

CODEN: LUTFD2/(TFRT-5417)/1-103/(1989)

# Farkoststyrning och förarmiljö

Michael Edler  
Fredrik Holst

Institutionen för Reglerteknik  
Tekniska Högskolan i Lund  
Oktober 1989

<b>Department of Automatic Control</b> <b>Lund Institute of Technology</b> P.O. Box 118 S-221 00 Lund Sweden		<i>Document name</i> MASTER THESIS	
		<i>Date of issue</i> October 1989	
		<i>Document Number</i> CODEN:LUTFD2/(TFRT-5417)/1-103/(1989)	
<i>Author(s)</i> Micael Edler, Fredrik Holst		<i>Supervisor</i> Rolf Johansson, Dan Lindahl	
		<i>Sponsoring organisation</i> Kockums Marine AB	
<i>Title and subtitle</i> Farkoststyrning och förarmiljö (Man-machine adaptation of the steering gear in a submarine)			
<i>Abstract</i> <p>The purpose of thesis has been to examine the possibility of implementing feeling to a submarine steering gear, by studying how different combinations of feedback factors affected the person who is steering. The study has been made with the help of a yawing simulator that was designed solely for this purpose.</p>			
<i>Key words</i> Man-machine interaction, control systems, systems analysis			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i>			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 103	<i>Recipient's notes</i>	
<i>Security classification</i>			

The report may be ordered from the Department of Automatic Control or borrowed through the University Library 2, Box 1010, S-221 03 Lund, Sweden, Telex: 33248 lubbis lund.

# Farkoststyrning och förarmiljö

av

Micael Edler

Fredrik Holst

vid

Institutionen för Reglerteknik

Lunds Tekniska Högskola

Augusti 1989

# Farkoststyrning och förarmiljö

av

Fredrik Holst

Micael Edler

vid

Institutionen för Reglerteknik

Lunds Tekniska Högskola

Augusti 1989

## **Title**

Man-machine adaption of the steering gear in a submarine.

## **English abstract**

The purpose of this thesis has been to examine the possibility of implementing feeling to a submarine steering gear, by studying how different combinations of feedback factors affected the person who is steering. The study has been made with the help of a yawing simulator that was constructed solely for this purpose.

## Förord

Rapporten ni nu håller i er hand kan ses som Fredrik och Mikael's farväl till Lunds Tekniska Högskola. Vi har haft privilegiet att under en lång och het sommar få göra vårt examensarbete för Kockums Marine AB. Vi tackar Dan Lindahl och Anders Olson på Kockums samt Rolf Johansson vid Institutionen för Reglerteknik för deras stöd under arbetet. Vi vill även tacka våra familjer för att de stått ut med oss, speciellt till Mikael's bebis som haft förståndet att vänta med sin födsel.

# Innehållsförteckning

<b>1 Referat</b>	<b>4</b>
<b>2 Introduktion</b>	
2.1 Arbetets syfte	5
2.2 Arbetets bakgrund	5
2.3 Rapportens uppläggning	6
<b>3 Styrkänsla</b>	
3.1 Styrkänsla - vad är det ?	7
3.2 Våra sinnens betydelse	7
3.3 Styrkänsla i en ubåt	8
3.4 Ett sätt att förbättra styrkänslan	9
<b>4 Teori</b>	
4.1 Varför en teoretisk modell ?	12
4.2 Använda beteckningar	12
4.3 Matematisk modell för hastigheter i girled	14
4.4 Tillgängliga parametrar	14
4.4.1 Hydrodynamisk kraftpåverkan	15
4.4.2 Girvinkelhastigheten $r$	18
4.4.3 Giraccelerationen	18
4.4.4 Sladdhastigheten $v$	18
4.4.5 Rattdämpning	19

<b>5</b>	<b>Styrsimulatorn</b>	
5.1	Hela konstruktionen	20
5.2	Styrratten	22
5.3	Motorn	22
5.4	Förstärkaren	22
5.5	Gränssnittet	22
5.6	AD/DA-kortet	23
5.7	Datorn	23
5.8	Programmet	23
<b>6</b>	<b>Tillvägagångssätt</b>	
6.1	Arbetet i sin helhet	28
6.2	Litteratursökningen	28
6.3	Simulatorutrustningen	28
6.4	Programvaruutvecklingen	30
6.5	Utprovningen av återkopplingsalternativ	30
<b>7</b>	<b>Resultat</b>	
7.1	Utrustningen	32
7.2	Styrkänslan	33
<b>8</b>	<b>Slutsatser</b>	
8.1	Vinkelgivaren	34
8.2	Styrratten	34
8.3	Återkopplingen	35
8.4	Lovande koncept ?	36
<b>9</b>	<b>Rekommendationer</b>	37



## **Appendix A**

Signalkonditioneringskretsar	39
Lågpasfilter	42

## **Appendix B**

Programflödesschema	43
Procedurförklaring	44
Programlistning	52

## **Appendix C**

Teknisk specifikation	83
-----------------------	----

## **Appendix D**

AD/DA-kort	86
------------	----

## **Appendix E**

Artiklar	88
----------	----

## **Appendix F**

Användarmanual	90
----------------	----

## **Referenser**

102
-----

# 1 Referat

En ubåt är en farkost med stor tröghet, vilket innebär att det tar lång tid innan den svarar på förarens rattrörelser. Dessutom finns det inga naturliga synintryck tillgängliga för föraren, vilket tillsammans med dynamiken gör att det är svårt att manuellt styra farkosten med precision. Vår uppgift har varit att ta fram ett sätt att ge ubåtsföraren bättre styrkänsla, genom att koppla information om hur ubåten beter sig till ubåtens styrratt. För att göra detta har en styrsimulator konstruerats i vilken utprovning av lämpliga återkopplingsparametrar har studerats. Vi har då funnit att återkopplingen bör vara enkel för att föraren skall kunna tillgodogöra sig informationen samt att styrratten inte bör gå att vrida mycket fortare än rodet. Vidare har vi funnit att kraften i ratten kändes bättre ju snabbare vi kunde uppdatera informationen. En lämplig frekvens att uppdatera den med ligger kring 100 Hz.

## 2 Introduktion

### 2.1 Arbetets syfte

Arbetet har gått ut på att undersöka ett sätt att förbättra möjligheten för föraren av en farkost med stor tröghet och utan fysisk koppling mellan roder och ratt att styra denna med ökad kontroll. Det sätt som vi studerat bygger på att man ska överföra information om hur farkosten rör sig via ratten till föraren.

Vårt arbete som bygger på ett tidigare examensarbete [1] har till stor del handlat om att utveckla ett verktyg för att prova ut vilken information som är lämplig för att styra det vridmoment föraren känner i ratten. Det verktyg som har utvecklats består dels av en ubåtsratt fastsatt i ett stativ tillsammans med en momentgivande motor och styrelektronik, dels av en dator som simulerar ubåtens uppträdande vid olika rattutslag och som beräknar den signal som används till att styra den till motorn kopplade till ratten.

### 2.2 Arbetets bakgrund

Känslan av kontroll när man framför en farkost är viktig eftersom den direkt påverkar hur exakt farkosten kan framföras. Om man bygger en större farkost blir de i styrningen inblandade krafterna så stora att någon direkt koppling mellan dessa och föraren inte längre är möjlig. I och med detta försvinner en direkt och naturlig möjlighet till att erhålla en känsla av kontroll, s k styrkänsla, genom att känna de krafter som verkar på rodren. Arbetet har gjorts åt Kockums Marine AB som tillverkar ubåtar. I den typen av farkost försvinner även två andra sätt att direkt uppleva styrkänsla - genom synintryck och vår förmåga att uppleva acceleration. Därför bör man istället skapa styrkänsla genom att indirekt påverka föraren med hjälp av tillgänglig information om farkosten och dess

beteende. I det tidigare utförda examensarbetet studerades vad informationen om en ubåts rörelser bestod av och hur man kunde utnyttja denna för att åstadkomma styrkänsla. För att kunna göra realistiska tester och prov konstaterades att man behövde en modell som simulerade den verkliga förarmiljön i en ubåt - en simulator.

Med hjälp av en matematisk modell av ubåtens dynamik gjordes ett förslag till hur simulatoren skulle konstrueras. Från och med detta stadium tog vi över och slutförde konstruktionen samt gjorde vissa utprovningar för att kunna demonstrera huruvida styrkänsla kan uppnås.

### **2.3 Rapportens uppläggning**

Författarna har haft ambitionen att skriva en lättläst rapport, då vi vill att så många som möjligt skall kunna tillgodogöra sig rapportens innehåll. Hänvisningar till referenser görs med hakparenteser [] och övriga hänvisningar i texten görs med hjälp av vanliga parenteser (sida, figur).

## 3 Styrkänsla

### 3.1 Styrkänsla - vad är det ?

Styrkänsla kan beskrivas som den känsla av kontroll föraren av en farkost upplever när han/hon styr. Rorgängaren av en Optimistjolle styr jollen genom att hålla i rorkulten och alternativt föra den i lä eller lovart. Nästan omedelbart svarar jollen på alla manövrar och de flesta av rorgängarens sinnen påverkas också av jollens rörelser.

Rorgängaren ser hur jollens kurs ändras, hur ljuset och skuggorna förändras. Vinden blåser in mot jollen från ett nytt håll, träffar rorgängaren i en ny vinkel och varje förändring innebär ett nytt ljud. Kraften som känns i rorkulten påverkas av vattenmassornas tryck mot rodrets ytor. Jollen kränger som en följd av vindens styrka och riktning. Alla dessa intryck bidrar till att rorgängaren kan styra jollen mycket precist och med hög kontroll - flera sinnen samverkar på ett naturligt, verkligt och för människan enkelt sätt. Detta kan vara ett exempel på den ena extremen av styrkänsla - den bästa.

Piloten av ett litet enmotorigt propellerplan styr planet genom att vrida en ratt och samtidigt dra den mot sig eller skjuta den ifrån sig. Även här hjälper många av pilotens sinnen till för att bidra till känslan av kontroll.

### 3.2 Våra sinnens betydelse

Problemet med att uppleva kontroll uppstår alltmer i takt med att föraren berövas på sinnesintryck som beskriver hur förflyttningen fortskrider i förhållande till hans/hennes tidigare position. Människan har lärt sig att bygga farkoster som alltmer möjliggör för henne att färdas på helt nya sätt. Man bygger supertankers, jumbojets, överljudsplan, månraketer, undervattensfarkoster, snabbtåg etc., men människan har inte

tillnärmelsevis utvecklats i samma hisnande takt. Hon har inte större möjlighet idag, än hon hade för tusen år sedan att tolka sina sinnesintryck. Mycket snabba eller långsamma förlopp är svåra att uppfatta och tolka för henne. Hon är helt beroende av sina sinnesintryck för att kunna bedöma hur hon förflyttar sig. Större och mer tekniskt komplicerade farkoster innebär ofta att mängden och kvaliteten på de naturliga och verkliga sinnesintrycken minskar - därmed ökar också behovet av att skapa bra hjälpmedel, som kan ersätta vad som förlorats (s88).

Vilket sinne är det då som påverkar styrkänslan mest ? Rent intuitivt kan man sluta sig till att det är vår syn som spelar den största rollen, eftersom det är det sinnet som utnyttjas mest vid framförandet av en farkost. Även hörseln hos människan är relativt bra och bidrar till styrkänslan. Människans balanssinne kommer inte till användning i någon större utsträckning när större farkoster framförs, med undantag av luftburna fordon, eftersom hennes kroppsposition då inte förändras nämnvärt. Med känseln uppfattar vi bl. a. tryck mot huden vilket har betydelse för styrkänslan. Vi kan uppleva att vi svänger, genom att vi trycks åt sidan mot någonting när farkosten kränger, att vi accelereras genom att vi trycks bakåt. Med känseln uppfattar vi även små rörelseförändringar hos något vi håller i handen. Ett annat sinne av betydelse i detta sammanhang är muskelsinnet som vi använder för att känna vikt och kraft. Men att känna någon finare kvantitativ skillnad med muskelsinnet är svårt om större muskelgrupper används.

### 3.3 Styrkänsla i en ubåt

En ubåt är en farkost som passar människan dåligt med avseende på styrkänslan. Vid manuell styrning har föraren att förlita sig på synintryck från en hastighetsmätare och en mätare som visar hur fort ubåten girar

samt en kompass. I bästa fall kan föraren även få information av en utkik. Några andra sinnesintryck existerar inte för föraren som hjälper honom att styra. Styrningen sker med hjälp av en ratt som inte alls påverkas av hur ubåten rör sig. Ratten är endast fjäderbelastad. Ubåtens rörelse är dessutom en långsam rörelse som ytterligare försvårar styrandet. Styrkänslan kan sägas ha nått nära den andra extremen - den sämsta.

Det som går att förbättra med avseende på styrkänslan av ubåten är synintrycken ifrån instrumenten och känselintrycken från ratten. Man kan även tänka sig att använda någon form av ljudinformation för att hjälpa upp styrkänslan. Vad det gäller instrumenteringen kan man tänka sig lite olika alternativ. Ett mycket lockande alternativ kan vara att presentera ubåtens läge i förhållande till ett horisontalplan och ett vertikalplan på en grafisk skärm. Skärmen skulle då t. ex. visa fyra rutade väggar som gick ihop tredimensionellt - upplevelsen skulle bli att man såg en slags korridor. Hastighet skulle då representeras av ett rutnätet rörde sig snabbare mot föraren, riktning av att vinkeln till väggarna förändrades i takt med den hastighet ubåten girade. I kombination med ett hjälpmedel som finns på t.ex. stora ytgående fartyg [2], en sk estimer, kunde styrkänslan förbättras avsevärt. En estimator är ett hjälpmedel som med hjälp av aktuell vinkel på rodret och tillgängliga tillståndsp parametrar förutsäger hur kursen kommer att vara en viss tidsperiod längre fram i tiden, om samma rodervinkel hade behållits. Resultatet kan visas grafiskt eller med siffror t. ex. angivande kursen.

### 3.4 Ett sätt att förbättra styrkänslan

Författarna har fått till uppgift att undersöka ett annat sätt att förbättra styrkänslan på ubåten - genom påverkan på ratten av ubåtens rörelse. Som tidigare nämnts innebär det vissa problem för människan att känna skillnader i tryck och kraft, men i jämförelse med att inte ha någon

verklighet alls knuten till ratten, måste skillnaden ändå bli stor om återkoppling används. Men att kvalitativt och kvantitativt försöka beskriva styrkänsla är givetvis mycket svårt - hur bedömer man känslor överhuvudtaget och kanske ännu svårare - hur kan man förmedla vad man upplevt till andra människor ?

Att en ratt används har berott på att det är vad som används idag för att styra en ubåt. Därmed inte sagt att det skulle vara det bästa - tvärtom kanske. I en tillämpning där det gäller för föraren att känna relativt små kraftförändringar måste en ratt anses som ett tämligen mediokert verktyg, eftersom alltför stora muskelgrupper tas i anspråk. En liten styrspak skulle kanske lämpa sig bättre [3]. Vidare avger ratten en styrsignal beroende på rattens läge - man skulle kunna tänka sig andra sätt, t. ex. att styrsignalen beror på det moment som utövas på ratten - eller styrspaken. Försök med detta i moderna jaktplan [3] har visat att styrsignalen kan förändras mycket snabbt men att det har varit svårt för föraren att känna konsekvenserna av sin påverkan. Roderutslagen blev stora i förhållande till det utövade momentet. Vidare kan man diskutera om styrsignalen från styrverktyget, i vårt fall ratten, skall uppvisa samma beteende oberoende av rattens relativa läge och/eller oberoende av hur läget förändrats i tiden, d v s skall rattens koppling till rodren vara i någon form olinjär istället för linjär.

### 3.5 Vilka parametrar ska påverka styrkänslan ?

De parametrar som finns tillgängliga för att påverka styrkänslan i en ubåt är hastigheten med vilken ubåten rör sig i sin egen längdriktning, hastigheten med vilken den rör sig i sidleds och hastigheten med vilken den vrider sig runt sin egen vertikala axel. Även accelerationen runt ubåtens vertikala axel och rodrens vinkel är tillgängliga. De krafter som verkar på rodren skulle också kunna göras tillgängliga. Vilken av eller



vilken kombination av dessa parametrar som skall användas är naturligtvis mycket svårt att säga. För att få reda på det måste försök och utprovningar göras. Men det är viktigt att komma ihåg att syftet med att låta ubåtens rörelser påverka ratten är att hjälpa föraren kontrollera dessa rörelser. Om påverkan sker på ett enkelt och okomplicerat blir möjligheten mycket större för föraren att tillgodogöra sig informationen han/hon förmedlas via ratten. Problemet ligger i att få föraren att "se" hur ubåten reagerar på vad han/hon gör med ratten genom att helt enkelt känna det i händerna.

Att en ratt används innebär i ännu högre grad att påverkan måste vara okomplicerad och att relativt tydliga förändringar krävs, eftersom hela armens muskler används för att vrida den. Att det rör sig om ett långsamt förlopp försvårar ytterligare, genom att man använder statisk kraftpåverkan från armarna på ratten istället för dynamisk. Vi känner nämligen förändringar sämre om musklerna utövar en statisk kraft.

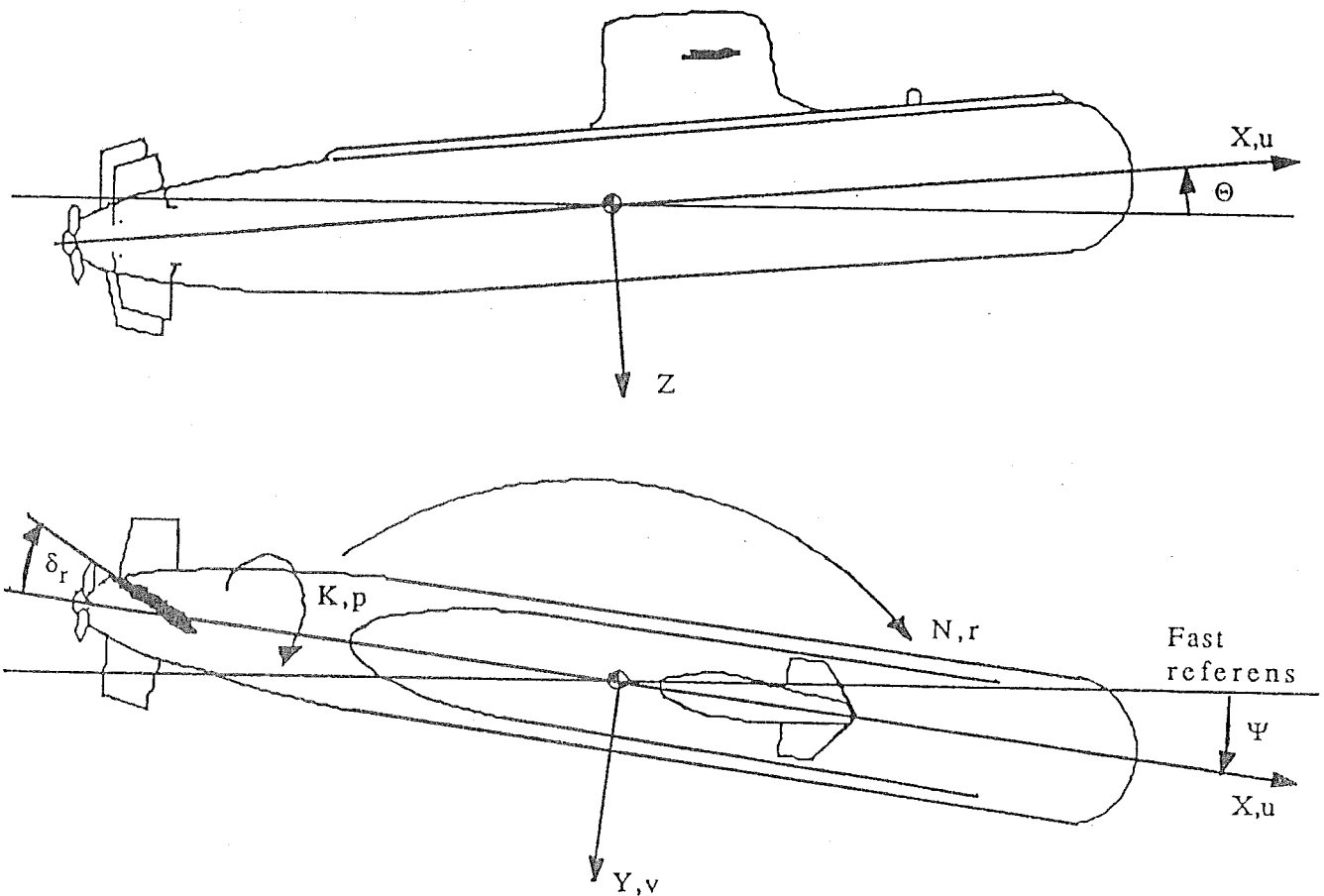
## 4 Teori

### 4.1 Varför en teoretisk modell ?

Det finns två möjligheter att prova styrkänslan i en ratt, det ena är att utföra försök i den verkliga miljön, i det här fallet en ubåt, och där köra farkosten med olika roderutslag i olika hastigheter för att på så sätt prova ut lämpliga parametrar att koppla till föraren via ratten. Det är naturligtvis ekonomiskt oförsvarbart att i ett inledande skede prova direkt i den verkliga miljön. Därför är vi hänvisade till den andra möjligheten nämligen att prova styrkänslan med hjälp av en modell av verkligheten. Den modell vi använder finns kort beskriven nedan och det är också denna modell som är implementerad i styrsimulatorens.

### 4.2 Använda beteckningar

För att beskriva en ubåts rörelser finns det ett standardiserat beteckningssätt [4]. Vid rörelser i girled (horisontalled) behövs de beteckningar som visas i figur 4.1 på nästa sida.



Figur 4.1 Beteckningar för en ubåts rörelse i girled.

Variabler:	$u$	Ubåtens hastighet i x-riktningen (skrovfast).
	$v$	Ubåtens hastighet i y-riktningen (skrovfast).
	$r$	Ubåtens vinkelhastighet kring z-axeln relativt vattnet.
	$p$	rollhastighet
	$\Psi$	Girvikel
	$\delta_r$	Girrodervinkel kan delas upp i kryssroderkomponenterna $\delta_1 + \delta_2 + \delta_3 + \delta_4$ .
	$n$	Propellervarvtal

Konstanter:	$\rho$	Vattnets densitet
	$D$	Propellerdiameter
	$g$	Tyngdaccelerationen
	$m$	Båtens massa
	$l_N$	Normerat avstånd mellan roder och tyngdpunkt
	$k_{T0}$	Propellerkonstant
	$k_{TJ}$	Propellerkonstant

### 4.3 Matematisk modell för hastigheter i girled

För att beskriva en ubåts rörelse fullständigt behövs sex ekvationer; tre translationer i x-, y- och z-led samt rotation kring de tre koordinataxlarna. Statens Skeppsprovninganstalt har tagit fram den modell [5] som vi har använt oss av i vår simulator. Modellen kan förenklas något utan att resultatet nämnvärt förändras. För det första kan antas att ubåten är symmetrisk, för det andra att hastigheten i x-led, u, endast beror av propellervarvtalet n. Styrsimulatorn har bara konstruerats för manövrar i girplanet vilket begränsar antalet frihetsgrader för ubåten till tre, varför antalet ekvationer kan minskas till tre, nämligen translation i x- och y-led samt rotation kring z-axeln. Efter en del räknande [1] kommer man fram till följande uttryck för ubåtens sladdhastighet v och girvinkelhastighet r.

$$\begin{bmatrix} v \\ r \end{bmatrix} = e^{A(t-t_0)} \begin{bmatrix} v_0 \\ r_0 \end{bmatrix} + \int_{t_0}^t e^{A(t-s)} B u(s) ds$$

Det är alltså med hjälp av denna formel som vi beräknar sladdhastigheten v och girvinkelhastigheten r som i sin tur används i kombination med hastigheten u för att beräkna ubåtens läge i förhållande till en startpunkt. Vi använder också sladdhastighet och girvinkelhastighet tillsammans med övriga tillgängliga parametrar vilka räknas upp senare i kapitlet för att komponera det uttryck som vi använder för att åstadkomma mer styrkänsla för föraren. När vi gör de här beräkningarna i vårt datorprogram, så finns A och B matriser inlagda för varje heltals hastighet mellan 0 - 20 knop samt för fyra olika sampelfrekvenser 25, 50, 75 och 100 Hz. A- och B-matriserna har beräknats i programpaketet Matlab.

### 4.4 Tillgängliga parametrar

När man bestämmer sig för att försöka åstadkomma en ökning av styrkänslan i styrratten, så ställs man direkt inför frågan om vilken

information som är lämplig att koppla till föraren via det moment som känns i ratten. De olika parametrar man kan återkoppla är den på rodret verkande hydrodynamiska kraften, girvinkelhastigheten, girvinkelaccelerationen, sladdhastigheten eller en parameter som motsvarar rodrets tröghet. Nedan följer en beskrivning av de nu uppräknade parametrarna, som alla finns tillgängliga i den använda modellen.

#### 4.4.1 Hydrodynamisk kraftpåverkan

Den hydrodynamiska kraftpåverkan på ett roder kan beskrivas med hjälp av Kutta och Joukowskis teorem [6], som säger att en vinge som är i vila i en ström, kommer att påverkas av två krafter dels en mot strömmen vinkelrät lyftkraft  $L$  enligt följande

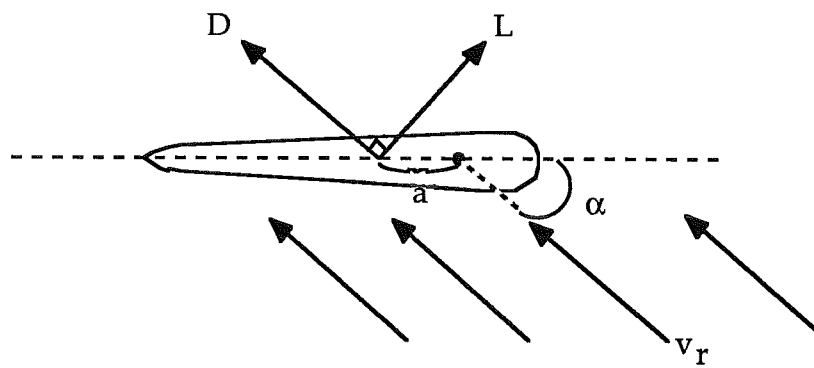
$$L = C_L \frac{1}{2} \rho A v_r^2 a \quad C_L = \text{Lyftkoefficienten}$$

samt en med strömmen riktad motståndskraft  $D$  [7]

$$D = C_D \frac{1}{2} \rho A v_r^2 \alpha^2 \quad C_D = \text{Motståndskoefficienten}$$

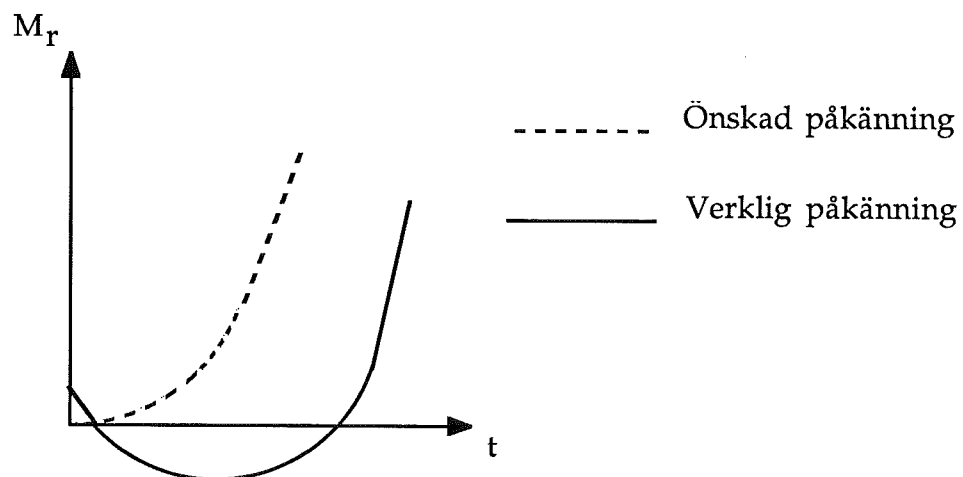
De båda krafterna kommer att angripa vingen (rodret) i en punkt på avståndet  $a$  från dess vridningsaxel (figur 4.2). Avståndet  $a$  är egentligen en funktion av vätskeströmmens attackvinkel, men är här approximerad till ett konstant värde. Avståndet  $a$  kommer då att vara den hävarm som lyftkraften och motståndskraften angriper rodret med. Detta ger det totala hydrodynamiska momentet som verkar på rodret i förhållande till roderaxeln lika med

$$M_r = L a \cos(\alpha) + D a \sin(\alpha)$$



Figur 4.2 Vätskeströmning mot en ving.

På grund av icke minimumfas effekt i rodret gäller ovanstående endast om rodret legat ute tillräckligt länge.



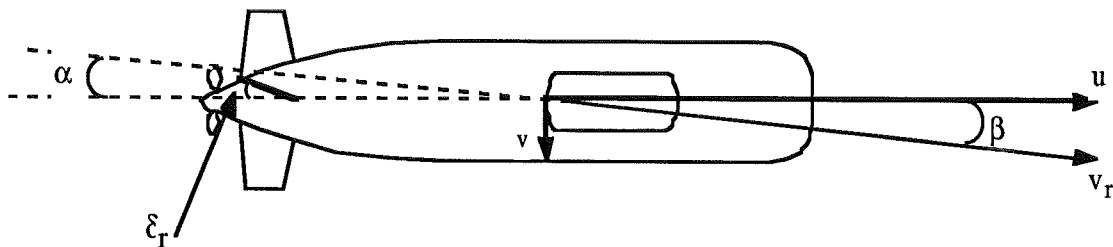
Figur 4.3 Minimumfas

Minimumfas effekten gör att ett litet roderutslag kommer att resultera i en motsatt rörelse av farkosten än den önskade (figur 4.3). Detta i sin tur medför att det inte är lämpligt att mäta upp den hydrodynamiska kraftpåverkan för att på så sätt få fram en återkopplingsbar parameter, eftersom kraften i ratten kommer att skifta tecken då man vrider ut ratten från dess neutralläge till ett rattutslag.

Vid en ren gir erhåller ubåten hastigheten  $u$  i x-led, sladdhastigheten  $v$  i y-led samt en rotationshastigheten  $r$  runt z-axeln. Ubåten kommer att möta en vattenström som svarar mot den resulterande hastigheten  $v_r$  med vinkeln  $\beta$  (figur 4.4) . Rodret träffas också av  $v_r$  men under vinkeln

$$\alpha = \delta_r - \beta$$

där  $\delta_r$  är rodervinkeln i förhållande till ubåtens x-axel.



Figur 4.4 Ubåt vid gir

Uttrycket för vridmomentet kring rodrets rotationsaxel blir nu

$$M_r = a \frac{1}{2} \rho A v_r^2 \left[ (\delta_r - \beta) C_L \cos(\delta_r - \beta) + (\delta_r - \beta)^2 C_D \sin(\delta_r - \beta) \right]$$

Intuitivt kan man se vattnets påverkan på rodret på så sätt att när man först lägger ut rodret, så kommer vattenmassorna att försöka återföra rodret till ursprungsläget, varför kraftverkan på rodret blir stor initialt. När ubåten börjar gira, kommer vattnet att strömma parallellt med rodret, varvid den återförande kraften minskar. Det är denna mänskliga intuition som skall hjälpa föraren att förstå farkostens uppträdande, om vi väljer att återkoppla vattnets kraftpåverkan på rodret till föraren via ratten.

Man skall observera att den hydrodynamiska påverkan på rodret dock inte är den enda som spelar roll, då man vill efterlikna de verkliga krafterna på rodret. Ytterligare påverkan utgörs av den tröghet rodrets massa ger upphov till, samt friktion i de styrleder som rodret styrs av.

#### 4.4.2 Girvinkelhastigheten $r$

Girvinkelhastigheten är en kontinuerlig parameter som går att återkoppla till föraren och den är ett direkt mått på hur fort farkosten svänger. Girvinkelhastigheten ökar med stegrande hastighet och roderutslag. När man lägger ut rodret mycket så växer girvinkelhastigheten snabbt till ett stationärt värde om ubåtens hastighet i x-led är hög. Det går betydligt långsammare och den stationära nivån blir lägre om ubåten färdas långsamt eller man har ett litet roderutslag.

#### 4.4.3 Girvinkelaccelerationen

Girvinkelaccelerationen fås genom att derivera girvinkelhastigheten, och den är ett mått på hur fort girvinkelhastigheten ändras. Girvinkelaccelerationen får stora värden, när rodret ändras snabbt och hastigheten är hög.

#### 4.4.4 Sladdhastigheten $v$

Sladdhastigheten är den hastighet som ubåten har vinkelrät mot hastigheten i x-led. Den är stor i början av ett roderutslag för att därefter minska i takt med att "rodret tar" och farkosten girar. Sladdhastigheten når sitt maxvärde, då man byter roderutslag från det ena ändläget till det andra.



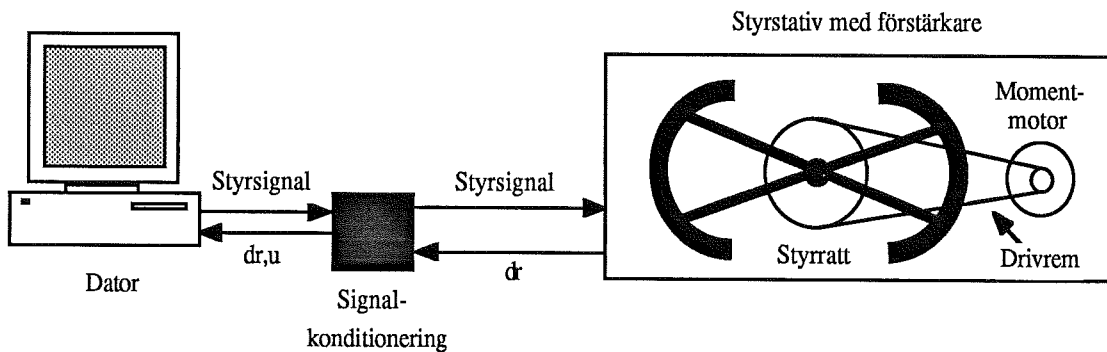
#### 4.4.5 Rattdämpningen

Genom att derivera signalen från ratten får man en signal, vars värde skall göra det möjligt att efterlikna den tröghet i rodret som finns i en verklig ubåt, där man inte kan röra rodret fortare än en viss maxhastighet. Den här signalen fugerar så att den blir stor då skillnaden mellan det gamla värdet som skall uppdateras och det nya värdet är stor. Detta kan man utnyttja för att minska den hastighet man kan vrida ratten med, så att man inte kan röra ratten fortare än rodret.

## 5 Styrsimulatore

### 5.1 Hela konstruktionen

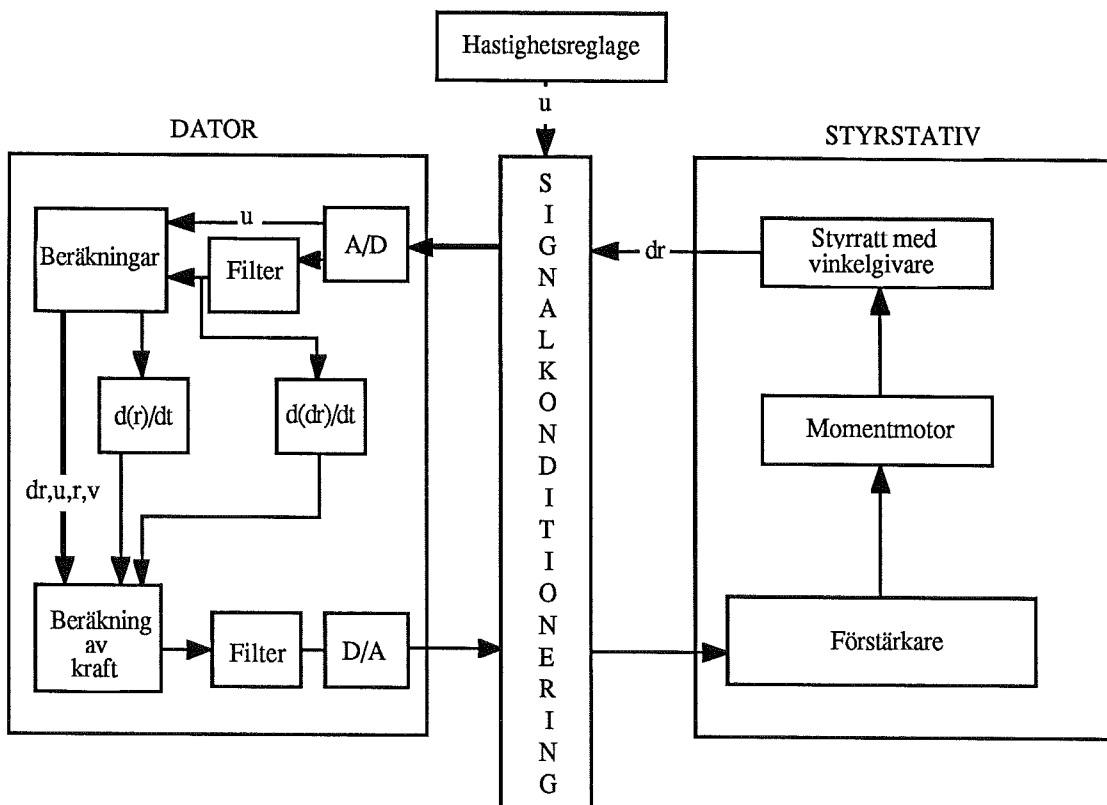
Styrsimulatore i sin helhet består av tre åtskiljbara byggblock - en persondator med programvara, ett styrstativ bestående av en förstärkare, en likströmsmotor och en styrratt med tillhörande vinkelgivare, och ett gränssnitt som förbinder datorn och styrstativet med varandra (se figur 5.1).



Figur 5.1 De tre byggblocken som utgör styrsimulatore.

Persondatorn innehåller en Intel 80386-processor och en 80387-matematikprocessor [12] samt ett 12-bitars AD/DA-kort för kommunikation med styrstativet (s87). På styrstativet är styrrattens axel förbunden via en kuggrem i gummi med likströmsmotorn, vars axel i sin tur är förbunden med axeln på en potentiometer. Potentiometern används följaktligen som vinkelgivare. Spänningen till potentiometern tas ifrån förstärkarkortet och signalen ifrån den passerar gränssnittet och når datorn via en AD-omvandlare och ett digitalt låpassfilter.

Styrsignalen kommer ifrån datorn via ett ej inkopplat digitalt låpassfilter, DA-omvandlaren och gränssnittet. Därefter når den förstärkaren, som är kopplad så att den skall styra motorn som en momentmotor. Innehållet i gränssnittet består av kretsar som anpassar signalnivån ifrån vinlelgivaren (ligger mellan -15 och 15 V) till AD-omvandlarens insignalområde (0 till 9 V) (s41), ifrån DA-omvandlaren (0 till 9 V) och förstärkaren (s40). Där finns även en vridpotentiometer med tillhörande kretsar som används till att ställa in ubåtens hastighet i knop (s39) och två stycken icke inkopplade analoga låpassfilter - ett med 20 Hz gränshfrekvens och ett med 30 Hz gränshfrekvens (s42). (Se figur 5.2)



Figur 5.2 Detaljerat blockschema över styrsimulatore och dess ingående delar.

## 5.2 Styrratten

Robust monterad på styrstativet sitter en likadan styrratt som finns ombord på ubåtar idag. På dess lagrade axel är ett drivhjul monterat. Drivhjulet är förbundet med axeln på en motor via en tandad kuggrem i gummi.

## 5.3 Motorn

Den motor som påverkar ratten via en gummirem är en likströmsmotor. Den styrs med hjälp av en förstärkare som är kopplad så att motorn skall avge ett moment till ratten. Motorn kan även kopplas så att den håller en viss hastighet. Motorns maxmoment är 6,4 Nm vilket är fullt tillräckligt (s84).

## 5.4 Förstärkaren

Till motorn är en förstärkare kopplad med vars hjälp man kan skicka en styrsignal till ratten. Man kan själv välja i vilken tillämpning motorn skall användas; momentgivande eller hastighetshållande. Förstärkaren ger användaren tillgång till en mängd signaler, som kan användas som hjälpmedel vid styrning av motorn. Man bestämmer enkelt förstärkningen genom att ställa in en potentiometer. Det finns möjlighet att begränsa den momentana storleken och medelvärdesstorleken på strömmen med två andra potentiometrar. (s83)

## 5.5 Gränssnittet

Inuti en svart plastlåda finns de analoga kretsar som anpassar alla signaler som går mellan datorn och styrstativet. Två stycken lågpasfilter, med 20 Hz respektive 30 Hz gränshfrekvens, finns också tillgängliga. En vridpotentiometer som används för att reglera hastigheten på ubåten sitter på lådans utsida. (s85)

### 5.6 AD/DA-kortet

För att konvertera de analoga signalerna till digitala och vice versa används ett instickskort avsett för en IBM PC-AT kompatibel, instucket i en av kortplatserna i datorn. Det innehåller 1 kanal 12-bitars digital till analog-omvandlare och 16 kanaler 12-bitars analog till digital-omvandlare. Omvandlingen kontrolleras mjukvarumässigt men i och med att en 386-baserad persondator med 20 MHz klockfrekvens används för att exekvera simulatorprogrammet, innebär det att en viss fördröjning måste ske i inläsningsrutinen vid analog till digital-omvandlingen för att kortet skall hinna utföra omvandlingen (s87). Den totala omvandlingstiden för varje kanal för analog till digital omvandling blir ca 0.8 ms och för digital till analog omvandling ca 0.2 ms. Det skiljer sig ifrån de värden som angetts i den tekniska specifikationen för kortet, vilket troligtvis beror på att fördröjningsrutinerna vid inläsningen inte optimerats. (s86)

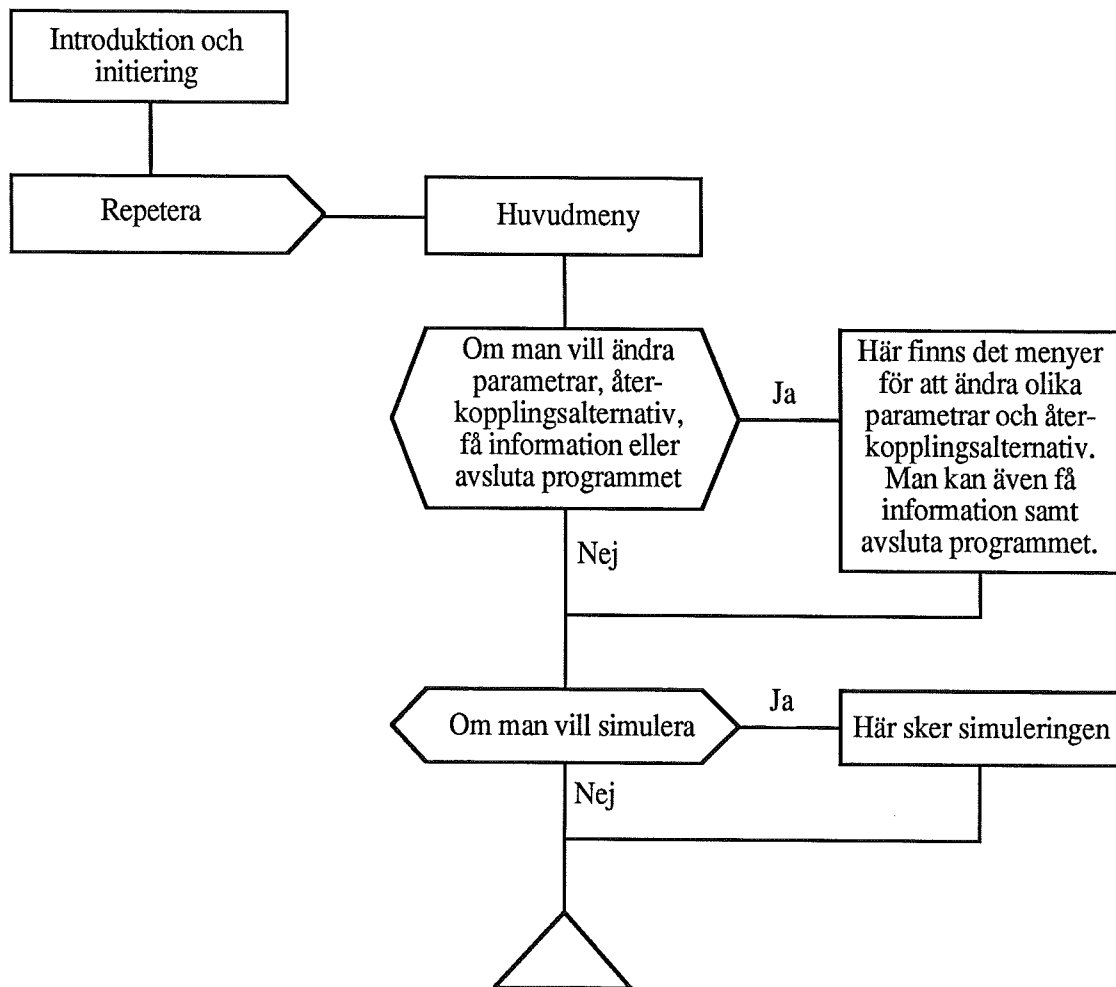
### 5.7 Datorn

För att simuleringen av ubåtens rörelser skall kunna presenteras grafiskt, samtidigt som styrsignaler skickas ut till förstärkaren, måste en tillräckligt snabb dator användas. Den dator konstruktörerna arbetat med har haft en 80386-processor med en 80387-coprocessor som klockats med 20 MHz. Tack vare datorns snabbhet har nya mätvärden ifrån rattens vinkelgivare och den hastighetsbestämmande potentiometern kunnat insamlats, och nya styrsignaler skickats iväg till förstärkaren med ett intervall på endast 10 ms. Det korta intervallet har spelat stor roll för den känsla som erhållits i ratten under pågående simulering.

### 5.8 Programmet

Det datorprogram som används för att kunna använda utrustningen har medvetet gjorts mycket användarvänligt och flera funktioner som avser att skydda användaren och utrustningen finns. Programmet ger

användaren möjlighet att påverka det moment motorn ger på ratten under simuleringens gång genom att välja mellan olika alternativ i olika menyer. I programmet beräknas de parametrar som behövs för att kunna simulera u-båtens rörelser i girled grafiskt på skärmen och för att därefter kunna beräkna den styrsignal motorn är i behov av för att kunna ge rätt moment på ratten. (se figur 5.3)



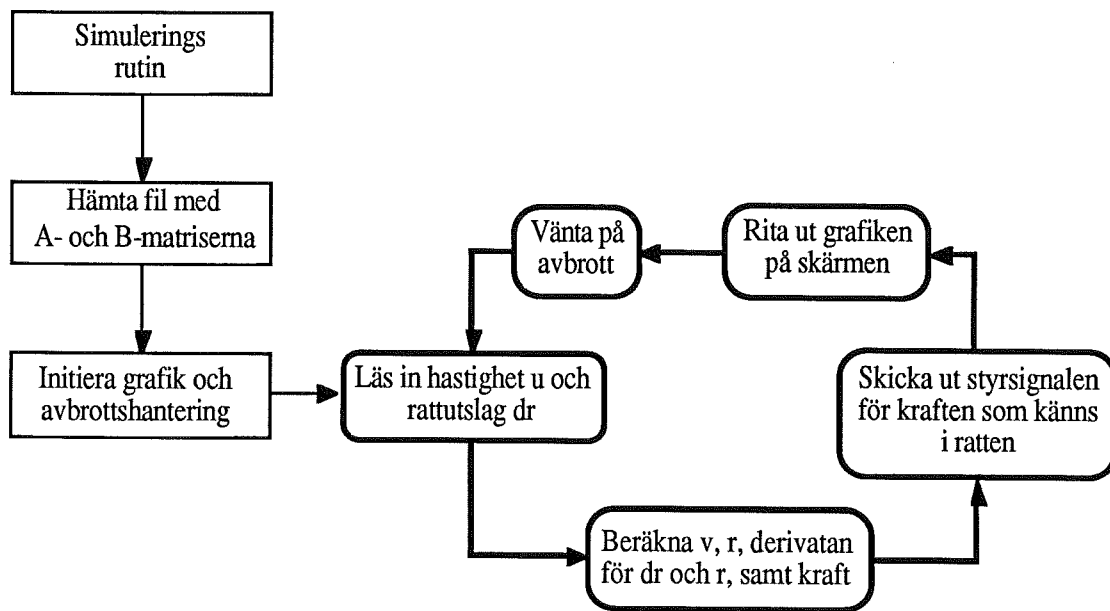
**Figur 5.3.** Uppbyggnaden av det program som används vid simulering. I en huvudmeny väljer användaren vilken aktivitet som skall utföras. Det finns möjlighet att välja vilken sammansättning av parametrar den

styrsignal som reglerar kraften på ratten skall ha. Även storleken på de ingående parametrarna kan mycket enkelt och snabbt ändras genom att välja respektive aktivitet i huvudmenyn. I programmet är digitala lågpasfilter och deriverande digitala regulatorer realiserade (regulatorerna presenterade som  $d(r)/dt$  och  $d(dr)/dt$  i figur 5.2) för att klara av den signalbehandling som krävs [8]. Ett menyalternativ kan väljas så att användaren kan ändra de matematiska konstanterna som används i dessa.

Samtliga konstanter tilldelas värden vid programmets uppstartande,  $s$   $k$  defaultvärden. Dessa värden har bedömts vara bra utgångsvärden för fortsatta försök och en mer noggrann intrimning av momentet som känns i ratten. Det är lätt att i programmet återfå dessa utgångsvärden efter att ha ändrat dem genom att välja rätt alternativ i huvudmenyn.

Orsaken till att programmet har fått denna utformning är att det skall vara så flexibelt som möjligt för användaren. Det går lätt och snabbt att ändra så att ett moment helt olikt det föregående känns i ratten. Olika användare kan mata in sina egna speciellt utprovade parametervärden till det uttryck för kraften som valts. Det är lätt att experimentera och prova helt nya återkopplingar på detta sätt.

Själva simuleringsrutinen fordrar en lite mer detaljerad figur (se figur 5.4).



Figur 5.4 Simuleringsrutinen åskådliggjord mer detaljerat.

När simuleringen startas, hämtas först den fil som innehåller de matrisvärden som behövs för att beräkna ubåten hastighet åt sidan, sladdhastigheten  $v$ , och hastigheten med vilken ubåten vrider sig runt sin egen vertikala axel, girvinkelhastigheten  $r$ . Därefter initieras de rutiner som behövs för att datorn skall kunna rita grafiskt på skärmen i färg och som innebär att mjukvaruavbrott görs med vald frekvens.

Nu kommer programmet att upprepa samma rutin med en förutbestämd frekvens, tills användaren vill avbryta eller tills det blir för stora variationer i styrsignalen till motorn. Rutinen innebär att datorn frågar efter insignaler på ingångarna på AD-omvandlaren, rodervinkeln  $dr$  och hastigheten  $u$ , och beräknar  $v$  och  $r$  med hjälp av rätt A- och B-matriser. I beräkningarna tar programmet hänsyn till att rodret har en viss maximal hastighet som det kan röra sig med. Även derivatorna för  $r$  och  $dr$  beräknas med hjälp av de digitala regulatorerna. Resultatet blir en ny



parameter, giracceleration och ytterligare en parameter som kallats för systemtröghet. Systemtrögheten är menad att simulera det motstånd som skulle kännas i ratten om ratt och roder var sammankopplade mekaniskt. Då skulle motståndet bero på rodrets massa, friktion i lager och begränsningen av den kraft, som hydralsystemet kan ge för att röra rodret. Systemtrögheten används alltså för att begränsa den hastighet med vilken ratten vrids. Sista beräkningen resulterar i den styrsignal, som sedan skickas ut till DA-omvandlaren och vilken sedan ger upphov till ett moment i ratten. De parametrar som föraren behöver för att kunna styra ubåten presenteras på grafiska mätare. Samtidigt ritas ubåtens position, i förhållande till föregående, upp på en slags karta på skärmen.

## 6 Tillvägagångssätt

### 6.1 Arbetet i sin helhet

Man kan dela upp arbetet i fyra olika avsnitt:

1. litteratursökning
2. montering och färdigställande av simulator utrustning
3. programutveckling
4. utprovning av lämpliga parametrar för att uppnå förbättrad styrkänsla

Arbetet har mest varit koncentrerat koncentrerat kring programutveckling och färdigställande av styrstativet.

### 6.2 Litteratursökningen

Litteratursökningen utfördes vid Lunds universitetsbibliotek under några veckor i maj 1989. Vid sökningen användes bibliotekets databas, LOLITA, samt de olika samlingar av vetenskapliga artiklar som finns i ämnet. Det finns tyvärr mycket lite material inom området människa-maskin kommunikation med inriktning mot styrning av farkoster med stor dynamik. Någon litteratur direkt kopplad till förarmiljön i en ubåt fann författarna aldrig. Däremot fanns annan litteratur som på ett eller annat sätt kan anses kopplad till problemet att framföra en farkost, till exempel haverirapporten över JAS 39 Gripens haveri våren 1989 [9]. Där konstaterades att en av de bidragande orsakerna till haveriet var styrspakens konstruktion och det faktum att de momentmotorer i styrspaken som skulle ge föraren styrkänsla inte var inkopplade.

### 6.3 Simulatorutrustningen

Före starten av vårt examensarbete hade en styrrigg beställts från verkstaden. Styrriggen fick kompletteras med en hållare för en enkel vinkelgivare; en trådlindad vridpotentiometer som beskrevs i föregående

kapitel.

Som tidigare nämnts så består styrriggen av ratt, motor, förstärkare samt potentiometer. Förstärkaren fick justeras och byglas för att ge ett moment proportionellt mot spänningen på styrsignalen. Motorn kopplades till förstärkaren och nätspänning anslöts till utrustningen. Vinkelgivarens spänningsmatning anslöts till förstärkarens spänningsaggregat.

Efter att ha provat förstärkare, motor och vinkelgivare var för sig kopplades de samman och prov med en tidigare konstruerad analog proportionell och deriverande regulator företogs. Dess uppgift var att förhindra alltför snabba ratt rörelser. Vi upptäckte då, att regulatorn gav en alldeles för hackig signal för att den skulle gå att återkoppla. Detta berodde på att när släpkontakten i potentiometern vrids från ett lindningsvarv till ett annat, så får man ett hack på signalen från vinkelgivaren, en s k 'spik'.

Eftersom en derivatadel fungerar så att den beräknar riktningskoefficienten för tangenten får man ett mycket stort värde även vid ett litet hack på signalen från vinkelgivaren. Lösningen på det här problemet var att lågpasfiltrera signalen innan vi skickar in den till PD-regulatorn. För den oinvidge kan man beskriva ett lågpasfilter som ett filter som tar bort snabbt varierande signaler som hack och brus men som släpper igenom den långsamt varierande styrsignalen.

Detta verkar ju vara ett undergörande recept, men även här är det så att det man vinner på gungorna förlorar man på karusellerna. Man får en tidsfördröjning som blir längre ju hårdare vi filtrerar. Tidsfördröjningen gör att man kan få ett återkopplat system som blir instabilt och börjar självsvänga. Trots nackdelen med tidsfördröjningen införde vi analoga lågpasfilter [10] både på ingången och utgången till PD-regulatorn [1]. Med lågpasfilterna inkopplade gick det att återkoppla signalen till den

momentgivande motorn med hyfsat resultat. Det var dock mycket svårt att ställa in regulatorn, varför beslut fattades om att implementera PD-regulatorn mjukvarumässigt senare under arbetet. De analoga lågpasfiltern orsakade lätt självsvängning av signalen, vilket gav oss anledning att implementera även dem i mjukvara.

Signalkonditioneringskretsarna utgjordes av totalt tre olika kretsar, vilka skulle anpassa de olika signalerna mellan dator och styrstativ (s39). När dessa var konstruerade sammankopplades hela utrustningen och efter en uppkoppling visade sig allting fungera som vi tänkt.

#### **6.4 Programvaruutvecklingen**

När arbetet med programutvecklingen påbörjades så fanns en början på ungefär 350 programrader. Detta "embryo" innehöll viss grafik samt den del av programmet som simulerar ubåten. Detta program har sedan utvecklats vidare och gjorts mer användarvänligt. Bland annat så har det implementerats rutiner för att på ett enkelt sätt förändra återkopplingsalternativen och dess konstanter, den grafiska presentationen har förbättrats, säkerhetssystem har införts och informationsmenyer har tillfogats programmet. Detta har lett till ett program omfattande mer än 1650 rader Turbo Pascal [11].

#### **6.5 Utprovning av återkopplingsalternativ**

Utprovningen påbörjades med att justera systemtrögheten d v s den deriverande regulator som verkade direkt på styrsignalen för att förhindra alltför snabba rattrörelser. Injusteringen av denna parameter är starkt beroende av lågpasfiltrering på insignalen, varför detta lågpasfilter justerades in samtidigt. Därefter följde utprovning av lämpliga återkopplingsalternativ. De olika alternativ som provades var:

- Den hydrodynamiska kraftpåverkan på rodret.
- Girvinkelhastigheten
- Girvinkelacceleration
- Sladdhastigheten
- Rattutslaget
- Hastighet

Alla gick att återkoppla så att man fick någon form av styrkänsla men man bör ha en rattdämpning adderad till samtliga för att få ett stabilt system. Mer information om resultatet av utprovningen finns i nästa kapitel.

## 7 Resultat

Arbetet har som tidigare nämnts skilt sig från de premisser som först gällde. I stället för att tonvikten lagts på utprovning av lämpliga återkopplingsalternativ, så har det kommit att fokuseras kring utvecklingen av ett lämpligt instrument för utprovning av dessa, vilket naturligtvis medfört att resultatet av arbetet förskjutits åt samma håll.

### 7.1 Utrustningen

På det hela taget fungerar och klarar utrustningen av det som den avsetts att göra på ett tillfredsställande sätt. Motorns snabbhet och maximala moment är fullt tillräckliga för att åstadkomma de momentförändringar som känns i styrratten. Datorn tillsammans med AD/DA-kortet ger möjlighet att uppdatera momentförändringen i styrratten med en frekvens på, för ändamålet, utmärkta 100 Hz. Programmet som används ger användaren möjlighet att enkelt ändra och byta utformning på den styrsignal, som åstadkommer momentförändringarna. Det gör utrustningen flexibel och mycket bra att använda som verktyg för att prova ut olika återkopplingar med. Givetvis måste en helt annan utrustning konstrueras för att användas i verklig tillämpning i en ubåt.

Några begränsningar sätter ändå utrustningen på den känsla som upplevs i styrratten och de är till största delen en följd av den vinkelgivare och styrratt som används.

Vid utvecklingen av styrsimulatorn har framkommit att om man vill använda någon form av deriverande ratt rörelsebegränsare bör man ha möjlighet att lågpasfiltrera signalen till denna. Annars riskerar man att få ett system som är störkänsligt. Trots att vi filtrerar signalen får vi en signal som gör att det känns hackigt när man rör ratten. Detta går inte att

komma ifrån genom hårdare digital filtrering, då man istället får för stor fördröjning vilket orsakar instabilitet och risk för självsvängning i ratten.

## 7.2 Styrkänslan

En av de första saker författarna lade märke till var att enkelhet var något eftersträvansvärt när det gäller att uppnå styrkänsla. Med detta menar vi att en återkoppling skall vara så enkel, att människan intuitivt förstår vad som avspeglas. Vidare fann vi att det är lättare att känna när kraften i ratten ökar än när den minskar på grund av att när musklerna är spända så minskar känslan för små förändringar.

## 8 Slutsatser

Under tre sommarmånader har arbete pågått med att ta fram och utprova ett simulatorverktyg för att undersöka om bättre styrkänsla i en ubåt kan uppnås. Den konstruerade simulatoren och de resultat vi erhållit vid utprovningen av alternativa återkopplingar har resulterat i de iakttagelser och slutsatser som presenteras nedan.

### 8.1 Vinkelgivaren

Den parameter som avser att motsvara den tröghet rodret har, vi har kallat den för systemtröghet, erhålls genom att signalen för rattvinkeln deriveras i en digital regulator. Signalen åstadkommer att ratten inte kan vridas med för stor hastighet och verkar även som en stabiliserande faktor på ratten. Den trådlindade potentiometer som används för att detektera rattens vinkel orsakar att denna signal får en 'spik' varje gång släpkontakten inuti potentiometern byter trådvarv. Skulle deriveringen ske direkt på signalen hade utsignalen ifrån regulatoren blivit mycket stor varje gång en 'spik' uppstått och därför olämplig att återkoppla till styrratten. Genom att ett lågpasfilter filtrerar bort 'spikarna' innan signalen når regulatoren undviks felaktiga styrsignaler men till priset av en fördröjning som otvivelaktigt påverkar styrkänslan. Att använda en trådlindad potentiometer som vinkelgivare medför alltså att ratten upplevs som antingen för ryckig eller för gungig beroende på hur stor filtrering som används. En vinkelgivare som ger en signal utan 'spikar' skulle lösa problemet (se Rekommendationer).

### 8.2 Styrratten

Som styrratt har en plastklädd järnratt med ca 30 cm diameter använts. När det gäller människans förmåga att uppfatta mindre förändringar av belastningen på armarna måste den här typen av styrratt anses som



mindre lämplig på grund av sin egen relativt stora massa. Ju större massa desto större tröghet och därmed ökar svårigheten att skapa små vinkel­förändringar av ratten med liten variation av styrsignalen, vilket har betydelse för styrkänslan.

Även om ratten gjorts i ett lättare material är det ändå inte säkert att föraren hade kunnat uppfatta små förändringar av belastningen på armarna. De stora muskelgrupper som finns i armarna klarar inte av att känna de små skillnaderna i det resulterande momentet på ratten. Dessutom handlar det till stor del att statistiskt känna dessa skillnader, något som är svårare än att känna dem dynamiskt. Det var nog inte bara utrymmesskäl som gjorde att man valde att använda ett styrhandtag i Sveriges nya stridsflygplan JAS 39 Gripen [3].

### 8.3 Återkopplingen

Efter att ha laborerat med olika parametersammansättningar och förstärkningar för den återkopplade kraften märkte vi, att det var svårt att tyda mer komplicerade kraftåterkopplingar, d v s där flera parametrar samverkade. Det var helt enkelt svårt att förstå hur ubåten svarade på ratt­rörelserna. Även giraccelerationen visade sig vara svår att reglera på och andra tidigare gjorda studier har visat detta [2].

Den parametersammansättning som vi tyckte gav oss mest information om ubåtens rörelser och som var lättast att uppfatta var girhastigheten tillsammans med ratt­dämpningen, i programmet kallad systemtrögheten. Girhastigheten kan sägas vara den parameter som lättast kan tolkas när den återkopplas och som dessutom innehåller mycket information om hur ubåten rör sig. Ratt­dämpningen bidrar till styrkänslan på det enkla sätt att det upplevs positivt att det finns en viss tröghet i ratten när den vrids. Det är dock fullt möjligt att en annan vinkel­givare och något annat att styra med kunnat innebära att ett annat alternativ blev bättre att

återkoppla.

#### 8.4 Lovande koncept ?

Tanken att förbättra styrkänslan för föraren av en ubåt, genom att återkoppla någon kombination av de parametrar som ger information om ubåtens rörelse, måste sägas vara god och ganska enkelt genomförbar. Den simulator som konstruerats och använts för att prova ut vilka parametrar som skall användas och till vilken grad styrkänslan kan förbättras har i sin nuvarande utformning de begränsningar som nämnts ovan. Det, tillsammans med otillräcklig utprovning, har inneburit att det är svårt för oss att ge mer än de indikationer som beskrivits men vi kan istället ge förslag på förbättringar och hur det fortsatta arbetet skulle kunna fortgå (se Rekommendationer).

## 9 Rekommendationer

De förbättringar som kan göras på simulatorutrustningen och några förslag till vad som kan göras ytterligare för att förbättra våra uppnådda resultat presenteras nedan.

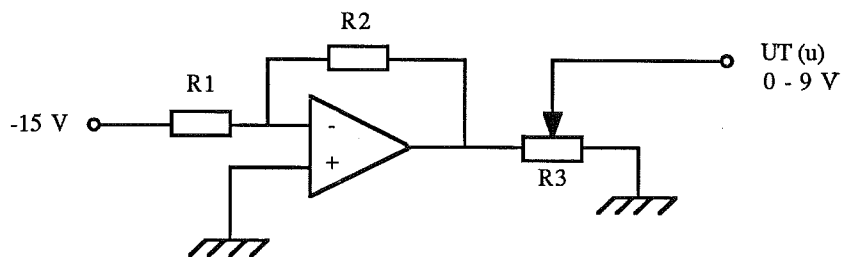
- Istället för att använda en trådlindad potentiometer som vinkelgivare bör en vinkelgivare med jämnare utsignal användas för att undvika för lång tidsfördröjning. Bästa resultatet fås om en digital vinkelgivare, som kan representera varje vinkel med en egen unik kod, t. ex. en s k 'Grey- kodad' vinkelgivare [13], kan användas.
- Överföringen mellan ratt och motor kan göras helt stel och inte som nu vara elastisk.
- Rattens fäste till sin axel kan göras med mindre glapp.
- Ratten kan bytas ut till en ratt med mindre massa eller någon form av styrhandtag för att förbättra möjligheten för föraren att känna små momentförändringar.
- Ett förartest eller liknande kan utformas för en grupp försökspersoner vilket kunde resultera i en kvalitativ- och kvantitativ bedömning av olika återkopplingsalternativ. Testet kunde utgöras av någon form av slalombana på sjökortet på skärmen som förarna måste styra igenom inom en viss tidsrymd och med ett minimum av påkörda hinder.
- Genom att använda en bildskärm med tre-dimensionell grafik som visar ubåtens rörelser i girled och i vertikalled, och/eller en estimator

som estimerar ubåtens kurs kontinuerligt och visar hur ubåten kommer att bete sig i framtiden skulle styrkänslan troligtvis förbättras. Genom att förändra de grafiska rutinerna i simulatorprogrammet 'Styrsim' kan även detta utprovas med den nu befintliga simulatorn.

## Appendix A

### Signalkonditioneringskretsar

De analoga kretsar som konstruerats och placerats i en svart låda märkt 'Gränssnitt' och som används för att anpassa signaler mellan styrstativ och dator presenteras nedan. Två stycken filter finns även tillgängliga men används inte eftersom vi filtrerar digitalt istället.



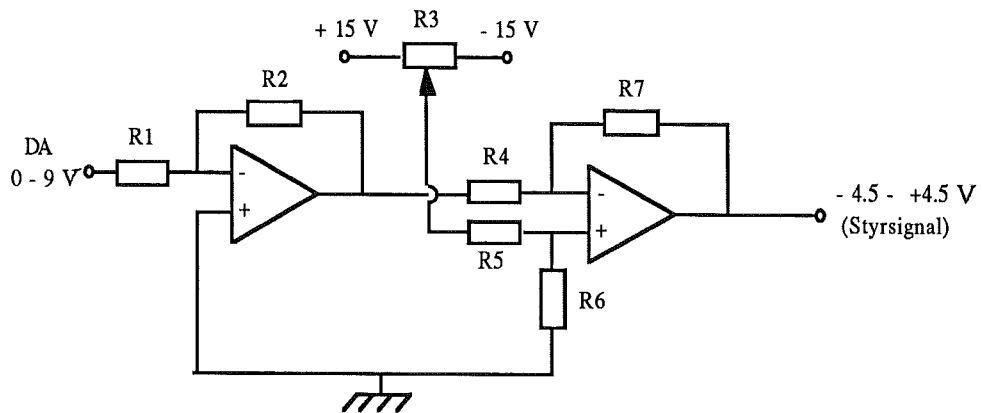
Figur A.1 Kretsen genererar signalen för hastighet (u)

### Komponentvärden

$$R1 = 100 \text{ k}\Omega$$

$$R2 = 60 \text{ k}\Omega$$

$$R3 = 20 \text{ k}\Omega$$



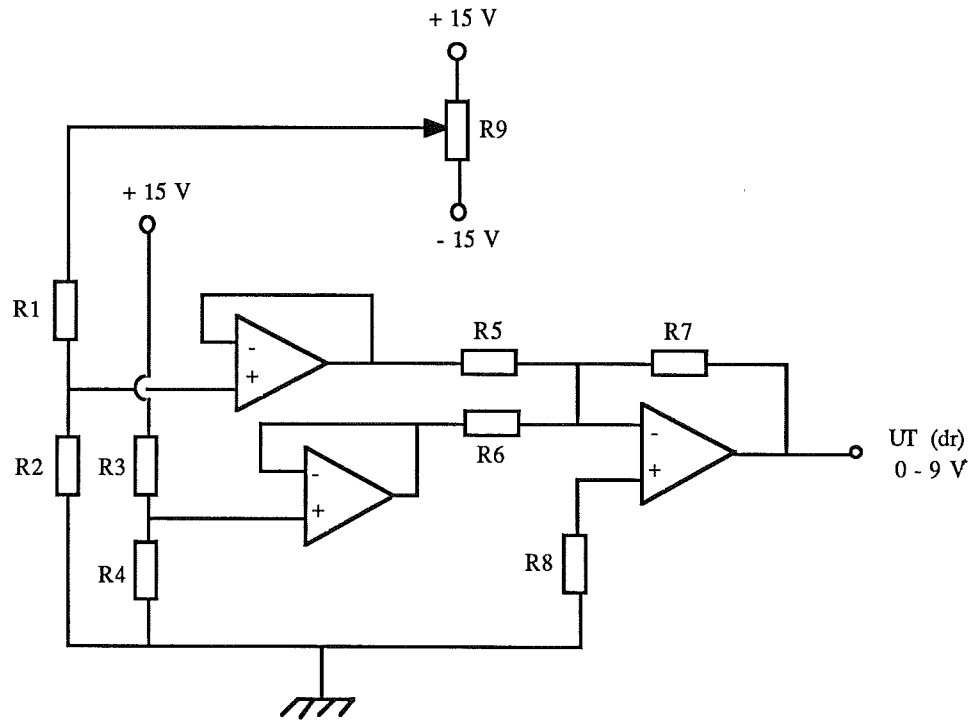
**Figur A.2** Kretsen transformerar styrsignalen ifrån datorn till att passa förstärkaren och motorn.

Komponentvärden

$$R1 = R2 = 8.2 \text{ k}\Omega$$

$$R3 = 0 - 50 \text{ k}\Omega$$

$$R4 = R5 = R6 = R7 = 10 \text{ k}\Omega$$



Figur A.3 Kretsen transformerar signalen ifrån vinkelgivaren till att passa AD-omvandlaren

#### Komponentvärden

$$R1 = 13 \text{ k}\Omega$$

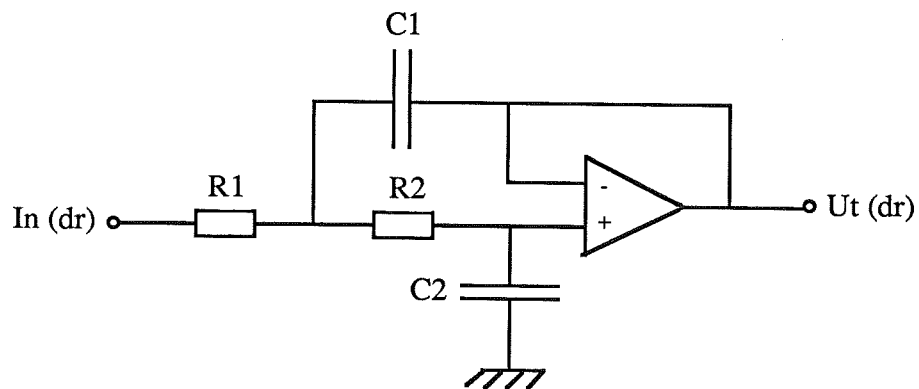
$$R2 = 120 \text{ k}\Omega$$

$$R3 = 100 \text{ k}\Omega$$

$$R4 = 47 \text{ k}\Omega$$

$$R5 = R6 = R7 = 82 \text{ k}\Omega$$

$$R8 = 27 \text{ k}\Omega$$



Figur A.4 Kretsen lågpassfiltrerar insignalen

Komponentvärden

$$f_0 = 20 \text{ kHz}$$

$$C1 = 135 \text{ nF}$$

$$C2 = 100 \text{ nF}$$

$$R1 = R2 = 68 \text{ k}\Omega$$

$$f_0 = 30 \text{ kHz}$$

$$C1 = 90 \text{ nF}$$

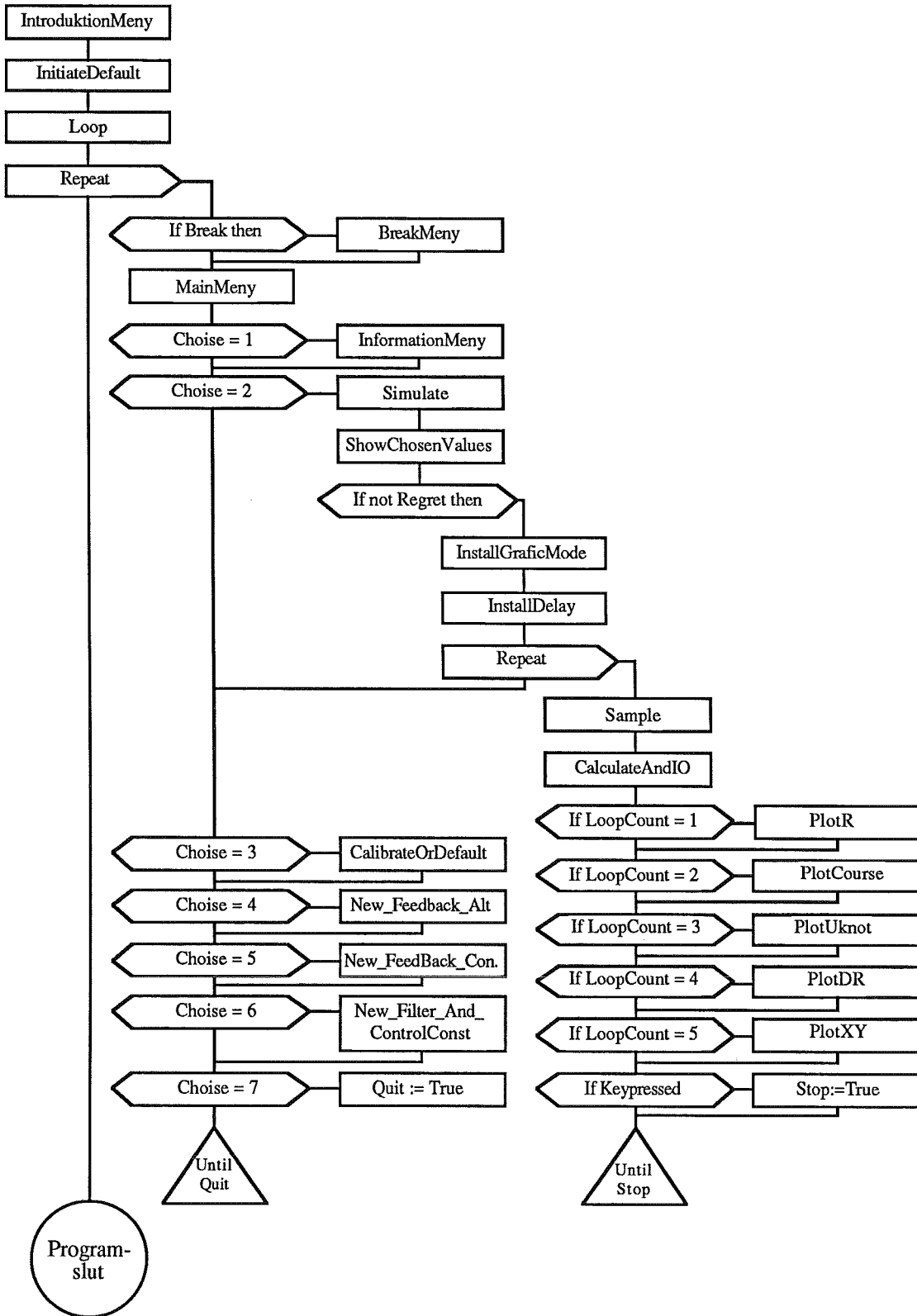
$$C2 = 68 \text{ nF}$$

$$R1 = R2 = 68 \text{ k}\Omega$$



# Appendix B

## Programflödesschema



**Beskrivning av samtliga procedurer och funktioner**

Programmet StyrSim är skrivet i Turbo Pascal [11] och består av ungefär 1650 programrader. Totalt ingår det 36 procedurer och 8 funktioner. Nedan följer en kort beskrivning av samtliga procedurer och funktioner.

**Procedurer****Ram**

Ritar en ram kring samtliga menyer.

**Noise**

Åstadkommer ljud med frekvensen Fr under tiden Ms.

Inparametrar: Fr: Fr är av typen integer.

Ms: Ms är av typen integer.

**FailSound**

Genererar en kort ljudslinga med tonhöjd beroende av Alt.

Inparametrar: Alt: Alt är av typen real.

**CrashSound**

Genererar ett kort buller.

**FunSound**

Genererar ett ljud vars frekvens stegas upp.

**IntroductionMeny**

Skriver ut en kort introduktion på skärmen när man startar upp programmet.

**InformationMeny**

Beskriver försöksutrustningen och de parametrar som man kan förändra i programmet.

**ShowChosenValues**

Visar valt återkopplingsalternativ och valda parametrar varje gång man skall in och simulera.

**BreakMeny**

Förklara varför simuleringen har avbrutits.

**ReadInputReal**

Läser ett inmatat reellt värde från skärmen samt kontrollerar att det är ett giltigt värde, om värdet är ogiltigt så skrivs ett felmeddelande ut på skärmen.

Inparametrar	Min:	Det minsta värde den inlästa parametern får ha. Min är av typen integer.
	Max:	Det största värde den inlästa parametern får ha. Max är av typen integer.
	Scale:	En skalningsfaktor som gör att de inlästa parametrarna hamnar i rätt intervall. Scale är av typen real.
Utparameter	Input:	Det värde vi läst in och multiplicerat med Scale som vi skickar till den anropande proceduren.

**ReadInputInteger**

Läser ett inmatat heltals värde från skärmen samt kontrollerar att det är ett giltigt värde, om värdet är ogiltigt så skrivs ett felmeddelande ut på skärmen.

Inparametrar	Min:	Det minsta värde den inlästa parametern får ha. Min är av typen integer.
	Max:	Det största värde den inlästa parametern får ha. Max är av typen integer.
Utparameter	Input:	Det värde vi läst in som vi skickar till den anropande proceduren.

**NewFeedBaConst**

Läser från tangentbordet in nya värden på konstanterna som används vid beräkning av återkopplingskraften.

**New\_FeedBack\_Alt**

Här väljer man det återkopplingsalternativ man vill använda vid simuleringen.

**New\_Filter\_And\_ControlConst**

Används för att ändra på konstanterna till lågpasfilter och PD-regulator samt sampel- frekvensen.

**Write\_A\_And\_B\_Matrix**

Skapar en fil för den iterativa formelns A och B-matriser.

**NewMatrix**

Läser in nya A och B-matriser om hastigheten u har förändrats mer än en knop.

**Read\_A\_And\_B\_Matrix**

Läser in en tidigare skapad fil med A och B-matriser för olika hastigheter, för att användas vid beräkning av r och v.

## **DA**

Skriver vår beräknade kraft på utgången på vårt ADDA-kort till vår momentgivande motor.

## **InitiateDefault**

Initierar defaultvärden på alla återkopplings- och filterparametrar samt sampelfrekvens.

## **CheckPowerOn**

Kontrollerar om spänningen är påslagen på styrstativet, om så inte är fallet så kommer en uppmaning om att åtgärda detta att komma upp på skärmen.

## **Calibrate**

Kalibrerar ratt och gasreglage.

## **Calibrate\_Or\_Default**

Visar en meny där man kan välja att kalibrera om ratt och gasreglage eller få tillbaka de defaultvärden som finns inlagda i programmet.

## **CheckSteeringWheelPos**

Återställer den återkopplade kraften till noll och ser till att ratten åter står i mittläget innan simuleringen startar. Om inte ratten står i horisontalläge så kommer ett meddelande att skrivas på skärmen så att det åtgärdas.

## **InstallGraficMode**

Introducerar den grafik vi använder under simuleringen.

## **InstallDelay**

Installerar ett klockavbrott med frekvensen SampleFr med hjälp av timer 0.

### **Back**

Återställer interrupt-, timer-, och skärmtillstånd till normalt läge samt lägger ut kraften 0 till ratten (dvs 2048).

### **PlotR**

Ritar upp den nya värdet på girvinkelhastighetsmätaren i grader per sekund.

### **PlotDR**

Ritar upp den nya värdet på rodervinkelmätaren i grader.

### **PlotUknot**

Ritar upp den nya värdet på U-båtens hastighet i knop.

### **PlotCourse**

Ritar upp den nya värdet på U-båtens kurs på en kompass.

### **PlotXY**

Ritar upp den nya värdet på U-båtens läge på vårt sjökort.

## **Funktioner**

### **AD**

Läser in ett värde mellan 0 och 4096 från kanalen Channel på AD-omvandlaren. Observera att vårt AD/DA-kortet inte hinner med och att vi därför måste göra en additon 4 gånger vid varje inläsning.

Inparameter      Channel: Den kanal på ADDA-kortet vi ska läsa ifrån.

Channel är av typen integer.

**Lowpass**

Är ett första ordningens lågpasfilter som filtrerar insignalen. Lowpass är en real.

Inparameter      Value:      Signalen som skall filtreras. Value är ett heltal.

**Lowpass2**

Är ett första ordningens lågpasfilter som filtrerar utsignalen. Lowpass2 är en real.

Inparameter      Value:      Signalen som skall filtreras. Value är ett heltal.

**Sign**

Ger +1 respektive -1 beroende på tecknet på rattvinkeln. Sign är ett heltal.

**DregDR**

Deriverande regulator som verkar på differensen mellan gammal och ny rattvinkel dr. Det är denna procedur som genererar det vi kallar systemtröghet. DregDR är ett decimaltal.

**DregR**

Deriverande regulator som verkar på differensen mellan gammal och ny girvinkel- hastighet. Man får därigenom girvinkelaccelerationen som vi använder i en typ av åter- koppling. DregR är ett decimaltal.

**Feedback**

Beräknar den signal som vi använder till att styra den momentgivande motorn på ratten.

Proceduren kontrollerar att signalen ligger inom det område som DA-omvandlaren klarar (0 - 4096). Man kontrollerar även att signalen inte förändras för mycket mellan två närliggande sampel, om så skulle vara

fallet så lägger man ut 2048 på DA-omvandlaren och två logiska variabler Stop och Break blir sanna och simuleringen avbryts. Feedback är ett heltal.

### **SecondOrderLowpass**

Ett andra ordningens lågpasfilter som för tillfället inte används men som kan kopplas in enligt anvisningar givna i användarmanualen. Filtret kan användas för att filtrera antingen insignalen eller utsignalen. SecondOrderLowpass är ett decimaltal.

Inparametrar      Value:      Det heltal som skall filtreras.

### **CalculateAndIO**

Läser in rodervinkel och hastighet från AD-omvandlaren och beräknar därefter hastigheten i sidled  $v$  samt girvinkelhastigheten  $r$ . Proceduren skickar även ut återkopplingssignalen Feedback till motorn.

### **Sample**

Beräknar ny position och kurs vidare hanteras även valet av vilken plottprocedur som skall göras vid varje sampel. Här kontrolleras även om någon tangent har tryckts ner, om detta varit fallet så sätts Stop till sann och simuleringen avbryts.

### **Simulate**

Är den procedur som anropas när vi vill simulera vald kraftåterkoppling för att känna hur den känns i ratten. Här installerar vi grafisk mod och klockavbrott med hjälp av InstallGraficMode och InstallDelay.

### **HuvudMeny**

Detta är den meny som ger användaren möjlighet att välja vad programmet skall utföra. Valet kontrolleras så att det är giltigt, om så inte är fallet så skrivs ett felmeddelande ut och användaren får välja igen.



## **Loop**

Det är i denna slinga vi ligger under tiden vi kör programmet och här ligger man enda tills dess att man avslutar programkörningen.

# **Programlistning**

av programmet

## **Styrsim**

Programlistningen innehåller kommentarer och även markeringar där föreslagna ändringar av programmet kan göras.

```
PROGRAM StyrSimulering;
(* Observera att programmet räknar med meter/s och vinkeln radianer *)
(* Roderutslaget multipliceras en faktor fyra för att erhålla dr *)
{$N+}

USES CRT,GRAPH,DBPLUS;

TYPE
  MatrixArray=ARRAY[1..6,0..20] OF REAL;
  MatrIndTyp =ARRAY[1..6] OF CHAR;

CONST
  Map          : ViewPortType =
    (x1: 20; y1: 20; x2: 320; y2: 460; Clip : ClipOn);
  Compass      : ViewPortType =
    (x1: 470; y1: 108; x2: 630; y2: 268; Clip : ClipOn);
  SpeedometerR : ViewPortType =
    (x1: 340; y1: 380; x2: 630; y2: 462; Clip : ClipOn);
  AnglometerDR : ViewPortType =
    (x1: 340; y1: 283; x2: 630; y2: 365; Clip : ClipOn);
  SpeedometerU : ViewPortType =
    (x1: 340; y1: 108; x2: 422; y2: 268; Clip : ClipOn);
  StopDisplay  : ViewPortType =
    (x1: 440; y1: 20; x2: 530; y2: 80; Clip : ClipOn);

  MatrInd :MatrIndTyp =('a','b','c','d','e','f');
  IntrMask      = $21;
  TimerCom      = $43;
  Kanal0        = $40;
  DrChannel     = 0;
  uChannel      = 1;
  ClFr          = 1.19E6;

VAR
  uKnot,OldUKnot,OldUP1,I,J,OldMask,WeelValue,
  XX,YY,OldRP1,RP1,Choise,XKoord,YKoord,FeedBackChoise,
  LoopCount,GraphDriver,GraphMode,OldPLUKnot,NConstR,
  NConst,s,DRP1,OldDRP1,UP1,SampleFr          : INTEGER;

  a,b,c,d,e,f,Ts,dr,drOffset,drMultFact,Delta,DConst,
  u,uMultFact,r,v,psi,x,y,cospsi,LowPassConst,K1,
  sinpsi,Alfa,Beta,LowPassConst2,OldFiltValue,TdConst,
  OldFiltValue2,OldValue,OldValueR,OldHelpVar,K6,K7,
  TdConstR,DConstR,BetaR,OldHelpVarR,Ordn2LowPassConst,
  OldY1Filt,OldY2Filt,K2,K3,K4,K5,K8,K9,K10,K11,K12,K13,
  K14,OldFeedBackValue          : REAL;

  A_B          :MatrixArray;
  SpeedPlot    :ARRAY[0..20] OF STRING
;
  fi          :FILE OF REAL;
  Quit,Stop,Calibrated,Regret,Moved,Break          :BOOLEAN;

(*****)
```

```
PROCEDURE Ram;  
(* Ritar en ram kring alla menyer *)  
BEGIN  
  ClrScr;  
  HighVideo;  
  TextColor(14);  
  Write('=====');  
  Write('|| EXAMENSARBETE                                KOCKUMS MA');  
  RINE AB ||');  
  Write('|| STYRKÄNSLA                                VAPENSEKTI');  
  ONEN ||');  
  Write('=====');  
  Write('||');  
  FOR I:=1 TO 19 DO  
    Write('||');  
  Write('=====');  
  Write('||');  
  TextColor(7);  
  HighVideo;  
END; (* Ram *)
```

(\*\*\*\*\*)

```
PROCEDURE Noise ( Fr,Ms :INTEGER);  
(* Åstadkommer ljud *)  
BEGIN  
  Sound(Fr);  
  DELAY(Ms);  
  NoSound;  
END; (* Noise *)
```

(\*\*\*\*\*)

```
PROCEDURE FailSound ( Alt :REAL);  
(* Ödesljud *)  
BEGIN  
  Noise(ROUND(440/Alt),300);  
  Noise(0,20);  
  Noise(ROUND(440/Alt),300);  
  Noise(0,20);  
  Noise(ROUND(440/Alt),300);  
  Noise(0,20);  
  Noise(ROUND(350/Alt),900);  
END; (* FailSound *)
```

(\*\*\*\*\*)

```
PROCEDURE CrashSound;  
BEGIN  
  FOR I:=1 TO 1000 DO  
    BEGIN  
      Noise(Round(Random*40),1);  
      Noise(Round(Random*45),1);  
    END;  
END;
```

END; (\* CrashSound \*)

(\*\*\*\*\*)

PROCEDURE FunSound;  
(\* Skapar introduktionsljudet \*)

BEGIN  
  I:=200;  
  J:=6;  
  REPEAT  
    REPEAT  
      Noise(I,J);  
      I:=I+30;  
    UNTIL I>4000;  
    REPEAT  
      Noise(I,J);  
      I:=I-30;  
    UNTIL I<200;  
    J:=J-1;  
  UNTIL J=1;  
END; (\* FunSound \*)

(\*\*\*\*\*)

PROCEDURE IntroductionMeny;  
(\* Ger kort förklaring av programmet \*)

BEGIN  
  HighVideo;  
  ClrScr;  
  Ram;  
  GoToXY(15,6);  
  Write('VÄLKOMMEN TILL MICAEL OCH FREDRIKS UBÅTS-SIMULATOR !');  
  GoToXY(10,9);  
  Write('Detta program simulerar en ubåts manövrar i girled. Simulatorn');  
  GoToXY(10,10);  
  Write('mäter rattvinkel och hastighet samt beräknar därefter ubåtens');  
  GoToXY(10,11);  
  Write('rörelse och ritar ut denna på skärmen. Även en kraft-term');  
  GoToXY(10,12);  
  Write('beräknas och skickas ut till en momentmotor som styr kraften');  
  GoToXY(10,13);  
  Write('som känns i ratten.');

  GoToXY(10,14);  
  Write('För att utprova styrkänsla har ni flera olika alternativa');  
  GoToXY(10,15);  
  Write('krafttermer att välja mellan. Ni har även möjlighet att ändra ');  
  GoToXY(10,16);  
  Write('filtreringen av in- och utsignal samt sampelfrekvensen. ');  
  GoToXY(33,19);  
  SetCursorOff;  
  Write('TRYCK RETURN !');

  FunSound;  
  REPEAT  
  UNTIL ReadKey=chr(13);  
  RestoreCursor;  
END; (\* IntroductionMeny \*)

(\*\*\*\*\*)

PROCEDURE InformationMeny;

(\* Förklarar använda parametrar och ritar blockschema \*)

BEGIN

SetCursorOff;

ClrScr;

Ram;

GoToXY(34,6);

WriteLn('ANVÄNDAR GUIDE !');

GoToXY(6,9);

WriteLn('Användar guiden är en kort information om de saker man bör känna ');

GoToXY(6,10);

WriteLn('till om programmet innan man startar en simulering.');

GoToXY(6,12);

WriteLn('Simulatorn består av datorn som har en 386-processor, en 387-matematik-');

GoToXY(6,13);

WriteLn('processor samt ett 12-bitars AD/DA-kort för kommunikation med en ratt en.');

GoToXY(6,14);

WriteLn('Datorn är kopplad till rattan via signalkonditioneringskretsar.');

GoToXY(6,17);

WriteLn(' ');

GoToXY(6,18);

WriteLn(' | «—————u,dr————— | ');

GoToXY(6,19);

WriteLn(' | DATOR | RATT+MOTOR | ');

GoToXY(6,20);

WriteLn(' | —————kraft————— » | ');

GoToXY(6,21);

WriteLn(' | ');

GoToXY(34,23);

WriteLn('TRYCK PÅ RETURN !');

REPEAT

UNTIL ReadKey=chr(13);

ClrScr;

Ram;

GoToXY(6,7);

WriteLn('För att kunna välja återkopplings-alternativ och sätta de konstanter');

;

GoToXY(6,8);

WriteLn('som används i signalbehandlingen ges nedan befogade förklaringar.');

GoToXY(6,11);

WriteLn('Parametrar: Intervall:');

GoToXY(6,14);

WriteLn('u = hastighet framåt ( 0 - 10,3 )');

GoToXY(6,15);

WriteLn('v = hastighet sidled ( -1.5 - 1.5 )');

GoToXY(6,16);

WriteLn('r = girhastighet ( -0.086 - 0.086 )');

GoToXY(6,17);

WriteLn('dr = rattutslag ( -2.1 - 2.1 )');

GoToXY(6,18);

WriteLn('Delta = 0.25\*dr ( -0.5 - 0.5 )');

GoToXY(6,19);

WriteLn('A = (0.25\*dr - v/u)<sup>2</sup> ( 0.0 - ± 0.5 )');

GoToXY(6,20);

```

WriteLn('DregDr = systemtrögheten      d(dr)/dt  = orsakar tröghet i ratten');

GoToXY(6,21);
WriteLn('DregR  = giraccelerationen   d(r)/dt');
GoToXY(34,23);
WriteLn('TRYCK PÅ RETURN !');
REPEAT
UNTIL ReadKey=chr(13);
ClrScr;
Ram;
GoToXY(30,6);
WriteLn('BLOCKSCHEMA I DETALJ !');
GoToXY(3,8);
WriteLn(' _____');
');
GoToXY(3,9);
WriteLn(' | _____ | _____ | _____ | _____ | _____');
');
GoToXY(3,10);
WriteLn(' | SIMULATOR |<| FILTER |<| A/D |<| -u,dr |<| S |<| RATT |<|');
');
GoToXY(3,11);
WriteLn(' | _____ | _____ | _____ | | I | _____ |');
');
GoToXY(3,12);
WriteLn(' | _____ |>| _____ |>| _____ | | G | _____ |');
');
GoToXY(3,13);
WriteLn(' | _____ | _____ | _____ | | N. | _____ |');
');
GoToXY(3,14);
WriteLn(' | d(r)/dt | | _____ | d(dr)/dt | | _____ | MOTO');
R | _____ |');
GoToXY(3,15);
WriteLn(' | _____ | _____ | _____ | | K | _____ |');
');
GoToXY(3,16);
WriteLn(' | _____ |<| _____ |<| _____ | | O | _____ |');
');
GoToXY(3,17);
WriteLn(' | _____ | _____ | _____ | | N | _____ |');
');
GoToXY(3,18);
WriteLn(' | BERÄKN.AV KRAFT |>| FILTER |>| D/A |>| kraft |>| D. |>| FÖRSTÄRKARE |>|');
');
GoToXY(3,19);
WriteLn(' | _____ | _____ | _____ | | _____ | _____ |');
');
GoToXY(3,20);
WriteLn(' _____');
');
GoToXY(34,23);
WriteLn('TRYCK PÅ RETURN !');
REPEAT
UNTIL ReadKey=chr(13);
ClrScr;
Ram;
GoToXY(30,6);
WriteLn('OBS !! KOM IHÅG ATT :');

```

```
GoToXY(6,8);
WriteLn('1. Inte vrida ratten med onödigt stark kraft, dvs TA INTE I !');
GoToXY(6,10);
WriteLn('2. Ändringar av parametrar bör ske i små steg. Utgå alltid ifrån');
GoToXY(6,11);
WriteLn('  default-värdena, som lätt sätts genom val i huvudmenyn. ');
GoToXY(6,13);
WriteLn('3. Omkalibrering behövs endast om någon mätare överstiger max-');
GoToXY(6,14);
WriteLn('  markeringen under simuleringens gång.  ');
GoToXY(6,16);
WriteLn('4. Systemtrögheten, kallad DregDR, bör alltid finnas med i den ');
GoToXY(6,17);
WriteLn('  återkopplade kraften av stabilitets-skäl. ');
GoToXY(6,19);
WriteLn('5. Den återkopplade kraften skall ligga mellan 0 och 4096. Värdet');
GoToXY(6,20);
WriteLn('  2048 motsvarar således kraften noll. ');
GoToXY(34,23);
WriteLn('TRYCK PÅ RETURN !');
REPEAT
UNTIL ReadKey=chr(13);
RestoreCursor;
END; (* InformationMeny *)

(*****

PROCEDURE ShowChosenValues;
(* Visar default-värden och återkopplingsalternativ *)
VAR Ch :CHAR;
BEGIN
  SetCursorOff;
  ClrScr;
  Regret:=False;
  Ram;
  GoToXY(10,6);
  Write('I SIMULERINGEN KOMMER FÖLJANDE PARAMETERVÄRDEN ATT ANVÄNDAS :');
  GoToXY(10,9);
  Write('Sampelfrekvens (Hz):                ',SampleFr:1);
  GoToXY(10,11);
  Write('Ingångens lågpasfilter:                ALFA=',LowPassConst:5:2);
  GoToXY(10,13);
  Write('Utgångens lågpasfilter:                ALFA=',LowPassConst2:5:2);
  GoToXY(10,16);
  Write('Återkopplingsalternativ:                ');
  GoToXY(10,18);
  CASE FeedBackChoise OF
    1: Write('Roderkraft+Giracceleration+Girvinkelhastighet+Systemtröghet');
    2: Write('Giracceleration plus systemtröghet');
    3: Write('Girvinkelhastighet plus systemtröghet');
    4: Write('Roderkraft plus systemtröghet');
    5: Write('Proportionell återkoppling med hastighetsberoende plus systemtrögh
et');
    6: Write('Stega');
  END;
  GoToXY(18,21);
  Write('Om ändra tryck på R, annars valfri tangent !');
  IF UpCase(ReadKey)='R' THEN
```



```

    Regret:=True;
  RestoreCursor;
END; (* ShowChosenValues *)

```

(\*\*\*\*\*)

```

PROCEDURE BreakMeny;
(* Förklarar varför programmet avbrutits *)
BEGIN

```

```

  IF Break THEN
  BEGIN
    SetCursorOff;
    ClrScr;
    Ram;
    GoToXY(7,11);
    TextBackground(12);
    TextColor(14);
    Write('┌───────────────────────────────────────────────────────────────────────────────────┐')
  ;
    GoToXY(7,12);
    Write('|| SIMULERINGEN AVBRÖTS P G A FÖR STORA VARIATIONER I RATTKRAFTEN !||')
  ;
    GoToXY(7,13);
    Write('|| KONTROLLERA OM LYSDIODEN PÅ FÖRSTÄRKARKORTET LYSER ! LYSER DEN: ||')
  ;
    GoToXY(7,14);
    Write('|| SLÅ AV SPÄNNINGEN TILLS DEN SLOCKNAT, SEN SLÅ PÅ DEN IGEN !           ||')
  ;
    GoToXY(7,15);
    Write('└───────────────────────────────────────────────────────────────────────────────────┘')
  ;
    FailSound(0.95);
    FailSound(0.7);
    DELAY(7000);
    Break:=FALSE;
    RestoreCursor;
    NormVideo;
    HighVideo;
  END;

```

```

END; (* BreakMeny *)

```

(\*\*\*\*\*)

```

PROCEDURE ReadInputReal (Min,Max :INTEGER; Scale :REAL; Var Input :REAL);
(* Läser inmatat värde och ger felmeddelande om ogiltigt värde matats in *)

```

```

VAR
  ErrorPos,CursorX,CursorY      :INTEGER;
  InputNumber                    :STRING;
BEGIN
  CursorX:=WhereX;
  CursorY:=WhereY;
  ReadLn(InputNumber);
  IF LENGTH(InputNumber) <> 0 THEN
  BEGIN
    Val(InputNumber,Input,ErrorPos);
    WHILE (Input < Min) OR (Input > Max) OR (ErrorPos <> 0) DO
    BEGIN

```

```
    SetCursorOff;
    TextBackground(12);
    TextColor(14);
    GoToXY(CursorX,CursorY);
    Write('FELAKTIGT VÄRDE !');
    DELAY(2000);
    GoToXY(CursorX,CursorY);
    TextBackground(0);
    TextColor(7);
    HighVideo;
    Write('          ');
    GoToXY(CursorX,CursorY);
    RestoreCursor;
    ReadLn(InputNumber);
    Val(InputNumber,Input,ErrorPos);
  END;
  Input:=Input*Scale;
END; (* ReadInputReal *)
```

(\*\*\*\*\*)

```
PROCEDURE ReadInputInteger (Min,Max :INTEGER; Var Input :INTEGER);
(* Läser inmatat värde och ger felmeddelande om ogiltigt värde matats in *)
VAR
  ErrorPos,CursorX,CursorY      :INTEGER;
  InputNumber                    :STRING;
BEGIN
  CursorX:=WhereX;
  CursorY:=WhereY;
  ReadLn(InputNumber);
  IF LENGTH(InputNumber) <> 0 THEN
  BEGIN
    Val(InputNumber,Input,ErrorPos);
    WHILE (Input < Min) OR (Input > Max) DO
    BEGIN
      SetCursorOff;
      TextBackground(12);
      TextColor(14);
      GoToXY(CursorX,CursorY);
      Write('FELAKTIGT VÄRDE !');
      DELAY(2000);
      GoToXY(CursorX,CursorY);
      TextBackground(0);
      TextColor(7);
      HighVideo;
      Write('          ');
      GoToXY(CursorX,CursorY);
      RestoreCursor;
      ReadLn(InputNumber);
      Val(InputNumber,Input,ErrorPos);
    END;
  END;
END; (* ReadInputInteger *)
```

(\*\*\*\*\*)

```
PROCEDURE New_FeedBack_Const;
(* Ändrar värdet på konstanterna som beräknar återkopplings- *)
(* kraften till Da-omvandlaren *)
BEGIN
  ClrScr;
  Ram;
  GoToXY(25,6);
  Write('VÄLJ ÅTERKOPPLINGS-KONSTANTER!');
  GoToXY(7,8);
  CASE FeedBackChoise OF
    1: BEGIN
      Write('Återkopplad signal= u*A*(K1*Sign+K2*A*Delta)-DregR+K3*r+DregDR+2
048');
      GoToXY(10,11);
      Write('K1 (0.0 - 10.0):      ',K1/100:8:0,' ');
      ReadInputReal(0,10,100,K1);
      GoToXY(10,13);
      Write('K2 (0.0 - 10.0):      ',K2/6000:8:0,' ');
      ReadInputReal(0,10,6000,K2);
      GoToXY(10,15);
      Write('K3 (0.0 - 10.0):      ',K3/4000:8:0,' ');
      ReadInputReal(0,10,4000,K3);
    END;
    2: BEGIN
      SetCursorOff;
      GoToXY(10,10);
      Write(' Parametrarna ändras endast i huvudmenyns val nr 6 !');
      GoToXY(34,16);
      Write('TRYCK RETURN !');
      REPEAT
      UNTIL ReadKey=chr(13);
      RestoreCursor;
    END;
    3: BEGIN
      Write('Återkopplad signal= DregDR+K4*r+2048');
      GoToXY(10,11);
      Write('K4 (0.0 - 10.0):      ',K4/4000:8:0,' ');
      ReadInputReal(0,10,4000,K4);
    END;
    4: BEGIN
      Write('Återkopplad signal= DregDR+u*A*(K5*Sign+K6*Delta*A)+2048');
      GoToXY(10,11);
      Write('K5 (0.0 - 10.0):      ',K5/100:8:0,' ');
      ReadInputReal(0,10,100,K5);
      GoToXY(10,13);
      Write('K6 (0.0 - 10.0):      ',K6/6000:8:0,' ');
      ReadInputReal(0,10,6000,K6);
    END;
    5: BEGIN
      Write('Återkopplad signal= Sign*(K7*Ln(K8*abs(dr)+1)*Ln(K9*u+1))+DregDR
+2048');
      GoToXY(10,11);
      Write('K7 (0.0 - 10.0):      ',K7/200:8:0,' ');
      ReadInputReal(0,10,200,K7);
      GoToXY(10,13);
      Write('K8 (0.0 - 10.0):      ',K8/0.6:8:1,' ');
      ReadInputReal(0,10,0.6,K8);
      GoToXY(10,15);
      Write('K9 (0.0 - 10.0):      ',K9/0.02:8:1,' ');
```

```
        ReadInputReal(0,10,0.02,K9);
        GoToXY(10,18);
        Write('Systemtrögheten ändras i huvudmenyns val nr 6 !');
    END;
{ 6: BEGIN
    Write('Återkopplad signal= ...HÄR SKRIVER NI IN EGET UTTRYCK OM NI VILL
...');
    GoToXY(10,11);
    Write('K10 (0.0 - 10.0):      ',K10/SKALFAKTOR SKRIVS HÄR!:8:1,' ');

    ReadInputReal(0,10,SKALFAKTOR SKRIVS HÄR!,K10);
    GoToXY(10,13);
    Write('K11 (0.0 - 10.0):      ',K11/SKALFAKTOR SKRIVS HÄR!:8:1,' ');

    ReadInputReal(0,10,SKALFAKTOR SKRIVS HÄR!,K11);
    GoToXY(10,15);
    Write('K12 (0.0 - 10.0):      ',K12/SKALFAKTOR SKRIVS HÄR!:8:1,' ');

    ReadInputReal(0,10,SKALFAKTOR SKRIVS HÄR!,K12);
    GoToXY(10,15);
    Write('K13 (0.0 - 10.0):      ',K13/SKALFAKTOR SKRIVS HÄR!:8:1,' ');

    ReadInputReal(0,10,SKALFAKTOR SKRIVS HÄR!,K13);
    GoToXY(10,15);
    Write('K14 (0.0 - 10.0):      ',K14/SKALFAKTOR SKRIVS HÄR!:8:1,' ');

    ReadInputReal(0,10,SKALFAKTOR SKRIVS HÄR!,K14);
    END; }
END; (* New_FeedBack_Const *)

(*****)
```

```
PROCEDURE New_FeedBack_Alt;
(* Väljer den sorts återkoppling man vill ha *)
VAR
    CursorX,CursorY,ErrorPos    :INTEGER;
    InputNumber                  :STRING;
BEGIN
    ClrScr;
    Ram;
    GoToXY(15,6);
    WriteLn('VÄLJ ÅTERKOPPLINGSALTERNATIV! FÖREGÅENDE VAL: ',FeedBackChoise);
    GoToXY(10,9);
    WriteLn('1...Roderkraft+Giracceleration+Girvinkelhastighet+Systemtröghet');
    GoToXY(10,11);
    WriteLn('2...Giracceleration plus systemtröghet');
    GoToXY(10,13);
    WriteLn('3...Girvinkelhastighet plus systemtröghet');
    GoToXY(10,15);
    WriteLn('4...Roderkraft plus systemtröghet');
    GoToXY(10,17);
    WriteLn('5...Proportionell återkoppling med hastighetsberoende');
    GoToXY(10,18);
    WriteLn('      plus systemtröghet');
```

```
(***** OM NI SKAPAR ERT EGET UTTRCK FÖR DEN ÅTERKOPPLADE KRAFTEN TAR NI *****)
(***** BORT KOMMENTARPARENTESERNA HÄREFTER OCH ÄNDRAR ÖVRE FELGRÄNS *****)
```

```
(***** TILL 6 ISTÄLLET FÖR 5 I WHILE SATSEN NEDAN.          *****)

(*GoToXY(10,20);          *)
(*WriteLn('6...Stega');  *)
  GoToXY(10,22);
  Write('Val: ');
  CursorX:=WhereX;
  CursorY:=WhereY;
  ReadLn(InputNumber);
  IF LENGTH(InputNumber) <> 0 THEN
  BEGIN
    Val(InputNumber,FeedBackChoise,ErrorPos);
    WHILE (FeedBackChoise < 1) OR (FeedBackChoise > 5) DO (* HÄR ÄNDRAR NI EV.*
)
  BEGIN (* FELGRÄNS TILL 6 *
)
    SetCursorOff;
    TextBackground(12);
    TextColor(14);
    GoToXY(CursorX,CursorY);
    Write('FELAKTIGT VÄRDE !');
    DELAY(2000);
    GoToXY(CursorX,CursorY);
    TextBackground(0);
    TextColor(7);
    HighVideo;
    Write('          ');
    GoToXY(CursorX,CursorY);
    RestoreCursor;
    ReadLn(InputNumber);
    Val(InputNumber,FeedBackChoise,ErrorPos);
  END;
END; (* New_FeedBack_Alt *)

(*****)

PROCEDURE New_Filter_And_ControlConst;
(* Ändrar ev. värden på lågpasiskonstanten Alfa samt regulatorkonstanterna *)
(* Td,N,K och sampelintervallet. *)
BEGIN
  ClrScr;
  Ram;
  GoToXY(19,6);
  Write('VÄLJ FILTER-KONSTANTER SAMT SAMPELFREKVENSI!');
  GoToXY(10,9);
  Write('Lågpas-konstant  $\alpha$  (insignal) (0.00 - 1.00) ',LowPassConst:8:2,'
');
  ReadInputReal(0,1,1,LowPassConst);
  GoToXY(10,11);
  Write('Lågpas-konstant  $\alpha$  (utsignal) (0.00 - 1.00) ',LowPassConst2:8:2,'
');
  ReadInputReal(0,1,1,LowPassConst2);
  GoToXY(10,13);
  Write('Sampelfrekvens i Hz (25,50,75,100) ',SampleFr:8,' ');
  ReadInputInteger(25,100,SampleFr);
  WHILE (SampleFr<>25) AND (SampleFr<>50) AND (SampleFr<>75) AND
    (SampleFr<>100) DO
```

```
BEGIN
  GoToXY(62,13);
  Write('          ');
  GoToXY(62,13);
  ReadInputInteger(25,100,SampleFr);
END;
Beta:=1+NConst/SampleFr/TdConst;
ClrScr;
Ram;
GoToXY(16,6);
Write('VÄLJ REGLER-KONSTANTER TILL SYSTEMTRÖGHETEN !');
GoToXY(10,9);
Write('Derivata konstant Td          (0.0 - 50.0)      ',TdConst:8:1,'      ');

ReadInputReal(0,50,1,TdConst);
Beta:=1+NConst/SampleFr/TdConst;
GoToXY(10,11);
Write('Heltals-konstant N          (1.0 - 50.0)      ',NConst:8,'      ');
ReadInputInteger(1,50,NConst);
Beta:=1+NConst/SampleFr/TdConst;
GoToXY(10,13);
Write('Derivata-förstärkning        (0.0 - 10.0)      ',DConst/11000:8:1,'
');
ReadInputReal(0,10,11000,DConst);
GoToXY(16,16);
Write('VÄLJ REGLER-KONSTANTER TILL GIRACCELERATION !');
GoToXY(10,19);
Write('Derivata-konstant Td          (0.0 - 50.0)      ',TdConstR/0.6:8:1,'
');
ReadInputReal(0,50,0.6,TdConstR);
Beta:=1+NConstR/SampleFr/TdConstR;
GoToXY(10,21);
Write('Heltals-konstant N          (1.0 - 50.0)      ',NConstR:8,'      ');
ReadInputInteger(1,50,NConstR);
Beta:=1+NConstR/SampleFr/TdConstR;
GoToXY(10,23);
Write('Derivata-förstärkning        (0.0 - 10.0)      ',DConstR/8E5:8:1,'
');
ReadInputReal(0,10,8E5,DConstR);
END; (* New_Filter_And_ControlConst *)
```

(\*\*\*\*\*)

```
PROCEDURE Write_A_and_B_Matrix;
(* Skapar en fil för A och B-matriser som används i beräkningarna *)
VAR
  Ch      :Char;
  I,J     :INTEGER;
  FilNamn :STRING[3];
BEGIN
  ClrScr;
  WriteLn;
  Write('Ange sampelfrekvensen i Hz ');Readln(SampleFr);WriteLn;
  Ts:=1/SampleFr;
  Str(SampleFr:2,FilNamn);
  ASSIGN(fi,FilNamn+'.DAT');
  WriteLn('Matriserna är på formen:');
  WriteLn(' a b          e');
```

```
WriteLn('A=      B=' );
WriteLn(' c d      f');WriteLn;
FOR I:=0 TO 20 DO
  FOR J:=1 TO 6 DO
    A_B[J,I]:=0;
REPEAT
  Write('u=(knop) ');ReadLn(i);
  FOR J:= 1 TO 6 DO
    BEGIN
      Write(MatInd[J], '=');ReadLn(A_B[J,I]);
    END;
  Write('Fler värden J/N ');ReadLn(Ch);
UNTIL UpCase(Ch)='N';
ReWrite(fi);          (* Skapa filen filnamn.DAT och skriver in *)
Write(fi,Ts);         (* Ts och Matriskoefficienterna för u=1..20 *)
FOR I:=0 TO 20 DO
  FOR J:=1 TO 6 DO
    Write(fi,A_B[J,I]);
Close(fi);
WriteLn;
END; (* Write_A_And_B_Matrix *)

(*****)

PROCEDURE NewMatrix;
(* När u ändrats mer än en knop läses nya A- och B matriser in *)
BEGIN
  a:=A_B[1,uKnot];b:=A_B[2,uKnot];c:=A_B[3,uKnot];d:=A_B[4,uKnot];
  e:=A_B[5,uKnot];f:=A_B[6,uKnot];
END; (* NewMatrix *)

(*****)

PROCEDURE Read_A_And_B_Matrix;
(* Läser in en tidigare skapad fil för den iterativa formeln *)
(*  $v, r(t+1) = A*v, r(t) + b*\delta(t)$ . A och B matriserna finns i den *)
VAR
  I,J      :INTEGER;
  FilNamn :STRING[3];
BEGIN
  Str(SampleFr:2,FilNamn);
  ASSIGN (fi,FilNamn+'.DAT');
  RESET (fi);
  Read(fi,Ts);
  FOR I:=0 TO 20 DO
    FOR J:= 1 TO 6 DO
      Read(fi,A_B[J,I]);
    CLOSE (fi);
  END; (* Read_A_And_B_Matrix *)

(*****)

FUNCTION AD (Channel:INTEGER):INTEGER;
(* Gör analog till digital-omvandling *)
CONST Baseadr=632;
```

Konvl =7;  
Length =4;

VAR Highb,A,C,J,Loop: INTEGER;  
    Dummy          : Real;

BEGIN  
  PORT[Baseadr]:=Channel;    (\* Adressering AD-channel (0-15) \*)  
  A:=PORT[Baseadr+3];      (\* Clear AD-register \*)  
  FOR J:=1 TO Konvl DO  
    BEGIN  
      A:=PORT[Baseadr+4];  
      Dummy:=1;  
      FOR Loop:=1 TO Length DO  
        Dummy:=2+3;  
      END;  
      FOR J:=1 TO Konvl DO  
        BEGIN  
          A:=PORT[Baseadr+5];  
          Dummy:=1;  
          FOR Loop:=1 TO Length DO  
            Dummy:=2+3;  
          END;  
          C:=PORT[Baseadr+2];  
          Ad:=TRUNC((C/16-INT(C/16))\*16)\*256+PORT[Baseadr+1];  
        END;  
      END;  
  END; (\* AD \*)

(\*\*\*\*\*)

PROCEDURE DA (Value:INTEGER);  
(\* Gör digital till analog-omvandling \*)  
CONST Baseadr=632;

VAR Highb,Lowb:INTEGER;

BEGIN  
  Highb:=TRUNC(Value/256);  
  Lowb:=Value-Highb\*256;  
  PORT[Baseadr+7]:=Highb;  
  PORT[Baseadr+6]:=Lowb;  
END; (\* DA \*)

(\*\*\*\*\*)

PROCEDURE InitiateDefault;  
(\* Initierar defaultvärden på alla återkopplings- och filterparametrar \*)  
(\* samt sampelfrekvens \*)

BEGIN  
  Quit:=FALSE;  
  Calibrated:= FALSE;  
  Break:=FALSE;  
  LowPassConst:=0.93;  
  LowPassConst2:=0.0;  
  Ordn2LowPassConst:=0.8;  
  DConst:=55E3;  
  TdConst:=5;  
  NConst:=1;  
  DConstR:=4E6;



```
TdConstR:=3;
NConstR:=1;
SampleFr:=100;
FeedBackChoise:=2;
K1:=500; K2:=30000; K3:=0; K4:=20000; K5:=500; K6:=30000; K7:=1000;
K8:=3; K9:=0.1; K10:=0; K11:=0; K12:=0; K13:=0; K14:=0;
Beta:=1+NConst/SampleFr/TdConst;
BetaR:=1+NConstR/SampleFr/TdConstR;
OldHelpVar:=0;
OldValue:=0;
OldHelpVarR:=0;
OldValueR:=0;
S:=0;
DA(2048);
END; (* InitiateDefault *)
```

(\*\*\*\*\*)

```
PROCEDURE CheckPowerOn;
(* Kontrollerar om spänningen är påslagen *)
BEGIN
  WHILE (AD(0) < 250) AND (AD(1) < 250) DO
    BEGIN
      SetCursorOff;
      ClrScr;
      Ram;
      TextBackground(12);
      TextColor(14);
      GoToXY(15,12);
      Write('┌───────────────────────────────────────────────────────────┐');
      GoToXY(15,13);
      Write('┌ SLÅ PÅ SPÄNNINGEN TILL SIMULATORUTRUSTNINGEN ! ────────────────────┐');
      GoToXY(15,14);
      Write('└───────────────────────────────────────────────────────────┘');
      TextBackground(0);
      TextColor(7);
      HighVideo;
      CrashSound;
      GoToXY(24,20);
      WriteLn('Tryck på Return när du är klar !');
      REPEAT
        UNTIL ReadKey=chr(13);
      RestoreCursor;
    END;
  END; (* CheckPowerOn *)
```

(\*\*\*\*\*)

```
PROCEDURE Calibrate;
(* Kalibrerar ratt och gasreglage *)
CONST
  RadAndSSPACnst = 0.06981317; (* 4*Pi/180 *)
  LeftRudderAngle = -30*RadAndSSPACnst;
  RightRudderAngle = 30*RadAndSSPACnst;
  TotalAngle = RightRudderAngle-LeftRudderAngle;
  MaxSpeed = 10.26666667; (* 20 knop i m/s *)
```

```
VAR
  LoValue,HiValue : INTEGER;
BEGIN
  CheckPowerOn;
  SetCursorOff;
  ClrScr;
  Ram;
  GoToXY(14,12);
  WriteLn('Sätt ratten i maximalt högerutslag (+30 grader ) !');
  GoToXY(22,14);
  WriteLn('Tryck på Return när du är klar !');
  REPEAT
  UNTIL ReadKey=chr(13);
  LoValue:=AD(drChannel);
  WriteLn(#7);
  WriteLn(#7);
  ClrScr;
  Ram;
  GoToXY(14,12);
  WriteLn('Sätt ratten i maximalt vänsterutslag (-30 grader ) !');
  GoToXY(22,14);
  WriteLn('Tryck på Return när du är klar !');
  REPEAT
  UNTIL ReadKey=chr(13);
  HiValue:=AD(drChannel);
  drOffset:=(LeftRudderAngle-LoValue)/(HiValue-LoValue)*TotalAngle;
  drMultFact:=TotalAngle/(HiValue-LoValue);
  WriteLn(#7);
  WriteLn(#7);
  ClrScr;
  Ram;
  GoToXY(20,12);
  WriteLn('Sätt fartreglaget i maximalt läge !');
  GoToXY(21,14);
  WriteLn('Tryck på Return när du är klar !');
  REPEAT
  UNTIL ReadKey=chr(13);
  HiValue:=AD(uChannel);
  uMultFact:=MaxSpeed/HiValue;
  Calibrated:=TRUE;
  RestoreCursor;
END; (* Calibrate *)

(*****)
```

```
PROCEDURE Calibrate_Or_Default;
(* Väljer om vi skall kalibrera eller sätta tillbaks defaultvärdena *)
VAR
  CH      :CHAR;
BEGIN
  ClrScr;
  Ram;
  GoToXY(10,12);
  WriteLn('Om ni vill kalibrera eller få tillbaka defaultvärdena tryck  ');
  GoToXY(10,14);
  Write('på K respektive D, annars tryck Return !');
  GoToXY(51,14);
  REPEAT
```

```
    CH:=UpCase(ReadKey);
UNTIL (CH=chr(13)) OR (CH='K') OR (CH='D');
IF CH='K' THEN
    Calibrate
ELSE IF CH='D' THEN
    InitiateDefault;
END; (* Calibrate_Or_Default *)
```

(\*\*\*\*\*)

```
FUNCTION LowPass (Value:INTEGER):REAL;
(* Implementerar ett första ordningens lågpasfilter *)
VAR P :REAL;
BEGIN
    P:=LowPassConst*OldFiltValue+(1-LowPassConst)*Value;
    OldFiltValue:=P;
    LowPass:=P;
END; (* LowPass *)
```

(\*\*\*\*\*)

```
PROCEDURE CheckSteeringWeelPos;
(* Ställer den återkopplade kraften till noll och ser till att ratten *)
(* står i mittläget innan simuleringen börjar *)
CONST
    KnotConv = 1.94805195; (* m/s till knop *)
BEGIN
    CheckPowerOn;
    SetCursorOff;
    r:=0;
    v:=0;
    A:=0;
    Delta:=0;
    Break:=FALSE;
    OldFeedBackValue:=2048;
    OldFiltValue:=2048;
    OldFiltValue2:=2048;
    OldY1Filt:=2048;
    OldY2Filt:=2048;
    OldHelpVar:=0;
    OldHelpVarR:=0;
    REPEAT
        Moved:=FALSE;
        WeelValue:=AD(0);
        IF (WeelValue < 2000) OR (WeelValue > 2100) THEN
            BEGIN
                ClrScr;
                Ram;
                GoToXY(20,11);
                Write('SÄTT RATTEN HORISONTELLT ! ');
                IF WeelValue < 2000 THEN
                    BEGIN
                        Write('(VRID MEDSOLS)');
                        FailSound(0.8);
                    END
                ELSE IF WeelValue > 2100 THEN
                    BEGIN
```

```
Write('( VRID MOTSOLS )');  
FailSound(0.9);  
END;  
GoToXY(28,14);  
Write('TRYCK DÄREFTER PÅ RETURN !');  
REPEAT  
UNTIL ReadKey=chr(13);  
WeelValue:=AD(0);  
WHILE (WeelValue < 2010) OR (WeelValue > 2070) DO  
BEGIN  
  ClrScr;  
  Ram;  
  GoToXY(20,10);  
  Write('TYVÄRR - INTE TILLRÄCKLIGT HORISONTELLT !');  
  GoToXY(25,12);  
  Write('FÖRSÖK IGEN ! ');  
  IF WeelValue < 2000 THEN  
  BEGIN  
    Write('(VRID MEDSOLS)');  
    FailSound(0.8);  
  END  
  ELSE IF WeelValue > 2100 THEN  
  BEGIN  
    Write('( VRID MOTSOLS )');  
    FailSound(0.9);  
  END;  
  GoToXY(27,16);  
  Write('TRYCK DÄREFTER PÅ RETURN !');  
  REPEAT  
  UNTIL ReadKey=chr(13);  
  WeelValue:=AD(0);  
END;  
END;  
ClrScr;  
Ram;  
GoToXY(27,10);  
Write('NU ÄR RATTEN HORISONTELL !');  
FOR I:=1 TO 2 DO  
BEGIN  
  GoToXY(13,13);  
  TextBackground(12);  
  TextColor(14);  
  Write(' ');  
  DELAY(100);  
  GoToXY(12,12);  
  Write(' ');  
  GoToXY(12,13);  
  Write(' || RUBBA DEN INTE UR SITT LÄGE INNAN SIMULERINGEN BÖRJAT ! || ');  
  GoToXY(12,14);  
  Write(' ');  
  WriteLn(#7);  
  WriteLn(#7);  
  TextBackground(0);  
  TextColor(7);  
  HighVideo;  
  DELAY(900);  
END;  
GoToXY(27,16);  
Write('TRYCK DÄREFTER PÅ RETURN !');
```

```

REPEAT
UNTIL ReadKey=chr(13);
FOR I:=1 TO 50 DO          (* För att erhålla stabilt dr-värde *)
  dr:=drOffset+LowPass(AD(drChannel))*drMultFact;
IF ABS(AD(0)-WeelValue)>20 THEN  (* Om man rubbat ratten *)
BEGIN
  Moved:=TRUE;
  ClrScr;
  Ram;
  GoToXY(13,13);
  TextBackground(12);
  TextColor(14);
  GoToXY(22,12);
  Write('┌───────────────────────────────────────────┐');
  GoToXY(22,13);
  Write('|| AJA BAJA, DAN ! INTE RÖRA RATTEN !! ||');
  GoToXY(22,14);
  Write('└───────────────────────────────────────────┘');
  TextBackground(0);
  TextColor(7);
  HighVideo;
  CrashSound;
  DELAY(2000);
END;
u:=AD(uChannel)*uMultFact;
uKnot:=ROUND(KnotConv*u);
NewMatrix;
v:=v*a+r*b+dr*e;
r:=v*c+r*d+dr*f;
OldValue:=dr;           (* För att derivata-delar skall bli noll *)
)
OldValueR:=r;          (* vid simuleringens början *)
UNTIL NOT Moved;
RestoreCursor;
END; (* CheckSteeringWeelPos *)

(*****)

PROCEDURE InstallGraficMode;
(* Introducerar grafisk mod *)
CONST
  PiHalf = 1.5707963;      (* pi/2 *)
BEGIN
  IF NOT Calibrated THEN
    Calibrate;
  CheckSteeringWeelPos;
  x:=375.0; y:=1075;
  u:=0; uKnot:=0;
  OlduKnot:=0; UP1:=129;
  psi:=PiHalf; dr:=0.0;
  XKoord:=80; YKoord:=75;
  RP1:=145;                (* Ger ett OldRP1 som inte suddar vid *)
  DRP1:=145;              (* Ger ett OldDRP1 som inte suddar vid *)
  Read_A_And_B_Matrix;    (* första anrop av PlotDR *)
  NewMatrix;
  GraphDriver := Detect;  (* Automatisk detektering av grafikhårdvara *)
  InitGraph(GraphDriver,GraphMode,'');
  IF GraphResult <> grOk THEN

```

```
Halt(1);
SetWriteMode(CopyPut); (* Grafikinitiering *)
SetFillStyle(SolidFill,LightGray); (* Färglägger bakgrunden *)
FloodFill(11,11,LightGray);
SetColor(Black);
SetTextJustify(CenterText,TopText); (* Rita ut text till sjökortet *)
OutTextXY(170,465,'SJÖKORT');
Line(250,470,280,470); (* x-axel skala *)
Line(255,468,255,472); (* Skalstreck *)
Line(275,468,275,472);
OutTextXY(300,465,'50 m');
SetColor(White);

WITH Map DO (* 300*440 *)
BEGIN
  Rectangle(Succ(x1),Succ(y1), (* Märk ut viewport Map *)
            Pred(x2),Pred(y2));
  SetViewPort(x1, y1, x2, y2, ClipOn);
  SetFillStyle(SolidFill,blue);
  FloodFill(10,10,Blue);
END;

SetViewPort(0, 0, GetMaxX, GetMaxY, ClipOn); (* viewport hela skärmen *)

WITH Compass DO (* 160*160 *)
BEGIN
  Rectangle(Succ(x1),Succ(y1), (* Märk ut viewport Compass *)
            Pred(x2),Pred(y2));
  SetViewPort(x1, y1, x2, y2, ClipOn);
  SetFillStyle(SolidFill,Black);
  FloodFill(10,10,GetMaxColor);
  OutTextXY(80,150,'KURS');
  Circle(80,75,50);
  Line(80,20,80,30); (* Markeringslinje N *)
  OutTextXY(80,5,'N');
  Line(95,33,99,23); (* Markeringslinje 20 *)
  OutTextXY(100,13,'20');
  Line(109,41,115,33); (* Markeringslinje 40 *)
  OutTextXY(122,25,'40');
  Line(119,53,128,48); (* Markeringslinje 60 *)
  OutTextXY(138,41,'60');
  Line(124,67,134,65); (* Markeringslinje 80 *)
  OutTextXY(145,59,'80');
  Line(125,75,135,75); (* Markeringslinje 90 Ö *)
  OutTextXY(147,71,'Ö');
  Line(124,83,134,85); (* Markeringslinje 100 *)
  OutTextXY(145,83,'100');
  Line(119,98,128,103); (* Markeringslinje 120 *)
  OutTextXY(142,100,'120');
  Line(109,109,115,117); (* Markeringslinje 140 *)
  OutTextXY(131,114,'140');
  Line(95,117,99,127); (* Markeringslinje 160 *)
  OutTextXY(112,128,'160');
  Line(80,120,80,130); (* Markeringslinje 180 S *)
  OutTextXY(80,135,'S');
  Line(65,117,61,127); (* Markeringslinje 200 *)
  OutTextXY(49,130,'200');
  Line(51,109,45,117); (* Markeringslinje 220 *)
  OutTextXY(28,116,'220');
```

```
Line(41,98,32,102);      (* Markeringslinje 240 *)
OutTextXY(18,100,'240');
Line(36,83,26,85);      (* Markeringslinje 260 *)
OutTextXY(13,83,'260');
Line(25,75,35,75);      (* Markeringslinje 270 V *)
OutTextXY(13,71,'V');
Line(36,67,26,65);      (* Markeringslinje 280 *)
OutTextXY(13,60,'280');
Line(41,53,32,48);      (* Markeringslinje 300 *)
OutTextXY(16,41,'300');
Line(51,41,45,33);      (* Markeringslinje 320 *)
OutTextXY(30,25,'320');
Line(65,33,61,23);      (* Markeringslinje 340 *)
OutTextXY(55,13,'340');
END;

SetViewport(0, 0, GetMaxX, GetMaxY, ClipOn);      (* viewport hela skärmen *)

WITH SpeedometerR DO (* 290*80 *)
BEGIN
  Rectangle(Succ(x1),Succ(y1),                      (* Märk ut viewport Speedometer *)
            Pred(x2),Pred(y2));
  SetViewport(x1, y1, x2, y2, ClipOn);
  SetFillStyle(SolidFill,Black);
  FloodFill(10,10,GetMaxColor);
  SetTextJustify(CenterText,TopText);
  OutTextXY(145,70,'GIRVINKELHASTIGHET (grad/s)');
  Line(15,46,275,46);      (* x-linje *)
  FOR LoopCount:=-8 TO 8 DO
    Line(145+LoopCount*15,47,145+LoopCount*15,50);
  SetTextJustify(CenterText,TopText);
  OutTextXY(25,55,'8');      (* Rita ut skalmarkeringen på x-axel *)
  OutTextXY(55,55,'6');
  OutTextXY(85,55,'4');
  OutTextXY(115,55,'2');
  OutTextXY(145,55,'0');
  OutTextXY(175,55,'2');
  OutTextXY(205,55,'4');
  OutTextXY(235,55,'6');
  OutTextXY(265,55,'8');
END;

SetViewport(0, 0, GetMaxX, GetMaxY, ClipOn);      (* viewport hela skärmen *)

WITH AnglometerDR DO (* 290*80 *)
BEGIN
  Rectangle(Succ(x1),Succ(y1),                      (* Märk ut viewport Anglometer *)
            Pred(x2),Pred(y2));
  SetViewport(x1, y1, x2, y2, ClipOn);
  SetFillStyle(SolidFill,Black);
  FloodFill(10,10,GetMaxColor);
  SetTextJustify(CenterText,TopText);
  OutTextXY(145,70,'RODERVINKEL (grader)');
  Line(15,46,275,46);      (* x-linje *)
  FOR LoopCount:=-6 TO 6 DO
    Line(145+LoopCount*20,47,145+LoopCount*20,50);
  SetTextJustify(CenterText,TopText);
  OutTextXY(25,55,'30');      (* Rita ut skalmarkeringen på x-axel *)
```

```
OutTextXY(45,55,'25');
OutTextXY(65,55,'20');
OutTextXY(85,55,'15');
OutTextXY(105,55,'10');
OutTextXY(125,55,'5');
OutTextXY(145,55,'0');
OutTextXY(165,55,'5');
OutTextXY(185,55,'10');
OutTextXY(205,55,'15');
OutTextXY(225,55,'20');
OutTextXY(245,55,'25');
OutTextXY(265,55,'30');
END;

SetViewPort(0, 0, GetMaxX, GetMaxY, ClipOn);    (* viewport hela skärmen *)

WITH SpeedometerU DO  (* 290*80 *)
BEGIN
  Rectangle(Succ(x1),Succ(y1),                  (* Märk ut viewport Speedometer *)
            Pred(x2),Pred(y2));
  SetViewPort(x1, y1, x2, y2, ClipOn);
  SetFillStyle(SolidFill,Black);
  FloodFill(10,10,GetMaxColor);
  SetTextJustify(CenterText,TopText);
  OutTextXY(41,140,'HASTIGHET');
  OutTextXY(41,150,'(knop)');
  Line(12,130,60,130);    (* x-linje *)
  Line(62,7,62,130);    (* y-linje *)
  FOR LoopCount:= 0 TO 10 DO
    Line(61,10+LoopCount*12,63,10+LoopCount*12);
  SetTextJustify(CenterText,TopText);
  OutTextXY(70,7,'20');    (* Rita ut skalmarkeringen på y-axel *)
  OutTextXY(70,19,'18');
  OutTextXY(70,31,'16');
  OutTextXY(70,43,'14');
  OutTextXY(70,55,'12');
  OutTextXY(70,67,'10');
  OutTextXY(70,79,'8');
  OutTextXY(70,91,'6');
  OutTextXY(70,103,'4');
  OutTextXY(70,115,'2');
  OutTextXY(70,127,'0');
END;

SetViewPort(0, 0, GetMaxX, GetMaxY, ClipOn);    (* viewport hela skärmen *)

WITH StopDisplay DO  (* 90*60 *)
BEGIN
  Rectangle(Succ(x1),Succ(y1),                  (* Märk ut viewport StopDisplay *)
            Pred(x2),Pred(y2));
  SetViewPort(x1, y1, x2, y2, ClipOn);
  SetFillStyle(SolidFill,Red);
  FloodFill(10,10,GetMaxColor);
  SetTextJustify(CenterText,TopText);
  SetTextStyle(Triplexfont,horizDir,4);
  OutTextXY(45,4,'STOP');
  SetTextStyle(DefaultFont,horizDir,1);
```



```
    OutTextXY(44,50,'PRESS KEY');  
END;  
LoopCount:=0;  
END; (* InstallGraficMode *)
```

(\*\*\*\*\*)

```
PROCEDURE InstallDelay;  
(* Installerar avbrott med frekvensen SampleFr mha timer 0 *)  
VAR  
    CountTo : INTEGER;  
BEGIN  
    OldMask:=PORT[IntrMask]; (* Sparar gammal interruptmask *)  
    PORT[IntrMask]:=$FC;      (* Maskar ut timer0-& tangentbordsinterrupt *)  
    CountTo:=ROUND(CIFr/SampleFr);  
    PORT[TimerCom]:=$36;      (* Öppnar kommandoregistret på timern *)  
    PORT[Kanal0]:=LO( CountTo ); (* Skriver in ny avbrottsfrekvens *)  
    PORT[Kanal0]:=HI( CountTo );  
END; (* InstallDelay *)
```

(\*\*\*\*\*)

```
PROCEDURE Back;  
(* Återställer interrupt- och skärmtillstånd till normalt läge och *)  
(* lägger ut kraften=0 på ratten *)  
BEGIN  
    PORT[IntrMask]:=OldMask;  
    PORT[TimerCom]:=$36;  
    PORT[Kanal0]:=255;  
    PORT[Kanal0]:=255;  
    CloseGraph;  
    DA(2048);  
END; (* Back *)
```

(\*\*\*\*\*)

```
PROCEDURE PlotR;  
(* Plottar girvinkelhastigheten på skärmen *)  
CONST  
    RadToGrad =57.29578;  
BEGIN  
    WITH SpeedometerR DO  
    BEGIN  
        OldRP1:=RP1; (* OldRP1 används för radering av gammalt r *)  
        RP1:=145+ROUND(r*RadToGrad*15);  
        SetViewPort(x1, y1, x2, y2, ClipOn);  
        SetColor(Black);  
        SetFillStyle(SolidFill,Black);  
        IF (OldRP1<145) AND (RP1>OldRP1) THEN  
            Bar(OldRP1,20,RP1,44)  
        ELSE IF (OldRP1>145) AND (OldRP1>RP1) THEN  
            Bar(RP1,20,OldRP1,44);  
        SetFillStyle(SolidFill,LightMagenta);  
        IF (RP1<145) AND (OldRP1>RP1) THEN  
            IF OldRP1>145 THEN  
                Bar(RP1,20,145,44)
```

```
ELSE
  Bar(RP1,20,OldRP1,44)
ELSE IF (RP1>145) AND (RP1>OldRP1) THEN
  IF OldRP1<145 THEN
    Bar(145,20,RP1,44)
  ELSE
    Bar(OldRP1,20,RP1,44);
END; (* SpeedometerR *)
END; (* PlotR *)
```

(\*\*\*\*\*)

```
PROCEDURE PlotDR;
(* Plottar rattvinkeln på skärmen *)
BEGIN
  WITH AnglometerDR DO
  BEGIN
    OldDRP1:=DRP1;      (* OldRP1 används för radering av gammalt r *)
    DRP1:=145-ROUND(dr*56.872);
    SetViewPort(x1, y1, x2, y2, ClipOn);
    SetColor(Black);
    SetFillStyle(SolidFill,Black);
    IF (OldDRP1<145) AND (DRP1>OldDRP1) THEN
      Bar(OldDRP1,20,DRP1,44)
    ELSE IF (OldDRP1>145) AND (OldDRP1>DRP1) THEN
      Bar(DRP1,20,OldDRP1,44);
    SetFillStyle(SolidFill,Yellow);
    IF (DRP1<145) AND (OldDRP1>DRP1) THEN
      IF OldDRP1>145 THEN
        Bar(DRP1,20,145,44)
      ELSE
        Bar(DRP1,20,OldDRP1,44)
    ELSE IF (DRP1>145) AND (DRP1>OldDRP1) THEN
      IF OldDRP1<145 THEN
        Bar(145,20,DRP1,44)
      ELSE
        Bar(OldDRP1,20,DRP1,44);
    END; (* SpeedometerDR *)
  END; (* PlotDR *)
```

(\*\*\*\*\*)

```
PROCEDURE PlotUKnot;
(* Plottar hastigheten på skärmen *)
BEGIN
  WITH SpeedometerU DO
  BEGIN
    OldUP1:=UP1;      (* OldUP1 används för radering av gammalt UP1 *)
    UP1:=129 - UKnot*6;
    SetViewPort(x1, y1, x2, y2, ClipOn);
    SetColor(Black);
    IF UP1>OldUP1 THEN
      BEGIN
        SetFillStyle(SolidFill,Black);
        Bar(36,OldUP1,60,UP1);
      END
    ELSE IF UP1<OldUP1 THEN
```

```
BEGIN
  SetFillStyle(SolidFill,LightBlue);
  Bar(36,UP1,60,OldUP1);
END;
END; (* SpeedometerU *)
END; (* PlotUKnot *)

(*****)
```

```
PROCEDURE PlotCourse;
(* Plottar kursen på skärmen *)
BEGIN
  WITH Compass DO
  BEGIN
    SetViewPort(x1, y1, x2, y2, ClipOn);
    Line(80,75,XKoord,YKoord); (* Raderar gammal kurs *)
    SetColor(White);
    XKoord:=80-ROUND(45*cospsi);
    YKoord:=75-ROUND(45*sinpsi);
    Line(80,75,XKoord,YKoord);
  END; (* Compass *)
END; (* PlotCourse *)

(*****)
```

```
PROCEDURE PlotXY;
(* Plottar positionen på skärmen *)
BEGIN
  IF x>747.5 THEN
    x:=2.5
  ELSE IF x<2.5 THEN
    x:=747.5;
  IF y>1100 THEN
    y:=2.5
  ELSE IF y<2.5 THEN
    y:=1100;
  XX:= ROUND(x*0.4); (* XX och YY skalas med antal *)
  YY:= ROUND(y*0.4); (* meter*0.4 *)
  WITH Map DO (* procedure PlotXY *)
  BEGIN
    SetViewPort(x1,y1,x2,y2,ClipOn);
    PutPixel(XX,YY,White);
  END; (* Map *)
END; (* PlotXY *)

(*****)
```

```
FUNCTION LowPass2 (Value:REAL):REAL;
(* Implementerar ett första ordningens lågpasfilter *)
(* att användas på utsignalen *)
VAR P2 :REAL;
BEGIN
  P2:=LowPassConst2*OldFiltValue2+(1-LowPassConst2)*Value;
  OldFiltValue2:=P2;
  LowPass2:=P2;
END; (* LowPass2 *)
```

(\*\*\*\*\*)

```
FUNCTION Sign :INTEGER;
(* Ger det tecken som rattvinkeln har, +1 eller -1 *)
BEGIN
  IF dr<0 THEN
    Sign:=-1
  ELSE IF dr>0 THEN
    Sign:=1
  ELSE
    Sign:=0;
END; (* Sign *)
```

(\*\*\*\*\*)

```
FUNCTION DregDR :REAL;
(* Implementerar en D-regulator som reglerar på felet *)
(* mellan nytt och gammalt värde av rattvinkeln *)
VAR
  Deviation,HelpVar : REAL;
BEGIN
  Deviation:=OldValue-dr;
  HelpVar:=1/Beta*OldHelpVar+(Beta-1)/Beta*Deviation;
  OldHelpVar:=HelpVar;
  OldValue:=dr;
  DregDR:=DConst*(HelpVar-Deviation);
END; (* DregDR *)
```

(\*\*\*\*\*)

```
FUNCTION DregR :REAL;
(* Implementerar en D-regulator som reglerar på felet *)
(* mellan nytt och gammalt värde av girvinkelhastigheten *)
VAR
  Deviation,HelpVar : REAL;
BEGIN
  Deviation:=OldValueR-r;
  HelpVar:=1/BetaR*OldHelpVarR+(BetaR-1)/BetaR*Deviation;
  OldHelpVarR:=HelpVar;
  OldValueR:=r;
  DregR:=DConstR*(HelpVar-Deviation);
END; (* DregR *)
```

(\*\*\*\*\*  
(\* HÄR ÄNDRAR MAN, OM MAN VILL ÄNDRA FORMEL FÖR DEN ÅTERKOPPLADE KRAFTEN \*)  
\*\*\*\*\*)

```
FUNCTION FeedBack :INTEGER;
(* Ger valt återkopplingsalternativ *)
VAR
  Value,A :REAL;
```

```
BEGIN
  A:=Alfa*Alfa ;
  CASE FeedBackChoise OF
    1: Value:= u*A*(Sign*K1+K2*Delta*A)-DregR-K3*r+DregDR+2048;
    2: Value:= DregDR-DregR+2048;
    3: Value:= DregDR-K4*r+2048;
    4: Value:= DregDR+(u*A*(K5*Sign+K6*Delta*A))+2048;
    5: Value:= Sign*(K7*Ln(K8*abs(dr)+1)*Ln(K9*u+1))+DregDR+2048;

  (***** HÄR FINNS MÖJLIGHET FÖR ER ATT SJÄLV SKAPA ETT UTTRYCK *****)
  (***** FÖR DEN ÅTERKOPPLADE KRAFTEN *****)

  (* 6: Value:=...använd konstanterna K10 - K14 i ert uttryck... *****)

  END; (* CASE *)
  IF Value < 10 THEN          (* För att undvika olinearitet *)
    Value:=10                (* hos AD-omvandlaren *)
  ELSE IF Value > 4085 THEN
    Value:=4085;
  IF ABS(OldFeedBackValue-Value) > 200 THEN  (* För att åstadkomma att *)
  BEGIN                                       (* simuleringen avbryts om *)
    Value:=2048;                             (* skillnaden blir för stor *)
    Stop:=TRUE;
    Break:=TRUE;
  END;
  OldFeedBackValue:=Value;
  FeedBack:= ROUND(LowPass2(Value));
END; (* FeedBack *)

(*****
*****

FUNCTION SecondOrderLowPass (Value:INTEGER):REAL;
(* Implementerar ett andra ordningens lågpasfilter, vilket kan användas *)
(* istället de första ordningens filter som finns på ingång och utgång *)
VAR Y1Filt,Y2Filt :REAL;
BEGIN
  Y1Filt:=Ordn2LowPassConst*OldY1Filt+(1-Ordn2LowPassConst)*Value;
  Y2Filt:=Ordn2LowPassConst*OldY2Filt+(1-Ordn2LowPassConst)*Y1Filt;
  OldY1Filt:=Y1Filt;
  OldY2Filt:=Y2Filt;
  SecondOrderLowPass:=Y2Filt;
END; (* SecondOrderLowPass *)

(*****
*****

PROCEDURE Calculate_And_IO;
(* Hämtar in Rodervinkel och hastighet. Beräknar v och r samt *)
(* skickar ut återkopplingskraften FeedBack till ratten *)
CONST
  KnotConv = 1.94805195; (* Omvandlar m/s till knop *)
BEGIN
  OlduKnot:=uKnot;
  dr:=drOffset+LowPass(AD(drChannel))*drMultFact;
  u:= AD(uChannel)*uMultFact;
  uKnot:=ROUND(KnotConv*u);
  IF ABS(uKnot-OlduKnot)>=1 THEN          (* Nya A- och B-matriser om *)
```

```
NewMatrix; (* u ändrats mer än 1 knop *)  
v:= v*a+r*b+dr*e;  
r:= v*c+r*d+dr*f;  
Delta:=0.25*dr;  
Alfa:= Delta-v/u;  
DA(FeedBack);  
END; (* Calculate_And_IO *)
```

(\*\*\*\*\*)

```
PROCEDURE Sample;  
(* Beräknar ny position och kurs och hanterar IO och plottning *)  
BEGIN  
  INLINE($F4); (* HLT *)  
  Calculate_And_IO;  
  psi:=psi+r*Ts;  
  sinpsi:=sin(psi);  
  cospsi:=cos(psi);  
  x:=x-(u*cospsi-v*sinpsi)*Ts;  
  y:=y-(u*sinpsi+v*cospsi)*Ts;  
  LoopCount:=LoopCount+1;  
  CASE LoopCount OF  
    1: PlotR;  
    2: PlotCourse;  
    3: PlotUKnot;  
    4: PlotDR;  
    5: BEGIN  
      PlotXY;  
      LoopCount:=0;  
    END;  
  END; (* CASE *)  
  IF KeyPressed THEN  
    Stop:=True;  
  END; (* Sample *)
```

(\*\*\*\*\*)

```
PROCEDURE Simulate;  
(* Denna procedur styr hela simuleringen *)  
BEGIN  
  ShowChosenValues;  
  IF NOT Regret THEN  
    BEGIN  
      InstallGraficMode;  
      InstallDelay;  
      Stop:=FALSE;  
      REPEAT  
        Sample;  
      UNTIL Stop;  
      Back;  
    END;  
  END; (* Simulate *)
```

(\*\*\*\*\*)

```
PROCEDURE MainMeny;
```

```
(* Ger användaren möjlighet att välja vad programmet skall utföra *)
VAR
  CursorX,CursorY,ErrorPos   :INTEGER;
  InputNumber                 :STRING;
BEGIN
  ClrScr;
  Ram;
  GoToXY(35,6);
  Write('HUVUDMENY !');
  GoToXY(10,8);
  WriteLn('1... Information om försöksuppställning och parametrar');
  GoToXY(10,10);
  WriteLn('2... Simulera');
  GoToXY(10,12);
  WriteLn('3... Omkalibrera eller återställa till defaultvärden'); {Skapa fil för
r A och B matriserna'};
  GoToXY(10,14);
  WriteLn('4... Välj vilka parametrar den återkopplade kraften skall bestå av');

  GoToXY(10,16);
  WriteLn('5... Välj konstanter till vald återkopplad kraft');
  GoToXY(10,18);
  WriteLn('6... Välj filter-och reglerkonstanter samt sampelfrekvens');
  GoToXY(10,20);
  WriteLn('7... Avsluta');
  GoToXY(10,22);
  Write('Ange alternativ: ');
  CursorX:=WhereX;
  CursorY:=WhereY;
  ReadLn(InputNumber);
  Val(InputNumber,Choise,ErrorPos);
  WHILE (Choise < 1) OR (Choise > 7) OR (InputNumber='') DO
  BEGIN
    SetCursorOff;
    TextBackground(12);
    TextColor(14);
    GoToXY(CursorX,CursorY);
    Write('FELAKTIGT VÄRDE !');
    DELAY(2000);
    GoToXY(CursorX,CursorY);
    TextBackground(0);
    TextColor(7);
    HighVideo;
    Write('                ');
    GoToXY(CursorX,CursorY);
    RestoreCursor;
    ReadLn(InputNumber);
    Val(InputNumber,Choise,ErrorPos);
  END;
END; (* MainMeny *)

(*****

PROCEDURE Loop;
(* Exekverar programmet tills dess att användaren vill avbryta *)
BEGIN
  REPEAT
    BreakMeny;
    MainMeny;
```

```
CASE Choise OF
  1: InformationMeny;
  2: Simulate;
  3: Calibrate_Or_Default;    {Write_A_And_B_Matrix;} (* TA BORT DETTA *)
  4: New_FeedBack_Alt;      (* OM NY SAMPEL- *)
  5: New_FeedBack_Const;    (* FREKVENS SKALL *)
  6: New_Filter_And_ControlConst; (* SKAPAS *)
  7: Quit:=TRUE;
END; (* CASE *)
UNTIL Quit;
END; (* Loop *)
```

```
(*****
(*                               HUVUDPROGRAM                               *)
*****)
```

```
BEGIN
  IntroductionMeny;
  InitiateDefault;
  Loop;
END. (* Huvudprogram *)
```

```
(*****
(*                               SLUT                               *)
*****)
```



## Appendix C

### Teknisk specifikation

#### AD/DA-kort

##### Analog till digital-omvandling

- 12 bitar
- 16 kanaler
- Spänningsområde för insignal är 0 - 9 V
- Succesiv approximation används som omvandlingsmetod [10]
- För varje kanal är omvandlingstiden 60  $\mu$ s

##### Digital till analog-omvandling

- 12 bitar
- Utsignalen varierar mellan 0 - 9 V
- Tiden för en utsignal att bli stabil är 0.5  $\mu$ s
- Icke-lineariteten är 0.2 %

Effektförbrukning : 2.2 W

Vikt : 170 g

Kortstorlek : 25 cm  $\times$  11 cm

Anslutningar : 25 ben av D-typ

(s86)

#### Förstärkarkort (LA-5600)

För styrsignalen(VCS) gäller :

- Insignal område :  $\pm 10$  V nominellt,  $\pm 60$  V maximalt
- Inimpedans : 6 k $\Omega$  minimum
- Skalningsområde: 0 - 100% av insignalens spänning (justeras med pot R28)

För den tillgängliga  $\pm 15$  V kraftförsörjningen gäller :

- Var och en av utgångarna får inte belastas med mer än 50 mA
- Utsignalen(V) :  $15 \text{ V} \pm 0.6 \text{ V}$  för både positiv och negativ polaritet
- Utsignalstabilitet : vid belastning på 0 - 50 mA varierar spänningen 1%

Strömförstärkning : 5 A/V

Temperatur drift på insignalen :  $15 \mu\text{V}/^\circ\text{C}$

Utsignalens bandbredd : DC till 2 kHz (med motor och last ca 50 - 500 Hz)

Förvaringstemperatur :  $-25^\circ\text{C}$  till  $85^\circ\text{C}$

Temperatur vid användande :  $0^\circ\text{C}$  till  $50^\circ\text{C}$

Bör förvaras i dammfritt utrymme

Övrigt : Om någon ändring av styrstativet skall göras eller någon justering av potentiometrarna på kortet, rekommenderas en noggrannare genomgång av förstärkaren [14].

### **Motor**

Kontinuerligt moment vid lågt varvtal : 1,5 Nm

Max kontinuerlig uteffekt : 350 W

Max toppmoment : 6,4 Nm

Max kontinuerlig ström : 6 A

Max toppström : 26 A

Max radiallyast : 130 N (25,4 mm ifrån lagret)

Vikt : 4,5 kg

Utväxling till styrratten : 3:1

Övriga data, se [15].

**Gränssnitt**

Matningsspänning :  $\pm 15$  V (tas lämpligen från styrstativet)

Strömförbrukning : max 20 mA/Ingång

**Vinkelgivare**

Potentiometerresistans : 10 k $\Omega$

Strömförbrukning : ca 3 mA

# Appendix D

## AD/DA-kort

### I. INTRODUCTION

12 bit ADDA CARD (IBM PC VERSION) is a high precision data conversion system. It contain 1 channel 12 bit Digital to analog (setting jumper 2 for select unipolar or bipolar) and 16 channel 12 bit analog to digital conversion (unipolar).

### II. SPECIFICATION

#### D/A

- 12 bits, 1 channel
- Output voltage 0-9v (adjust by VR).
- Current settling time 500 nsec.
- Nonlinearity 0.2%

#### A/D:

- 12 bits, 16 channel.
- Input voltage range 0-9v
- Successive Approximation Method.
- Conversion time 60 usec (each channel).

I/O PORT ADDR: &H 278-27F

POWER COMSUMPTION: 2.2W

NET WEIGHT: 170g

CARD SIZE: 25cm × 11cm

### III. PACKING

- ADDA Interface Card (with D type 25 pin connector)
- User's manual

### IV. INSTALLATION

1. make sure that the power is off
2. remove the cover of your PC
3. install the ADDA CARD in any slot
4. replace the cover
5. insert the DEMONSTRATION DISKETTE in drive A
6. switch on the power

### V. OPERATION

1. wait until the cursor appear
2. key in the BASIC TEST PROGRAM then type RUN (Ref NOTE 4.)
3. the screen will display:
  - WHICH SELECTION DO YOU WANT?
  - 1. A/D MODE
  - 2. D/A MODE
4. If you select "1. A/D MODE", screen will display each value (from 0 to 4095 of the 16 channels.)
5. If you select "2. D/A MODE", pin 12 of D-TYPE connector will output 16 steps SAW-TOOTH WAVE.

## VI. BASIC PROGRAMMING UNDER MS-DOS

PORT = 760 (WHICH SELECT &H 2FB-2FF)  
PORT = 632 (WHEN SELECT &H 278-27F)

1. ANALOG TO DIGITAL (A/D) PROCEDURE:
  - (1) OUTPUT CHANNEL NUMBER TO PORT  
OUT PORT, CHANNEL
  - (2) CLEAR REGISTER  
INP (PORT + 3)
  - (3) START CONVERT  
FOR I = 1 TO 7  
A = INP (PORT + 4)  
NEXT I  
FOR I = 1 TO 7  
A = INP (PORT + 5)  
NEXT I
  - (4) READ HIGH BYTE (4-BITS)  
C = INP(PORT + 2)  
HB = (C/16 - INT(C/16))\* 16
  - (5) READ LOW BYTE (B-BITS)  
LB = INP(PORT + 1)
  - (6) DATE:  
AD = HB\* 256 + LB
2. DIGITAL TO ANALOG (D/A) PROCEDURE
  - (1) OUTPUT HIGH 4-BITS  
OUT PORT + 7, HDATA
  - (2) OUTPUT LOW B-BITS  
OUT PORT + +6, LDATA

### NOTE:

1. I/O PORT ADDRESS (&H 278-27F)
  - &H278 output A/D channel number (low 4 bits)
  - 279 input low 8 bits data
  - 27A input A/D high 4 bits data (low nibble)
  - 27B clear A/D Register
  - 27C A/D conversion loop (low)
  - 27D A/D conversion loop (high)
  - 27E output D/A low 8 bits data
  - 27F output D/A high 4 bits data (low nibble)

### 2. D TYPE CONNECTOR PINOUT

PIN	FUNCTION	PIN	FUNCTION
13	GND	25	GND
12	DAOUT	24	-5V
11	+5V	23	GND
10	GND	22	AD-CH15
9	AD-CHO	21	AD-CH14
8	AD-CH1	20	AD-CH13
7	AD-CH2	19	AD-CH12
6	AD-CH3	18	AD-CH11
5	AD-CH4	17	AC-CH10
4	AD-CH5	16	AD-CH9
3	AD-CH6	15	AD-CH8
2	AD-CH7	14	-12V
1	+12V		

down	1	2	3	4	5	6	7	8	9	10	11	12	13	up
	I												I	

### 3. BASIC TEST PROGRAM:

```

10 CLS
12 BA = 632
14 LOCATE 10, 25: PRINT "1 AD DEMO"
16 LOCATE 12, 25: PRINT "2 DA DEMO"
18 A$ = INKEY $
20 IF A$ = "1" THEN 100
22 IF A$ = "2" THEN 200
24 GOTO 18
100 CLS
102 PRINT: PRINT
105 FOR J = 0 TO 15
107 OUT BA, J
110 FOR I = 1 TO 7: A = INP(BA+4): NEXT I
130 FOR I = 1 TO 7: A = INP(BA+5): NEXT I
140 C = INP (BA+2)
150 HB = (C/16-INT(C/16)) * 16
160 LB = INP (BA+1)
170 AD = HB * 256 + LB
180 PRINT "CHANNEL = "; AD
190 NEXT J
195 GO TO 102
200 CLS
202 LOCATE 10, 25: PRINT "OUTPUT STAIR-STEP WAVEFORM"
205 FOR I = 0 TO 15
210 OUT BA+7, I
220 NEXT I
230 GOTO 205

```

## Appendix E

Artiklar saxade ur SvD måndagen den 12 juni 1989

# Styrspakens betydelse mycket omdiskuterad

Styrspaken är länken mellan piloten och flygplanet. Vilken roll spaken spelade i JAS-haveriet den 2 februari är mycket kontroversiellt. Saab och Statens haverikommission har olika uppfattningar och även inom Saabs flygdivision går meningarna isär.

Av haverikommissionens första delrapport framgår att den främsta orsaken till haveriet var att det uppstod ett slags egen-svängningar i planets datoriserade styrsystem. Orsaken till detta var framför allt fel i styrsystemets programvara.

Men enligt kommissionen bidrog dessutom själva styrspakens konstruktion till haveriförloppet. När piloten upplevde att planet inte lydde hans order kunde han alltför lätt göra kraftigare och kraftigare manövrer med spaken. Till slut var kraschen ett faktum.

- Det som hänt är ett skol-

exempel på bristande anpassning mellan människa och maskin. Med nuvarande egenskaper är planet för svårt att flyga, konstaterade haverikommissionens ordförande Olof Forssberg den 15 mars.

### Belönad med medalj

Inom Saab-Scania anser man däremot att spaken inte hade den betydelse för haveriet som haverikommissionen hävdar:

- Vi har en annan syn än kommissionen på detta, säger Saabs flygutprovningsschef, Milton Mobärg, numera med specialansvar för styrsystemet, till SvD.

Den styrspak som satt i det kraschade JAS-planet var inte färdigbyggd. Vissa funktioner saknades, vilket kan ha försvårat pilotens arbete. Detta fick konstruktören Lennart Nordström att reagera redan efter den första flygningen (se artikeln ovan).

Civilingenjör Lennart Nordströms specialitet är just anpassningen mellan människa och maskin. För sina insatser på området belönades han 1987

med flygtekniska föreningens Thulinmedalj i silver. Hans konstruktioner används i dag i flygplanssystemen Draken, Viggen, JAS-Gripen samt i det helikopterburna robotssystemet Helitow.

### Spak fick toppbetyg

JAS styrspak bygger på nya, helt unika konstruktionslösningar. Att konstruera en styrspak är mycket svårt, det visar erfarenheter från det amerikanska flygplanet F 16 där det tog 15 år att utveckla spaken. När prototypen till JAS styrspak flygprovades i USA fick den emellertid toppbetyg av amerikanska testpiloter.

Men inom Saab-Scania fanns uppenbarligen olika uppfattningar om hur den färdiga konstruktionen skulle göras och 1985 kopplades Lennart Nordström bort från arbetet.

Saab-Scania valde en engelsk firma, Page, som leverantör av styrspaken. I stället för Saab Combitech som tagit fram prototypen.

Enligt uppgifter från Försvarets materielverk har firman Page

inte klarat av att tillverka de momentmotorer som skall ingå i styrspaken så att de fick tillräckligt hög precision. Haverikommissionen har också bekräftat att momentmotorerna inte var inkopplade i det kraschade planet.

Det är uppenbarligen detta Lennart Nordström reagerat på. Enligt det material han lämnat till haverikommissionen spelar momentmotorerna en viktig roll för att hjälpa piloten vid svåra situationer. Som när provplanet skulle landa före kraschen, i turbulenta vindar samtidigt som andra delar i styrsystemet krånglade.

### Dämpa rörelser

Momentmotorerna har flera funktioner:

□ Dels att dämpa pilotens handrörelser så att de inte blir för kraftiga (ju fortare piloten rör spaken desto mer motstånd möter handen).

□ Dels att ge piloten en information direkt i handen, beroende på de manövrer han gör med planet. Han behöver inte "vänta

ut" flygplanet rörelser för att få begrepp om hur planet lyder hans manövrer.

Konstruktionen bygger på att den mänskliga handen är ett mycket känsligt instrument. Via handen kan piloten uppfatta förändringar via spaken så snabbt som på 10-20 millisekunder.

Detta blir möjligt genom att momentmotorerna styrs av datorer. Motorerna arbetar olika, beroende på situationen. Det ger piloten möjlighet att styra med millimeternoggrannhet vid till exempel landning. Eller göra snabba manövrer, som stora svängar i hög fart, utan att behöva ta i för mycket.

### "Spelar liten roll"

Milton Mobärg vid Saab-Scania uppger för SvD att han anser att momentmotorerna spelar en mycket liten roll för styregenskaperna. Enligt Mobärg skulle de bara ha betydelse när piloten skall precisionssikta för att skjuta iväg planets vapen.

- Vi har aldrig velat ha de där

momentmotorerna, säger Milton Mobärg vidare.

Milton Mobärqs uppgifter motsägs av att konstruktionen av momentmotorer ingick i det kontrakt som slöts med tillverkaren Page. Nu uppger Mobärg att Page fått i uppdrag att titta på andra och enklare lösningar.

Några skäl att ändra på styrspaken innan JAS lyfter igen finns inte i dag, enligt Milton Mobärg.

En rakt motsatt uppfattning framförs i de dokument som konstruktören Lennart Nordström lämnat till haverikommissionen. En förenklad styrspak skulle orsaka stora problem för flygvapnets piloter, anser han:

"Det kan innebära att en vanlig förbandsförare ej klarar en precisionslandning på en vägbas under besvärliga förhållanden. Endera krävs en extremt skicklig flygförare eller för den vanliga förbandsföraren att det ej blåser."

Mikael Holmström

# Saab avfärdade konstruktör

Omedelbart efter JAS-planets första flygning i december 1988 kom en varning för brister i styrsystemet. Varningen utfärdades av den konstruktör som utformat planets styrspak, men ledde inte till några åtgärder inom Industrigruppen JAS.

Dokumentet ingår i haverikommissionens slutliga utredning av JAS-kraschen.

Just brister i JAS-planets styrsystem har tidigare pekats ut som huvudorsak till haveriet den 2 februari. SvD kan nu redovisa helt nya uppgifter om den styrspak som satt i det kraschade planet.

Redan i den första delrapporten från Statens haverikommission den 15 mars angavs att brister i själva styrspaken var en av de faktorer som "direkt påverkat" haverieroloppet.

Men sedan dess har Statens haverikommission fått tillgång till interna dokument från Saab-Scania. Dessa nya dokument visar: □ Att en varning för bristerna i spaken fanns före haveriet, redan efter första flygningen. □ Att spakens utformning ifrågasattes internt inom Saab-Scania. □ Dokumenten har lämnats över till kommissionen av den civilingenjör vid Saab-Scania som ursprungligen konstruerade styrspaken i JAS-planet. Han heter Lennart Nordström och har varit anställd vid Saabs flygdivision sedan 1948.

**"Har lyssnat"**  
- Vi har lyssnat på Nordström, säger Rune Lundin, utredningschef vid Statens haverikommission till SvD.  
Rune Lundin antyder att Saab-Scania hittills viftat bort Lennart Nordströms kritik alltför lättvindigt.



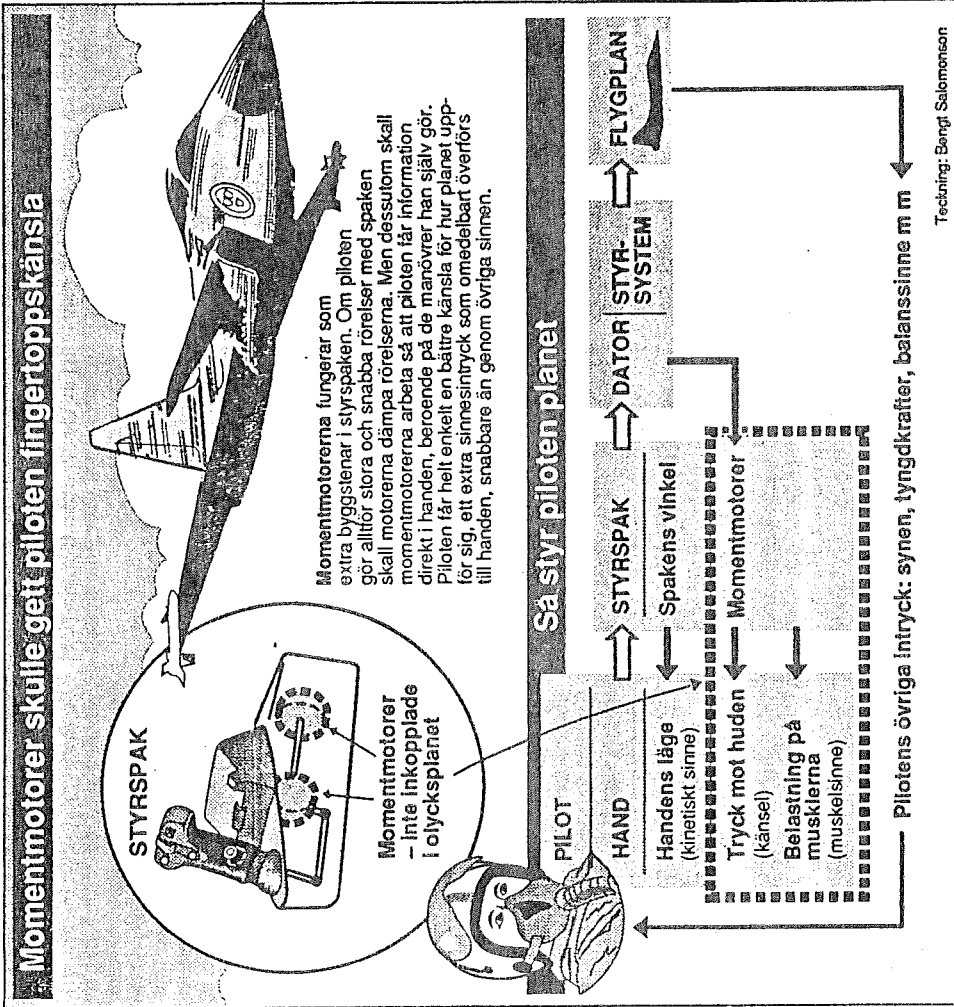
JAS har en minimal styrspak där den rörliga delen ryms i pilotens hand, i Vigen är däremot styrspaken halvmetern lång.



Profvflygaren Stig Holmström flyger i en av de simulatorer som använts för utvecklingen av styrspaken. Han styr med högerhanden och håller vänstra handen på planets dragkraftsreglage. Foto: Saab

- Antingen får man avfärda vad han säger på bättre grunder - eller göra en annan lösning i serieflygplanen, säger Rune Lundin.

Ett av dokumenten som nu finns hos kommissionen är en inlämnad skrivelse, ställd till VD i Industrigruppen JAS, Harald Schröder, samt flera tekniskt ansvariga för JAS inom Saab-Scania. Skrivelsen på tre maskinskrivna A-4-sidor är daterad måndagen den 12 december. Det var första arbetsdagen



Teknisk: Bengt Salomonson

Män har bedömt att man inte behöver göra mer än de åtgärder man redan vidtagit, säger Harald Schröder och understryker att det inte är ovanligt med olika uppfattningar i tekniska frågor.

De olika uppfattningarna om JAS-planets styrspak ledde för några år sedan till att Lennart Nordström kopplades bort från projektarbetet. Trots att styrspaken var hans idé, och trots att Saab fått patent på hans lösningar.

## "Nya haverier"

Själv vill Lennart Nordström inte uttala sig, utan hänvisar till de dokument han lämnat till haverikommissionen. I ett av dessa, daterat den 26 april i år, varnar Nordström för att bristerna i styrspaken kan leda till nya haverier.

"Dessa brister måste elimineras före fortsatta flygningar med flygplan 39 (JAS), då man annars får räkna med ytterligare tillbud eller haverier på grund av otillfredsställande manöveregenskaper."

Haverikommissionen har också fått ett åtgärdsprogram som Lennart Nordström tagit fram för att rätta till de brister han anser finns. Men inom haverikommissionen har man svårt att avgöra om det är konstruktören Nordströms eller Saabledningens lösning som är den rätta.

- Det handlar om den allra mest avancerade tekniken och vi har inga laboratorier för att avgöra vad som är rätt eller fel. Det är en mycket komplex fråga där olika konstruktörer har olika uppfattningar, säger Rune Lundin, och tillägger:

- Men vår uppfattning är att Nordströms synpunkter inte kan avfärdas. Vi kommer i vår slutrapport att föreslå att man tittar närmare på det här innan man beslutar om vilken styrspak som skall sitta i serieflygplanen.

Mikael Nordström

aldrig med Nordström, utan provflygningarna fortsatte och vid den sjätte flygningen havererade planet.

## Kopplades bort

Industrigruppen JAS VD Harald Schröder bekräftar för SvD att han känner till Lennart Nordströms skrivelse och hävdar att den behandlats på ett riktigt sätt: - Vi delar inte författarens bedömning på ett antal punkter.

efter JAS premiärflygning, fredagen den 9 december.

## Brevet en varning

- Brevet från den 12 december kan betraktas som en varning, säger Rune Lundin.

I skrivelsen hänvisar konstruktören Lennart Nordström till att Saabs chefprofvflygare efter första profvflygningen haft anmärkningar om "för hög spakänslighet" och fortsätter:

"Spaksystemet i profvflygplanet avviker från ett av mig tidigare rekommenderat och optimalt utförande."

I profvflygplanet fanns alltså en mer förenklad styrspak än vad konstruktören själv föreslagit, systemet var inte färdigbyggt (se artikel nedan).

Brevet slutar med ett förslag om ett möte så att "ändamålsenliga åtgärder" kan vidtas.

Något sådant möte hölls dock

# Appendix F

## Användarmanual

### Innehållsförteckning

1. Utrustning
2. Kontroll av anslutningar
3. Start av programmet
4. Huvudmenyn
5. Huvudmenyalternativ
6. Hur Ni skapar nya sampelfrekvenser
7. Hur Ni byter till andra ordningens lågpasfilter
8. Hur Ni komponerar ert eget uttryck för den återkopplade kraften
9. När behövs oscilloscop ?
10. Felsökning



## Användarmanual

OBSERVERA : SPÄNNINGEN TILL STATIVET MÅSTE VARA FRÅNSLAGEN OM NI INTE ÄR INNE OCH KÖR PROGRAMMET 'STYRSIM.EXE' !!

### 1. Utrustning

Det som behövs för att kunna köra programmet 'Styrsim.exe' som finns på disketten med namnet 'STYRSIM' är följande :

- \* persondator med 386-processor med tillhörande matematik-processor och 12-bitars AD/DA-kort
- \* stativ med styrratt, förstärkare och momentmotor
- \* gränssnitt för signaler mellan dator och förstärkare - finns i en svart låda märkt 'Signalkonditionering'

### 2. Kontroll av anslutningar

Innan spänningen slås till på stativet och innan start av programmet skall alla anslutningar kontrolleras. Det skall finnas förbindelse mellan :

- i) stativ (märkning: DATA) och datorn (via mångparskabel)
- ii) stativ (märkning: +15 V) och signalkonditioneringsutrustning (märkning: +15 V)
- iii) stativ (märkning: -15 V) och signalkonditioneringsutrustning (märkning: -15 V)
- iv) stativ (märkning: Rattvinkel 0-9V) och signalkonditioneringsutrustning (märkning: Rattvinkel 0-9V)

- v) stativ (märkning: Rattvinkel  $\pm 15V$ ) och signalkonditioneringsutrustning (märkning: Rattvinkel  $\pm 15V$ )
- vi) stativ (märkning: signaljord) och signalkonditioneringsutrustning (märkning: GND)
- vii) stativ (märkning: Kraft från dator 0-9V) och signalkonditioneringsutrustning (märkning: Kraft 0-9V)
- viii) stativ (märkning: VCS) och signalkonditioneringsutrustning (märkning: Kraft  $\pm 5V$ )

### 3. Start av programmet

Nu kan programmet 'Styrsim.exe' startas vilket ligger i underbiblioteket 'Simulate'. Markera med hjälp av piltangenterna ordet 'Simulate' och tryck därefter på vagnretur. Markera 'Styrsim.exe' och tryck vagnretur - programmet startas!

### 4. Huvudmenyn

Efter introduktionen och ljudslungan kommer huvudmenyn upp på skärmen. Där finns alla de alternativ som behövs för att kunna använda simulatören tillgängliga. Genom att skriva siffran som står framför det alternativ som önskas går man vidare i programmet. Nu kan spänningen till styrstativet slås på !

### 5. Huvudmenyalternativ

#### 1. Information om försöksuppställning och parametrar

...Här fås viktig information om simulatoruppställningen, tillgängliga parametrar och vad Ni speciellt bör tänka på när Ni simulerar. En ovan användare bör läsa igenom detta.

#### 2. Simulera

...Först får Ni några viktiga kontrolluppgifter som visar hur signalen ifrån

vinkelgivaren på ratten kommer att behandlas på sin väg genom datorn tills dess att ett vridande moment känns i ratten. Detta för att Ni skall få en chans att kontrollera att Ni ingenting glömt och en möjlighet att ångra Er.

Om Ni ånkrat Er och tryckt på tangenten 'R' hamnar Ni återigen i huvudmenyn varifrån Ni kan åtgärda vad Ni ånkrat.

Har Ni inte ånkrat Er och inte slagit till spänningen på styrstativet kommer ett ljud och en skylt på skärmen att upplysa Er om det.

Nästa steg innebär en kalibrering av ratten och hastighetsreglaget som sitter på den svarta lådan 'Gränssnitt'. Först förs ratten varsamt till det högra ändläget och sedan till det vänstra ändläget. Sen vrids hastighetsreglaget motsols till sitt ändläge och kalibreringen är klar. Kalibrering behöver Ni normalt endast göra denna första gång.

Om nu inte ratten är ställd helt horisontellt kommer ett ljud och en skylt att tala om vilket håll ratten skall vridas åt tills det är gjort. Detta för att åstadkomma en nollställning av den kraft som skickas ut till ratten när Ni startar simuleringen. När ratten ställts horisontellt skall den lämnas i det läget tills skärmen ritat ut grafiken som tillhör simuleringen. Rubbar man ratten ur sitt läge för tidigt får man omedelbart bakläxa enligt ovan.

Nu skall Ni befinna Er i en simulering och kan bekanta Er med instrumentpanelen och kartan som syns på skärmen. Vrider Ni på ratten kommer instrumenten att reagera och en vit markering ritas samtidigt upp ubåtens kurs på kartan.

Det är meningen att all hantering av ratten skall ske varsamt - med tanke på utrustningens ömtålighet - är man ovarsam kan man råka ut för att kraften i ratten plötsligt försvinner och grafiken ersätts med en skylt. Detta kan bero på en av följande orsaker :

- \* Ni har i förhållande till vald kraftåterkoppling på ratten med tillhörande parametervärden vridit alltför hastigt på ratten.

- \* Ni har vridit på ratten samtidigt som Ni vridit på hastighetsreglaget och därigenom råkat uppnå för stor förändring av den återkopplade kraften.
- \* Genom att för stora parametervärden används i beräkningen av den återkopplade kraften har inte förstärkarkortet på styrstativet orkat med och har därför automatiskt kopplat ifrån. Detta upptäcks genom att en liten röd lysdiod tänts på förstärkarkortet. Innan nästa simulering påbörjas måste spänningen slås av till styrstativet, lysdioden skall slockna och spänningen slås på igen.

Antingen Ni väljer att avsluta simuleringen eller om Ni blev "utkastade" hamnar Ni i huvud-menyn igen.

### 3. Omkalibrera eller återställning till defaultvärden

...Här erbjuds Ni två alternativ och väljer ett genom att trycka på rätt tangent , 'K' eller 'D'.

'K' innebär att Ni får göra en omkalibrering enligt ovan beskrivna kalibrering. Det kan vara aktuellt om Ni i första simuleringen märkte att hastighetsmarkeringen i sitt maxläge antingen steg över eller inte nådde upp till 20-knops markeringen, eller om rodervinkel-markeringen i endera ändläge antingen gick över eller inte nådde fram till 30-graders markeringarna.

Tryker Ni på 'D' sker automatiskt återställning till de utprovade värden som konstruktörerna ansett vara bra att utgå ifrån.

Trycker Ni på 'Return' återgår Ni till huvudmenyn utan att några förändringar gjorts.

#### 4. Välj vilka parametrar den återkopplade kraften skall bestå av

...Här kan Ni se vilka olika sammansättningar den till ratten återkopplade kraften kan ha. Det skall sägas att systemtrögheten är den term som åstadkommer känslan av tröghet när man vrider på ratten. Av stabilitetsskäl har konstruktörerna valt att ha med denna term i alla uttrycken. Ni väljer antingen ett nytt alternativ eller fortsätter med det föregående genom att trycka 'Return'.

#### 5. Välj konstanter till vald återkopplad kraft

...Valt återkopplingsalternativ visas i dess fullständiga form för att Ni skall kunna göra förnuftiga bedömningar hur Ni skall ändra de ingående parametrarna. Är återkopplingsalternativ 2 valt innebär det att endast två derivatatermer ingår i kraftuttrycket för den återkopplade kraften. Dessa ändras genom att välja alternativ 6 i huvudmenyn.

Att tänka på när Ni ändrar parametervärdena är att det kanske inte lönar sig med alltför stora hopp, eftersom Ni då riskerar att bli 'utkastade' ifrån simuleringen.

#### 6. Välj filter- och reglerkonstanter samt sampelfrekvens

...Innan insignalen behandlas matematiskt passerar den ett första ordningens lågpasfilter. Hur mycket signalen filtreras beror på hur konstanten alfa är satt. Ett lågt värde innebär liten filtrering men också liten fördröjning - ett ganska svårt motsatsförhållande i detta fall - och ett stort värde på alfa innebär hög filtrering men samtidigt lång fördröjning. Ni kan ändra alfa eller gå vidare genom att trycka 'Return'. Observera dock att om en trådlindad potentiometer används som vinkelgivare på styrstativet måste det finnas filtrering på insignalen p g a senare derivataverkan, eftersom insignalen är alltför brusig för att kunna deriveras direkt. Alfa bör då ej heller väljas mindre än 0.7 .

Innan utsignalen som påverkar momentet på ratten skickas ut ifrån datorn till förstärkaren på styrstativet passerar även den ett första ordningens lågpasfilter. Konstruktörerna har dock här valt att inte filtrera alls med hänsyn till den fördröjning som då orsakas. Det är naturligtvis lätt för Er att själv att prova med viss filtrering genom att ändra alfa.

Ni har möjlighet att byta från första ordningens lågpasfilter till ett andra ordningens i stället, antingen på ingången eller på utgången - se 'Hur Ni byter till andra ordningens lågpasfilter'.

Ni har fyra alternativa sampelfrekvenser att välja mellan - 25, 50, 75 och 100 Hz. 100 Hz används som default och det finns ingen anledning att ändra sampelfrekvens såvida Ni inte vill kontrollera vad som händer. Det finns en annan anledning till att ändra sampelfrekvens men det kräver att Ni gjort ändringar inuti själva programmet 'Styrsim.pas' - i funktionen 'FeedBack' - och där satt samman en så tidskrävande formel för den återkopplade kraften att sampeltiden inte räcker till. För att kunna upptäcka detta måste Ni ta hjälp av ett oscilloskop , som Ni kopplar till styrstativets utgång märkt 'Styrsignal', och granskar signalen med en sveptid på ca 10 ms. Då kan Ni se om de olika sampelintervallen är lika långa - är de det är allt väl och 100 Hz går bra att sampla med.

Ytterligare en anledning kan finnas till att Ni vill ändra sampelfrekvens - se vidare 'Hur Ni skapar fler sampelfrekvenser'.

För att påverka det konstruktörerna kallar 'Systemtröghet' kan Ni sedan ändra de tre konstanterna 'Td', 'N' och derivata förstärkning. Genom att Ni själva manipulerar de olika värdena kan Ni kanske uppnå bättre verkan än konstruktörerna - det är inte så enkelt. En viss försiktighet bör iaktas när Ni ändrar förstärkningen - små ändringar rekommenderas.

Giraccelerationen beräknas genom derivering av girhastigheten och påverkas genom ändring av konstanterna enligt ovan.

## 7. Avsluta

...Programmet avslutas.

**OBS !** Glöm inte att slå av spänningen till styrstativet. Om spänningen står på innan programmet har startats upp igen kan en spänning som motsvarar ett vridande moment på ratten innebära att ratten börjar rotera okontrollerbart. Detta innebär inte bara att hela styrstativet troligtvis totalhavererar utan vad viktigare är att användaren riskerar att bli allvarligt skadad !

## 6. Hur Ni skapar nya sampelfrekvenser

Det finns möjlighet att skapa fler möjliga sampelfrekvenser genom att i programmet 'Styrsim.pas' ändra i menyval 3 i proceduren 'Loop'. Där flyttar Ni kommentarparenteserna runt 'Write\_A\_And\_B\_Matrix' till 'Calibrate\_Or\_Default' istället. Därefter kompilerar Ni programmet och exekverar. Välj sen menyval 3 i huvudmenyn och nu förväntas Ni skriva in den nya sampelfrekvens Ni vill skapa samt de matrisvärden som behövs för simulatorns beräkningar. Matrisvärdena får Ni genom att köra 'MatLab'. Var noggrann under inskrivningen så sparar Ni tid. Matrisvärden från 1 - 20 knop behövs. Nu har Ni skapat en fil för den nya sampelfrekvensen och kan återställa programmet i sin ursprungliga form. Ändra även i proceduren 'New\_Filter\_And\_ControlConstants' där Ni skriver in den nya sampelfrekvensen i write-satsen 'Sampelfrekvens i Hz (25,50,75,100)' och i while satsen direkt efter den. Nu är den nya sampelfrekvensen klar att användas efter en ny kompilering av programmet. Ni bör även skapa en ny 'EXE'-fil.

## 7. Hur Ni byter till andra ordningens låpassfilter

Vill Ni använda ett andra ordningens låpassfilter på ingången istället för

ett första ordningens måste Ni ändra inuti programmet 'Styrsim.pas'. I proceduren 'Calculate\_And\_IO' byter Ni ut 'LowPass(AD(drChannel))' mot 'SecondOrderLowPass(AD(drChannel))'. I proceduren 'New\_Filter\_And\_ControlConst' byter Ni ut de ställen det står 'LowPassConst' till 'Ordn2LowPassConst' istället. Kompilera och exekvera sedan.

Samma sak går även att göra på utgången, dock ej samtidigt. Då ändrar Ni i funktionen 'FeedBack' där det står 'ROUND(Lowpass2(Value))' till 'ROUND(SecondOrderLowpass(Value))'. I proceduren 'New\_Filter\_And\_ControlConst' byter Ni ut de ställen det står 'LowPassConst2' till 'Ordn2LowPassConst' istället. Kompilera och exekvera sedan.

### 8. Hur Ni komponerar ert eget uttryck för den återkopplade kraften

Konstruktörerna har komponerat fem alternativa sätt att räkna ut den resulterande återkopplade kraften som verkar på ratten. Kanske har Ni ett eget förslag som Ni vill prova. Det innebär att lite ändringar måste göras inuti programmet 'Styrsim.pas'.

- i) I funktionen 'FeedBack' tar Ni bort kommentarparenteserna runt alternativ 6 och skriver in det uttryck Ni själv har komponerat. Använd så många av konstanterna K10 - K14 som Ni behöver, de övriga sätts till noll ändå.
- ii) I proceduren 'New\_FeedBack\_Alt' tar Ni bort kommentarparenteserna runt raderna som berör alternativ 6. Även några rader längre ner måste den övre gränsen i while-satsen ändras till 6 istället.
- iii) I proceduren 'New\_FeedBaConst' tar Ni bort kommentarparenteserna runt de rader som berör konstanterna K10 -K14. Justera de felgränser och den skalfaktor som proceduren



'ReadInputReal' behöver som inparametrar för att åstadkomma felmeddelande och att K10 - K14 skrivs ut som de hade värden mellan 0.0 - 10.0 .

Nu är programmet klart att kompileras och exekveras. Skapa sen även en ny 'EXE'-fil. Att tänka på är hur konstanterna skall väljas så att styrsignalen till ratten blir rimlig. Polariteten av signalen är viktig för att förhindra att ratten vrids åt fel håll - ta gärna oscilloskop till hjälp innan Ni kopplar in styrsignalen på styrstativet.

### 9. När behövs oscilloskop ?

Har potentiometern gått sönder och en ny måste installeras behövs oscilloskopet för att kalibrera potentiometern mot ratten - spänningen noll ut ifrån potentiometern skall motsvara horisontalläget för ratten.

Om Ni misstänker att sampeltiden inte räcker till kontrollerar Ni genom att mäta signalen märkt 'Styrsignal' på styrstativet. Med en sveptid på ca 10 ms kan Ni se om alla sampel är lika långa.

Om kraften i ratten inte motsvarar det Ni förväntade Er kan Ni kontrollera genom mätning på styrsignalen enligt ovan. Det kan ibland vara en fördel att kunna titta på styrsignalen innan Ni kopplat in den till styrstativet för att se om den är rimlig och överhuvud taget lönar sig att koppla in. Speciellt om Ni komponerat ett eget uttryck för den återkopplade kraften kan det vara väsentligt att titta på styrsignalen för att se om den t. ex. har rätt polaritet - annars kanske ratten vrids åt fel håll.

Oscilloskop använder Ni även vid felsökning av utrustningen, se 'Felsökning'.

## 10. Felsökning

**Fel :**

Ni kastas ur simuleringen direkt innan Ni ens rör ratten.

**Åtgärd :**

Ratten står inte riktigt horisontellt. Försök att rubba den lite åt något håll och prova igen.

**Fel :**

Ni kan knappt röra vid ratten innan Ni slängs ur.

**Åtgärd:**

Ändra konstantvärden för parametrarna ingående i den återkopplade kraften. Återgå eventuellt till defaultvärdena.

**Fel :**

Ni känner ingen kraft alls i ratten.

**Åtgärd:**

Kontrollera att alla anslutningar är gjorda.

Kontrollera att potentiometern är hel.

Ändra eventuellt konstanter värden.

Kontrollera att strömbegränsaren på förstärkarkortet utlösts (lysdioden tänd). Är så fallet måste förstärkningen minskas, antingen mjukvarumässigt eller genom att vrida på potentiometer R28, innan omstart. Slå ifrån spänningen på styrstativet och lysdioden slocknar.

**Fel :**

Ratten uppför sig alltför ryckigt.

**Åtgärd :**

Konstanten alfa för lågpasfiltret till insignalen kan vara satt till ett för lågt värde.

Konstanterna för 'Systemtröghet' är inte bra.

**Fel :**

Ratten upplevs som alltför gungig.

**Åtgärd :**

Konstanten alfa för lågpasfiltret till insignalen kan vara satt till ett för högt värde.

**Fel :**

Ratten upplevs som instabil.

**Åtgärd :**

Derivata förstärkningen till 'Systemtröghet' kan vara för liten.

**Fel :**

Mätarna på skärmen betar sig onormalt.

**Åtgärd :**

Kontrollera att potentiometern är hel.

Gör en omkalibrering.

**Fel :**

Mätarna för hastighet och rodervinkel når inte till sina ändlägen.

**Åtgärd :**

Är felet litet lämnas det därhän - det kan bero på elasticitet i gummiremmen ifrån motor till ratt. Är det stort - gör en omkalibrering.

## Referenser

- [1] Människa-maskin anpassning av ubåts manöverorgan  
Tomas Bergman, Anders Olson  
Institutionen för Reglerteknik, LTH, 1988
- [2] Human Factors in Engineering and Design  
Mark S. Sanders, Ernest J. McCormick  
6th ed. Mc Graw, Hill 1987
- [3] Utveckling av Styrspak för Flygplan  
Lennart Nordström, Saab Scania AB  
Industri Gruppen JAS
- [4] Revised Standard Submarine Equations of Motion  
J. Feldman, David W. Taylor  
Naval Ship Research and Development Center  
DTNSRDC/SPD-0393-09 June 1979
- [5] Appendix Mathematical Model of a Submarine  
Report 3242-3  
SSPA
- [6] Mecanics of Fluids  
W. J. Duncan m. fl.  
Edward Arnold 1970
- [7] Flight Safety Aerodynamics  
Ange Roed  
Aerotech 1974
- [8] Datorer i Automation, del 1 och del 2  
Gustaf Olsson  
Institutionen för Industriell Elektroteknik och Automation  
LTH, 1988
- [9] JAS-39-1-haveriet 1989-02-02  
Statens Haverikommission, 890315
- [10] Analog och Digital Mikroelektronik  
Institutionen för Tillämpad Elektronik, LTH, 1984
- [11] Turbo Pascal 1988  
Borland International

- [12] Programmera 286/386  
William H. Murray, Chris H. Pappas  
McGraw-Hill 1987
  
- [13] Modern Industriell Mätteknik 1989, Givare  
Grahm, Jubrink, Lauber  
Bokförlaget Teknikinformation 1989
  
- [14] LA-5600 Linear Amplifier Instruction Manual  
Motor and Control Systems Division  
Robbins and Myers
  
- [15] Motorer från Östergrens: Electro-Craft  
DC, Servomotorer, 1987
  
- [16] Artiklar ur SvD den 12 juni 1989  
"Styrspakens betydelse mycket omdiskuterad" och  
"Saab avfärdade konstruktör" av  
Mikael Holmström