



K. Mellberg

**DIGITALA SYSTEM vid  
SPECIELLA  
TILLÄMPNINGAR**

*Standard Radio & Telefon AB*

AVDELNING RADIO och ELEKTRONIKSYSTEM  
BARKARBY • SWEDEN



Fruent Önskan - 64.

DIGITALA SYSTEM VID SPECIELLA TILLÄMPNINGAR

av

Civilingenjör Kjell Mellberg

STANDARD RADIO & TELEFON AB

Johannesfredsvägen 9 - 11

EROMMA

Fruent Ölafsson - 64.

DIGITALA SYSTEM VID SPECIELLA TILLÄMPNINGAR

av

Civilingenjör Kjell Mellberg

STANDARD RADIO & TELEFON AB

Johannesfredsvägen 9 - 11

EROMMA

Föreliggande kompendium utgör underlag till föreläsningar i rubricerade ämne hållna av författaren vid "Påbyggnadskurs i digitala system" anordnad i Stockholm läsåret 1962-63 av Kungl. Överstyrelsen för yrkesutbildning.

Innehållet i detta kompendium är egendom tillhörig firma STANDARD RADIO & TELEFON AB, Bromma. Eftertryck helt eller delvis må ej göras utan medgivande från denna firma.



INNEHÅLLSFÖRTECKNING

<u>Kapitel</u>		<u>Sida</u>
	Ad notam	I
	Innehållsförteckning	II
1.	<u>INLEDNING</u>	1
1.1.	Definition av specialtillämpningar	1
1.2.	Generella maskiner i specialtillämpningar kontra specialmaskiner	2
2.	<u>TYPER AV SPECIELLA TILLÄMPNINGAR</u>	3
2.1.	<u>Icke tidsbundna system</u>	3
2.1.1.	Lösning av linjära algebraiska ekvationer	3
2.1.2.	Lösning av (partiella) differentialekvationer	3
2.2.	<u>Reelltidsystem</u>	4
2.2.1.	Processreglering	4
2.2.1.1.	Allmänt	4
2.2.1.2.	Mätning av process. Sampling	8
2.2.1.3.	Styrning av processer	10
2.2.1.4.	Återkoppling. Stabilitetsproblem	11
2.2.1.5.	Inverkan av sampling	11
2.2.1.6.	Inverkan av tidsfördröjningar	22
2.2.1.7.	Inverkan av brus	31
2.2.1.8.	Läges- och kursfel	34
2.2.1.9.	Följning	36
2.2.2.	Simulation	44
2.2.3.	Digitalteknik inom kommunikation	46
2.2.3.1.	Datakommunikation	46
2.2.3.2.	Automatiska telefonväxlar	51
3.	<u>DIGITALA ÜTRUSTNINGAR VID SPECIALTILLÄMPNINGAR</u>	55
3.1.	<u>Systemkrav</u>	55
3.2.	<u>Generella maskiner i specialtillämpningar</u>	57
3.2.1.	Allmänt	57
3.2.2.	Prioritetsavbrott	58
3.2.3.	Direkt minnesaccess	64
3.2.4.	Maskinsnabbhet. Lagrade och fasta program	67
3.2.5.	Platasökning. Dataadresserade minnen	70

<u>Kapitel</u>		<u>Sida</u>
3.3.	<u>Speciella digitalsystem</u>	72
3.3.1.	Digitalstyrda servon	72
3.3.2.	Detalänkterminaler	76
3.3.2.1.	Allmänt	76
3.3.2.2.	Sändarterminaler	77
3.3.2.3.	Mottagarterminaler	79
3.3.3.	Den digitala differentialanalysatorn	81
3.3.3.1.	Allmänt	81
3.3.3.2.	Principen för den digitala differentialanalysatorn	81
3.3.3.3.	Praktisk utformning av DDA	84
3.3.4.	Den binära multiplikatorn	85
3.3.4.1.	Allmänt	85
3.3.4.2.	Binär multiplikator (modulator)	85
3.3.4.3.	Direkt användning av den binära integratorn för funktionsgenerering	88
3.3.4.4.	Indirekt användning av integratorn för utförande av vissa matematiska operationer	90
3.3.4.5.	Generering av sinus-cosinus-funktioner	93
3.3.4.6.	Metod att förbättra noggrannheten hos en binär modulator genom att använda avrundande steg	95
3.3.5.	Kontinuerlig datapresentation	95
3.3.5.1.	Allmänt	95
3.3.5.2.	Direktskrivningsmetoden	96
3.3.5.3.	Rastermetoden	97
3.3.5.4.	Generering av godtyckliga kurvor	100
4.	<u>DRIFTSÄKERHET</u>	101
4.1.	Felfrekvens och medeltid mellan fel	101
4.2.	Införande av redundans. Majoritetsval	103
4.3.	Några slutsatser	109
	<u>LITTERATURREFERENSER</u>	110

# 1. INLEDNING

## 1.1. Definition av specialtillämpningar

Gränsen mellan generella och speciella tillämpningar inom dataområdet är ofta mycket svårdefinierad. I detta fall skall med generella tillämpningar menas användning av maskiner för vetenskapliga beräkningar och för kontorsbruk, där maskinens anknytning till omvärlden är mycket lös och ofta begränsas till mänskligt handhavande av in/ut-media såsom hålkort, håltremsor, magnetband och skrivmaskiner. Arbetsuppgifterna består normalt av matematisk behandling och/eller sortering av data från och till dessa media. En generell maskin skall kunna användas för olika typer av sådana arbetsuppgifter genom omprogrammering i maskinens minne. Dels på grund av den erforderliga generaliteten i programmeringsmöjligheterna och dels på grund av de ofta mycket långa programmen och stora behandlade datamängderna krävs mycket stora minnen i maskinerna. Däremot är normalt kraven på behandlingssnabbhet inte särskilt stora, utan maskinen får ta den tid på sig, som den behöver. Denna frihet gäller givetvis inom vissa gränser, så att maskinen kan utnyttjas ekonomiskt.

Med speciella tillämpningar skall här avses sådana användningar, där arbetsuppgifterna för en maskin är mer eller mindre bestämda, även om de kan vara både många och komplicerade. Anknytningen till omvärlden är ofta mycket fast och förmedlas över omvandlande organ eller transmissionsmedia förutom via mänskliga operatörer. Vanligen framkommer mycket speciella och höga krav på vissa prestanda hos maskinerna närmast ifråga om snabbhet och simultanbetjäning av yttre organ. För varje tillämpningsfall erfordras ofta speciella yttre organ såväl som speciella åtgärder i själva maskinen.

Speciella tillämpningar kan kanske p.g.a. benämningen ge intryck av att vara sällsynta inom databehandlingsområdet, men faktum är, att de på senaste tid växt till en väldig omfattning p.g.a. utnyttjande vid processregleringar och transmissionsteknik.

De tekniska problemen vid konstruktion av generella maskiner består huvudsakligen i att skaffa användaren ett lätthanterligt instrument med sådan flexibilitet, att han själv kan anpassa detsamma till önskade arbetsuppgifter. I de speciella tillämpningarna kan dessa krav kvarstå åtminstone i stora system, men dessutom måste en stor del av konstruktionsarbetet inriktas på att anpassa maskinen till den systemmiljö, i vilken den skall arbeta. Av konstruktören krävs alltså i detta fall en god kännedom om



systemet i sin helhet samt företeelser inom det samma, vilket gör arbetet svårare men samtidigt intressant och omväxlande. Därför skall även en del här ägnas åt vissa viktiga systemtyper och problem sammanhängande med dessa.

## 1.2. Generella maskiner i specialtillämpningar kontra specialmaskiner

Vid planering av digitala system av specialkaraktär ställs men ofta inför valet av bästa utrustning, för att systemet skall fungera tillfredsställande och samtidigt vara ekonomiskt i tillverkning och drift. Man har två huvudalternativ; det ena är att använda generella datamaskiner och programmera dessa, så att de kan fullgöra systemfunktionerna eventuellt med tillsats av vissa speciella delar. Det andra alternativet är att konstruera utrustningar speciellt utformade efter de arbetsproblem de har att lösa, dvs utan den generella maskinens flexibilitet. Valet dikteras som sagt dels av möjligheterna att uppfylla systemfunktionerna och dels av ekonomiska hänsyn. Den generella datamaskinen har för många tillämpningar vissa begränsningar; ofta gäller detta snabbheten p.g.a. att maskinen måste programmeras efter givna normer som inte tillåter några genvägar för speciella operationer. I ett sådant fall kan en speciellt konstruerad maskin ofta klara uppgiften, genom att dess utformning tar speciell hänsyn till det givna problemet helt eller delvis på bekostnad av möjligheter till andra arbetsuppgifter. I andra fall, i synnerhet i större system, kan den generella maskinen vara alltför exklusiv och dyrbar, p.g.a. att endast fåtal typer av operationer krävs. I ett sådant fall blir den generella maskinen dåligt utnyttjad och ersättes lämpligare med en speciellt konstruerad utrustning, som endast innehåller de delar som behövs för fullgörande av de specificerade uppgifterna. I det senare fallet kommer emellertid alltid en annan faktor med i spelet, och det är konstruktionsarbetet för en speciell utrustning. Ofta kan en sådan konstruktion drivas direkt från hyllan utan måste nykonstrueras med ty åtföljande konstruktionskostnad och konstruktionstid. Om det väger på gränsen i ett sådant fall, kan det många gånger vara tillrådligt att ta en generell maskin antingen som den är eller med sådana nödvändiga modifieringar, som inte kräver alltför stor konstruktionsinsats.

Ytterligare en faktor talar för det sistnämnda valet. En speciell konstruktion kan inte förväntas utgå i så stora upplagor som en mera generell sådan, varför konstruktionskostnaden kommer att bli en avgörande faktor vid valet. Vidare är det oft så i specialtillämpningar, att systemkraven är svåra att i förväg specificera på ett så fullständigt sätt, att senare

ändringar helt kan undvikas. Även här kommer en generellare lösning att premieras genom billigare ändringsförfarande (exempelvis omprogrammering). Förutsättningen är dock givetvis alltid, som nämnts att den generella lösningen kan uppfylla systemkraven varför många gånger de generella maskinerna måste förses med vissa speciella drag, vilket närmare skall diskuteras i fortsättningen.

## 2. TYPEN AV SPECIELLA TILLÄMPNINGAR

I detta avsnitt diskuteras olika problemtyper som kräver någon form av specialtillämpningar för att lösas på ett optimalt sätt. Systemlösningar presenteras också i vissa fall, men utan detaljerad genomgång av kraven på de olika utrustningarna. Dessa senare genomgås längre fram, då en överblick över problemställningarna erhållits genom detta avsnitt.

### 2.1. Icke tidsbundna system

#### 2.1.1. Lösning av linjära algebraiska ekvationer

Ett problemområde, som man väl vanligen direkt associerar med generella datamaskiner, är lösning av ekvationer, som framkommer i så gott som alla vetenskapliga beräkningar. Av dessa ekvationer förekommer mycket ofta den linjära typen med antalet oberoende variabler varierande mellan ett fåtal och nästan oändligt många. Nu är det emellertid så, att vid många variabler den generella maskinen är svårprogrammerad och långsam. Den är också rätt dyrbar, samtidigt som dess flexibilitet dåligt utnyttjas. I vissa fall har man därför funnit det ekonomiskt att utveckla speciella ekvationslösare, som emellertid knappast har allmänt intresse. De berörs därför inte här, utan den intresserade hänvisas till litteraturförteckningen i slutet av detta kompendium (ref. 1-2).

#### 2.1.2. Lösning av (partiella) differentialekvationer

En annan typ av ekvationer, som ofta dyker upp vid vetenskapliga beräkningar, kanske speciellt i matematisk fysik, är differentialekvationerna. Komplexa sådana ekvationer har sedan länge vanligen lösts med hjälp av analogsystem av mekanisk eller elektronisk konstruktion. I dag görs detta också vanligen med hjälp av elektroniska analogmaskiner exempelvis vid studium av invecklade fysikaliska förlopp. Emellertid har analogsystemen sina begränsningar vad gäller noggrannhet och flexibilitet, varför i vissa tillämpningar (i synnerhet icke tidsbundna) önskemålet om digitala metoder är ganska starkt (framställning av tabeller etc). Man använder därvid kända numeriska metoder, som oftast går ut på att differentialekvationerna ersätts av ekvationer med ändligt små differenser i en serie av punkter inom det



intressanta området av funktionen.

Även i detta fall kan den generella maskinen användas, men för vissa tillämpningar kan man konstruera speciella utrustningar, som är både billigare, snabbare och lättare att programmera. Sådana digitalutrustningar berörs närmare i avsnittet om digitala differentialanalysatorer.

## 2.2.

### Reelltidsystem

I föregående avsnitt har sådana tillämpningar berörts, där inga särskilda tidskrav ställts annat än att lösningsledningarna inte bör bli alltför långa. Uppsättningen av problemen görs också normalt av mänskliga operatörer utan någon direkt bundenhet till omgivande händelser. I detta avsnitt skall däremot behandlas sådana tillämpningar, där tidsanknytningen till fysikaliska skeenden är mycket fast och dessa skeendens variationsfrekvenser bestämmer de tider, som finns tillgängliga för databehandlingen. Problemet gäller därvid oftast att utgående från data, som varierar med tiden, bilda andra data under den tid, som ursprungsdata är aktuella. Problemställningen uppträder vid två huvudtyper av tillämpningar nämligen processreglering och kommunikation.

### 2.2.1

#### Processreglering

#### 2.2.1.1.

##### Allmänt

Som namnet säger, innebär processreglering övervakning och styrning av fysikaliska processer. Fig. 1 visar ett typiskt blockschema för ett sådant system. Det reglerande organet, datamaskinen, insamlar data om processens

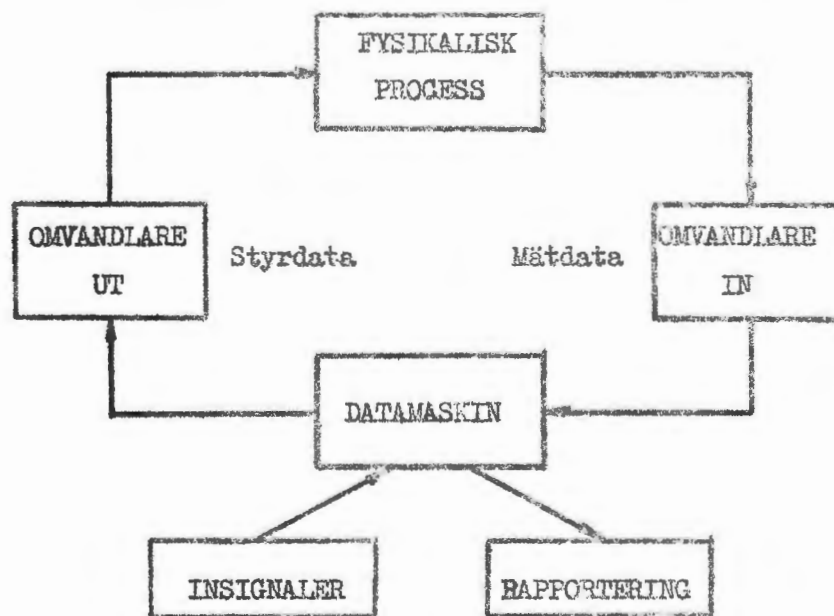


FIG. 1

tillstånd, utvärderar dessa data enligt givna instruktioner, som oftast innebär en bedömning av processens utveckling, samt återlämnar data, som påverkar processen i önskad riktning om den avviker från det riktiga tillståndet. Här framgår den bundenhet till verklig tid, som systemet har. Datamaskinen måste nämligen dels hinna ta emot mätdata tillräckligt ofta, för att processens detaljerade beteende skall kunna följas. Vidare måste styrdata beräknas och utmatas i god tid, innan processen spårar ur. Databehandlingshastigheten i datamaskinen är alltså beroende av processens variationshastighet eller om man så vill dess tidskonstant. Systemet påminner alltså i hög grad och är faktiskt också ett återkopplat servosystem, där som bekant alltför stora tidsfördröjningar i loopen kan åstadkomma självsvängningar i systemet.

I stora komplexa system kan det inträffa, att processen plötsligt ändrar karaktär och kräver en ny typ av databehandling i maskinen. Detta innebär, att den senare måste växla program för att kunna styra processen på rätt sätt. Antingen kan denna programväxling initieras genom signaler från processen, eller också får datamaskinen lov att med ledning av processens iakttagna utveckling själv bestämma växlingen.

Andra problem i dessa typer av tillämpningar är kravet på direkt kommunikation mellan datamaskin och process. Eftersom de flesta processer är av analog karaktär, erfordras oftast omvandlare från signaler i analog form till signaler i digital form och vice versa mellan processen och datamaskinen. Ofta erfordras också mer eller mindre kontinuerlig mätning av flera processvariabler och likaså utmatning av flera styrsignaler, för att processen skall kunna styras på erforderligt sätt. Detta innebär flera in- och utgångar till datamaskinen, som denna skall kunna betjäna inom de tillgängliga begränsade tidsperioderna. Samtidigt kan även information från andra organ behöva tillföras systemet exempelvis uppgifter om tidpunkter, då processen skall ändras och hur den skall ändras beroende på diverse yttre betingelser eventuellt i andra anknutna processer. Datamaskinen kan också behöva lämna information om processens beteende till övervakare och dylikt ofta via fjärrförbindelser och datapresentationsutrustningar. Det hela resulterar i stora krav på maskinens kommunikationsmöjligheter med yttre organ av varierande typ och på dess möjligheter att ideligen växla automatiskt mellan olika arbetsuppgifter.

Processregleringsystem är i regel på ett eller annat sätt återkopplade enl. fig. 1. Givetvis kan man tänka sig kedjan uppbruten, så att exempelvis ingen mätning sker (okontrollerad styrning) eller ingen styrning (mät-

och följsystem). Båda systemtyperna förekommer, dock ofta som självständiga länkar i återkopplade system. Återkopplingen är vanligen önskvärd i synnerhet vid styrning eftersom detta ger möjlighet till kontroll av styrningen.

Här skall några exempel på återkopplad processreglering ges. Det kanske allra enklaste åtminstone uppbyggdsmässigt sett är ett regelrätt s.k. "digitalt servo", som kan exemplifieras med numerisk styrning av en axels vridning. Fig. 2 visar blockschema för en sådan anordning. Indata, dvs de data som anger axelns önskade vridningsrörelse (ofta kallas värdena på indata för "börvärden"), jämförs med de verkliga data för axelvinkeln (ofta kallade "ärvärden"), som erhålls från en digital kodskiva monterad på servomotorns axel. Det digitala resultatet av jämförelsen, dvs axelfelet,

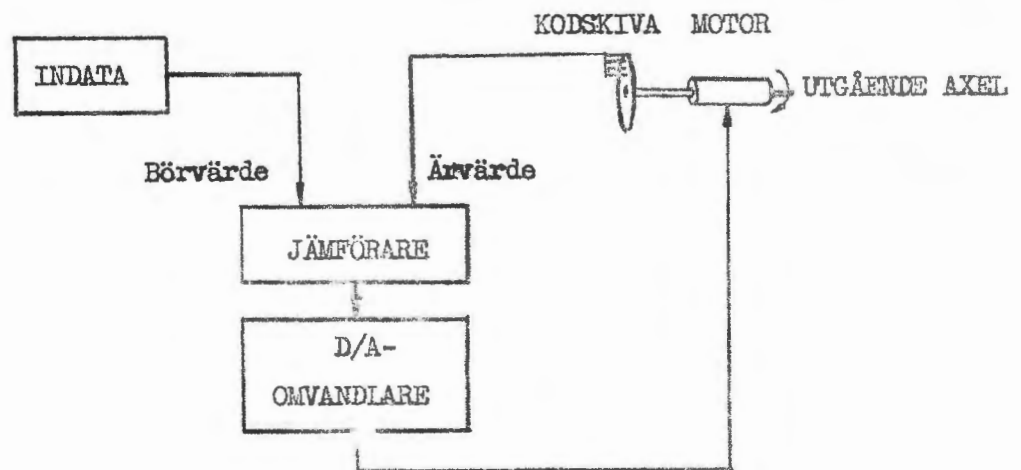


FIG.2

omvandlas i en digital-analog-omvandlare till en analogispänning så beskaffad, att den styr servomotorn åt det håll som tenderar att minska axelfelet. I omvandlaren kan den styrande analogispänningen göras godtyckligt stor i förhållande till det digitalt uppmätta felet. Detta innebär, att servomotorn alltid kan styras in i rätt läge med godtycklig noggrannhet, genom att kodskivans noggrannhet görs tillräckligt stor. Det är alltså klart, att ett digitalservo av denna typ kan återbilda en axelrörelse med en noggrannhet, som är otänkbar på analog väg. Knuten ligger just däri, att digitala värden med godtyckligt stor noggrannhet kan jämföras utan försämring av denna noggrannhet, vilket inte är möjligt med analogteknik p.g.a. de analoga jämförarorganens egna inneboende fel.

Anordningen i fig. 2 är huvudsakligen densamma som i fig. 1, varvid jämförare, DIAN, motor och kodskiva i den förra figuren motsvarar resp. datamaskin,



ut-omvandlare, process och in-omvandlare i den senare.

Ett mer komplicerat fall kan få illustreras av ett system för automatisk lufttrafikövervakning (Air Traffic Control) enligt fig. 3. Den övervakade processen är i detta fall ett eller flera flygplan, omvandlare in

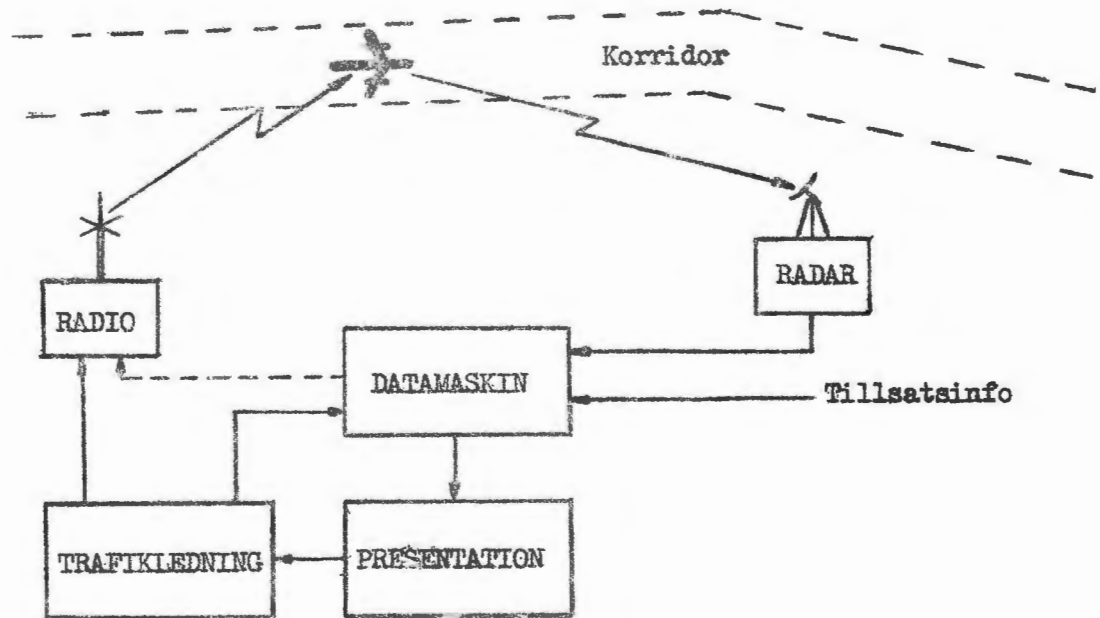


FIG. 3

utgörs av radarn, datamaskinerna av en ofta komplicerad utrustning, som följer flygplanens väg och med ledning av indata i form av order från trafikledningen och tillsatsinformationer (väderuppgifter, tidsuppgifter, andra radarinformationer, flygvägar etc.) bedömer riktigheten av flygplanförlyttningarna, samt omvandlar ut av talradioförbindelse mellan trafikledningen och flygplanen. Över den sistnämnda dirigeras trafiken enligt beslut grundade på datautrustningens beräkningar erhållna via en presentationsutrustning (PPI:n, tabellindikatorer etc.). I ett ännu mer automatiserat system kan talradioförbindelsen från trafikledningen ersättas av en dataförbindelse direkt från maskinen (streckat i fig. 3).

Systemet enligt fig. 3 är rätt komplicerat p.g.a. den mängd databehandling som skall utföras i reell tid (målföljningsberäkningar, kontroll mot bestämda flygbanor, tidberäkningar, förmedling av data från olika inomvandlare och till olika utomvandlare exvis för presentation) och den mängd samtidigt förlopp, som skall övervakas (flera flygplan, flera presentationsorgan etc.).

I det beskrivna systemet har antagits, att flygtrafiken skall kanaliseras i vissa tillåtna korridorer och att det är datautrustningens uppgift att övervaka och leda denna kanalisering. Systemet kan modifieras, så att de anbefallda flygvägarna görs beroende av en yttre variabel exempelvis ett rörligt mål. Detta tillämpas vid luftförsvaret, där jaktplan eller robotar leds mot fiendliga flygande mål. Systemet får då utseendet enligt fig. 4. Indata till systemet, som i fig. 3. representerades av stridsledningens

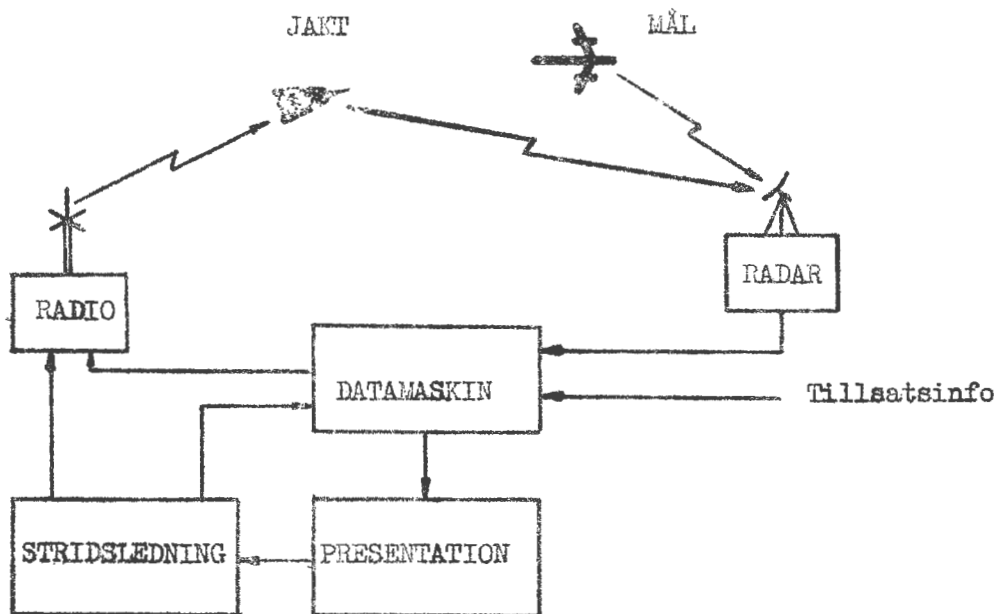


FIG. 4

beslut och information om tillåtna flygvägar, representeras i fig. 4 av målets rörelser och det är systemets uppgift att inta ett sådant tillstånd att motvapnet i form av jakt eller robot får kontakt med målet. Här kommer alltså processens är- och börvärden via samma medium till datautrustningen, och de har dessutom samma karaktär, varför det krävs av datamaskinen (eventuellt med hjälp av operatörer) att den skall kunna särskilja de två typerna av information för möjlighet att övervaka och styra processen.

#### 2.2.1.2. Mätning av process. Sampling

Av det föregående har framgått, att man i princip kan uppdelat ett processregleringssystem i följande funktioner förutom processen själv: mätning av processens utveckling, jämförelse av processdata, ärvärdena, med någon typ av börvärden, vilket utgör själva databehandlingen, samt styrning av processens vidare utveckling med hänsyn till börvärdena.

Följningen av en process innebär omhändertagande av mätdata från varierande



typer av omvandlare. Oftast är sådana omvandlare s.k. analog-digital-omvandlare, eftersom flertalet processer är av analog karaktär. De flesta omvandlare återfinns bland huvudtyperna kodskivor, digitala spänningsmätare och inkrementgivare (exvis pulsgivare). Metoderna att anknypa omvandlarna till databehandlingsmaskinen varierar. Ofta sker anknypningen direkt, medan i vissa fall andra digitalutrustningar måste utnyttjas för anpassning mellan omvandlare och maskin. Det senare gäller ofta för generella maskiner i stora system, där viss förbehandling av mätdata måste göras i specialutrustningar före matning till den centrala maskinen.

I de mest komplicerade systemen kan faktiskt en eller flera fullständiga datamaskiner behövas som anpassningsorgan eller buffertar mellan den centrala maskinen och omvärlden för exempelvis uppsamling, sortering och sällning av i systemet mottagen information. Det stora informationsflödet av normalt både nyttig och överflödigt karaktär kan därigenom reduceras stegvis till så nyttig information som möjligt, innan den centrala maskinen belastas därmed.

En annan typ av anpassningsbehov mellan mätorgan och central maskin möter man i de fall mätvärdena skall överföras längre sträckor (telemetering). I detta fall gäller det att erhålla en anpassning till och från överföringsmediet, som normalt utgörs av radio- eller telefonlänkförbindelse. Dessa anpassningsorgan är i sig själva digitalutrustningar av varierande komplexitet och skall beröras närmare i avsnitt 2.2.3.1.

Ur systemsynpunkt är det högst intressant, hur stort dataflödet mellan omvandlarna och den centrala maskinen blir. Detta flöde bestämmer ju en del av belastningen på maskinen, varför det gäller att välja sådana dataanknytnings- och nedbringningsorgan som nedbringar belastningen till ett minimum med behållande av systemekonomin. Å andra sidan bestäms anknypnings sättet och dataflödet av typ av följd process samt typ av omvandlare och anpassningsorgan. Innan vi går närmare in på dessa saker, så låt oss se på ett grundläggande drag hos informationsmatning till digitala utrustningar. På grund av den relativa lätthet med vilken digital information kan kopplas mellan olika digitala utrustningar samt lagras i minnen behövs inte någon kontinuerlig förbindelse mellan exempelvis mätorgan och centralutrustning, som oftast är fallet i analogisystem, utan dataförmedlingen kan inskränkas till de tidpunkter, då det är nödvändigt för att processen skall kunna följas nöjaktigt. Förfarandet kallas allmänt sampling, och den yttersta gränsen för ovannämnda inskränkning i dataförmedlingen, dvs den lägsta samplingsfrekvensen, bestäms av samplingsteoromet. Även om man normalt inte kan gå så

långt ned i samplingsfrekvens, som teoremet tillåter, p.g.a. att alltför komplicerade filterfunktioner i så fall behövs i maskinen, så inser man, att ju långsammare en process förändrar sig desto lägre kan samplingsfrekvensen tillåtas vara. Mellan samplingtidpunkterna för en viss förbindelse kan då sampling på andra förbindelser göras, om sådana finns. Problemet är att få den centrala maskinen att kunna sampla olika datakällor och att även kunna skifta sina beräkningar mellan de olika mottagna datatyperna.

Samplingproblemet kan i princip uppdelas i två huvudfall, som ger olika krav på datamaskinen. Vi kan här kalla dem fallen med samlade resp. samlade omvandlare. Med samlade omvandlare menas då sådana omvandlare, som är ständigt förbundna med processen och som på order från exempelvis den centrala maskinen mäter och omvandlar resp. processdata och förmedlar dem till maskinen. I ett sådant fall kan den centrala maskinen alltid programmeras för en viss avsökningssekvens för omvandlarna, dvs behandlingen av processdata för de olika förbindelserna läggs i sekvens i maskinens program, och då maskinen kommer fram till behandling för en viss förbindelse, avläses omvandlaren för denna förbindelse. Som exempel på samlingsbara omvandlare kan nämnas kodskivor anbringade på roterande axlar samt digitala spänningsmätare. I vissa fall kan även inkrementgivare räknas dit, nämligen om de innehåller ett eget minne för mätdata.

Med samlade omvandlare menas sådana omvandlare, som endast vid vissa tidpunkter kan erhålla mätvärden från den övervakade processen. Typiska exempel på sådana omvandlare är radarstationer och telemeterförbindelser för fjärröverföring av data. Till samlade omvandlare kan även räknas inkrementgivare utan eget minne. Omvandlare av denna typ styrs alltså inte av den centrala maskinen utan samlar själva mätdata vid varierande tidpunkter beroende på typ av omvandlare. Det är alltså inte möjligt att synkronisera avsökningen med någon viss behandlingscykel i den centrala datamaskinen, utan den senare måste konstrueras, så att den kan mottaga data vid vilka tidpunkter som helst och ofta också skifta beräkningsprogram vid sådan yttre sampling.

### 2.2.1.3

#### Styrning av processer

Processstyrning innebär i regel utsändning från den centrala maskinen av elektriska impulser för drivning av motororgan, som ställer in processparametrarna på önskat sätt (inställning av ventiler i kemiska processer, skärverktyg vid maskinell bearbetning, rotermotorer vid styrning av



flygplan och robotar etc.). Vid styrning liksom vid mätning måste oftast någon typ av omvandlare finnas mellan maskinen och det styrda processorganet vanligen då digital-analogomvandlare. Ibland fungerar även mänskliga operatörer som omvandlare exempelvis mellan visuellt presenterade order från maskinen och manuell påverkan av processorgan.

Även vid styrning är det önskvärt med sampling på så sätt, att maskinen endast vid vissa tidpunkter matar styrdata till ett visst organ och däremellan matar andra organ eller utför annat arbete. Likaså kan man också i detta fall skilja mellan samplade och samplande omvandlare. Exempel på de förra är digital-analogomvandlare och på de senare telemeterförbindelser för fjärrstyrning. Konsekvenserna i de båda fallen för den centrala maskinen är desamma som vid följning.

Den lagring av data mellan samplingsögonblicken, som vid följning kan göras i den centrala maskinens minne måste i styrfallet göras i omvandlaren i s.k. samplingshållkretsar. Dessa hållkretsar kan antingen ligga på digitalsidan och utgöras av register eller på analogsidan och utgöras av någon typ av kondensatorhållkretsar.

#### 2.2.1.4 Återkoppling. Stabilitetsproblem

Då man som i fig 1 avsnitt 2.2.1.1. knyter ihop process, inorgan, databehandlingsutrustning och utorgan får man ett återkopplat styrsystem, som i sin allmännaste form kan anges enligt fig 5. Ett sådant system kan som bekant ge stabilitetsproblem beroende på de i systemet rådande frekvens-,

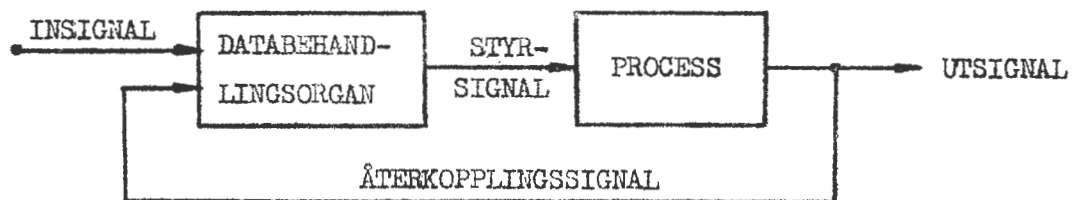


FIG. 5

fas- och brusförhållandena. Dessa problem förvärras i allmänhet av den ovannämnda samplingen och även av tidsfördröjningar i systemet p.g.a. databehandlingen. Här är inte rätta platsen för något detaljerat studium av dessa frågor, men för att några av de viktigaste problemställningarna vid sådana här system skall belysas, genomgås i det följande en del enkla exempel på hur stabiliteten kan påverkas av systemutformningen.

#### 2.2.1.5 Inverkan av sampling

Låt oss anta, att systemet enligt fig. 5 är samplat så att såväl in-,

styr- som återkopplings signaler endast erhålles med bestämda tidsintervall, och låt oss vidare för enkelhets skull anta, att samplingstidpunkterna är desamma för alla signalerna. Låt oss till sist förutsätta, att utsignalen skall anta insignalens värden så bra som möjligt. Detta ger oss ett enkelt fall för att iakttaga beteendet hos ett återkopplat system vid olika samplingsfrekvenser, tidsfördröjningar och systemsnabbheter. De kurvor för olika förlopp, som visas i det följande, är högst idealiserade och förenklade och tjänar endast till att underlätta förståelsen för de fenomen, som behandlas.

Fig 6 visar responsen i systemet för det fall, att insignalen är en stepfunktion. För enkelhets skull har här liksom i det följande antagits, att systemets styrda förändring sker rätlinjigt med en derivata som är proportionell mot styrsignalens amplitud. Fig 6 visar två fall, varav det ena har ett samplingintervall  $T_1$ , som överstiger systemtidskonstanten  $t$ , och det andra ett intervall  $T_2$ , som är hälften av  $T_1$  och understiger  $t$ . Med hjälp av felsignalkurvorna nedtill i figuren har de idealiserade insvängningsförloppen härletts. Det framgår av dessa, att det längre samplingsintervallet dels ger översvängningar i systemet och dels kräver betydligt längre tid för insvängning än det kortare samplingsintervallet. I detta idealiserade fall skulle tydligen den snabbaste insvängningen erhållas för  $T = t$ . Ju snabbare systemets respons är, desto högre bör alltså samplingsfrekvensen vara. Detta krav kan dock i viss mån kringgåås med hjälp av knep, som skall beröras längre fram. Ett samplingsintervall understigande systemtidskonstanten ger aperiodisk insvängning, medan ett längre intervall ger en insvängning med underkritisk dämpning, som blir lägre och lägre för att vid  $T = 2t$  ge odämpad svängning (självsvängning). Läsaren kan själv övertyga sig om detta med hjälp av samma grafiska metod, som använts i fig 6.

I fig 7 illustreras några fall, då insignalen är en rampfunktion (linjärt föränderlig funktion). Kurvor för fyra olika responstider (tidskonstanter) har angivits. Den långsammaste är lika med samplingsintervallet (under ett samplingsintervall ändrar sig då systemet med ett värde lika med det samplade felets amplitud), nästföljande lika med  $2/3$  (ändring 1.5 ggr felets amplitud), den tredje hälften och den fjärde  $1/3$  av samplingsintervallet. Med ledning av felkurvorna nedtill i fig 7 har responskurvorna beräknats. Det framgår av figuren, att den långsammaste responsen ger den snabbaste insvängningen varefter systemet följer insignalen med en viss eftersläpning  $\Delta$ . Den näst långsammaste

———— = samplingsperiod  $T_1$   
----- = " "  $T_2$

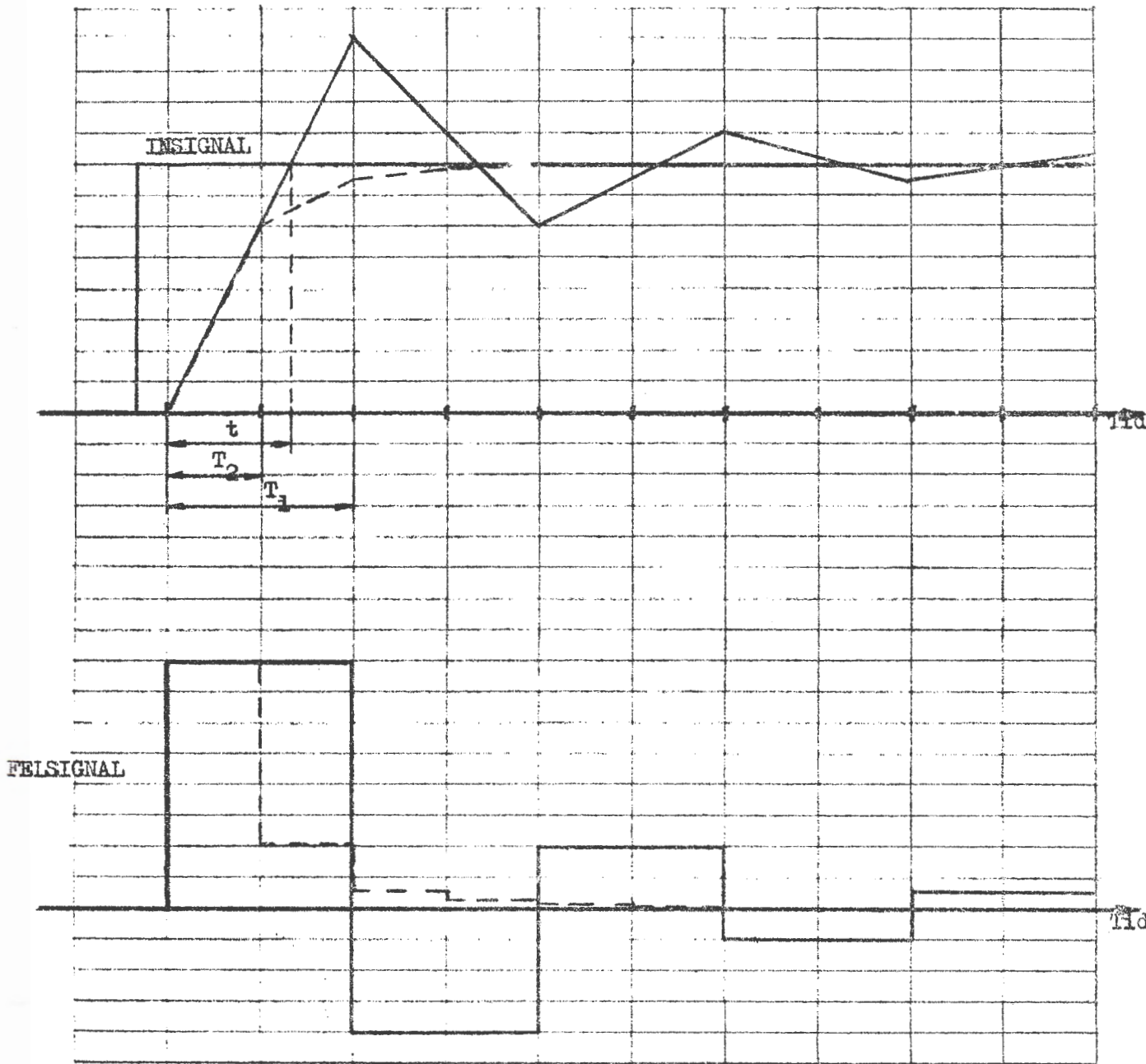


FIG. 6

Transientförlopp i samplat system.



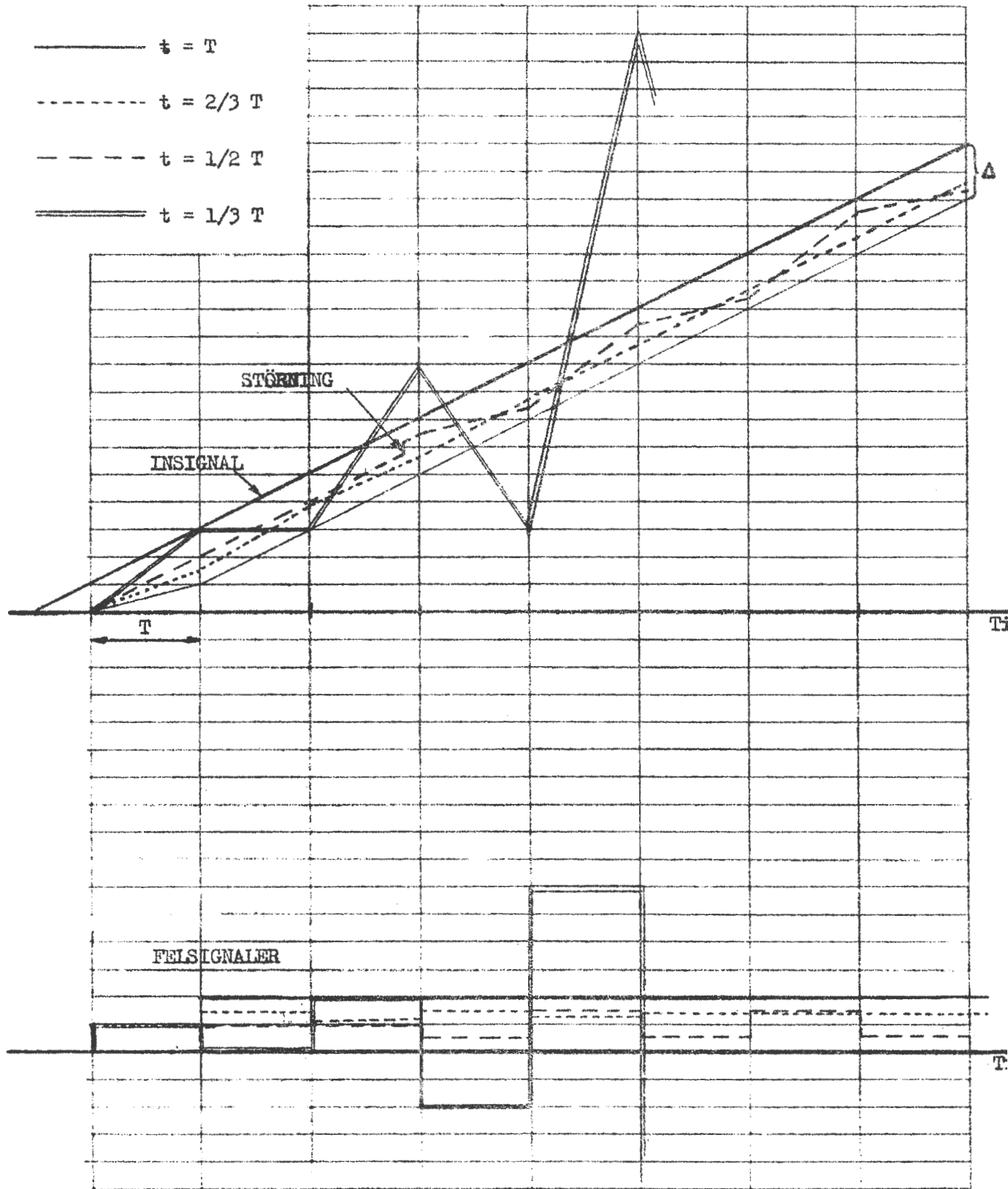


FIG. 7

Transientförlopp i samplat system.

responserna ger en periodisk dämpad insvängning mot en jämn följdning av insignalen. I detta fall är eftersläpningen endast  $2/3 \Delta$ , vilket är naturligt eftersom detta system är 1.5 ggr snabbare än det föregående. Om tidskonstanten minskas till hälften av samplingsintervallet (tredje kurvan), inträder en hackig följdning, som aldrig utslätas som i de två föregående fallen utom i ett labilt tillstånd nämligen då eftersläpningen i samplingsögonblicket är  $\Delta/2$ . I detta fall blir följdningen jämn, men en störning, som ger en avvikelser från denna förelse, för systemet tillbaka till den hackiga följdningen. Om slutligen tidskonstanten görs mindre än halva samplingsintervallet, inträder självsvängning i systemet (fjärde kurvan. Jfr även fig 6, som visar självsvängning för  $T \geq 2t$ ).

Av kurvorna i fig 7 framgår också, att ett långsammare system som naturligt är ger större eftersläpning än ett snabbare. Stor eftersläpning är i många tillämpningar en nackdel, varför relativt snabba system eftersträvas. Detta för emellertid med sig behov av en relativt hög samplingsfrekvens, vilket också är en nackdel i vissa fall då systemet sätter en övre gräns för denna frekvens. Som exempel på ett sådant fall kan nämnas fjärröverföring av styrdata på smalbandslänk (exempelvis telefonlinje). Av bandbreddsskäl är samplingsfrekvensen för utmatning i detta fall begränsad, men ändå kan givetvis krav på liten eftersläpning finnas.

Ett sätt att lösa problemet samplingsfrekvens kontra eftersläpning är metoden med s.k. prediktering (förutsägelse) av systemets utveckling genom interpolation från den redan kända utvecklingen. I ett linjärt föränderligt system enl. fig 7 är det tämligen enkelt att prediktera den framtida utvecklingen genom linjär interpolation. Fig 8 visar, hur detta skulle gestalta sig för en insignal enl. fig 7. Snabbheten hos systemet i fig 8 är lika med den långsammaste i fig 7, dvs tidskonstanten lika med samplingsintervallet. I första samplingsögonblicket känner datautrustningen felet  $\Delta$ . Den enklaste predikteringen är att anta, att insignalen förändras sig lika mycket ytterligare under påföljande samplingsintervall, varför systemfelet i så fall skulle bli  $2\Delta$  vid sampling nr 2 om systemet stode stilla. Det är just detta förväntade fel som skall förekommas, varför det också bör styra systemet (se nedersta styrsignalkurvan i fig 8). Emellertid ligger inte första samplingsögonblicket ett helt samplingsintervall från tiden för insignalens start, varför predikteringen blir felaktig och ett kvarstående fel  $= \Delta$  upptäcks vid sampling nr 2. Ur sampling nr 1 och 2 av insignalen har datautrustningen beräknat, att insignalen skall flytta sig ytterligare  $2\Delta$  till sampling nr 3. Totalt skall systemet så-

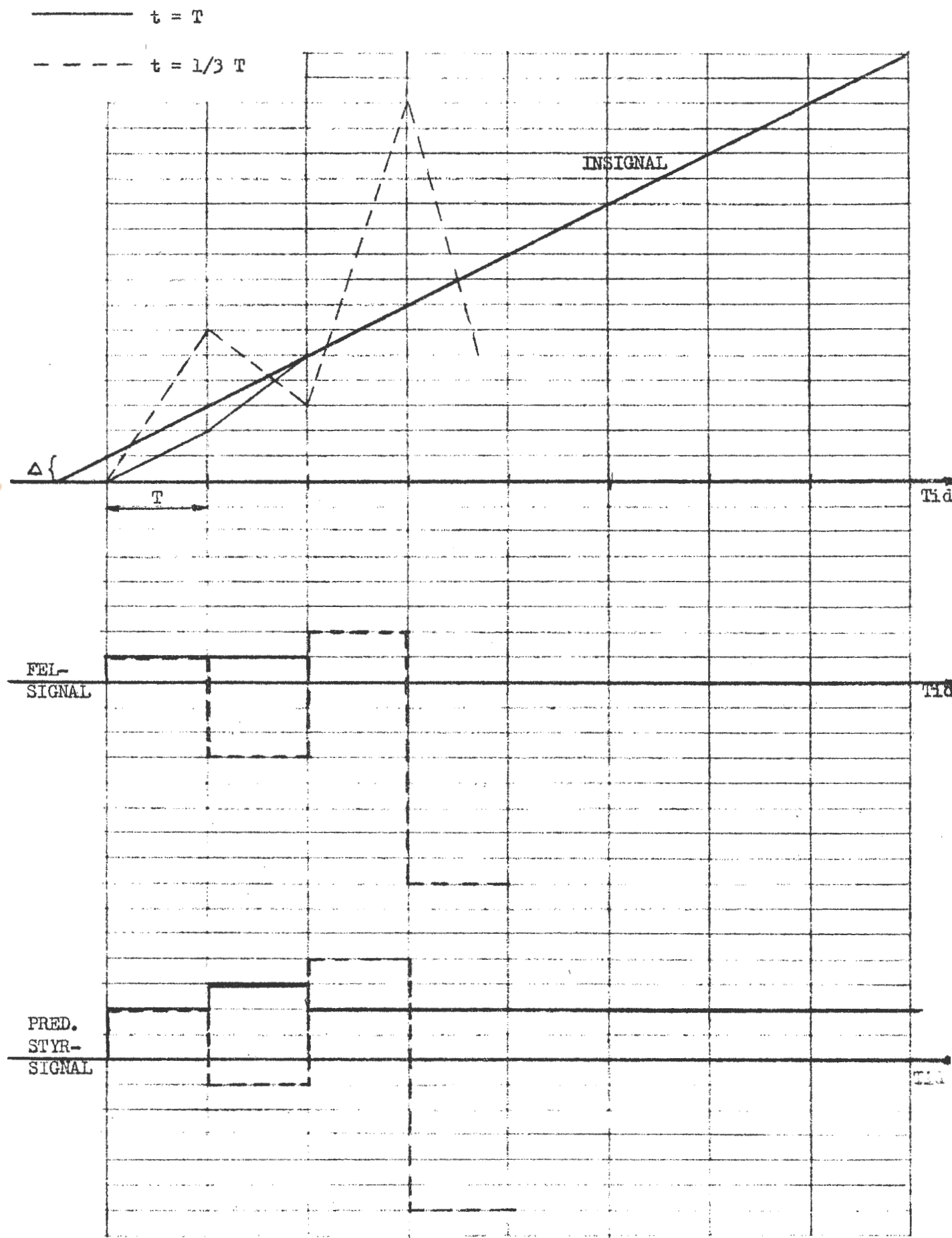


FIG. 8

Transientförlopp i samplat och predikterat system.



ledes ta igen en förflyttning av  $3\Delta$ , vilket bestämmer styrsignalen. Vid sampling nr 3 är felet noll, och följningen blir perfekt med hjälp av styrsignaler bestämda av den predikterade förändringen i insignalen.

Den perfekta följningen illustrerad i fig 8 förutsätter givetvis att systemtidskonstanten är exakt lika med samplingsintervall. I praktiken kan ett sådant antagande inte göras, varför någon eftersläpning eller någon framförhållning alltid måste finnas  $\epsilon n$  förra vid för lång och den senare vid för kort tidskonstant. Vid tillräckligt kort tidskonstant uppträder också självsvängningar (streckade kurvan i fig 8).

Problem på ovannämnda sätt orsakade av missanpassade systemtidskonstanter kan åtgärdas i datautrustningen genom införande av s.k. digitala kompensrande och stabiliserande nät. Detta innebär helt enkelt att in- och utdata utsätts för en matematisk behandling, som ger resultat liknande dem man erhåller i analoga filter. Förutom att den digitala metoden är mycket flexibel genom att den är programmerbar, så är den också den enda tänkbara vid mycket långa samplingsintervall. De digitala filtermetoderna vid samplade system kan vara mycket komplicerade och skall inte närmare beröres här utom med ett par enkla exempel, som kan låta oss ana vilka möjligheter som finns.

I princip går de flesta kompensationsmetoderna ut på att minnas in- och ut signaler från ett varierande antal samplings tidpunkter. Normalt måste båda typerna signaler minnas, varför den ovan behandlade predikteringen inte ger fullt utbildad filterverkan utan svängningar kan uppstå (fig 8 streckad kurva).

Om vi tittar närmare på varför svängning uppstår för den snabbare systemresponsen i fig 8, så är det tydligen p.g.a. att datautrustningen endast mäter och beräknar den förändring, som systemet bör undergå till nästföljande samplingsögonblick samt ger styrsignal efter detta mått. Däremot bedömer datautrustningen inte, hur systemet reagerar på styrsignalen. Om ett snabbt system reagerar alltför kraftigt på en korrigerande styrsignal, resulterar detta endast i en ännu kraftigare styrsignal åt motsatt håll vid nästföljande sampling, och självsvängningen är i gång. Om däremot styrsignalens amplitud kunde dämpas allteftersom systemresponsen är kraftigare, skulle svängningstendenserna kunna upphävas. Systemresponsen kan då exempelvis anges med uttrycket  $R_n = \frac{U_n - U_{n-1}}{S_{n-1}}$ , där  $U_n$  anger utsignalens värde vid samplings tidpunkten  $n$  och  $S_{n-1}$  styrsignalens utmatade värde vid samplings tidpunkten  $n-1$ . Om vidare samplingsfelet vid tidpunkten  $n$  betecknas  $\Delta_n = I_n - U_n$  och

den predikterade förändringen hos insignalen I mellan tidpunkterna n och n+1 betecknas  $\Delta_{pn} = I_{n+1} - I_n$ , kan styrsignalen lämpligen dämpas enligt uttrycket:

$$S_n = (\Delta_n + \Delta_{pn}) / R = (\Delta_n + \Delta_{pn}) \frac{S_{n-1}}{U_n - U_{n-1}}$$

Detta uttryck säger, att styrsignalen vid samplingstidpunkten n skall vara sådan att dels det iakttagna systemfelet  $\Delta_n$  och dels den beräknade systemförändringen  $\Delta_{pn}$  till nästföljande samplingstidpunkt skall kunna tas igen av systemet under samplingsintervallet n — n+1. Därvid används erfarenhet från tidigare samplingsintervaller om hur systemet reagerar för givna styrsignaler. Den ovan givna formeln är skriven i en form, som tydligt anger principen för styrsignalens bildande. Om emellertid uttrycken för  $\Delta_n$  och  $\Delta_{pn}$  sätts in i formeln, kommer denna endast att innehålla in- och utsignalvärden förutom den senaste styrsignalens värde:

$$S_n = (I_n - U_n + I_{n+1} - I_n) \frac{S_{n-1}}{U_n - U_{n-1}} ;$$

$$S_n = \frac{I_{n+1} - U_n}{U_n - U_{n-1}} \cdot S_{n-1}$$

Ur denna formel kan man se, att datautrustningen behöver lagra värdena U och S från föregående samplingsstillfälle samt kunna prediktera I-värdet för nästkommande. Eftersom denna beräkning också sker på grundval av lagrade I-data från tidigare samplingsintervaller, så innebär den i princip möjligheten att lagra tidigare I-värden. I det enklaste fallet, linjär prediktering, är  $I_{n+1} = I_n + I_n - I_{n-1} = 2I_n - I_{n-1}$ , dvs I-värdet från föregående sampling behöver lagras. Totalt betyder detta, att maskinen skall kunna lagra I, U och S från föregående sampling, I och U från innevarande sampling samt utföra beräkningen

$$S_n = \frac{2I_n - I_{n-1} - U_n}{U_n - U_{n-1}} \cdot S_{n-1} \dots \dots \dots (1)$$

Formeln I är i själva verket mer komplicerad till sin innebörd än det enkla utseendet låter förmoda. Den kan nämligen utvecklas, så att även  $S_{n-1}$  uttrycks enbart i I- och U- termer:

$$S_n = \frac{I_{n+1} - U_n}{U_n - U_{n-1}} \cdot S_{n-1} = \frac{I_{n+1} - U_n}{U_n - U_{n-1}} \cdot \frac{I_n - U_{n-1}}{U_{n-1} - U_{n-2}} \cdot S_{n-2} =$$



$$= \frac{I_{n+1} - U_n}{U_n - U_{n-1}} \cdot \frac{I_n - U_{n-1}}{U_{n-1} - U_{n-2}} \cdot \frac{I_{n-1} - U_{n-2}}{U_{n-2} - U_{n-3}} \dots \frac{I_2 - U_1}{U_1 - U_0} \cdot S_0 \dots \dots \dots (2)$$

Man kan alltså se, att formeln i själva verket innebär lagring av alla tidigare samplingsvärden ända tillbaka till de första.  $S_0$  och  $U_0$  är startvärden, som vi skall behandla senare. I ett linjärt föränderligt system, där exakt följning uppstått, blir uttrycket ovan emellertid betydligt enklare, eftersom en linjär funktion kan bestämmas av ett fåtal I- och U-värden. Antag, att exakt följning uppstått efter k:te samplingen. Alla samplade I- och U-värden med index k och högre är då parvis lika, dvs  $I_{k+i} = U_{k+i}$ .

Formel 7 kan då skrivas:

$$S_{k+1} = \frac{I_{k+2} - I_{k+1}}{I_{k+1} - I_k} \cdot S_k$$

Men  $I_{k+2} - I_{k+1} = I_{k+1} - I_k$ , eftersom förändringen är linjär. Alltså är  $S_{k+1} = S_k$ ,  $S_{k+2} = S_{k+1} = S_k$  etc. och

$$S_n = S_k, \dots \dots \dots (3)$$

dvs styrsignalen är konstant från det ögonblick exakt följning uppnåtts.

Formel 1 är ofullständig i det avseendet, att om  $U_n - U_{n-1} = 0$  får man ett obestämt uttryck. Detta fall kan inträffa i ett vilande system, som plötsligt börjar förändra sig. Före starten av denna förändring är styrsignalen ( $S_0$  i formel 2) noll och det först samplade U-värdet ( $U_1$ ) lika med vilotillståndet ( $U_0$ ).  $U_1 - U_0$  såväl som  $S_0$  är då noll, dvs hela uttrycket obestämbart. Eftersom man i detta läge inte kan beräkna systemresponsen, är det enklast att lämna odämpad styrsignal, dvs  $S_0 / (U_1 - U_0) = 1$ . Ett praktiskt sätt att uppnå detta är att sätta responsen  $R_n = (U_n - U_{n-1} + \epsilon) / (S_{n-1} + \epsilon)$ , där  $\epsilon$  är godtyckligt litet, så att responsvärdet normalt avviker från sitt rätta värde mycket lite. I detta fall blir  $0 + \epsilon / 0 + \epsilon = 1$ , vilket önskades. Ovannämnda omväg behövs dock normalt inte i digitala system, eftersom digitala räkneenheter oftast ger resultatet 1 för divisionen 0/0. Formel 1 får därför kvarstå i sin enkla form.

Låt oss nu se, hur vi kan hjälpa upp självsvängningsfallet enligt den streckade kurvan i fig 8 med hjälp av den ovan diskuterade kompensationsmetoden. Resultatet visas i fig 9 (heldragen kurva). Responsen för systemet är som i fig 8 sådan, att tidskonstanten är 1/3 av samplingsintervallet. Tabell 1 visar en uppställning av de parametrar, som bestämmer den ritade kurvan. Alla signalamplituder är där givna i enheter svarande mot 5 mm-sträckor i figuren.

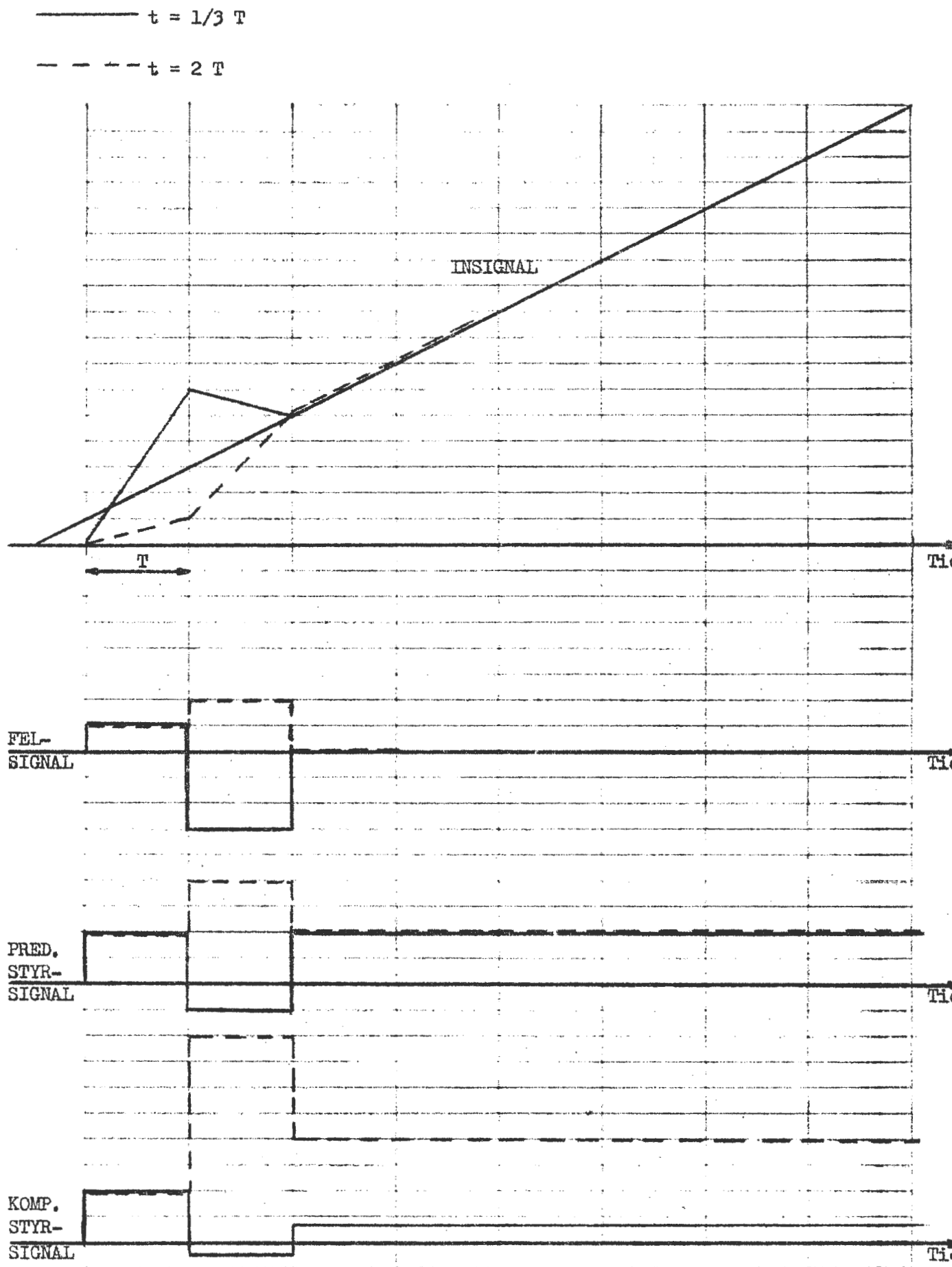


FIG. 9

Transientförlopp vid kompensering.

$t = 1/3 T$

$i$	$I_i$	$I_{i+1} = 2I_i - I_{i-1}$	$U_i$	$S_i = (I_{i+1} - U_i) / (U_i - U_{i-1}) S_{i-1}$	$U_{i+1} - U_i = 3S_i^x$
1	1	2	0	2	6
2	3	5	6	-1/3	-1
3	5	7	5	2/3	2
4	7	9	7	2/3	2

$^x \Delta U_i = 3S_i$ , eftersom tidskonstanten är  $1/3$  samplingsintervall.

$t = 2T$

$i$	$I_i$	$I_{i+1} = 2I_i - I_{i-1}$	$U_i$	$S_i = (I_{i+1} - U_i) / (U_i - U_{i-1}) S_{i-1}$	$U_{i+1} - U_i = 1/2 S_i$
1	1	2	0	2	1
2	3	5	1	8	4
3	5	7	5	4	2
4	7	9	7	4	2

TABELL 1

För uppritning av fig. 9



Av fig 9 framgår, att systemet svänger in sig med maximal snabbhet och sedan följer nivåerna exakt trots systemets snabba respons i förhållande till samplingsintervallens längd.

Som jämförelse visas även i fig 9 (streckad kurva) det fallet, att systemtidskonstanten endast är hälften av samplingsintervallet. Insvingning mot fullständig följlning sker i detta fall lika snabbt som för det ovan diskuterade. Skillnaden är bara den, att styrsignalen i det senare fallet måste vara så mycket kraftigare för att systemet skall hinna följa insignalen.

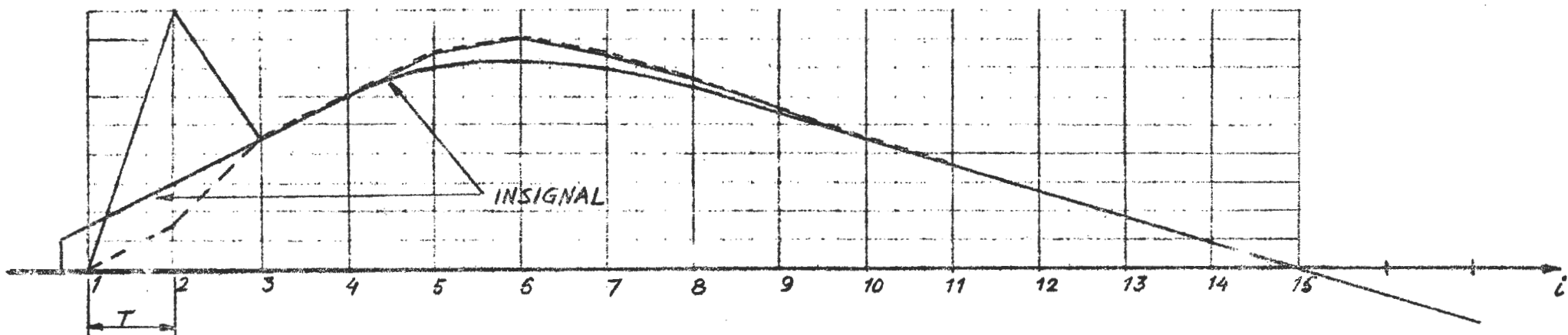
Låt oss slutligen undersöka ett mer generellt fall, där insignalen varierar kontinuerligt förutom att den innehåller en stepfunktion. Ett sådant fall illustreras av fig 10, där två systemfall angivits; ett har en tidskonstant lika med  $1/3$  av ett samplingsintervall (heldragen kurva) och ett en tidskonstant lika med 2 samplingsintervall (streckad kurva). Tabell 2 visar de beräkningsvärden, som ligger till grund för uppritande av kurvorna. Enheten är liksom i tabell 1 en sträcka på 5 mm.

Helt i enlighet med det föregående visar kurvorna i fig 10, hur det snabba systemet svänger över i starten. Sedan följlning uppnåtts, kröker insignalen. Följlningen släpper därvid något, på grund av att insignalen avviker från det varje gång predikterade nya läget medan systemet styrs exakt mot detta predikterade läge. P.g.a. detta senare förhållande spelar det heller ingen roll, vilken tidskonstant systemet har. Följlningen kommer att bli likadan i alla fall, vilket också framgår av fig 10.

#### 2.2.1.6. Inverkan av tidsfördröjlningar

I det föregående har hela tiden antagits, att datamaskinen inte ger någon fördröjlning, utan att en styrsignal som erhållits av en viss mätning matas ut under samma samplingsögonblick som då mätningen görs. I praktiken måste man dock räkna med, att maskinen behöver viss tid på sig att utföra sina beräkningar. Som vi skall se i fortsättningen kan sådana fördröjlningar inverka menligt på systemet, då fördröjlningens värde upptar en väsentlig del av ett samplingsintervall. Stora fördröjlningar kan uppträda exempelvis i följande fall:

- a) datamaskinen används enbart för behandling av ett fåtal funktioner. För att utrustningen skall bli billig, önskar man göra den så långsam, att beräkningstiden utfyller så stor del av samplingsintervallet som möjligt.



—  $t = \frac{1}{3}T$   
 - - -  $t = 2T$

FIG.10

KOMPENSERING I OLINJÄRT SYSTEM.

$t = 1/3 T$

$i$	$I_i$	$I_{i+1} = 2I_i - I_{i-1}$	$U_i$	$S_i = (I_{i+1} - U_i) / (U_i - U_{i-1}) S_{i-1}$	$U_{i+1} - U_i = 3S_i$
1	1,5	3	0	3	9
2	3	4,5	9	-1,5	-4,5
3	4,5	6	4,5	0,5	1,5
4	6	7,5	6	0,5	1,5
5	7	8	7,5	1/6	0,5
6	7,2	7,4	8	-0,2	-0,6
7	6,9	6,6	7,4	-4/15	-0,8
8	6,3	5,7	6,6	-0,3	-0,9
9	5,4	4,5	5,7	-0,4	-1,2
10	4,5	3,6	4,5	-0,3	-0,9
11	3,6	2,7	3,6	-0,3	-0,9 följning

etc.

$t = 2 T$

$i$	$I_i$	$I_{i+1} = 2I_i - I_{i-1}$	$U_i$	$S_i = (I_{i+1} - U_i) / (U_i - U_{i-1}) S_{i-1}$	$U_{i+1} - U_i = 1/2 S_i$
1	1,5	3	0	3	1,5
2	3	4,5	1,5	6	3
3	4,5	6	4,5	3	1,5
4	6	7,5	6	3	1,5

etc. enligt första tabellen.

TABELL 2.

För uppritning av fig.10



- b) processen är inte åtkomlig varken för mätning eller styrning annat än vid bestämda tidpunkter utanför datamaskinens kontroll, varvid tidpunkterna för mätning resp. styrning kan vara olika. Exempelvis kan detta inträffa, om roterande omkopplare (mekaniska eller elektroniska) används för förmedling av mät- resp. styrsignaler mellan datamaskin och olika funktioner. Ett annat fall är, då data förmedlas via telemeterförbindelse varvid denna bestämmer samplingstidpunkterna.
- c) datamaskinen måste vänta på kompletterande data, innan beräkningarna kan utföras. Sådan väntan kan bero på maskinkonstruktionen; exempelvis kan data behöva hämtas från långsamma minnen (magnetband etc.). Väntan kan också bero på att kompletterande data måste samplas i sin tur via samplare, som inte styrs av maskinen själv.

Innan vi går in på att diskutera orsakerna till de besvär fördröjningar kan orsaka, så låt oss se ett par exempel på hur problemen yttrar sig. Fig 11 visar fallet med en rampfunktion som insignal och där ingen prediktering förekommer alltså motsvarande fallet i fig 7. Två fall har angivits det ena (heldragen kurva) med en fördröjning motsvarande ett samplingsintervall och det andra (streckad kurva) med halva denna fördröjning. I båda fallen är systemets tidskonstant lika med samplingsintervallet. Kurvorna visar, att den längre fördröjningen skapar en ständig oscillation medan den kortare ger en stadig följd precis som motsvarande fall i fig 7 men efter en betydligt längre insvängning. Följningen sker med en eftersläpning  $\Delta$ , och omkring denna eftersläpning oscillerar även systemet med den längre fördröjningen. Oscillationen beror tydligen på att då systemet uppnått det rätta följtillståndet (dvs med eftersläpningen  $\Delta$ ) styrsignalen grundar sig på ett tidigare tillstånd dvs en tidigare felsignal. Systemet tillåts därför inte att stanna i det rätta följtillståndet utan svänger över. Vid stora fördröjningar kan då felaktigheten i styrsignalen bli så stor, att över-svängningarna blir tillräckligt kraftiga för att åstadkomma oscillation.

Ett predikterat system enl. fig 8 kommer givetvis att ge samma problem som det ovannämnda vid tidsfördröjning, eftersom båda systemen uppträder likadant så när som på att det ena får ett predikterat tillägg i styrsignalen. Detta gör endast, att eventuella svängningar kommer att ha insignalen i stället för eftersläpningsläget som medelläge. Fig. 12 illustrerar ett fall med en tidsfördröjning lika med ett samplingsintervall och en systemsnabbhet lika stor som för fallen i fig 11.

Att inte heller kompensationsmetoden enligt fig 9 går särskilt bra att an-

———— Fördröjning = T

- - - - - " - - - = 1/2 T

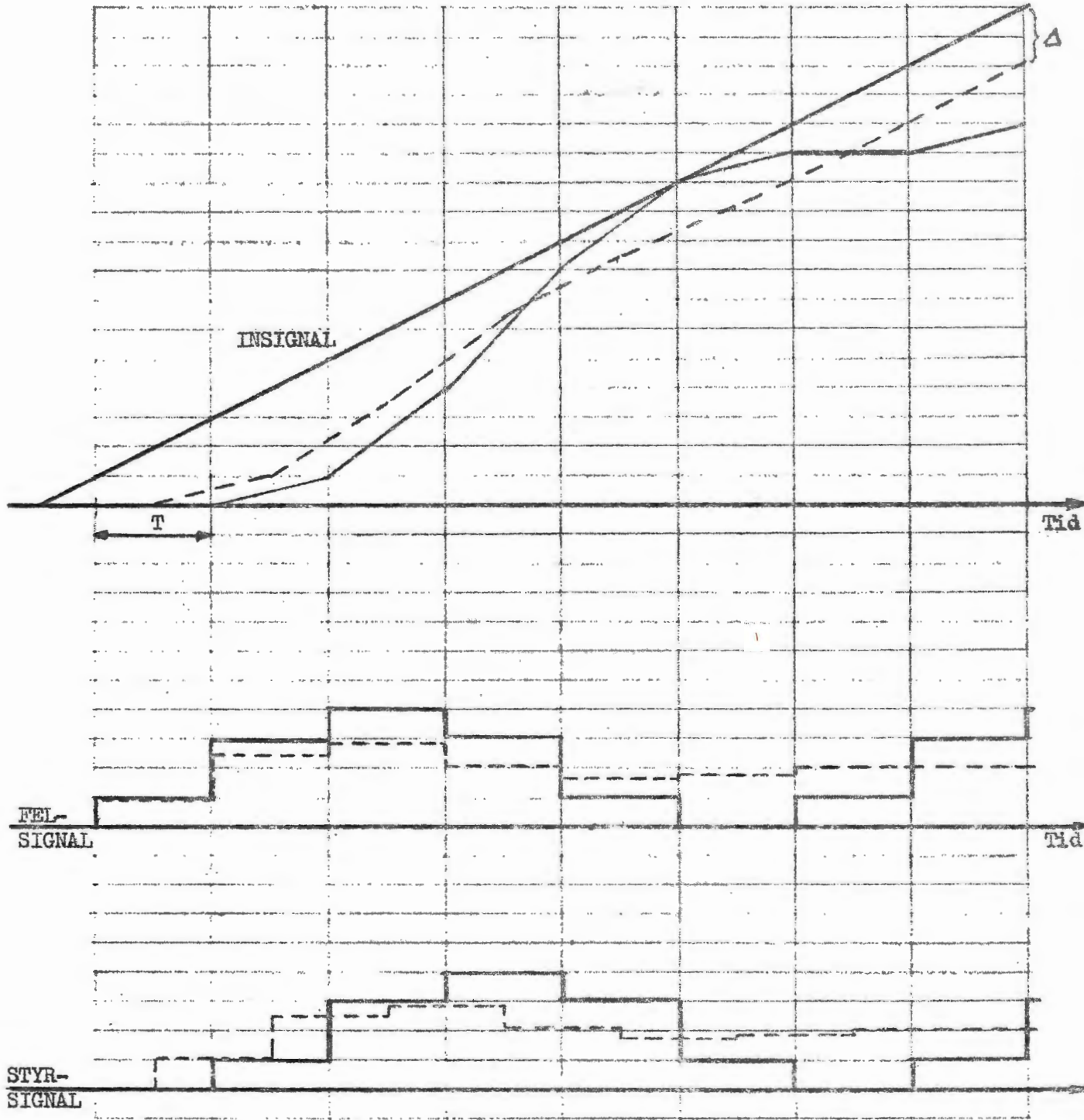


FIG. 11

Inverkan av fördröjning mellan mätning och styrning.

Ingen prediktering.

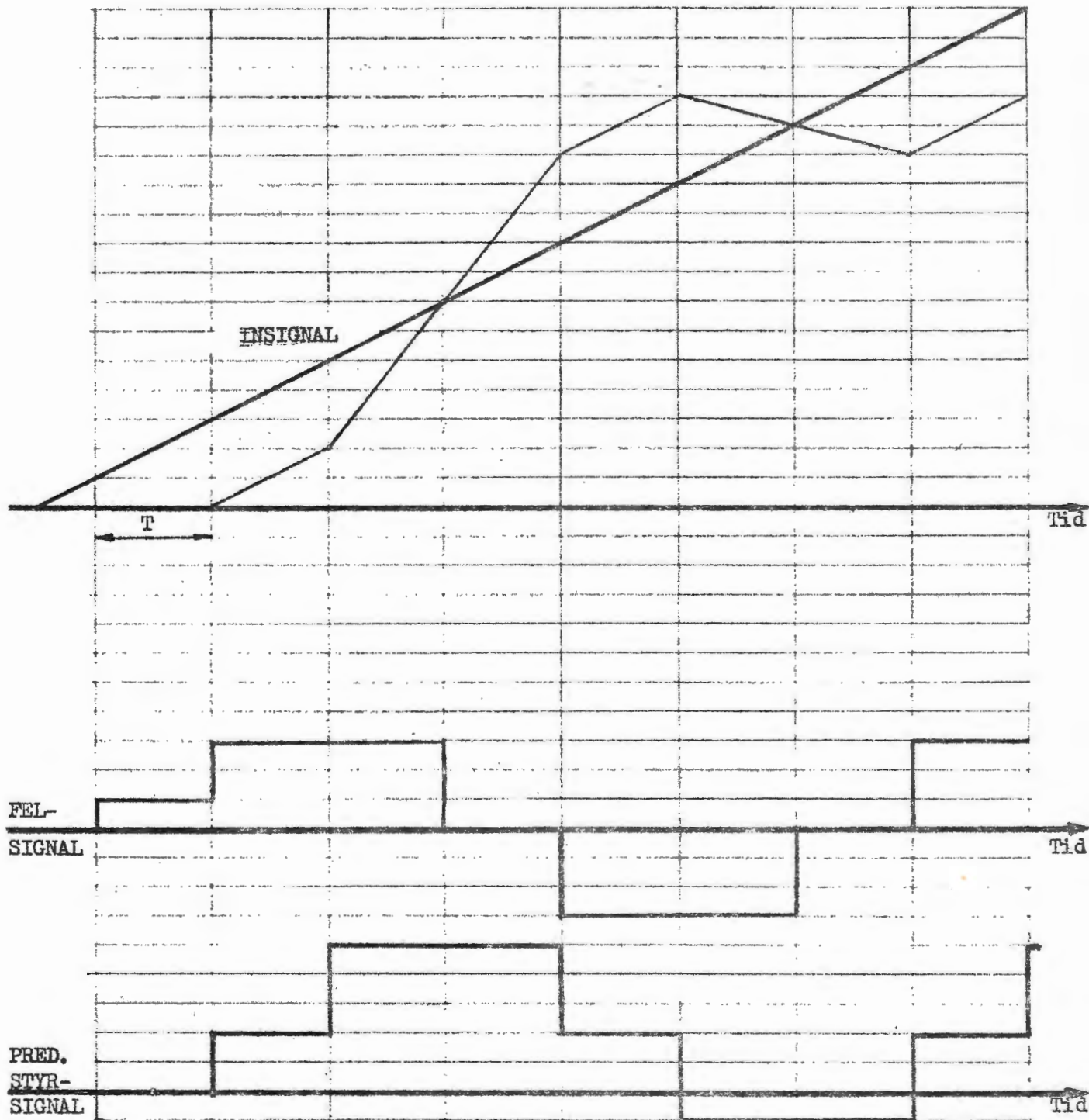


FIG. 12

Inverkan av fördröjning mellan mätning och styrning  
vid prediktering



vända inser man, eftersom en fördröjd respons i systemet kommer att ge till resultat en mycket kraftig styrsignal som kan sätta systemet i svängning. Mätningen av responsen på grundval av systemvärden i samplingsögonblicken blir inte heller riktig, eftersom styrsignalen p.g.a. tidsfördröjningen kan ändra värde någonstans under ett samplingsintervall. Systemförändringen under ett sådant intervall kan därför vara högst olinjär och därför inte alls motsvara systemresponsen.

För att man skall komma till rätta med fördröjningsproblemet, måste man tydligen införa en uppskattning av systemvärdena vid de tidpunkter, då styrdata matats ut dvs vid samplingstidpunkterna för styrdata. Likaså får man försöka prediktera fortsatt inverkan av styrdata, som redan matats ut men inte givit full verkan fram till den tidpunkt då en mätsampling görs. Den förra åtgärden syftar till att ge de systemtillstånd, mellan vilka systemet rör sig rätlinjigt, och den sista förhindrar, att responsen felaktigt beräknas för låg så att styrdata förstärks med eventuell instabilitet som följd.

Låt oss betrakta ett visst samplingsögonblick  $n$ , där systemtillståndet (utsignalen) uppmätts till värdet  $U_n$  och samtidigt insignalen har värdet  $I_n$ . Fördröjningen i utmatningen betecknas  $d$ , som antas vara mindre än samplingsintervallet  $T$ . Jämförelsen mellan  $U_n$  och  $I_n$  skall som förut resultera i en styrsignal  $S_n$ , som modifierats med hänsyn till systemresponsen. För beräkning av den senare antar vi att vi kan beräkna utsignalen vid den tidpunkt, så  $S_{n-1}$  matades ut, dvs vid tidpunkten  $n-1+d$ . Detta tillstånd kallar vi  $U_{d(n-1)}$ . Om beräkningen av  $U_{d(n-1)}$  stämmer med det verkliga förhållandet, är systemresponsen tydligen sådan, att förändringen  $U_n - U_{d(n-1)}$

åstadkommit under tiden  $T-d$  av styrsignalen  $S_{n-1}$ . Alltså är  $R = \frac{T}{T-d}$ .

$\cdot (U_n - U_{d(n-1)}) \cdot \frac{I}{S_{n-1}}$  räknat på ett helt samplingsintervall. Utsignalen vid

utmatningstidpunkten för  $S_n$  (vid tiden  $n+d$ ) kommer då att vara  $U_{d(n-1)} + \frac{T}{T-d} (U_n - U_{d(n-1)})$  och vid samma tidpunkt kommer felet mellan insignal och

utsignal att vara  $I_n + \frac{d}{T} (I_n - I_{n-1}) - \left[ U_{d(n-1)} + \frac{T(U_n - U_{d(n-1)})}{T-d} \right]$ , efter-

som även insignalen har rört sig under tiden  $d$  och med ett tänkt belopp linjärt beroende av tidigare värden. Detta fel plus förändringen i  $I$  skall hämtas igen, tills  $S_{n-1}$  utmatas vid tidpunkten  $n+1+d$ . Tiden för denna föränd-

ring är tydligen ett samplingsintervall och således förändringen lika med  $I_n - I_{n-1}$ , om förändringen antas fortfarande att vara linjär. Alltså skall systemet totalt förändra sig värdet

$$I_n + \frac{d}{T} (I_n - I_{n-1}) + I_n - I_{n-1} = \left[ U_{d(n-1)} + \frac{T(U_n - U_{d(n-1)})}{T-d} \right] =$$

$$= I_n + \left(1 + \frac{d}{T}\right) (I_n - I_{n-1}) - \frac{T U_n - d U_{d(n-1)}}{T-d}, \text{ vilket}$$

dividerat med ovan angivna respons ger styrsignalen  $S_n$ .

$$S_n = \frac{I_n + \left(1 + \frac{d}{T}\right) (I_n - I_{n-1}) - \frac{T U_n - d U_{d(n-1)}}{T-d}}{\frac{T}{T-d} (U_n - U_{d(n-1)})} \cdot S_{n-1} \dots \dots \dots (4)$$

Ovanstående formel har använts för att i fig 13 (heldragen kurva) illustrera följning av samma olinjära insignal som i fig 10. Fördröjningen  $d$  har därvid satts till en halv samplingsperiod. Systemtidskonstanten är lika med ett samplingsintervall.

Formeln (4) får följande utseende för det fall, att fördröjningen är större än eller lika med ett samplingsintervall

$$S_n = \frac{I_n + \left(1 + \frac{d}{T}\right) (I_n - I_{n-1}) - \frac{T U_{n+1} + (d-T) U_{d(n-1)}}{2T-d}}{\frac{T}{2T-d} (U_n - U_{d(n-1)})} \cdot S_{n-2}$$

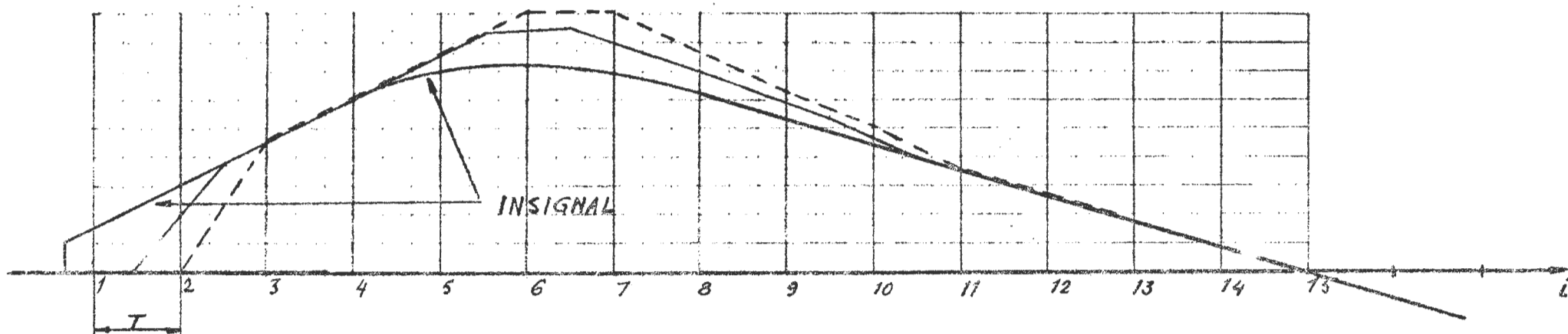
För det vanligaste fallet, att  $d=T$  blir denna formel enklare:

$$S_n = \frac{3I_n - 2I_{n-1} - U_{n+1}}{U_n - U_{n-1}} \cdot S_{n-2}$$

$$S_n = \frac{3I_n - 2I_{n-1} - U_n}{U_n - U_{n-1}} \cdot S_{n-2} - S_{n-1} \dots \dots \dots (5)$$

Formel (5) har använts till att beräkna den streckade systemkurvan i fig 13. De båda kurvorna i denna figur jämförda med kurvorna i fig 10 visar, att ju större tidsfördröjningen är desto större blir också följningsfelet vid olinjär variation i insignalen. Dessutom är ett mer fördröjt system senare i inhämtning av följningen, vilket är helt naturligt.

Kompensationsförfarandena vid fördröjningar i styrningen kräver av maskinen ytterligare minnesmöjligheter jämfört med fallet utan fördröjningar. För ett system enligt formel (5) måste maskinen alltså minnas två av de före-



— FÖRDRÖJNING =  $\frac{1}{2} T$  } SYSTEMTIDSKONSTANT =  $T$   
 - - - " =  $T$

FIG. 13

INVERKAN AV TIDSFÖRDRÖJNING

I SAMPLAT KOMPENSERAT SYSTEM



gående styrsignalerna och i ett system enligt formel (4) ett beräknat värde hos  $U$  vid en styrdatasampling. Detta värde  $U_{d(n-1)}$  beräknas kontinuerligt enligt formel (4) i sista termen i täljaren, eftersom  $U_{d(n-1)} = \frac{TU_{n-1} - dU_{d(n-2)}}{T-d}$ . Maskinen skall alltså minnas denna predikterade storhet från samplingsögonblick till samplingsögonblick. Formel (4) innebär också ett väsentligt försvårande av beräkningsarbetet.

Ett system motsvarande formel (4), dvs där styrdatautmatningen ligger mellan två samplingsögonblick, är mycket känsligt för störningar. Systemets tillstånd, då en styrsignal utmatas, kan ju enligt ovan endast beräknas och inte kontrolleras genom mätningar. Eftersom den beräknade systemresponsen grundar sig på detta tillståndsvärde, blir dennas riktighet beroende av riktigheten i det beräknade tillståndet och kan alltså förryckas om systemet av någon anledning avviker från det antagna tillståndet. Ett sådant förhållande leder till en instabilitet, som för maskinen ter sig som instabilitet i systemresponsen. Genom antagandet, att denna respons endast kan variera mycket långsamt, kan systemet stabiliseras genom medelvärdesbildning av ett antal beräknade responser. Detta för emellertid med sig ännu större krav på maskinen och undviks helst. Om därför en väsentlig fördröjning i systemet är nödvändig, är det ofta lämpligare att göra denna lika med ett samplingsintervall, eftersom systemtillståndet då alltid kan mätas samtidigt som systemet utsätts för en  $f$  rändring. Eventuella avvikelser i systemet upptäcks då snabbt och responseräkningen kan alltid grundas på sanna systemmätvärden.

#### 2.2.1.7. Inverkan av brus

I det föregående har det alltid antagits, att systemet har möjlighet att uppnå god följdning då insignalen förändrar sig rätlinjigt. Många typer av insignaler förändrar sig emellertid aldrig exakt rätlinjigt utan i bästa fall med vissa fluktuationer omkring ett rätlinjigt medelvärde. Detta gäller exempelvis, om insignalen härrör från mätningar av en föränderlig storhet. Antingen kan därvid mätstorheten fluktuera eller också fluktuerar det digitala mätvärdet p.g.a. kvantiseringsfel. Det sistnämnda förekommer som regel vid processregleringar och förtjänar därför att beröras något närmare. Kvantiseringsfel är som bekant de fel man erhåller mellan det exakta värdet av den uppmätta storheten och motsvarande digitala mätvärde. P.g.a. att det senare har ändlig upplösning, kan det inte alltid återge det exakta mätvärdet utan i stället det digitala värde som ligger närmast mätvärdet.

Kvantiseringsfelets maximala storlek är plus eller minus halva kvantiseringssteget, om mätutrustningen är optimalt inställd.

Fig 14 illustrerar grovt schematiskt, vad som kan inträffa p.g.a. kvantiseringsfel. Insignalen är en rampfunktionen och mätutrustningen har en upplösning på en ruta i vertikal led. Punkterna omkring insignalen anger närmaste digitalvärde vid varje sampling. Låt oss i första hand studera systemsvaret enligt den heldragna kurvan. Kurvorna nedtill anger dels uppmätt felsignal, dels predikterad förändring lika med skillnaden mellan de två senast uppmätta insignalvärdena och dels styrsignalen som summan av felsignal och predikterad förändring (systemtidskonstant = samplingsintervall). För åskådliggörelse skull har det inkonsekvent antagits, att styrsignalen har finare upplösning än mätvärdena. Insignalen blir här behäftad med rätt kraftiga fluktuationer, som systemet söker följa med viss eftersläpning, varvid avvikelserna från insignalen kan överstiga mätnoggrannheten.

En förbättring av systemet kan åstadkommas med den metod, som illustreras av den streckade kurvan över systemsvaret. Här tillåts systemet inte följa fluktuationerna i insignalen till 100% utan förändringarna i styrsignalen dämpas ner till hälften:

$$S_n = S_{n-1} + 1/2 [ (\Delta_n + \Delta_{pn})/R - S_{n-1} ]$$

$$S_n = 1/2 [ S_{n-1} + (\Delta_n + \Delta_{pn})/R ] \dots\dots\dots (6)$$

Korrektionen med hänsyn till den uppmätta förändringen hos insignalen är alltså i detta fall reducerad till hälften. Den reducerade korrektionens inverkan på systemet är ungefär densamma som för ett lågpasfilter. En typisk insvängning kan också märkas i början av förloppet i figuren.

Fig 14 kan illustrera både det fall, då brus erhålles av kvantiseringen och det då bruset finns i insignalen redan från början. I det senare fallet kan felet bli större, men principen är ändå densamma.

Brus av det slag, som här berörts, tillsammans med fördröjningar, som behandlades i föregående avsnitt, kan om båda är av tillräcklig storlek göra systemet instabilt av skäl som redan diskuterats (se sista stycket i föregående avsnitt). De metoder, som här angivits för stabilisering av ett system, har alla karaktären av filterfunktioner, där systemutvecklingen bedöms ur ett flertal tidigare systemtillstånd. Normalt för lågpasfilter är, att de inför en viss tröghet i systemet, dvs varje olinjär förändring i insigna-

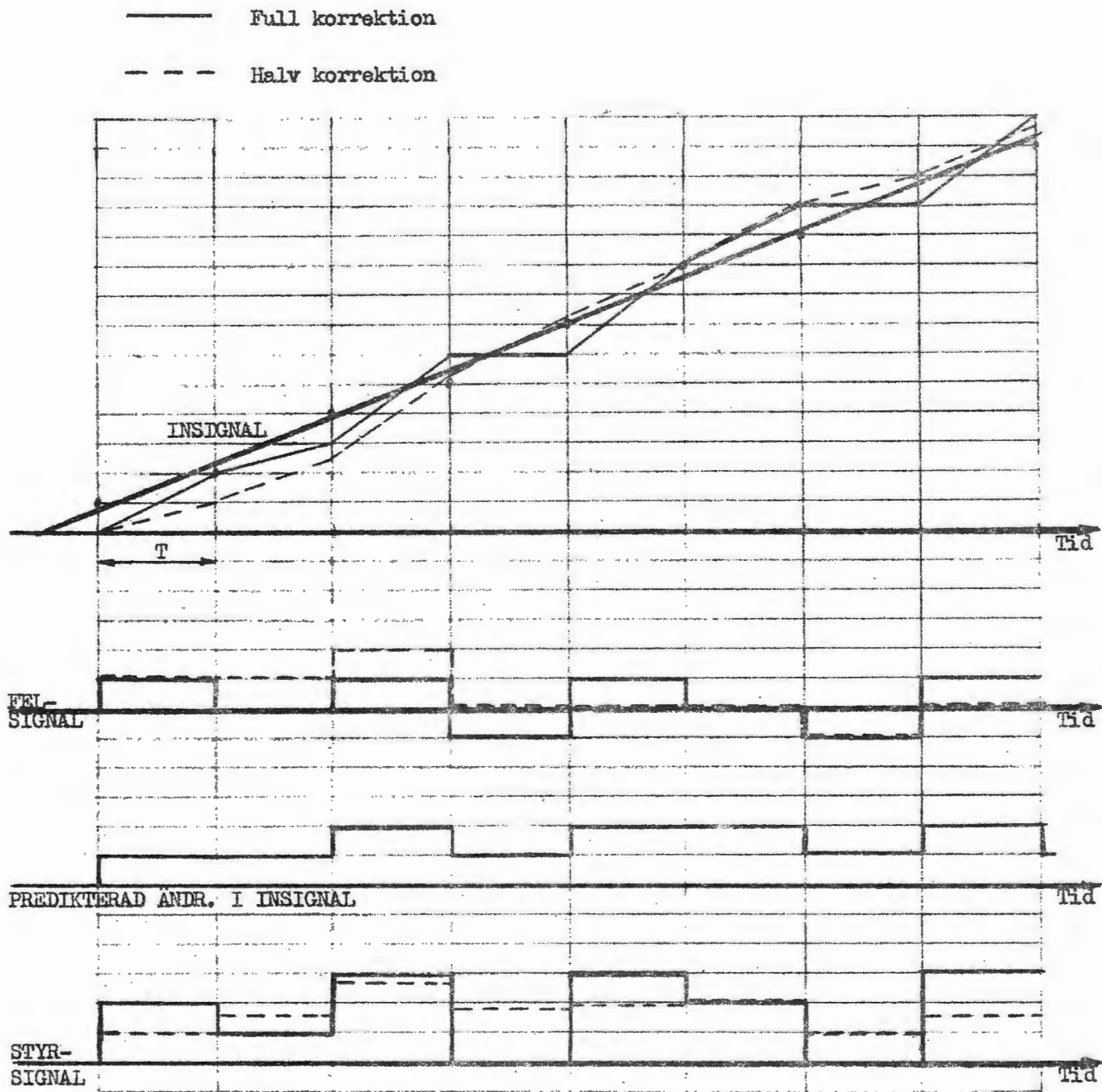


FIG. 14

Inverkan av kvantiseringsbrus



len ger till resultat en transient i utsignalen, så att den senare tar viss tid på sig att åter ansluta sig till den förra. Denna inverkan förmärks i stabiliseringstypen enligt ekv. 6, där insvängningen är logaritmisk precis som i ett analogfilter. I typen enligt ekv. 1 (jfr exvis fig 10) uppstår också transienter, som dock svänger in betydligt snabbare p.g.a. prediktering. Den tröghet man erhåller i ett sådant system, är främst betingad av samplingen. Ju högfrekventare samplingen är, desto mindre är trögheten. Den sistnämnda kompenseringstypen är emellertid ofullkomlig såtillvida, att brus kommer att skapa vissa stokastiska fel mellan utsignal och insignal p.g.a. att ingen korrektionsdämpning finns (jfr heldragen kurva för systemsvaret i fig 14). Om en sådan dämpning införes, blir följningsfelet mindre vid likformig förändring i insignalen. Vid olikformig förändring förvärras emellertid den av korrektionsdämpningen orsakade trögheten avvikelsen under transienten. Vid olikformigt föränderliga system måste man alltså kompromissa mellan brus känslighet och manöverkänslighet, dvs felen p.g.a. brus måste vägas mot felen p.g.a. dålig följsamhet, så att ett optimalt system erhålles. Problemet har behandlats teoretiskt i litteraturen och berörs ej närmare här. Den intresserade hänvisas till litteraturförteckningen i slutet.

#### 2.2.1.8

##### Läges- och kursfel

I det föregående har hela tiden den reglerade processen antagits vara ett system, som inte kan ändra tillstånd diskontinuerligt. Därför har datamaskinen beräknat styrsignalen så, att ett iakttaget fel mellan in- och utsignal skall hämtas igen under ett samplingsintervall. Emellertid kan man givetvis också tänka sig att göra en snabbare lägeskorrektur, varefter systemet styrs vidare enbart efter den predikterade förändringen hos insignalen (se fig 15). Båda reglerfallen kan ha sitt berättigande. I fig 16 kan man se, att den snabba korrekturen ger mindre medelavvikelse från insignalen än den långsamma vid olinjär förändring. Snabbkorrekturen ger alltså högre manöverkänslighet och därmed bättre följsamhet. Den nedre delen av fig 16 visar emellertid, att systemets riktning under transienten kan avvika rätt kraftigt från insignalens riktning. Ett sådant förhållande kan försämra följningen avsevärt, om av någon anledning en mätsampling uteblir under olinjäritetsperioden. I detta fall är den långsamma korrekturen bättre, eftersom den ger systemet en senare riktning. Det är ytterligare en nackdel, som vidlåder snabbkorrektionsmetoden, och det är, att eventuellt brus i insignalen vidarebefordras till utsignalen plus att den senare erhåller brus redan vid olinjär förändring (fig 16). Detta kan vara skadligt, om ut-



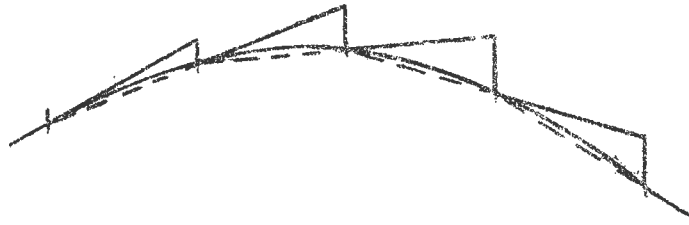
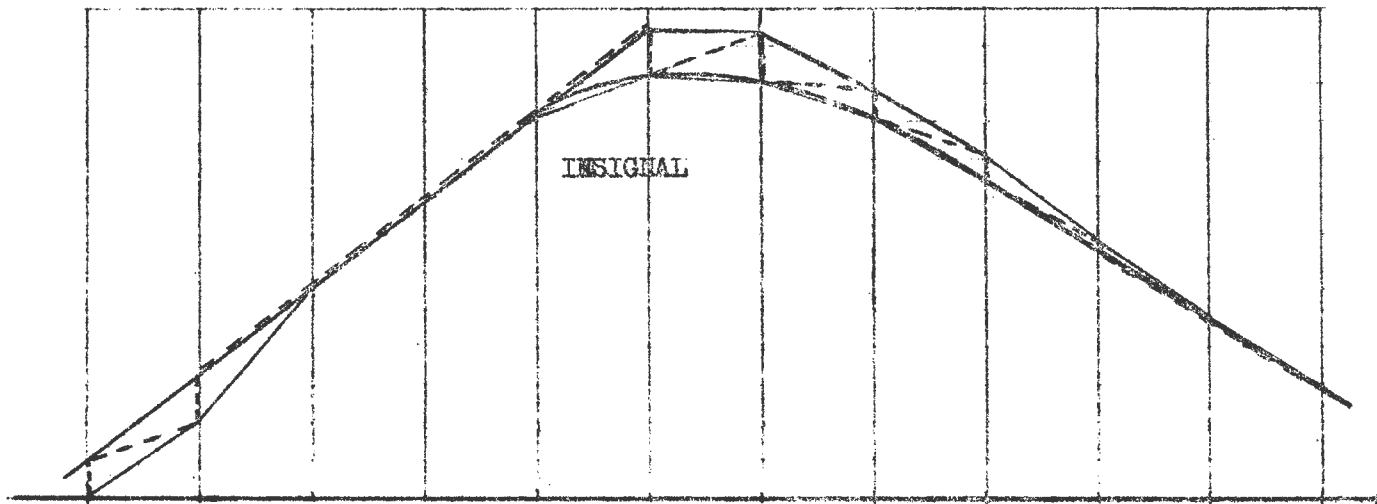
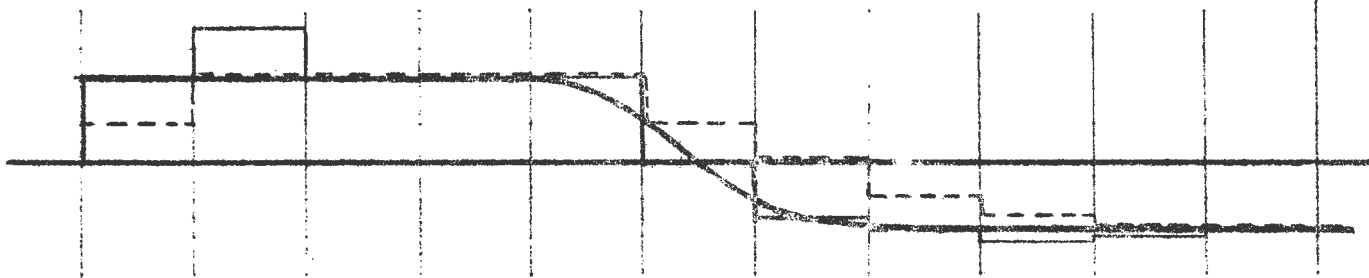


FIG. 15

— Lägeskorrektions tid =  $T$   
- - - " = 0



Positionskurva



Hastighetskurva

FIG. 16

Inverkan av korrektionssnabbhet

signalen i sin tur skall användas som informationskälla.

#### 2.2.1.9

##### Följning

Vi har i det föregående studerat stabilitetsproblem i reglerade system under olika betingelser. Studierna har för åskådlighetens skull gällt enstaka kontrollerade variabler, där det inte antagits vara några större problem att mäta deras tillstånd. Beroende på karaktären hos systemsnabbhet, sampling, tidsfördröjningar och brus har olika goda resultat av följsamhet mellan in- och utsignaler erhållits. Även vid mindre god följsamhet kan man säga, att systemet varit under kontroll så länge det inte uppstått självsvängningar. Det finns emellertid andra tillämpningar, där systemkontrollen helt kan äventyras även vid måttliga brister i följsamheten. Detta gäller bl.a. i system, där flera variabler skall mätas av samma mätorgan utan att det senare har förmåga att avgöra, vilka variabler som har vilka mätvärden. I ett sådant system är det reglerorganets (datamaskinens) uppgift att sortera ut de mätvärden, som hör till resp. variabler, för att en systemövervakning skall vara möjlig. Vi kan här kalla ett sådant system för ett "följsystem", eftersom en av reglerorganets uppgifter är att följa de övervakade storheterna bland mängden av hopblandade variabelvärden. Det mest kända exemplet på ett följsystem torde vara automatisk lufttrafikövervakning medelst radar. Här nedan skall beröras några aspekter på ett sådant system för att ge en praktisk belysning åt de tidigare diskuterade stabilitetsproblemen och främst då brus- och manöverkänslighet.

Dagens starkt ökade flygtrafik och höga flyghastigheter håller på att göra automatisk lufttrafik till en nödvändighet, eftersom mänskliga operatörer dels har en begränsad kapacitet och dels är benägna att falla offer för den mänskliga faktorn i form av felbedömningar, som kan vara katastrofala. Kapaciteten belastas främst av arbetet att övervaka en växande mängd flygplan och att samordna deras trafik, så att dels tidtabeller kan hållas och dels katastrofer i form av exempelvis kollisioner kan undvikas. Inom militärt luftförsvaret har automationsbehovet kanske accentuerats ännu kraftigare och då främst p.g.a. de ökade flyghastigheterna, som skurit ner de tillgängliga förvarningstiderna så mycket att mänsklig bedömning av ett luftläges utveckling blir alltför långsam om inte hjälp kan erhållas från automatiska hjälpmedel.

De uppgifter, som kontrollenheten (datamaskinen) har i ett luftövervakningssystem, är primärt att övervaka de olika flygplanens rörelser och bedöma deras färd i förhållande till flygkorridorer, andra flygplan, flygfält, an-

fallsmål etc. För att denna övervakning skall kunna göras, måste maskinen införskaffa en mängd uppgifter om flygplanen i form av läges-, kurs- och fartdata, identifiering etc. och ofta sammanställa dessa med andra typer av data som väder- och basinformationer, order från operatörer etc.

Det är just vad gäller läges- och rörelseparametrarna hos flygplanen, som datamaskinens följningsuppgifter uppträder i ett automatiskt system. Lägesuppgifterna för flygplanen kommer in via en eller flera rundspanande radarstationer, som i det fallet fungerar som mätorgan eller processorvandler. Självklart kan en sådan radar dock inte hänföra de olika mottagna radarekonorna till vissa flygplan, utan lämnar alla data via samma signalutgång och i samma principiella form. Det är alltså kontrollenhetens uppgift att bland mängden information från radarn följa den, som gäller för ett specifikt flygplan. Datamaskinens första uppgift är alltså att sortera radarinformationen på de olika flygplanens förflyttningsbanor (tracks). Denna uppgift kan maskinen principiellt lösa genom att för varje flygplan beräkna en bana ur erhållna positionsuppgifter från radarn samt extrapolera (prediktera) denna bana till en förväntad position för planet vid kommande radarvarv. Det erhållna eko, som då sammanfaller med det förväntade läget för ett visst flygplan, antas höra till dettas bana och prediktering med positionsjämförelser kan fortsätta varv efter varv, varvid en följning av flygplanet erhålles. Detta ideella fall försvåras av en hel del faktorer, varav följande är de viktigaste:

- flygplanen flyger inte rätlinjigt utan svänger ibland
- p.g.a. reflexer, strålavböjningsfel och fel i radarn erhålls in-exakta positionsbestämmelser i radarn (mätbrus)
- radarekon kan bortstöras, varvid mätningar uteblir vissa radarvarv
- störningar eller andra flygplan ger mätvärden, som kan sammanblandas med de önskade.

Fig 17 visar, hur en flygbana kan se ut på ett PPI, då ovannämnda faktorer inverkar. Då en mänsklig operatör följer ett mål på PPI, försöker han hitta sådana ekon, som tillsammans med andra gamla ekon kan bilda en acceptabel bana på PPI:t. På samma sätt får datamaskinen i ett automatiskt system söka arbete, dvs den får associera varje nyinmätt måleko med lagrade uppgifter om äldre sådana för att se om en bana kan bildas, som vad gäller flyghastigheter och kursförändringar rimligtvis kan härröra från ett flygplan. Ett sådant associeringsförlopp kan vara oerhört omfattande, om luftrummet innehåller många flygplan och störningar. Av praktiska skäl begränsar man därför ofta associeringsarbetet på så sätt, att varje i maskinen följt bana ges en mätlucka som placeras över den predikterade mätpositionen. Fig 18 visar princi-

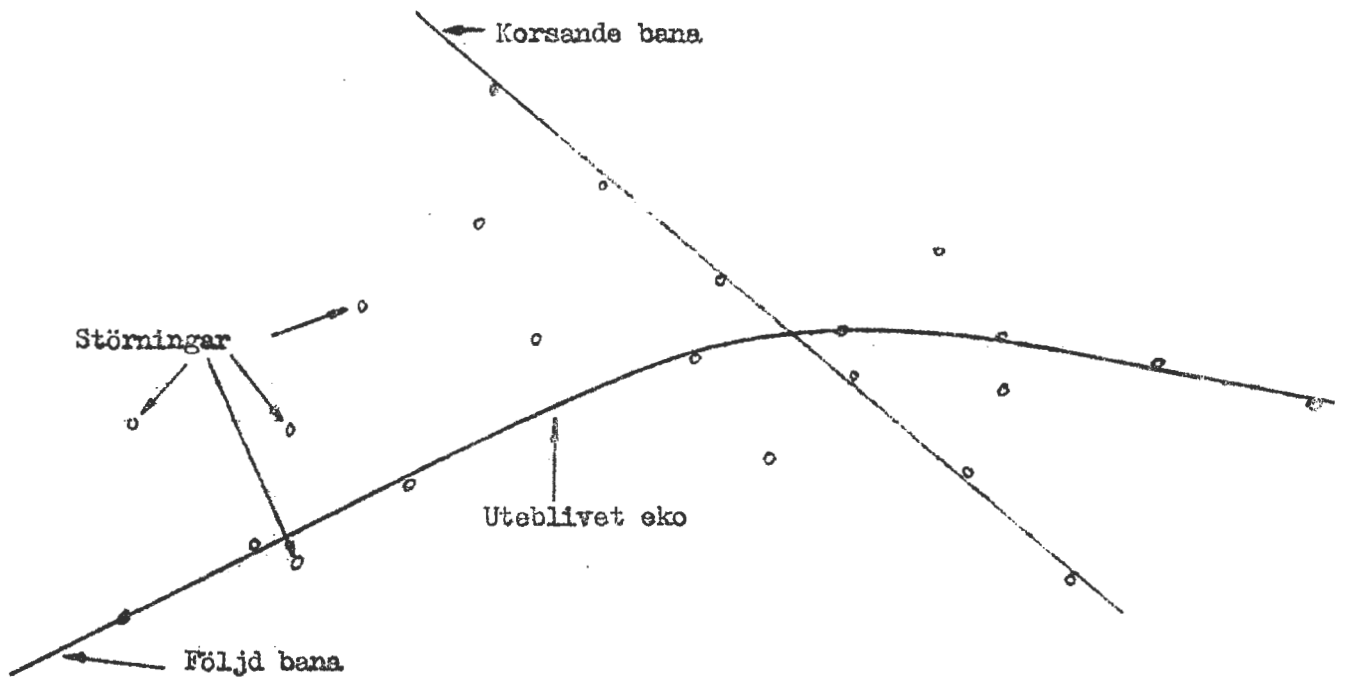


FIG. 17

Komplikationer vid målföljning

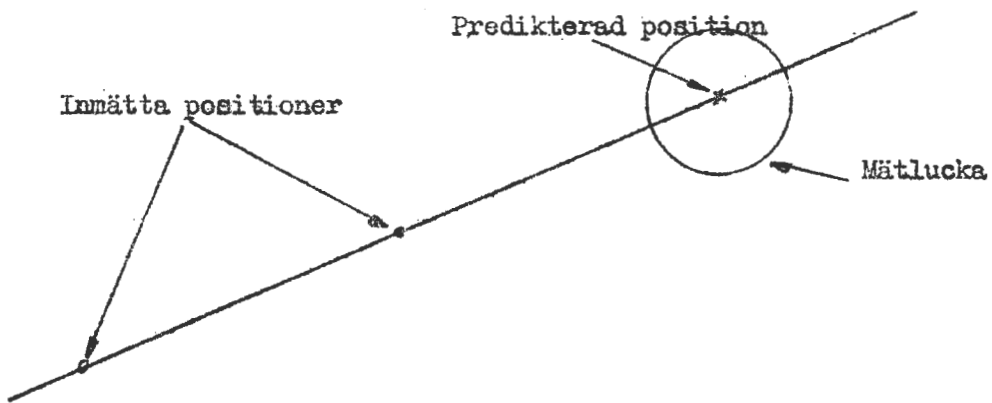


FIG. 18

Luckbildning



pen för detta förfarande. Luckans storlek bestäms av den osäkerhet man beräknar att predikteringen har p.g.a. undanmanövrer, mätbrus och avvikelser från en konstant antennrotationstid. Ett eko, som mottas inom luckan, antas automatiskt tillhöra den flygbana som luckan tillhör, varför associeringen blir tämligen enkel. Eventuella avvikelser mellan uppmätt ekoposition inom luckan och den predikterade positionen används sedan till att ge banan nödiga korrekitioner, så att följningen kan upprätthållas vid avvikelser mellan sann och predikterad flygbana. Den ovannämnda associeringen blir givetvis något mer komplicerad, om flera ekon tas emot inom mätluckan. Mer invecklade bedömningar får då göras, som grundar sig på kännedom om eventuella andra korsande flygbanor, brusförhållanden, förhistoria hos den aktuella banan etc. En nackdel med luckmetoden är naturligtvis, att följning av en bana inte kan startas automatiskt utan måste initieras av operatör, som exempelvis kan mata in startdata till datamaskinen efter upptäckt av flygbana på PPI.

Sedan associeringen väl är genomförd, dvs då processvariabelns mätvärde utvalts, skall som nämnts den predikterade flygbanan (dvs den predikterade processutvecklingen) korrigeras med hänsyn till detta mätvärde. Vi är då tillbaka till vad vi tidigare diskuterat om övervakning och styrning. Följningen sker i detta fall i ett två- eller tredimensionellt system, men låt oss för enkelhets skull endast betrakta en enda koordinat exempelvis X. Antag som i fig 19, att vid två mätningstillfällen  $t_1$  och  $t_2$  med samplingsintervallet T (= radars effektiva rotationstid, vilken kan variera p.g.a. målrörelser etc.) X-värdena  $X_1$  och  $X_2$  uppmätts. Därur har sedan hastighetskoordinaten  $\dot{X}_2 = (X_2 - X_1)/T$  beräknats. Till tidpunkten  $t_3$  predikteras lägesvärdet  $X_{P3} = X_2 + T \cdot \dot{X}_2$ . Snellertid uppmätts vid  $t_3$  det sanna positionsvärdet  $X_{M3}$ . Maskinen upptäcker då ett fel  $\Delta X_3 = X_{M3} - X_{P3}$ . Samtidigt beräknar den en ny hastighetskoordinat  $(X_{M3} - X_2)/T = \dot{X}_2 + \Delta X_3/T$ . Som tidigare diskuterats, kan man nu antingen låta korrigera systemet fullständigt (snabb korrektion) eller endast delvis (filtrering eller glättning; jfr ekv 6 i föreg. avsnitt). Detta kan då gälla både läges- och hastighetskorrektionen. Vi skriver därför följande allmänna formler för en glättad korrektion:

$$\begin{cases} X_3 = X_{P3} + a \cdot \Delta X_3 & \dots\dots\dots(7) \\ \dot{X}_3 = \dot{X}_2 + b \cdot \Delta X_3 / T & \dots\dots\dots(8) \end{cases}$$

a och b är här de s.k. filter- eller glättningskonstanterna. Ju mindre värden

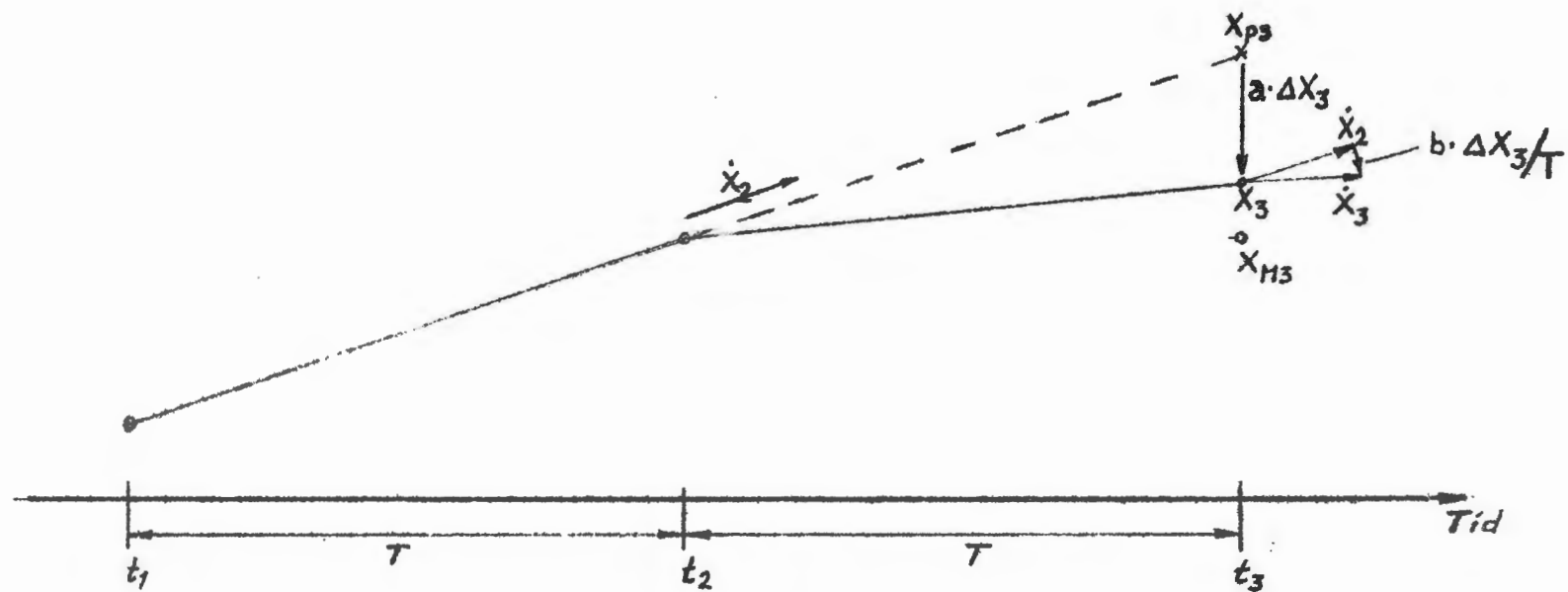


FIG. 19

PREDIKTERING OCH KORREKTION AV FÖLJNING

a och b ges, desto kraftigare blir glättningen, dvs desto mindre hänsyn tas till avvikelser mellan sanna och predikterade mätvärden. Stora värden på a och b ger däremot snabba korrektioner, dvs avvikelser följs med liten tröghet, vilket tyvärr då också gäller avvikelser p.g.a. störningar. Fig 20 visar två följningsfall för en bana med såväl brus som sväng; det ena (heldragen kurva) gäller för  $a=b=0,5$  och det andra (punkterad kurva) för  $a=b=1$ . Under de raka delarna av banan ger det glättade fallet ( $a=b=0,5$ ) en bättre följning än det oglättade, dvs i det förra fallet kan mätluckorna vara mycket små, och alltså är risken för störning liten. I svängen däremot släpar det glättade systemet efter avsevärt mer än det oglättade, dvs i det fallet kräver det förra systemet större luckor för att ha möjlighet att mäta in målekon och blir alltså störningskänsligare. En kompromiss mellan kraftig glättning med liten störkänslighet och liten glättning med hög manöver- och störkänslighet måste alltså väljas för att systemet skall ha optimal följningsförmåga. Tänkbart är också att växla glättning i olika fall; exempelvis kan korrektionen göras kraftigare, om maskinen upprepade gånger märker att mätvärdena avviker kraftigt från den predikterade banan.

Inverkan av glättning i predikterade system av ovan diskuterade typ har varit föremål för en hel del teoretisk behandling. Man har därvid kommit fram till vissa lagar för stabilitet och störkänslighet, som kan illustreras i diagram av den typ, som visas i fig 21 och 22. Den förra visar stabilitets-triangeln för a och b, inom vilken systemet är stabilt. Den streckade kurvan visar orten för kritisk dämpning, inom vilken systemet är överkritiskt dämpat. Fig 22 visar en del av samma diagram nämligen området  $a = b \leq 1$ , vilket normalt är det intressanta. Där har också kurvor över störkänsligheten inritats. Definitionen av störkänsligheten är därvid kvadratroten ur förhållandet mellan medelkvadraten för störning i ut- resp. insignal. Ur det senare diagrammet kan man utläsa, att störkänsligheten alltid är mycket liten om lägeskorrektionen a är liten. Däremot gäller inte detta alltid om b är liten. Vill man alltså ha mycket lite brus i utsignalen, bör man göra a så litet som möjligt (jfr fig 16 heldragna kurvan). De optimala värdena på a och b är svåra att beräkna och bestäms kanske bäst med hjälp av ett stort antal praktiska försök, som kan ge ett tillräckligt underlag för statistisk bestämning av optimalvärdena. De teoretiska värdena förrycks normalt också något i ett praktiskt system, där samplingsintervallen inte alltid är konstanta, vilket man antar vid teoretiska studier.

Av ovanstående framgår, att ett följssystem är mycket kritiskt med avseende på godheten i följningen. Största vikt bör läggas vid en optimering av korrek-

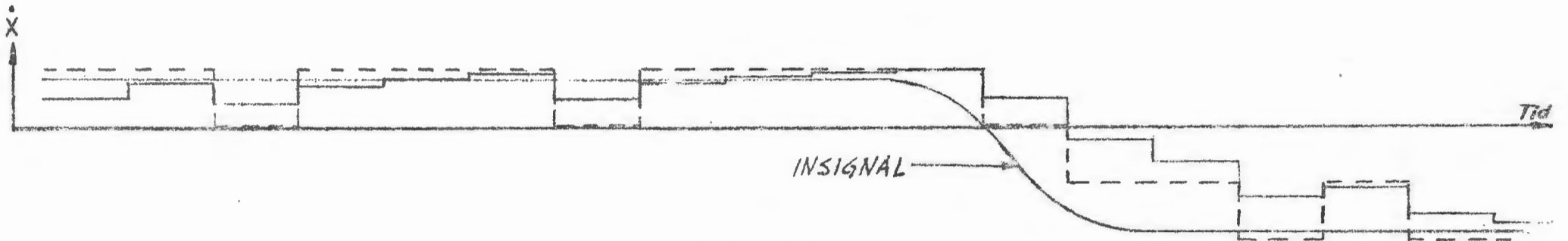
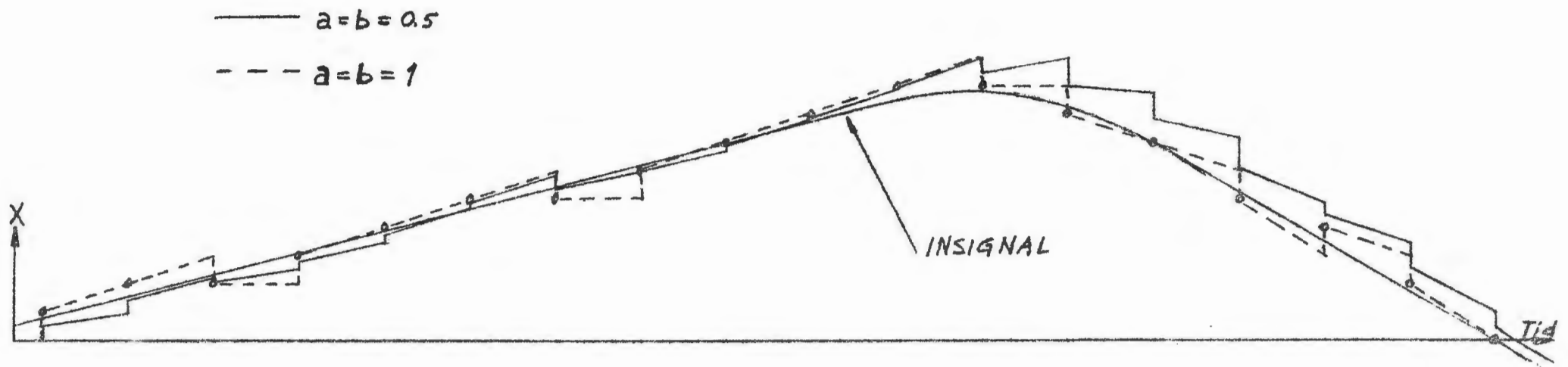


FIG. 20

FÖLJNING AV OLINJÄRT FÖRLOPP



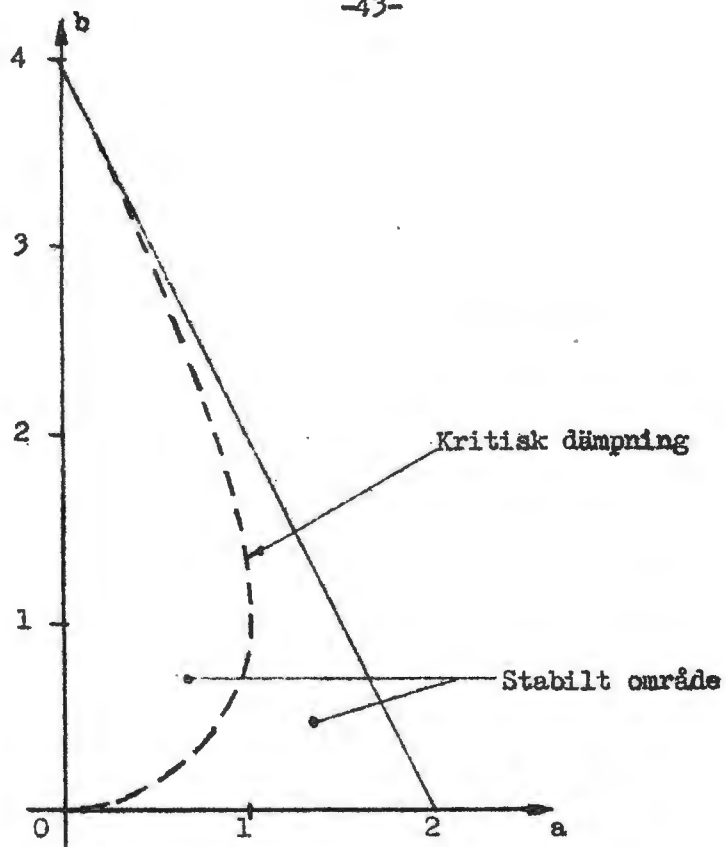


FIG. 21

Stabilitetsfigur för följsystem

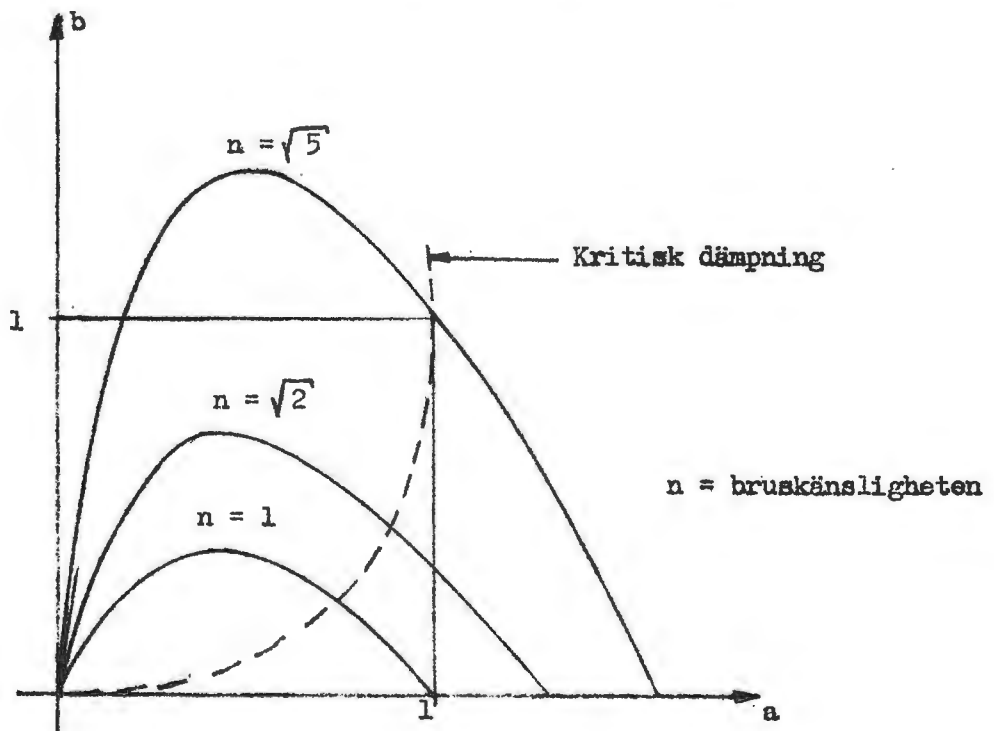


FIG. 22

Bruskänslighetskurvor för följsystem

tionsparametrarna, om man vill uppnå en säker funktion i ett ekonomiskt system. För att vara riktigt säker på att erhålla säker följdning borde man egentligen låta datamaskinen arbeta på samma sätt som en operatör vid ett PPI. Om ett flygplan exempelvis svänger i en bana som går genom ett område med ett flertal falska ekon, kan operatören inte alltid omedelbart avgöra, hur planet svänger eftersom flera ekon kan dyka upp i närheten av den position han förväntar sig. Han måste då på prov tänka sig flera utvecklingar av flygbanan och vänta ett eller flera radarvarv, tills så många ekon dykt upp att han säkert kan bedöma hur banan svängt. Fig 23 visar ett tänkt exempel, hur två falska ekon ger upphov till två extra provbanor. Normalt kanske man behöver vänta längre än vad som är angivet i figuren, innan en provbana avbryts eftersom avsaknad av eko kan bero på att detta är bortstört från en riktig bana. Man inser, att om man skall fordra ovannämnda möjligheter hos en maskin så kan belastningen på densamma bli avsevärd även för en måttlig grad av falska ekon. Ett sådant system kräver alltså mycket hög kapacitet hos maskinen, för att risken för övermättnad skall kunna undvikas. Alternativet är att endast välja en provbana för varje gång och då välja den, som är mest sannolik. Om detta resulterar i tappad följdning, kan det hela hjälpas upp igen av den övervakande operatören.

### 2.2.2.

#### Simulation

Termen "simulation" innebär utbyte av ett givet system med ett ersättnings-system en "simulator", som reagerar på samma sätt som det ursprungliga systemet för påverkan av visst intressant slag. En simulator används i de fall, då utnyttjande av originalsystemet är omöjligt, opraktiskt eller oekonomiskt vid studium av systemet. Sådant studium kan vara önskvärt exempelvis för konstruktion och provning av utrustningar som skall arbeta tillsammans med systemet eller för upplärning och träning i användandet av ett system (trainers). En simulator är alltså ett typiskt reelltidsystem, och det är också ganska uppenbart önskvärt, att simulatören skall vara omställbar för olika beteendemönster för att göra simulationen så variationsrik som möjligt. P.g.a. att en simulation ofta är ganska komplicerad, har tidigare analogisystemen varit dominerande som simulatorer, men utvecklingen inom digitalområdet har gjort det mer och mer möjligt att utnyttja digitalmaskiner med deras stora flexibilitet. En annan fördel med de senare är att de tillåter godtycklig noggrannhet och därmed ett noggrannare felstudium, som dessutom lämpar sig för registrering och matematisk behandling. Flexibiliteten kan utnyttjas till att avbilda logiska och stokastiska beteendemönster hos en process och programmerings- och minnesmöjligheterna till lagring av iakttagna och registre-

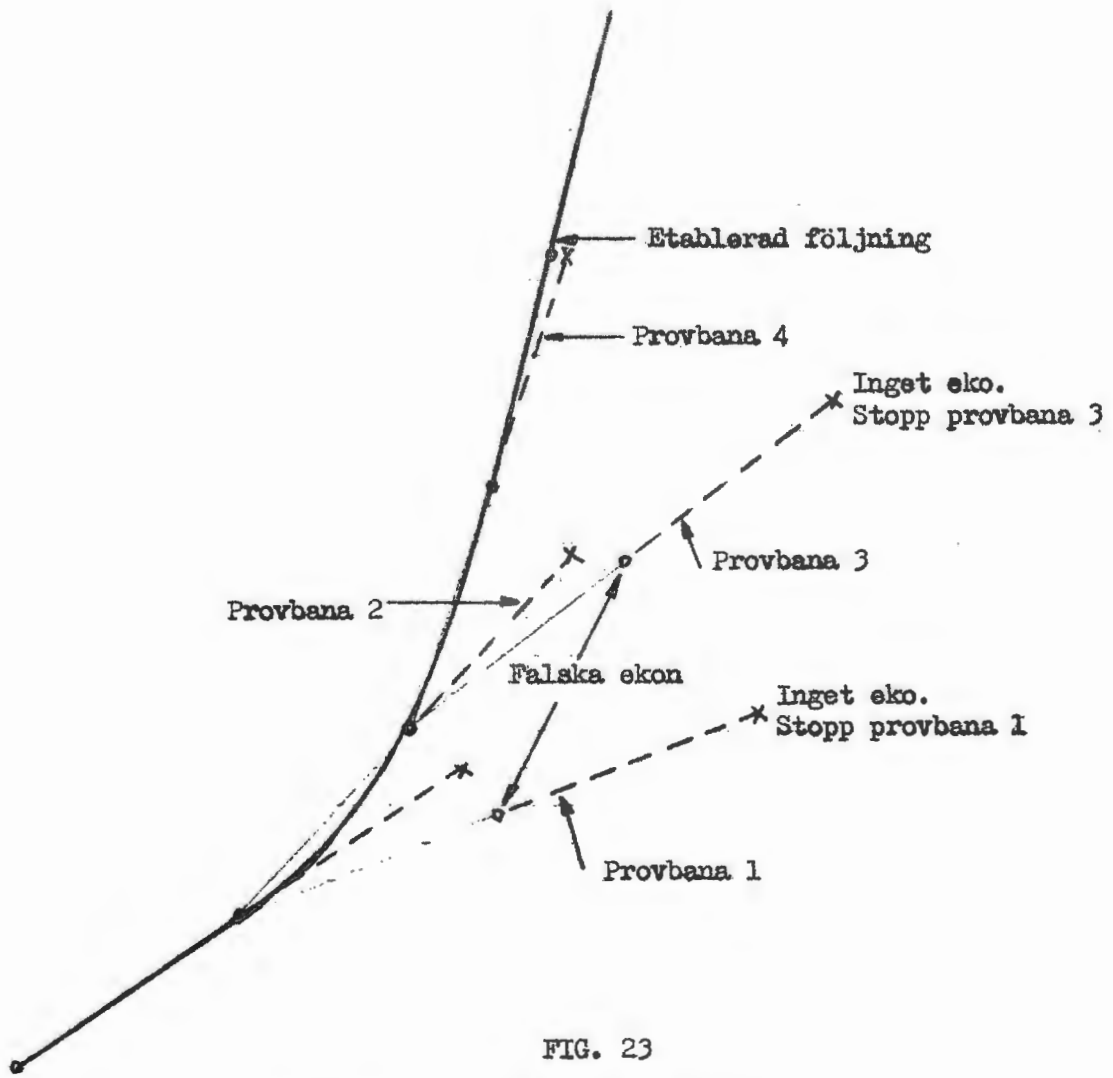


FIG. 23

Alternativbanor vid störd följdning.

rada beteendemönster, som om och om igen kan studeras exempelvis för optimering av en utrustning eller för stokastisk bedömning av en sådans uppträdande i systemet. I många fall har man funnit det fördelaktigt att använda hybrider av analogi- och digitalutrustningar, som ger fördelarna av båda typerna utrustningar.

Inom processregleringen har vi studerat datamaskinens uppgift som övervakare av en process. Vid simulation är det emellertid processen själv eller delar därav, som datamaskinen kan behöva representera. Som nämnts är det processens reaktioner för olika slag av påverkan, som skall simuleras, och eftersom de matematiska lagarna för sådana reaktioner kan vara högst komplexa, är det ofta så, att det ställs krav på den simulerande utrustningen som är av samma slag eller högre än vad som gäller för processreglerande utrustningar.

Beroende på vilka typer av processer eller vilka delar av en process som skall simuleras, måste oftast omvandling mellan analoga och digitala data och vice versa göras vid in- och utgångarna i simulatören. Sampling och tidsuppdelad behandling i den simulerande datamaskinen är därför vanligt förekommande, varför i stort sett samma samplings- och prediktionsproblem gäller för simulation som för processreglering. Återkopplingsproblemen däremot behöver inte nödvändigtvis finnas, eftersom en återkoppling normalt inte uppstår förrän en sluten kedja bildas tillsammans med ett reglerorgan, och det är då i detta som återkopplingsproblemen normalt uppstår.

Den interna databehandlingen i en simulator kan vara helt artskild från motsvarande i en processreglerande maskin. P.g.a. att det ofta är en fysikalisk process och dess förändring med tiden som skall simuleras, gäller beräkningarna ofta integrationer av differentialekvationer av varierande komplexitet. En mycket använd digitalutrustning i sådana sammanhang är därför den digitala differentialanalysatören, som behandlas i ett senare avsnitt.

### 2.2.3. Digitalteknik inom kommunikation

#### 2.2.3.1. Datakommunikation

Förmedling av information är ett område, där människan oförtröttligt sökt utvidga sina möjligheter. Då elektrotekniken togs i tjänst för detta ändamål för telegrafi, telefoni, radio etc., växte antalet kommunikationsförbindelser ut till en väldig omfattning, vilket gjorde att man snart började intressera sig för olika metoder att optimera förbindelserna. Främst skedde detta av ekonomiska hänsyn; förbindelser kostar pengar, och man vill därför kunna överföra så mycket information som möjligt på så få och billiga förbindelser som möjligt. Studier av informationsteorin tog fart och man började förstå



de lagar, som bestämmer förmedlingskapaciteten hos olika typer av förbindelser. Informationsteorin visade, att optimala överföringsegenskaper teoretiskt sett borde uppnås genom användande av diskreta informationer i stället för kontinuerliga som exempelvis i analogöverföringar av typ telefoni. I praktiken har man funnit, att denna optimering är lättast att åstadkomma med hjälp av binärsystem där alltså varje informationselement består av en bit och varje meddelande av en binär kod. Sådana koder kan sedan överföras med hjälp av någon typ av pulsöverföring. Den första elektriska kommunikationstypen, telegrafen, var just av detta slag och den är än i dag i överförd form, telex, en expanderande kommunikationsform.

En av de uppenbara fördelarna med pulsöverföring av diskreta data ligger däri, att data kan överföras med godtycklig noggrannhet om bara kapacitet därför finns i överföringskanalen. Detta är inte möjligt vid överföring av kontinuerliga signaler p.g.a. distorsion och störningar i modulatorer, transmissionsmedia och demodulatorer. För ett diskret, kodat meddelande är det däremot så, att meddelandet består av en viss sammansättning i tiden av pulser och pulsmellanrum, och mottagarsidan behöver bara känna dessa pulser och mellanrum för att exakt återbilda meddelandet. Relativt stor amplitud- och fasdistorsion kan tillåtas, innan ett så enkelt system upphör att fungera, och eftersom ett kodat meddelande kan göras godtyckligt noggrant genom att antalet pulslägen ökas (ökat antal bitar i ett binärt system), så inser man, att en noggrann överföring är möjlig med tämligen enkla medel. Motsvarande fenomen gäller även för digitala databehandlingsutrustningar, där dataflödena också är pulsförbindelser. När därför utvecklingen på det digitala databehandlingsområdet ledde till behov av kommunikation inte endast mellan olika delar i datautrustningarna utan också mellan olika datautrustningar belägna geografiskt långt från varandra, låg det nära till hands att välja den kommunikationsmetod, som mest liknade den som gäller internt i utrustningarna dvs pulsöverföring. Denna överföringstyp är också den enda, som kan utnyttja noggrannheten i datautrustningarnas informationer. Av olika typer pulsöverföringar har den pulskodmodulerade (PCM) blivit den viktigaste, emedan mönstret av pulser i detta fall kan göras exakt likt ett binärt meddelande vilket innebär att förmedlingen mellan datautrustning och transmissionsmedium kan bli rätt enkel. En annan fördel med sådan kodning är att ändutrustningarna kan göra logiska bedömningar på den överförda informationen, vilket bl.a. är viktigt vid adressering av meddelanden till ett urval av flera mottagare. Fig 24 visar, hur en dataöverföringslänk mellan två datautrustningar ser ut i princip. Ofta ligger bitarna i ett meddelande parallellt i den till

länken anslutna datamaskinen, eller om den arbetar i serieform har den ofta mycket högre frekvens än datalänken, varför ett lagringsorgan normalt måste finnas i sändarterminalen där en omvandling till serieutmatning av meddelandenas bitar kan göras i en takt som är anpassad till överföringslänkens bandbredd. Den erhållna pulsserien passerar en modulator före matning till länken. Den vanligaste typen av modulation är idag frekvensmodulation (vid pulsöverföring kallad frekvensskift) p.g.a. dess större möjligheter till störhållfasthet. På mottagarsidan finns en motsvarande utrustning för omvandling tillbaka till parallell digital form för den mottagande datamaskinen.

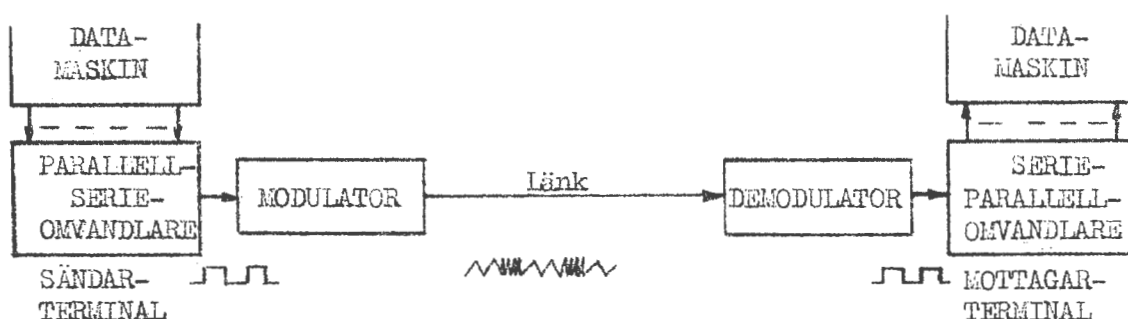


FIG.24

Det ovan skisserade allmänna fallet har på senare tid kommit till användning i stora processövervakningssystem exempelvis för luftbevakning, där flera databehandlingscentraler kan förmedla data till varandra, eller för automatisk väderdatabehandling, där ett flertal prognoscentraler kan behöva samarbeta. Dessa omfattande kommunikationsnät finns f.n. i USA men torde inom en snar framtid komma att växa fram mer allmänt.

Ett annat fall av datakommunikation är s.k. "telemetering", vilket innebär överförande av olika typer av mätdata till en central databehandlingsutrustning. I det fallet matas sändarterminalen ofta från någon typ av analog-digitalomvandlare för processvariabler men även från exempelvis digitala manöverorgan i form av omkopplare etc.

Trots PCM-överföringens goda motståndskraft mot störningar kan dessa inte helt undvikas utefter de ofta rätt långa transmissionsvägarna. En stor del arbete har därför ägnats åt möjligheterna att minska risken för felöverföringar. Detta är viktigt i synnerhet för sådan dataförmedling, där ett meddelande endast överförs en gång vilket gäller exempelvis vid överföring av beräkningsdata till och från en databehandlingscentral. Detta slag av överföring har blivit mer och mer utnyttjat för hyrning av beräkningstid i datacentraler från avlägsna beställare. För att öka överföringssäkerheten har man därför uppfunnit en mängd mer eller mindre invecklade metoder omfattande redundant kodning med felupptäckt och felkorrigering, omsändning



etc. Dessa faciliteter ställer rätt stora krav på sändar- och mottagarterminalernas databehandlande förmåga, varför dessa i sådana fall kan sägas vara små fristående datautrustningar som förmedlar och anpassar dataflödet mellan kommunicerande utrustningar och överföringsmediet.

Vid datakommunikation i reelltidssystem, vilka ovanstående exempel normalt inte tillhör, får man ibland lov att lägga lite andra synpunkter på överföringen. I sådana system får huvudvikten läggas vid överföringssystemets möjligheter att förmedla data till destinationsorten, innan de blivit inaktuella för utnyttjande i den gällande processövervakningen. Enligt vad som tidigare diskuterats är det ju högst viktigt att inte alltför stora fördröjningar mellan mätningar och styrande åtgärder uppstår i reglerkedjan. Man har därför i regel rätt stora krav på överföringens kapacitet uttryckt i bånd eller antalet överförda bitar per sekund. Detta gör också, att man söker undvika att förlänga meddelandena genom redundant kodning för felkorrigering. Den uteblivna felkorrigeringen kan oftast avhjälpas, tack vare att exempelvis en övervakad variabel i en process mäts upprepade gånger varför en felöverföring snart kan rättas till. Dock kan det vara olämpligt att låta ett felaktigt värde tas om hand av den mottagande utrustningen, varför man normalt söker upptäcka fel och blockera meddelandena där fel upptäcks. Felupptäckten fordrar viss extra information i överföringen, som oftast består av paritetskontroll och regelbundet överförda testmeddelanden av känd sammansättning.

I reelltidssystem kräver man som nämnts ofta en relativt hög kapacitet hos överföringen för att överförda data inte skall förlora sin aktualitet. Däremot kan det ofta inträffa att överföringsbehovet inte uppträder så tätt, att länken blir jämnt belastad. Detta innebär givetvis dålig ekonomi och man försöker i sådana fall att lägga förmedling av så många olika data som möjligt i en och samma överföring. Uppdelning på de olika dataslagen görs då på tidsbasis i det att varje typ av meddelande får sin del av tiden i överföringen (tidmultiplex). Denna form av multiplex lämpar sig utmärkt för pulskodad överföring, eftersom datafördelningen lätt kan göras med hjälp av digitala logiska kretsar. Man vinner normalt också bandbredd med denna metod gentemot andra multiplexförfaranden exempelvis frekvensmultiplex, där frekvenskanalerna måste läggas rätt långt från varandra för att överhörning skall undvikas. Sådana problem existerar inte i tidmultiplexsystem, utan där kan informationerna tätpackas i det tillgängliga frekvensutrymmet.

Ovan har nämnts, att transmissionskapaciteten är en viktig faktor vid bedöm-

ning av en överförings användbarhet i ett reelltidssystem. Kapaciteten är bestämd av överföringsfrekvensen, som i sin tur är beroende av det frekvensområde som förbindelsen kan överföra. Transmissionshastigheten uttrycks normalt i enheten "baud", och antalet baud innebär antalet överförda informationselement eller bitar per sekund. Vid denna definition antar man, att pulsmoduleringen utföres utan pulsmellanrum mellan successiva ettor i koden (non return to zero) varför antalet baud även kan definieras som maximala antalet växlingar per sekund mellan puls och icke puls.

Normala telegrafförbindelser kan överföra ett frekvensområde av drygt 100 Hz och tillåter i praktiken en transmissionshastighet av 50 baud, medan en telefonlinje kan överföra 2400-3000 Hz med en transmissionshastighet av 1200-1800 baud. Speciella telefonlinjer använda för radioöverföring, s.k. programledningar, tillåter upp till 50 kHz dvs c:a 25 kilobaud. Dessa förbindelser är de i dag vanligaste i stigande kostnadsgrad, men även bredbandsförbindelser över radiolänk utnyttjas i synnerhet som inlagrade kanaler i mikrovågslänkar, varvid man enligt amerikanska uppgifter överfört ända upp till megabaud.

Normalt använder man i alla fall som sagt telegraf- och telefonlinjer. Man skiljer därvid mellan privata och offentliga förbindelser. En privat förbindelse ligger normalt i samma nät som det offentliga men används inte i normal drift utan är kopplat förbi telex- och telefonstationer. Sådana förbindelser uthyrs ofta för speciella ändamål. Den privata förbindelsen är normalt alltid ansluten till de kommunicerande organen och blir därför dyrbar i hyra men ger i gengäld en alltid tillgänglig förbindelse. Den är dessutom inte utsatt för de svåra pulsstörningar, som ofta uppträder i offentliga nät p.g.a. reläslutningar i automatväxlar. I motsats därtill har man på det offentliga nätet större möjligheter att koppla upp olika förbindelser plus det förhållandet, att man endast betalar för den tid man verkligen sänder på förbindelsen. I processregleringar är emellertid den ständiga beredskapen hos en förbindelse mycket viktig, varför privata förbindelser är de vanligaste. Dessa är som nämnts inte heller så utsatta för störningsrisker som de offentliga förbindelserna, varför inte så stor vikt behöver läggas vid att lägga in stor kapacitetskrävande redundans för felupptäckt.

Kostnaderna för upprätthållande av förbindelser blir en allt större faktor i de kolossala datakommunikationssystem, som så småningom börjar växa upp runtom i världen. Behovet av sådana system är f.n. kanske störst på militärsidan, dels för informationsförmedling från olika typer av utspridda informa-



tionskällor till centrala behandlingsplatser men även för övervakning av sådana källors driftstillstånd och funktioner. Inom strategin är dessutom naturligtvis sambandstjänsten mellan olika stridsenheter en avgörande faktor. På den civila sidan kan man emellertid också skönja ett ökat behov av vidsträckt förgrenad datakommunikation exempelvis inom affärslivet, industrin och ett sådant viktigt område som väderövervakning.

Det är naturligt att vidsträckta och komplicerat hopknutna kommunikationsnät kräver många förbindelser. Ur ekonomisk synpunkt kan det betyda oerhört mycket, om dessa förbindelser är maximalt belastade eller inte. Vad man strävar efter, är att fördela hela dataflödet så, att mindre belastade vägar får ta över en del av arbetet från överbelastade vägar varigenom en sammanlagring erhålles som tillåter minimering av förbindelsenätet. I ett stort system är det givet, att sådan optimering måste utföras automatisk p.g.a. de snabba växlingar i belastningsfördelningen som kan inträffa. För sådan automatisering har man i USA (bl.a. ITT och IBM) med stor framgång börjat använda stora datamaskiner, som placeras i viktiga knutpunkter i kommunikationssystemet. Dessa maskiner kan dels övervaka belastningen på olika datavägar och fördela dataflödet; dels kan de också sammanställa och omforma informationer från vissa förbindelser till informationsgrupperingar för olika datakonsumenter på andra förbindelser etc. Sådana maskiner är konstruerade med huvudvikt på samtidiga inkopplingsmöjligheter för en mängd yttre organ, samtidigt som flexibiliteten i programmeringen bör vara stor för varierande arbetsuppgifter.

#### 2.2.3.2. Automatiska telefonväxlar

En teknikgren som har många drag gemensamt med digital databehandling är telefonväxeltekniken. I en telefonväxel skall information kopplas till och från olika transmissionskanaler enligt givna förutsättningar och denna koppling utföres med hjälp av logiska element på i många fall samma sätt som i en datamaskin. Skillnaden är i huvudsak den, att en telefonväxel innehåller så mycket mer möjliga informationsvägar än en datamaskin, medan den senare har större antal olika sätt att arbeta och har större snabbhetskrav.

Hittills har reläer varit dominerande både som logiska element och talbanomkopplare i telefonväxlar. En ofta förekommande logisk enhet är den s.k. väljaren, som kan avsöka förbindelser i tur och ordning. Väljarna kan antingen vara utvecklingar av reläer eller också elektromekaniska roterande avsökare. De äldre automatväxlarna innehåller seriekopplade väljare, som stegas fram till rätt mottagarledning styrda av sifferimpulser från nummerskivan hos den anropande abonnenten. Detta system är opraktiskt för större abonnentgrupper,

eftersom det kräver rätt många väljare vilka får begränsad kapacitet p.g.a. det omfattande sökandet efter lediga förbindelser. Väljarna i sådana växlar innehåller också alla logiska element som behövs för framstegning och utval av lediga förbindelser. Metoderna utvecklades så småningom, så att de på nummerskivan inslagna siffrorna lagrades i register i växelstationerna. Därigenom tilläts snabbare sifferöverföring från abonnent till växel och snabbare utställning av väljarna, då dessa kunde arbeta med optimal snabbhet i stället för efter fingerskivan. Nästa vidareutveckling var införande av markerarsystem. En markerare innehåller alla de logiska element, som behövs för utställning av väljarna till ledningar, och vad som gör markerarsystemet ekonomiskt är att markerarna kan fördela sitt arbete i tiden på en mängd eller t.o.m. alla väljare i en växelstation. Väljarna kan därigenom göras enkla, snabba och driftsäkra. Markerarna är alltså ett slags allmänna övervakare av förbindelserna i en växel. Det är uppenbart, att bl.a. snabbhetskraven på markerarna är stora i en större växel p.g.a. den mängd tidsuppdelade kopplingar och bedömningar som behöver göras. Det ligger därför nära till hands att låta markerararbetet utföras av en central snabb datamaskin. De speciella drag en sådan maskin skulle få är förbindelseorganen med abonnenterna och styrorganen för kontakterna i abonnentförbindelserna.

Fig 25 visar ett enkelt schema över hur ett datamaskinstyrt telefonnät kan se ut. Systemet består av en transmissionsdel, där den fysikaliska sammankopplingen mellan abonnenterna sker, och en databehandlande del, som tar emot

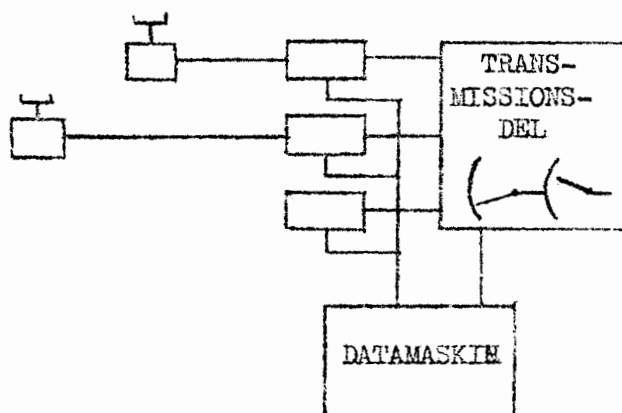


FIG. 25.

information från abonnentledningar (via register) och uppkopplade förbindelser samt verkställer upp- och nedkopplingar i transmissionsdelen.

Kravet på datamaskinen i ovan skisserade system är, att den måste vara mycket snabb om den skall vara gemensam för hela växelstationen. Dessutom måste dess

förslitning vara liten även för hög trafik.

Man kan tänka sig ett par olika typer av växlar i ett datamaskinstyrt system. Det ena kan kallas rumsuppdelat och det andra tidsuppdelat system.

#### Rumsuppdelning

Med ett rumsuppdelat system menas ett väljarnät uppbyggt av elektromekaniska eller elektroniska kontakter, där en kontakt endast kan förmedla ett samtal åt gången. Till denna grupp räknas givetvis de idag normalt använda elektromekaniska kontakterna.

Kraven på kontakterna i ett sådant system är, att de skall vara snabba för att ge korta beläggningstider för datamaskinen. De skall dessutom dra lite effekt för att bättre anpassas till normala datamaskinutgångar. Sedan har man givetvis alltid önskemålet, att de skall vara små så att stora växlar kan bli praktiskt utförbara. Mycket arbete har lagts ned på att konstruera goda elektroniska kontakter, men hittills har man inte lyckats få fram några, som både i funktion och ekonomi kan tävla med elektromekaniska sådana.

#### Tidsuppdelning

I ett tidsuppdelat system utnyttjas samma talkontakt för flera talförbindelser, genom att de olika förbindelserna tilldelas var sitt tidselement. Fig 26 illustrerar principen. Två pulsfrekvenser med skilda faser moduleras av

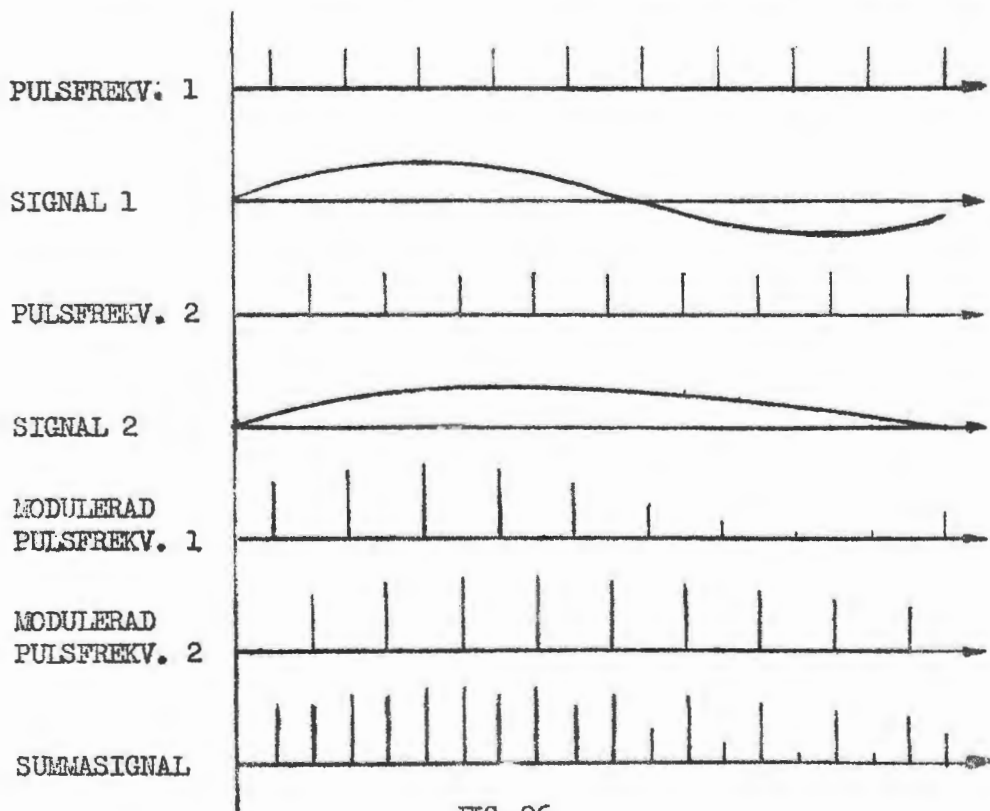


FIG. 26



var sin talsignal, varefter de blandas på överföringsledningen. Inmatnings-terminalen är därvid försedd med kontakter, som sluts i bestämda ögonblick för varje inkommande abonnentlinje. Motsvarande finns även vid utmatnings-terminalen från den gemensamma ledningen "filen". Före utmatningen måste givetvis de utvalda pulstågen återbildas till de ursprungliga talsignalerna i filter. Man spar här i hög grad kontaktpunkter och ledningar i den centrala uppkopplingsdelen, eftersom gemensamma s.k. filer kan användas för en mängd samtal samtidigt. Väljarkontakterna måste i ett tidmultiplexsystem vara så snabba, att elektromekaniska sådana inte kan komma ifråga. P.g.a. pulsningen är det emellertid möjligt att åstadkomma kontakter med goda slutningsegenskaper med hjälp av halvledare. De kan också tillåtas bli något dyrare eftersom de är ganska få i en växel. I dag praktiska pulsfrekvenser på en fil varierar mellan ett par hundra kHz och c:a en MHz med filen uppdelad på 20 till 100 talförbindelser. Kontakterna måste slutas i sina bestämda ögonblick med hjälp av pulser från någon typ av minnen, som ofta är cirkulerande exempelvis trum- eller nickeltrådsminnen. Dessa styr samtidigt samtliga kontakter ingående i en talbana. Vid uppsättning och nedkoppling av samtal måste de i dessa minnen lagrade informationerna kunna ändras från den centrala datamaskinen.

#### Databehandling

Principen för databehandlingsutrustningen framgår av fig 27. Utrustningen känner tillståndet på anslutna ledningar med hjälp av avsökare. Datautrustningen kan anslutas till väljarnätet för att avläsa tillståndet, sätta upp nya förbindelser eller koppla ned gamla. För att kunna sända tonsignaler till

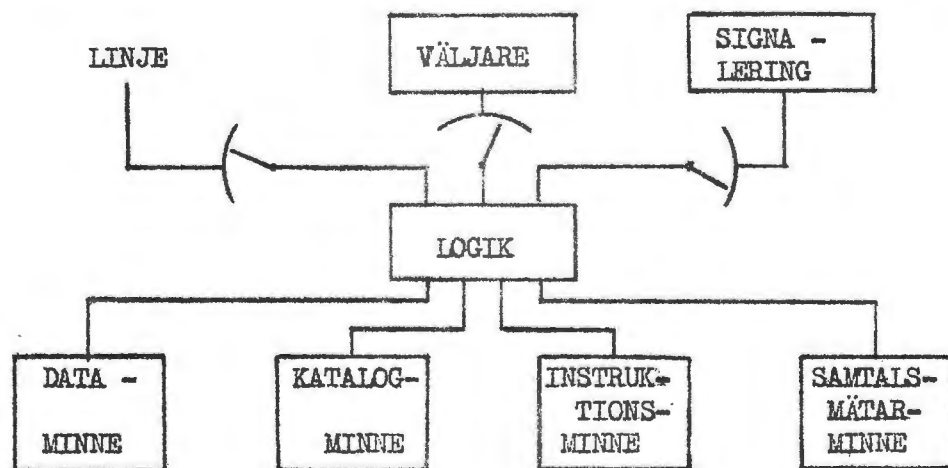


FIG. 27

abonnenter och andra stationer måste datautrustningen kunna manövrera ett signalsändande don.



Datautrustningen består i huvudsak av följande delar:

Dataminnet: där lagras tillfälligt information främst från anropande abonnent samt över uppkopplade tillstånd i väljarnätet.

Katalogminnet: där lagras något varaktigare information om exempelvis abonnenternas katalognummer och klassmarkeringar, lednings- och riktnummeröversättningar, uppgifter om alternativa trafikvägar etc.

Samtalsnätminne: där lagras som namnet säger uppgifter, varpå debitering skall grundas.

Instruktionsminne: i detta minne lagras som i vanliga datamaskiner instruktioner för maskinens arbetssätt, exempelvis vid vilka tidpunkter de olika minnena och utorganen skall avläsas, förses med ny information eller fräntas gammal.

Datautrustningen avsöker successivt alla ledningar och jämför tillståndet på dessa ledningar med tillståndet i dataminnet. Om någon förändring skett, undersöks tillståndet i väljarnätet. Om förändringen är ett anrop, avläses den sökta abonnentens nummer och lagras tillfälligt i logikdelen, varifrån katalogdelen tillfrågas om abonnentens s.k. multipelnummer. Med hjälp av detta nummer kan maskinen i dataminnet avläsa, om abonnenten är ledig eller upptagen.

Uppgifterna i katalogminnet om alternativvägar är särskilt värdefulla vid högt trafik på vissa förbindelser, varvid andra mindre belastade kan uppsökas och därigenom väntetiderna för abonnenterna nedbringas utan att kapaciteten på de ansträngda ledningarna ökas. Detta medför mer ekonomiskt utnyttjande av de tillgängliga förbindelserna.

Datamaskinstyrda telefonnät befinner sig ännu rätt mycket på försöksstadiet. Metoden är emellertid lovande och torde vara den enda ekonomiska utvägen att klara av det ständigt växande telefonnätet världen över.

### 3. DIGITALUTRUSTNINGAR VID SPECIALTILLÄMPNINGAR

#### 3.1 Systemkrav

Som nämndes i inledningskapitlet, gäller specialtillämpningar inom digitaltekniken oftast reelltidsystem, där den digitala utrustningen tjänar som kontrollutrustning med stark bundenhet till systemets eller processens förlopp. Digitalutrustningens arbetssätt kan därför normalt inte väljas godtyckligt utan blir starkt påverkat av de krav, som systemet uppställer. Man kan därför säga, att i ett sådant fall databehandlingsutrustningarna oftast är processororienterade medan däremot i många fall av generell databehandling

exempelvis inom affärsbranschen databehandlingssystemen är datamaskinorienterade.

Den ovannämnda systeminriktningen hos en kontrollutrustning i ett reelltids-system betingas främst av att den måste ombesörja datakommunikationen med processen vid de tillfällena, då data dyker upp, oftast utan att kunna lagras dessa i mellanninnen av karaktären hålkort eller magnetband för bekvämare åtkomst för datamaskinen. Det senare förekommer dock vid s.k. off-line system, som emellertid endast är användbara för mycket långsamma processer.

En annan faktor som bidrar till systeminriktningen är anslutningarna mellan kontrollutrustning och process. I vissa fall isynnerhet vid måttligt snabba processer är det enbart anslutningsutrustningarna dvs oftast omvandlare, som ger digitalsystemet dess specialkaraktär. Särskilt gäller detta om processen är så pass långsam, att omvandlarna kan förses med databuffertar för synkronisering av dataförmedling med datamaskinens kommunikationstillfällen. Vid många anslutningar mellan kontrollenhet och process blir emellertid sådana buffertar dyrbara, så att även i måttligt snabba system direkt kommunikation med kontrollenheten kan vara fördelaktig.

Beroende på art av system och grad av automatisering kan datautrustningen ha funktioner av olika kvalifikationsgrad. I huvudsak kan tre typer av sådana funktioner särskiljas:

- 1) Insamling och registrering av data
- 2) Insamling, registrering och processbedömning
- 3) Komplet systemövervakning.

Funktionen 1) innebär, att datautrustningen kontinuerligt avsöker alla omvandlare för mätdata och andra informationsgivare för en process samt registrerar dessa data i minnen. En sådan utrustning kallas ofta datalogg. I vissa fall kan omvandlarna för mätdata utgöra en del av dataloggen, i synnerhet om typen av omvandling kan standardiseras exempelvis till enbart omvandling av spänningsvärden. I andra fall är detta inte möjligt p.g.a. omvandlarnas speciella karaktär eller deras belägenhet exempelvis vid telemetering.

En datalogg fungerar ofta som ett mellansteg i en större process mellan den samma och den centrala kontrollenheten. Dataloggens uppgift därvid är att systematisera och eventuellt reducera datanängderna för att därigenom avlasta den centrala enheten. Dataloggens arbete blir därigenom inriktad på administrering och lagring av data, medan beräkningsarbetet är minimalt.

Givetvis är en datalogg också användbar i off-line system för insamling och registrering av data för eventuell senare databehandling, som kan gälla vid

testprocesser (vindtunnelprov, flygprov etc.).

Funktionen 2) är en utökning av 1) till att omfatta även beräkningar, som skall utföras på uppsamlade processdata. Beroende på mängden processvariabler och komplexiteten i deras beteende blir kraven på maskinens beräkningskapacitet högst varierande. Kontrollenheter av denna typ står ofta i förbindelse med något slag av presentationsorgan för förmedling av processbedömningar till en operatör, som i sin tur skall verkställa eventuella ingrepp i processen efter kontrollenhetens bedömningar och alltså sluta regleringskedjan. Beroende på systemsnabbheten kan presentationen ske via skrivmaskiner (ev. off-line) eller kontinuerliga presentationsutrustningar.

Funktionen 3) är den mest avancerade och kan ge en helt automatiserad reglerkrets i det att denna kan slutas av kontrollenheten utan mänsklig inblandning. Ett mellanstadium mellan 2) och 3) är därvid det fall, då kontrollenheten beräknar kompletta inställningsdata för processen som sedan en operatör manuellt kan åtgärda. I det fallet kan kontrollutrustningen även arbeta off-line, om snabbhetskravet tillåter detta. I det helt automatiserade fallet däremot måste givetvis datamaskinen stå i ständig kontakt med såväl in- som utomvandlerare. Det automatiska fallet ställer givetvis de allra största krav på den kontrollerande datamaskinen. Den måste vara programmerad för alla möjliga uppträdanden hos processen inkluderande start och stopp av densamma. Dessutom kräver man oftast, att maskinen själv snabbt kan upptäcka och signalera eventuella fel samt lokalisera dem, så att processen inte äventyras genom felaktiga åtgärder från maskinen.

Ovanstående fall är olika avancerade typer av vad man kallar integrerad automation, dvs automation som tar hänsyn till samfällid inverkan av så många parametrar som möjligt för optimering av en process.

Fortsättningen av detta kapitel kommer att behandla konstruktiva problem i digitalutrustningar för specialtillämpningar. Såväl den generella datamaskinens möjligheter som vissa intressanta specialutrustningar skall beröras, de förra med speciell hänsyn till de möjligheter som vissa nya principer för datamaskiner har givit upphov till.

## 3.2. Generella maskiner i specialtillämpningar

### 3.2.1. Allmänt

I inledningskapitlet diskuterades i korthet några faktorer, som påverkar valet om generella datamaskiner eller specialutrustningar skall användas vid specialtillämpningar. Det framhölls därvid, att den generella maskinen normalt är



flexiblare än en specialutrustning men att samtidigt ekonomi och användbarhet är faktorer som har avgörande påverkan av valet. Ekonomiska hänsyn kan emellertid vara dubbelriktade till sin verkan. I ett litet system med ett fåtal enkla processvariabler är det ofta billigare med en liten specialutrustning än med en större datamaskin, som inte kan fullt utnyttjas. Finns det däremot möjlighet att ge datamaskinen andra uppgifter för att utnyttja en eventuell överkapacitet, kan givetvis förhållandet bli det motsatta, om tillräckligt många små utrustningar kan ersättas av en enda större. Detta gäller också för större system, där många variabler skall behandlas samtidigt. I synnerhet är därvid en större datamaskin önskvärd, om variablerna inte är helt oberoende av varandra eftersom i ett sådant fall en integrerad behandling av variablerna kan vara nödvändig. Det är alltså uppenbart, att ju fler särskilda databehandlingar en maskin kan utföra samtidigt, desto mer ekonomisk blir den i förhållande till specialutrustningar. Den andra faktorn vid valet, användbarheten, är givetvis i hög grad utslagsgivande. En maskins användbarhet bestäms enligt ovan bl.a. av dess möjlighet att samtidigt utföra olika arbetsuppgifter samt naturligtvis också av dess kapacitet bestämd av minnesutrymme, snabbhet och noggrannhet. Krav på ökade kapaciteter har tillfredsställts dels p.g.a. i huvudsak komponenttekniska framsteg, som inte nödvändiggör några större avsteg från äldre normer för maskinkonstruktioner, men också p.g.a. en del nya organisatoriska tankegångar, som skall beröras nedan. Önskemålen om simultanbearbetning däremot har utvecklat ett speciellt drag hos nya maskintyper, den s.k. avbrottstekniken, som kanske mer än något annat ger datamaskinen förmåga att anpassa sitt arbetssätt till beteendet i tiden hos en process. Denna teknik tillåter maskinen att själv planera sitt arbete efter från början uppställda optimeringsregler bestående av prioritetssättningar av olika behandlingsuppgifter.

### 3.2.2. Prioritetsavbrott

Avbrottstekniken innebär, att datamaskinen på order av speciella signaler från anslutna yttre enheter, brytsignaler, skall kunna göra programhopp, dvs avbryta pågående uppgifter till förmån för andra viktigare sådana samt återuppta det avbrutna arbetet, då det nya är klart. Arbetssättet påminner om vanliga programhopp till subrutiner (wheelerhopp) men kompliceras av det förhållandet, att avbrotthopp inte är programmerade som vid wheelerhopp. De förra är i motsats till de senare asynkrona med maskinprogrammet, vilket försvårar utförandet.

Som vid de flesta logiska operationer i datamaskiner kan man gå två vägar, om man vill förse en generell maskin med avbrottsmöjligheter: den nödvändiga



logiken kan inläggas som en speciell subrutin i minnet eller den kan byggas in kretsmässigt i maskinens kontrollorgan. Denna inbyggda logik kan vi här kalla "avbrottsorgan". Terminologin tycks ännu inte vara fastlagd på svenska språket, varför olika benämningar kan påträffas (exempelvis programvalsenshet).

För att det skall kunna förhindras att avbrott görs för mer än en brytsignal i taget vid eventuell samtidig signalering från yttre organ, ordnas brytsignalerna i prioritetsordning allt efter vikten eller brådskan för de olika yttre organen. Denna prioritetsordning måste lagras i maskinen från början. Pågår ett program med viss prioritet, tillåts inga avbrott initierade av brytsignaler med lägre prioritet men däremot av sådana med högre prioritet.

Ett avbrottsorgan kan vara uppbyggt av i princip två register; det ena håller reda på det löpande programets prioritet genom en lagrad kod och det andra lagrar prioritetskoderna för mottagna anrop. Låt oss kalla det förra registret för "prioritetsregister" och det senare för "avbrottsregister".

Fig 28 visar i ett flödesdiagram, hur ett avbrott via avbrottsorgan kan verkställas. Datamaskinen jämför prioritetsskoderna i prioritets- och avbrottsregister efter varje operation i det löpande programmet. Om någon avbrottsprioritet är högre än prioriteten för det löpande programmet, avbryter maskinen omedelbart det senare, eljest fortsätter maskinen till nästa instruktion i det löpande programmet. Vid avbrott måste alla uppgifter för det avbrutna programmet, dvs delresultat, pågående instruktion och prioritet, läggas undan i minnet för att sedan kunna hämtas fram, då återgång till samma program åter kan ske. Detta tillåter, att maskinen fortsätter beräkningarna från avbrottet i stället för att behöva ta om dem från början.

Om vissa speciella instruktioner införes, kan ovan angivna avbrottsmetod även realiseras helt genom programmering. I detta fall erfordras dock flera programsteg, innan maskinen kan avgöra om det löpande programmet skall brytas. Fig 29 visar i ett flödesdiagram, hur ett dylikt avbrott kan verkställas. Brytsignaler får i detta fall lov att tas in som data från yttre enhet, varför räkneorganet i maskinen måste rensas innan någon prioritetsjämförelse kan ske. Prioritetsregistret motsvaras i detta fall av speciell minnescell. Skall prioritetsjämförelse göras efter varje operation, är det tydligt, att denna jämförelse kommer att uppta det mesta av maskinens tid. Vid glesare jämförelser blir tiden för åtgärdande av avbrott längre, varför den programmerade metoden är helt underlägsen. Detta gäller i ännu högre grad, om avbrottsorganet kan förses med fasta uppkopplingar av de instruktioner som i det fallet behöver utföras för dataundanläggning etc, varigenom avbrottsförberedelse och återhopp går fortare än vid minnesprogrammering för motsvarande. Eftersom

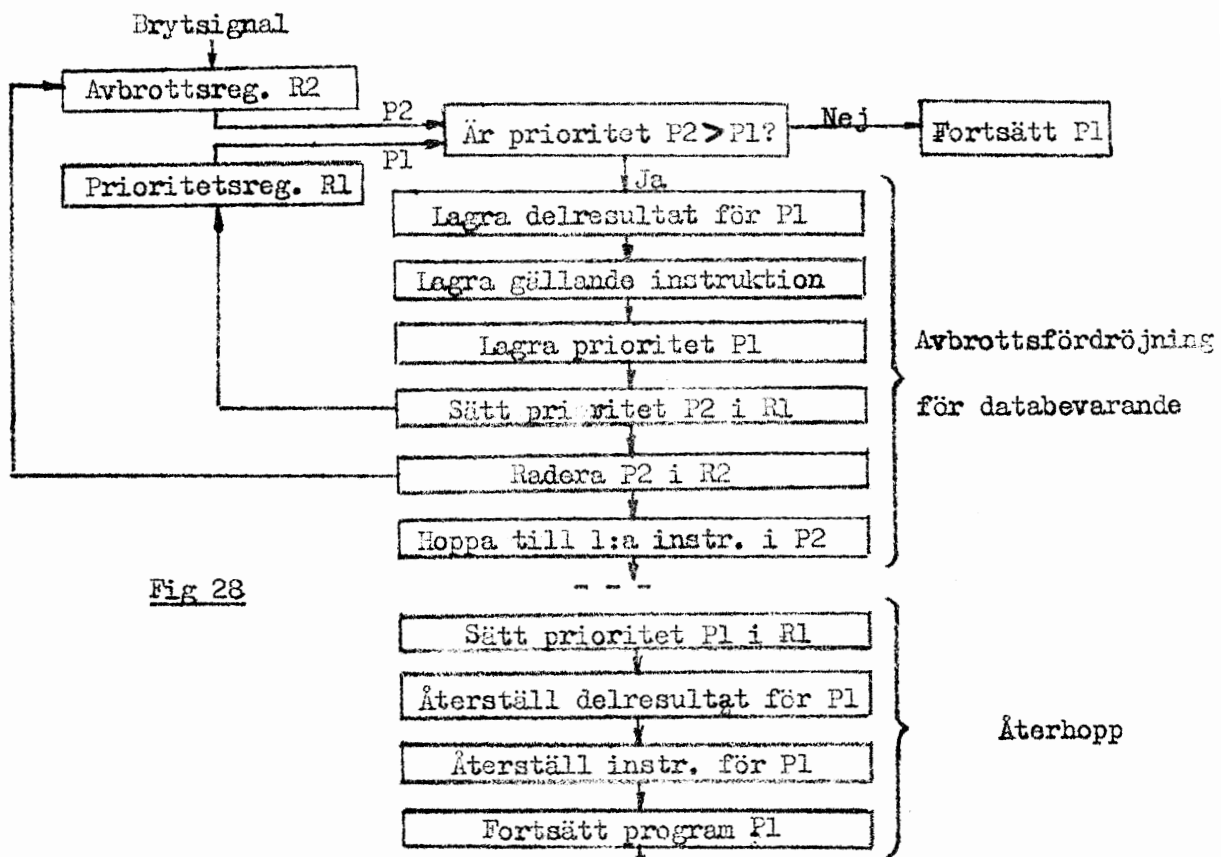


Fig 28

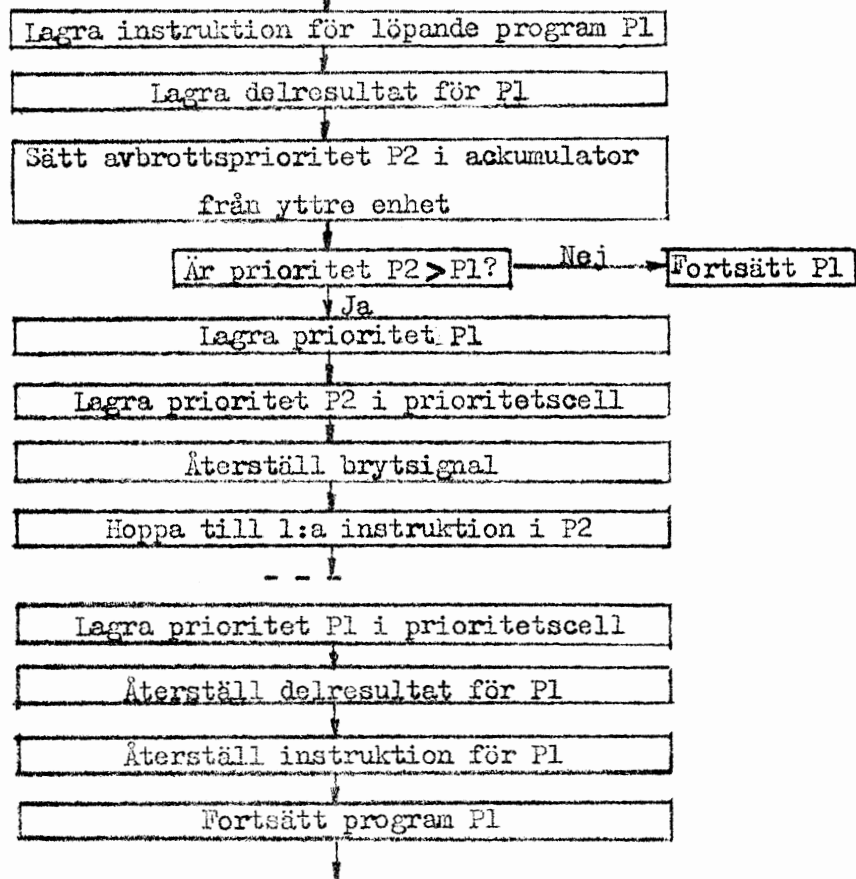


Fig 29

dessa instruktioner normalt är oföränderliga, är det också fullt möjligt att använda fast uppkopplade rutiner för dem i avbrottsorganet.

Ytterligare komplikationer för programmerade avbrott kan man se, om flera brytsignaler kommer samtidigt så att dessa först måste jämföras sinsemellan. Det ovan anförda talar därför starkt för införande av särskilt avbrottsorgan i datamaskiner för simultanbehandling av flera förlopp. I fortsättningen antas därför, att sådan utrustning finns i maskin med avbrottsmöjligheter.

En komplikation, som man alltid måste räkna med i en maskin med avbrottsmöjligheter är upprepade avbrott utan någon som helst ordning mellan resp. brytsignaler. Denna ordning beror på att brytsignalerna alstras av processförhållanden, som inte kontrolleras av datamaskinens program. Ett brytande program kan alltså själv brytas, så att maskinen kan ha ett flertal program under arbete som endast är delvis färdigbehandlade. Eftersom ett avbrutet program alltid uppsöks på nytt (se fig 28) då tid därför finns, kommer emellertid alla program att så småningom slutföras. Ett villkor för detta är emellertid, att maskinen som i fig 28 kan börja från den punkt i programmet, där avbrottet skedde. På detta sätt kan alltså maskinen själv fördela sitt arbete, så att den vid högbelastning endast utför de viktigaste arbetsuppgifterna och spar de mindre viktiga för tillfällen med lägre belastning.

Diagrammet i fig 30 visar ett exempel på ett händelseförlopp vid flera hopblandade avbrott. Processen ger en brytsignal med prioriten 3, medan datamaskinen bearbetar ett program P1 med lägsta prioritet 1. Maskinen bryter P1 och hoppar direkt till brytsignalens program P3 men lagrar alla delresultat för P1. Medan P3 bearbetas, inkommer en brytsignal för program P5, varvid P3 bryts och maskinen hoppar till program P5. Medan P5 bearbetas, er-

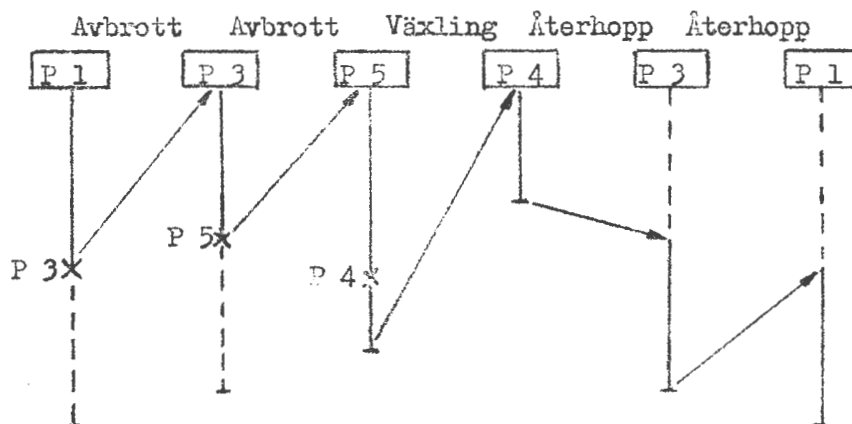


FIG. 30

hålles brytsignal för program P4, som dock har lägre prioritet än P5 varför



det senare slutföres innan programväxling sker till P4. Först då P4 är färdigbearbetat, kan maskinen hoppa tillbaka till P3 på den plats i P3, där avbrott skedde. P3 slutföres, varefter till sist P1 kan behandlas. Avbrotten sker alltså i växande prioritet och återhoppet i fallande.

Förutom för simultanbehandling av flera processprogram är avbrottstekniken väl lämpad för kontroll av datautrustningen själv. Om den lägsta prioriteten avdelas för ett automatiskt maskintestprogram, kommer datamaskinen automatiskt att genomlöpa ett sådant under ledig tid. Fig 31 visar ett tänkbart sätt, där maskinen kör vissa av de aktuella programmen med kända data, varvid svaren kan checkas. För mer allmän kontroll kan ett sådant testprogram även innefatta körning av samtliga instruktioner med värsta tänkbara instruktions- och datakombinationer.

Om vid körning av ett testprogram ett fel upptäcks, kan maskinen antingen enbart signalera detta fel eller vid processer, som är ömtåliga för felaktig påverkan från datamaskinen, koppla över styrfunktionerna till en manuell kontroll samt ge larm. Om felet är av sådan art att den normala testrutinen inte kan lokalisera det, bör maskinen automatiskt koppla in ett diagnosprogram, som är av sådan omfattning att fellokalisering kan ske varefter diagnosresultatet meddelas operatören via skrivmaskin eller annat presentationsorgan.

För dubbelcheckning ifall larmfunktionen skulle tänkas manka, kan man tänka sig, att maskinen även presenterar resultatet av ett framgångsrikt genomfört testprogram som angivits i fig 31.

Den angivna metoden för självtest innebär, att maskinens normala arbetsuppgifter inte behöver avbrytas för genomförande av testen. Eventuellt inkommande brytsignaler från processen ger automatiskt hopp ifrån testprogrammet, varigenom processbehandlingen inte störs. Metoden med återhopp till avbrottsstället medger också, att relativt långa testrutiner körs, så att så god checkning som möjligt erhålls. Hur lång testen än är, kan den alltid förr eller senare slutföras, genom att små bitar i sänder genomlöps under ledig maskintid. Därför är det t.o.m. möjligt att lagra långa testprogram utanför själva den centrala maskinen exempelvis på hållremsa eller magnetband, så att det centrala minnets kapacitet inte belastas.

Avbrottstekniken kan givetvis även användas för automatisk felupptäckt för själva processen, om larm erhålles från denna exempelvis för otillåtna systemtillstånd. Datamaskinen kan då hoppa till särskilda larmrutiner, som först söker kontrollera datamaskinen själv med omvandlare inräknade. Ger dessa inget fel, larmas om processfel, samtidigt som maskinen i de fall det är möj-

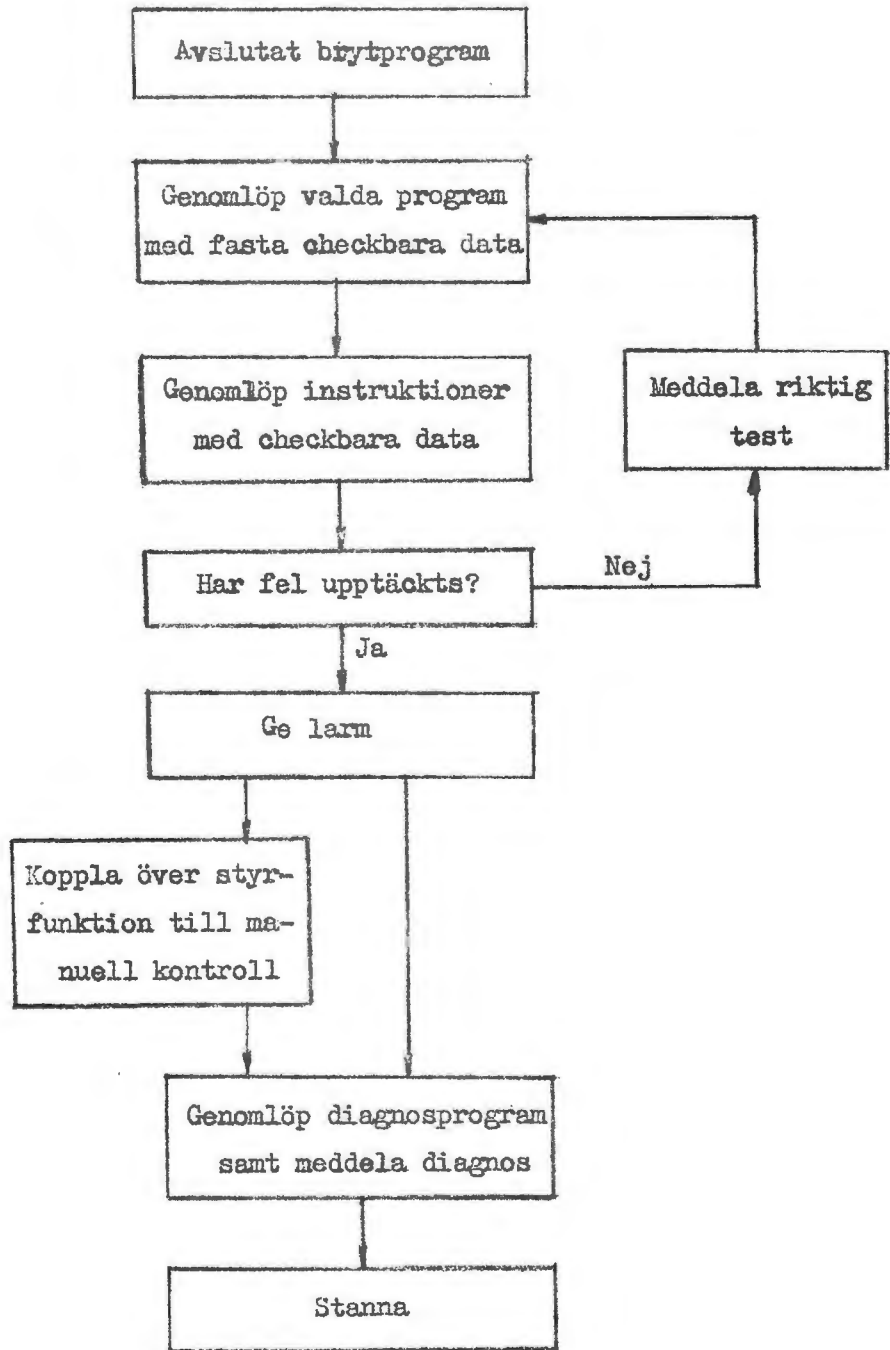


Fig 31

ligt försöker styra processen enligt bestämda program för att av processens beteende försöka utrona felkaraktären.

En viktig katastrofåtgärd som en datamaskin med hjälp av avbrott kan vidta är databevarande vid kraftavbrott i datamaskinen. Därvid får larm för kraftavbrott (från normalt befintlig detektor i kraftförsörjningen) ge en brytsignal, som åstadkommer att maskinen lägger undan alla delresultat och instruktioner i ett minne som inte förstör sitt innehåll vid kraftavbrott (exempelvis ferritminne). Därefter stannar maskinen. Då kraften återkommer, ges ny signal, så att alla data och instruktioner kan framhämtas igen till arbets-kretsarna som normalt är av dynamisk karaktär och alltså sårbara vid kraftbortfall. På detta sätt undviks data- och instruktionsbortfall med åtföljande nyinläsning från hållremsor etc. Detta skydd är synnerligen önskvärt vid automatisk inkoppling av exempelvis reservaggregat, om huvudkraften bortfaller. Eftersom undanläggningen av data tar en viss tid, skyddar denna åtgärd dock endast mot transienta kraftbortfall och inte mot t.ex. kortslutning av en viktig matningsspänning.

### 3.2.3. Direkt minnesaccess

Den i föregående avsnitt berörda avbrottstekniken har inneburit ett stort steg framåt för möjligheten att använda digitala datamaskiner som centrala kontrollorgan i stora processregleringssystem. I förhållande till ett programmerat avbrottssystem har metoden med inbyggt avbrottsorgan den fördelen, att den ständiga avkänningen av brytsignaler inte behöver ta någon extra programtid och därigenom belasta maskinens kapacitet. Emellertid lider avbrottsmetoden, som den illustrerats i fig 28, av åtminstone en svaghet och det är den nödiga tiden för utförande av ett avbrott, som betingas av kraven på lagring av data för avbrutet program. Låt oss säga, att denna tid för en normal maskin rör sig om c:a 100  $\mu$ s. För vissa typer av processer kan detta vara en alltför lång tid att hålla data, som skall lämnas till datamaskinen eller vänta på data från maskinen. I synnerhet gäller detta, om en stor del av maskinens arbete består av kommunikation med yttre enheter för datainsamling och data-distribution. Omflyttning av data för avbrutna program, som strängt taget är onyttigt arbete, kan i sådana fall uppta en betydande del av maskinkapaciteten.

Ett sätt att undvika dataomflyttning vid avbrott är givetvis att låta det pågående programmet behandlas färdigt, innan programväxling sker. För att anropande organ i det fallet inte skall behöva vänta för länge på åtkomst av maskin, görs avbrott endast i den till det löpande programmet hörande minnessekvensen, som gäller hämtning av instruktioner samt förmedling av data mellan



minne och räkneorgan. Under detta avbrott kan räkneorganet stå stilla eller självständigt utföra längre aritmetiska operationer som multiplikation och division, om sådana råkar vara för handen. Avbrottet i minnessekvensen görs därvid endast för utmatning eller lagring av data för anropande yttre organ. Kommunikation kan ske direkt mellan minne och yttre organ, om maskinen är försedd med s.k. yttre busledning eller samlande överföringsledning som anges i fig 32. Även räkneorganet kan anslutas till denna busledning men också via

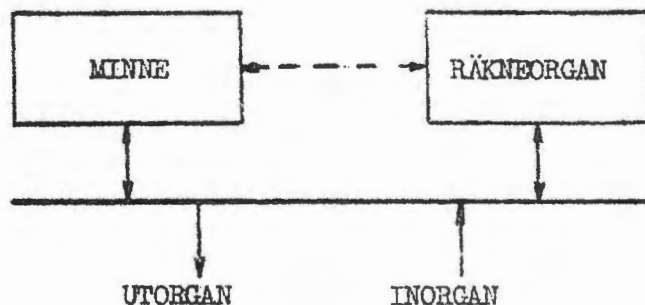
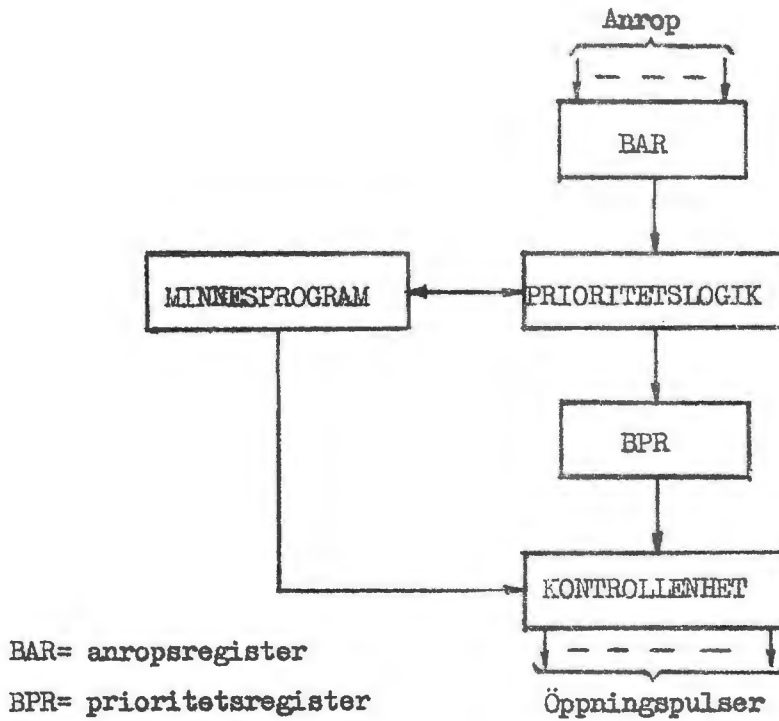


FIG: 32

separat förbindelse, om högre förmedlingssnabbhet önskas.

Vid direktkommunikation mellan yttre organ och minnet är det givet, att det förre måste kunna lämna information till maskinen dels om vilket yttre organ det gäller och dels om adress i minnet, där data önskas hämtade eller lagrade. En typ av yttre organ som kan ges sådana möjligheter är kontrollbord, från vilka alltså operatörer kan mata in data i eller avläsa data från önskade platser i minnet med hjälp av direktkommunikation. Andra typer av sådana organ kan vara exempelvis presentationsutrustningar, som är programmerade för successiv genomsökning av alla data i minnet eller datalänkar över vilka information innehållande uppgift om minnesplats levereras. En tredje typ är andra kompletta datamaskiner, varav framgår att denna metod medger direkt kommunikation mellan flera datamaskiner utan större tidsförlust.

Den här angivna typen av kommunikation med en datamaskin kräver, att denna innehåller ett avbrottsorgan även för minnet, ett organ som vi i detta fall kan kalla "anropsorgan", eftersom inga avbrott i förut angiven bemärkelse orsakas av densamma. Anropsorganet kan vara av samma principiella konstruktion som avbrottsorganet dvs försett med register för lagring av prioritet för pågående minnesoperation (prioritetsregister) och för anropande organ (anropsregister) samt med prioritetslogik för successivt verkställande av anrop alltefter prioritet. Till detta anropsorgan behöver normalt även räkneorganet anslutas, för att yttre organ i trängande fall skall kunna komma åt minnet



BAR= anropsregister  
BPR= prioritetsregister

Fig 33

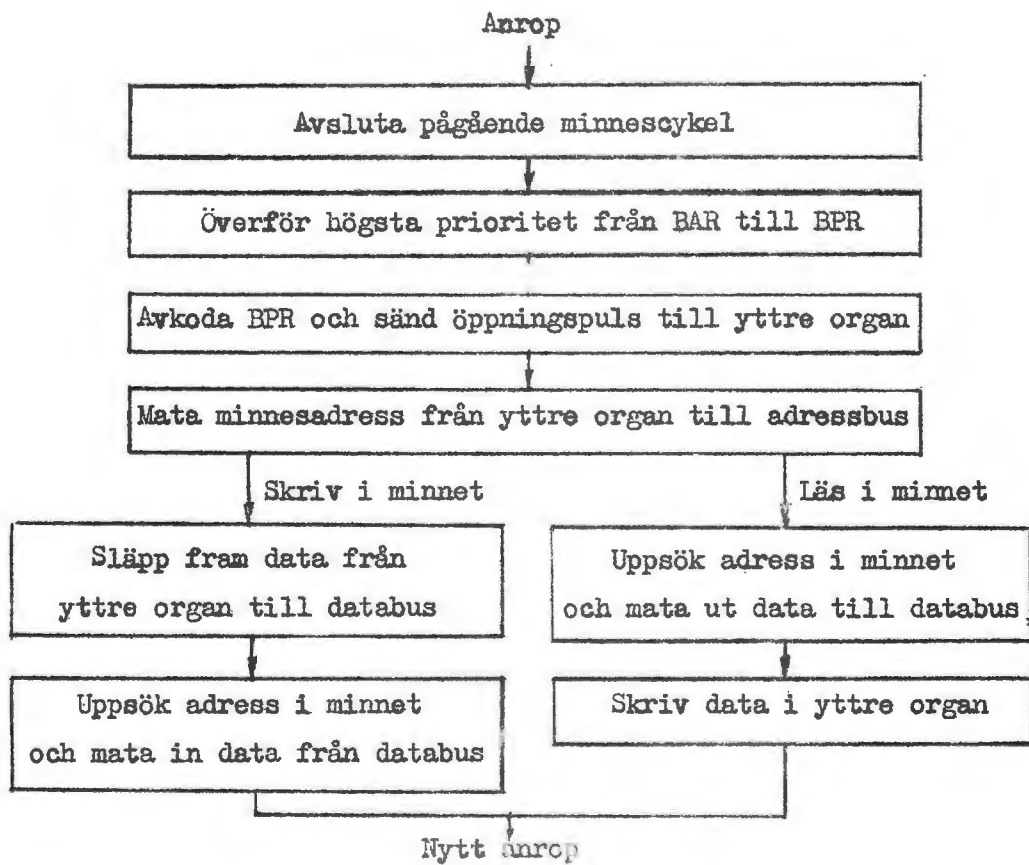


Fig 34

mitt i ett räkneprogram. Fig 33 visar principschemat för anropsorganet och fig 34 flödesschemat vid anrop.

Vad som hittills berörts, har enbart gällt datakommunikation direkt med utvald adress i minnet. I vissa fall kan en begäran om dataavlämning till maskinen vara sammankopplad med begäran om visst räkneprogram. I ett sådant fall anropas även avbrottsorganet, och då det begärda programmet tas om hand, inleds detta med hämtning av data i den minnescell som uppsöktes direkt från det yttre organet. Detta förutsätter då, att denna adress är fast för resp. yttre organ.

Det är tydligt, att metoden med direktaccess till en datamaskins minne förutom nedskärning av dataomflyttningar även reducerar behovet av databuffertar i yttre organ. Detta betingas av den snabbhet med vilken yttre organ, åtminstone sådana med hög prioritet, kan avlämna data till minnet vid exempelvis klumpvis uppträdande data med hög förmedlingsfrekvens.

I föregående avsnitt omnämndes möjligheten att med hjälp av avbrottstekniken kunna köra mycket långa testrutiner partiellt under ledig maskintid. För att långa program av denna typ inte skall blockera andra program vid utnyttjande av ovannämnda princip om avslutning av program, kan de styckas upp i delar, som var för sig får ge brytsignal till avbrottsorganet. Samma brytsignal kan därvid användas av alla delar i ett program. Avbrott kan i detta fall normalt ske med relativt långa tidsintervall; huvudsaken är, att alla begärda program behandlas inom den reelltidsram som gäller för processen. Vid styrning av mycket högfrekventa förlopp kan man dock vara tvingad att utföra programavbrott, för att styrdata skall kunna erhållas snabbast möjligt. I sådana fall används direktaccessen till minnet endast för kommunikation med data, som ej åtföljs av begäran om räkneprogram. Givetvis kan för olika tillämpningsfall varianter av ovannämnda förfaranden tänkas. Variationsmöjligheterna är emellertid så många, att det inte är möjligt att täcka dem i denna översiktsbeskrivning.

#### 3.2.4. Maskinsnabbhet. Lagrade och fasta program

Man säger ofta, att en datamaskins snabbhet bestäms av minnets snabbhet som vanligen är den begränsande faktorn. Detta är framför allt sant för maskiner med lagrade program, dvs där alla programpunkter är lagrade i maskinens minne och måste hämtas därifrån steg för steg. Att minnets snabbhet är sämre än räknekretsarnas är helt naturligt med den typ av minnen, som i dag är tillgängliga. Självklart gäller detta förhållande för trumminnen och långsammare typer p.g.a. den långa accesstiden, men även s.k. direktaccessminnen (random access memories) med högsta tillgängliga snabbhet är långsammare än räknekretsarna. Detta förhållande beror på att en minnesoperation kräver flera



successiva kretsoperationer samt drivning av relativt hög ström. Minnen för lika snabb läsning som skrivning fordrar också en dubbeloperation, normalt tömning (läsning) plus återställning (skrivning). En minnesoperation är därför i allmänhet mer komplicerad än en enkel operation i räkneorganet och därför långsammare. Så länge även enkla operationer programmeras i minnet, blir maskinens snabbhet begränsad av minnessnabbheten. I generell användning av datamaskiner är emellertid minneslagring av program högst önskvärd, eftersom detta medger att maskinen själv under databehandlingens gång ändrar programmet efter behov, en facilitet som visat sig mycket användbar i modern databehandling.

Som tidigare nämnts, är digitalutrustningens snabbhet ofta en av de avgörande faktorerna för dess användbarhet i processreglering. Samtidigt är det ofta så, att behovet av programmodifiering inte är särskilt stort. Maskinen programmeras i allmänhet en gång för alla för vissa arbetsuppgifter, som utförs mycket ofta utan större variationer. I sådana fall kan utnyttjande av s.k. fasta program höja maskinens snabbhet avsevärt. Ta som ett mycket extremt exempel, att en viss maskins uppgift till 50% är att utföra rotutdragningar och att en sådan operation med fast program tar endast 10% av tiden för motsvarande med minneslagrat program. Med fast programmerad rotutdragning kan därvid 45% av maskintiden inbesparas.

Orsaken till att de fasta programminnen kan göras snabbare än de föränderliga har flera orsaker beroende på vilken typ av fast minne som väljs. En välkänd typ av minne för snabb avläsning är ferritminnet med s.k. icke raderande avläsning utnyttjande exempelvis flerhålskärnor eller partiell switchning av enhålskärnor. Normalt går inskrivning av information i sådana minnen mycket långsammare än läsning, vilket emellertid spelar mindre roll eftersom inskrivning endast behöver ske vid ändring av de lagrade programmen. Sådan inskrivning kan då exempelvis göras från hållremsa. Till fasta ferritminnen hör vidare sådana, där de enskilda kärnorna i en matrisuppsättning kan förhindras att avge signal p.g.a. läs- eller drivtrådarna kortsluts förbi resp. kärnor, så att en önskad kombination av pulser erhålls vid läsning av varje adress. Sådana minnen är omprogrammerbara genom mekanisk bygling över kärnorna. Ytterligare ett sätt att åstadkomma fasta minnen är att utnyttja en mängd små lufttransformatorer ofta bestående av etsade induktanser i metallfolie, som parvis läggs nära varandra. Med ännu ett etsat folie inskjutet mellan ett sådant par kan de parvisa induktanserna avskärmade från varandra (metall i foliet mellan induktanserna) eller inte (metallen bortetsad). I de senare fallen erhålls koppling vid pulsning av induktanserna. I ett sådant fall

sker omprogrammering genom inskjutning av mellanfolier med nya mönster av skärmande metall. Man kan även gå tillväga på motsatt vis och i stället förstärka kopplingen mellan två induktanser genom instoppande av ferritstavar i spolfält.

De ovannämnda minnena brukar även kallas semipermanenta programminnen, eftersom informationsinnehållet med relativt enkla medel kan ändras. Det är också möjligt att konstruera minnen med hjälp av uppkopplade diodmatriser, där diodförbindningar bildar de önskade koderna (se fig 35). Ett sådant minne är i regel fast och måste lödas för omprogrammering, så vitt inte mångpoliga kontak-

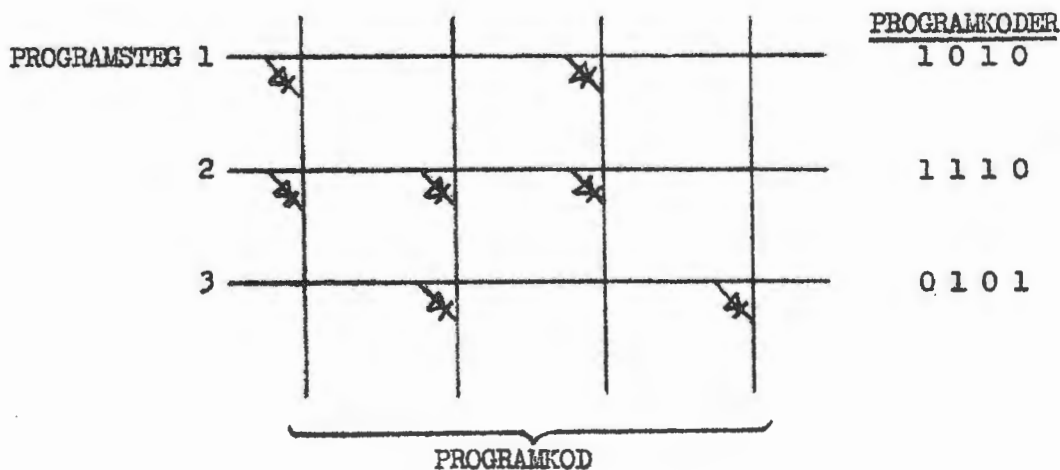


Fig 35.

ter med varierande stiftförbindningar används till att koppla in olika diodkombinationer. Denna typ av minne är mycket snabb och även billig för mindre minnen men lämpar sig inte för större sådana.

I vissa specialbyggda datamaskiner med bestånda arbetsprogram kan hela maskinprogrammet ligga i ett fast minne av ovan angivet slag. Alla data däremot läggs i ett arbetsminne av konventionell typ exempelvis ett normalt ferritminne. I andra fall, där programmeringen kan behöva ändras under databehandlingen, kan det fasta programmet användas för en del kortare program (subrutiner), till vilka maskinen kan hoppa från det minneslagrade programmet. Det fasta programmet kan därvid behandlas som en del av arbetsminnet ur adresseringspunkt och hopp från minneslagrat till fast program kan göras med hjälp av Wheelerhopp eller på order av brytsignaler. Givetvis måste vid varje hopp



klockfrekvensen för minnet anpassas till resp. minnes snabbhet. Exempel på ett lämpligt program att lägga in i fasta minnen är det tidigare nämnda avbrotts- och återhoppsprogrammet, vilket bör utföras så snabbt som möjligt.

Tidsvinsterna med fasta program gör sig mest gällande för sådana rutiner, där det normala minnet sällan behöver uppsöka nya data. Sådana rutiner är exempelvis ekvationer och serieutvecklingar för enstaka variabler. Villkoret är givetvis dock, att sådana beräkningar används mycket ofta i maskinen. I sådana fall kan minnet långa perioder bli obelastat av beräkningsrutiner och får i stället möjlighet att betjäna yttre organ med hjälp av direktaccess, som tidigare omnämnts (se avsnitt 3.2.3). En kombination av fast programmering och direkt minnesaccess kan därför i många tillämpningar avsevärt höja maskin-kapaciteten.

### 3.2.5. Datasökning. Dataadresserade minnen

Inom alla områden av databehandling ställs ofta en datamaskin inför problemet att söka rätt på en viss information i ett minne, utan att maskinen på förhand känner minnesadressen för denna information. Det normala är i sådana fall, att den sökta informationen är känd och att man är intresserad av andra informationer som sammanhör med den sökta. Det vardagligaste exemplet på sådan informationsbestämd sökning är väl utnyttjandet av telefonkatalog eller nummerbyrå för uppletande av nummer för känd abonnent eller abonnent för känt nummer. Inom processregleringen kan ett exempel ur tidigare berörda systemfall tas nämligen följning av en processvariabel exempelvis vid automatisk målföljning i radar (se avsnitt 2.2.1.9). De i radarn mottagna flygplanekona skall associeras med i datamaskinen upprättade flygbanor. Den kända informationen är i detta fall det mottagna ekots positionsdata och den önskade är alla viktiga data för den flygbana, som kan innefatta dessa positionsdata. Principen för sökningen är alltså en jämförelse mellan mätdata och motsvarande predikterade data för alla flygbanor. Som nämdes i avsnitt 2.2.1.9 är ett sådant sökförfarande genom successiv jämförelse mycket omfattande vid många följningar, varför en vanlig metod är att gå motsatta vägen och låta predikterade flygbanepositioner i stället leta efter sambörande mätdata genom luckbildning. Denna metod är mycket effektiv men kräver även den en hel del strängt taget överflödigt databehandling. Parametrarna för alla lagrade banor måste nämligen ständigt genomsökas i minnet, så att luckbildning kan ske vid de tidpunkter då resp. flygplan ger eko i radarn. En nackdel med metoden är dessutom som tidigare nämnts, att någon automatisk uppstartning av nya banor inte kan ske eftersom inga luckor finns för dessa. I det fallet är metoden med generell jämförelse däremot användbar.

Det är tydligt, att sådana typer av informationssökning som angivits ovan skulle kunna genomföras avsevärt effektivare med ett minne som kan adresseras med



hjälp av minnesinformation i stället för minnesposition. Forskningen på sådana minne har intensifierats på senare tid och en del experimentmodeller byggts. Minnena kallas varierande för dataadresserade eller associativa minnen eller katalogminnen. I ett dataadresserat minne jämför man den kända informationen med motsvarande delinformation (märkdata; i det ovan exemplifierade fallet positionsdata för ekon) i samtliga minnesadresser under en enda minnescykel. Som resultat erhålles i princip minnesadressen för lagrade data, som eventuellt koincerat med den kända informationen. Denna adress används sedan till att på konventionellt sätt hämta fram all övrig samhörande information, som önskas, eller för att skriva in ny information.

De dataadresserade minnena är mer komplicerade än de konventionella och kräver än så länge stort uppbåd av komponenter för lagring av större datamängder. Experiment med olika komponenter har gjorts i USA både med magnetiska och supraledande minneselement. De senare i form av tunnfilmkryotroter tycks vara de mest lovande dels p.g.a. kompaktitet och dels p.g.a. enkla möjligheter att samtidigt fungera som minnes- och logikelement. Här finns inte utrymme för någon genomgång av minnernas funktionsätt, eftersom detta kräver en ganska ingående beskrivning av krets- och komponentegenskaper. Några litteraturreferenser har i stället givits i slutet av kompendiet. Här nedan skall endast ett mycket litet och enkelt minne beskrivas för att förklara principen för dataadressering.

Fig. 36 visar ett magnetiskt minne för 3 ord om vardera 3 bitar. Varje ord förekommer i två former, en sanna form och en komplementär, placerade i var sin del av minnet. Vid sökning placeras märkinformationen i dataregistret. Varje bit i detta register styr drivningen av motsvarande bit i minnet (i alla ord samtidigt) på så sätt, att en nolla i registret driver den sanna sidan medan en etta driver komplementetsidan. Drivningen åstadkommer, att en kärna som lagrar en etta ger puls på den genomsdragna läsledningen medan en nolla ingenting ger. Det sätt på vilket drivningen av de båda sidorna sker åstadkommer, att ett koincerande ord inte ger signal på läsledningen för någon bit vilket antytts i figuren med ettor och nollor på läsledningarnas utgångar. Detta gäller endast det koincerande ordet. Genom avkodning av utsignalerna kan alltså adressen för detta ord erhållas.

Nya data kan skrivas in i minnet exempelvis genom kombinerad drivning på läs- och drivledningar de senare styrda av inskrivningsdata i dataregistret.

P.g.a. att dataadresserade minnen åtminstone f.n. är avsevärt dyrare än normala minnen för större datamängder kan de med fördel användas enbart för lagring av märkdata tillsammans med adress till ett normalt minne, där övriga

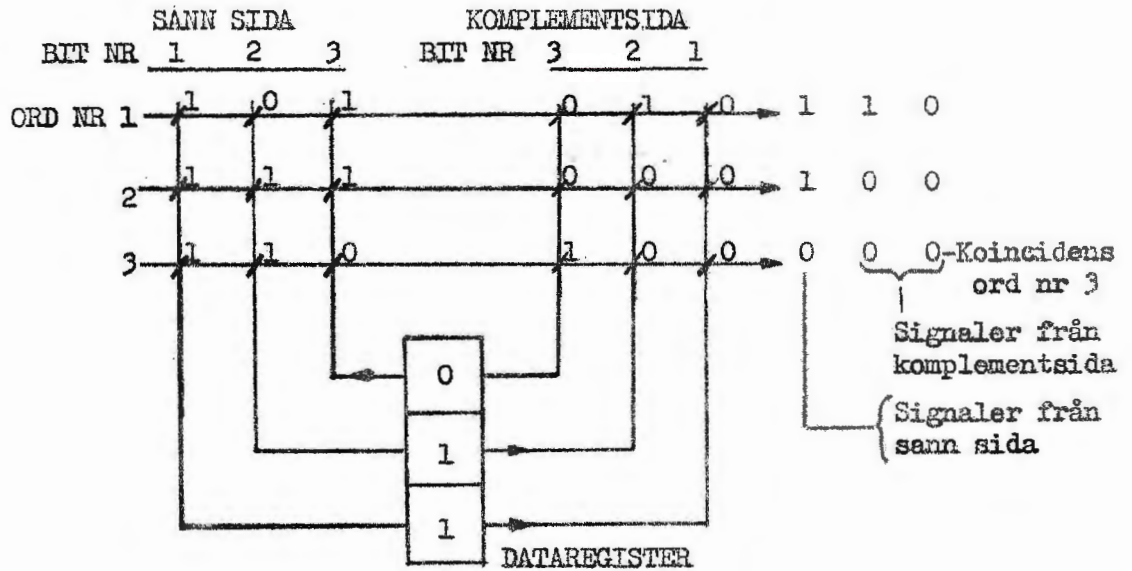


Fig. 36

sammanhörande data finns. Därigenom kan storleken av det dataadresserade minnet hållas nere.

### 3.3. Speciella digitalsystem

Som tidigare nämnts finns många speciella tillämpningar, där antingen omfattningen av databehandlingen är så begränsad att en specialutrustning är billigare än en generell datautrustning eller också är kraven så speciella, att en generell maskin inte kan uppfylla dem. Fortsättningen av detta avsnitt skall ge exempel på några sådana specialutrustningar. Området speciella digitalutrustningar är emellertid så överväldigande stort, att inte ens något försök till översikt görs här, utan framställningen inriktas på några typer av utrustningar, som fått vidsträckt användning i reelltidssystem.

#### 3.3.1. Digitalstyrda servon

Redan i avsnitt 2.2.1.1 gavs som exempel på en enkel typ av processreglering ett återkopplat digitalstyrt servo. Sådana servon används ofta för noggrann numerisk styrning av axelrörelser i en mängd olika tillämpningar. Eftersom denna utrustning i sin enkelhet är ganska representativ för ett processregleringssystem, skall här nedan några konstruktionsproblem beröras.

Ett ofta förekommande fall, är att servosystemet styrs från en datamaskin, som med vissa tidsmellanrum matar styrdata till servot (indata i fig 2). Som tidi-

gare nämnts kan man då råka ut för problemet, att samplingsfrekvensen för denna utmatning av olika skäl måste göras låg i förhållande till servots naturliga frekvens. Om jämförelsen mellan är- och börvärden görs endast vid dessa samplingstillfällen, kan risk för självsvängning föreligga (jfr fig 7). Av detta skäl måste man ofta förse servosystemet med en egen digital återkoppling, som samplas med mycket högre frekvens utan att datamaskinen störs. Med en mycket liten ändring av principschemat i fig 2 på sätt som visas i fig 37, kan ett sådant system åstadkommas. Indata från den styrande datamaskinen lagras i ett

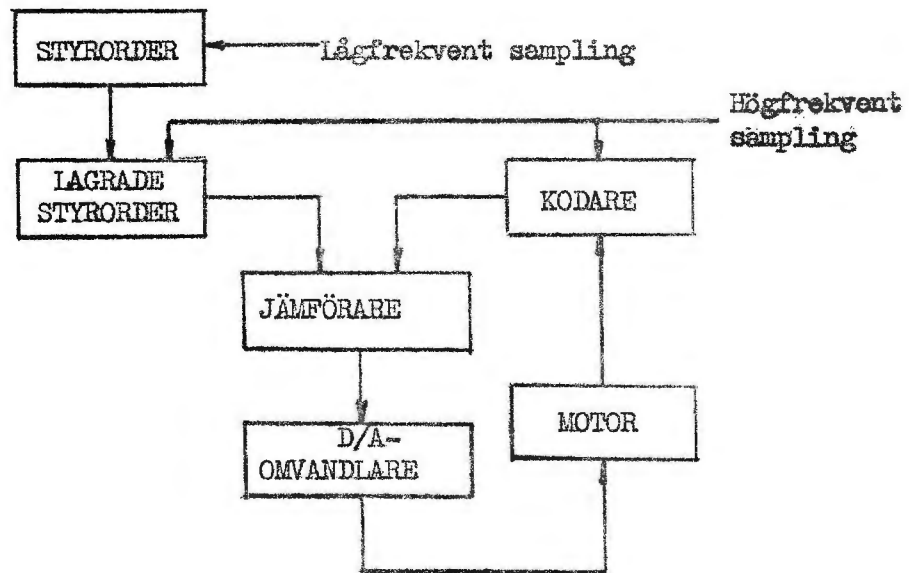


Fig 37

särskilt register mellan samplingstidpunkterna för datamaskinen. Dessa lagrade värden samt ärvärdena från den på motoraxeln monterade kodskivan samplas med högre frekvens och jämförs för bildande av det styrande felet. Om den senare samplingsfrekvensen är tillräckligt hög i förhållande till servots egenfrekvens, kommer servot att mellan varje datamaskinsampling svänga in sig stabilt mot styrdata.

Fig 38 visar ett mer detaljerat schema över hur ett digitalservo kan se ut. På order från det styrande systemet (exempelvis datamaskin) samplas en styrorder (indata) och lagras i ett styrorderregister. Mellan dessa samlingar utför ett internt program i servosystemet ett antal arbetscykler för intern sampling. Programmet inleder varje cykel med sampling av den på den styrda axeln monterade kodskivan och lagrar avläst värde i ett ärvärde register. Normalt är kodskivan cykliskt kodad, varför en omvandling till normal kod måste göras innan ärvärdet kan användas. Denna omvandling kan antingen göras i parallell form vid samplingen (som i fig) eller i serieform i registret före vi-



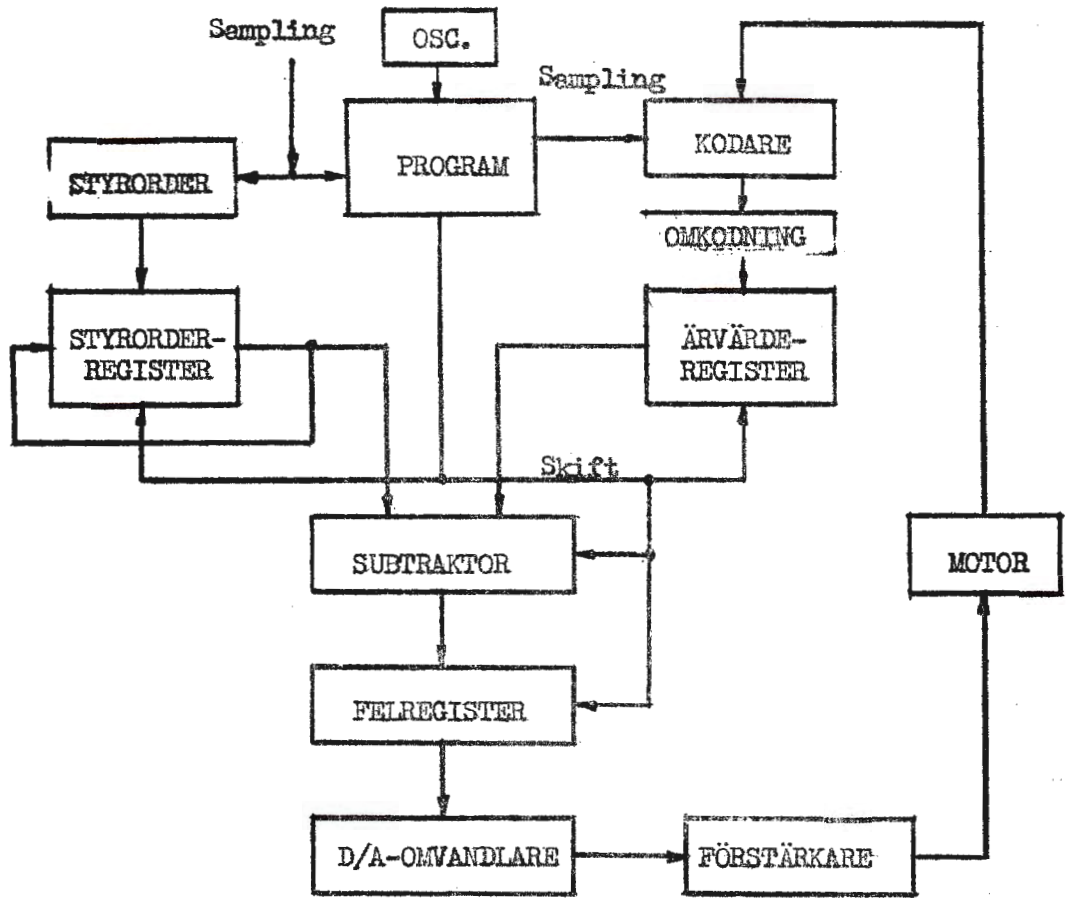


Fig 38

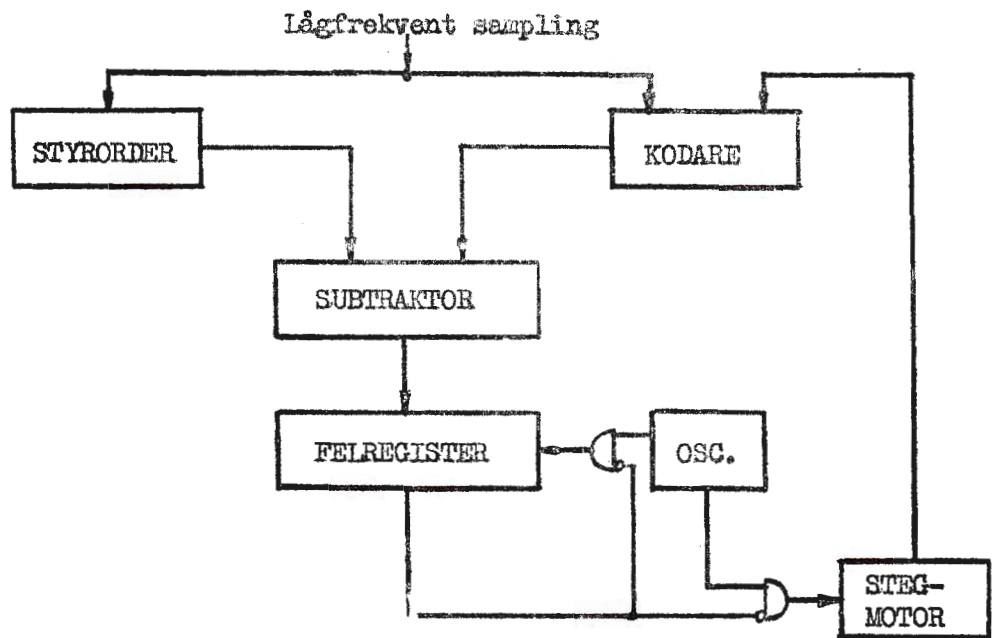


Fig 39

dare användning. Efter denna sampling med omvandling startar programmet ett skiftförlopp, varvid styrorder- och ärvärderegister skiftar sina data in i en subtraktionskrets där skillnaden mellan de skiftande data beräknas och sedan lagras i ett felregister. Digitalvärdet i detta register matar en digitalanalogomvandlare, som via förstärkning styr motorn för den utgående axeln.

#### "Närmaste väg"-problemet

Då den utgående axeln roterar förbi det läge, som i kodskivan indikeras som nolläge, uppträder en diskontinuitet i den avlästa koden, eftersom denna växlar mellan nollvärde och maxvärde. Antag att servot skall svänga in från ett läge  $360^\circ$  till ett läge nära  $0^\circ$ . Det får då rotera nästan ett helt varv baklänges och riskerar att temporärt svänga över nolläget. Om detta sker, fås återigen ett helt varvs fel i baklängesriktningen, varför servot hela tiden kommer att rotera och inte stabilisera sig. Även om detta inte sker, så är det tydligt, att servot ibland måste vrida sig större delen av ett varv medan det är önskvärt att låta det kunna röra sig närmaste vägen mellan två inställningslägen.

Olika metoder har använts för att lösa detta "närmaste väg"-problem, men följande torde vara den enklaste. Antag för enkelhets skull, att servot är försett med en 4-bitars kodskiva som efter omkodning ger koderna 0000 för  $0^\circ$ , 1000 för ett halvt varv och 1111 för  $15/16$  varv. Om man vid den aritmetiska jämförelsen i subtraktorn låter den högsta biten fungera som teckensiffra, är "närmaste väg"-problemet löst. Med denna metod får alla vinklar mellan  $180^\circ$  och  $360^\circ$  negativa binärkoder i form av komplementtal. Ett halvt varv får koden 1.000 dvs minus ett halvt varv.  $15/16$  varv blir 1.111 eller  $-1/8$  av ett halvt varv (talområdet är plus och minus ett halvt varv). Genom denna tolkning av koden får felet efter jämförelsen sådant tecken, att servot styrs närmaste vägen till börvärdet oberoende av om diskontinuitetspunkten (i detta fall vid  $180^\circ$ ) passeras eller ej. Antag att servot skall röra sig från vinkeln 0.111 ( $7/16$  varv) till vinkeln 1.001 ( $9/16$  varv eller  $-7/8$  av ett halvt varv). Felet vid jämförelsen blir då  $1.001 - 0.111 = 0.010$  dvs två delningar i positiv riktning, varför servot vrider sig via  $180^\circ$  till börvärdet. Lika lätt passerar servot över  $0^\circ$ . Antag att börvärdet är 1.111 ( $15/16$  varv) och ärvärdet 0.001 ( $1/16$  varv). Felet blir då  $1.111 - 0.001 = 1.110$  dvs två delningar i negativ riktning, varför servot vrider sig via  $0^\circ$  till börvärdet. Om högsta siffran inte tolkats som teckensiffra, skulle felet blivit 1110 dvs 14 delningar i positiv riktning, och servot hade fått vrida sig  $14/16$  varv.

Fig 39 visar en annan typ av servo, där motorn är en stegmotor driven av en oscillator. Efter beräkning och lagring av felet i felregistret räknas detta register ner av oscillatoren, samtidigt som motorn stegas fram i motsvarande

grad. Då felet är noll, avbryts stegningen. Oscillatorn kan ersättas av kodskivan, om denna fungerar som pulsgivare. I detta fall sätts felregistret direkt från det styrande organet med värdet på den förändring, som servot skall undergå. Denna servotyp används bl.a. vid numerisk styrning av bearbetningsmaskiner. Nackdelen med metoden är, att pulsfel ger bestående lägesfel.

### 3.3.2. Datalänkterminaler

#### 3.3.2.1. Allmänt

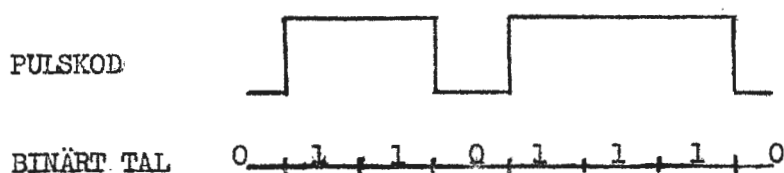
Som omnämndes i avsnitt 2.2.3.1 är det ofta fördelaktigt att använda speciella utrustningar för anpassning mellan kommunicerande datasystem och förbindelsemediet (jfr fig 24). Anpassningsbehovet gäller därvid främst informationens överföringstyp (parallellform i datamaskin kontra serieform i förbindelsemediet), förmedlingsfrekvens, säkerhetskodning mot störningar i förbindelsen samt modulationstyp. Ett annat anpassningsbehov i detta sammanhang gäller synkronisering mellan sändande och mottagande system, vilken också kan åstadkommas samtidigt med den ovannämnda anpassningen. Den digitala delen av en sådan anpassningsutrustning (dvs exklusive modulatorerna) kallas dataterminaler och är av två slag, sändande och mottagande.

För erhållande av lägsta nödvändiga bandbredd på överföringen är det fördelaktigt att sända kodpulserna utan pulsmellanrum mellan successiva ettor (se fig 40 a). Baudtalet på förbindelsen blir i detta fall detsamma som antalet överförda bitar per sekund.

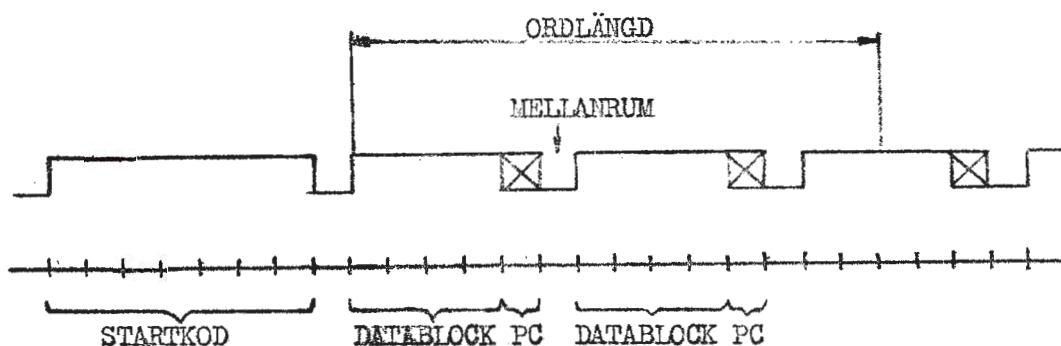
Den synkronisering som erfordras mellan sändare och mottagare, för att den senare skall kunna detektera meddelandena på länken, är av två slag dels synkronisering per puls och dels synkronisering per meddelande. Den senare gör, att mottagaren vet var ett överfört meddelande börjar resp. slutar, och åstadkommes genom en startkod i början av varje meddelande. I det fall man utnyttjar pulsöverföring med pulsmellanrum är denna kod enkel att införa, eftersom den i princip kan bestå enbart av exempelvis en dubbel pulslängd. I fallet utan pulsmellanrum är detta inte möjligt, eftersom en sådan kod också kan uppträda var som helst i ett meddelande. Man kan lösa problemet genom att uppdelar varje meddelande i block med viss längd, som är åtskilda av en nolla per block. Startkoden görs då längre än ett block och kan således detekteras genom sin längd. Ett sådant meddelande kan ha utseendet enligt fig 40 b. En delinformation (ord) inom ett meddelande kan omfatta ett eller flera block eller delar av block. För varje block inför man ofta redundans i form av en paritetssiffra (PC: förkortning för eng. parity check), som fyller ut blocket så att det alltid innehåller exempelvis udda antal ettor. Beroende på om antalet ettor i ett



block är udda eller jämnt blir alltså paritetssiffran en nolla eller en etta.



a/



b/

Fig 40

Denna siffra används sedan för blockkontroll i mottagarterminalen.

### 3.3.2.2. Sändarterminaler

Fig 41 visar ett förenklat blockschema över en sändarterminal. Hjärtat i denna är en oscillator med en frekvens lika med baudtalet på överföringen. Denna oscillator styr ett program, som sammansätter de utgående meddelandena. Primärinformationen från det dataavlämnande systemet (exempelvis en datamaskin) överförs i parallellform till ett skiftregister, sedan programmet givit klarsignal för ny information och erhållit startsignal från datamaskinen. Programmet bildar därefter startkod, om det gäller början av ett nytt meddelande. Efter startkod och skiljenolla (fig 40b) lämnar programmet en följd av pulser till registret, som skiftar dettas information i serieform till modulatern för modulerad överföring på länken (jfr fig 24). Under utskiftningen räknas antalet utsända ettor i en paritetsräknare PR, som i princip utgörs av en vippe synkroniserad med skiftpulserna. Då ett block utskiftats, stoppas skiftningen och PR avläses i en paritetsgrind PC. Beroende på PR:s värde utmatas från denna en etta eller nolla till modulatern, varefter PR nollställs. Efter ny skil-

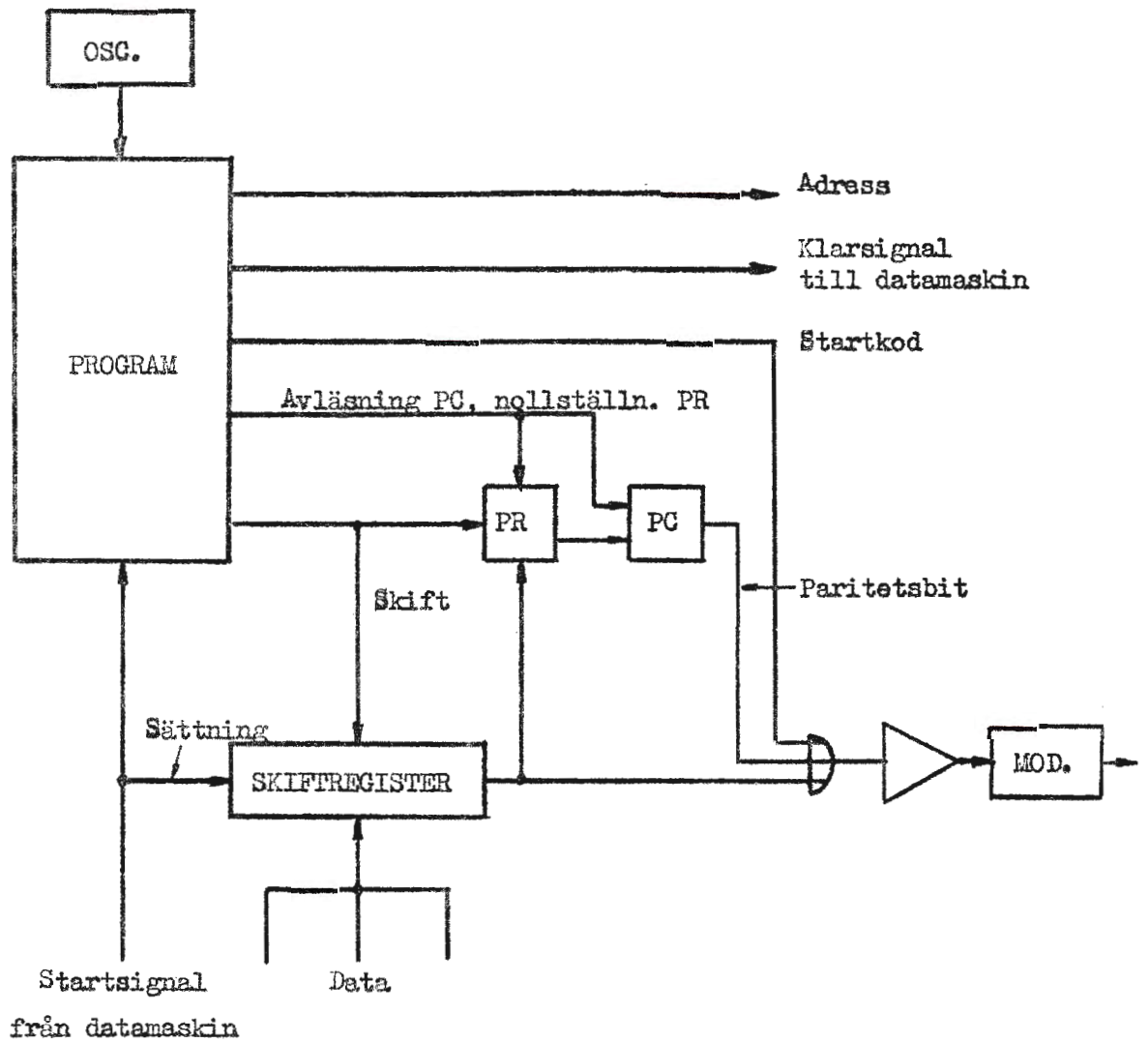


Fig 41

jenolla återupptas skiftningen osv.

Om kommunikationen mellan sändare och datamaskin är tillräckligt snabb exempelvis om den senare har avbrottsmöjligheter, kan ordlängden i sändarens register hållas nere till storleksordningen längden av ett ord inom meddelandet. För varje gång ett ord skiftats ut, avger då sändaren klarsignal till datamaskinen för uthämtning av nytt ord, som måste erhållas inom en pulstid. I detta fall måste sändaren ofta lämna någon typ av adress, som talar om för maskinen vilken delinformation som önskas.

En sändarterminal av beskrivet slag kan givetvis förmedla data från andra system än datamaskiner. Ett exempel är avsökning av statistiska minnen i form av omkopplare, reläkontakter etc. I ett sådant fall kan sändningsregistret undvaras och programmet avsöker i stället successivt bit för bit i det statistiska minnet under blocktiderna.

### 3.3.2.3. Mottagarterminaler

Fig 42 visar det förenklade blockschemat för en mottagarterminal. Uppbyggnaden är likartad sändarfallet. Liksom i detta organiseras i mottagaren arbetet av en oscillator med tillhörande program. Oscillatorfrekvensen är densamma som baudtalet på överföringen. För åstadkommande av frekvensöverensstämmelse mellan sändare och mottagare är bådas oscillatorer ofta kristallstyrda. Möjligt är också att på mottagarsidan använda en resonanskrets med högt Q-värde och resonansen avstämd till överföringens baudtal. En sådan ger svänghjulverkan och rättar också in sig i fas med pulserna från länken, vilket är nödvändigt för att mottagaren skall kunna känna av inkommande pulser. Vid puls-sändning utan pulsmellanrum krävs emellertid ett mycket högt Q-värde och därigenom långa insvängningstider. Man kan i stället på digital väg åstadkomma infasning av en kristallstyrd till baudtalet på länken avpassad frekvens.

På samma sätt som i sändaren kontrollerar programmet i mottagaren informationsfördelningen i ett meddelande. En separat enhet, synkdetektorn, upptäcker genom räkning av antalet bitar i pulserna eventuellt förekommande startkoder. Förekomsten av en sådan synkroniserar programmet meddelandevis, så att det kommer att veta tidslägena för alla i meddelandet ingående delar. Under blocktid skiftas länkdata in i ett skiftregister, och under samma tid räknas antalet ettor udda jämt i en paritetsräknare PR. I slutet av varje block undersöker programmet paritetsförhållandet i en paritetskontroll PC. Om detta är felaktigt under flera block, ges larm. På samma villkor som för sändaren kan mottagaren nöja sig med att i skiftregistret lagra enbart delinformationer i ett meddelande och för varje inskiftad sådan ge avlämningsignal till exempelvis en data-



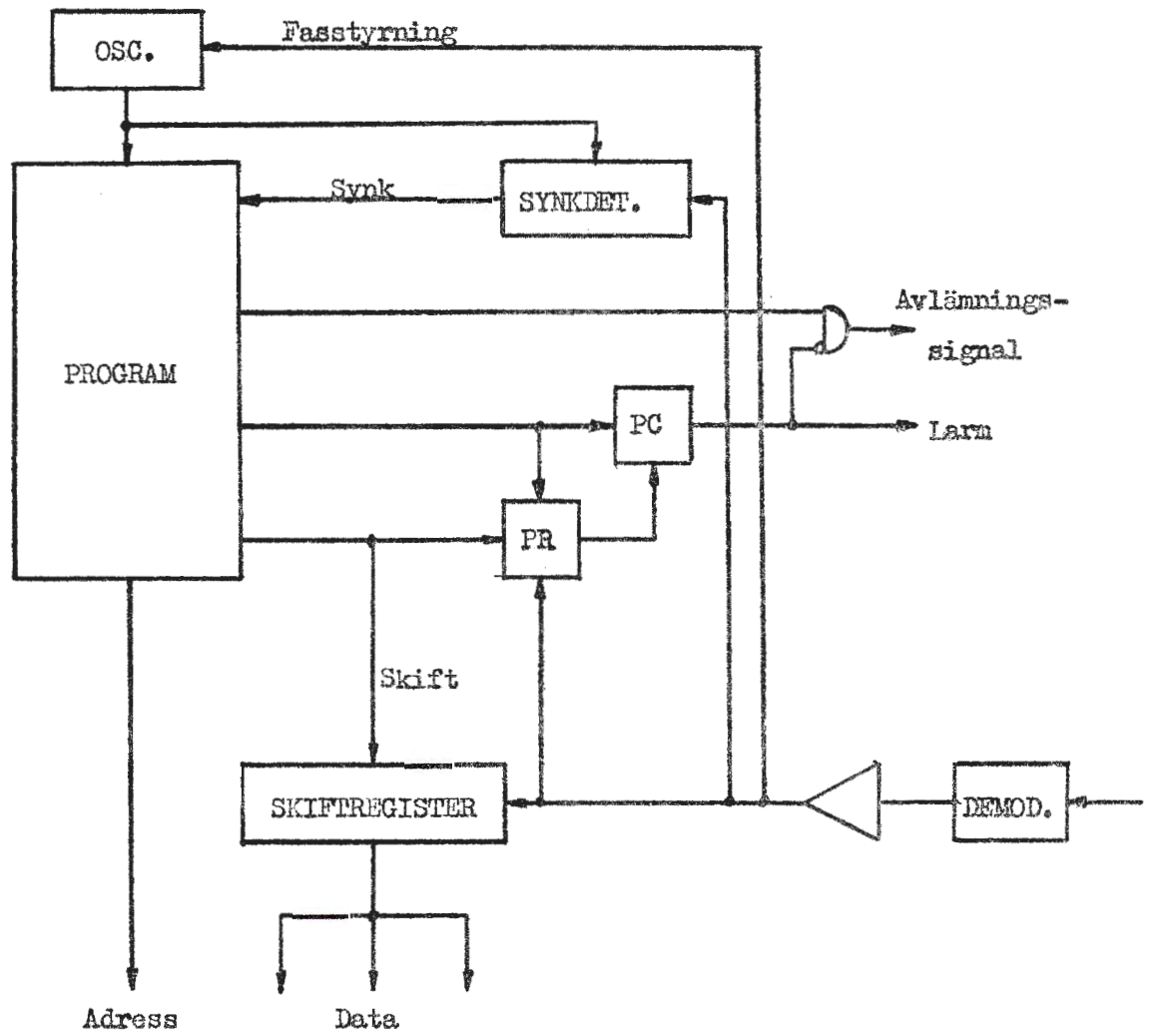


Fig 42

maskin som datakonsument. Sådan avlämningsignal blockeras dock om någon paritetssiffra i informationen varit felaktig. Även i detta fall kan adressuppgift om delinformation behöva lämnas till maskinen.

### 3.3.3. Den digitala differentialanalysatorn

#### 3.3.3.1 Allmänt

Den digitala differentialanalysatorn (DDA) konstruerades ursprungligen som en relativt billig digital datamaskin för lösning av differentialekvationer. Denna maskintyps möjligheter har senare lett till omfattande användning i reelltidssystem av skilda slag; för kontrolländamål och liknande används små specialutrustningar av denna typ och för simulering av större system mer komplicerade och snabba typer med generellare användningsmöjligheter. En intressant och viktig utvecklingsgren är sammanslagning av generell datamaskin och differentialanalysator i en enda maskin.

En DDA är, på samma sätt som den analoga motsvarigheten, normalt uppbyggd av ett antal databehandlande element, mest integratorer, som var för sig kräver mycket lite utrustning. Dessa element ansluts till varandra på olika sätt, beroende på typ av problem som skall lösas. Förbindelserna mellan elementen förmedlar snarare förändringar i de genererade funktionerna än värdena själva för dessa senare. Den förmedlade informationsmängden blir därigenom mindre än för normal digital databehandling, samtidigt som de matematiska beräkningarna blir enkla även för komplicerade funktioner.

Differentialtekniken är användbar för beräkningar på alla typer av kontinuerliga funktioner och därför också på komplicerade reelltidsföreteelser. Fördelarna med DDA framför den analoga motsvarigheten gäller främst dess överlägsna noggrannhet p.g.a. digitaltekniken. Gentemot den generella datamaskinen har den fördelarna att för vissa applikationer vara billigare och snabbare, medan den p.g.a. det speciella arbetssättet har en begränsad flexibilitet jämfört med en generell maskin.

I fortsättningen skall en kortfattad beskrivning ges över DDA:s arbetsprinciper, varefter några smärre specialutrustningar av samma typ genomgås mer detaljerat.

#### 3.3.3.2. Principen för den digitala differentialanalysatorn

Grundelementet i DDA är integratorn, som i huvudsak är en fast programmerad beräkningsutrustning. Den arbetar cykliskt på inkrementära insignaler (differenser) och bildar likaledes inkrementära utsignaler. Under en viss cykel, säg den n:te, mottar integratorn inkrementära insignaler dels på en primär in-

gång ( $\Delta X_n$ ) och dels på en sekundär ( $\Delta Y_n$ ) och avger en inkrementär utsignal ( $\Delta Z_n$ ) (se fig 43). X är därvid integrationsvariabeln och Y den funktion som

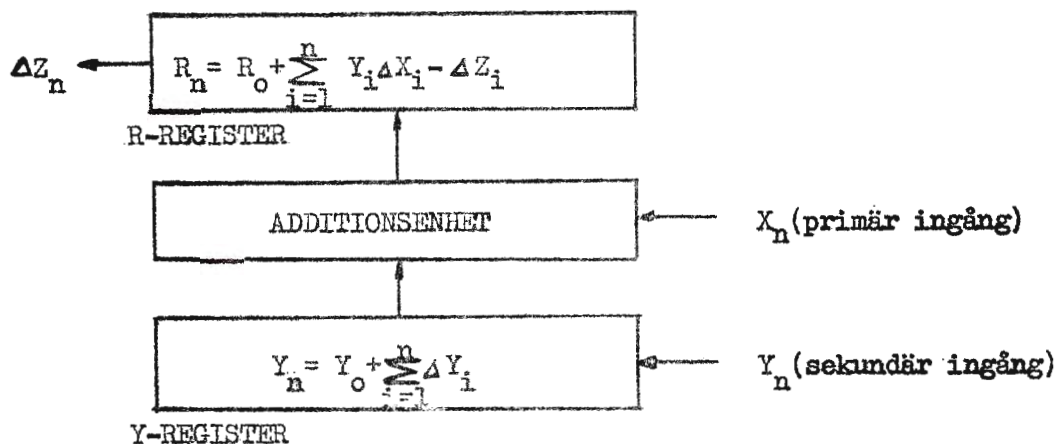


Fig 43

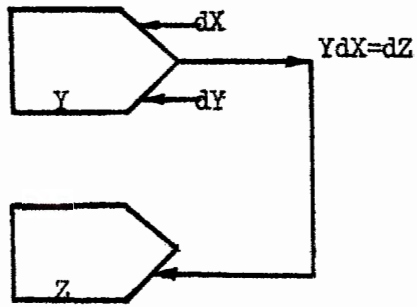
integreras. Inkrementen  $\Delta X_n$  och  $\Delta Y_n$  ges olika värden (skalas) beroende på de behandlade funktionernas gränser. X ändras alltid ett inkrement åt gången, medan Y kan ändras med godtyckliga belopp. Skalningen görs dock så, att storleken av Y begränsas till att ligga i området -1 till +1.

Integratorn består i princip av två register (se fig 43), Y-registret och R-registret. Under den n:te behandlingscykeln adderas  $Y_n$  till R-registret, om  $\Delta X_n = +1$  eller subtraheras från R om  $\Delta X_n = -1$ . Om  $\Delta X_n = 0$ , ändras inte R. Efter den n:te cykeln har additionsenheten totalt summerat värdet  $\sum_{i=1}^n Y_i \Delta X_i$ . R-registret begränsas emellertid till att kunna lagra enbart värden mellan -1 och +1. Om dessa gränser överskrids, flödar R-registret över, så att det efteråt kommer att innehålla det överflödande värdet minskat med 1 enligt nedanstående decimala exempel. Överflödet utgör därvid just inkrementet  $\Delta Z_i$

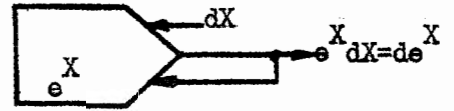
i	$Y_i \Delta X_i$	$R_i$	$\Delta Z_i$	$Z_i$
0		0	0	0
1	0,1	0,1	0	0
2	0,2	0,3	0	0
3	0,3	0,6	0	0
4	0,4	0,0	1	1
5	0,5	0,5	0	1
6	0,6	0,1	1	2
7	0,7	0,8	0	2

Tabell 3



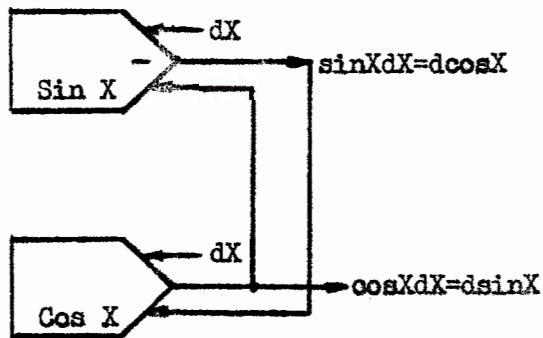


a/  
Integration av Z



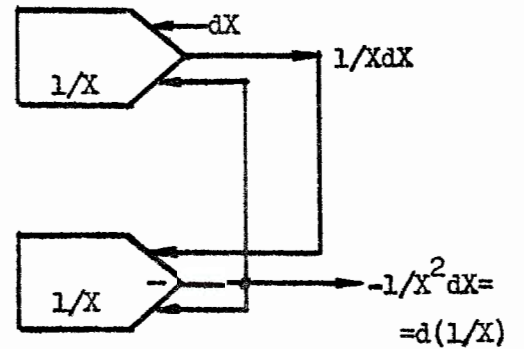
$YdX=Y$   
 $YdX=dY$   
 $dY/dX=Y$   
 $Y=e^X$

b/  
Integration av  $e^X$



-  $YdX^2=Y$   
 $-YdX^2=d^2Y$   
 $d^2Y/dX^2=Y$   
 $Y = \sin X (\cos X)$

c/  
Integration av  $\sin X$  och  $\cos X$



-  $Y^2 dX=Y$   
 $-Y^2 dX=dY$   
 $dY/dX=-Y^2$   
 $Y=1/X$

d/  
Integration av  $1/X$

Fig 45

I detta fall måste givetvis integratorn kunna känna, att  $Y_i - X_i$  är negativ och med separat signal markera att  $\Delta Z_i$  också skall vara negativ.

### 3.3.3.3. Praktisk utformning av DDA

Den praktiska utformningen av en DDA varierar beroende på funktionskraven. Eftersom maskintypen p.g.a. sitt arbetssätt är snabb, kan man ofta nöja sig med en seriemaskin, där alla integratorer behandlas efter varandra i tid av en och samma uppsättning aritmetiska element som i sin tur behandlar siffrorna i varje integrator i serieform. I andra fall, där snabbhetskravet är högre, måste parallellmaskiner användas, där flera eller alla integratorer arbetar samtidigt med var sitt enkla räkneorgan.

I en seriemaskin placerar man Y- och R-registren i ett minne, som avses cykliskt för matning till räkneorganet. En ofta använd minnestyp är därför trumminnet, som ger en relativt billig utrustning med stor kapacitet. Nackdelen med denna minnestyp är, att den inte kan synkroniseras med yttre förlopp exempelvis i snabba reelltidssystem utan erfordrar statiska buffertminnen för signaler från sådana förlopp. Frågan är också, om inte de prissänkningar på snabbaccessminnen av ferritkärnstyp som inträtt på senaste tid kommer att göra dessa mer tilltalande som minnen även i DDA. Flexibiliteten blir större med sådana minnen, eftersom den cykliska genomsökningen inte är så starkt bunden som i trumfallet. Ett sådant system kan göras mycket snabbt, om räknekretsarna arbetar i parallellform vilket lämpar sig för samarbete med exempelvis ferritminnen. Efter önskemål kan faciliteter införas, som närmar DDA till en generell datamaskin vad gäller flexibiliteten. Det är också fullt möjligt att använda DDA-teknik i generella maskiner, om dessa programmeras på rätt sätt, och detta gäller speciellt sådana maskiner, som har snabbaccessminne emedan i detta fall integratorerna kan uppsökas successivt utan tidsförluster.

En DDA av parallelltyp har för varje integrator ett minne, som för ekonomins skull ofta utgöres av en fördröjningsledning för cirkulerande data i serieform. Räknekretsarna blir därigenom också billiga, eftersom dessa likaså bör arbeta i serieform. De olika integratorerna är synkroniserade och hopkopplas fysikaliskt för varje speciellt problem på samma sätt som i den analoga DDA. Uppbådet för denna typ av maskin är normalt mycket stort och används endast för mycket speciella krav.

### 3.3.4. Den binära multiplikatorn

#### 3.3.4.1. Allmänt

Inom databehandlingstekniken är man ofta intresserad av att på ett enkelt sätt framställa olika matematiska funktioner, t.ex. beräkning av sinus och cosinus, kvadratrötter o.d. Man tillgriper då i vanliga fall tämligen snabbt konvergerande serieutvecklingar. Dessa är så beskaffade, att de på ett enkelt sätt kan utföras på den aktuella maskintypen. Analyserar man emellertid dessa lösningar kretsmässigt, finner man att de vanligen inte är de mest direkta, utan att de ger ett dåligt utnyttjande av de enskilda komponenterna. I många speciella datasystem vill man ofta ha en enkel kretsmässig lösning för en enda del av en större enhet, där denna del endast används till ett enda slag av beräkning och där man inte är intresserad av en mångsidig användbarhet.

Hos analogmaskiner kan man på ett enkelt sätt erhålla olika slag av förlopp genom integration. Där kan man t.ex. erhålla sinus och cosinus genom två stegs integration av differentialekvationen

$$\ddot{y} + k_1 y = k_2$$

I detta avsnitt diskuteras möjligheterna att på digital väg med en speciell digital integrator utnyttja en del av de metoder, som användes vid analogmaskiner. Med dessa möjligheter kan man kombinera analogmaskinens enkelhet med digitalmaskinens stabilitet och noggrannhet.

Många av de beräkningsmetoder, som används i detta avsnitt, påminner rätt mycket om dem, som förekommer hos digitala differentialanalysatorer.

#### 3.3.4.2. Binär multiplikator (modulator)

Om man studerar en konventionell binärräknarkedja, som räknar framåt, finner man att resp. bits "1"-sida går positiv enligt ett visst mönster med halva räknefrekvensen på första räknarsteget, fjärdedelen på andra osv. (se fig 46.a och b). Nederst i fig 46 b visas "1"-sidornas positiva flanker för resp. flip-flop. Här kan man observera, att inte någon av dessa pulser kommer i samma tidsögonblick som någon annan 1-side-puls. Man kan säga att utsignalen från 1-sidan på steg 1 har den högsta signifikansen i frekvenshänseende av alla 1-sidorna.

Med ett grindsystem enl. fig 46 c kan man införa villkor  $a_i$  (kan vara antingen 1 eller 0) på resp. grind och erhålla en summa av pulser vid utgången, vars antal per tidsenhet T är:



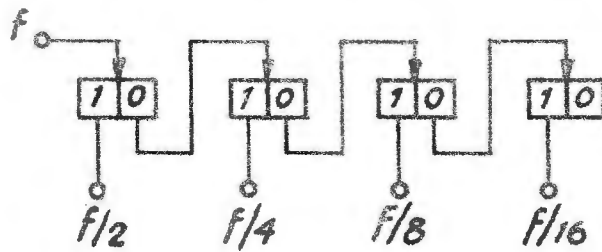


FIG. 46 a

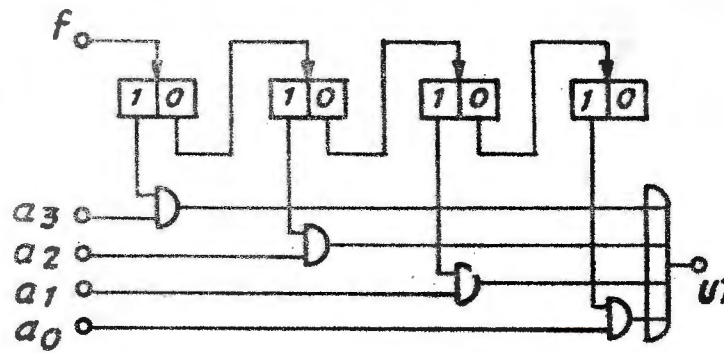


FIG. 46 c

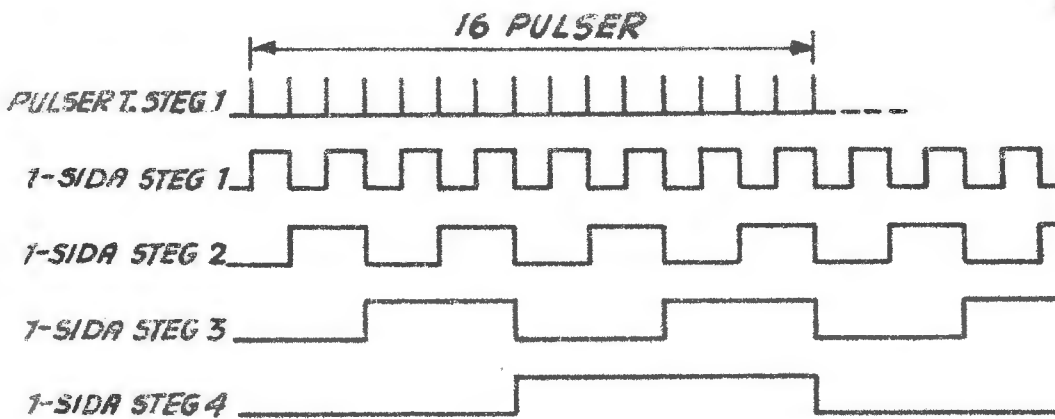
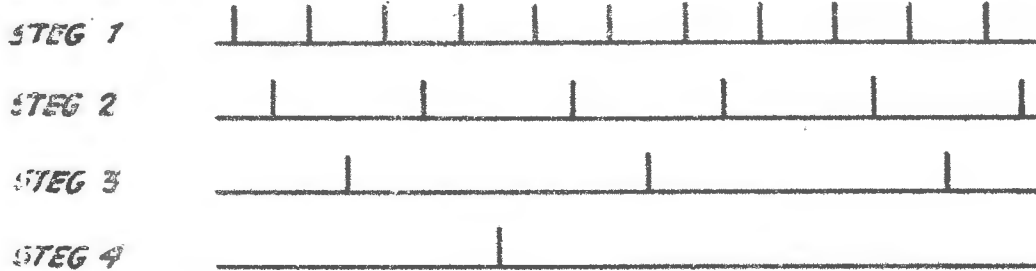
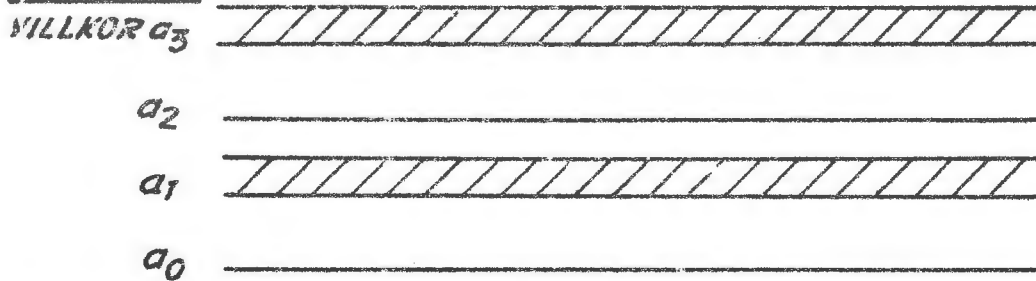


FIG. 46 b

1-SIDORS FLANKER:



EXEMPEL 1:

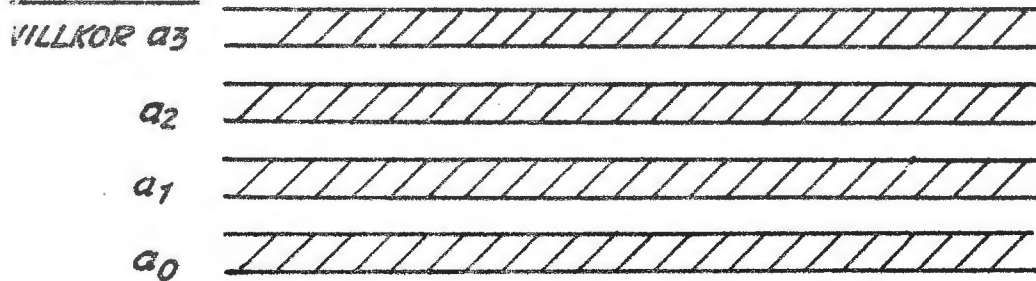


TALET  $10_{10} = 1010_2$



FIG. 46 d

EXEMPEL 2:



TALET  $15_{10} = 1111_2$



PULSUPPEHÅLL

$$fT \sum_{i=0}^{n-1} a_i 2^{-n+1} = fT \cdot 2^{-n} \sum_{i=0}^{n-1} a_i 2^i$$

där  $f$  är den frekvens den drivande räknaren erhåller. Antalet pulser på utgången är alltså proportionellt mot det binära talet

$$\sum_{i=0}^{n-1} a_i 2^i$$

Man bör observera att den mest signifikanta siffran i detta tal skall styra grinden från den i frekvenshänseende mest signifikanta flip-flopen. Man har således erhållit en metod att multiplicera eller modulera frekvensen  $f$  med ett binärt tal

$$k = \sum_{i=0}^{n-1} a_i 2^i$$

så att man får en lägre frekvens

$$f_{ut} = f \cdot 2^{-n} \sum_{i=0}^{n-1} a_i 2^i = f \cdot k / 2^n$$

Man kan alltså tala om en frekvensmodulering, och därför kommer i fortsättningen beteckningen multiplikator och modulator omväxlande att användas.

Pulserna i den erhållna frekvensen är ej jämnt fördelade i tiden, utan vissa förskjutningar av de enskilda pulserna äger rum, vilka ger upphov till momentana fel i antalet inräknade pulser.

Studera antalet pulser som erhålles under en cykel för räknaren, som i det följande kallas den drivande räknaren. Om denna räknare har  $n$  steg kräver den  $2^n$  pulser för att gå runt ett varv. Väljes tidsenheten  $T$  så att just så många pulser anländer till drivräknaren, dvs. så att  $f \cdot T = 2^n$ , erhålles efter modulering

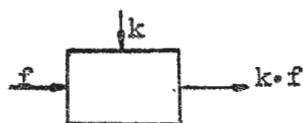
$$2^n 2^{-n} \sum_{i=0}^{n-1} a_i 2^i = \sum_{i=0}^{n-1} a_i 2^i$$

Det största tal, med vilket modulering kan ske, är det där alla  $a_i$  är 1. Då erhålles på utgången det maximala antalet pulser per cykel

$$\sum_{i=0}^{n-1} 2^i = \frac{2^n - 1}{2 - 1} = 2^n - 1$$

alltså en puls mindre än antalet pulser till den drivande räknaren. Detta uppehåll inträffar, när alla räknarstegen går från "1" - till "0"-läge samtidigt. Pulsbidraget från närmast högre steg, om systemet utökas med en bit, kommer att falla i varannan av dessa pulsluckor.

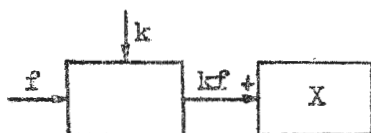
Den binära modulatorens representeras i fortsättningen med denna blocksymbol.  $f$  betecknar den drivande frekvensen och  $k$  det modulerande talet.



Moduleringen belyses av exempel som visas i fig 46 d.

#### Ren integration av en variabel.

Från den binära modulatorens erhålles ett pulståg, som har en lägre frekvens än den drivande. Om detta pulståg inräknas i en räknare kan man uppfatta räknarens värde som tidsintegralen av den modulerande storheten.



Under tidsintervallet  $t$  anländer till räknaren  $k \cdot f \cdot \Delta t$  pulser. Alltså kan tillskottet i  $X$  tecknas

$$\Delta X = kf \Delta t$$

$$X = \sum X = f \sum k \Delta t$$

Låter man nu  $\Delta t \rightarrow 0$  erhålles i det ideella fallet

$$\frac{dx}{dt} = kf$$

eller om man integrerar ett steg

$$X = \int kf \cdot dt = f \int k \cdot dt$$

De fall, då den modulerande storheten är konstant, kallas vanligen linjär modulation.

#### 3.3.4.3 Direkt användning av den binära integratoren för funktionsgenerering

##### Framställning av högregradsfunktioner

Om derivatan av en funktion införes som det modulerande talet till den binära integratoren, kommer räknaren att anta ett tidsberoende, som är detsamma som funktionens. Om ett konstant tal är modulerande kommer räknaren att stiga linjärt. Är det modulerande talet linjärt växande med tiden kommer räknaren



ha ett kvadratisk tidsberoende osv.

Andragsförlopp kan alltså erhållas enl. fig 47a, där man använder den dri-

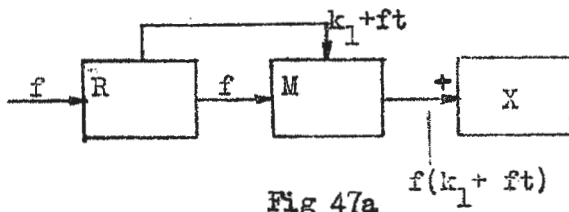


Fig 47a

vande räknaren hos modulatorens som modulerande storhet, eftersom denna räknarens värde stiger linjärt med tiden.

$$\Delta x = f(k_1 + ft) \Delta t$$

låt  $\Delta x \rightarrow 0$

$$\frac{dx}{dt} = fk_1 + f^2 t ; \quad x = \frac{f^2}{2} \cdot t^2 + fk_1 \cdot t + k_2$$

där  $k_1$  resp.  $k_2$  är begynnelsevärdena hos driv- resp. integrationsräknaren.

Högregradsförlopp kan erhållas genom att integrera från 0:te graden steg för steg upp till önskad grad enl. fig 47b.

3:e graden

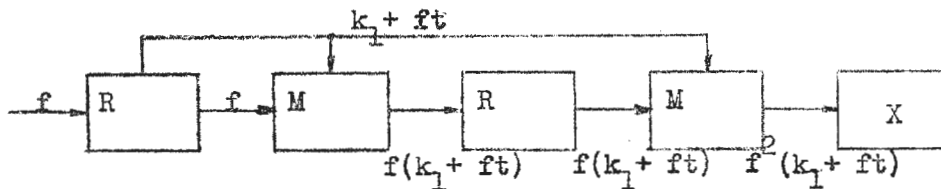


Fig 47b

Framställning av exponentialförlopp.

Kännetecknande för en exponentialkurva är, att derivatan i en godtycklig punkt är lika med kurvans värde i punkten så när som på några konstanter. Genom att driva modulatorens med funktionens värde skulle man erhålla en exponentiell förändring.

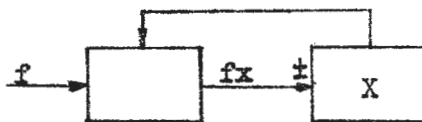


Fig 48.

Med den speciella binära integratorn kan denna funktion realiseras på det sätt, som visas i fig 48. Den integrerande räknarens värde återföres som villkor till den binära modulatorens.

Med den speciella binära integratorn kan denna funktion realiseras på det sätt, som visas i fig 48. Den integrerande räknarens värde återföres som villkor till den binära modulatorens.

$$\frac{dx}{dt} = \pm fx; \quad \frac{dx}{x} = \pm f dt$$

$$\ln x = \pm ft + C$$

$$\pm ft + C$$

$$x = e$$

då  $t = 0$ , har  $x$  värdet  $k_1$ .  $\therefore x = k_1 e^{\pm ft}$

Således kan både stigande och fallande exponentialförlopp erhållas. Observeras bör, att begynnelsevärdet kommer in som en konstant faktor, dvs. det måste vara skilt från noll, när räkning sker framåt i den uppsamlande räknaren. I annat fall kommer det dröja oändlig tid, innan beräkningen är slutförd. Vid räkning bakåt kan begynnelsevärdet väljas godtyckligt mellan 0 och 1 och räknarens värde kommer i detta fall att asymptotiskt närma sig 0.

3.3.4.4. Indirekt användning av integratorn för utförande av vissa matematiska operationer

Frekvenskännande koppling

Genom ett konstgrepp kan man få integratorn att verka "deriverande". Man använder en reversibel räknare, vars positiva ingång drives med en frekvens  $k \cdot f$ , medan dess negativa ingång är via en modulator förönden med utgången på räknaren. Räknaren är alltså återkopplad (fig 49).

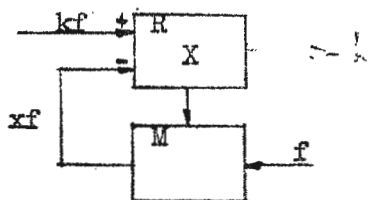


Fig 49.

$$\frac{dx}{dt} = kf - fx; \quad \frac{dx}{dt} + fx = kf$$

$$x = k(1 - e^{-ft}),$$

då begynnelsevärdet för  $x$  är 0.

Efter ett insvängningsförlopp kommer alltså räknaren att visa  $k$ , den bråkdel av grundfrekvensen som infrekvensen utgör. Varierar man infrekvensen, kommer räknaren att följa efter med viss eftersläpning. Tar man bort den positiva ingången, kommer kopplingen att vara exakt den, som gav ett negativt exponentiellt förlopp. "Deriveringen" är inte riktig i matematisk mening. Man kommer endast att kunna direkt avläsa en viss frekvens' förhållande till en given frekvens. Ändringar kommer omedelbart att kunna indikeras. Om man kopplar ett pulståg till en räknare och samtidigt till ovanstående anordning, kan man säga att räknaren kommer att visa integralen över pulståget och ovanstående anordning räknarens derivata.

Anordningen kan alltså användas för att mäta en frekvens relativt en annan frekvens, t.ex. en frekvens från en annan modulator. I fig 50 visas insväng-

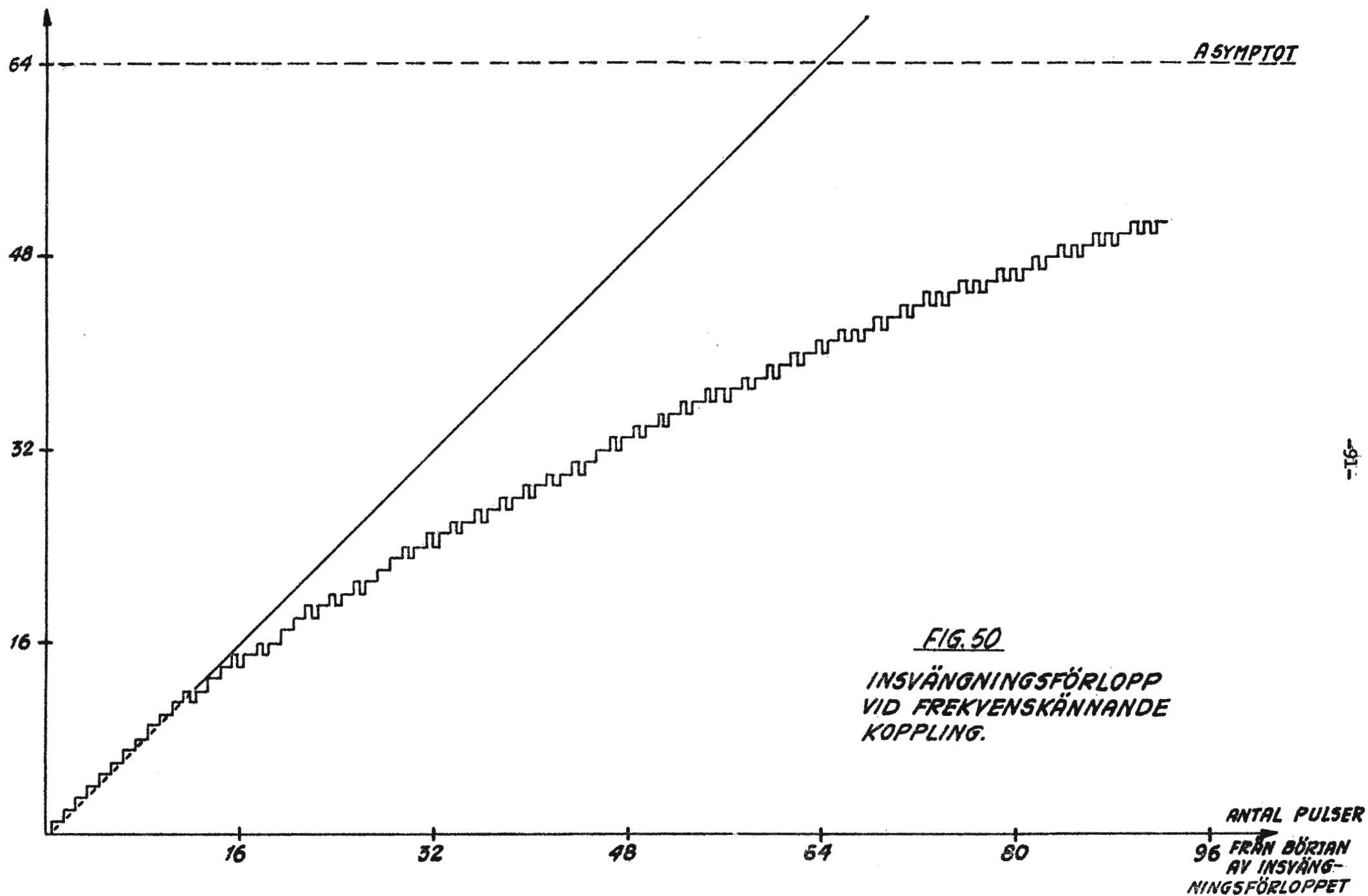


FIG. 50  
 INSVÄNGNINGSFÖRLOPP  
 VID FREKVENSKÄNNANDE  
 KOPPLING.

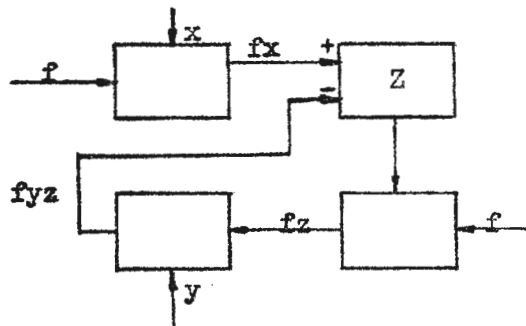
ANTAL PULSER  
 96 FRÅN BÖRJAN  
 AV INSVÄNG-  
 NINGSFÖRLOPPET



ningsförloppet för kopplingen, då begynnelsevärdet för den reversibla räknaren är noll och den mätta frekvensen är densamma som referensfrekvensen, d.v.s.  $k = 1$ . Förloppet är utfört vid 6 bitar och räknarens värde skall alltså svänga in mot ett 64 minsta enheter högre värde. Tidsenheten indelas i två moment, ett då räkning sker framåt, och ett då räkning sker bakåt.

Division

Med den i föregående avsnitt angivna anordningen kan man utföra division. Den reversibla räknarens positiva ingång anslutes till en modulator, till vars grindar dividenden föres. Den negativa ingången förenas via två modulatorer med räknarens utgång (fig 51).



$$\frac{dz}{dt} = f \cdot x - fyz$$

$$\frac{dz}{dt} + fyz = fx$$

$$Z = \frac{x}{y} (1 - e^{-fyt}) \text{ då utgångsläget är noll.}$$

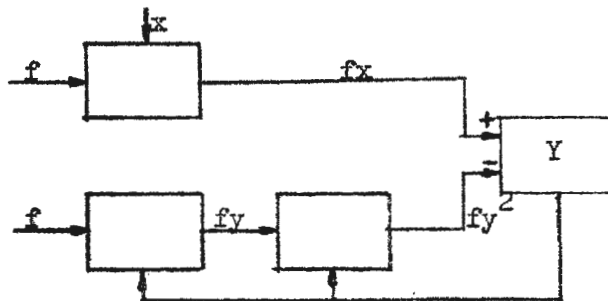
Fig 51.

Räknaren kommer alltså att efter en insvängning visa kvoten mellan  $x$  och  $y$ .

För att anordningen skall kunna fungera måste  $0 < x < y < 1$ , ty den reversibla räknaren kan ej visa större värde än  $1 - 2^{-n}$ , där  $n$  betecknar antal bitar. Modulatorerna kan inte heller moduleras med större tal.

Rotutdragning

I de två föregående kopplingarna för frekvensmätning och division har principen varit, att lösningen erhållits då den positiva och den negativa ingången balanserar varandra. Detta kan utnyttjas för att beräkna andra funktioner. Låter man t.ex. en variabel  $x$  balanseras av en annan variabels kvadrat kan man ordna rotutdragning (fig 52).



$$\frac{dy}{dt} = fx - fy^2$$

$$\frac{dy}{dt} + fy^2 = fx$$

$$y_p = \sqrt{x}$$

Fig 52.

$$\frac{dy}{y^2} = -f dt; \quad -\frac{1}{y} = -ft + c; \quad y_{\text{hom}} = \frac{1}{ft + c_1}$$

$$y = \sqrt{x} + \frac{1}{ft + c_1}$$

3.3.4.5. Generering av sinus-cosinus-funktioner

Differentialekvationen  $\frac{d^2x}{dt^2} + \omega^2x=0$  har den homogena lösningen  $x = A\sin\omega t$  eller  $x = B\cos\omega t$  beroende på begynnelsevillkoren. Med en analogi-maskin skulle problemet kunna lösas om  $\frac{d^2x}{dt^2}$  integreras två gånger och sedan jämförs med  $\omega^2x$ . Detta kan även göras med den binära modulatorens kopplad som integrator (se fig 53).

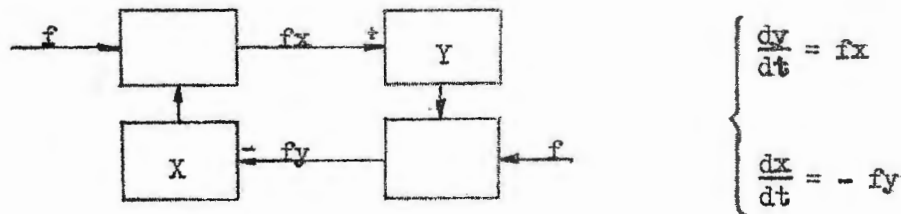


Fig 53.

$$\frac{d^2y}{dt^2} = f \frac{dx}{dt} = -f^2y; \quad \frac{d^2y}{dt^2} + f^2y = 0; \quad y = Ae^{jft} + Be^{-jft}$$

Antag att  $y = 0, x = 1$  då  $t = 0$

$$x = \frac{1}{f} \frac{dy}{dt} = j (Ae^{jft} - Be^{-jft})$$

Begynnelsevillkoren ger  $\begin{cases} A + B = 0 \\ j(A - B) = 1; \quad A = -\frac{j}{2}; \quad B = \frac{j}{2} \end{cases}$

$$x = \frac{e^{jft} + e^{-jft}}{2} = \cos ft$$

$$y = \frac{e^{jft} - e^{-jft}}{2j} = \sin ft$$

Man kan även, som differentialekvationerna uttrycker saken, teckna de två integrationerna på så sätt att "sinus" har derivatan "cosinus" och "cosinus" derivatan "minus sinus".

Speciellt för den binära integratorn kan det uppstå en del tolkningsproblem. Antag att systemet har  $n$  bitar. Funktionsvärdet 0 representeras i räknarna av  $n$  st 0-or medan Funktionsvärdet 1 representeras av  $n$  st 1-or. Detta bety-

der att räknarna aldrig skall kunna visa funktionsvärdet 1, utan det största tal de kan visa är  $1 - 2^{-n}$ . Det man läser i räknarna är alltså binärbråk, där binärpunkten står omedelbart till vänster om högsta binära siffran.

Om räknarens storlek är  $n$  bitar kommer det antal pulser som fordras för att stega fram från funktionsvärdet 0 till 1 vara  $2^n$  (i verkligheten  $2^n - 1$ ... då slutvärdet är  $n$  st 1-or). För att modulatorens skall lämna detta antal pulser, måste modulatorräknarna mottaga  $2^n \cdot \frac{\pi}{2}$  pulser per  $90^\circ$ . Detta tal är emellertid inte en jämn 2-potens, vilket i många fall är önskvärt. Man kan då använda en räknare med  $n + 1$  bitar och till den koppla en modulator med konstant modulering med  $\pi/4$ . Detta ger totala antalet pulser till sinus-cosinus-modulatorräknaren  $2^{n+1} \cdot \frac{\pi}{4} = 2^n \cdot \frac{\pi}{2}$ .

Fig. 54 visar sin- och cos-kurvor erhållna med 5 bitars modulatorer

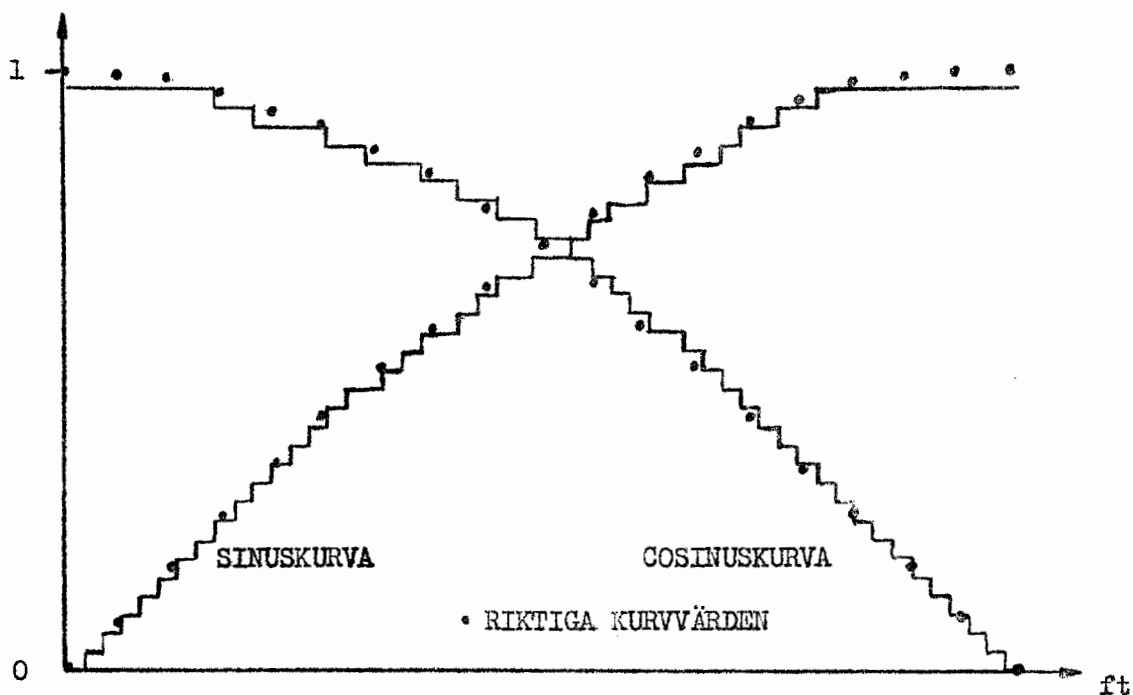


Fig. 54.



### 3.3.4.6. Metod att förbättra noggrannheten hos en binär modulator genom att använda avrundande steg

Hos den ordinära binära modulatern kan det uppstå avsevärda fel i värdet av den uppsamlade räknaren, eftersom de pulser som erhålles från modulatern endast kan komma vid vissa bestämda tider. Man får alltså fel som är större än rena avrundningsfel. Vill man emellertid ha bättre noggrannhet vid givet antal bitar får man tillgripa någon metod för att minska felen.

För en linjär n-stegsmodulator kan maximalfelet uttryckt i antal delningar anges av följande formel:

$$|\Delta n| = 7/18 + n/6 + 1/9(-1)^n \cdot 2^{-n}$$

Av denna formel framgår att absolutfelet kan göras godtyckligt litet om tillräckligt många steg används.

Man kan alltså driva modulatern med ett större antal bitar än önskat och med någon anordning avrunda till önskat antal. Om man helt enkelt låter bli att avläsa de lägsta stegen vid en sådan modulation är det inte säkert att man erhåller mindre fel än om man arbetar med önskat antal bitar direkt.

Genom att modifiera denna avläsningsmetod något kan man förbättra noggrannheten. Vid korrekt avrundning skall den uppsamlade räknaren slå om till nästa värde, då felet (dvs. skillnaden mellan räknarens värde och den önskade kontinuerliga kurvan) just har nått värdet 1/2 minsta avläst enhet. Man kan därför snedställa den uppsamlade räknaren, så att den har värdet 1/2 minsta enhet när beräkningen börjar (räkning framåt antages). Räknaren kommer då att ta emot pulser som är värda  $2^{-m}$ , där m anger antalet icke avlästa steg i den lägre delen av den uppsamlade räknaren. När nu räknaren mottagit pulser, som sammanlagt är värda en halv minsta enhet kommer alltså räknaren stå i läget 1 enhet. Räknaren kommer alltså att presentera värden som är korrekt avrundade.

Denna metod kan användas vid såväl linjär som olinjär modulation. Vid sinus-cosinus-generering har metoden undersökts och visat sig ge en avsevärd förbättring såväl vid ett ringa antal bitar som vid 11 bitar. Detta på bekostnad av snabbheten. För att kunna driva systemet fordras  $2^m$  ggr fler pulser än vid modulering utan avrundande steg.

### 3.3.5. Kontinuerlig datapresentation

#### 3.3.5.1. Allmänt

I de flesta fall av databehandling finns ett behov av kommunikation från data-

maskin till mänskliga operatörer eller bedömare. I generell databehandling sker denna förmedling oftast via klartext från skrivmaskiner anslutna till datamaskinerna. Denna kommunikation har mer karaktären off-line. I reelltidstillämpningar exempelvis av typen 2 i översikten i avsnitt 3.1. förekommer behov av direkt och kontinuerlig presentation av processtillstånd för operatörer, så att dessa med minsta tidsfördröjning kan erhålla upplysningar och handla efter dessa.

Här skall endast beröras ett par grundtyper av presentation, som arbetar på digital basis och är av den karaktären att de är direkt anpassbara till exempelvis en datamaskin. Utrustningar innehållande lampor, analoga karaktärbildare (exempelvis karaktärrör) eller mekaniska anordningar berörs däremot inte.

Den mängd data som skall presenteras i ett reelltidssystem kan vara ganska stor, varför utmatning ur ekonomisk synpunkt bör ske i tidmultiplex genom så få presentationsgeneratorer som möjligt. Detta i sin tur kräver mycket snabb utmatning och presentation för en enskild information. Presentation av data på katodstrålerör (s.k. tabellrör) är därvid en praktisk lösning p.g.a. skrivorganets snabbhet. I luftövervakningssystem kan viss del information gällande flygplan också skrivas direkt på en operatörs PPI-skärm tillsammans med den normala radarbilden. Viss upprepningsfrekvens krävs vid denna typ av presentation och denna beror på rörets efterlysningsförmåga samt på önskad flimmerfrekvens.

Man kan i princip skilja på två grundtyper av datapresentation på katodstrålerör. Den ena är direkt skrivning med hjälp av successiva punkter i en symbol och den andra är den s.k. rastermetoden, där elektronstrålen sveper i ett raster över en viss yta av skärmen och lysas upp i de punkter där den skär en symbollinje.

### 3.3.5.2. Direktskrivningsmetoden

Direktskrivningen i den form den här skall beskrivas och som lämpar sig väl för kommunikation med en datamaskin grundar sig på en matris av tillgängliga punkter ur vilken de för varje figur behövliga punkterna väljs. Fig 55 visar en matris på 5 x 7 punkter i vilken punkterna för siffran 5 markerats. Inom en sådan matris kan varje punkt anges av ett visst par  $\bar{x}$ - och  $\bar{y}$ -spänningar för utstyrning av den skrivande elektronstrålen. Varje sådant spänningspar kan i digital form anges av två koder, som efter omvandling till analogform ger det önskade spänningsparet. I en 5 x 7-matris, som kodas i binär form, erfordras tydligen 3 bitar för vardera  $\bar{x}$ - och  $\bar{y}$ -spänningarna. Siffran i fig 55 kan då skrivas, om följande kodpar successivt omvandlas till analogspänningar (början i  $x = 1, y = 2$ , se tabell 5).

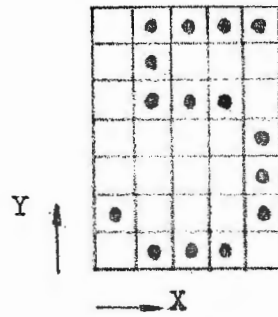


Fig 55.

X	Y
001	010
010	001
011	001
100	001
101	010
101	011
101	100
100	101
011	101
010	101
010	110
010	111
011	111
100	111
101	111

Tabell 5

De successiva kodparen kan givetvis matas direkt från en datamaskin, men detta skulle belasta dess kapacitet i enorm grad vid stor presentationsvolym och skrivningen skulle ändå inte kunna göras med för katodstråleröret maximal snabbhet, såvitt inte de allra snabbaste datamaskintyperna skulle användas. En annan metod är att låta en speciell utrustning, en s.k. symbolgenerator, sköta frammatningen av koderna, medan symbolvalet görs från datamaskinen symbolvis eller om grupper av symboler. Koderna för varje symbol kopplas då upp fast i en matris i symbolgeneratorm. Vid skrivning av viss symbol avses dennas matris successivt som i tabell 5. Fig 56 visar, hur datamaskinen överlämnar en symbolkod till generatorm. Denna avkodas och väljer ut en viss symbolmatris, så att punktavsökarpulserna kopplas till denna. De successiva X- och Y-koderna matas till var sin digital-analogomvandlare, som ger de önskade avlänkningsspänningarna till presentationsröret.

Placeringen av varje symbol på katodstrålerörets skärm kan exempelvis beordras av datamaskinen, samtidigt som den beordrar symboltyp. Utstyrningen av elektronstrålen till detta läge görs då i separat digital-analogomvandlare.

### 3.3.5.3. Rastermetoden

Rastermetoden bygger även den på en matris av punkter som i fig 55. Fig 57



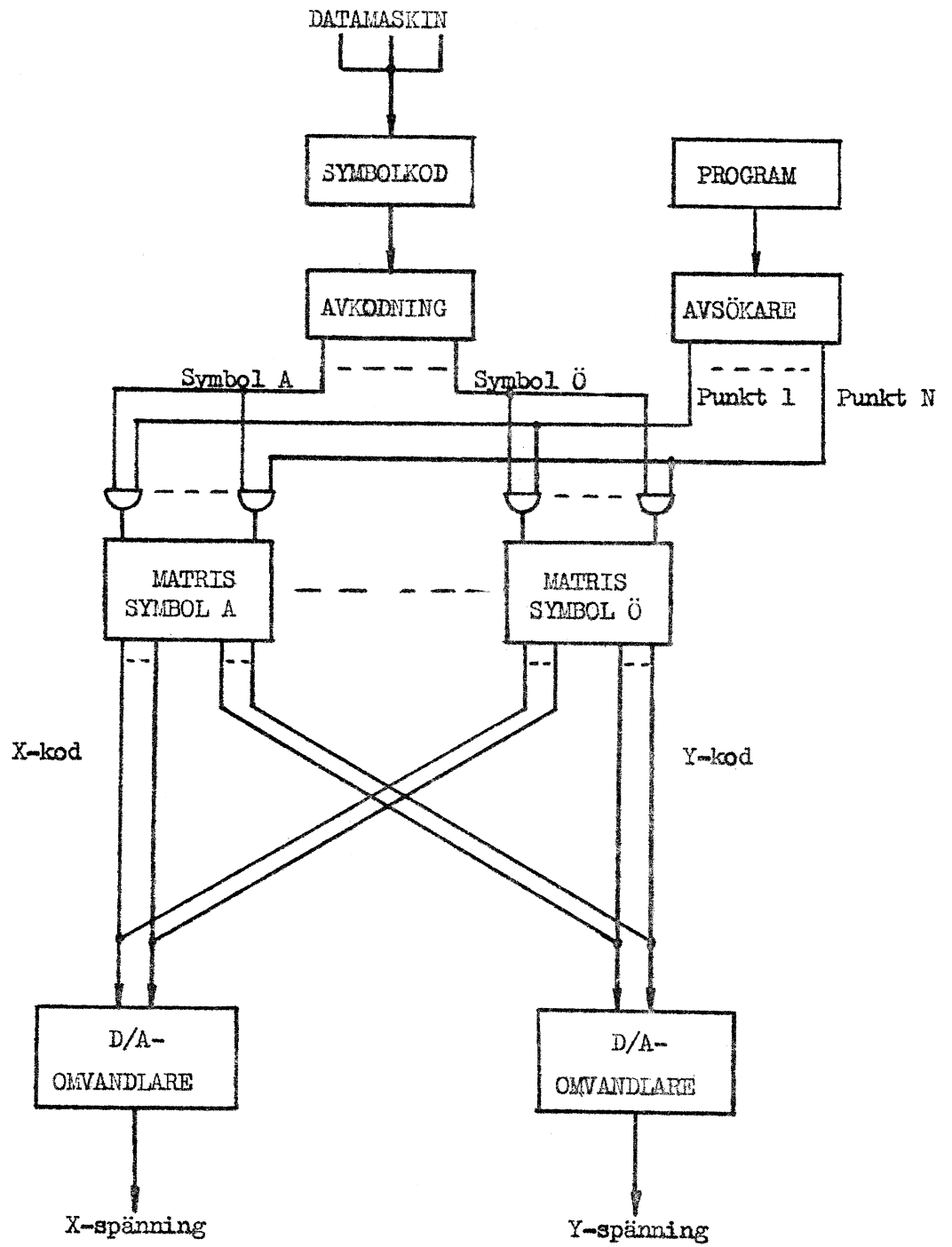


Fig 56

visar, hur en sådan matris byggts upp av magnetkärnor och hur en metalltråd dragits genom de kärnor som bildar siffran 5.

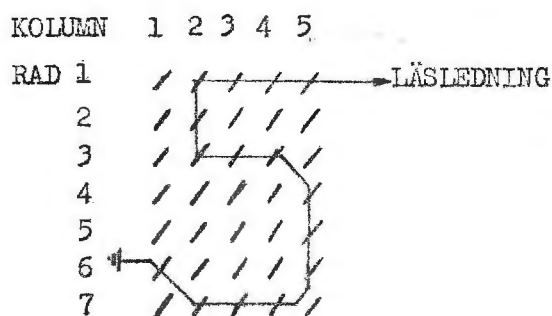


Fig 57.

Tillvägagångssättet vid rasterskrivningen är nu följande: kärnorna, som antas fyrkantmagnetiserade, befinner sig alla i ett magnetiskt tillstånd, som betecknas 0-tillstånd. En drivledning genom alla kärnor i rad 1 (ledningen ej utritad i fig) pulsas med en ström, som sätter alla dessa kärnor i 1-tillstånd. Därefter pulsas en likadan ledning genom alla kärnor i kolumn 1, som strävar att ge kärnorna 0-tillstånd. Kärnan i första raden och första kolumnen switchar då tillbaka till 0-tillstånd. Därefter pulsas kolumn 2, kolumn 3 etc. Alla de kärnor i rad 1, som genomlöps av den i fig 57 utritade "lästråden", kommer då att ge en puls på denna, då kärnorna sätts tillbaka i 0-tillstånd av kolumndrivningen. Då alla kolumner pulsats, sätts rad 2 i 1-tillstånd, varefter alla demnas kärnor avläses genom pulsning av kolumner. På detta sätt förfäres genom hela matrisen, som alltså kommer att läsas av som med ett TV-svep. Det program som bildar denna avsökning kan också bilda analoga x- och y-svep i takt med avsökningen. Dessa används att avlänka elektronstrålen i tabellröret i takt med avsökningen, och varje gång en puls erhålles på läsledningen, lysas strålen upp så att den önskade figuren erhålles ur lysande punkter.

En kärnmatris av ovan beskrivet slag kan förses med en mängd lästrådar för olika figurer i samma matris. Utval av symbol görs då genom grindning av resp. lästråds utsignaler. Därigenom kan flera olika symboler skrivas samtidigt på olika ställen. Symbolval från datamaskin sker i princip på samma sätt som i fig 56 med symbolkod och avkodning. Skillnaden är som sagt, att avkodningen i detta fall grindar en lästrådsignal.

Denna typ av symbolgenerator är givetvis mycket kompakt i förhållande till den förstnämnda typen men är i gengäld långsammare, eftersom den i exempelvis fallet med en 5 x 7-matris alltid måste avsöka 35 punkter medan den andra typen avsöker c:a 15 punkter.

3.3.5.4. Generering av godtyckliga kurvor

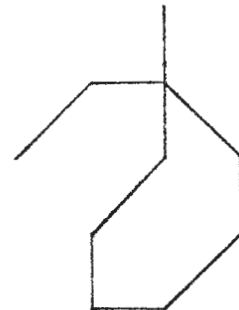
I en del tillämpningar har man behov av en kontinuerlig presentation av mer invecklade kurvor såsom kartkonturer på PPI vid luftövervakning, transienter i simulatorer etc, som inte kan utföras med de generatorer som beskrivits i de två föregående avsnitten p.g.a. kurvornas utsträckning.

Emellertid kan punktskrivningsmetoden med fördel användas även i detta fall i utvidgad form. Man kan således tänka sig, att alla punkter i exempelvis en karta är lagrade i ett minne som x- och y-koder och att man vid presentation av söker detta minne successivt. För en kurva innehållande tusentals punkter blir emellertid det erforderliga minnesutrymmet i så fall ganska stort, eftersom man ofta kräver minst 10 bitars upplösning i vardera koordinaten. Dessutom måste minnet uppsökas mycket ofta, om snabb skrivning av kurvan önskas. Ett eventuellt önskemål att lägga in kurvdata i en maskins normala minne kan därför vara svårt att uppfylla.

En metod som sänker behovet av minnesutrymme är skrivning med hjälp av inkrement, vilken kan användas för alla kontinuerliga kurvor. Enligt denna metod utgörs de från minnet utmatade koderna av uppgifter om skrivstrålens förflyttningar. Om strålen för varje gång endast rör sig en enda delning i x- och y-koordinaterna, är det tydligt att endast 8 nya lägen för varje gång kan komma ifråga (se fig 58 a). Varje förflyttning kan således beskrivas med en kod

7	0	1
6	●	2
5	4	3

a/



Kodföljd:  
1234560100

b/

Fig 58.

om endast 3 bitar. Fig 58 b. visar, hur en godtycklig figur kan skrivas med hjälp av en följd sådana koder. Fig 59 anger principen för hur den skrivande generatoren kan utformas. Ett register för vardera koordinaten används som räknare och integrerar inkrementen, som bildar strålens väg. Vid start av varje kurva sätts dessa register med startvärden från minnet. Förflyttningskoderna avkodas så att de för vart och ett av de 8 nya lägena kan tilldela räknarna de rätta inkrementen  $\Delta x$  och  $\Delta y$ .

P.g.a. att koderna i detta fall är så korta, kan flera tänkas lagrade parallellt i en minnesadress. Utmatning av flera  $\Delta$ -värden kan då göras samtidigt till  $\Delta$ -registret i generatoren, som själv sedan tar hand om dem i tur och ord-



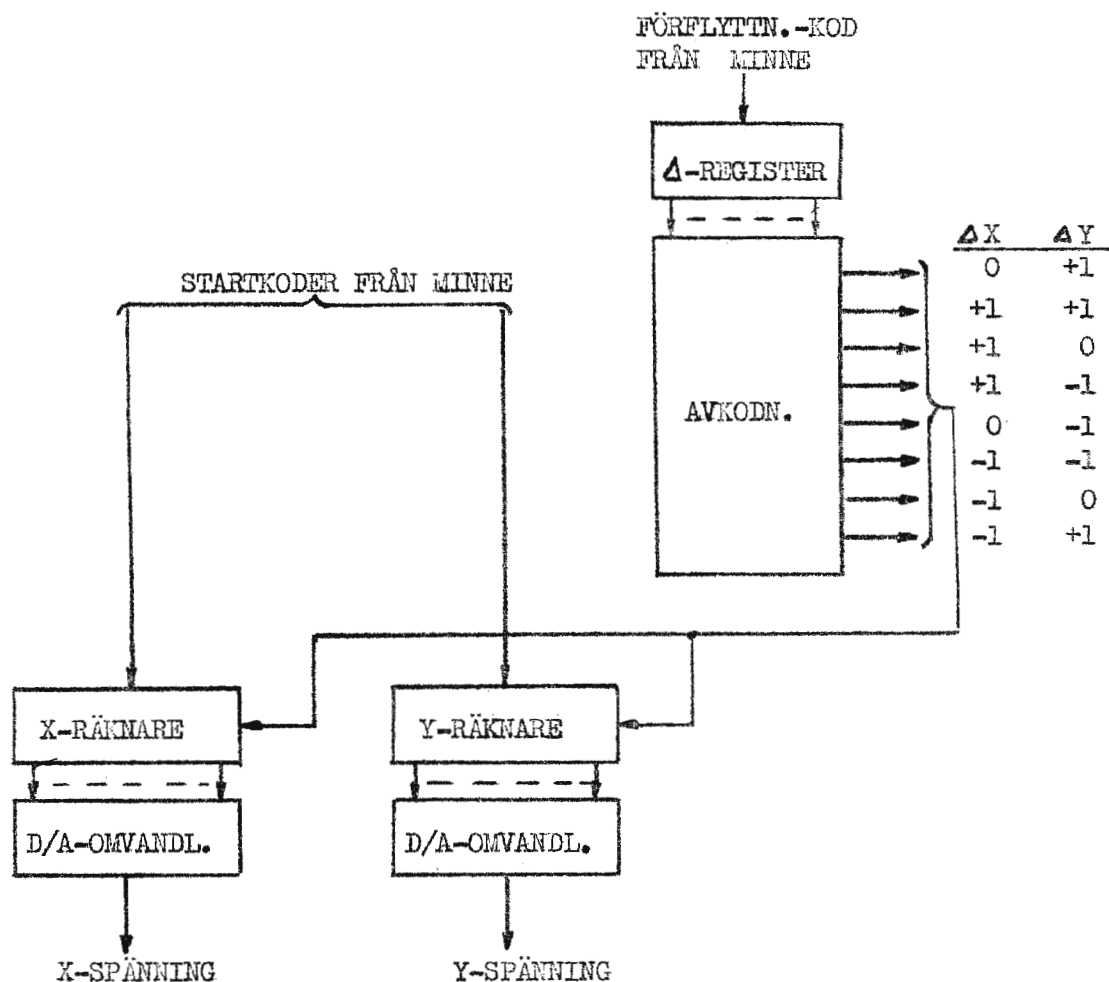


Fig 59.

ning. Därigenom minskas belastningen på minnet.

Ofta önskar man hålla avståndet mellan två successiva punkter större än en delning i x och y, för att mängden punkter skall hållas nere. De 8 nya lägen, som anges i fig 58, kan då vara för få för att ge tillräcklig noggrannhet i kurvritningen. Antalet nya lägen kan emellertid fördubblas. för varje gång en ny bit tillfogas förflyttningsskoden. Avkodaren i fig 59 blir då något mer komplicerad men inte så mycket, att det har någon större betydelse.

#### 4. DRIFTSÄKERHET

##### 4.1. Felfrekvens och medeltid mellan fel

Digitala utrustningar kräver p.g.a. sin karaktär ett relativt stort uppått av komponenter jämfört med analogsystem. Digitalteknikens framsteg har därför varit intimt förbundna med tillförlitligheten hos de komponenter, som stått till buds. Behovet av allt större och mer komplicerade system driver fram bättre och bättre komponenter, och i och med halvledarteknikens utveckling togs ett stort steg mot den grad av tillförlitlighet som erfordras i praktiska större digitalsystem.

Hur goda komponenter man än framställer, kan man dock aldrig komma ifrån en liten sannolikhet för att en komponent skall uppvisa felaktiga värden antingen p.g.a. inneboende dolda fel, som så småningom uppträder och då oftast i form av mycket stor värdeförändring s.k. katastroffel eller också p.g.a. förslitning dvs förändring med tiden under normal drift (åldring). Den senare typen av fel söker man helt undvika dels genom sådan underbelastning av varje komponent, att dess åldring blir mycket liten, och dels genom så konservativ dimensionering av kretsarna, att dessa kan fungera även med de förändringar i komponentdata som kan komma ifråga. Den förra typen av fel däremot, som normalt uppträder statistiskt fördelade, är svårare att undvika och måste alltid räknas med, även om felsannolikheterna är mycket små. Man kan nämligen alltid tänka sig ett så stort system, att antalet komponenter och därmed antalet felkällor gör hela systemets tillförlitlighet otillåtet låg.

Praktiska erfarenheter från ett stort antal skilda tillämpningsområden har visat, att felfrekvensen för en komplicerad utrustning efter en viss för varje utrustning specifik tid (inkörningstiden) antar ett konstant värde. Komponentfelen uppträder härvid slumpvis och är av katastrofkaraktär.

Om felfrekvensen betecknas  $\lambda$ , gäller följande uttryck för sannolikheten R, att en utrustning skall fungera till tiden t:

$$R(t) = e^{-\lambda t} \dots \dots \dots (9)$$

Det inverterade värdet av  $\lambda$  anger medeltiden mellan fel och betecknas här m.

$$m = 1/\lambda \dots \dots \dots (10)$$

Antag, att utrustningen består av  $n_i$  komponenter av i:te slaget, där  $i = 1, 2, \dots, j$ . Felfrekvensen för varje sådan komponent betecknas  $\lambda_i$ , varvid gäller en total felfrekvens  $n_i \lambda_i$  för hela i:te gruppen. Hela utrustningens felfrekvens blir därvid:

$$\lambda = \sum_i^j n_i \lambda_i \dots \dots \dots (11)$$

Felfrekvensen  $\lambda$  anges oftast i antalet fel per  $10^6$ . Felstudier på ett antal större utrustningar har givit följande ungefärliga  $\lambda$ -värden för olika komponenttyper:

transistorer (kisel)	0.2
dioder (kisel)	0.04
kondensatorer(papper)	0.01
motstånd (kolskikt)	0.015

Man får räkna med, att de angivna värdena kommer att förbättras elteftersom produktionstekniken går framåt.

Antag som ett exempel ett system med följande komponentfördelning:

15 000 transistorer
45 000 dioder
30 000 kondensatorer
<u>60 000 motstånd</u>
150 000 komponenter

Med de ovan exemplifierade  $\lambda$  -värdena erhålles ur ekv (11):

$$\lambda = 0,005 \text{ fel/tim}$$

Ekv. (10) ger medeltiden mellan fel:

$$m = 200 \text{ tim}$$

Detta sätt att ange en utrustnings driftsäkerhet är både praktiskt och åskådligt. Sambandet mellan m och funktionssannolikheten R framgår av ekv (9) i omskriven form:

$$R(t) = e^{-t/m}$$

Ett system, som består av n st utrustningar med var sitt m-värde får följande funktionssannolikhet enligt ekv. (11):

$$R(t) = e^{-t \sum_{i=1}^n \lambda_i} = e^{-t \sum_{i=1}^n 1/m_i} = \prod_{i=1}^n R_i(t) \dots\dots\dots (12)$$

dvs funktionssannolikheten för hela systemet är lika med produkten av alla utrustningars funktionssannolikheter, vilket också direkt framgår av grundläggande sannolikhetskalkyl.

Om ett system består av exempelvis två utrustningar med m-värdena 200 resp. 100 timmar, erhålles ett totalt m-värde:

$$1/m = 1/200 + 1/100$$

$$m = \frac{200 \cdot 100}{300} = 67 \text{ timmar}$$

4.2. Införande av redundans. Majoritetsval

Av det föregående framgår, att medeltiden mellan fel kan bli så liten för större system att detta blir praktiskt oanvändbart p.g.a. ständigt underhåll.



I vissa fall exempelvis inom databehandling av off-line-karaktör kan man undvika, att felet ger störningar i maskinarbetet genom att periodiskt anslå viss tid till marginalprov och systematiskt utbyte av komponenter. Under dessa tider kan då maskinen inte användas för sitt vanliga arbete.

I många fall speciellt i reelltidssystem förekommer krav på kontinuerlig drift av datautrustningen. Man kan därvid i princip skilja mellan två driftstyper, dels den som är tillgänglig för reparation av uppträdande fel och dels den som inte är det. Den sista typen är givetvis den svåraste, men den förra kompliceras av kravet, att systemet måste fungera under reparation samt att så fort fel uppträder ge signal om detta som samtidigt lokaliserar felet.

#### Icke reparabla system

Det normala sättet att hjälpa upp tillförlitligheten hos en utrustning är att införa redundans dvs överflödiga komponenter, som kan träda in i stället för felande sådana. Man kan därvid skilja på två typer av redundans, dels komponentredundans där varje komponent har en eller flera ersättare och dels kretsredundans där en eller flera kretsar bildar enheter som har en eller flera likadana enheter som ersättare. Fig 60 visar olika typer av redundanta kopplingar, som dock endast är användbara för komponentredundans, eftersom fel i enstaka delar kan påverka andra delar.

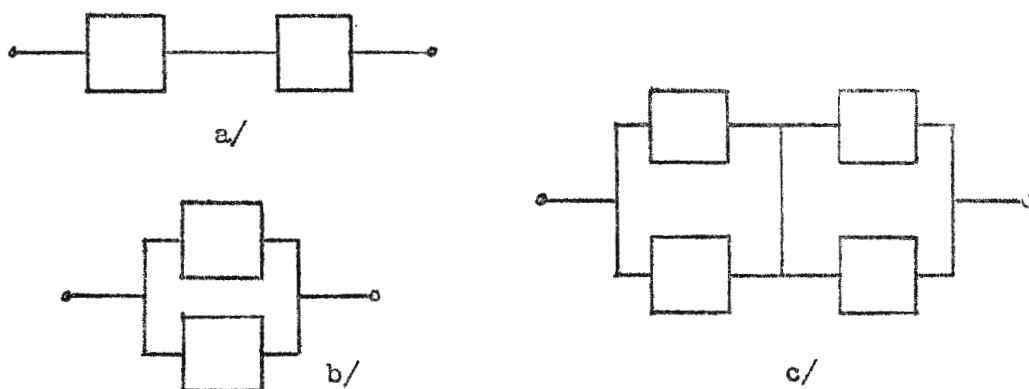


Fig 60.

Av dessa kopplingar är serieparallellkopplingen den mest universella, eftersom den kan tåla såväl avbrott som kortslutning i varje gren. Det är dock tydligt att dessa typer av redundans förutom att de ökar antalet komponenter även kan ge problem vid kretskonstruktion. Antag exempelvis, att serieparallellkopplingen består av kapacitanser  $C$ . Inget fel ger totalresistansen  $C$ , medan olika felmöjligheter som dock inte ger totalfel kan få kapacitansen att variera mellan  $C/2$  och  $2C$ . Sådana variationer måste alltså kretsen tåla, vilket kan vara mycket svårt att åstadkomma. Det är heller inte lätt att

genom test exempelvis vid inkörning erhålla upplysning om kretstillståndet, eftersom vissa felaktiga komponenter inte ger sig till känna i funktionen. Om kopplingen exempelvis i fig 60.b. skulle användas för hela kretsar eller systemblock, måste de olika blockens in- och utgångar isoleras från varandra, så att exempelvis en kortslutning på ett blocks utgång inte låser en annan utgång. Denna isolering är möjlig att utföra medelst dioder. Dock kvarstår ett annat problem, och det är att avgöra, vilket av blocken som går fel. Ett sätt att lösa detta är att använda felupptäckande koder i alla kommunikationer mellan blocken, men kretsarna för att bilda och kontrollera dessa koder blir normalt relativt komplexa och kan sänka driftsäkerhetsvinsten avsevärt, om de har en felsannolikhet i närheten av den som gäller för de kontrollerade blocken.

En enklare metod att realisera är införande av s.k. majoritetsval mellan de redundanta blocken. Fig 61 visar, hur  $m$  st funktionsblock parallellkopplats till en gemensam ingång samt hur utgångarna avkänns av en väljande switch. Under förutsättning, att switchen kan välja ett funktionsdugligt block, är sannolikheten för systemfel endast  $q^m$ , om  $q$  är felsannolikheten under viss tid för ett enskilt block. För  $n$  sådana redundanta kopplingar är totala felsannolikheten lika med  $1 - (1 - q^m)^n$  (funktionssannolikheten =  $(1 - q^m)^n$ ), vilken kan fås att gå mot noll vid tillräckligt stora  $m$  oberoende av värdet på  $n$ .

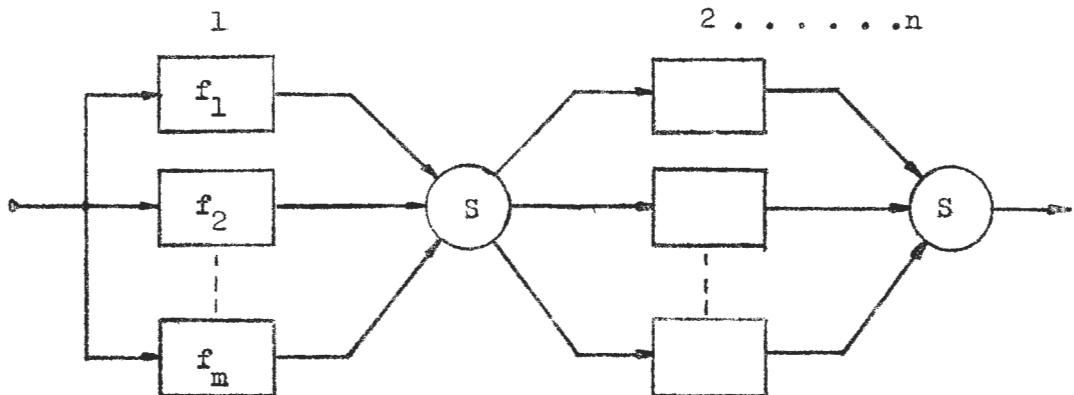


Fig 61.

Fig. 62 visar en annan redundant koppling, där switchen kontrollerar flera funktionsgrupper. I detta fall är felsannolikheten lika med  $[1 - (1 - q)^n]^m$ , som går mot värdet 1 för stora  $n$  oberoende av värdet på  $m$ . Av dessa två exempel framgår, att det är fördelaktigt att lägga valet efter så få funktioner dvs på så låg nivå som möjligt. Detta gäller dock med viss modifikation be-

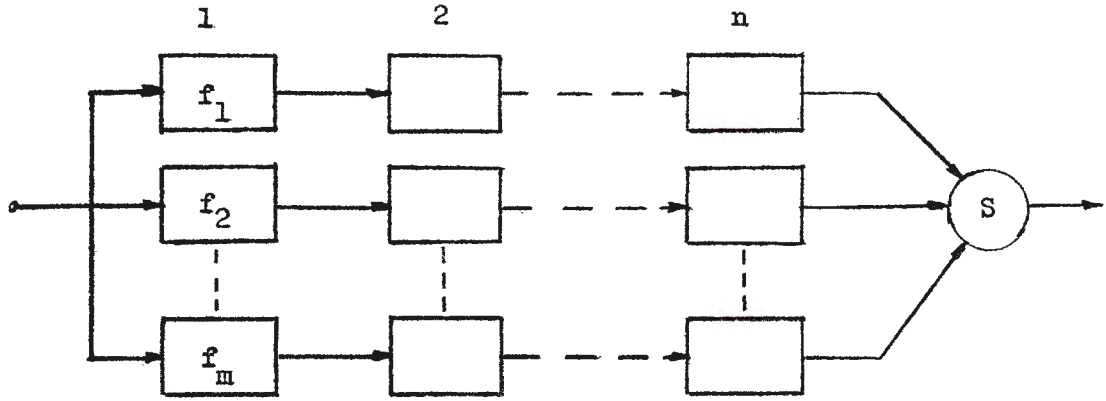


Fig 62

roende på hur tillförlitlig switchen själv är.

För att switchen skall kunna välja en korrekt utsignal måste viss extra information finnas. Ett sätt att lösa detta problem är att låta switchen jämföra alla signaler och välja den typ, som förekommer i största antal (majoritetsval). En vanlig redundanskonfiguration är därvid tre parallella funktionsblock anslutna till en majoritetslogik, som väljer sådana utsignaler som förekommer på minst två utgångar samtidigt. Om man först antar att majoritetslogiken är felfri, så är enda möjligheten för den redundanta gruppen att gå fel, att mer än ett funktionsblock felar. Funktionssannolikheten för en grupp är då:

$$R = p^3 + 3 (1-p)p^2 = p^2 (3-2p) \dots \dots \dots (13)$$

För ett system om n sådana grupper:

$$R_1 = [ p^2 (3-2p) ]^n$$

Tag som exempel, att n=100 och p=0.99. Då är  $R_1 = 0.97$ , medan i ett icke redundant fall  $R_0 = p^n = 0.37$ .

Den relativa vinsten i tillförlitlighet gentemot ett icke redundant system kan uttryckas:

$$R_1/R_0 = [ p^2 (3-2p) ]^n / p^n = [ p(3-2p) ]^n$$

Fig 63 visar sambandet mellan  $R_1/R_0$  och funktionssannolikheten p. Någon vinst kan tydligen inte erhållas, om  $p \leq 0.5$ . Majoritetsval på funktionsblock, som är så stora att tillförlitligheten är för låg, lönar sig alltså inte.

Om felmöjligheterna för majoritetslogiken tas med i beräkningarna, blir givetvis tillförlitligheten i systemet sämre. Antag en funktionssannolikhet  $p_m$  för



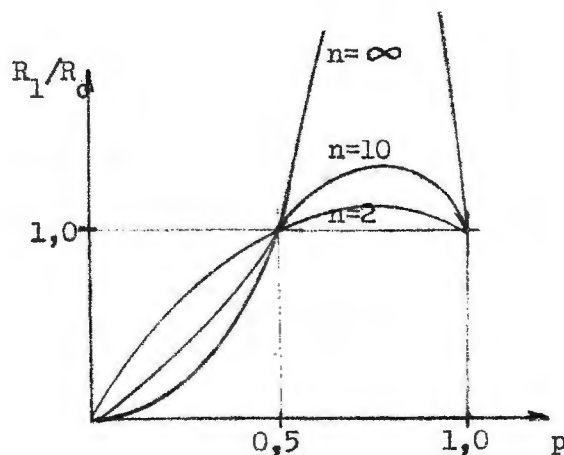


Fig 63.

logiken. Systemtillförlitligheten blir då:

$$R_2 = [p_m p^2 (3-2p)]^n$$

Om  $p_m = p$  fås:

$$R_2 = p^n [p^2 (3-2p)]^n = R_0 \cdot R_1$$

Eftersom  $R_1 < 1$  är  $R_2 < R_0$  dvs sämre än det icke redundanta systemet.  $p_m$  måste alltså vara större än  $p$ , vilket visar att majoritetsvalet inte får läggas på alltför låg nivå så att logikens felsannolikhet närmar sig den som gäller för de kontrollerade blocken.

Ett sätt att förbättra tillförlitligheten i majoritetslogiken är att införa även tre sådana som i fig 64. I det fallet kan en majoritetslogik felas. De

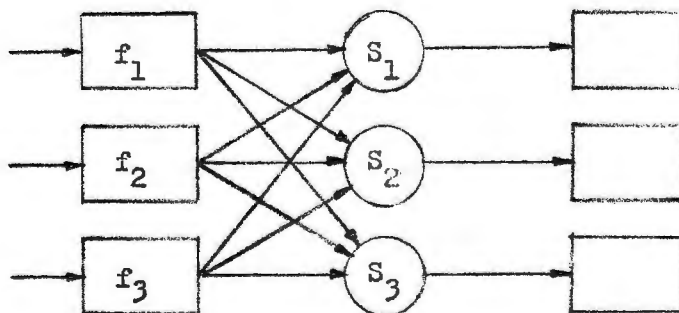


Fig 64.

två riktiga vidarebefordrar rätta data till nästa grupp om tre funktionsblock, där riktigt majoritetsval alltså kan göras. Med denna koppling erhålles även isolering av ingångarna för efterföljande block i en redundant grupp.

Fig. 65 visar i logisk form schemat för en majoritetsgrind.

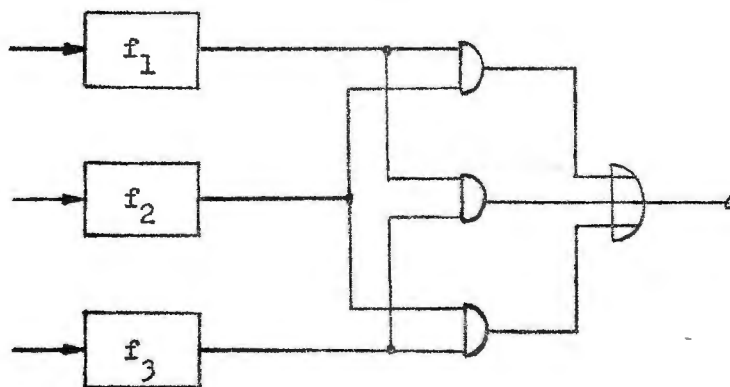


Fig 65.

Reparabla system

För system som är tillgängliga för reparation kan man erhålla mycket goda tillförlitlighetsvärden, om felaktigt funktionsblock indikeras och omedelbart utbytes. För beräkning av sådan tillförlitlighet är medeltiden för istandsättning av felaktig del av intresse. Låt oss kalla denna tid för  $\tau$ . Samolikheten  $q$  för att en redundant del skall vara ur funktion är då förhållandet mellan medelreparationstid och medeltid mellan fel (brukstiden) för denna del:

$$q = \tau/m = \tau \cdot \lambda \quad \dots \dots \dots (14)$$

Antag ett system med majoritetsval, där tillförlitligheten kan uttryckas med ekv (13).

$$q_s = 1 - R = 3 q^2 - 2 q^3$$

$q_s$  är då samolikheten för funktionsfel för hela det redundanta systemet. Insättning av uttrycket för  $q$  i ekv (14) ger:

$$q_s = 3\tau^2 \lambda^2 (1 - \tau \lambda) = \tau \lambda_s$$

där  $\lambda_s$  är felfrekvensen för det redundanta systemet.

$$\lambda_s = 3\tau \lambda^2 (1 - \tau \lambda)$$

Antag som i det tidigare exemplet, att  $\lambda = 0.005$  fel/tim och att medelreparationstiden per fel är 1 tim.

$$\lambda_s = 3 \cdot 25 \cdot 10^{-6} (1 - 5 \cdot 10^{-3}) \approx 75 \cdot 10^{-6}$$

$$m_s = 1/\lambda_s = 10^6 / 75 \approx 13.400 \text{ timmar}$$

#### 4.3. Några slutsatser

Det på sistnämnda sätt erhållna värdet för mellantid mellan funktionshindrande fel är mycket bra och pekar på en möjlighet att nedbringa uppbyggnaden för erhållande av redundansen. Trots att majoritetsvalet lagts på en hög nivå, uppnås god driftsäkerhet. Den höga nivån innebär i många fall stor komplexitet inom funktionsblocket exempelvis motsvarande en datamaskin. Därmed är också möjligheterna större att utan nämnvärd ökning av komplexiteten införa självcheckning i funktionsblocket exempelvis med hjälp av felupptäckande koder eller program (jfr avsnitt 4.2. femte stycket). I ett sådant fall behöver majoritetsval inte tillgripas, utan det räcker med två parallella funktionsblock, av vilka det felande kopplar ur sig själv eller ger larm om att urkoppling bör ske.

Det finns givetvis olika grader av de krav, som ställs på redundansens övertagande av arbetet vid uppträdande fel. Som tidigare nämnts, kan de variera från långvarigt funktionsbevarande i system, som inte är åtkomliga för reparation av redan uppståndsna fel, till övertagande av funktion medan felaktig del repareras. I det senare fallet kan också olika krav särskiljas. Det svåraste är, att redundant del automatiskt och omedelbart skall övertaga funktion vid fel, vilket kräver minst två parallellarbetande block för varje redundant funktion med automatisk urkopplingsmöjlighet för varje block. De parallella blocken kan antingen arbeta helt parallellt både på in- och utgångar (exempelvis vid majoritetsval) eller kan reservfunktionen endast vara ansluten till ingångarna men kopiera förloppen i den arbetande funktionen för att vid fel i denna kopplas in i funktionskedjan (stand by-körning). Den senare metoden förenklar bl.a. vissa synkroniseringsproblem.

I andra fall beroende på typ av övervakat system kan viss begränsad tid tillåtas för urkoppling av felande funktion och inkoppling av reserv. I ett sådant fall erbjuder sig möjligheten att konstruera systemet utan parallella funktioner, nämligen om systemet har möjlighet att själv exempelvis genom testrutiner och diagnosprogram (jfr fig. 31) detektera och lokalisera fel samt ge information till servicepersonal om nödvändigt utbyte av felande enheter.

Med tanke på den uppståndsökning som automatisk inkoppling av redundanta funktioner innebär, är det överhuvudtaget nödvändigt av ekonomiska och praktiska skäl att för varje system göra klart för sig, hur långa utbytetider som kan tolereras. Vad man strävar efter, är normalt ett icke parallellt system för manuellt utbyte, eftersom detta både är billigare och ger möjligheter till mer avancerade tekniska lösningar p.g.a. sin enkelhet.



LITTERATURREFERENSER

Referenser för avsnitt 2.1.

1. The USAF-Fairchild Specialized Digital Computer - J.J. Stone  
Mathematical Tables and Other Aids to Computation, 7(41):35-37 (Jan 1953)
2. A Special Purpose Digital Computer - J.P. Walker Jr  
Mathematical Tables... 7(3):190-195 (July 1953)

Referenser för avsnitt 2.2.

3. The Stability of Sampled-Data Systems - J. Tschauner  
Proceedings of the IEEE, Vol 51, No 4 (April 1963)
4. Designing Sampled Data Systems - C.M. Cundall and V. Latham  
Control Engineering, Oct. 1962 and Jan. 1963
5. Analog, Digital and Combined Analog-digital Computers for  
Real-time Simulation - C.G. Blayer and H. Mori  
Proc. Eastern Joint Computer Conf. Washington DC, Dec. 9-13, 1957, pp 104-110
6. The Philosophy of PCM - B.M. Oliver, J.R. Pierce and C.E. Shannon  
Proc. IRE 36(11):1324-1331 (Nov. 1948)
7. Om elektroniska telefonväxlar - K. Katzeff  
Elektronik nr 1, 1963
8. Mathematical Models of Air Traffic Control Systems - P. Kugel  
Computers and Automation, Vol 10 No 11 (Nov. 1961), pp 10-15
9. Automated Information-Processing Assistance for Military Systems -  
- M.O. Kappler - Computers and Automation, Vol 10 No 11-12 (Nov-Dec. 1961)

Referenser för avsnitt 3.1.

10. Integrated Automatic Control Systems - Applications and Frontiers -  
- J.R. Moore - Computers and Automation, Vol 11 No 11-12 (Nov-Dec. 1962)

Referenser för avsnitt 3.2.

11. Associative Memories - Electronic Design, Febr. 1, 1963
12. Data Addressed Memory Using Thin Film Cryotrons - V.L. Newhouse and R.E. Fruin  
Electronics, May 4, 1962, pp 31-36

Referenser för avsnitt 3.3.

13. Design Features of Current Digital Differential Analyzers - E.L. Braun  
IRE Convention Record, part 4, New York 1954, pp 87-97
14. Methods of Simulating a Differential Analyzer on a Digital Computer -  
- F. Lesh- J. Assoc. Computing Machinery 5(3):281-288 (July 1958)
15. CRT Display of Alpha-Numeric Information Applied to Radar Data Handling-  
- J.S. Johnston- Journal Brit. IRE, Aug, 1961, pp 139-143
16. On the Encoding of Arbitrary Geometric Configurations- H. Freeman  
IRE Trans. on Electronic Computers, June 1961, pp 260-268

Referenser för kapitel 4.

17. Digital-Circuit Reliability through Redundancy - A.A. Sorensen  
Electronics Reliability & Microminiaturization, Vol 1, 1962, pp 27-37,  
Pergamon Press, Great Britain
  18. Improvement of Electronic-Computer Reliability through the Use  
of Redundancy - W.G. Brown, J. Tierney and R. Wasserman  
IRE Trans. on Electronic Computers, Sept. 1961, pp 407-416
  19. Design of a Repairable Redundant Computer - R. Teoste  
IRE Trans. on Electronic Computers, Oct. 1962, pp 643-649
  20. An Automatic Self-Checking and Fault-Locating Method - F. Lee  
Samma publikation som ref. 19, pp 649-654
-