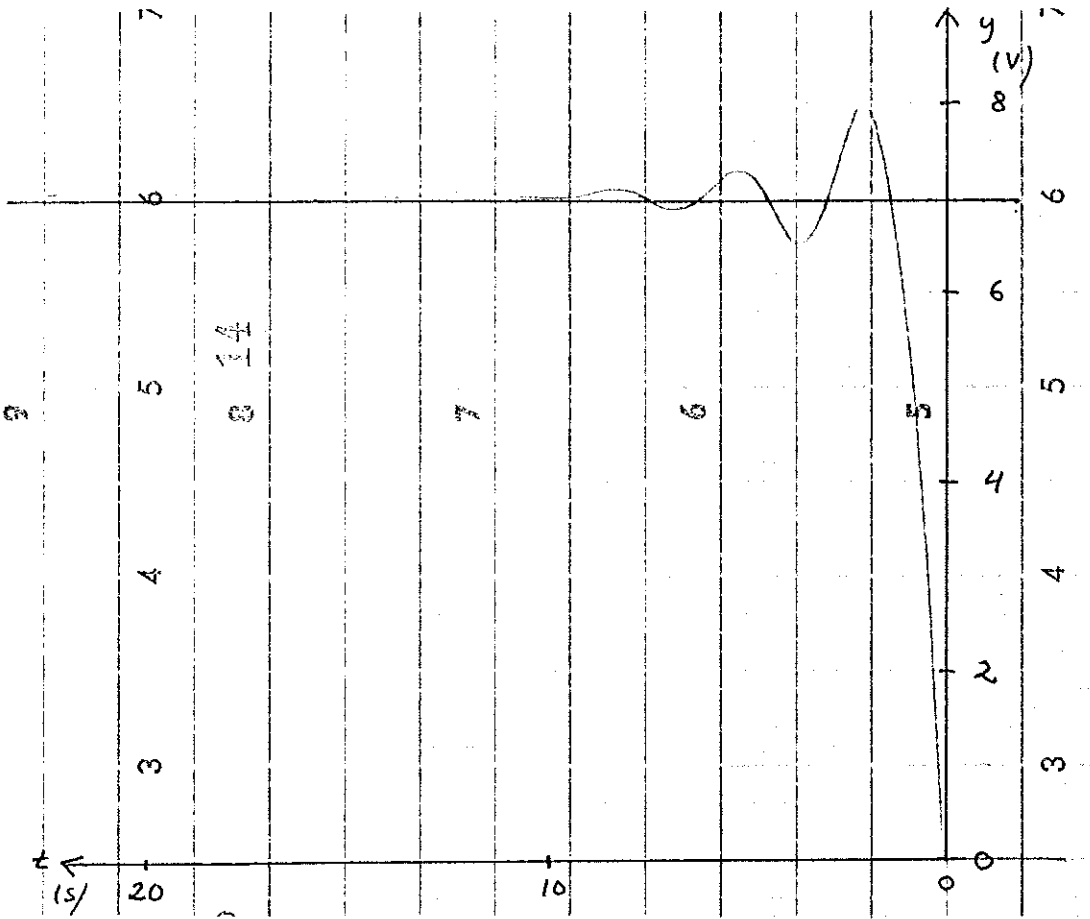
Figur 6

System: G_1 $T_s = 1 \text{ s}$ $y_r = 5 \text{ V}$ Störning: $\pm 0,8 \text{ V/s}$

Variation av T_s

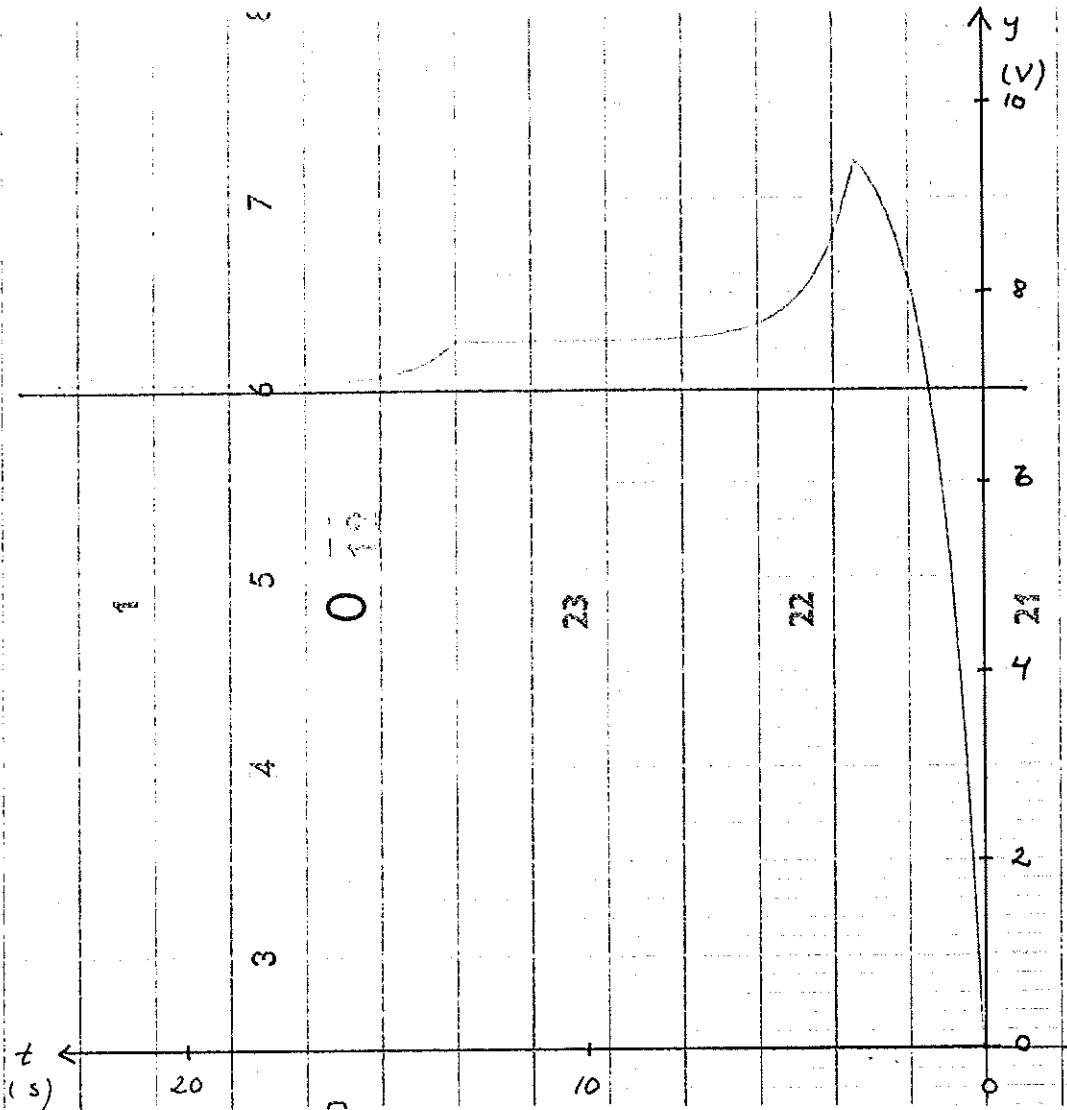
Figur 7 och 8 visar utsignalen vid samplingstiderna 0,22 s respektive 10 s då systemet är $G_1 = \frac{1}{s+1}$, $y_r = 7 \text{ V}$ och $y(0) = 0 \text{ V}$. Dessa kurvor kan jämföras med figur 1 där motsvarande försök har körts med samplingstiden 1 s. Samplingstiden 0,22 s är ungefär lika med exekveringstiden för KVOTR och är alltså minsta möjliga samplingstid. Den har erhållits genom att ha en hög frekvens hos interruptsignalen (100 Hz).

Av diagrammen framgår att systemet är stabilt för alla T_s . Det svänger mindre med ökande storlek på T_s men samtidigt blir överslängen större. Vid $T_s = 10 \text{ s}$ ligger utsignalen onödigt länge på värdet 7,5 V, systemet så att säga bara väntar på nästa sampling.



Figur 7

System: G_1
 $T_s = 0,22 \text{ s}$
 $y_r = 7 \text{ V}$
 $y(0) = 0 \text{ V}$



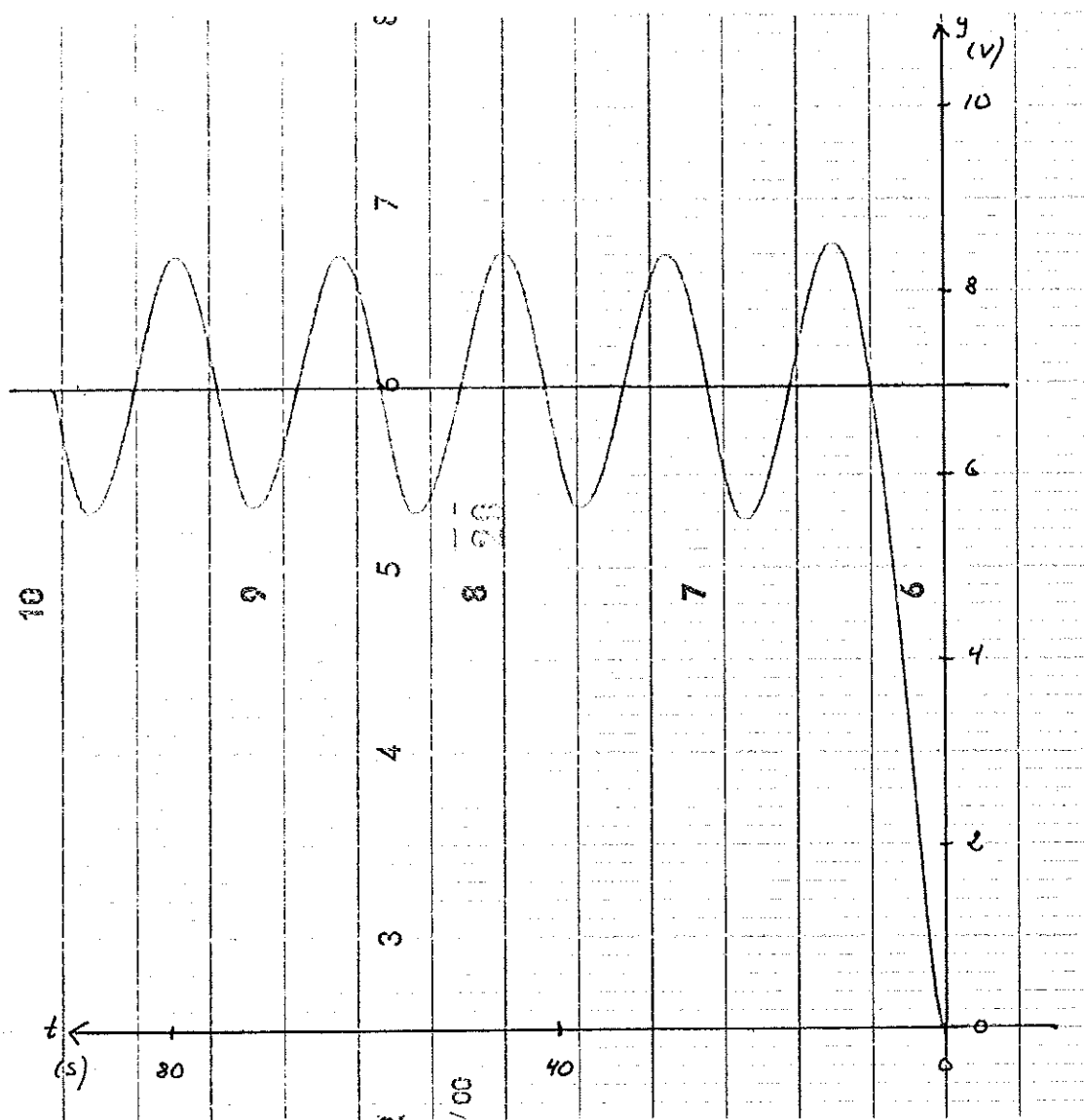
Figur 8

System: G_1
 $T_s = 10 \text{ s}$
 $y_r = 7 \text{ V}$
 $y(0) = 0 \text{ V}$

Figur 9 och 10 visar utsignalen vid samplingstiderna 1 s respektive 2 s då systemet är $G_2 = \frac{1/8}{(s+1/4)(s+1/2)}$, $y_r = 7$ V och $y(0) = 0$ V. Dessa kurvor kan jämföras med figur 2 där motsvarande försök har körts med samplingstiden 10 s. Observera tidsskalan.

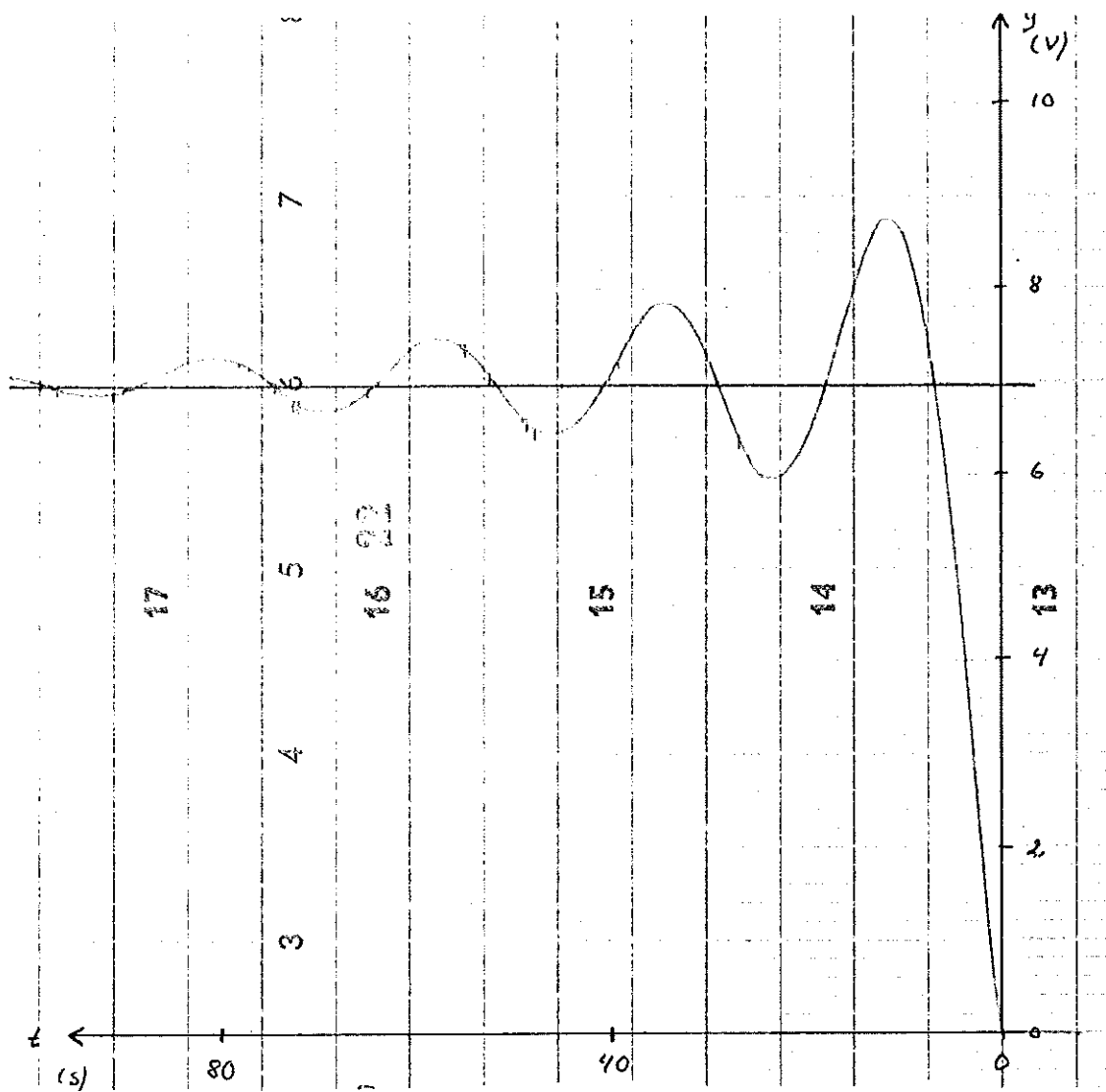
Även här ökar svängningarna med minskande T_s medan överslängen minskas. Vid $T_s = 1$ s dör ej svängningen ut. Det finns således en stabilitetsgräns mellan $T_s = 1$ s och $T_s = 2$ s.

Överhuvudtaget visar dessa försök att kvotregulatorns inverkan på systemet är i mycket hög grad beroende av samplingstiden. Denna bör alltså vid praktisk tillämpning av kvotregulatorn nogra utprovas.



Figur 9

System: G_2
 $T_s = 1$ s
 $y_r = 7$ V
 $y(0) = 0$ V



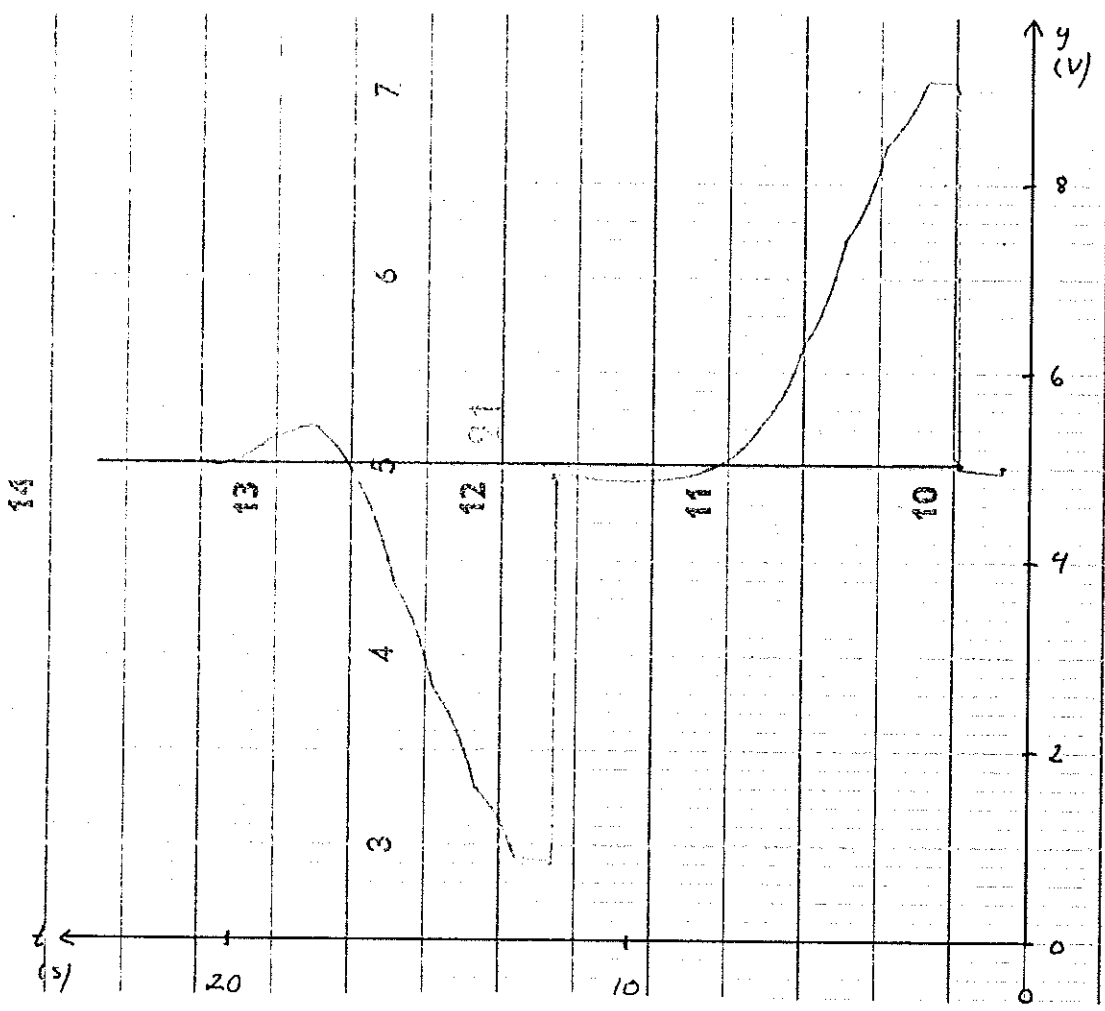
Figur 10

System: G_2
 $T_s = 2$ s
 $y_r = 7$ V
 $y(0) = 0$ V

Begränsning av insignalens ändringsmöjligheter

Då utsignalen svänger kraftigt svänger även insignalen ordentligt. Genom att dämpa insignalens svängningsmöjligheter bör även utsignalen "lugnas ner". Rutinen DUB begränsar insignalsändringen till maximalt $1/4$ av hela intervallet.

I figur 11 och 12 visas subrutinens inverkan på stegstörning och rampstörning. Figurerna skall jämföras med figur 3 respektive 6 där försöksbetingelserna var desamma. Den största fördelen med begränsningen är den stora nedminskningen av överslängen efter en negativ stegstörning, vilket dock sker på bekostnad av snabbheten i kompenseringen. I övrigt fås endast en något jämnare signal.



Figur 11

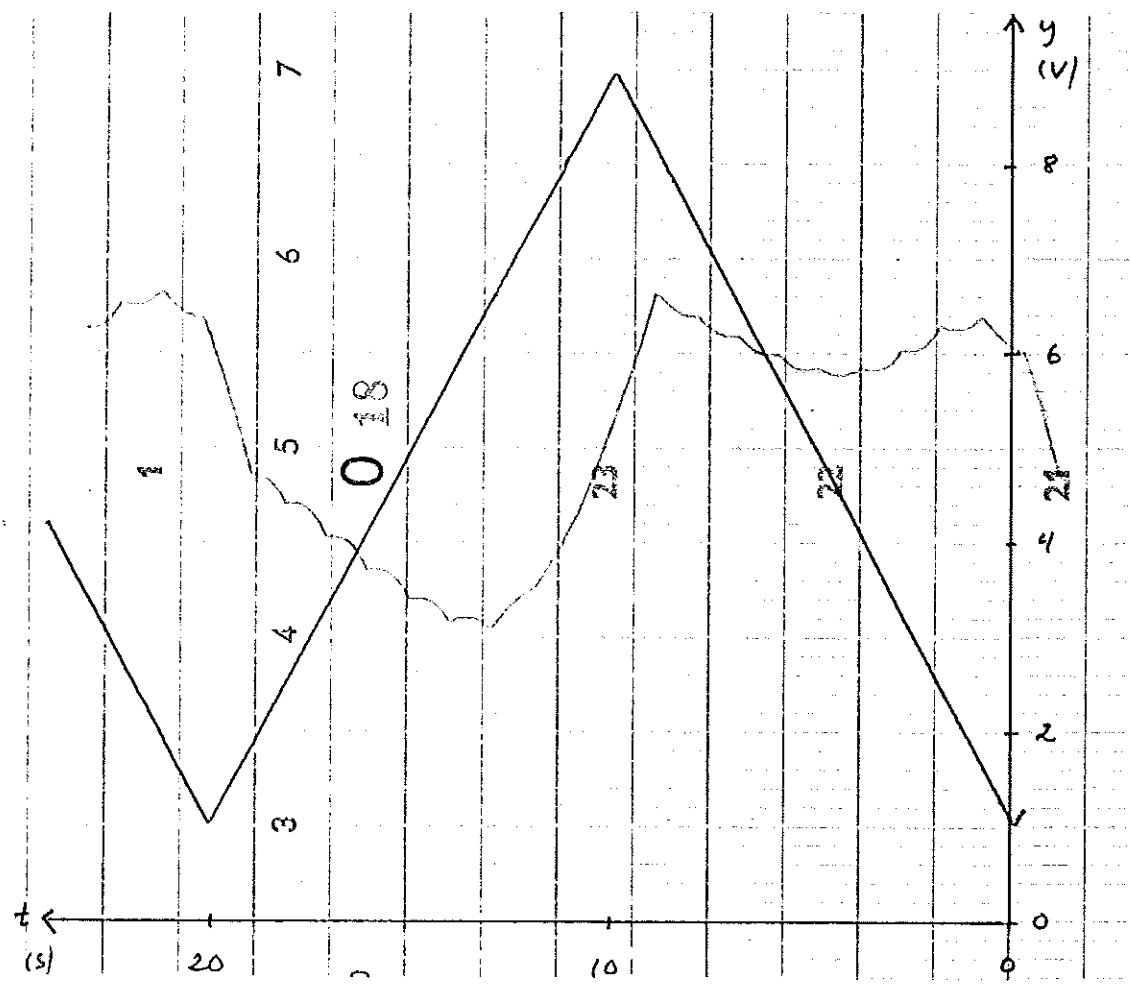
DUB

System: G_1

$T_s = 1$ s

$y_r = 5$ V

Störning: ± 4 V



Figur 12

DUB

System: G_1

$T_s = 1$ s

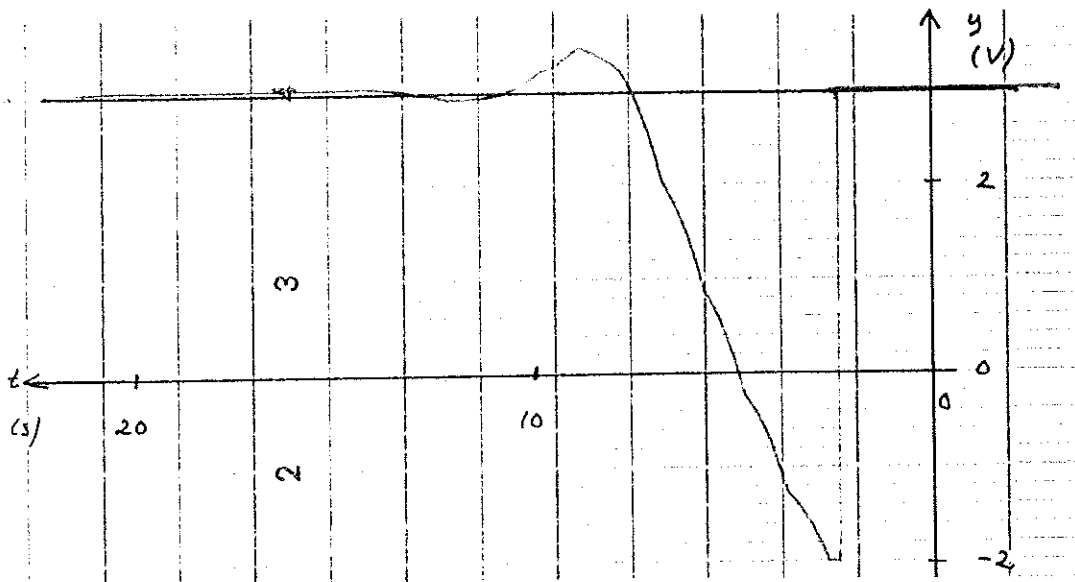
$y_r = 5$ V

Störning: $\pm 0,8$ V/s

Framkoppling

Om utsignalen $y(t)$ blir negativ vid någon störning fungerar kvotregulatorn ej matematiskt om systemet ej framkopplats enligt figur på sidan 6. I vår uppställning tolkas emellertid ett negativt y som $y = 0$ V (på grund av S/F-omv) vilket medför att maximal styrsignal ställs ut. Detta ökar utsignalen så att denna åter kan bli positiv.

Experiment har gjorts med följande förutsättningar: system $G_1 = \frac{1}{s+1}$, $T_s = 1$ s, $y_r = y(0) = 3$ V. En stegstörning på minus 5 V lades på systemet. Resultatet med framkoppling ($y_0 = 3$ V) respektive utan framkoppling kan studeras i figur 13 och 14. Störningssvaren är nästan identiska. Framkoppling är alltså onödig med vår implementering av kvotregulatorn.

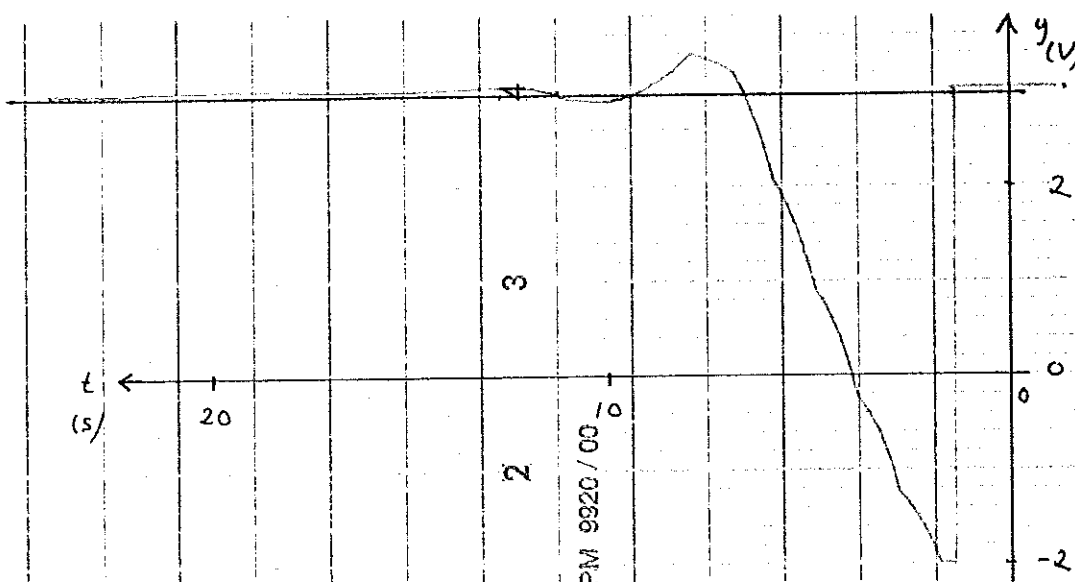


Figur 13

Med framkoppling

System: G_1 $T_s = 1$ s $y_r = 3$ V $y(0) = 3$ V

Störning: - 5 V

 $y_0 = 3$ V

Figur 14

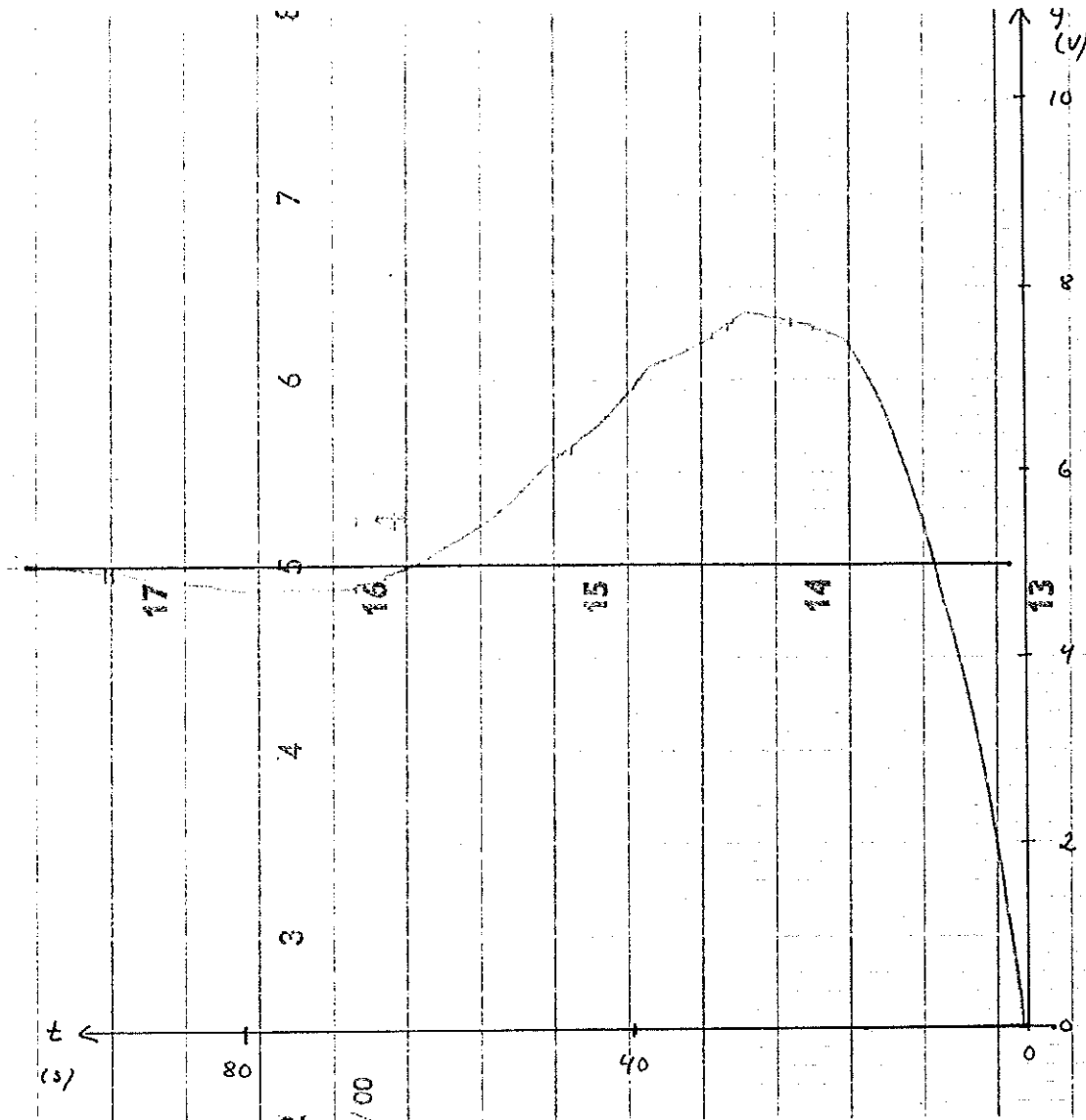
Utan framkoppling

System: G_1 $T_s = 1$ s $y_r = 3$ V $y(0) = 3$ V

Störning: - 5 V

Långsamt system

Kvotregulatorn fungerar bäst då styrsignalen sakta ändrar sig med små amplituder från $u_{\max}/2$. Långsamma system kräver långa samplingsperioder för att styrsignalen ej skall göra för kraftiga utslag från mittvärdet och orsaka svängningar. Detta medför att det tar lång tid att nå det stationära tillståndet. Ett exempel ges i figur 15 där systemet är $G_3 = \frac{1/10}{s+1/10} = \frac{1}{10s+1}$, $T_s = 10$ s, $y_r = 5$ V och $y(0) = 0$ V. Observera att tidsskalan är 4 gånger den som använts vid övriga försök på 1:a ordningens system. Förutom den långa tid som krävs blir dessutom överslängen enormt stor. Regulatorn fungerar alltså ej bra på långsamma system.



Figur 15

System: G_3
 $T_s = 10$ s
 $y_r = 5$ V
 $y(0) = 0$ V

II Vattenvärmning *****

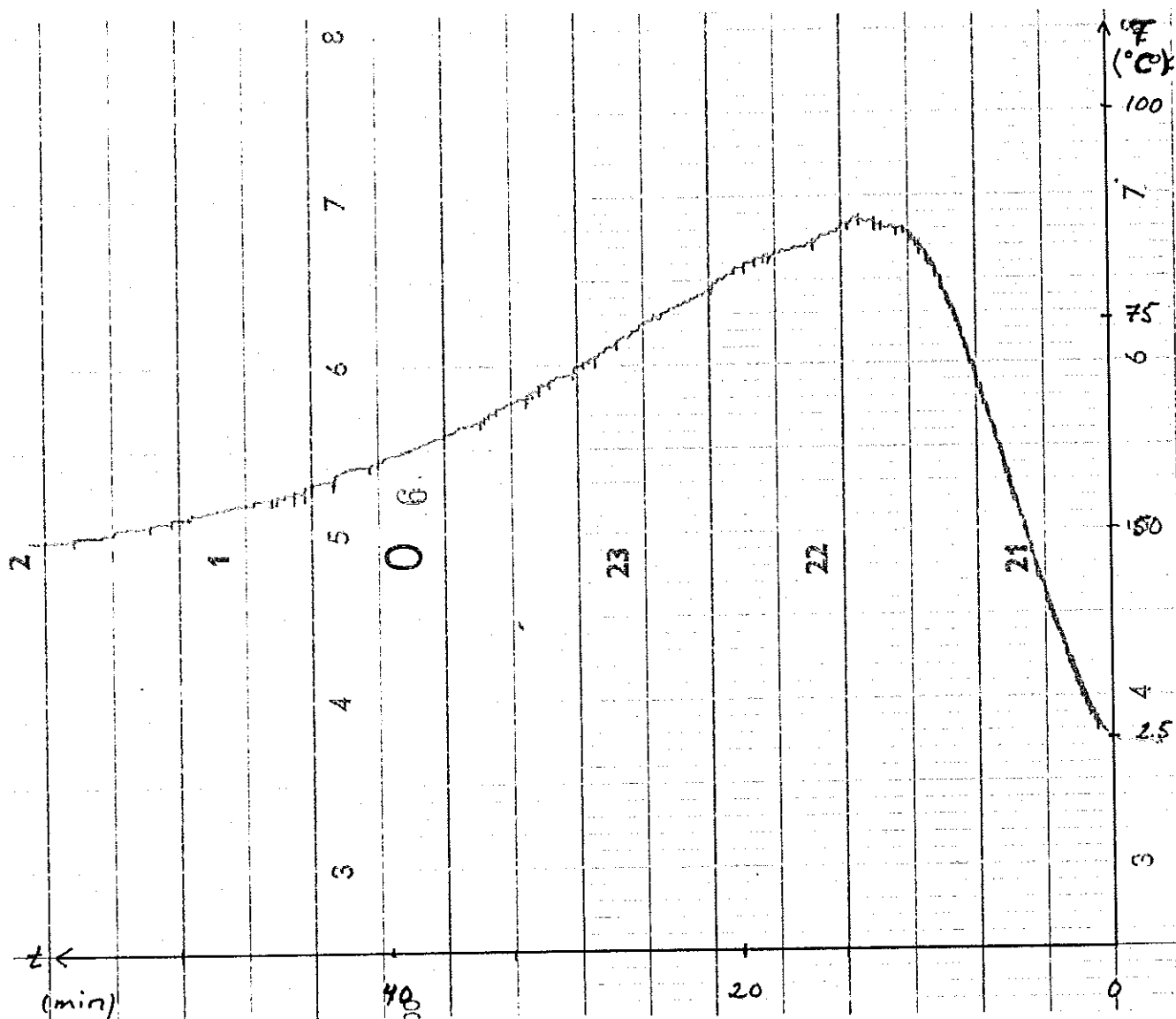
Systemet är idetta fallet $\frac{1}{2}$ liter vatten i ett kokkärl på en värmeplatta. Systemet är olinjärt och är dessutom olika trögt för uppvärmning respektive avkylning, dvs det kan karakteriseras med olika tidskonstanter. Dessa är båda stora och vi får samma problem som vid långsamma linjära system. T_s måste väljas stort för att insignalen ej skall variera för snabbt och orsaka svängningar.

I figur 16 visas hur systemet reagerar vid följande betingelser: $y_r = 50$ °C, $y(0) = 25$ °C och $T_s = 5$ minuter. Vi får en mycket stor översläng och det tar nästan 1 timme innan referensvärdet nås efter densamme. Regulatorn är alltså ej bra för detta system.

Figur 16

System: Vatten i ett kokkärl

$$T_s = 5 \text{ min} \quad y_r = 50 \text{ } ^\circ\text{C} \quad y(0) = 25 \text{ } ^\circ\text{C}$$



7.

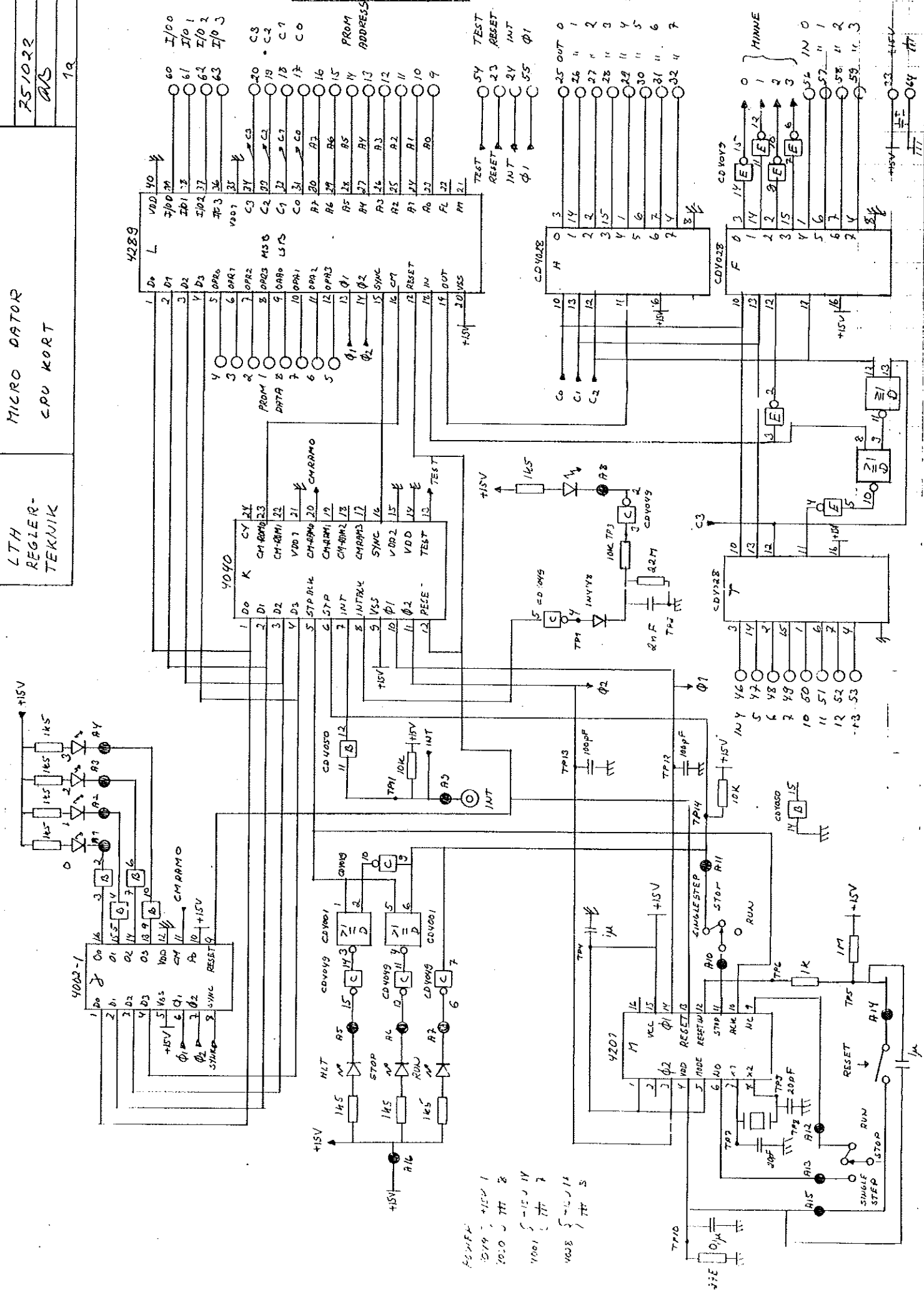
R E F E R E N S E R

1. Lars Jensen " Digital styrning av klimatprocesser ".
2. INTEL MCS-40 " Users manual ".
3. Motorola, Semiconductor Data Library Volume 5
" CMOS Integrated Circuits ".
4. P. Wolgast - J. Miszczuk " Programmeringshjälpmedel för INTEL 4040 ".
5. Hallberg - Danielsson " Digital teknik Del 5-6 ".

LTH
REGLER-
TEKNIK

MICRO DATOR
CPU KORT

R51022
GAS
1a



- 5024 15V 1
- 5020 15V 2
- 5001 15V 3
- 5002 15V 4
- 5003 15V 5

100
 0000
 0000

64-pin kontakt datt till rack

2

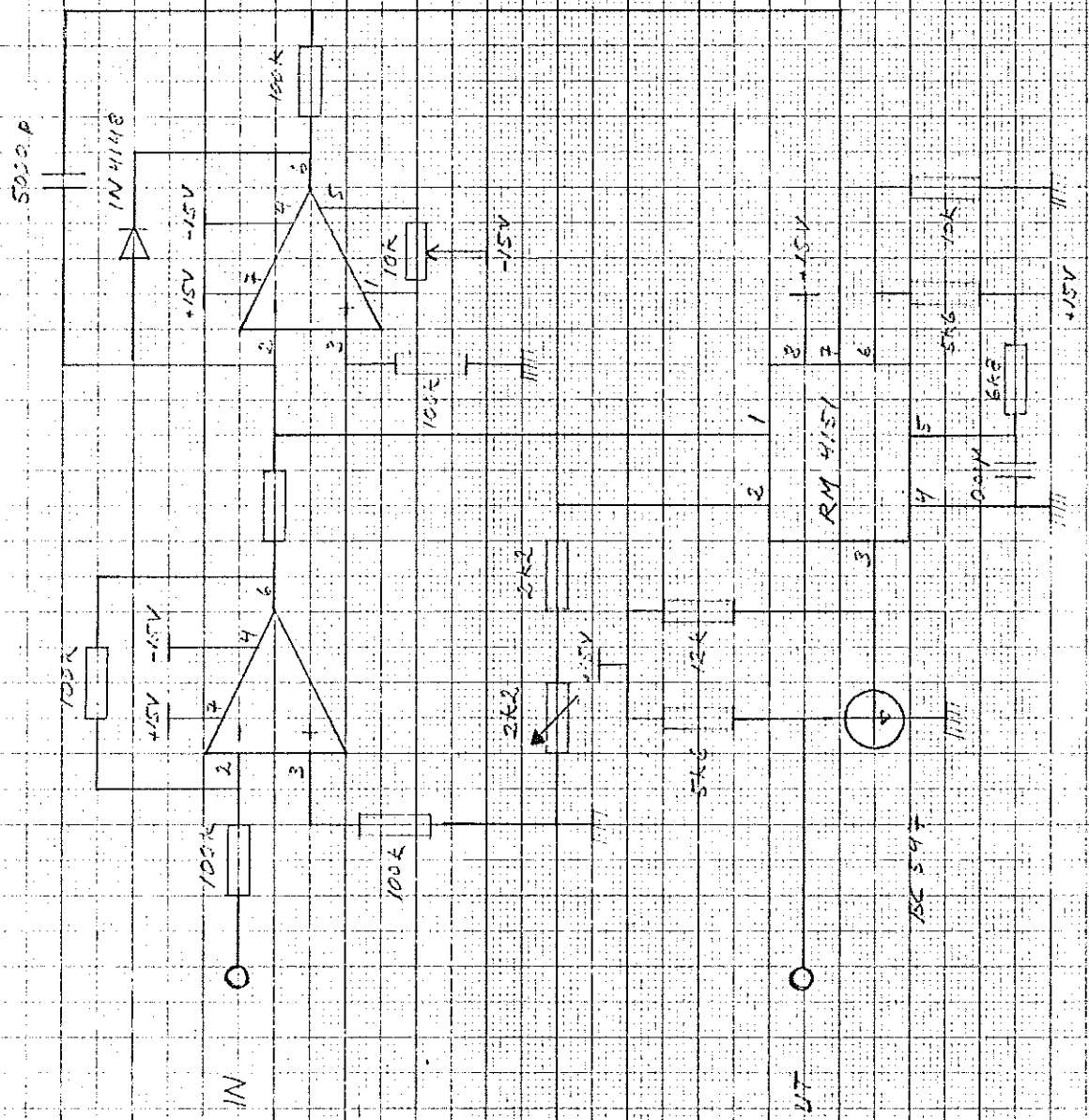
76 85-10
 8/10 11/10

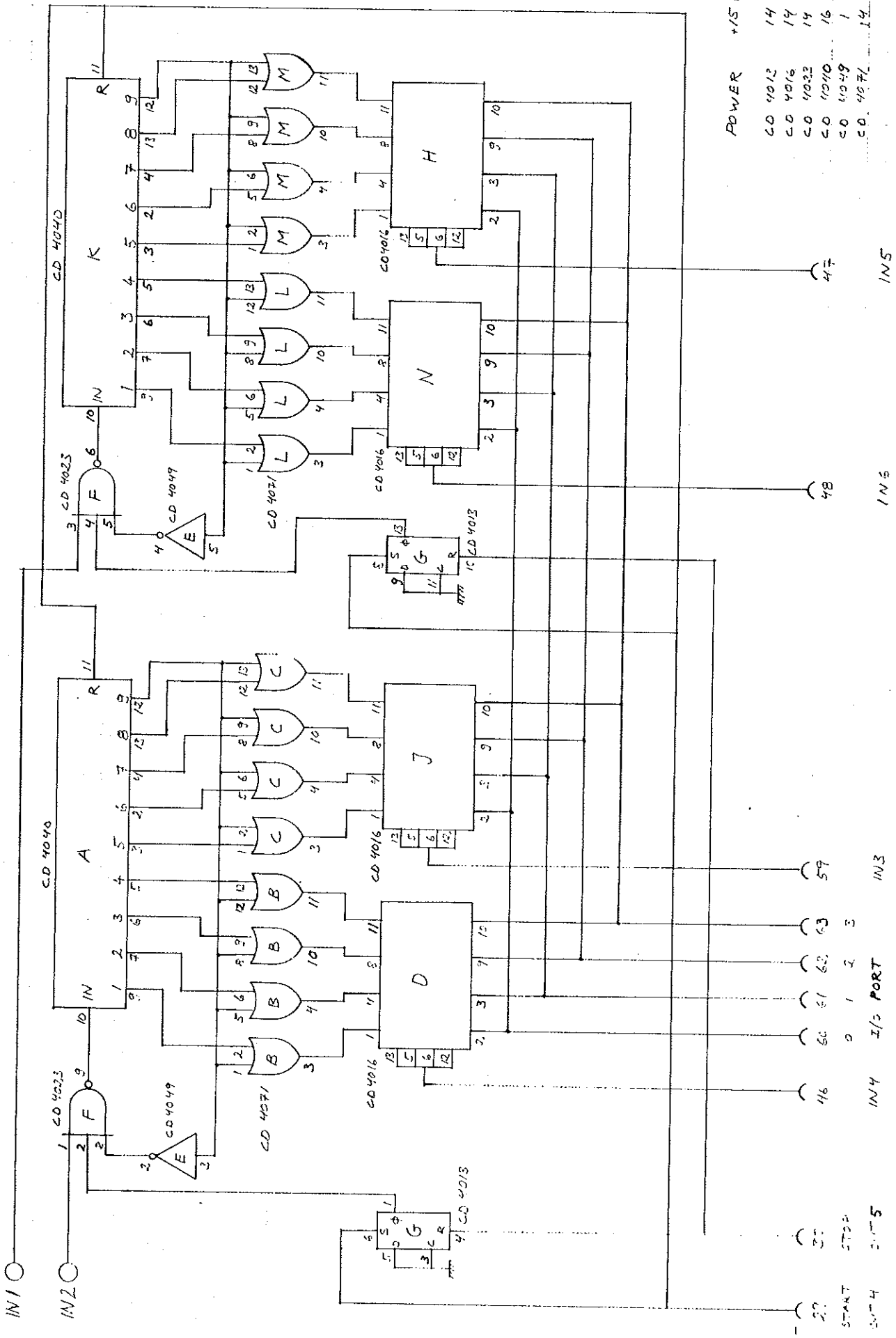
-15V	33	1	OPR 3
JACK SWA 110	34	2	OPR 2
-15V	35	3	OPR 1
	36	4	OPR 0
	37	5	OPR 3
	38	6	OPR 1
	39	7	OPR 1
	40	8	OPR 0
	41	9	40
	42	10	41
	43	11	42
	44	12	43
	45	13	44
IN 4	46	14	45
IN 5	47	15	46
IN 6	48	16	47
IN 7	49	17	48
IN 8	50	18	49
IN 9	51	19	50
IN 10	52	20	51
IN 11	53	21	52
TEST	54	22	17.2 kV ₂
Φ 1	55	23	300 M ₂
Φ 10	56	24	RES 07
IN 1	57	25	INT
IN 2	58	26	OUT 0
IN 3	59	27	OUT 1
I/O 0	60	28	OUT 2
I/O 1	61	29	OUT 3
I/O 2	62	30	OUT 4
I/O 3	63	31	OUT 5
JORD (ORIGINAL)	64	32	OUT 6

1.11
S/F - 01V

S/F - 01V

3
Rohr Braun
76-02-33



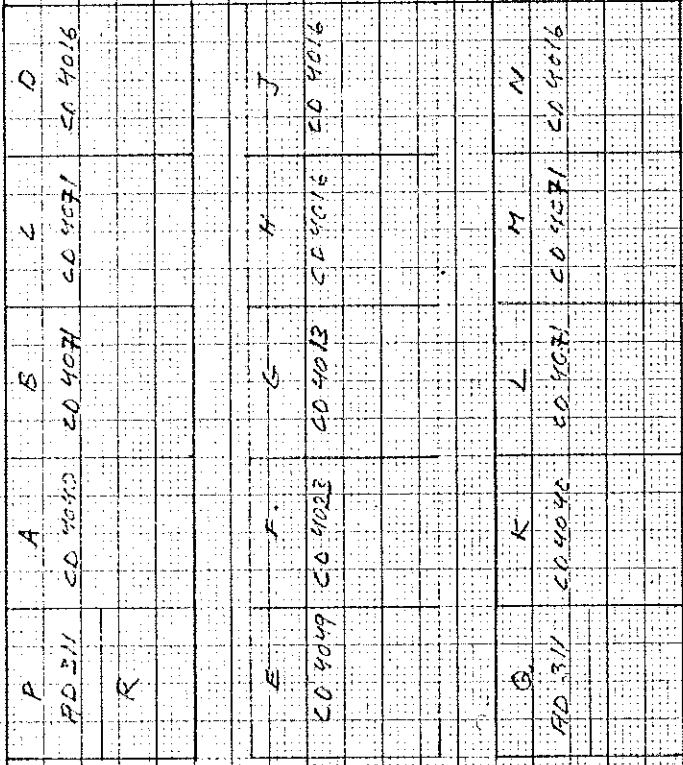


POWER	+5V	0V
CO 4012	14	7
CO 4016	14	7
CO 4023	14	7
CO 4040	16	8
CO 4049	1	8
CO 4071	14	7

LTH
REGUL
TEKNIK

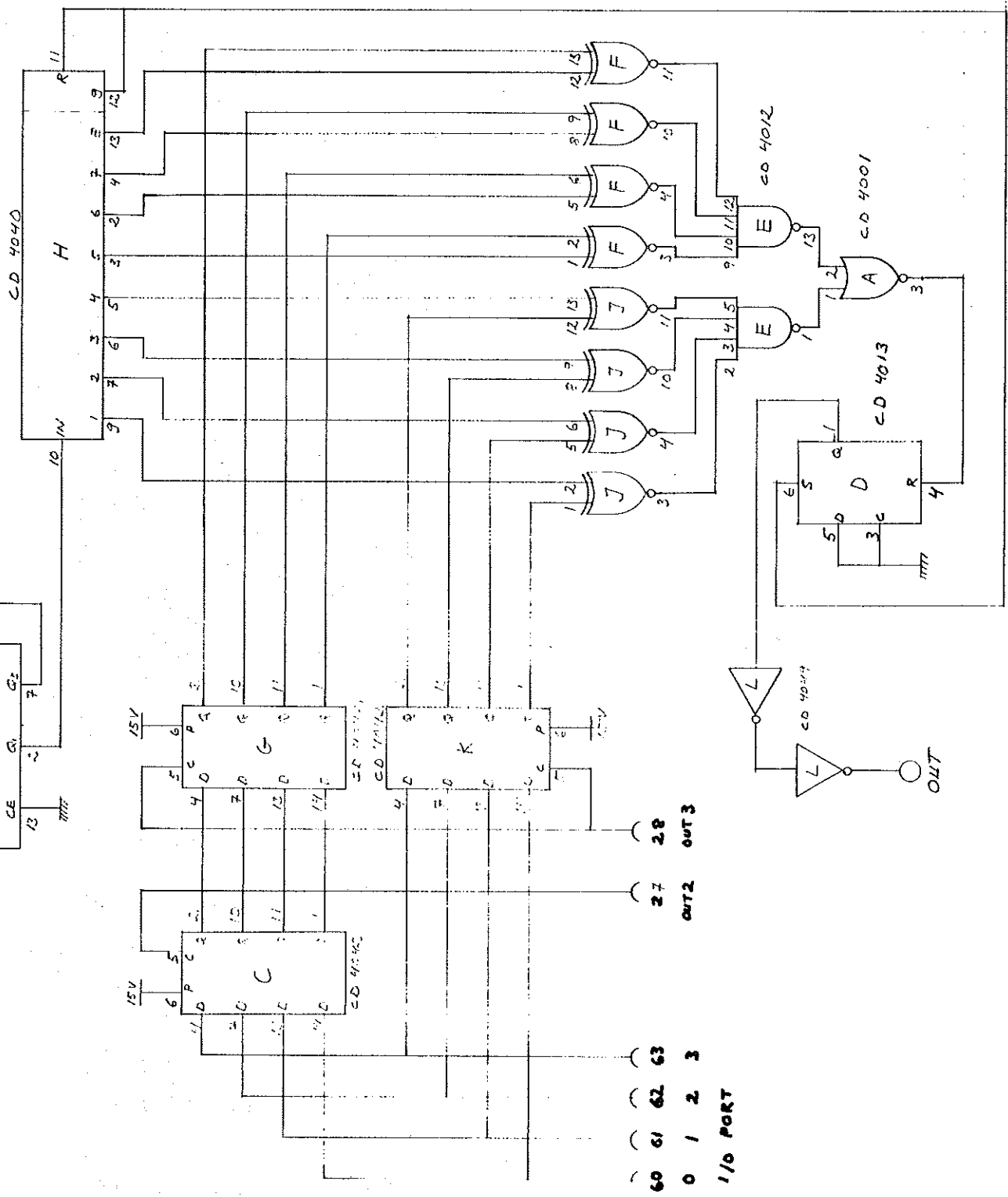
PULSRÄKNARE
KRETSLAYOUT

5
760512
Anders Nilsson



POWER 7.5V

CD 4001	14
CD 4012	14
CD 4013	14
CD 4017	16
CD 4042	16
CD 4047	1
CD 4074	14



CD 4077

60 61 62 63
0 1 2 3
I/O PORT

27 28
OUT2 OUT3

OUT

LTH
REGLER-
TEKNIK

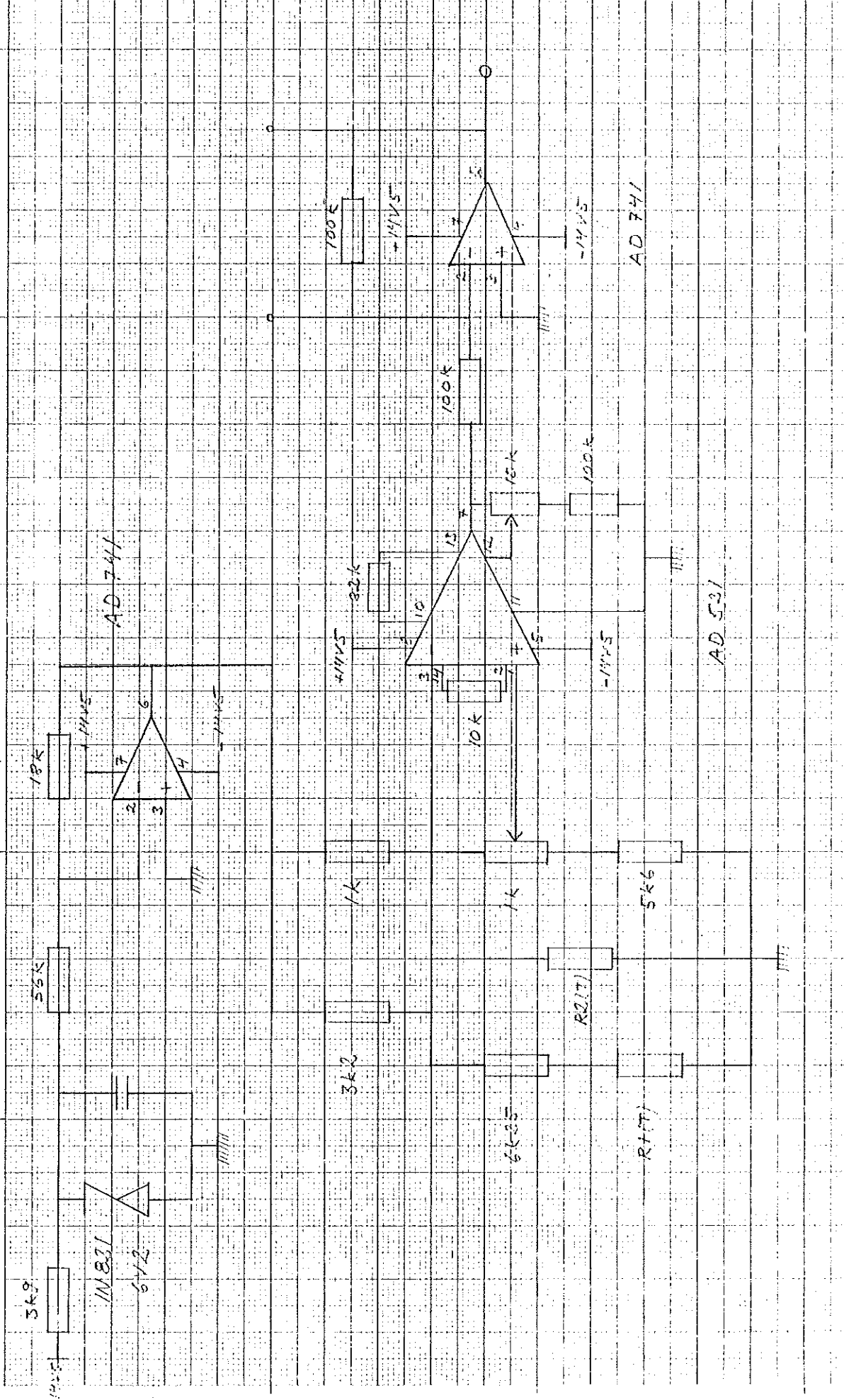
PULSLÅNGDSBIVÄR
KREISLAYOUT

7
760512
Anders Nilsson

9	8	7
CD 4011	CD 4017	CD 4012

D	E	F	G
CD 4013	CD 4012	CD 4017	CD 4012

L	H	J	K
CD 4019	CD 4010	CD 4017	CD 4012

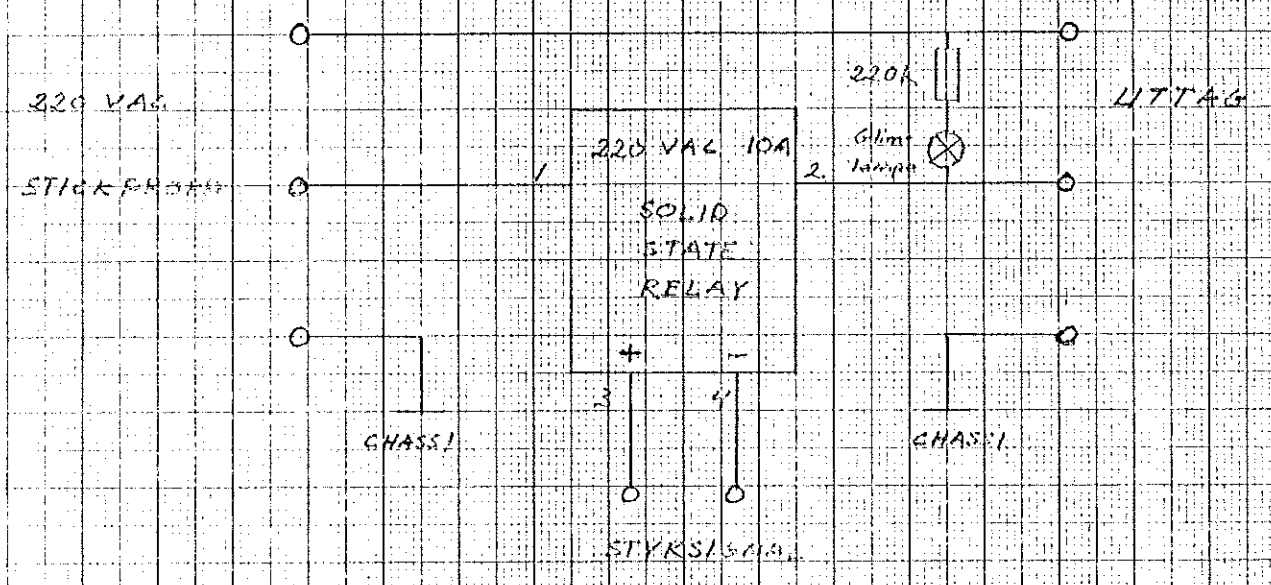


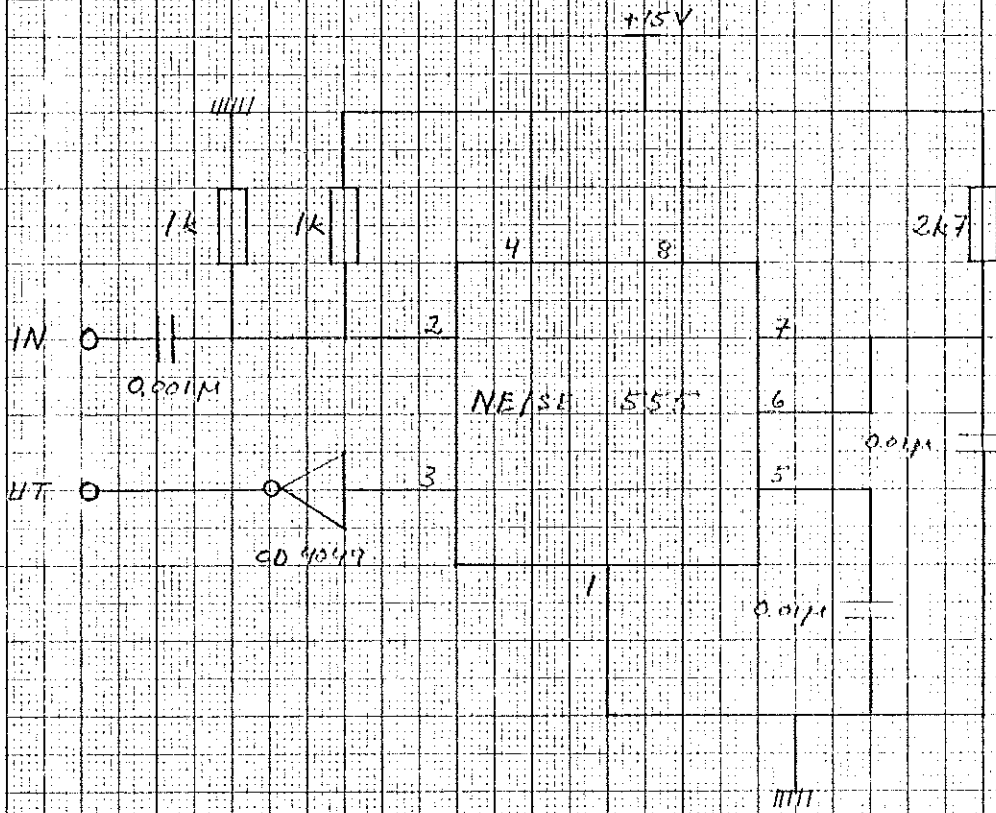
LTH
REGLER
TERMIN

EFFEKTREGULATOR

9

76-05-13
Anders Nilsson





4040 AND 4004
BASIC INSTRUCTION SET

(Those instructions preceded by an asterisk (*) are 2 word instructions that occupy 2 successive locations in ROM)
MACHINE INSTRUCTIONS (Logic 1 = Low Voltage; Negative Voltage; Logic 0 = High Voltage = Ground)

MNEMONIC	OPR D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	OPA D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	DESCRIPTION OF OPERATION
NOP	0 0 0 0	0 0 0 0	No operation
*JCN	0 0 0 1 A ₂ A ₁ A ₀ A ₂ A ₁ A ₀	C ₁ C ₀ C ₄ A ₁ A ₀ A ₁ A ₀	Jump to ROM address A ₂ A ₁ A ₀ if C ₁ A ₁ A ₀ is 101. If C ₀ A ₁ A ₀ is 101, then the same ROM address is used. If C ₁ A ₁ A ₀ is 101, then the same ROM address is used. If C ₀ A ₁ A ₀ is 101, then the same ROM address is used. If C ₁ A ₁ A ₀ is 101, then the same ROM address is used.
*JCN	0 0 1 0 D ₂ D ₁ D ₀ D ₂ D ₁ D ₀	R R R 0 D ₁ D ₀ D ₁ D ₀	Fetch immediate (direct) from ROM OPR D ₂ D ₁ D ₀ to index register per location RRR (2)
SRC	0 0 1 0	R R R 1	Send register control. Send the address (contents of index register per RRR) to ROM and RAM at X ₂ and X ₃ time in the Instruction Cycle
FIN	0 0 1 1	R R R 0	Fetch address from ROM. Send contents of index register per location RRR out as an address. Data fetched is placed into register per location RRR at A ₁ and A ₀ time in the instruction cycle
JR	0 0 1 1	R R R 1	Jump unconditional to ROM address A ₂ A ₁ A ₀
*JUN	0 1 0 0 A ₂ A ₁ A ₀ A ₂ A ₁ A ₀	D ₂ D ₁ D ₀ A ₁ A ₀ A ₁ A ₀	Jump to successive ROM address A ₂ A ₁ A ₀ if A ₁ A ₀ is 101. (A ₂ is in effect.)
*JMS	0 1 0 1 A ₂ A ₁ A ₀ A ₂ A ₁ A ₀	D ₂ D ₁ D ₀ A ₁ A ₀ A ₁ A ₀	Increment contents of register RRR (3)
INC	0 1 1 0	R R R 0	Increment contents of register RRR (3)
*SZ	0 1 1 1 A ₂ A ₁ A ₀ A ₂ A ₁ A ₀	R R R 0 A ₁ A ₀ A ₁ A ₀	Increment contents of register RRR (3) to ROM address A ₂ A ₁ A ₀ if within the same ROM (the contents of the instruction) if RRR is 0. Otherwise skip to the next instruction in memory.
ADD	1 0 0 0	R R R 0	Add contents of register RRR to accumulator with carry
SUB	1 0 0 1	R R R 1	Subtract contents of register RRR to accumulator with borrow
LD	1 0 1 0	R R R 0	Load contents of register RRR to accumulator
XCH	1 0 1 1	R R R 1	Exchange contents of index register RRR and accumulator
SHL	1 1 0 0	D D D 0	Shift left (down 1 level in stack) and load data 0000 to accumulator
LDM	1 1 0 1	D D D 1	Load data 0000 to accumulator

NEW 4040 INSTRUCTIONS

MNEMONIC	OPR D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	OPA D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	DESCRIPTION OF OPERATION
HLT	0 0 0 0	0 0 0 1	Halt - inhibit program counter and data buffers
BBS	0 0 0 0	0 0 1 0	Branch back from interrupt and restore the previous SRC. The Program Counter and index register control are restored to their pre-interrupt value
LCR	0 0 0 0	0 0 1 1	The contents of the COMMAND REGISTER are transferred to the ACCUMULATOR
ORA	0 0 0 0	0 1 0 0	The 4 bit contents of register RRR are logically "OR-ed" with the ACCUM.
ORS	0 0 0 0	0 1 0 1	The 4 bit contents of index register RRR are logically "OR-ed" with the ACCUMULATOR
ANI	0 0 0 0	0 1 1 0	The 4 bit contents of index register RRR are logically "AND-ed" with the ACCUMULATOR
ANI	0 0 0 0	0 1 1 1	The 4 bit contents of index register RRR are logically "AND-ed" with the ACCUMULATOR
DB0	0 0 0 0	1 0 0 0	DESIGNATE ROM BANK 0. (DB-ROM ₀ becomes enabled)
DB1	0 0 0 0	1 0 0 1	DESIGNATE ROM BANK 1. (DB-ROM ₁ becomes enabled)
SRI	0 0 0 0	1 0 1 0	SELECT INDEX REGISTER BANK 0. The index register is 7.
SRI	0 0 0 0	1 0 1 1	SELECT INDEX REGISTER BANK 1. The index register is 7.
ENI	0 0 0 0	1 1 0 0	ENABLE INTERRUPT
DISI	0 0 0 0	1 1 0 1	DISABLE INTERRUPT
RPM	0 0 0 0	1 1 1 0	READ PROGRAM MEMORY

INPUT/OUTPUT AND RAM INSTRUCTIONS

(The status and input/output instructions have been previously selected by the last SRC instruction executed.)

MNEMONIC	OPR D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	OPA D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	DESCRIPTION OF OPERATION
WRM	1 1 1 0	0 0 0 0	Write the contents of the accumulator into the previously selected RAM main memory character
WRM	1 1 1 0	0 0 0 1	Write the contents of the accumulator into the previously selected RAM status character
WRM	1 1 1 0	0 0 1 0	Write the contents of the accumulator into the previously selected ROM status character
WPM	1 1 1 0	0 0 1 1	Write the contents of the accumulator into the previously selected RAM status character
WRM (4)	1 1 1 0	0 1 0 0	Write the contents of the accumulator into the previously selected RAM status character 0
WRM (4)	1 1 1 0	0 1 0 1	Write the contents of the accumulator into the previously selected RAM status character 1
WRM (4)	1 1 1 0	0 1 1 0	Write the contents of the accumulator into the previously selected RAM status character 2
WRM (4)	1 1 1 0	0 1 1 1	Write the contents of the accumulator into the previously selected RAM status character 3
SRM	1 1 1 0	1 0 0 0	Read the previously selected RAM main memory character from the accumulator
SRM	1 1 1 0	1 0 0 1	Read the previously selected RAM status character from the accumulator
RDM	1 1 1 0	1 0 1 0	Read the previously selected ROM main memory character into the accumulator (I/O Lines)
ADM	1 1 1 0	1 0 1 1	Add the previously selected RAM main memory character to accumulator with carry
RDM (4)	1 1 1 0	1 1 0 0	Read the previously selected RAM status character 0 into accumulator
RDM (4)	1 1 1 0	1 1 0 1	Read the previously selected RAM status character 1 into accumulator
RDM (4)	1 1 1 0	1 1 1 0	Read the previously selected RAM status character 2 into accumulator
RDM (4)	1 1 1 0	1 1 1 1	Read the previously selected RAM status character 3 into accumulator

ACCUMULATOR GROUP INSTRUCTIONS

MNEMONIC	OPR D ₅ D ₄ D ₃ D ₂ D ₁ D ₀	DESCRIPTION OF OPERATION
CLR	1 1 1 1	Clear both (Accumulator and carry)
CLC	1 1 1 1	Clear carry
IAC	1 1 1 1	Increment accumulator
CMC	1 1 1 1	Complement carry
DMA	1 1 1 1	Rotate left (Accumulator and carry)
RAR	1 1 1 1	Rotate right (Accumulator and carry)
TCC	1 1 1 1	Transfer carry to accumulator and clear carry
DAC	1 1 1 1	Decrement accumulator
TCS	1 1 1 1	Transfer carry subtract and clear carry
STC	1 1 1 1	Set carry
DAA	1 1 1 1	Decimate adjust accumulator
IGP	1 1 1 1	Increment program counter by one out of four code to a bank code
DCL	1 1 1 1	Deplete command line

NOTES:

- (1) The condition code is assigned as follows:
C₁ = 1 - Zero condition
C₁ = 0 - Non-zero condition
C₂ = 1 - Jump if accumulator is zero
C₂ = 0 - Jump if carry/link is a 1
C₃ = 1 - Jump if carry/link is a 0
C₃ = 0 - Jump if register pair is 0
- (2) RRR is the address of 1 of 8 index register pairs in the CPU.
- (3) RRRR is the address of 1 of 16 index registers in the CPU.
- (4) Each RAM chip has 4 registers, each with twenty 4-bit characters subdivided into 16 main memory characters and 4 status characters. Chip number, RAM register and main memory character are addressed by an SRC instruction. For the selected chip and register, however, status character locations are selected by the instruction code (OPA).

PAGE 1 KVOTR SRC KVOTR

```
1 00000 R 000100 A *G          KVOTR
2 00000 R 000005 R *G
3 00001 R 000000 A *G
4 00002 R 000000 A
5
6
7 00003 R 000321 A *G
8 00004 R 000002 A
9
10
11
12 00005 R 000040 A *G          REG
13 00006 R 000000 A *G
14 00007 R 000330 A *G
15 00010 R 000346 A
16 00011 R 000360 A
17 00012 R 000347 A
18 00013 R 000042 A *G          LOOP
19 00014 R 000120 A *G
20 00015 R 000044 A *G          FIM R4,140 /R4=6
    00016 R 000140 A *G          FIM R6,100 /R6=4
    00017 R 000046 A *G
    00020 R 000100 A *G          .EJECT
```

.TITLE KVOTR
.GLOBAL MULTH, DIVIH, DUB
JUN BEG

NOPI
/INTERRUPT ROUTINE

LDM 1 /AC=1

BBS

/MAIN PROGRAM

FIM R0,0

LDM 10 /LOAD U=10000000 INTO STATUS CHARACTER 2 AND 3 IN THE RAM

WR2

CLR

WR3

FIM R2,120 /R2=5

FIM R4,140 /R4=6

FIM R6,100 /R6=4

.EJECT

21	00021 R	000047 A	*G
22	00022 R	000342 A	
23	00023 R	000120 A	*G
	00024 R	000137 R	*G
24	00025 R	000043 A	*G
25	00026 R	000342 A	
26	00027 R	000352 A	
27	00030 R	000041 A	*G
28	00031 R	000344 A	
29	00032 R	000045 A	*G
30	00033 R	000352 A	
31	00034 R	000041 A	*G
32	00035 R	000345 A	
33	00036 R	000041 A	*G
34	00037 R	000354 A	
35	00040 R	000260 A	*G
36	00041 R	000355 A	
37	00042 R	000261 A	*G
38	00043 R	000356 A	
39	00044 R	000262 A	*G
40	00045 R	000357 A	
41	00046 R	000263 A	*G
42	00047 R	000120 A	*G
43	00050 R	000156 E	*G

SRC R6 /LET THE PULSE COUNTER COUNT FOR 100 MS

WRR
JMS ETS

SRC R2

WRR
RDR
SRC R0 /READ THE FOUR MOST SIGNIFICANT BITS OF YR ON INPUT IN 1

WR0
SRC R4 /WRITE THEM INTO STATUS CHARACTER 0

RDR
SRC R0 /READ THE FOUR LEAST SIGNIFICANT BITS OF YR

WR1
SRC R0 /WRITE THEM INTO STATUS CHARACTER 1

R00
XCH R0

RD1
XCH R1

RD2
XCH R2 /<R2,R3>=U

RD3
XCH R3

JMS MULTH /<R4,R5,R6,R7>=YR*U

.EJECT

44	00051 R	000042 A	*G
	00052 R	000060 A	*G
45	00053 R	000054 A	*G
	00054 R	000100 A	*G
46	00055 R	000056 A	*G
	00056 R	000120 A	*G
47	00057 R	000055 A	*G
48	00060 R	000342 A	
49			
	00061 R	000120 A	*G
	00062 R	000137 R	*G
50			
	00063 R	000057 A	*G
51	00064 R	000342 A	
52			
	00065 R	000043 A	*G
53	00066 R	000352 A	
54			
	00067 R	000260 A	*G
55			
	00070 R	000055 A	*G
56	00071 R	000352 A	
57			
	00072 R	000261 A	*G
58			
	00073 R	000120 A	*G
	00074 R	000154 E	*G
59			
	00075 R	000246 A	*G
60			
	00076 R	000034 A	*G
	00077 R	000105 R	*G
61			

FIN R2,060 /R2=3

FIN R12,100 /R12=4

FIN R14,120 /R14=5

SRC R12 /LFT THE PULSE COUNTER COUNT FOR 100 MS

MRR
JMS ETS

SRC R14

MRR
SRC R2 /READ Y ON INPUT IN 2 AND PUT IT INTO <R0,R1>

RDR
XCH R0

SRC R12

RDR
XCH R1

JMS DIVIH /<R6,R7>=U=YR*U:Y

LD R5

JFZ R /IF U=0 THEN R7= 1

.EJECT

PAGE	4	KVOTR	SRC	KVOTR
62		00100	R 000247 A *G	
63		00101	R 000034 A *G	
		00102	R 000105 R *G	
		00103	R 000362 A	
64		00104	R 000267 A *G	
65		00105	R	R
66		00105	R 000120 A *G	
		00106	R 000155 E *G	
67		00107	R 000040 A *G	
		00110	R 000000 A *G	
68		00111	R 000042 A *G	
		00112	R 000040 A *G	
69		00113	R 000044 A *G	
		00114	R 000060 A *G	
70		00115	R 000041 A *G	
71		00116	R 000247 A *G	
72		00117	R 000347 A	
73		00120	R 000246 A *G	
74		00121	R 000346 A	
75		00122	R 000043 A *G	
76		00123	R 000342 A	
77		00124	R 000247 A *G	
78		00125	R 000045 A *G	
79		00126	R 000342 A	
80				

```

LD R7
JFZ B
IAC
XCH R7
JMS DUR
FIW R0,0
FIW R2,040 /R2=2
FIW R4,060 /R4=3
SRC R0 /LOAD U INTO STATUS CHARACTERS 2 AND 3
LD R7
WR3
LD R6
WR2
SRC R2 /SEND U TO THE OUTPUT MONUL
WRP
LD R7
SRC R4
WRP
EJECT

```

```

81 00127 R 000014 A /WAIT FOR INTERRUPT
82 00130 R 000360 A /AC=0
83 00131 R 000000 A NOP
84 00132 R 000024 A JTZ D /IF AC=0 THEN GO TO D ELSE GO TO NEXT INSTRUCTION
85 00133 R 000131 R A *G
86 00134 R 000015 A
87 00135 R 000100 A A *G
88 00136 R 000013 R A *G
89
90 00137 R 000050 A A *G ETS / DELAY SUBROUTINE
91 00140 R 000000 A A *G FIM R10,0
92 00141 R 000052 A A *G
93 00142 R 000000 A A *G
94 00143 R 000170 A A *G ISZ R8,C
95 00144 R 000143 R A *G
96 00145 R 000171 A A *G ISZ R9,C
97 00146 R 000143 R A *G
98 00147 R 000172 A A *G ISZ R10,C
99 00150 R 000143 R A *G
00151 R 000000 A A NOP
00152 R 000000 A A NOP
00153 R 000300 A A *G PRL 0
00154 R 000154 A A *G .END
00155 R 000155 E E *F
00156 R 000156 E E *F

```

SIZE=00157

NO ERROR LINES

1	00000	R	000044	A	*G
2	00000	R	000000	A	*G
3	00001	R	000000	A	*G
4	00002	R	000046	A	*G
	00003	R	000000	A	*G
5	00004	R	000241	A	*G
	00005	R	000272	A	*G
	00006	R	000243	A	*G
	00007	R	000273	A	*G
6	00010	R	000120	A	*G
7	00011	R	000103	R	*G
8					
9	00012	R	000251	A	*G
10	00013	R	000267	A	*G
11	00014	R	000250	A	*G
12	00015	R	000266	A	*G
13					

MULTH
 .TITLE MULTH
 .GLOBAL MULTH
 FIM R4,0

FIM R6,0

CHANGE R1,R3,R10,R11 /R1 OCH R3 LAGGS I R10 OCH R11

/OCH MULT4 ANROPAS SOM MULTIPLICERAR
 /R10 OCH R11 OCH LAGGER RESULTATET I
 /R5 OCH R9.

LD R9
 XCH R7 /LAGG IN PRODUKTEN I RATT REGISTER.

LD R6
 XCH R6

.EJECT

14

00016 R 000241 A *G

*G

00017 R 000272 A *G

*G

00020 R 000242 A *G

*G

00021 R 000273 A *G

*G

00022 R 000120 A *G

*G

00023 R 000103 R *G

*G

15

00024 R 000250 A *G

*G

17

00025 R 000265 A *G

*G

18

00026 R 000246 A *G

*G

19

00027 R 000361 A

*G

20

00030 R 000211 A *G

*G

21

00031 R 000266 A *G

*G

22

00032 R 000032 A *G

*G

23

00033 R 000035 R *G

*G

24

00034 R 000145 A *G

*G

CHANGE R1,R2,R10,R11 /B*C ADDERAS TILL R5 OCH R6

/OCH MULT4 ANROPAS.

LD R6

XCH R5

LD R6

CLC R7

ADD R6

XCH R6

JFC NOCY1

INC R5 /INGEN CARRY TILL R4 MOJLIG.STORSTA

.EJECT

PAGE 3 MULTH SRC MULTH

25 00035 R NOCY1 CHANGE R0,R3,R10,R11 /MULLISA T&L R5 OCH R6 AR 1110 1111.
26 00035 R /A*D ADDERAS TILL R5 OCH R6.

00035 R 000240 A *G

00036 R 000272 A *G

00037 R 000243 A *G

00040 R 000273 A *G

00041 R 000120 A *G

00042 R 000103 R *G

00043 R 000246 A *G

00044 R 000361 A

00045 R 000211 A *G

00046 R 000266 A *G

00047 R 000032 A *G

00050 R 000053 R *G

00051 R 000361 A

00052 R 000145 A *G

LD R6

CLC R9

ADD R9

XCH R6

JFC NOCY2

CLC R5 /INGEN RISK ATT R5 BLIP 0.
INC R5

.EJECT

27

28

29

30

31

32

33

34

35

00053 R 000245 A *G

NOCY2

LD

R5

36

00054 R 000210 A *G

ADD

R2

37

00055 R 000265 A *G

XCH

R5

38

00056 R 000032 A *G
00057 R 000061 R *G

JFC

NOCY3

39

00060 R 000144 A *G
00061 R

INC

R4

40

00061 R 000240 A *G
00062 R 000272 A *G
00063 R 000242 A *G
00064 R 000273 A *G

NOCY3

CHANGE

R0,R2,R10,F11

/A*C ADDERAS TILL R4 OCH R5.

41
42

00065 R 000120 A *G
00066 R 000163 R *G
00067 R 000361 A

CLC

R5

43

00070 R 000250 A *G

LD

R5

44

00071 R 000204 A *G

ADD

R4

45

00072 R 000264 A *G

XCH

R4

46

00073 R 000245 A *G

LD

R5

47

00074 R 000361 A

CLC

R9

48

00075 R 000211 A *G

ADD

R5

49

00076 R 000265 A *G

XCH

R5

50

00077 R 000032 A *G

JFC

OUT

51

00101 R 000144 A *G

INC

R4

52

00102 R
00102 R 000300 A *G

OUT

0

PAGE	S	MULTH	SRC	MULTH	MULTH	FIN	R8,0	/MULT4 MULTIPPLICERAR 2 4-BITARS TAL
53		00103	R	000050	A *G			
		00103	R	000000	A *G			
		00104	R	000000	A *G			
54						FIN	R12,0	/SOM FINNS I R10 OCH R11.
55		00105	R	000054	A *G			/RESULTATET LAGRAS I R8 OCH R9.
56		00106	R	000000	A *G			
57		00107	R	000335	A *G	LDW	-3	/RAKNAREN F12 SATTS = -3.
58						XCH	R12	
59		00110	R	000274	A *G	JUN		NOSHIPT /INGA VANSTERSKIPT FORSTA GANGEN.
60		00111	R	000100	A *G			
61		00112	R	000125	R *G			
62		00113	R	000255	A *G	SHIFT	R13	/I R13 LAGRAS DE 4 MEST SIGNIFIKANTA
63						LD		/BITARNA SOM UPPKOMMER GENOM VANSTER-
64		00114	R	000361	A	CLC		/SKIPTEN,AR O FRAN BORJAN.
65		00115	R	000365	A	PAL		/INGEN CARRY VID VANSTERSKIPT.
66						XCH	R13	
67		00116	R	000275	A *G	LD	R10	
68		00117	R	000252	A *G			
69		00120	R	000365	A	PAL	R10	
70		00121	R	000272	A *G	XCH		
71		00122	R	000032	A *G	JFC		NOSHIPT /TA VARA PA CARRYN
72		00123	R	000125	R *G	INC	R13	
		00124	R	000155	A *G			.EJECT

PAGE	6	MULTI	SRC	MULTI	NOSHIFT	LD	R11	
73		00125	R					/HOGERSKIFTA TAL 2.
74		00125	R	000253	A *G			/OM VALD TECKENBIT AR 0
75								/INGEN ADDITION.
76		00126	R	000366	A			
77						RAR	R11	
						XCH		
78		00127	R	000273	A *G			/OM CARRYN AR 0 INGEN ADDITION.
						JFC	NOADD	
79		00130	R	000032	A *G			
		00131	R	000145	R *G			
		00132	R	000361	A			
80						CLC	R3	/ADDERA TILL DE 4 WEST
		00133	R	000250	A *G	LD		
81								/SIGNIFIKANTA RITARNA.
82						ADD	R13	
83		00134	R	000215	A *G	XCH	R8	
		00135	R	000270	A *G	LD	R9	/ADDERA TILL DE 4 MINST
84								/SIGNIFIKANTA RITARNA.
85		00137	R	000361	A	CLC	R10	
86						ADD		
87		00140	R	000212	A *G	XCH	R9	
88						JFC	NOADD	/LAGG TILL CARRYN.
89		00141	R	000271	A *G			
		00142	R	000032	A *G			
		00143	R	000145	R *G	ING	P6	
90								
91		00144	R	000150	A *G			

.EJECT

PAGE	7	MULTM	SRC	MULTM	NOAND	LD	P12	/FARDIGT ?
92		00145	R					
		00145	R	000254	A *G			
93		00146	R	000024	A *G	JTZ	OUTM	
		00147	R	000153	R *G			
94		00150	R	000154	A *G	INC	R12	
95		00151	R	000100	A *G	JUN	SHIFT	
		00152	R	000113	R *G			
96		00153	P			OUTM	BBL	0
		00153	R	000300	A *G			
				000000	A			
97				SIZE=00154	A			

NO ERROR LINES
 .END

PAGE	1	DIVIH	SRC	DIVIH		DIVIH	TITLE	DIVIH
1							.TITLE	DIVIH
2							.GLOBL	DIVIH,ROTH
3							FIN	R2,0
4								
5							CLC	
6							LD	R0
7							PAR	
8							XCH	R0
9							LD	R1
10							PAR	
11							XCH	R1
12							LD	R2
13							PAR	
14							XCH	R2
15							CLC	
16								
17								
18								
19							LD	R4
20							PAR	
21							XCH	R4
22							LD	R5

/ HOGERSKIFTA R0,R1 OCH R2 ETT STEG

/ HOGERSKIFTA R4-R7 ETT STEG
/ SISTA BITEN TILL R9

.EJECT

23 00021 R 000366 A
 24 00022 R 000265 A *G
 25 00023 R 000246 A *G
 26 00024 R 000366 A
 27 00025 R 000266 A *G
 28 00026 R 000247 A *G
 29 00027 R 000366 A
 30 00030 R 000267 A *G
 31 00031 R 000320 A *G
 32 00032 R 000365 A
 33 00033 R 000271 A *G
 34 00034 R 000120 A *G
 00035 R 000113 R *G
 35 00036 R 000022 A *G
 00037 R 000071 R *G
 38 00040 R 000327 A *G
 00041 R 000270 A *G
 00042 R
 39 00042 R 000120 A *G
 00043 R 000130 E *G
 40 00044 R 000150 A *G
 41 00045 R 000250 A *G
 42 00046 R 000024 A *G
 00047 R 000077 R *G

RAP R5

XCH R6

LD R6

RAR R6

XCH R6

LD R7

PAR P7

XCH P7

LD 0

RAL R9

XCH R9

JMS SUPT

JTC MAX

LDW -11

XCH R8

JMS ROTH

INC R8

LD R8

JTZ OUT

ROTAND

/DIVISORN.
/OKTALT

/RR AR RAKNAREN

/HOPPA OM DIVIDENDEN AR > ELLER =

.EJECT

44 00050 R 000120 A *G JMS AD /OM RESULTATET AV ADDITIONEN AR

00051 R 000100 R *G

/POSITIVT ELLER 0 AR CARRYN=1.

45 00052 R 000032 A *G JFC ROTADD

00053 R 000042 R *G

/I R9 LAGGS RESULTETET IN.

46 00054 R 000151 A *G ROTSUB INC R9

00055 R 000120 A *G

/R6 OCH R7 SKIFTAS EFTER VARJE GANG.

47 00056 R 000130 E *G JMS ROTH

00057 R 000150 A *G

INC R8

48 00060 R 000250 A *G LD R8

00061 R 000024 A *G

JTZ OUT

49 00062 R 000077 R *G JMS SUPT /OM RESULTATET AV SUBTRAKTIONEN AR

00063 R 000120 A *G

JTC ROTSUB

50 00064 R 000113 R *G JUN ROTADD /POSITIVT ELLER 0 SA AR CARRYN=1.

00065 R 000022 A *G

JUN ROTADD

51 00066 R 000054 R *G JFC ROTADD /MAX STALLER UT MAXIMALT TAL,

00067 R 000100 A *G

LDM 17 /DVS TALET NARMAST UNDER 1.

52 00070 R 000042 R *G XCH R6

00071 R 000071 R *G

LDM 17

53 00072 R 000266 A *G

.EJECT

54 00073 R 000337 A *G

55

56

57

58

59

60

61

PAGE	4	DIVIH	SRC	DIVIH		
62		00074	R 000267	A *G	XCH	R7
63		00075	R 000100	A *G		
		00076	R 000077	R *G	JUN	OUT
64		00077	R			
		00077	R 000300	A *G	R9L	0
65		00100	R 000361	A	AD	
66		00101	R 000246	A *G	CLC	R6
67		00102	R 000202	A *G	LD	
					ADD	R2
68		00103	R 000266	A *G	XCH	R6
69		00104	R 000245	A *G	LD	R5
70					ADD	R1
71		00105	R 000201	A *G	XCH	R5
72		00106	R 000265	A *G	LD	R4
73		00107	R 000244	A *G	ADD	R0
74		00110	R 000200	A *G	XCH	R4
75		00111	R 000264	A *G	B8L	0
76		00112	R 000300	A *G	CLC	
77		00113	R 000361	A	LD	R6
78					SUB	R2
79		00114	R 000246	A *G		
80		00115	R 000222	A *G		
81					.EJECT	

/SUBROUTINEN AD ADDERAR TALET I R0

/OCH R1 TILL TALET I R4 OCH R5.

/SUBROUTINEN SUPT SUBTRAHERRAR TALET I
/R0 OCH R1 FRAN TALET I R4 OCH R5.

PAGE	5	DIVIH	SRC	DIVIH		
82					XCH	R6
		00116	R 000266	A *G	CMC	
83		00117	R 000363	A	LD	R5
84						
		00120	R 000245	A *G	SUR	R1
85						
		00121	R 000221	A *G	XCH	R5
86						
		00122	R 000265	A *G	CMC	
87		00123	R 000363	A	LD	R4
88						
		00124	R 000244	A *G	SUB	R0
89						
		00125	R 000220	A *G	XCH	R4
90						
		00126	R 000264	A *G	SRL	0
91						
		00127	R 000300	A *G	.END	
			000000	A		
92		00130	R 000130	E *F		
			SIZE=00131			

NO ERROR LINES

LINE	ROTH	SRC	ROTH	TITLE	ROTH
1				.TITLE	ROTH
2				.SLOHL	ROTH
3				CLC	
4				LD	R9
5				RAR	
6				XCH	R9
7				LD	R7
8				RAL	
9				XCH	R7
10				LD	R6
11				RAL	
12				XCH	R6
13				LD	R5
14				RAL	
15				XCH	R5
16				LD	R4
17				RAL	
18				XCH	R4
19				RBL	0
20					

SIZE=00021 NO ERROR LINES

00020 R 000300 A *G
000000 A

PAGE	1	DUB	SRC	DUB
1				
2				
3				
4				
5				
6				
7				
8				
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19				

```

.TITLE DUB
:GLOBL DUB
FIL R0,0
/THIS SUPERROUTINE LIMITS DU=U(T)-U(T-1)

FIL R6,040
/ <R6,R9>=A

SRC R0
RD2
XCH R2
/ <R2,R3>=U(T-1)

RD3
XCH R3

CLC
LD R7
/ <R4,R5>=DU

SUB R3
XCH R5

CMC
LD R6

SUB R2
XCH R4

JTC E
/IF DU<0 THEN <R4,R5>=-DU=IDU1

.EJECT

```

PAGE	2	DUB	SRC	DUB
20		00023	R 000360	A
21		00024	R 000225	A *G
22		00025	R 000265	A *G
23		00026	R 000363	A
24		00027	R 000320	A *G
25		00030	R 000224	A *G
26		00031	R 000264	A *G
27		00032	R 000140	A *G
28		00033	R 000361	A
29		00034	R 000245	A *G
30		00035	R 000231	A *G
31		00036	R 000265	A *G
32		00037	R 000363	A
33		00040	R 000244	A *G
34		00041	R 000230	A *G
35		00042	R 000264	A *G
36		00043	R 000032	A *G
37		00044	R 000071	R *G
38		00045	R 000240	A *G
39		00046	R 000024	A *G
40		00047	R 000062	R *G

E

CL3
 SUB R5
 XCH R5
 CMC
 LDM 0
 SUB R4
 XCH R4
 INC R0
 CLC
 LD R5
 SUB R9
 XCH R5
 CMC
 LD R4
 SUB R8
 XCH R4
 JFC F
 /
 LD R0
 JTZ G
 .EJECT

/R=<R4,R5>=IDU1-A

/IF B>0 THEN U=<R4,R7>=U(T-1) + A OR U=U(T-1) - A

DEPENDING ON THE SIGN OF DU

PAGE	3	DUB	SRC	DUF	
41		00050	R 000361	A	CLC
42					LD R3
43		00051	R 000243	A *G	SUB R9
44		00052	R 000231	A *G	XCH R7
45		00053	R 000267	A *G	CMC
46		00054	R 000363	A	LD R2
47		00055	R 000242	A *G	SUB R8
48		00056	R 000230	A *G	XCH R6
49		00057	R 000266	A *G	JUN F
50		00060	R 000100	A *G	CLC
51		00061	R 000071	R *G	LD R3
52		00062	R 000361	A	
53		00063	R 000243	A *G	ADD R9
54		00064	R 000211	A *G	XCH R7
55		00065	R 000267	A *G	LD R2
56		00066	R 000242	A *G	ADD R8
57		00067	R 000210	A *G	XCH R6
58		00070	R 000266	A *G	NOP
59		00071	R 000000	A	BRL 0
		00072	R 000300	A *G	
			000000	A	

SIZE=00073 NO ERROR LINES •END