

CODEN: LUTFD2/(TFRT-5214)/1-061/(1978)

MIKRODATORBASERAT POSITIONERINGSSYSTEM  
FÖR ASEA IRB-SYSTEM

PETER NOREN

Lund Institute of Technology  
Department of Automatic Control  
Oktober 1978

TILLHÖR REFERENSBIBLIOTEKET

UTLANAS EJ

Dokumentutgivare  
Lund Institute of Technology  
Handläggare Dept of Automatic Control  
Björn Wittenmark  
Författare

Dokumentnamn  
REPORT LUTFD2/(TFRT-5214)/1-061/(1978)  
Utgivningsdatum  
October 1978  
Dokumentbeteckning  
Arendebeteckning

Peter Noren

1014

Dokumenttitel och undertitel

1014  
Mikrodatorbaserat positioneringssystem för ASEA IRb-system  
(Mikrodatorbased positioningsystem for ASEA IRb-system)

Referat (sammmandrag)

1014  
In this report one way of doing mikrodatorbased positioning of a robot axis is described.  
Todays position-regulator has been transfered to software.  
A Resolver to Digitalconversion has been made and an interface to the Drive-unit constructed.

Referat skrivet av

1014  
Author

Förslag till ytterligare nyckelord

1014

Klassifikationssystem och -klass(er)

1014

Indextermer (anga källa)

1014

Omfång

1014  
61 pages

Övriga bibliografiska uppgifter

56T2

Språk

1014  
Swedish

Sekretessuppgifter

1014

ISSN

60T4

ISBN

60T6

Dokumentet kan erhållas från

1014  
Department of Automatic Control  
Lund Institute of Technology  
Box 725, S-220 07 LUND 7, Sweden

Mottagarens uppgifter

62T4

Pris

1014

DOKUMENTTABLAD enligt SIS 62 10 12

Blankett LU 11:25 1976-07

Författare - Author

Peter Norén

Godkännare - Approved by

Hans Skoog, YBR *XS*

Uppdragsgivare - Requested by

Hans Skoog, YBR

Titel - Title

Mikrodatorbaserat positions-  
system för ASEA IRb-system

Examensarbete för LTH

## Teknisk rapport Technical Report

Från - From

YBR

Datum - Date

1978-09-22

TR-YB0096

Reg.

6397

Sida - Page

1

Utredning, teoretisk  
undersökning - Analysis,  
theoretical investigation

Provnings, experim. under-  
sökning - Test, experi-  
mental investigation

Delrapport

Slutrapport

Provnings/undersökning avslutad  
Test/investigation finished

Ordernr - Ref. No.

Debiteras ordernr

Pkt/Akt

812.110

Antal textsidor - No. of pages of text

60

Antal bilagesidor - No. of supplem. pages

### Sammanfattning - Summary

I denna rapport beskrivs ett sätt att utföra en mikrodatorbaserad positionering av en robotaxel.

Nuvarande lägesregulator för axeln, har överförs till mjukvara. En resolver till digitalomvandling har gjorts, och ett drivdonsinterface har utarbetats. Lägesregleringen i det mikrodatorbaserade systemet, har jämförts med lägesregleringen i dagens system. Någon större skillnad har inte kunnat påvisas. För att undersöka samplingstidens inverkan på regleringen i det mikrodatorbaserade systemet, har denna ökat i steg om 10 ms. Därvid har funnits, att samplingstiden inte behöver vara så kort som 10 ms för att en tillfredställande reglering skall erhållas.

Skillnaden i pris mellan hårdvara och mjukvara, gör att man i ett system med tillgänglig datorkapacitet bör överväga att lägga så mycket hårdvarufunktioner som möjligt i program. Detta arbete är ett första steg i att överföra samtliga funktioner på dagens axelkort till mjukvara.

### Distribution

YB, YBA, YBR, YBS, YTU

Enbart sida 1 - Page 1 only

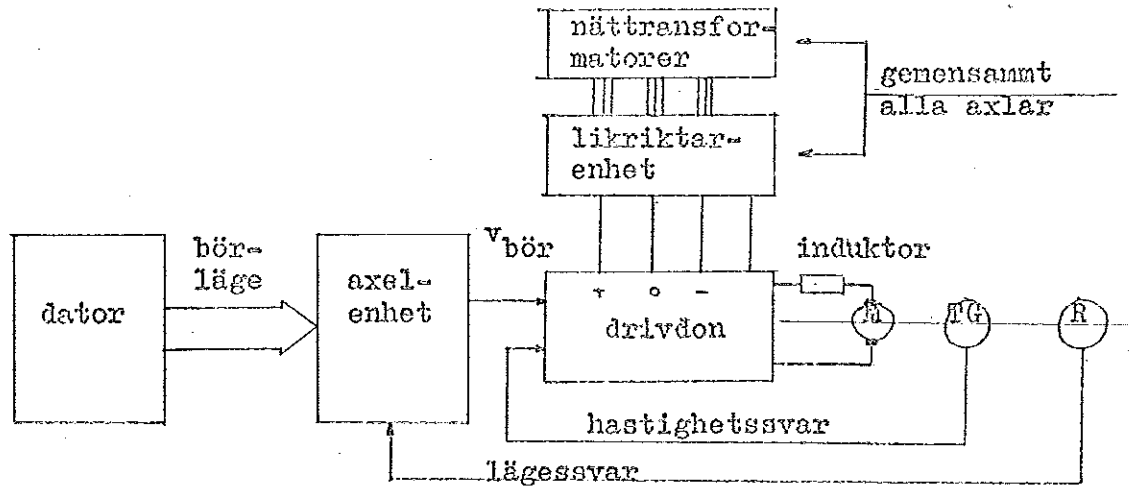
YL

Innehållsförteckning

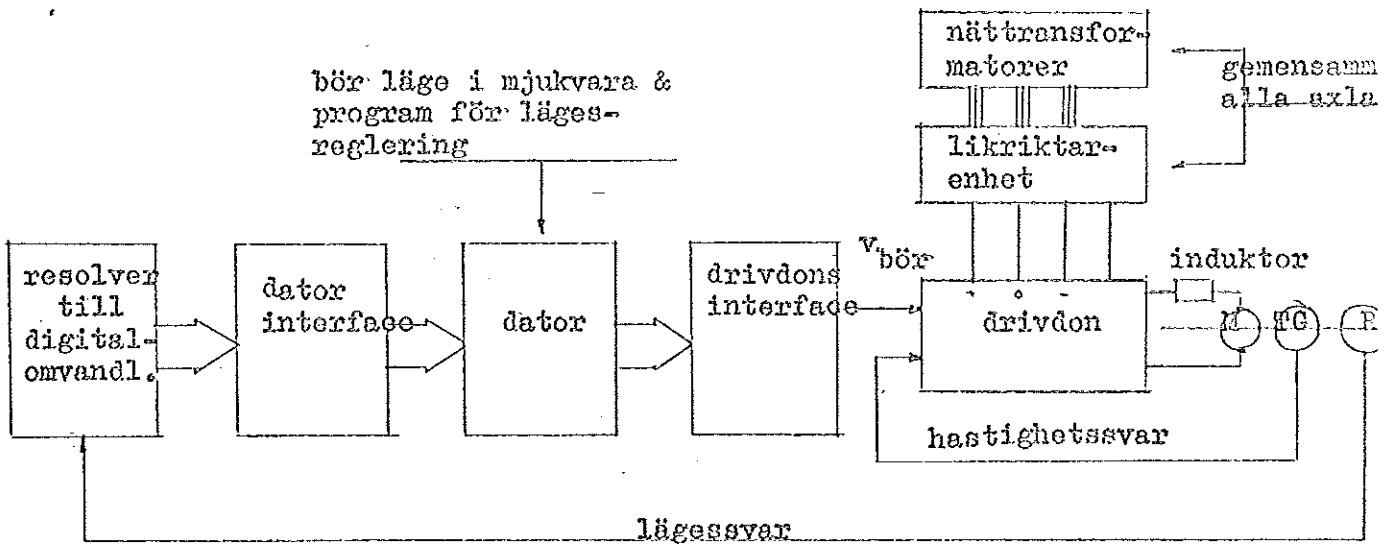
## FÖRORD

- 1 BESKRIVNING AV KOMPONENTER I DAGENS POSITIONERINGSSYSTEM
  - 1.1 Datorn
  - 1.2 Lägesmätning med resolver
  - 1.3 Axelkort
  
- 2 ALLMÄNT OM RESOLVER TILL DIGITAL-  
OMVANDLING
  - 2.1 Fasanaloga mätmetoden
  - 2.2 Omvända fasanaloga mätmetoden
  - 2.3 Amplitudanaloga mätmetoden
  
- 3 HÅRDVARA FÖR MIKRODATORBASERAD POSITIONERING AV 5 AXLAR
  - 3.1 Resolver till digitalomvandling
  - 3.2 Drivdons interface
  
- 4 KONSTRUKTIONSLÖSNINGAR FÖR MIKRO-  
DATORBASERAD POSITIONERING AV EN AXEL
  - 4.1 Resolver till digitalomvandling,  
dator interface
  - 4.2 Program för lägesreglering
  - 4.3 Drivdons interface
  
- 5 PROV
  - 5.1 Allmänt om proven
  - 5.2 Lägesreglering i hårdvara jämfört  
med lägesreglering i mjukvara
  - 5.3 Lägesreglering i mjukvara med  
ändring av samplingstiden

## FÖRORD



Figur 1 Blockschemat över positionering i dagens system, en axel.



Figur 2 Blockschemat över mikrodatorbaserad positionering, en axel.

Figur 1 visar i blockschemat form, principen för positionering i dagens system. Datorn är under förflyttningsfasen ovetande om robotaxelns verkliga läge.

Figur 2 får definiera vad jag kallar mikrodatorbaserad positionering. Datorn har under förflyttningsfasen fullständig vetskap om robotaxelns verkliga läge.

Mikrodatorbaserad positionering har både en teknisk och en ekonomisk sida. För att förenkla resonemanget antas att ASEA kommer att byta mikroprocessor, vid en övergång till nämnda metod, och att bara metoder för resolver till digitalomvandling, som är långsammare än systemets samplingstid inte kan accepteras. CPU-snabbhet, minnesbehov och in/utgångar skall då inte längre vara något problem.

Då tekniskt acceptabla lösningar alltid finns, blir den kritiska faktorn ekonomin. Figur 1 och Figur 2 ger därvid en ganska klar bild av vad som försvinner och vad som tillkommer i systemet.

**FÖRSVINNER:** Axelenhet

**TILLKOMMER:** Resolver till digitalomvandlare, datorinterface och drivdonsinterface

Med tanke på att systemet har 5 axlar, skall man vara särskilt observant på möjligheten av att dela på de krets-lösningar som ryms under TILLKOMMER, när man tekniskt specificerar en lösning.

Helt allmänt kan man säga om den mikrodatorbaserade positioneringen att förutom den rent ekonomiska vinsten som kan göras, så föreligger också en möjlighet att göra en mera sofistikerad lägesreglering av axeln, både vad gäller den tid som åtgår för positioneringen, och den bana som skall följas.

Kapitel 1 skall leda fram till en förståelse om hur positionering går till i dagens system, för den som redan är välbekant med detta, rekommenderas att kapitlet hoppas över. I kapitel 2 redovisar jag olika metoder för resolver till digitalomvandling. I kapitel 3 redogöres för en möjlig hårdvarulösning, vid positionering av 5 axlar. Kapitel 4 beskriver den konstruktion, för en axel som användes vid de prov som redovisas i kapitel 5.

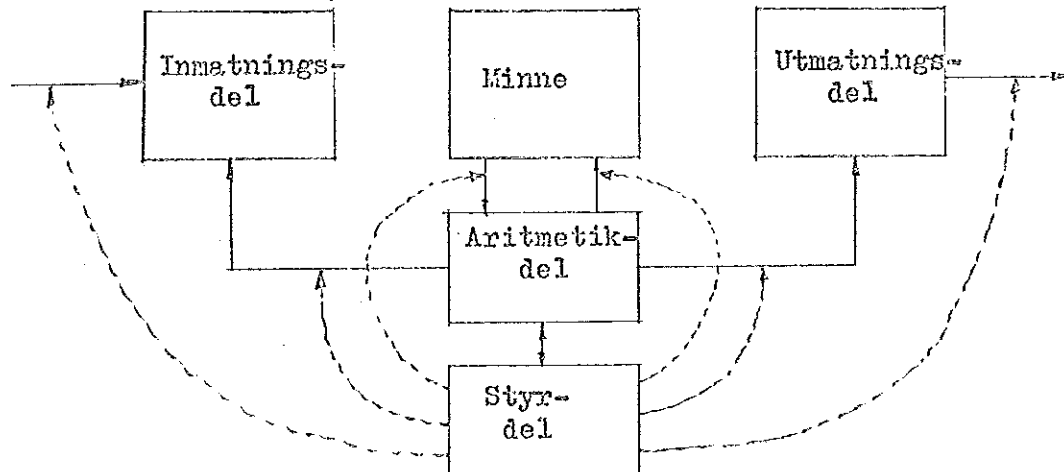
Slutligen vill jag rikta ett tack till, Hans Colleen som varit min handledare vid ASEA, för de råd och den ökade kunskap om systemet han gett mig, Ulf-Göran Norefors som speciellt utvecklat den metod för resolver till digitalomvandling som användes vid prov, men som även genom sin stora kunskap om hela robotsystemet bidragit till att arbetet kunnat gå framåt i rask takt, och Lennart Beckman som hjälpt mig vid programutveckling och programkörning.

# 1 BESKRIVNING AV KOMPONENTER I DAGENS POSITIONERINGSSYSTEM

## 1.1

### Datorn

Allmänt kan en dator delas upp i fem huvudbeståndsdelar enligt figur 3.



Figur 3 Datorns huvudbeståndsdelar

-inmatningsdel för inläsning av datainformation från yttre process

-aritmetikdel för databehandling

-minne för lagring av dels olika in- och utdata från respektive styrsystemenheter till processen, dels robot-systemets styrprogram

-styrdel för övervakning och kontroll av vad som skall utföras

-utmatningsdel för utläsning av datainformation till yttre processen

Samtliga delar i datorn utgöres av elektronikmoduler uppbyggda på kretskort som kan pluggas in i ett elektronikramverk. Aritmetik och styrdelarna består av funktionenheterna CENTRALENHET och AVBROTT OCH SPÄNNINGSÖVERVAKNING, se figur 4, sidan 7.

### Minnet

Minnet är organiserat i minnesord på vardera 8 bitar. Totalt kan 16k adresseras. Minnet är uppdelat i två delar. Den ena delen innehåller datorinstruktionsord och fasta datavärden.

Den andra delen av minnet innehåller dataord som definierar robotens rörelsemönster. Innehållet i dataorden kan alltså ändras under programmets gång eller vid programmering av roboten, medan datorinstruktionsorden definierar datorns så kallade styrprogram. Detta är fast. Därför användes i styrsystemet två olika typer av minnen. För dataorden används skriv/läsminne och för instruktionsorden läsminne.

#### Centralenhet, avbrottsystem och spänningsövervakning

Se figur 4.

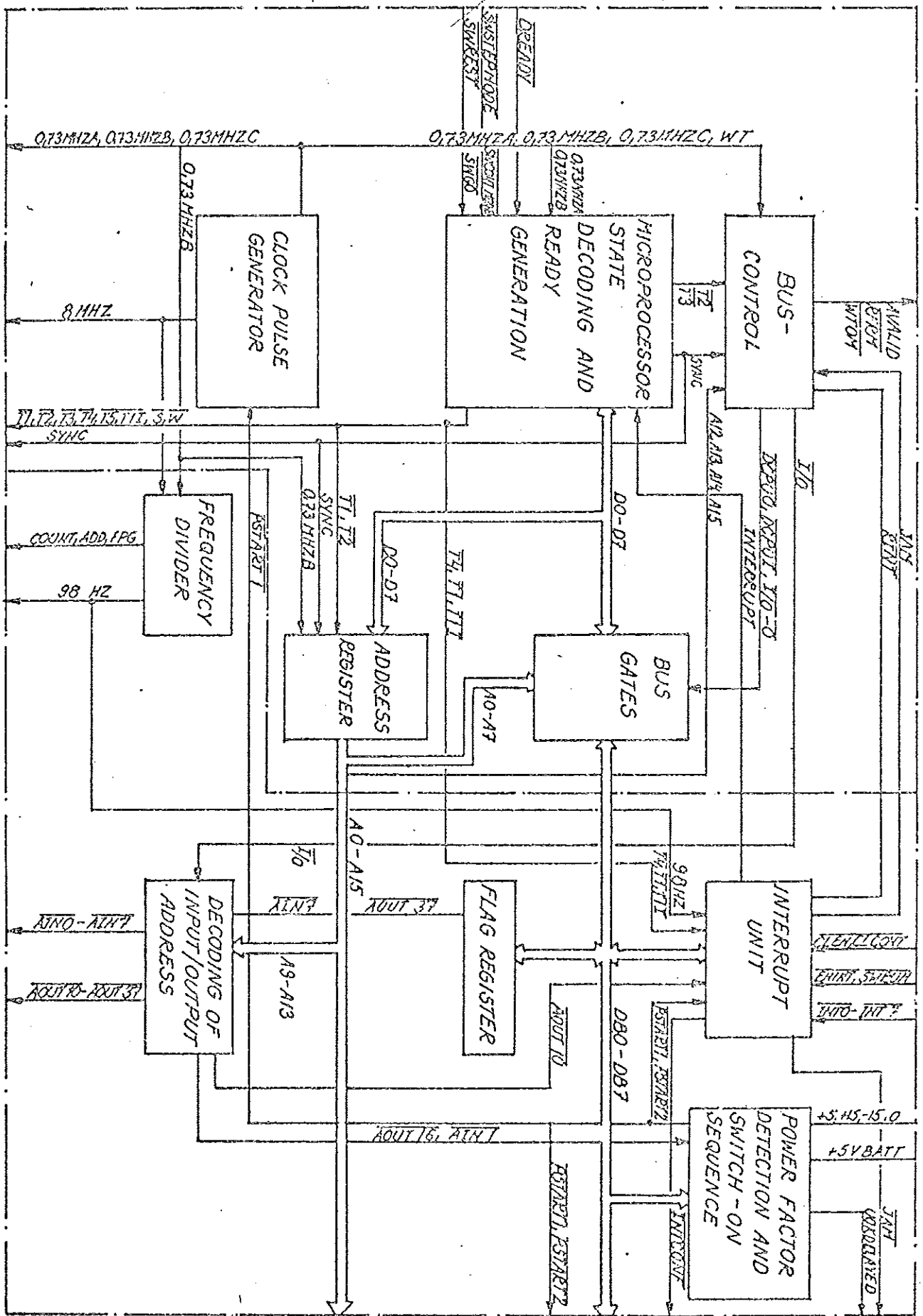
Enheterna utgör tillsammans med minnet en dator. Här finns bland annat en mikroprocessor, INTEL 8008, klockpulsgenerator och kretsar för att styra en 8 bitars databuss. I blocket klockpulsgenerator bildas ett antal pulser som används dels av mikroprocessorn, dels för att bilda olika styrsignaler som behövs i systemet bland annat en 98 Hz signal som utgör systemets samplingsfrekvens. All dataöverföring mellan funktionsenheterna i styrelektroniken sker via en 8 bitars databuss DBO-DB7. På denna buss sker överföringen i båda riktningarna vilket är möjligt genom tidsmultiplexing. Denna tidsmultiplexing verkställes från busskontrollblocket som via tillståndsutkodaren känner av i vilket tillstånd mikroprocessorn befinner sig i. Mikroprocessorn som under en typisk maskincykel genomlöper 5 tillstånd, sätter under tillstånd 1 upp de 8 minst signifikanta adressbitarna vilka läggs in i adressregistret och under nästa tillstånd sättes 8 nya bitar upp, av vilka 2 är dont care. De 6 mest signifikanta bitarna läggs in till adressregistret och en minnesadress är klar. De tre följande tillstånden användes sedan för att hämta data och exekvera instruktionen. Om det är fråga om en I/O-operation, kodas in-utadressen ut via A9-A13 och ackumulator innehåll som utgöres av bitarna A0-A7 länkas ut på databussen. Blocket för bussgrindar är ett buffertsteg mellan mikroprocessorn, databussen DBO-DB7 och adressbussen A0-A15. Bussen D0-D7 är den buss på vilken mikroprocessorn primärt sänder respektive slutligen tar emot data. Bussarna är av tri-state typ vilket innebär att utgångarna kan anta tre olika tillstånd nämligen logiskt 0 eller logiskt 1 när enheten är adresserad annars ett högimpeditivt tillstånd som inte belastar bussen. Avbrottsenheten länkar in avbrott till mikroprocessorn. En tillslagssekvens alstras och gör så att mikroprocessorn börjar med att exekvera styrprogrammet. I detta block kontrolleras också att alla elektronikspänningar är rätt. Flaggregistret slutligen kan direktadresseras från mikroprocessorn som använder detta för att lättare komma åt data som den ofta manipulerar med.

#### Mikroprocessorn

INTEL 8008, är en 8 bits parallell CPU-enhet av MOS-typ. 8008 adresserar med 14 bitar och klarar alltså ett minne på 16k. Enheten har 48 instruktioner i sin repertoar. I medeltal tar en instruktion 22 $\mu$ s.



Fig. 2-1 Block diagram of central unit with interrupt and power failure detection unit. 7 epis



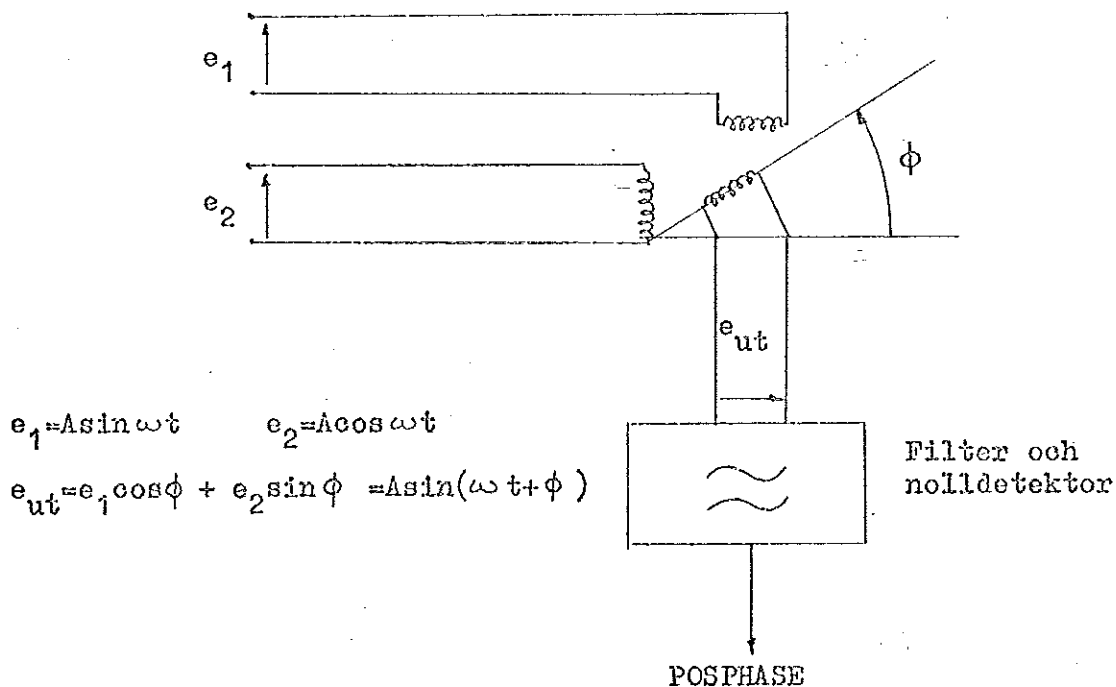
Figur 4 Centralenhet, avbrottsystem och spänningsövervakning

## 1.2

## Lägesmätning med resolver

Resolvern är en typ av elgon vilken har två mot varandra vinkelräta statorlindningar och en rotorlindning. Statorlindningarna matas med två sinussignaler med samma amplitud men med inbördes 90 graders fäsförskjutning. Rotorspänningen kommer då att bli en sinussignal med konstant amplitud men med en fäsförskjutning exempelvis relativt sinusinspänningen, som kommer att variera linjärt med rotorns vinkeläge i rummet. (Se även kapitel 2 avsnitt 1, om den fasanaloga metoden)

Matningsspänningens frekvens är 2 kHz. Den från rotorn erhållna signalen filtreras och omvandlas till en fyrkant-signal. Denna som även har frekvensen 2 kHz, kallas POSPHASE. Se figur 5.

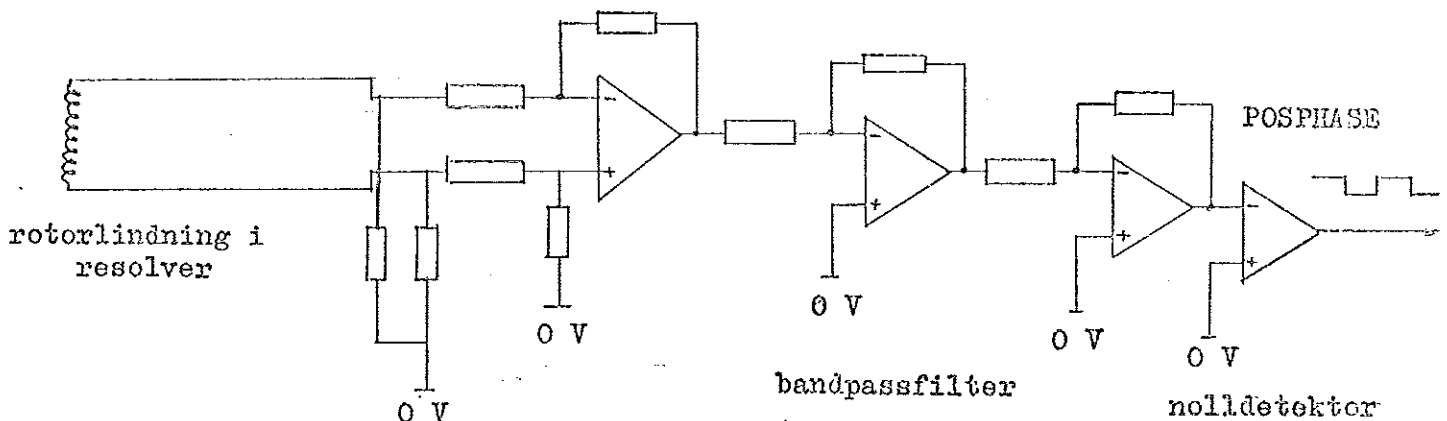


Figur 5 Resolver

Resolvern är kopplad till robotaxelns drivmotor så att om robotaxeln flyttar sig 200 robotinkrement kommer resolvern att rotera ett varv. Den minsta förflyttningen utgöres av ett robotinkrement.

### Filter och nolldetektor

Den signal som återföres från resolverns rotorlindning består av grundtonen på 2 kHz och en mängd övertoner härhörande från matningen med fyrkantvågor. Det är dock endast grundtonen som är användbar som läggsignal, varför övertoner filteras bort i ett bandpassfilter. För att underlätta denna filtrering innehåller fyrkantvågorna för resolvermatningen spikar, vilka undertrycker övertoner. Bandpassfiltrets uppbyggnad framgår av figur 6.



för förstärkare

Figur 6 Filter och nolldetektor

Det består av en lågpasdel och en högpasdel. Lågpasdelen innehåller två OP-förstärkare, varav den första har differentialingång, som eliminerar störningar gemensamma för de båda signalledningarna. Ett RC-filter på kortingången filtrerar bort högfrekventa yttre störningar och gör mätgivarsignalen symmetrisk kring noll volt. Högpasdelen avser eliminera lågfrekventa störningar och likspänningsnivån, som annars skulle medföra fel vid detekteringen av nollgenomgången.

Nolldetektorn består av en snabb OP-förstärkare med hög förstärkning. Den är kopplad som en jämförare med liten hysteres, för att få ett väldefinierat omslag. Nolldetektorn blir även mindre känslig för brus. Den så erhållna signalen utgör ärvärdesfasen.

### 1.3 Axelkort

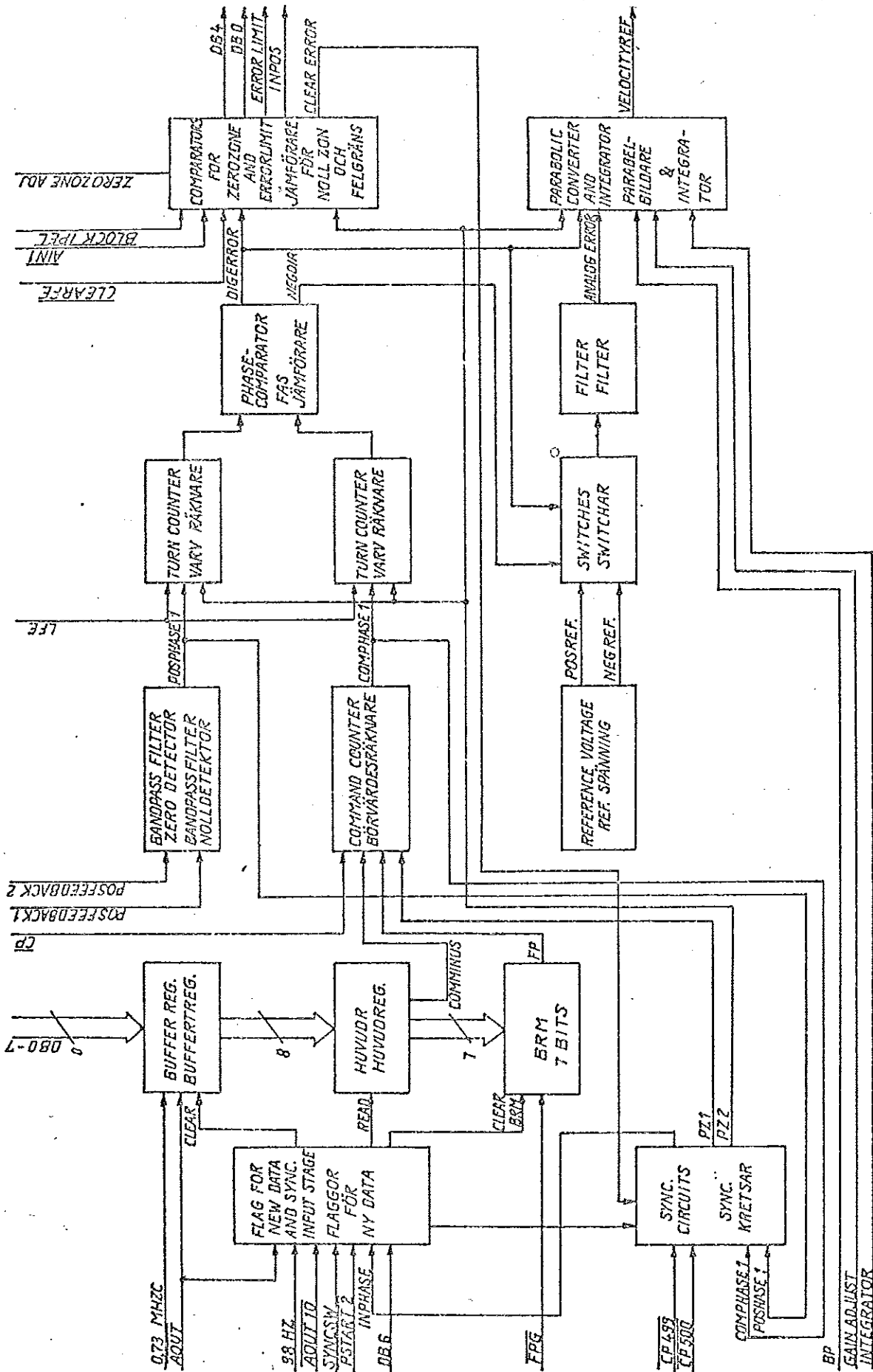
Funktionsenhetens uppgift är dels att utföra en noggran digital/analog omvandling av det från datorn beordrade läget för en robotaxel, dels att jämföra detta beordrade läge med robotaxelns verkliga läge och ur skillnaden bilda en hastighetsreferens för drivdonet. En enhet krävs för varje axel.

Principen för en förflyttning av en robotaxel i dagens system är enligt följande.

En ny positionering börjar med att den nya punktens koordinatvärden läses in i börvärdesregistret, medan robotens verkliga läge representeras av ett ärvärdesregister. De båda registernas innehåll anger läget relativt en fix utgångspunkt, vilken är robotens synkroniseringspunkt. I ett tredje register lagras ett tal  $ds$  på sju bitar plus en teckenbit, som anger den beordrade hastigheten och rörelseriktningen för positioneringen. Under förflyttningen arbetar datorn enligt ett samplingsförfarande. Detta innebär att den regelbundet (var 10:e ms) jämför talen i börvärdes- och ärvärdesregisterna. Om skillnaden är större än  $ds$ , lägger datorn ut detta tal till axelenheten tillsammans med teckenbiten för att motsvarande förflyttning skall verkställas. Samtidigt uppdateras datorns ärvärdesregister. Då skillnaden bör- ärvärdesregister blivit mindre än  $ds$ , läggs resten ut under ett sista samplingsintervall och ärvärdesregistret sätts lika med börvärdesregistret. Därefter är datorn passiv och inväntar kvittenssignal från axelenheten före nästa programblock exekveras. Detta gäller dock bara om man finpositionerar. Signalen INPOS från axelenheten är wire-and kopplad till de andra axlarna, och datorn väntar således tills dess att alla axlarna befinner sig i position. Om man grovpositionerar lägger datorn ut vad som är kvar att lägga ut under det sista samplingsstillfället och övergår sedan till att utföra nästa robotinstruktion. Lägesregulatorn på axelenheten arbetar enligt den fasanaloga metoden. Bör och ärvärde för robotaxeln representeras alltså av fasläget hos två fyrkantvågor. Börvärdet genereras elektroniskt på axelenheten medan ärvärdet utgöres av svaret från resolvern. Skillnaden mellan signalernas fasläge genererar den analoga felsignal som användes som hastighetsreferens av drivdonet. I systemet kommer en eftersläpning att byggas upp tills hastighetsreferensen är så stor som den programmerade. Då datorn lagt ut sitt sista förflyttningsinkrement och börvärdesfasen inte ändras mer, kommer eftersläpningen att användas som retardationssträcka.

Figur 7 visar ett blockschema över STYRKRETS FÖR AXEL och nedan följer en kortfattad funktionsbeskrivning.

1-0 övergången hos POSPHASE1 representerar ärvärdessignalens fasläge. Beroende på vilken maximal eftersläpning som tillåtes i systemet delas POSPHASE1 och COMPHASE1 i en varvräknare som kan ställas in för 32 eller 64 varv.



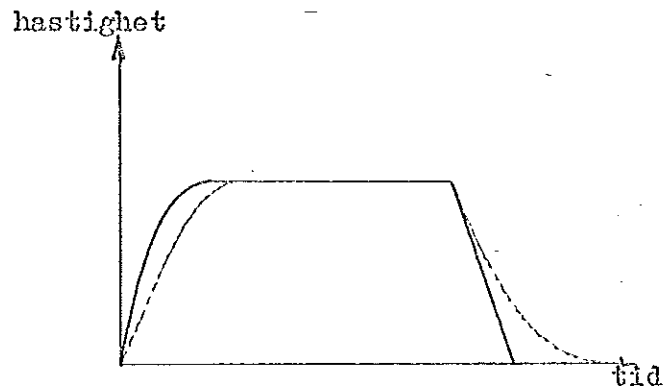
Figur 7 Blockschemata över STYRKRETS FÖR AXEL

Fig. 8-1 Blockschemata över STYRKRETS FÖR AXEL

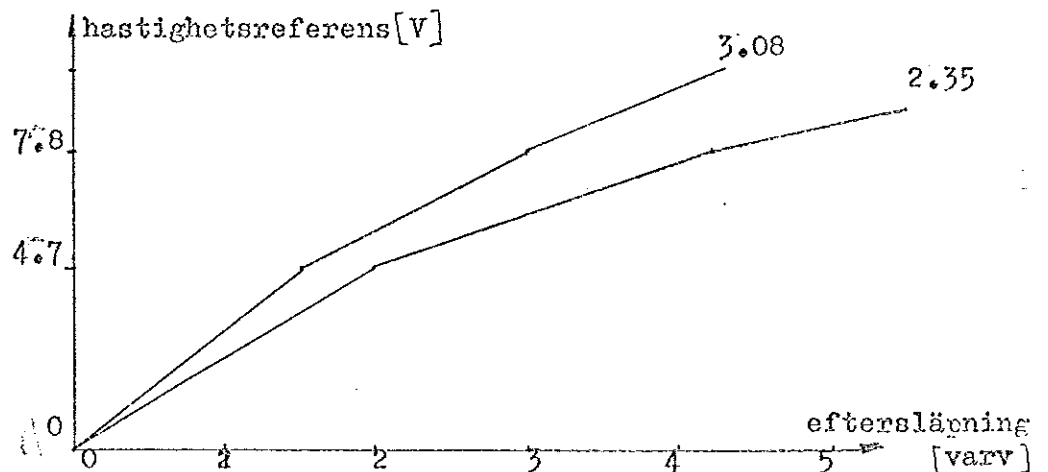
Börvärdesfasen är 1-0 övergången hos fyrkantvågen COMPHASE1. Fasläget hos denna signal kan via börvärdesräknaren, vilken är en cirkulerande modulo 1000-räknare förskjutas i steg om  $1/1000$  period, styrt från klockpulssignalen CP på 2 MHz. Fasläget kan förskjutas i såväl positiv som negativ riktning beroende på om en klockpuls hoppas över eller räknas två gånger. Dessa steg är således mätsystemets upplösning och kallas resolverinkrement. Faslägsjämföraren bildar en digital felsignal DIGERROR som i en omkopplare normeras till en väldefinierad amplitud. Om sedan NEGDIR är ett kopplas en negativ referensspänning in annars en positiv. På så sätt fås en fyrkantvåg vars medelvärde är ett noggrant mått på felets storlek. Signalen medelvärdesbildas i ett lågpassfilter. ANALOGERROR får sedan gå in på en parabelbildare vars utsignal är hastighetsreferens för drivdonet.

#### Parabelbildare

För att få minimal retardationstid och retardationssträcka har en parabelbildare införts i den slutna lägesregleringen. Idealt skall då hastighetsreferensen vara proportionell mot roten ur lägesfelet. Detta ger konstant retardation, se figur 8. Figur 9 visar parabelbildarens karakteristik.



Figur 8 Förflyttning med  $\text{---}$  och utan  $\text{- - -}$  parabelbildare



Figur 9 Parabelbildarens karakteristik (förstärkningen under första brytpunkten 3.08 respektive 2.35 V/varv)

2

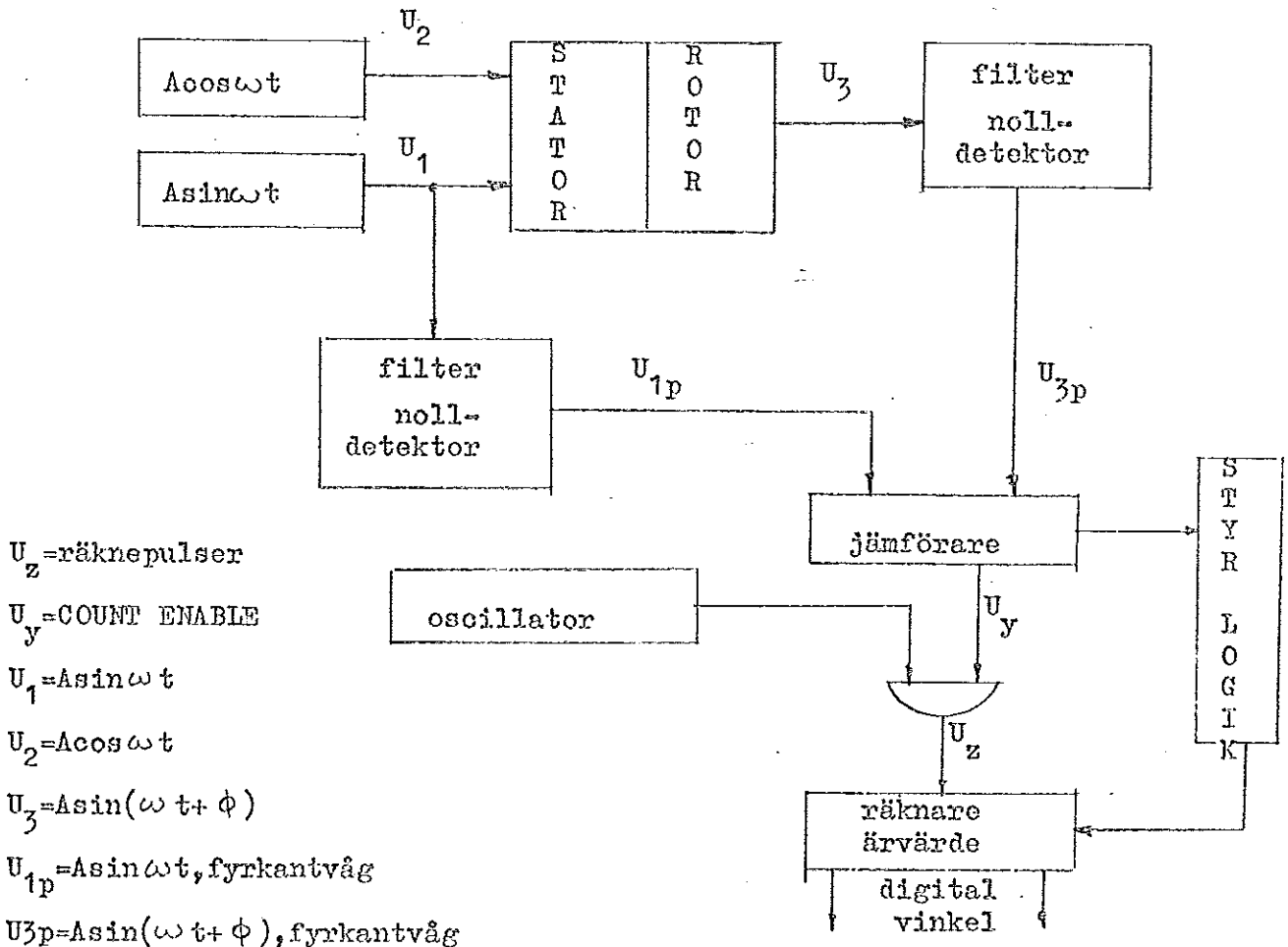
Allmänt om resolver till digitalomvandling

2.1

Fasanaloga mätmetoden

Metoden innebär att statorlindningarna matas med två spänningar av samma amplitud men med 90 graders inbördes fasförfskjutning. Som tidigare visats kommer då rotorspänningen att bli  $A\sin(\omega t + \phi)$ , det vill säga en spänning som relativt en av matningsspänningarna, har ett fasläge som direkt tjänar som ett mått på axelns läge (inom 360 grader).

Principiellt sker digitaliseringen av läget enligt figur 10. Det som skiljer de olika sätten, är hur man väljer att göra jämförelsen mellan signalerna  $U_{1p}$  och  $U_{3p}$ .

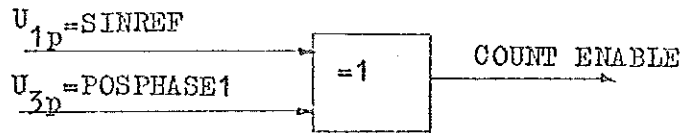


Figur 10 Digitalisering av axelläge vid fasanalog mätmetod.

Vid utvecklingen av den lösning som skulle användas vid utprovning av den mikrodatorbaserade positioneringen, diskuterades om jämförelsen kunde göras med en modulo-2 grind.

Jämförare=modulo-2 grind

Signalen COUNT ENABLE bildas enligt figur 11.

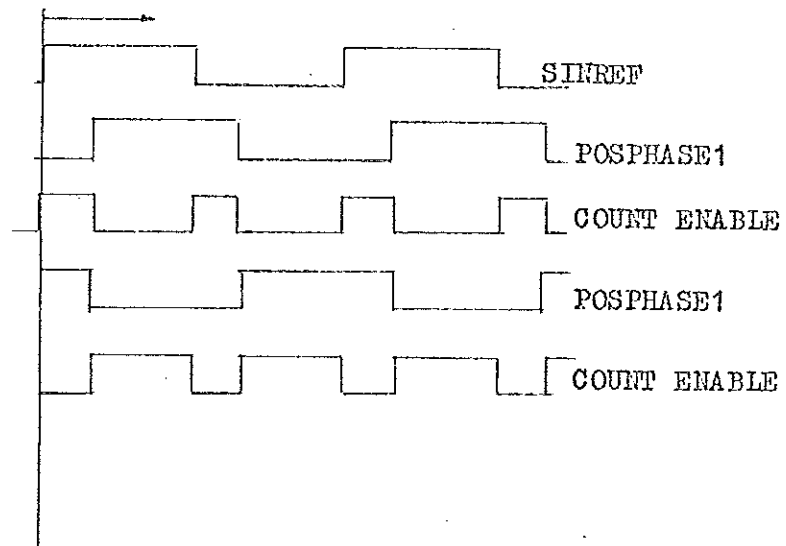


Figur 11 COUNT ENABLE med modulo-2 grind

Den filtrerade och nollgenomgångsdetekterade signalen från rotorlindningen motsvaras av POSPHASE1 i dagens system. På funktionsenheten REFERENSRÄKNARE OCH RESOLVERFÖRSTÄRKARE genereras signalen SINREF.

Figur 12 visar hur signalen COUNT ENABLE ser ut vid positionering i positiv riktning.

t=0



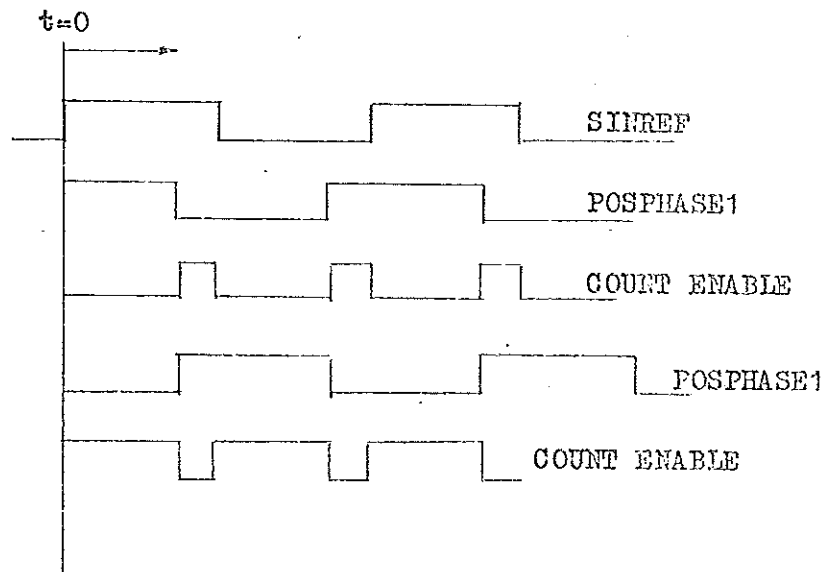
Figur 12 COUNT ENABLE då fasläget hos POSPHASE1 rör sig åt höger, motsvarande positiv riktning

Anm

För enkelhets skull visar figur 12 två positiva vinklar med axeln stillastående.



Då axeln rör sig i negativ riktning kommer COUNT ENABLE att se ut enligt figur 13.



Figur 13 COUNT ENABLE då fasläget hos POSPHASE1 rör sig åt vänster, motsvarande negativ riktning.

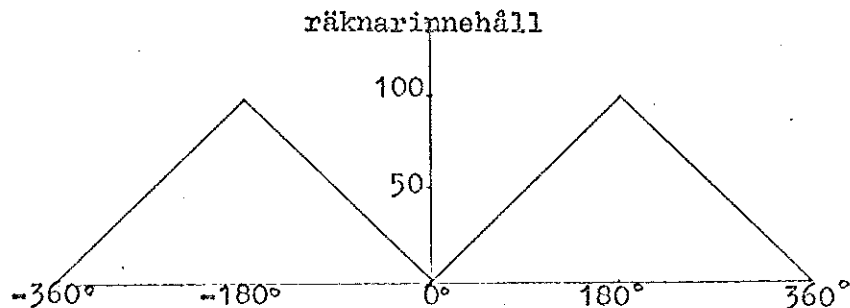
Ann

För enkelhets skull visar figur 13 två negativa vinklar med axeln stillastående.

Med hjälp av figur 12 och figur 13 kan man nu rita upp ett diagram som kan användas för att koda ut räknar innehåll.

Eftersom 360 graders fasdifferans mellan SINREF och POSPHASE1 motsvarar 200 robotinkrement får man att 180 graders fasdifferans skall innebära ett räknarinnehåll på 100.

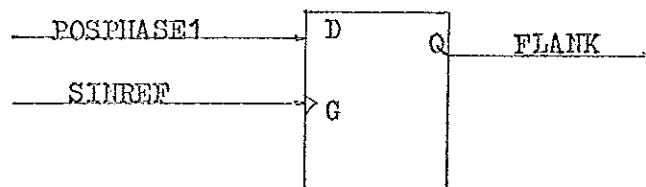
Figur 14 visar räknarinnehållet som funktion av fasskillnaden mellan SINREF och POSPHASE1, med positiv fasskillnad motsvarande ett positivt läge på robotaxeln och negativ fasskillnad motsvarande ett negativt läge på axeln.



Figur 14 Fasskillnad-räknarinnehåll

För att kunna räkna ut hur mycket och i vilken riktning som axeln vridit sig mellan två samplingsstillfällen måste man först och främst maximera hastigheten till 99 inkrement per sample, men man måste också känna till på vilken sida om +180 respektive -180 grader man ligger.

Därför bildas signalen FLANK enligt figur 15.



Figur 15 Signalen FLANK

FLANK blir 1 om SINREF:s 0-1 övergång inträffar när POSPHASE1 är 1. Annars blir FLANK 0.

Med hjälp av figurerna 12 och 13 får man att,

$-360 \leq \phi < -180$	FLANK=0
$-180 \leq \phi < 0$	FLANK=1
$0 \leq \phi < 180$	FLANK=0
$180 \leq \phi < 360$	FLANK=1

Kännedom om värdet av FLANK och att maximal hastighet är 99 inkrement per sample är nu fullt tillräcklig för att robotaxelns position skall kunna bestämmas i mjukvaran som sköter positioneringsberäkningarna.

Eftersom FLANK är relaterat till det värde som man har sist läst ned till datorinterfacet (buffertregister), får man se till att samma signal som laddar buffertregistret också laddar en tri-stabil vippra med FLANK, som får tillfogas i datorinterfacet.

Överhuvudtaget ger denna metod genom sina diskontinuiteter och det på det hela taget asynkrona förloppet upphov till att man måste införa en del kompenseringselektronik varför jag avstår från att föra resonemanget ned på kretslösning-nivå. Jag vill bara fastlägga att metoden är fullt möjlig att använda men att den slutliga mängden elektronik inte lär understiga den som prov-lösningen innehåller.

Sammanfattningsvis kan man säga att den fasanaloga mätmetodens stora fördel är att den elektronik som åtgår för digitaliseringen är okomplicerad, medan en nackdel är att filtreringen och fastläggandet av nollgenomgången måste ske noggrant för att inte mätprecisionen skall påverkas.

2.2

Omvända fasanaloga mätmetoden

Rotorlindningen matas med en växelspanning av konstant amplitud

$$U_3 = A \sin \omega t$$

De spänningar som då tages ut från två i rummet 90 grader förskjutna statorlindningar kommer att bli

$$U_1 = U_3 \cos \phi$$

$$U_2 = U_3 \cos(\phi + 90) = -U_3 \sin \phi$$

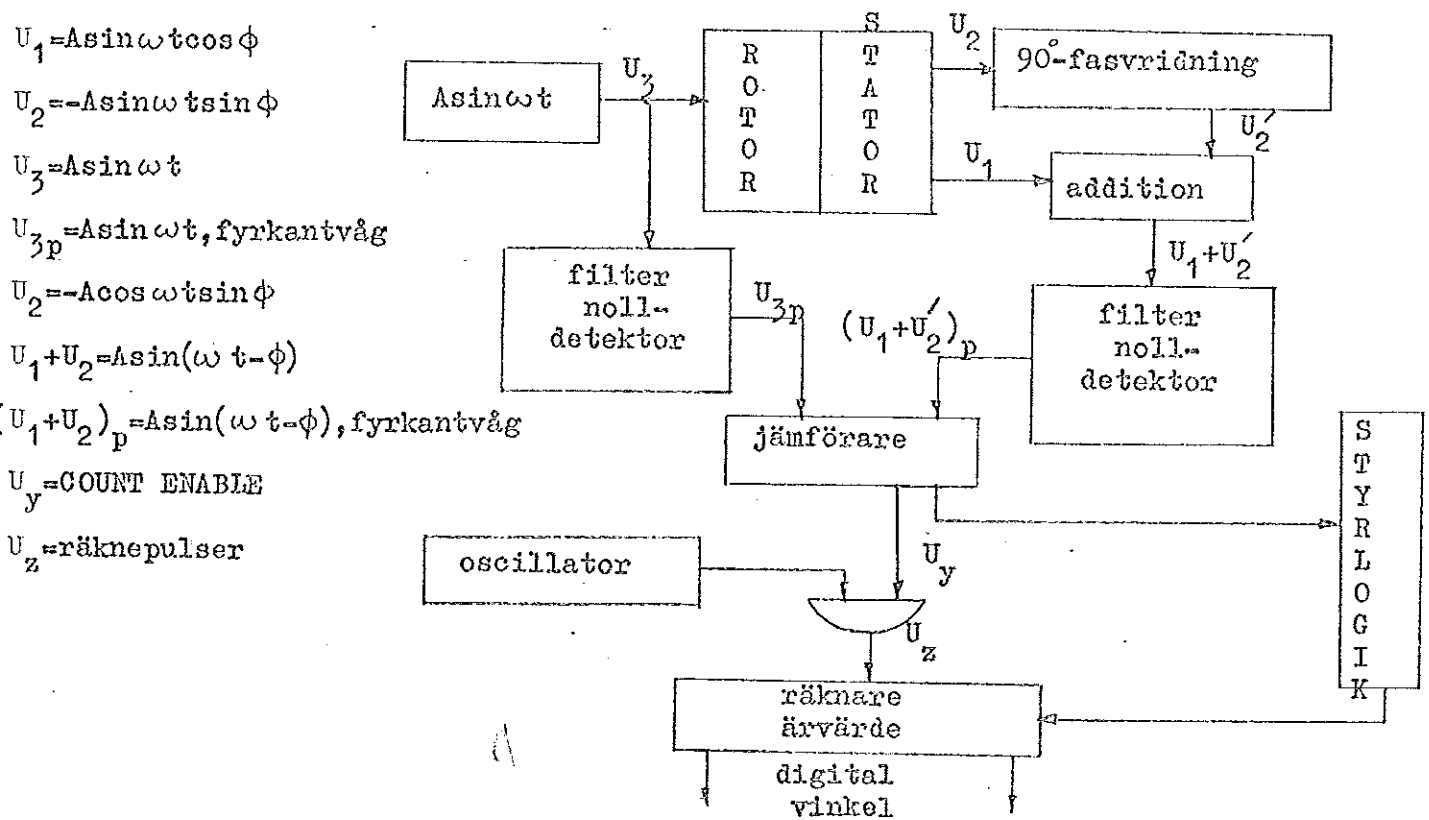
Signalen  $U_2$  fasvrides därför  $90^\circ$  och adderas till  $U_1$

$$U_1 + U_2' = A(\sin \omega t \cos \phi - \cos \omega t \sin \phi)$$

$$U_1 + U_2' = A \sin(\omega t - \phi)$$

En jämförelse mellan  $U_3$  och  $(U_1 + U_2')$  ger nu en möjlighet att fastställa fasförskjutningen.

Figur 16 visar principen för digitalisering av fasskillnaden, då man använder den omvända fasanaloga mätmetoden.



Figur 16 Digitalisering vid omvänd fasanalog mätmetod.

Från figur 16 framgår att digitaliseringen av fasdifferansen då man använder den omvända fasanaloga mätmetoden, kräver mera elektronik för att skapa de signaler som skall in till blocket-jämförare- men att excitationsförfarandet är för- enklat, jämfört med den fasanaloga mätmetoden.

Samma metoder som användes i jämföraren vid fasanaloga mätsystem kan användas här.

### 2.3

#### Amplitudanaloga mätmetoden

Metoden innebär att man exciterar rotorlindningen och sedan tar ut tvåfas eller trefas spänningar från statorlindningarna. Resolver- eller syncroformat på spänningarna. Sedan arbetar man direkt på de amplitudmodulerade statorspänningarna och omvandlar inte dessa till en resulterande spänning med en viss fasförskjutning relativt excitationsspänningen som i omvända fasanaloga mätmetoden.

Resolverformat på statorspänningarna får man om statorlindningarna är 90 grader förskjutna i förhållande till varandra i rummet, syncroformat erhålles då man har trefas statorlindningar.

Om man exciterar rotorn med  $A\sin\omega t$  får man i resolver fallet ut statorspänningarna

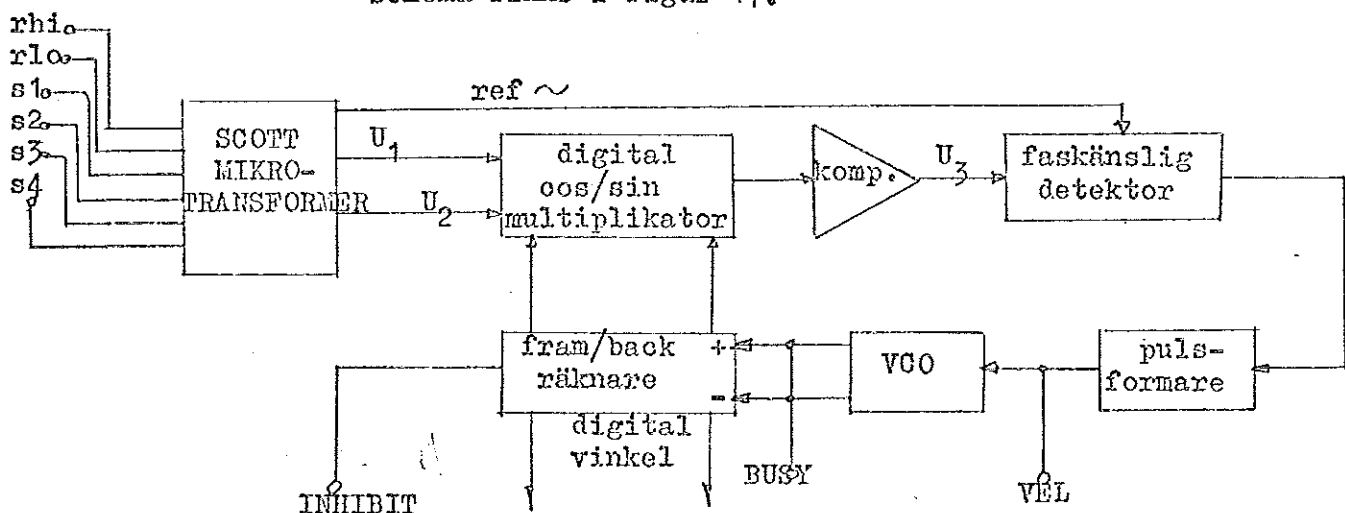
$$A\sin\omega t\cos\phi$$

$$A\sin\omega t\sin\phi$$

Speciella IC-kretsar finns på marknaden som tar hand om de bägge statorspänningarna och levererar vinkeln på sina utgångspinnar.

#### Tracking converter

Resolver till digitalomvandlaren kommer från Memory Devices och försäljes av bolaget NAXAB i Solna. IC-kapseln i blockschema finns i figur 17.



Figur 17 Tracking converter enligt Memory Devices

Kapsel funktion

Modulen är konstruerad för att kunna ta emot signaler i både resolver- och syncroformat. Kapseln arbetar internt med resolverformat på signalerna, varför signaler i syncroformat först får passera en Scott transformer, för omvandling. Detta ingångssteg fungerar bara som en buffert för resolverformat signaler.

Med excitationsspänningen  $A \sin \omega t$  och fasskillnaden  $\phi$  gäller att

$$U_1 = A \sin \omega t \sin \phi$$

$$U_2 = A \sin \omega t \cos \phi$$

Omvandlingsprocessen går nu till så att  $U_1$  multipliceras med  $\cos \theta$  och  $U_2$  med  $\sin \theta$ , där  $\theta$  är fram/back räknarens tillstånd. I komparatören bildas en felsignal

$$U_3 = A \sin \omega t \sin(\theta - \phi)$$

Den faskänsliga detektorn och den spänningsstyrda oscillatören bildar en loop som strävar att nolla  $\sin(\theta - \phi)$ . När detta är gjort finns vinkeln  $\phi$  på fram/back räknarens utgångspinnen.

Omvandlingen är klar då signalen BUSY är låg. Spänningarna som uppträder på BUSY-pinnen korresponderar mot de pulser som den spänningsstyrda oscillatören levererar för att inkrementera eller decrementera räknaren.

Vinkeln på utgången representeras genom vanlig binär viktning där den mest signifikanta biten representerar 180 grader, nästa 90 grader och så vidare.

Spänningsanslutningarna är: excitationsspänningen till r1i och r1o, statorspänningarna till s1 s3 respektive till s2 s4.

INHIBIT: öppnar loopen och blockerar uppdateringen av räknaren

VEL: spänning som motsvarar axelns vridningshastighet. Kan användas istället för tachometer.

Ur datablad för kapseln hämtar man följande.

1. 10, 12 eller 14 bitars upplösning av 360 grader
2. Tracking hastighet större än 75 varv/s. Om excitationsspänningen väljes till  $A \sin \omega t$ , med  $\omega = 2.6$  kHz
3. Återhämtningstid vid 180 graders stegändring av vinkeln, 50 ms.
4. BUSY pulsvidd 0.5-1.25  $\mu$ s

Slutsatser

1. Upplösningen är fullt tillräcklig, då det ju räcker med 200 delar för ett varv.
2. Den max. hastighet som axeln körs med idag är 64 varv/s.

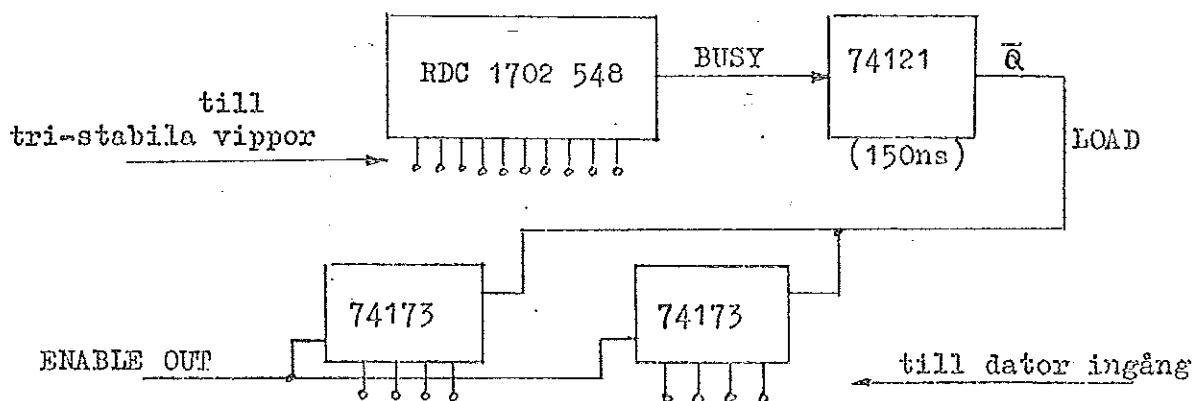
Denna hastighet ligger under kapselns kapacitet, om man exiterar med frekvensen 2.6 kHz istället för som nu med 2 kHz

3. Kapselns återhämtningstid vid stegändringar är inte försumbar. Detta innebär att man måste använda en kapsel per axel.
4. Vid användning av 10 bitars kapseln motsvarar 1 varv 1023 inkrement i räknaren. Med en hastighet av 64 varv/s alltså 64·1023 pulser/s. Tiden mellan två BUSY pulser blir  $\approx 15 \mu\text{s}$ . Med en BUSY pulsvidd på  $\approx 1 \mu\text{s}$  har man  $\approx 14 \mu\text{s}$  på sig för att läsa räknarens innehåll. Då datorn behöver  $\approx 2.75 \mu\text{s}$  för att läsa ett buffertregister finner man att läsningsoperationen går bra då axeln körs med maximal hastighet

#### Interface med dator

Om man använder 10 bitars kapseln får man ett enklare dator-interface om de två minst signifikanta bitarna betraktas som don't care.

Figur 18 visar principen för datorinterfacet.



Figur 18 Princip för datorinterface

Metoden innebär att de tri-stabila vipporna kontinuerligt uppdateras 150 ns efter BUSY signalens 0-1 övergång.

RDC 1702 548 är en kapsel med

RDC: resolver till digitalomvandling

1702: 10 bitars upplösning av 360 grader

5: temperaturintervall på 0-70 °C

4: matningsspänningsfrekvens på 2.6 kHz

8: spänningsnivå på 11.8 V för mätsignal och 26 V för referens

Fördelar med metoden

Störningsokänslig, tål svår industrimiljö och ger en takometer signal.

Nackdelar

Metoden blir dyr, kostnaden per kapsel ligger i storleken 1000 kronor. Inga möjligheter finns för delning av en kapsel mellan flera axlar, då återhämtnings tiden inte är försumbar.

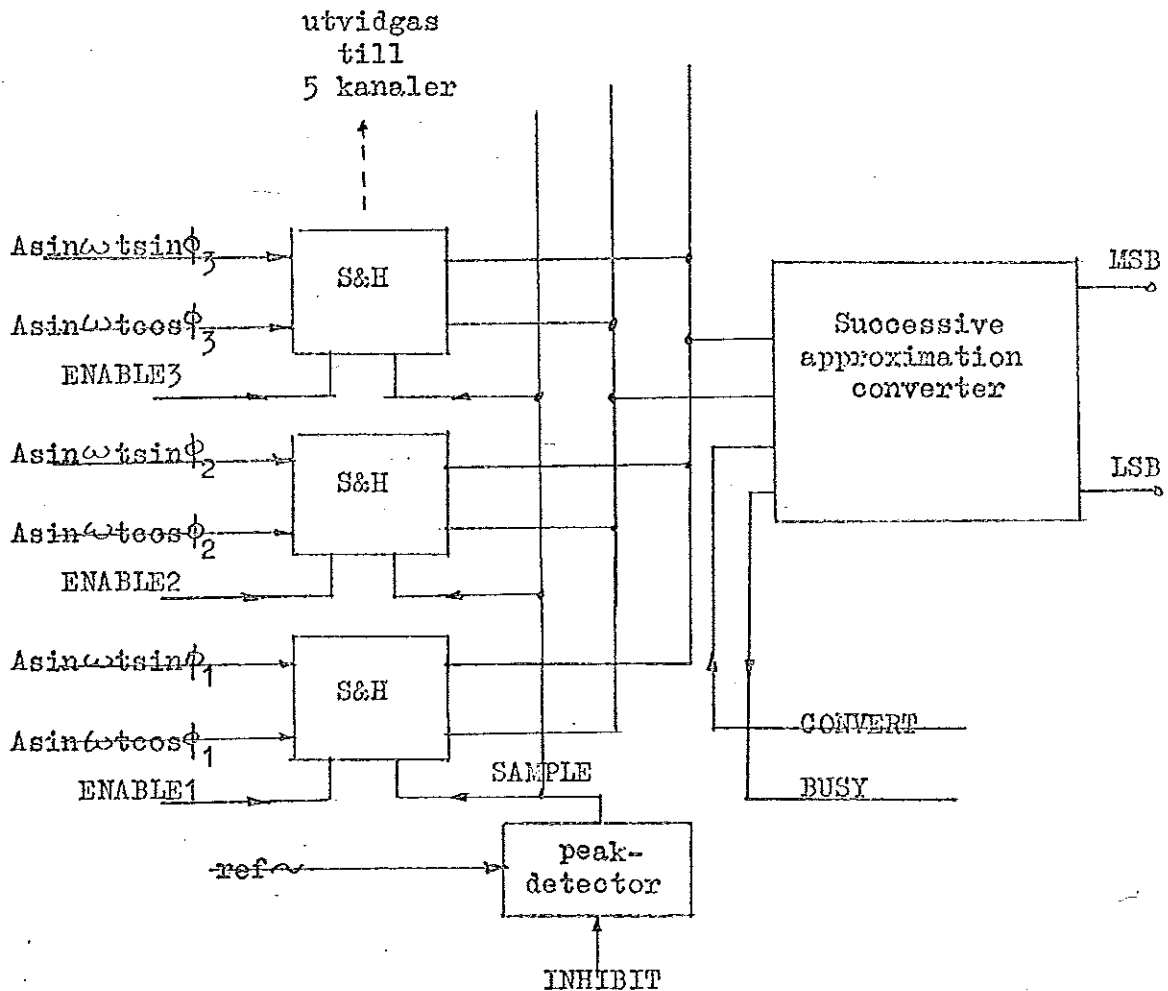
3

Hårdvara för mikrodatorbaserad positionering av 5 axlar

3.1

Resolver till digitalomvandling

Den största olägenheten med tracking convertern som redovisades i kapitel 2 avsnitt 3, var att den inte kunde delas av flera axlar. Memory Devices har emellertid en IC-krets (Successive approximation converter), som kan användas i ett multiplexat system. Se figur 19.



Figur 19 Multiplexat system för resolver till digitalomvandling.

De på referensspänningens toppvärde samplade resolver-signalerna  $Asin \omega t \cos \phi$  och  $Asin \omega t \sin \phi$ , utgör insignaler till omvandlaren. Omvandlaren väljer först kvadrant genom att addera  $90^\circ$ ,  $180^\circ$  eller  $270^\circ$ . Den ingående sin/cos-multiplikatorn kommer därför alltid att arbeta i första kvadranten. Utgångarna från multiplikatorn subtraheras sedan för att ge en felsignal som efter att ha förstärkts orsakar att approximationsregistret ökar eller minskar, register utgången är sedan den sökta vinkeln.



För 8 bitars noggrannhet i den digitala vinkeln kommer man ned till  $\approx 40 \mu\text{s}$  i omvandlingstid, vilket är acceptabelt. Lämpligen låter man då datorn läsa värdet på en axel, lagra undan detta i ett register, göra ENABLE på en ny axel och ge CONVERT COMMAND till omvandlaren och sedan reglera den inlästa axeln. På så sätt kommer den korta omvandlingstiden inte alls att märkas i systemet. Datorn har hela tiden vettigt arbete att utföra.

För 5 axlar går det åt en omvandlare, 5 Sample&Hold kretsar och en Peak detector. Datorinterfacet kan göras likadant som för Tracking convertern. Summa, en måttlig samling komponenter som bör rymmas på 1/4 kort.

#### Nackdel med metoden

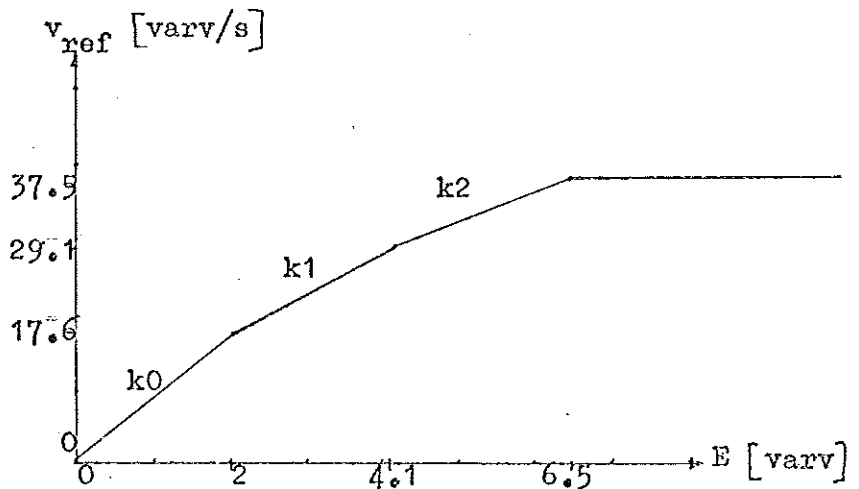
En allmän nackdel med samplande metoder är känsligheten för brus på mätsignalerna. Emellertid, med den grova upplösning av 360 grader som tolereras, borde detta ej vara något problem.

### 3.2

#### Drivdonsinterface

I den följande framställningen skall jag dra upp riktlinjer som man kan följa när man konstruerar ett drivdonsinterface, gemensamt för alla 5 axlarna och med en billig 8 bitars D/A-omvandlare.

Hastighetsreferensen som skall läggas ut till drivdonet som funktion av eftersläpningen  $E$ , visas i figur 20. (Hastighetsreferensen är klippt vid 37.5 varv/s vilket motsvarar en spänning på 10 V)



Figur 20 Hastighetsreferensen som funktion av eftersläpningen.

Förstärknings faktorerna i parabelbildaren är:

$$k_0 = 8.81 \text{ [1/s]}$$

$$k_1 = 5.46 \text{ [1/s]}$$

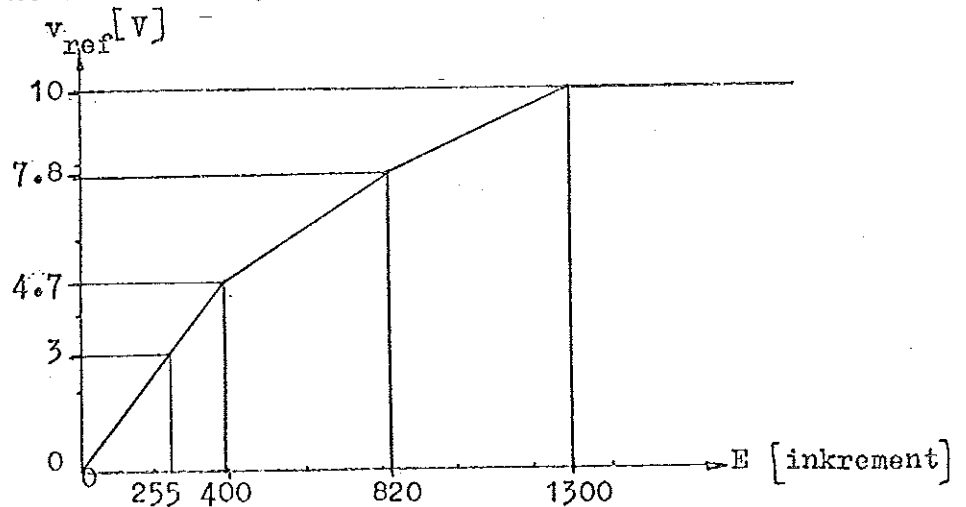
$$k_2 = 3.52 \text{ [1/s]}$$

En 8 bitars D/A-omvandlare utnyttjas, med avseende på upplösningen optimalt om man inte lägger in någon teckenbit till den, utan bara arbetar med positiva spänningar på utgången.

Upplösningen blir då  $2^8 - 1 = 255$  bitar.

Man vill nu ha stor upplösning på hastighetsreferensen vid låga hastigheter (eftersläpning upp till 2 varv). Medan man vid högre hastigheter tillåter en grövre upplösning på hastighetsreferensen.

Därför delas det område inom vilket eftersläpningen varierar, in i 4 olika delar. Figur 21 visar områdesindelningen, med hastighetsreferens axeln graderad i spänning och eftersläpnings axeln i robotinkrement.



Figur 21 Områdesgränser i parabelbildaren

Område 1  $0 \leq E < 255$

Område 2  $255 \leq E < 400$

Område 3  $400 \leq E < 820$

Område 4  $820 \leq E \leq 1300$

Med denna områdesindelning ser man att upplösningen i hastighetsreferensen bibehålles för  $E < 400$ .

Ann

Det är inget problem att behålla upplösningen i hastighetsreferensen i hela det område inom vilket eftersläpningen varierar. Områdena 3 och 4 i figur 21 får då vart och ett delas upp i två områden.

Kopplingen i blockschemaform, av drivdonsinterfacet visas i figur 22. Vissa delar av figur 22 finns uppförstorade i separata figurer, nämligen:

Operationsförstärkarkoppling för område 1, i figur 22a.

Operationsförstärkarkoppling för område 2, i figur 22b.

Operationsförstärkarkoppling för område 3, i figur 22c.

Operationsförstärkarkoppling för område 4, i figur 22d.

Funktionsbeskrivning figur 22, drivdonsinterface.

Datorn noterar att till exempel  $\phi$  axelns eftersläpning skall beräknas. Datorn utför beräkningen och gör sedan ett antal test för att bestämma inom vilket område eftersläpningen ligger. Om datorn finner att den ligger i områdena 1 eller 2, subtraheras den undre områdesgränsen från eftersläpningen och resten läggs ut till D/A-omvandlaren. Låg eftersläpningen däremot i område 3 eller 4, ändrar datorn även på upplösningen hos hastighetsreferensen genom att dividera eftersläpningen med 2, innan utlägget sker. Därefter läser datorn ut 3 bitar kontrollinformation till ett buffertregister. Kontrollbitarna är till för att rätt spänning, med tecken, skall kunna länkas fram till Sample&Hold kretsarna.

Datorn måste sedan ge kommandot SAMPLE till  $\phi$  axelns Sample&Hold krets. Detta gör den genom att lägga ut 8 bitar varav en är noll, till ett annat buffertregister. Den Sample&Hold krets som hör till  $\phi$  axeln kommer då att sampla in sin spänning.

Till sist innan datorn går över till att behandla nästa axel, gör den HOLD på Sample&Hold kretsen, genom att läsa ut bara ettor till buffertregistret.

Ann

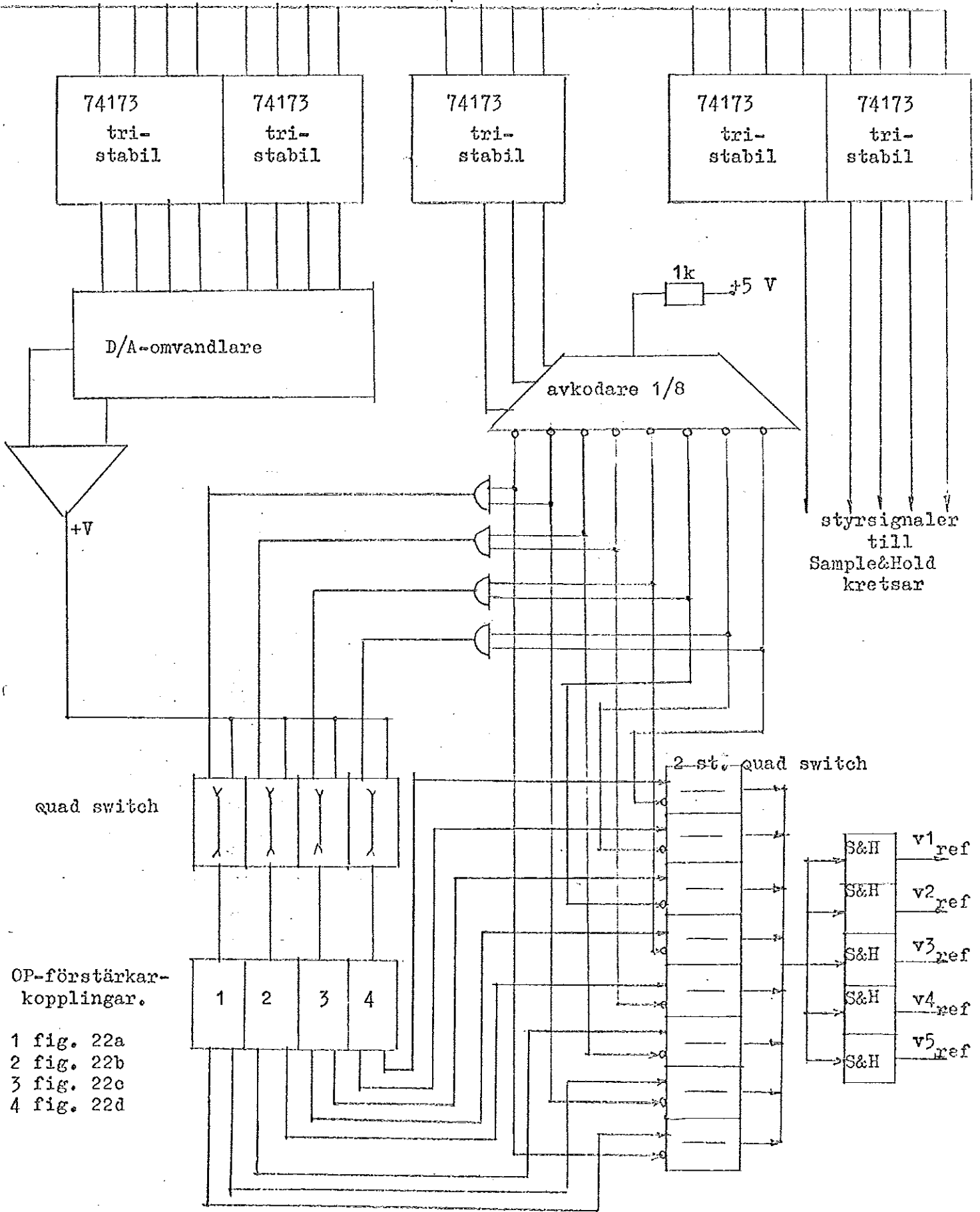
Sample&Hold kretsen behöver bara  $\approx 4$   $\mu$ s för att ladda in spänning. Hold-tiden är typiskt 2s. Spänningarna tillåts ligga  $\pm 10$  V.

Ann

Quad switchen släpper igenom spänning på ingången, om kontroll-ingången är låg. Därför är avkodaren försedd med inverterare på sina utgångar.

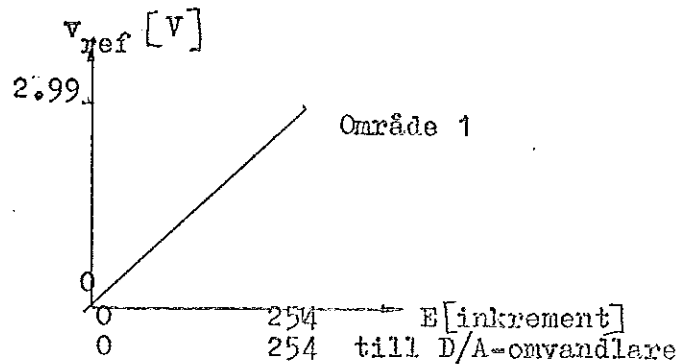
Summering av metoden

Det är datorns kännedom om eftersläpningen i systemet som banar väg för möjligheterna att dela resurser mellan flera axlar. Det skissade drivdonsinterfacet är billigt och komponenterna rymms på ungefär ett halvt kort.

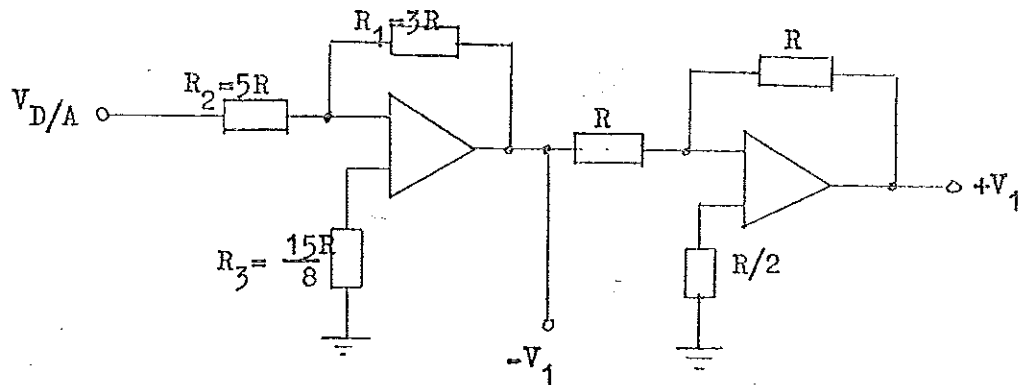


Figur 22 Drivdonsinterface i blockschemaform

Operationsförstärkarkoppling, område 1.



Vid alla nedanstående resistansberäkningar förutsättes att 255 i D/A-omvandlaren svarar mot att denna levererar utspänningen +5 V.



Figur 22a OP-förstärkar koppling för område 1.

Motståndet  $R_1$  väljes godtyckligt till  $3R$ .

Samband mellan spänningar:  $V_1 = \frac{R_1}{R_2} \cdot V_{D/A}$

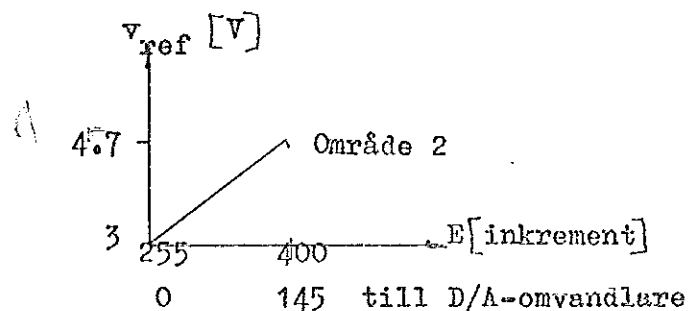
Från övre områdesgränsen fås,

$$2.99V = \frac{3R \cdot 254 \cdot 5V}{R_2 \cdot 255}$$

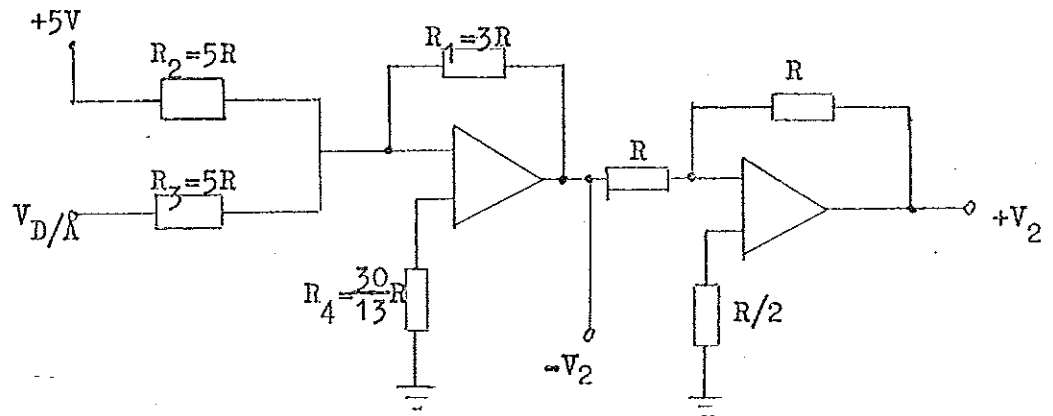
Varför  $R_2$  väljes till  $5R$

För att minimera offset-fel väljes  $R_3 = R_2 // 3R = \frac{15}{8} R$

Operationsförstärkarkoppling, område 2.



Egentligen är övre områdesgränsen 399 inkrement, men man får fram rätt resistansvärden, med enklare räkningar, om man tar 400 inkrement istället.



Figur 22b OP-förstärkar koppling för område 2.

$$V_2 = \frac{R}{R_2} \cdot 5V + \frac{R_1}{R_3}$$

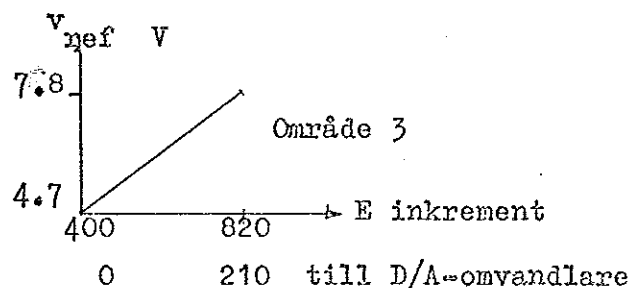
Från områdesgränserna fås:

$$3V = \frac{3R}{R_2} \cdot 5V \quad \text{varför } R_2 \text{ väljes till } 5R.$$

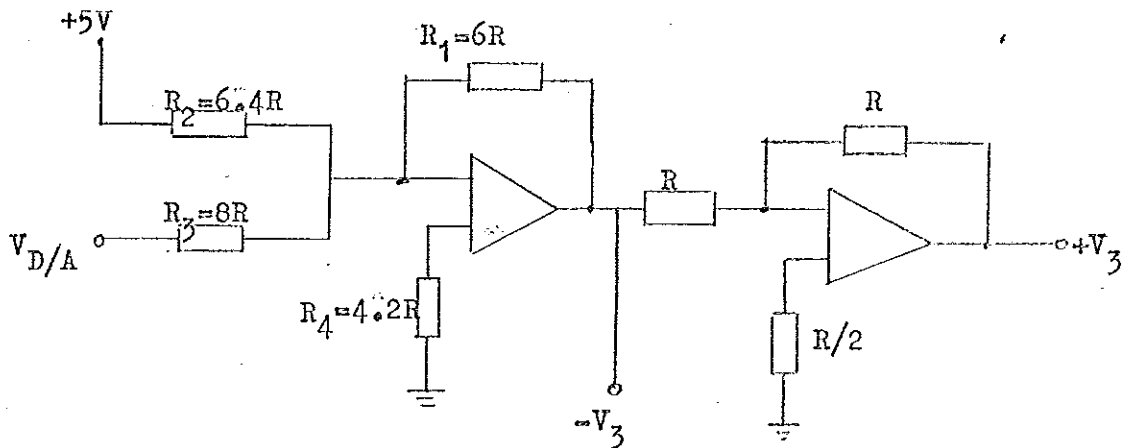
$$4.7V = \frac{3R}{5R} \cdot 5V + \frac{3R}{R_3} \cdot \frac{145}{255} \cdot 5V \quad R_3 \text{ väljes till } 5R.$$

För att minimera offseten väljes  $R_4$  till  $\frac{30}{13} R$

### Operationsförstärkarkoppling, område 3.



Eftersläpningen inom detta område varierar med 420 inkrement. Någon 1:1 skalning till bitarna i D/A-omvandlaren går inte att göra. Vi befinner oss i det område där upplösningen på hastighetsreferensen måste minskas. Lösningen på problemet består i att datorn efter att ha fastställt att eftersläpningen ligger i detta området, subtraherar undre områdesgränsen och dividerar resten med två. Sedan väljer man sina resistanser på ett sådant sätt att, rätt spänning levereras till drivdonet.



Figur 22o OP-förstärkar koppling för område 3.

$$V_3 = \frac{R_1}{R_2} \cdot 5V + \frac{R_1}{R_3} \cdot V_{D/A}$$

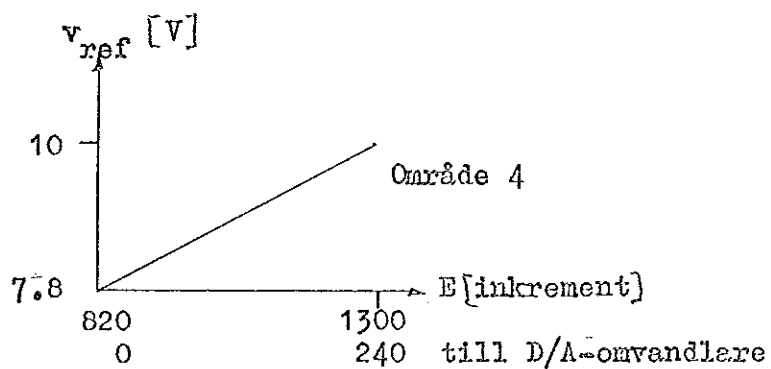
Från områdesgränserna fås:

$$4.7V = \frac{6R}{R_2} \cdot 5V \quad R_2 \text{ väljes till } 6.4R$$

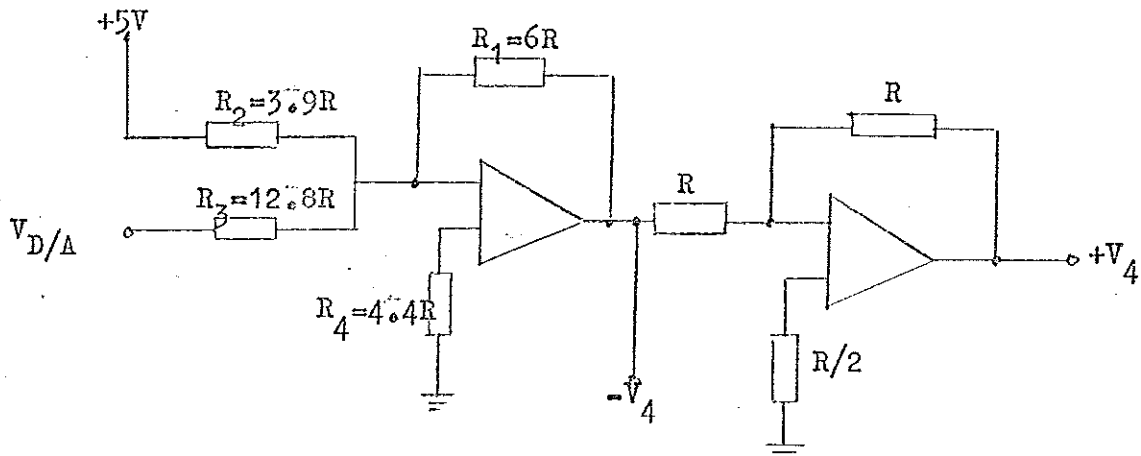
$$7.8V = \frac{6R}{6.4R} \cdot 5V + \frac{6R \cdot 210}{R_3 \cdot 255} \cdot 5V \quad R_3 \text{ väljes till } 8R$$

För att minimera offseten väljes  $R_4$  till  $4.2R$

#### Operationsförstärkarkoppling, område 4.



Man gör precis som för område 3, för att få resistansvärdena, det vill säga först division med två i datorn för att ändra upplösningen.



Figur 22d OP-förstärkar koppling för område 4.

$$V_4 = \frac{R_1}{R_2} 5V + \frac{R_1}{R_3} V_{D/A}$$

Från områdesgränserna fås:

$$7.8V = \frac{6R}{R_2} 5V \quad R_2 \text{ väljes till } 3.9R$$

$$10V = \frac{6R}{3.9R} 5V + \frac{6R}{R_3} \frac{240}{255} 5V \quad R_3 \text{ väljes till } 12.8R$$

För att minimera offseten väljes  $R_4$  till  $4.4R$ .

Anm

Man kan tänka sig andra upplösningar av hastighetsreferensen i områdena 3 och 4. Upplösningen behöver inte vara lika i de bägge områdena, som ovan.

Man kan till exempel dividera med 8 i stället för med två i område 4, och på så sätt försämra upplösningen en faktor 4. Resistansen  $R_1$  i figur 22d skall då bytas mot  $24R$ .



4

Konstruktionslösningar för mikro datorbaserad positionering av en axel

4.1

Resolver till digitalomvandling, dator interface

#### Inledning

Den fasanaloga metoden enligt figur 10, sidan 13, ligger till grund för konstruktionen.

De signaler vars inbördes faslägen jämföres är:

Den filtrerade och nollgenomgångsdetekterade matnings-  
spänningen SINREF.

Den filtrerade och nollgenomgångsdetekterade rotorspänningen  
POSPHASE1.

För filtrering och nollgenomgångsdetektering användes  
kretsen i figur 6, sidan 9.

Jämförelsen mellan signalerna göres på så sätt att, SINREF:s  
0-1 övergång får starta en räknare, medan POSPHASE1:s 0-1  
övergång stoppar räkningen. Räknarens innehåll kommer därvid  
att utgöra ett mått på fasskillnaden inom 360 grader.

#### Anm

Axelns hastighet begränsas till 17.6 varv/s.

#### Kretsar för klockpulsgenerering.

För att erhålla en konsekvent synkronisering i metoden,  
bildas 3 klockpulser på 400 kHz, tidsmässigt icke samman-  
fallande. Dessutom bildas en fjärde klockpuls, också den  
på 400 kHz, för att leverera räknepulser till en 256-räknare.

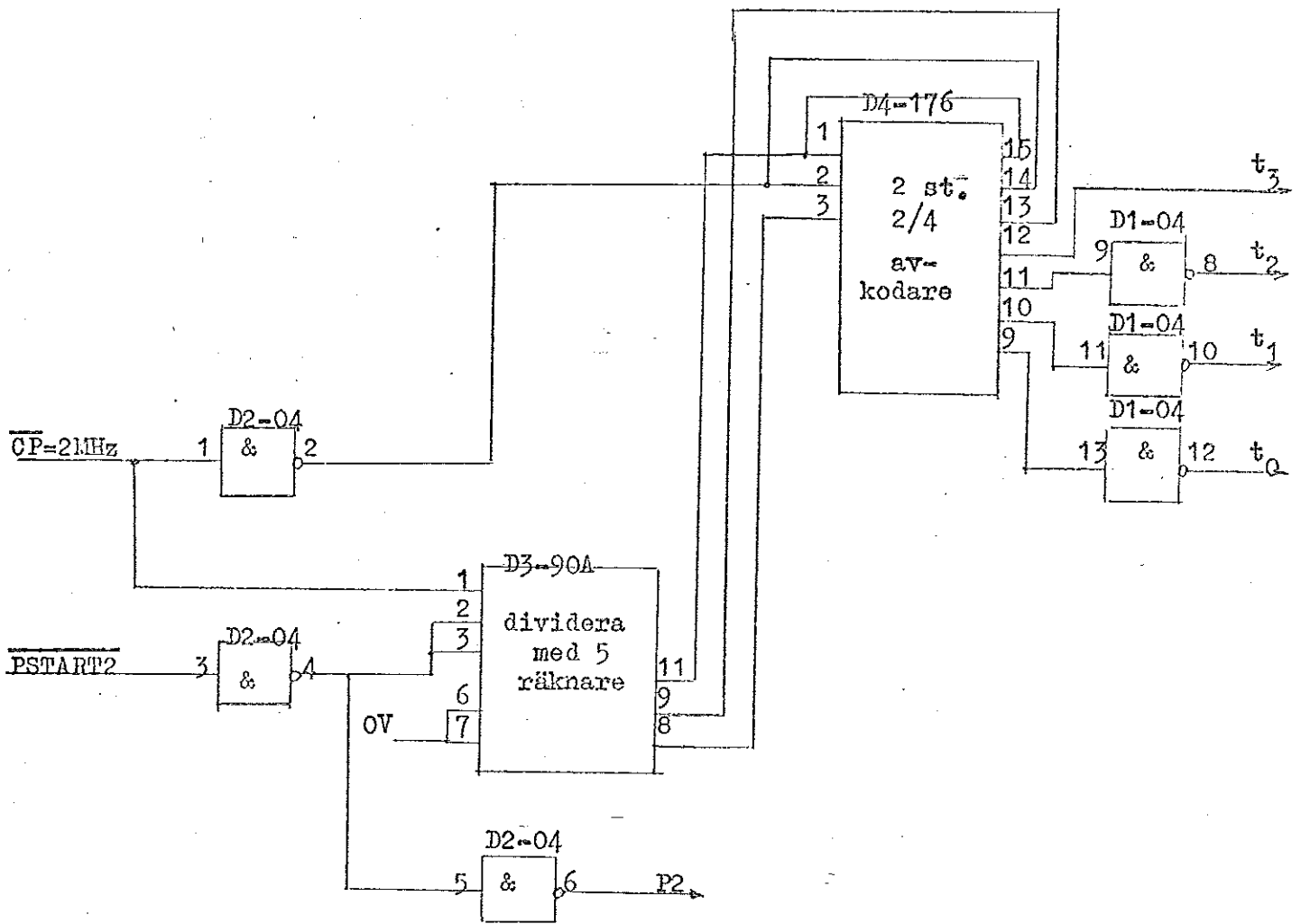
#### Anm

Att klockpulsfrekvensen valts till just 400 kHz, beror på att  
om SINREF och POSPHASE1 ligger 360 grader fasförskjutna  
relativt varandra, räknaren kommer att innehålla 200. Detta  
är den upplösning av 360 grader som vi vill ha.

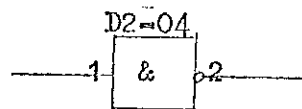
Figur 23 visar de kretsar som användes för klockpuls  
genereringen. Figuren återfinnes på sidan 32.

Klockpulsgenereringen går till så att:

Signalen  $\overline{CP}$  på 2MHz delas ned i en dividera med 5 räknare.  
Denna räknare levererar adresser till en avkodare som på  
sina utgångar ger klockpulserna. Signalen PSTART2 användes  
för att se till så att räkningen börjar från tillståndet  
000. PSTART2 blir sann under en kort tid vid spännings-  
tillslag. FAN-OUT på PSTART2 ökas genom två gångers inver-  
tering. Den erhållna signalen kallas P2. För att få spikfria  
utgångar från avkodaren användes CP som enable-signal.



Figur 23 Kretsar för klockpulsgenerering.



Förklaring till ritteknik.

D2: IC-kapsel nummer 2.

O4: Kapsel typen är 7404. 74 är underförstått.

1 och 2 : Pinnar på IC-kapsel.

Fasjömfönare

I detta block användes klockpulserna  $t_0$ ,  $t_1$  och  $t_2$  för att tidsmässigt kontrollera det förlopp enligt vilket signalerna COUNT ENABLE, COUNT DISABLE, RESET och KLOCK bildas. Denna del av resolver till digital omvandlaren visas i figur 24, sidan 34.

Kopplingens funktion kan förklaras genom att man betraktar två fall av fasskillnad mellan SINREF och POSPHASE1. (Axeln tänkes stå stilla)

Fall 1. Fasskillnad +1 inkrement.

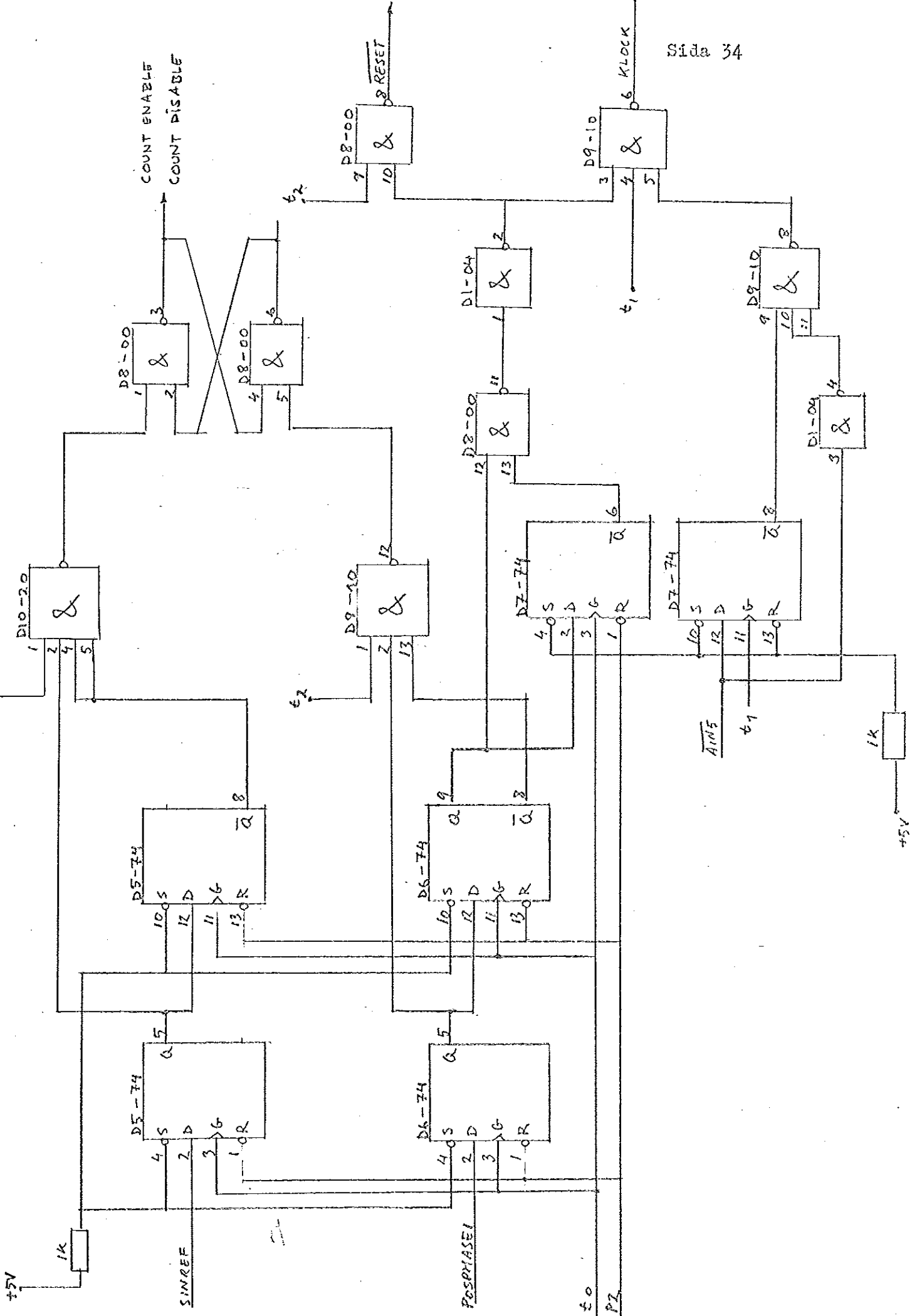
Betrakta de bägge övre D-vipporna.

Denna vippkoppling fungerar så att när SINREF har gjort en 0-1 övergång och  $t_0$  klockat in en etta i den första vippan, så ligger en puls på SR-vippans set-ingång och väntar på  $t_1$ . När  $t_1$  kommer, ställer SR-vippan in sig så att dess Q utgång blir COUNT ENABLE. På  $t_2$  händer ingenting i kopplingen.  $t_3$  levererar en räknepuls till räknaren som kommer att få innehållet 1. Nästa  $t_0$  kommer sedan att skifta in ettan från SINREF till den andra D-vippan. Set-ingången på SR-vippan kommer att blockeras. Under den tid av 2 kHz intervallet som återstår kommer SINREF inte att påverka några signaler i kopplingen.

De bägge undre D-vipporna som hör till POSPHASE1 fungerar likadant som de övre. När POSPHASE1 gjort sin 0-1 övergång kommer en puls att ligga på SR-vippans reset ingång och vänta på  $t_2$ . När  $t_2$  kommer förändras SR-vippans Q utgång till COUNT DISABLE. Räknepulsen vid  $t_3$  kommer då inte att kunna uppdatera räknaren. Vid nästa  $t_0$  kommer ettan från POSPHASE1 att skiftas igenom den andra D-vippan. Den så ettsatta Q-utgången på denna vippa kommer att ge KLOCK via  $t_1$  och RESET vid  $t_2$ . En tredje vippa blir sedan ettsatt vid åter nästa  $t_0$  och blockerar signalerna KLOCK och RESET. En etta kommer att ligga i ett buffertregister och räknaren är nollställd. Inga förändringar noteras längre i nätet, men samma procedur upprepas nästa gång SINREF gör en 0-1 övergång.

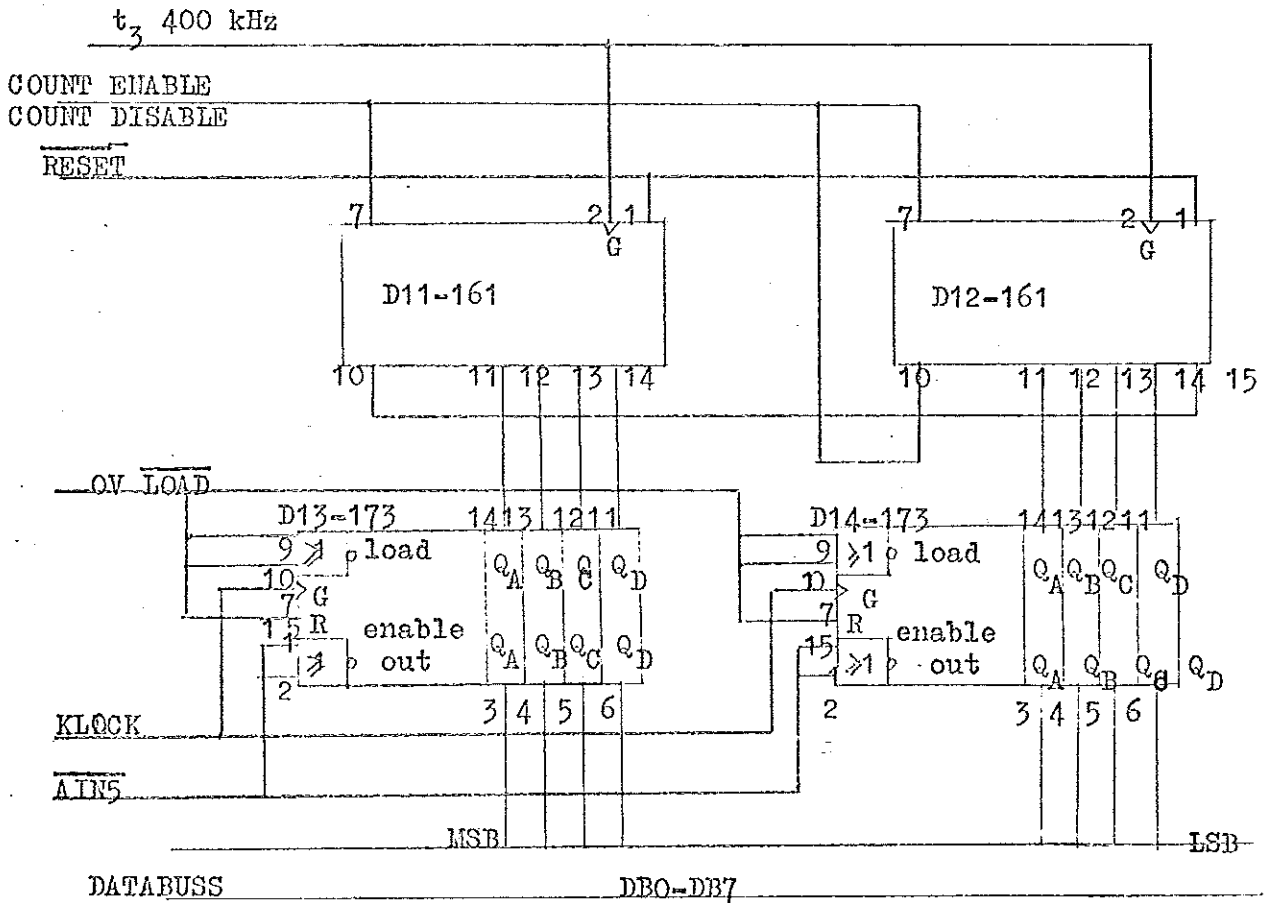
Fall 2. Fasskillnad 0 inkrement.

Den set-puls som släpps igenom vid  $t_1$  och ger COUNT ENABLE åtföljs direkt (vid  $t_2$ ) av en reset-puls som ger COUNT DISABLE. Räknaren kommer inte att uppdateras och dess innehåll blir 0. Signalerna KLOCK och RESET bildas precis som i Fall 1.



Figur 24 Fasjämförare

Räknare och interface till dator



Figur 25 Räknare-buffertregister

Buffertregistret har konstant  $\overline{LOAD}$ -signal genom OV. Inläsning sker på 0-1 övergången hos KLOCK.

Anm

Kopplingen med den nedre D-vippan i figur 24 ser till att ett nytt värde på fasskillnaden läses in till buffertregistret, om inte AIN5 (ingång 5 användes) är låg när 0-1 övergången hos  $t_1$  kommer. Buffertregistret behöver nämligen en viss minsta tid för att laddas rätt.



Här kommer ett nytt värde att läsas ned till buffertregister

Här kommer inte ett nytt värde att läsas ned

4.2

## Program för lägesreglering

Inledning

Det program som tagits fram sköter lägesregleringen av en axel vid körning i hastigheter mindre än 17.6 varv/s. Denna begränsning i hastighet gör att hårdvaran för drivdonsinterfacet blir billigare och mjukvaran enklare. Om man vill lägesreglera en axel men tillåta högre hastigheter har man två möjligheter.

Alternativ 1

Man tillåter sämre upplösning på hastighetsreferensen vid höga hastigheter.

Man kan då förfara enligt vad jag redogort för i kapitel 3 avsnitt 2. Uteslut bara vad som gäller Sample&Hold kretsarna och gå igenom lägesregleringsprogrammet en gång.

Alternativ 2

Man tillåter inte sämre upplösning i hastighetsreferensen vid höga hastigheter.

Antingen använder man en D/A-omvandlare som ger en tillräcklig upplösning över hela arbetsområdet, eller också gör man en uppdelning av parabelbildaren i ett tillräckligt antal områden, bestämmer inom vilket område som eftersläpningen ligger och lägger ut till D/A-omvandlaren, samt ser till att rätt OP-förstärkar koppling användes.

Den mikrodatorbaserade positioneringen gör det möjligt att lägga en del av systemets hårdvara i program. De funktioner som finns på dagens axelkort går lätt att överföra till mjukvara. Som ett exempel skall jag skissahur roboten skulle kunna synkroniseras.

Synkronisering

Precis som i dagens system kräver man att roboten skall synkroniseras innan något användarprogram tillåts exekvera. Oberoende av i vilket läge robotaxlarna befinner sig, sluter operatören en synkroniseringsswitch, som aktiverar ett speciellt programblock för synkronisering. I detta block modifieras systemets nödstoppfunktion, så att alla axlar körs till sina ändlägen. Efter det att nödstopp registrerats på alla axlar, undersöker programmet om resolverarna befinner sig i riktiga lägen (ett godtyckligt valt nolläge som kan markeras med ett synligt hack för att underlätta service på injustering av resolver). En kontakt skall då vara till, om inte så måste resolverarna justeras. Eftersom den mikrodatorbaserade positioneringen innebär att datorn känner alla axlarnas är-värden, kan axlarna köras såg 50 resolvervarv och denna punkt sedan väljas som synkroniseringspunkt.

Flödesschema över lägesregleringsprogram

Figur 26, sidan 38 visar det totala flödesschemat, medan figur 27, sidan 39 och figur 28, sidan 40 är uppförstoringar av vad som i figur 26 kallas resolverblock respektive vrefblock.

I programmet användes 8 dataswitchar, som kan läsas in till datorns ackumulator med INPO operationen.

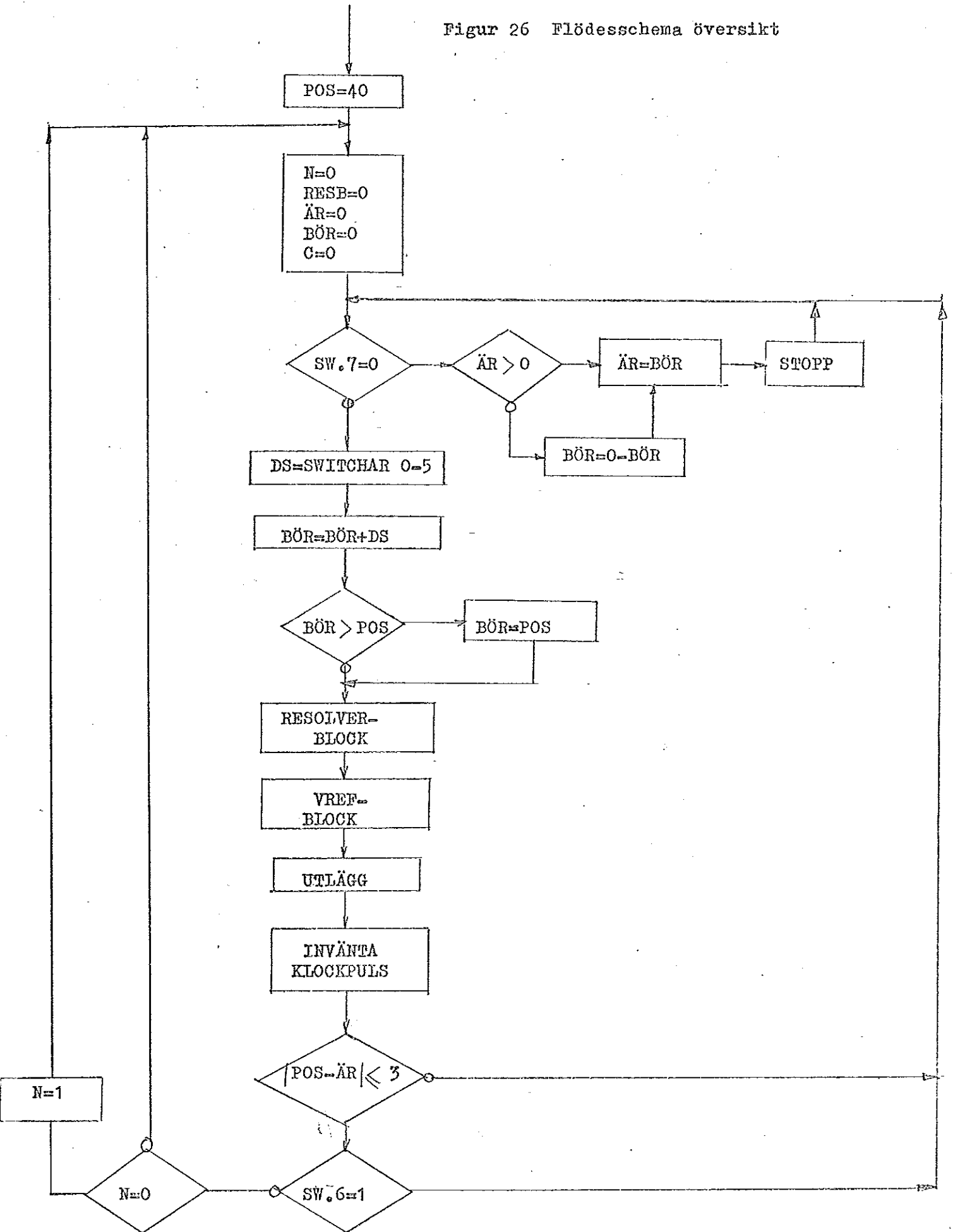
switch 7: Nödstopp om switchen=0, genom att hastighetsreferensen sättes till 0V.

switch 6: Om switchen=0 sker positionering tillbaka till den punkt som axeln befann sig i då programmet startades, en oscillerande rörelse utföres alltså. Annars har man vad som i dagens system kallas finpositionering med väntetid.

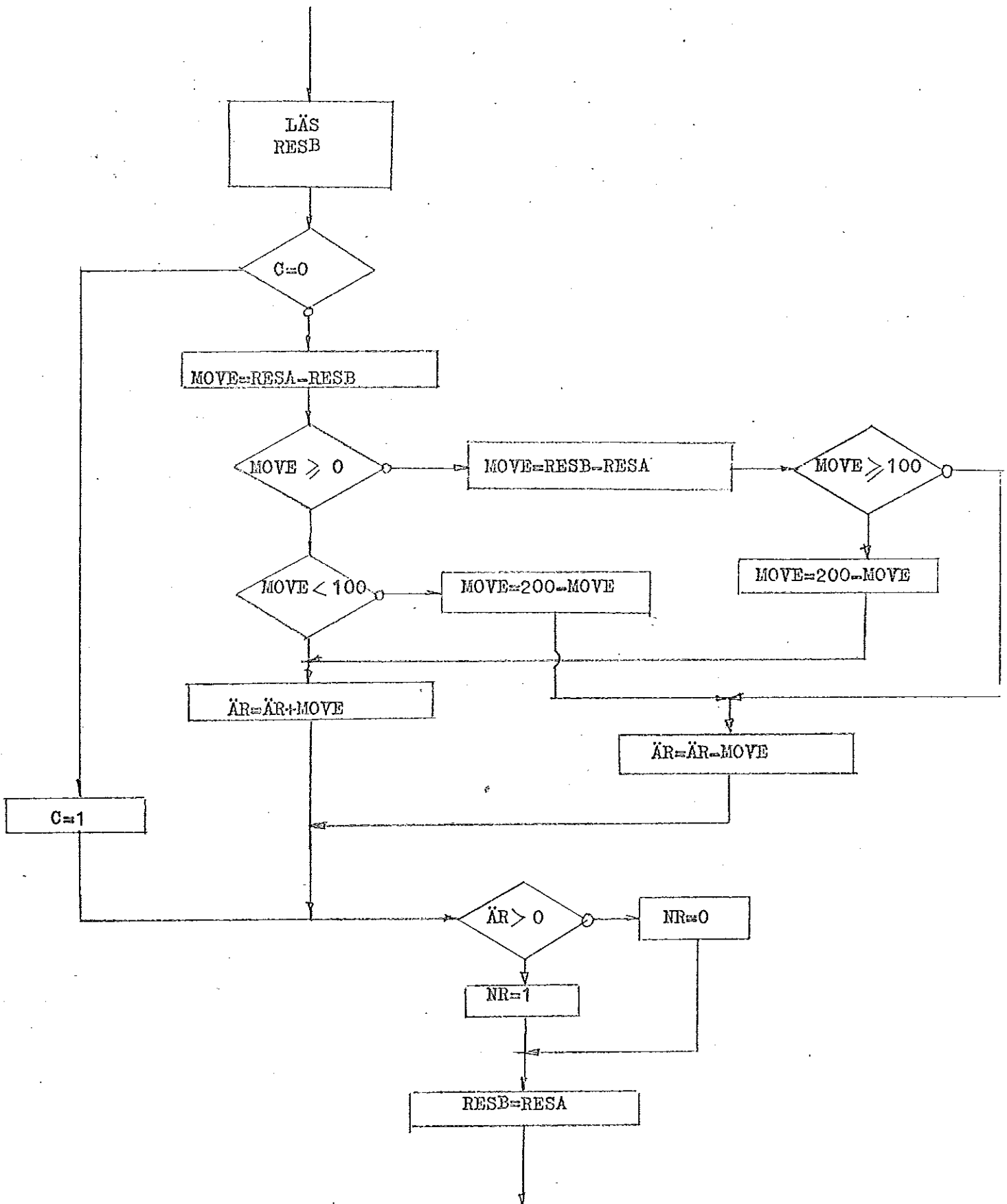
switcharna 0-5: Anger direkt hastighetsreferensen i robotinkrement. Tillåtes högst vara 0.176.200 inkrement/10ms.

Programmet startas på switch 7 och får axeln att utföra en oscillerande rörelse mellan två finpunkter definierade från program adressen POS. Den programmerade hastigheten läses från switcharna 0-5. Axelns rörelse mellan två samplingstillfällen beräknas i RESOLVERBLOCK. Konventionell lägesreglering utföres i VREFBLOCK. För övrigt se kommentarer i assembler program.

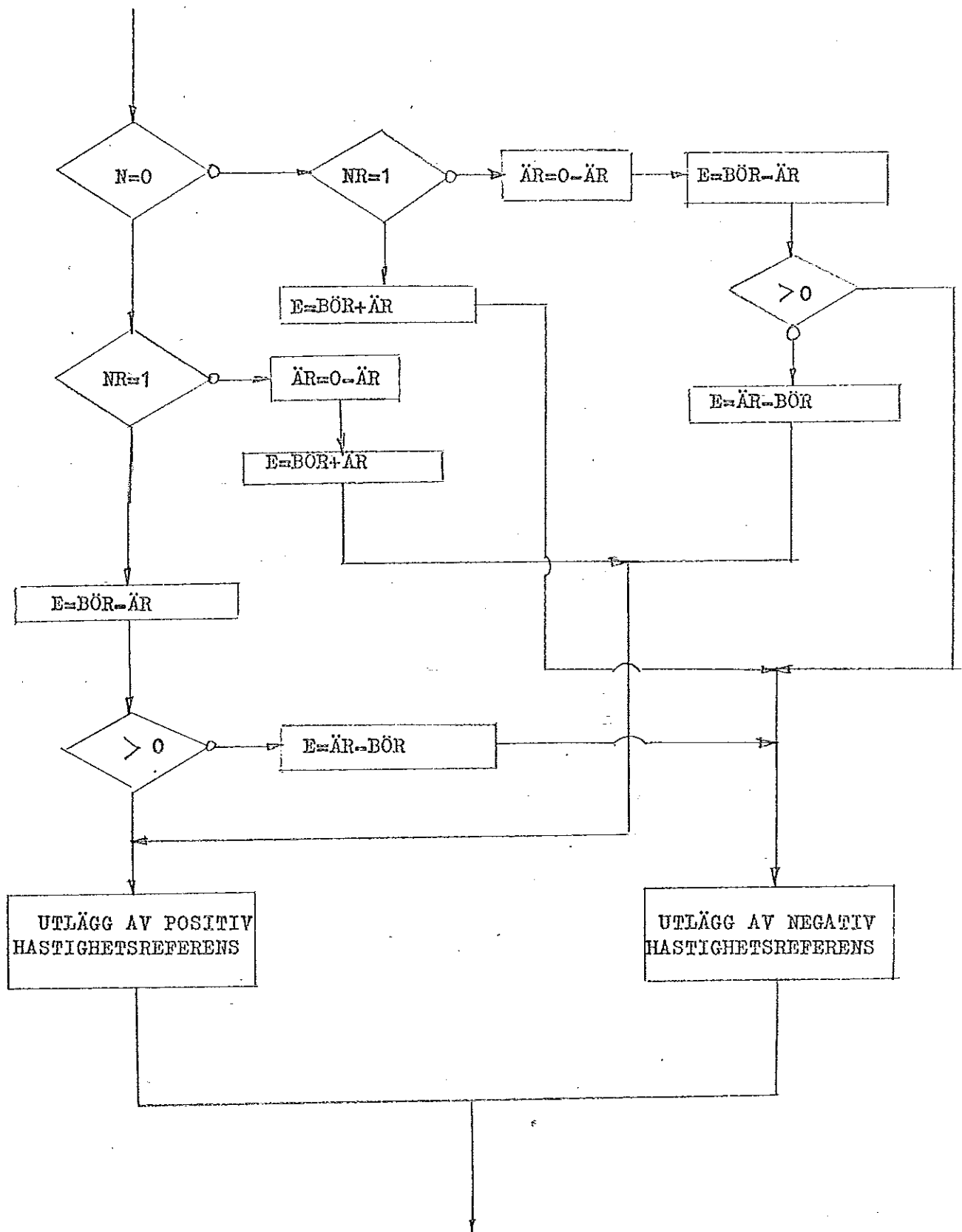
Figur 26 Flödesschema översikt







Figur 27 Resolverblock för beräkning av utförd förflyttning mellan två samplingstillfällen.



Figur 28 Vrefblock -slutet i block för utlägg.

Assembler program för lägesreglering av en axel

Programmet är skrivet i INTEL 8008 assembler och sköter lägesregleringen på i medeltal 2.5 ms.

LABEL	OPERATION	OPERAND	COMMENTS
	ORG	40 Q	
	RET		Återhopp efter interrupt
/	ORG	2000 Q	Program startadress
	SHL	POS2	As(POS2)
	LMI	0	POS2=0
	INL		As(POS1)
	LMI	8	POS1=8
APOS	LLI	1	As(N)
	LMI	0	N=0
BPOS	LLI	2	As(BR2)
	LMI	0	BR2=0
	INL		As(BR1)
	LMI	0	BR1=0
	LLI	9	As(AR2)
	LMI	0	AR2=0
	INL		As(AR1)
	LMI	0	AR1=0
	INL	7	As(C)
	LMI	0	C=0
	LLI	7	As(RESB)
	LMI	0	RESB=0
			Avläsning av data switch 7. Switchen hindrar programmet att startas, eller ger stopp om exekvering pågår.
POS3	INP	0	
	NDI	377 Q	
	JTS	RUN	
	LAI	377 Q	Nödstopp: v <sub>ref</sub> =0
	OUT	34 Q	
	OUT	35 Q	
	LLI	2	As(BR2)
	LBM		BR2 till Reg B
	INL		As(BR1)
	LCM		BR1 till Reg C
	LLI	10	As(AR1)
	LAM		AR1 till Reg A
	NDI	377 Q	
	JTS	TWOK	Hoppa om ÅR < 0 för att bilda tvåkomplementet
	LAC		BR1 till Reg A
	IMA		AR1=BR1
	LAB		BR2 till Reg A
	DCL		As(AR2)
	IMA		AR2=BR2
	JMP	POS3	Testa switch 7 igen

LABEL	OPERATION	OPERAND	COMMENTS
TWOK	LAI	0	2-komplement av BÖR
	SUB		0-BR2
	DCL		As(BR2)
	LMA		AR2=BR2
	LAI	0	0 till Reg A
	SBC		0-BR1-borrow
	INL		As(BR1)
	LMA		AR1=BR1
	JMP	POS3	Testa om klart för körning
RUN	NDI	77 Q	
	LLI	2	As(BR2)
	ADM		DS+BR2
	LMA		Reg A till BR2
	JFC	ITALOM	Hoppa om BÖR i talområdet
	LAI	1	Inkrementera MSB av BÖR
	INL		As(BR1)
	ADM		BR1+1
	LMA		Reg A till BR1
	JFC	ITALOM	Om BÖR utanför talområdet, sättes BR1=BR2=377 Q
	LMI	377 Q	As(BR2)
	DCL		BR2=377 Q
	LMI	377 Q	Test om BÖR > POS <sub>1</sub> i så fall sättes BÖR=POS
ITALOM	LLI	2	As(BR2)
	LBM		BR2 till Reg B
	INL		As(BR1)
	ICM		BR1 till Reg C
	INL		As(POS2)
	LAM		POS2 till Reg A
	INL		As(POS1)
	LEM		POS1 till Reg E
	SUB		POS2-BR2
	LAE		POS1 till Reg A
	SBC		POS1-BR1-borrow
	JFC	POS4	Klart om POS > BÖR
	DCL		As(POS2)
	LAM		POS2 till Reg A
	LLI	2	As(BR2)
	LMA		BR2=POS2
	INL		As(BR1)
	LAE		POS1 till Reg A
	LMA		BR1=POS1
			Resolverblock: beräkning av är- värdet på läget, test av om slingan genomlöpes en första gång.
POS4	INP	5	Läs buffertregister =RESA
	LLI	6	Lägg i RESA
	LMA		

LABEL	OPERATION	OPERAND	COMMENTS
	LLI	11	C=0 innebär att programmet
	LAM		genomlöpes en första gång vid
	NDI	377 Q	positionering i viss riktning
	JFZ	CONE	
	LAI	1	C var 0. C sättes =1
	LMA		
	JMP	POS5	
CONE	LLI	6	Bilda MOVE
	LAM		RESA till Reg A
	INL		As(RESB)
	SUM		RESA-RESB
	JTC	TEST1	MOVE < 0, hoppa
	LBA		MOVE till Reg B
	SUI	100	MOVE-100
	JFC	TEST2	MOVE > 100, hoppa
PLUS	LLI	9	Här göres AR=AR+MOVE
	LAM		AR2 till Reg A
	ADB		AR2+MOVE
	LMA		AR2=AR2+MOVE
	JFC	POS5	Klart om ej carry till AR1
	INL		As(AR1)
	LAI	1	1 till Reg A
	ADM		AR1+1
	LMA		AR1=AR1+1
	JMP	POS5	Klart
TEST1	LAM		RESB till Reg A
	DCL		As(RESA)
	SUM		RESB-RESA
	LBA		MOVE till Reg B
	SUI	100	MOVE-100
	JTC	MINUS	MOVE < 100, hoppa
	LAI	200	200 till Reg A
	SUB		200-MOVE
	LBA		MOVE till Reg B
	JMP	PLUS	Hoppa till AR=AR+MOVE
TEST2	LAI	200	200 till Reg A
	SUB		200-MOVE
	LBA		MOVE till Reg B
MINUS	LLI	9	Här göres AR=AR-MOVE
	LAM		AR2 till Reg A
	SUB		AR2-MOVE
	LMA		AR2=AR2-MOVE
	INL		As(AR1)
	LAM		AR1 till Reg A
	SBT	0	AR1-borrow
	LMA		AR1=AR1-borrow

LABEL	OPERATION	OPERAND	COMMENTS
POS5	LLI	10	As(AR1)
	LAM		AR1 till Reg A
	NDI	377 Q	
	JFS	OK	Hoppa om $\text{ÄR} > 0$
	LLI	0	As(NR), för tecken på $\text{ÄR}$
	LMI	0	NR=0
	JMP	UK	
OK	LEI	0	As(NR)
	LMI	1	NR=1
UK	LLI	6	As(RESA)
	LAM		RESA till Reg A
	INL		As(RESB)
	LMA		RESB=RESA
			VREF-block. Hastighetsreferensen bestäms av eftersläpningen.
	LLI	2	As(BR2)
	IBM		BR2 till Reg B
	INL		As(BR1)
	ICM		BR1 till Reg C
	LLI	9	As(AR2)
	LDM		AR2 till Reg D
	INL		As(AR1)
	LEM		AR1 till Reg E
	LEI	1	As(N). N=1 om negativ riktning
	LAM		N till Reg A
	NDI	377 Q	
	JFZ	NEG	Hoppa om negativ positioneringsriktning
	DCL		As(NR)
	LAM		NR till Reg A
	NDI	377 Q	
	JTZ	MOD1	Hoppa om $\text{ÄR}$ negativt
	LAB		BR2 till Reg A
	SUD		BR2-AR2
	LEA		LSB(E) till Reg L
	LAC		BR1 till Reg A
	SBE		BR1-AR1-borrow
	LHA		MSB(E) till Reg H
	JFC	TABP	Till TABP om positiv eftersläpning (normalfallet)
	LAD		AR2 till Reg A
	SUB		AR2-BR2
	LEA		LSB(E) till Reg L
	LAE		AR1 till Reg A
	SBC		AR1-BR1-borrow
	LHA		MSB(E) till Reg H
	JMP	TABN	TABN för att korrigera för översläng

LABEL	OPERATION	OPERAND	COMMENTS
MOD1	LAI	0	Tvåkomplement av AR
	SUD		0-AR2
	LDA		AR2 till Reg D
	LAI	0	0 till Reg A
	SBE		0-AR1-borrow
	LEA		AR1 till Reg E
	LAB		E=BÖR+ÄR
	ADD		BR2+AR2
	LLA		LSB(E) till Reg L
	LAC		BR1 till Reg A
	ACE		BR1+AR1+carry
	LHA		MSB(E) till Reg H
	JMP	TABP	
NEG	DCL		As(NR)
	LAM		NR till Reg A
	NDI	377 Q	
	JFZ	MOD3	Hoppa om ÄR positiv
	LAI	0	Tvåkomplement av ÄR
	SUD		0-AR2
	LDA		AR2 till Reg D
	LAI	0	0 till Reg A
	SBE		0-AR1-borrow
	LEA		AR1 till Reg E
	LAB		BR2 till Reg A
	SUD		BR2-AR2
	LLA		LSB(E) till Reg L
LAC		BR1 till Reg A	
SBE		BR1-AR1-borrow	
LHA		MSB(E) till Reg H	
JFC	TABN		Till TABN om negativ efter- släpning (normalfallet)
LAD		AR2 till Reg A	
SUB		AR2-BR2	
LLA		LSB(E) till Reg L	
LAE		AR1 till Reg A	
SBC		AR1-BR1-borrow	
LHA		MSB(E) till Reg H	
JMP	TABP		Korrigerar för översläng
MOD3	LAB		E=BÖR+ÄR
	ADD		BR2+AR2
	LLA		LSB(E) till Reg L
	LAC		BR1 till Reg A
	ACE		BR1+AR1+carry
	LHA		MSB(E) till Reg H
	JMP	TABN	

LABEL	OPERATION	OPERAND	COMMENTS
TABP	LDI	377 Q	Reg D märker positiv VREF
	JMP	KORR	Maska ut bitarna till D/A- omvandlaren
TABN	LDI	177 Q	Reg D märker negativ VREF
KORR	LAL		LSB(E) till Reg A
	NDI	370 Q	Maska ut de 5 MSB
	RRC		
	RRC		Läge 000-----
	LCA		
	LAH		MSB(E) till Reg A
	RRC		
	RRC		
	RRC		Läge 00-00000
	ADC		Läge 00-----
	LHA		MSB(VREF) till Reg H
	LAL		LSB(E) till Reg A
	NDI	7	Läge 00000----
	LLA		LSB(VREF) till Reg L
	LAH		
	XRI	377 Q	Invertera
	NDD		För rätt teckenbit
	OUT	34 Q	MSB(VREF)+teckenbit till OUT 34
	LAL		
	XRI	377 Q	
	OUT	35 Q	LSB(VREF) till OUT 35 Q
			Gör enable interrupt
	LAI	2	
	OUT	10 Q	
	HLT		CPU väntar på klockpuls
			Test av om axeln är i position och slutning av slingan
	SHL	POS2	POS2 till Reg B
	LBM		As(POS1)
	INL		POS1 till Reg C
	ICM		As(AR2)
	ILL	9	AR2 till Reg D
	IDM		As(AR1)
	INL		AR1 till Reg E
	TEM		Bilda  POS-ÄR
	ILL	0	As(NR)
	LAM		NR till Reg A
	NDI	377 Q	
	JTZ	NABS	Hoppa om ÄR mindre än 0
TWICE	LAB		POS2 till Reg A
	SUD		POS2-AR2
	LDA		LSB(POS-ÄR) till Reg D
	LAC		POS1 till Reg A
	SBE		POS1-AR1-borrow
	JFC	OKY	Hoppa om POS > ÄR



LABELL	OPERATION	OPERAND	COMMENTS
	LJI	9	As(AR2)
	LAM		AR2 till Reg A
	SUB		AR2-POS2
	LDA		LSB(AR&POS) till Reg D
	LAE		AR1 till Reg A
	SBC		AR1-POS1-borrow
OKY	LEA		Testa om $ \text{POS}-\text{AR}  \leq 3$
	LAD		MSB(abs.-belopp) till Reg E
	SUI	3	LSB(abs.-belopp) till Reg A
	LAE		LSB-3
	SBI	0	MSB till Reg A
	JFC	POS3	MSB-borrow
			Hoppa om större än 3
NABS	LAI	0	Tvåkomplementet av AR
	SUD		$0-\text{AR2}$
	LDA		AR2 till Reg D
	LAI	0	0 till Reg A
	SBE		$0-\text{AR1-borrow}$
	LEA		AR1 till Reg E
	JMP	TWICE	Hoppa till testen $ \text{POS}-\text{AR}  \leq 3$
SW6	IHP	0	Läs dataswitcharna
	NDI	100 Q	
	JFZ	POS3	Finpositionering med väntetid om switch 6=1
	LBI	11	As(N)
	LAM		N till Reg A
	NDI	377 Q	
	JFZ	APOS	Byt till positiv riktning
	IMI	1	
	JMP	BPOS	Byt till negativ riktning

## DATALISTA

NR	ORG	DEF	COMMENTS
	4000 Q		Datalistans begynnelseadress
NR	DEF	0	Testvariabel
N	DEF	0	Definierar förflyttningsriktning
BR2	DEF	0	LSB av börvärde
BR1	DEF	0	MSB av börvärde
POS2	DEF	0	LSB av slutpositioneringspunkt
POS1	DEF	0	MSB av slutpositioneringspunkt
RESA	DEF	0	Resolvervärde vid t.
RESB	DEF	0	Resolvervärde vid t-samplingstid
MOVE	DEF	0	Utförd förflyttning per sample
AR2	DEF	0	LSB av ärvärde
AR1	DEF	0	MSB av ärvärde
C	DEF	0	Testvariabel
	END		Slut assembler program

## 4.3

## Drivdonsinterface

Drivdonsinterfacet kan delas upp i 4 delar.<sup>43</sup>

- A.<sup>43</sup> Buffertregister koppling.
- B.<sup>43</sup> D/A-omvandlar koppling.
- C.<sup>43</sup> OP-förstärkar koppling.
- D.<sup>43</sup> Switch koppling.

Delarna A, B och C visas i figur 29 på sidan 43<sup>43</sup>. Del D finns på sidan 51 i figur 31.

#### A.<sup>43</sup> Buffertregister koppling.

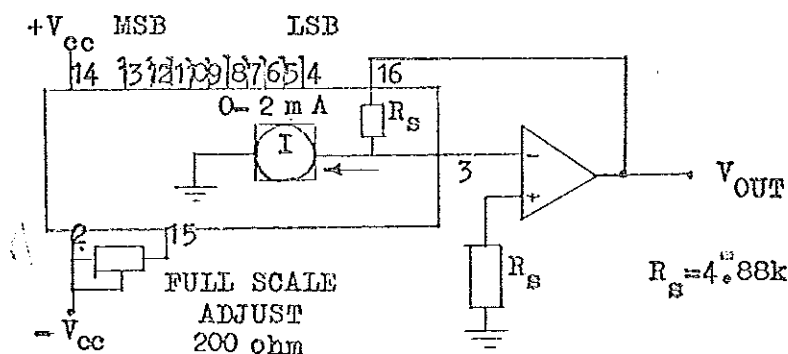
Med signalen  $\overline{AOUT34Q}=0$  klockas databussens innehåll in till ett buffertregister på signalen 0.73 MHzC:s framkant. Detta buffertregister består av två 4 bitars tristabila vipper.

Ett annat buffertregister användes för att med signalen  $\overline{AOUT35Q}=0$  på 0.73 MHzC:s framkant, ta hand om de tre minst signifikanta bitarna av databussen. Detta buffertregister består av en 4 bitars tristabil vippra, med bara tre av bitarna anslutna till databussen.

Buffertregisternas utgångar har konstant ENABLE signal genom OV. Alla utgångarna utom den som svarar mot databussens mest signifikanta bit DB7 är anslutna till D/A-omvandlaren. DB7 biten användes av switch kopplingen som switch villkor (teckenbit).

#### B.<sup>43</sup> D/A-omvandlar koppling.

Den använda D/A-omvandlaren har konstant ström karakteristik på utgången för ett givet ingångsord. Spänningstillämpning är implementerad genom att en yttre OP-förstärkare och ett återkopplingsmotstånd i kapseln används. Då D/A-omvandlaren är kopplad som på drivdonsinterfacet säges den arbeta i UNIPOLAR MODE. Figur 30 visar kopplingen som finns i figur 29 isolerad från övrig elektronik.



Figur 30 D/A-omvandlare i UNIPOLAR MODE

$V_{OUT}$ : Kopplingen kan leverera spänningar mellan 0 och 10V.

FULL SCALE ADJUST: Med hjälp av potentiometern på 200 ohm justeras fullt utslag till 10V.

Kodning: Omvandlaren använder inverterad binär logisk kod. Detta innebär att en låg logisk ingångs nivå aktiverar biten. Därför inverteras eftersläpningen i mjukvaran innan utlägg sker.

Skalningen mellan robotinkrement och D/A-omvandlar bitar är gjord 1:1 och 10 bitars omvandlaren användes för att inte upplösningen i hastighetsreferensen skall försämrans.

#### Amm

Det är ett visst slöseri med komponenter att inte låta teckenbiten gå in på D/A-omvandlaren så att man arbetar med både positiva och negativa spänningar på dess utgång. Om man då inte inverterar teckenbiten i mjukvaran kommer en OP-förstärkare och switchen i figur 29 att försvinna. (Det är den inverterande OP-förstärkaren som försvinner.) I gengäld skall man injicera halva toppvärdesströmmen genom utgången, men detta kostar bara ett motstånd och en diod.

Anledningen till att detta inte gjordes var att switchen kostade 2 kronor och att ASEA brukar använda metoden med switchning.

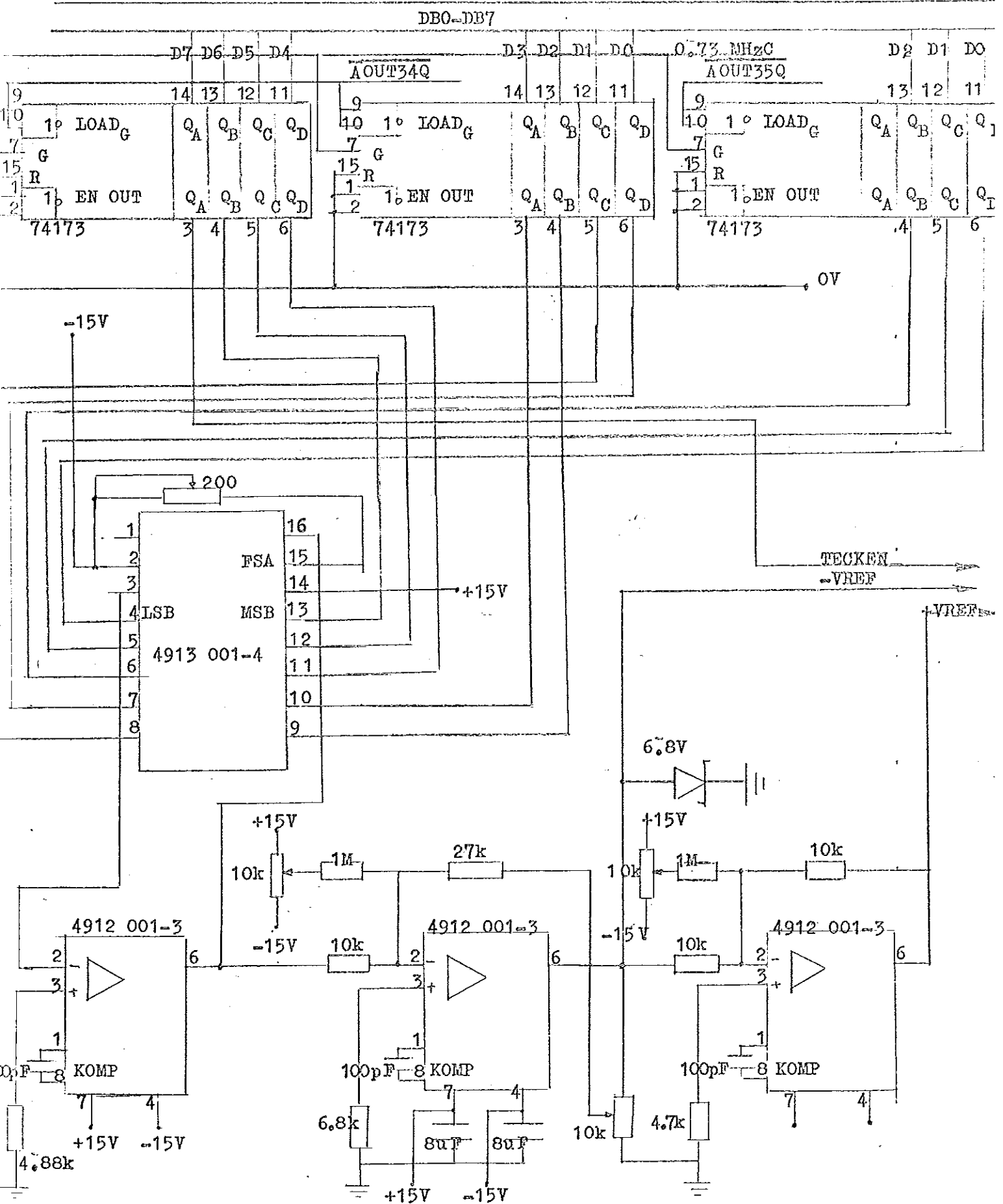
#### C. OP-förstärkar koppling

Mellan D/A-omvandlar kopplingen och switchen ligger två OP-förstärkare som skall se till att spänningsnivåerna på de signaler som levereras till switchen skall överensstämma med det analoga systemets, för en given eftersläpning. I den första OP-förstärkaren sker förstärkningen vilken har gjorts inställbar via en potentiometer, och i den andra inverteras den förstärkta spänningen, så att OP-förstärkar kopplingen levererar spänningarna +VREF och -VREF till switchen.

Potentiometrarna i återkopplingslooparna är till för att man skall kunna justera bort offseten.

8  $\mu$ F:s kapacitanserna på den ena OP-förstärkaren är till för att minska inverkan av variationer i matningsspänningarna +15 och -15V.

Zenerdioden på 6.8V skyddar switchen för spänningar större än 6.8V.



Figur 29 Buffertregister, D/A-omvandlare och OP-förstärkare i drivdonsinterfacet

D. Switch koppling.

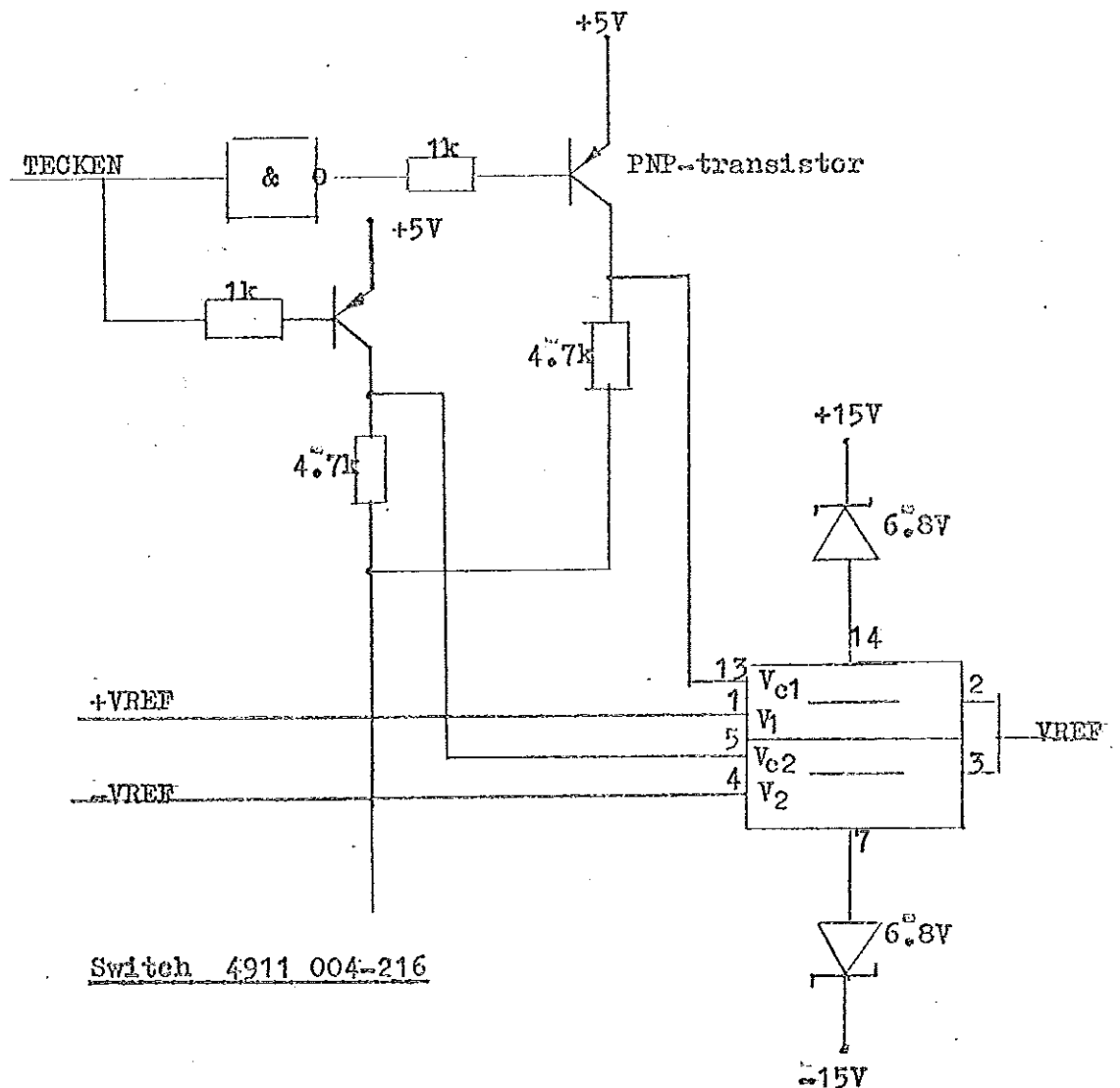
Kopplingen är till för att med hjälp av teckenbiten från datorn switcha in rätt hastighetsreferens till drivdonet.

Kopplingens funktion:

TECKEN=1      +VREF kopplas in  
TECKEN=0      -VREF kopplas in

Ämn

Genom att välja den dyrare switchen 4913 004-2 kan man låta teckenbiten gå direkt in på switchens kontrollringångar och slipper arrangemanget med PNP-transistorerna och zenerdioderna för att lägga spänningsnivåerna rätt.



Figur 31 Switch koppling.

5  
Prov

5.1  
Allmänt om proven

$\phi$ -axeln positionerades med,

A: lägesreglering i hårdvara enligt dagens system

B: lägesreglering i mjukvara enligt de konstruktionslösningar som redovisats i kapitel 4

A förkortas nedan till dagens system och B till digitalt system.

Syftet med proven var dels att jämföra regleringen i dagens system och digitalt system, dels att studera hur regleringen i det digitala systemet blev då samplingstiden ökades.

Vid jämförelsen mellan de bägge positioneringsmetoderna gjordes prov vid:

- 1.<sup>o</sup> DS=0 inkrement/10ms, driftreglering.
- 2.<sup>o</sup> DS=1 inkrement/10ms, finpositionering med väntetid.
- 3.<sup>o</sup> DS=25 inkrement/10ms, finpositionering med väntetid.
- 4.<sup>o</sup> DS=0 inkrement/10ms, laststörning.

För att utvärdera samplingstidens inverkan på regleringen i det digitala systemet gjordes prov vid:

- 1.<sup>o</sup> DS=0 inkrement/TS 10ms, driftreglering.
- 2.<sup>o</sup> DS=TS inkrement/TS 10ms, finpositionering med väntetid.
- 3.<sup>o</sup> DS=0 inkrement/TS 10ms, laststörning.

(TS · 10ms är den vid proven använda samplingstiden, TS heltal)

Förstärkningen i dagens system.

Vid eftersläpningen 1 varv=200 inkrement mättes hastighetsreferensen till 2,32V.

Då 10V motsvarar 37,5 varv/s finner man att förstärkningen

$$k_0 = 8,7 \left[ 1/s \right]$$

Förstärkningen i digitalt system.

För att kunna justera in förstärkningen så att den överensstämde med den i dagens system, skrevs ett program in i R/W-minnet via teletypen

```
LAI 346 Q
OUT 34 Q
LAI 7
OUT 35 Q
```

Programmet skriver ut det inverterade värdet av 200 till D/A-omvandlaren.

Hastighetsreferensen mättes med en digital voltmeter och justerades in till 2.32 V med potentiometern för "injusterings av förstärkning" på drivdonsinterfacet.

Anm

Innan förstärkningen kunde korrigeras till att bli samma som i dagens system, justerades offseten och gjordes full scale adjustment.

För offset justeringen skrevs programmet

```
LAI 377 Q
OUT 34 Q
OUT 35 Q
```

och potentiometrarna ställdes in så att man fick 0V i hastighetsreferens.

För full scale adjustment skrevs programmet

```
LAI 0
OUT 34 Q
OUT 35 Q
```

Fullt utslag=10V ställdes in med FSA potentiometern.  
(Mätning på D/A-omvandlarens utspänning)

Om kurvor på dagens system.

Finpositionering med väntetid får man genom att från programmeringsenheten knappa in programmet

```
PKTF
WAIT
PKTF
WAIT
PROGR END
```

Detta program får axeln att oscillera mellan två punkter. Hastigheten i rörelsen bestäms av switchen HASTIGHET AUTO.

```
AUTO 5 motsvarar 25 inkrement/10ms
AUTO 1 motsvarar 1 inkrement/10ms
```

Hastigheten 0 inkrement/10ms får man genom att trycka in PROGR STOP knappen när ett program exekveras. Kurvor över hastighetsreferens och hastighetssvar har tagits upp på skrivare. För kurvorna som avser finpositionering med väntetid visas retardationsfasen. Ingångsfilter på 92k och 0.47  $\mu$ F har använts på bägge skrivarkanalerna.

Om kurvor på digitalt system.

Hastighetsreferens ställes in från switcharna 0-5. Finpositionering med väntetid från switch 6. Programmet startas och stoppas på switch 7.

Till skillnad från i dagens system stoppar man inte programmet när man vill ha  $v_{ref}=0$  inkrement/sample, utan måste ställa switcharna 0-5 i läge 0. Vid stopp av programmet på switch 7 frikopplas nämligen den programmdel som sköter lägesregleringen.

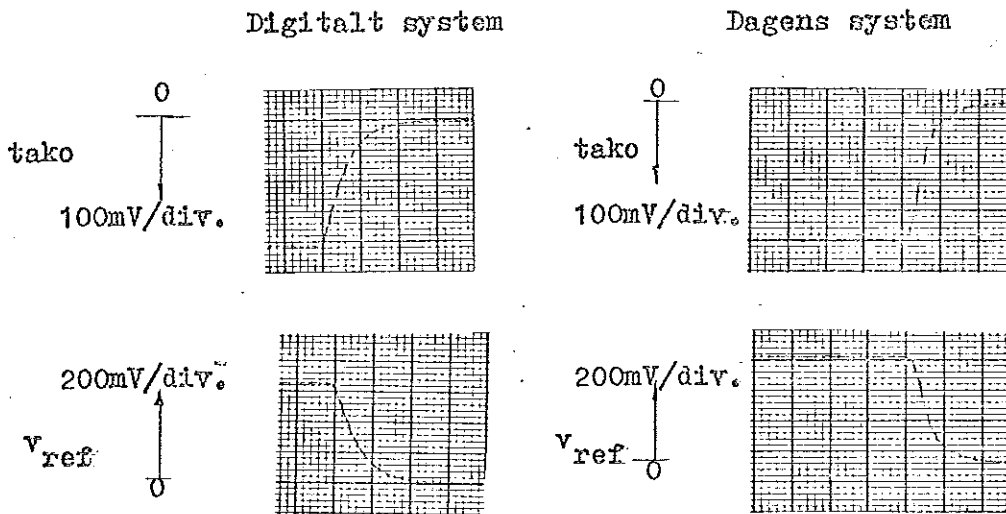
Kurvor över hastighetsreferens och hastighetssvar har tagits upp på skrivare. För kurvorna som avser finpositionering med väntetid visas retardationsfasen. Inga ingångsfiler till skrivaren har använts.

## 5.2

Lägesreglering i hårdvara, jämfört med lägesreglering i mjukvara

I detta avsnitt är samplings tiden 10ms och skrivarens pappershastighet hela tiden 25mm/s.

DS=25 inkrement/10ms, finpositionering med väntetid.



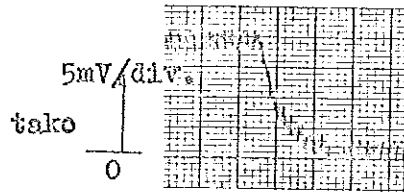
Dagens system är något snabbare än det digitala. Det borde inte vara så och en del förklaringar kan man tänka sig, som att: Förstärkningen i de bägge systemen trots allt inte är lika. När förstärkningen i dagens system bestämdes mättes eftersläpningen på oscilloskop skärm. Mätfel här slår igenom på den förstärkning som ställes in i det digitala systemet. Två olika skrivare användes för att ta upp kurvorna. Små skillnader i pappershastigheter påverkar direkt kurvornas utseende.

Med tanke på de felmöjligheter som finns tycker jag att retardationsfasen i det digitala systemet ganska väl överensstämmer med dagens systems.

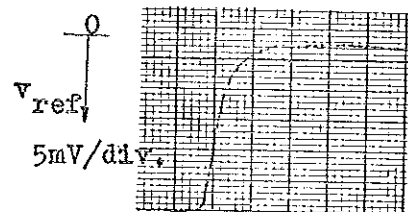
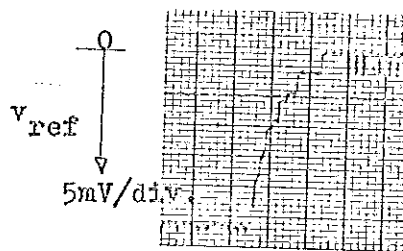
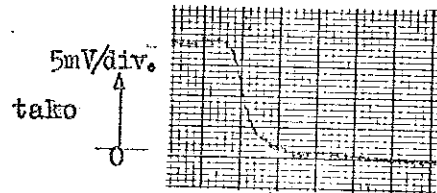


DS=1 inkrement/10ms. finpositionering med väntetid.

Digitalt system



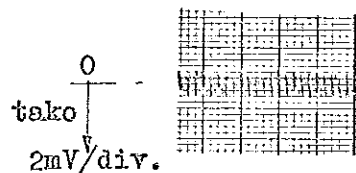
Dagens system



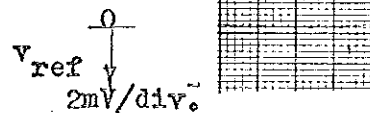
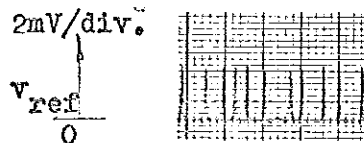
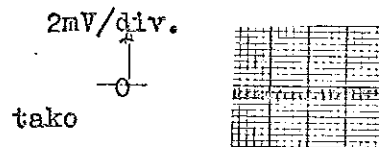
Samma som för DS=25 inkrement/10ms, ingen större skillnad mellan positioneringsmetoderna.

DS=0 inkrement/10ms, laststörning och driftreglering.

Digitalt system



Dagens system



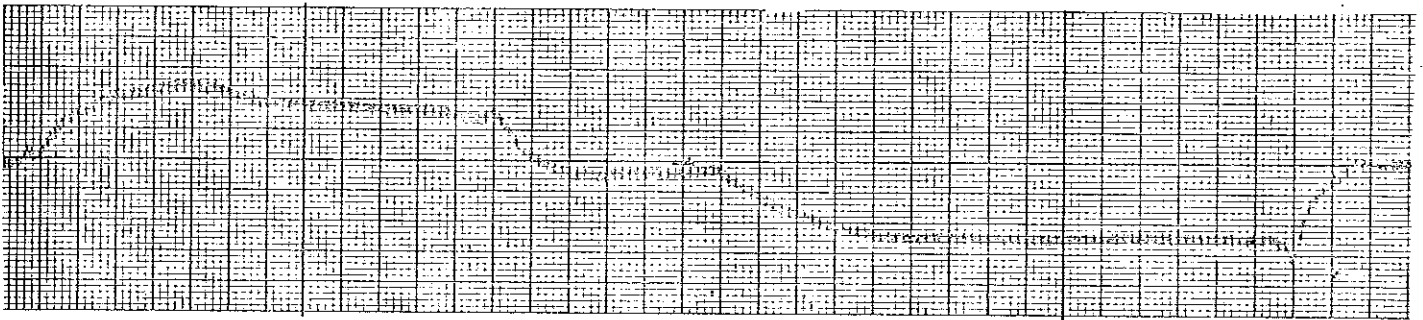
## Driftreglering

Den digitala driftregleringen skall sätta in när felet blir 1 inkrement.

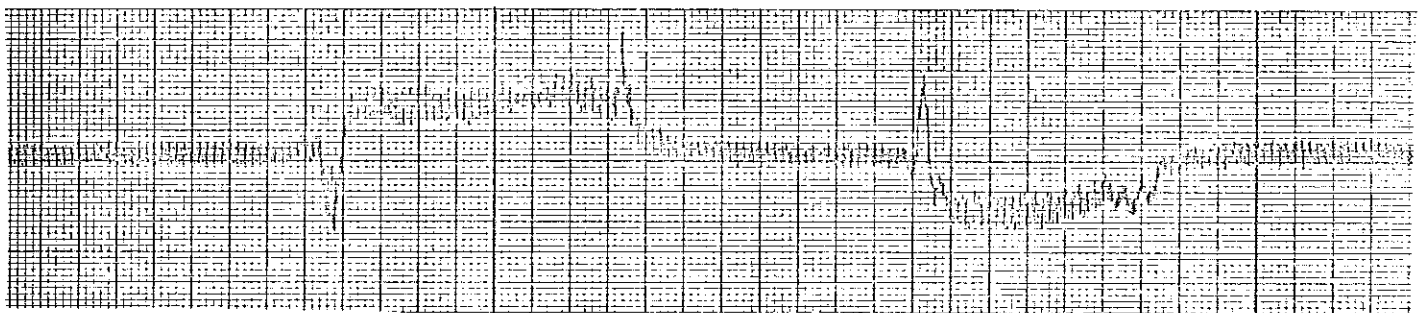
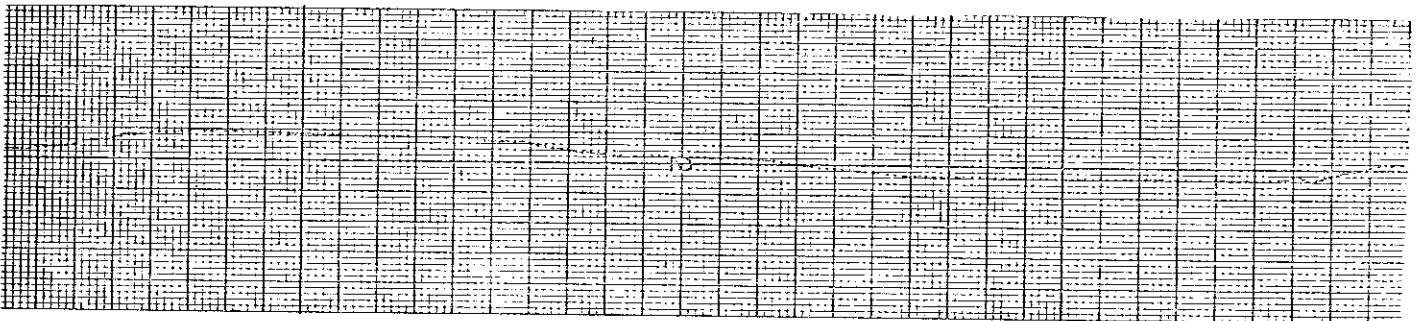
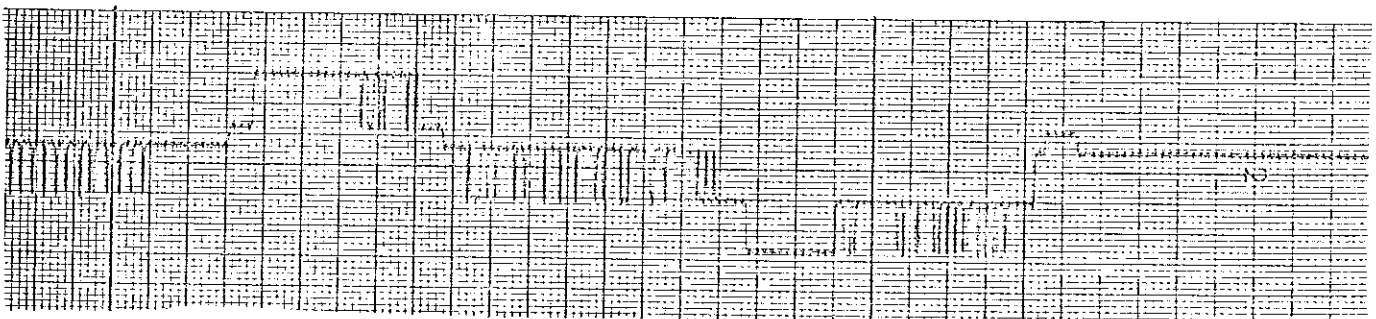
Med förstärkningen 3.7 1/s i det digitala systemet skall 200 inkrement motsvara en hastighetsreferens på 2.32 V.

alltså 1 inkrement motsvarar  $11.6\text{mV}$   
 Ur diagrammet får man att 1 inkrement motsvarar  $15\text{mV}$ .  
 D/A-omvandlarens noggrannhet är plus minus 5mV varför detta värde kan tolereras.

Dagens system, tåko 2mV/div



Digitalt system, tåko 2mV/div

Dagens system,  $v_{ref}$  2mV/divDigitalt system,  $v_{ref}$  2mV/div

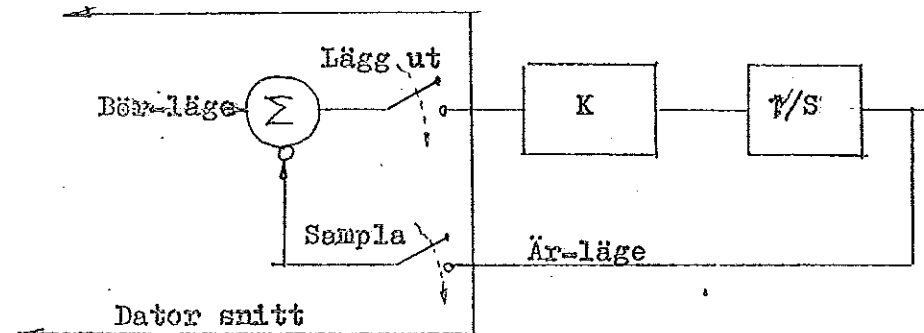
Laststörning i såväl positiv som negativ riktning.<sup>3</sup>  
 (Laststörningen är inte normerad utan varierar något från fall till fall)

Regleringen ser ungefär likadan ut i bägge systemen. Tåko-  
 topparna i det digitala systemet kan man förmoda har tagits  
 bort av filtret på skrivaren för dagens system.

5.3

## Lägesreglering i mjukvara, med ändring av samplingsfrekvensen

Reglertekniskt kan positioneringssystemet med en god approximation beskrivas av följande blockschema.



Vid dessa prov ändrades den period med vilken man samplade är-värdet av läget. Den så beräknade eftersläpningen lades sedan ut till D/A-omvandlaren på drivdonsinterfacet och fick ligga på tills nästa eftersläpning beräknats.

Avsikten med proven var speciellt att studera hur regleringen blev med avseende på svängningar i takometer signalen, då axeln kördes i hastigheten 1 inkrement/10ms.

Program för ändring av samplingstiden

```
I lägesregleringsprogrammet bytes blocket för "Enable
Interrupt"
LAI 2
OUT 10 Q
HLT
```

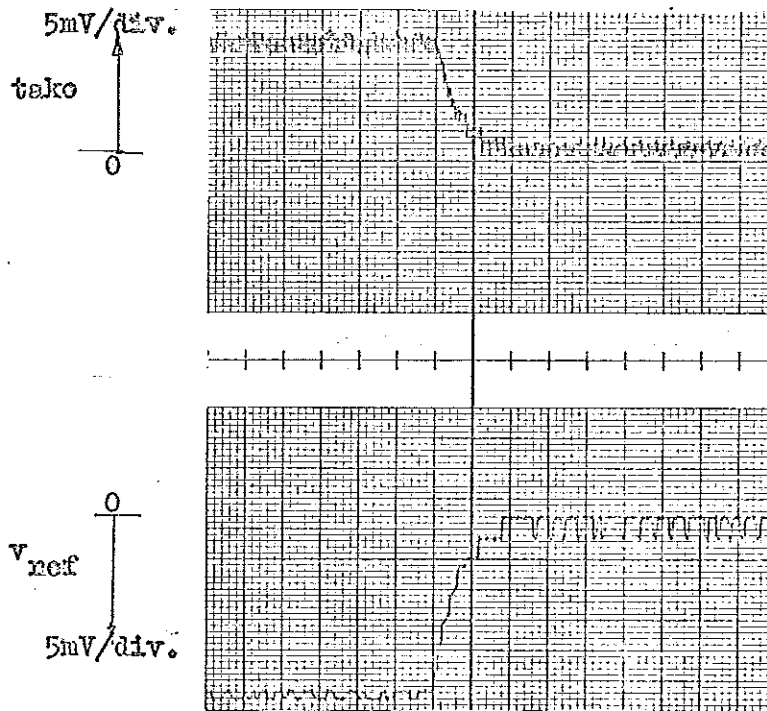
```
ut mot CALL WAIT
LAA
```

Sedan lades följande program sekvens in på ledig minnes plats

```
10MS LAI 2
      OUT 10 Q
      HLT
      RET
WAIT CALL 10MS
      CALL 10MS
      .
      .
      .
      RET
```

Varje anrop av subrutinen 10MS ger 10 ms:s väntetid. Vid till exempel 40 ms:s samplingstid skall 4 stycken CALL 10MS åtföljda av en RET instruktion förekomma.

TS=20 ms, hastighetsreferens 2 inkrement/20ms.



Man ser här att tako signalen svänger något i det område där axeln borde röra sig med konstant hastighet.

En utredning om hur långt upp i samplingstid man kan gå vid mikrodatorbaserad positionering göres av Horst Gose i WT-Zeitschrift für industrielle fertigung, Springer-Verlag. Enligt Gose kan man vänta sig god reglering för samplingstider  $T$  som är mindre än en sjättedel av systemets tidskonstant.

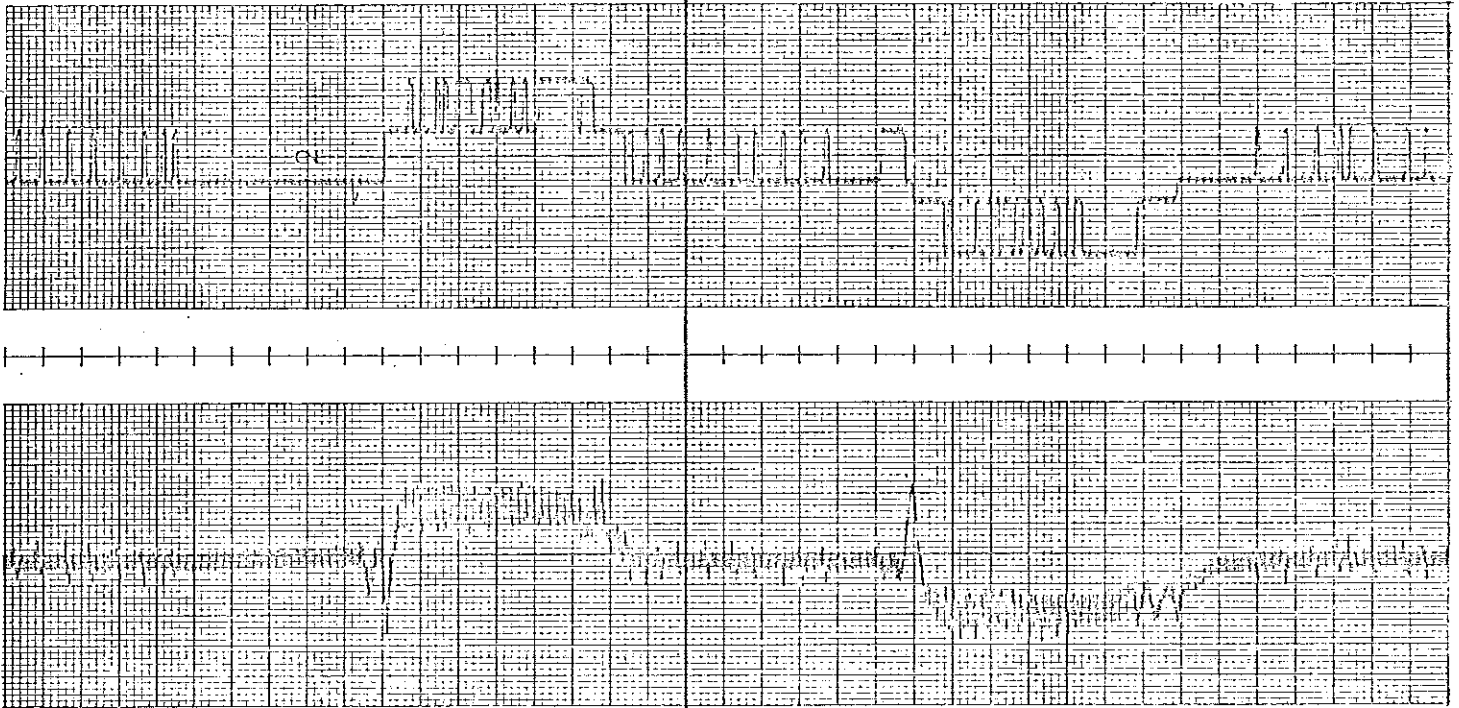
I vårt fall med förstärkningen  $k_0=8.7$  1/s bör enligt Gose samplingstiden kunna ökas till 19ms utan att regleringen försämras märkbart jämfört med den analoga regleringen.

Teorin stämmer ganska bra. Vid visuell betraktelse av axeln verkade regleringen gå bra.

Referens Horst Gose, Geschwindigkeitsproportionale Lagesollwertemittlung bei CNC mit Mikroprozessor.  
WT-Zeitschrift für industrielle Fertigung  
Springer Verlag 1977.

TS=20ms, hastighetsreferens 0 inkrement/20ms.

V<sub>ref</sub> 2mV/div.

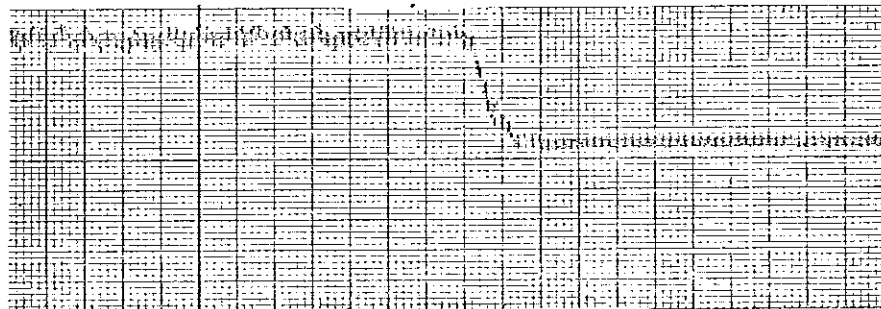


tako 2mV/div.

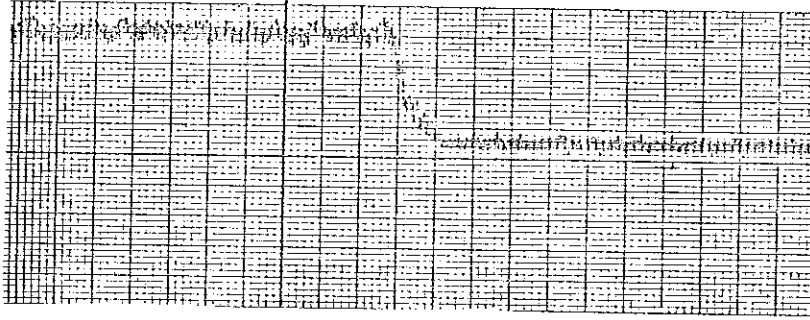
Lastströmmingar i mittpartierna och driftreglering i ytterkanterna av diagrammen.

Tako signalen är något svängigare än vid TS=10ms, men avvikelser från det analoga systemet är inte så stora.

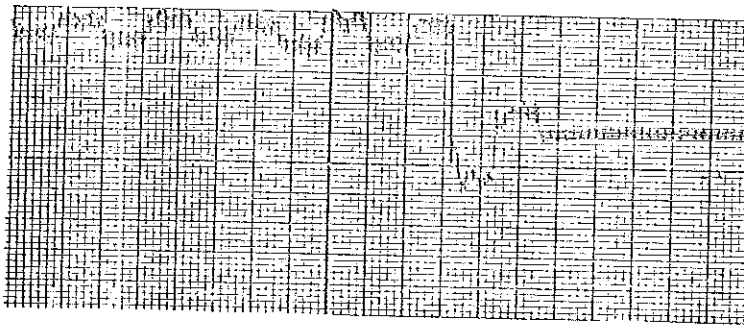
Takometer signal för TS=30, 40 och 220ms, hastighetsreferens 30 inkrement/30ms, 40 inkrement/40ms och 220 inkrement/220ms.



TS=30ms tako 5mV/div.



TS=40ms tako 5mV/div



TS=220ms tako 5mV/div

Trenden är klar, avvikelserna från konstant hastighet blir större och större ju mer man ökar samplingstiden.

Dessutom får man översläng vid höga samplingstider tack vare att den stationära hastighetsreferensen får axeln att röra sig förbi positioneringspunkten.

#### Sammanfattning av proven.

Den konstruktionslösning som användes vid den mikrodator-baserade positioneringen ger en reglering som inte väsentligt skiljer sig från den man har i dagens system.

Vid ökning av samplingstiden och med kravet på att regleringen i det digitala systemet inte märkbart skall skilja sig från regleringen i dagens system, ligger den största samplingstiden som kan användas i närheten av 20ms, vilket stämmer väl med teorin.