

ISSN 0280-5316  
ISRN LUTFD2/TFRT--5696--SE

# Styrning av undervattensfarkost

Tao Ge  
Karl Johan Nilsson

Institutionen för Reglerteknik  
Lunds Tekniska Högskola  
November 2002



<b>Department of Automatic Control</b> <b>Lund Institute of Technology</b> <b>Box 118</b> <b>SE-221 00 Lund Sweden</b>		<i>Document name</i> MASTER THESIS	
		<i>Date of issue</i> November 2002	
		<i>Document Number</i> ISRN LUTFD2/TFRT--5696--SE	
<i>Author(s)</i> Tao Ge and Karl Johan Nilsson		<i>Supervisor</i> Tore Hägglund, LTH Peter Jonsson, Teknisk Geologi LTH	
		<i>Sponsoring organization</i>	
<i>Title and subtitle</i> Styrning av undervattensfarkost (Control of a submarine vehicle).			
<i>Abstract</i> This report summarizes a master-thesis project where the goal has been to control a submarine vehicle. The vehicle is used to take video films of the bottom of the sea. For this reason, it is desired to keep the vehicle at a constant distance over the sea bottom.  The work consist of a model-building phase, where a frequency-response experiment is performed on the vehicle. Based on this model, several different control strategies are suggested and tested through both simulations and real test on the submarine vehicle.			
<i>Keywords</i>			
<i>Classification system and/or index terms (if any)</i>			
<i>Supplementary bibliographical information</i>			
<i>ISSN and key title</i> 0280-5316			<i>ISBN</i>
<i>Language</i> Swedish	<i>Number of pages</i> 60	<i>Recipient's notes</i>	
<i>Security classification</i>			

The report may be ordered from the Department of Automatic Control or borrowed through:  
University Library 2, Box 3, SE-221 00 Lund, Sweden  
Fax +46 46 222 44 22      E-mail ub2@ub2.se



# Styrning av undervattensfarkost

Tao Ge  
Karl Johan Nilsson



## Innehållsförteckning

1	Introduktion.....	4
2	Beskrivning av undervattensfarkosten.....	5
2.1	Tekniska data.....	5
2.2	Styrprogrammet från GuildLine .....	6
3	Fysikalisk modell .....	7
3.1	Kabelns egenskap .....	7
3.2	Frekvensberoende processmodell.....	9
3.3	Roderdynamik .....	11
3.4	Linjäriserande variabelbyte .....	12
4	Framtagande av matematiska modeller .....	14
4.1	Identifieringsexperiment .....	14
4.2	Identifieringsuppbyggnad .....	14
4.3	Data från den trimmade farkostens process .....	14
4.4	Skattade Bodediagram .....	15
4.5	Parametriska linjära modeller.....	17
4.6	Identifiering av olinjära processer .....	20
4.7	Modell med växlande förstärkning.....	21
4.8	Olinjära modeller skattade med polynomansättning .....	21
4.9	Validering av Olinjära modeller .....	23
5	Regulatorstrukturer.....	24
5.1	Tillståndsestimering.....	24
5.2	PID-regulator.....	24
5.3	MRAS baserad på Lyapunovfunktion. ....	25
5.4	MRAS baserad på MIT struktur .....	26
6	Simulering i Simulink.....	28
6.1	Maximal roderhastighet .....	28
6.2	Maximalt Roderutslag.....	28
6.3	Maximalt avstånd från jämviktspunkt .....	28
6.4	Simulering med PID regulator och binär förstärkning .....	28
6.5	Simulering med adaptivt Liapunovbaserad MRAS-regulator.....	29
6.6	Simulering med MRAS enligt MIT-rule .....	31
7	Regulatorstest i Öresund .....	32
8	Hårdvara.....	34
8.1	Datorkraft .....	34
8.2	Motorservo .....	34
8.3	Signaler från GPS och styrboxen.....	35
8.3	Uppkoppling.....	35
9	Identifieringsprogram .....	36
9.1	Identifieringsprogram: "ident(selffrekvt).vi" .....	36
9.2	Identifieringsprogram: "ident.vi" .....	39
10	Regulatorprogrammet .....	41
10.1	Att använda programmet.....	41
10.2	Programmets uppbyggnad.....	42
11	Förslag till fortsatt arbete.....	43
11.1	Hårdvaror förslag.....	43
11.2	Mjukvaror förslag .....	43
	Referens .....	47
	Bilaga.....	48

# **1 Introduktion**

Vid avdelningen för Teknisk Geologi vid LTH finns en undervattensfarkost, som bl.a. används för att videofilma havsbotten. Farkosten, s k MiniBAT, är tillverkad av GuildLine i Canada med syfte att mäta temperaturen, syrehalt, klorofyll mm i olika vattennivåer. Farkosten var då designad att röra sig sinusformigt i vattnet med maximalt djup på 100m. Regleringen i originalprogrammet klarar dock inte att köra på konstant höjd över havsbotten.

Farkosten bogseras efter ett moderfartyg med hjälp av en 4 mm tjock vajer. Beroende på vilket djup man vill köra kan vajerlängden justeras manuellt. Med ökad längd fås farkosten att gå djupare och vice versa.

Farkosten saknar både formskrov och trimtankar varför farkostens horisontalläge enbart stabiliseras med hjälp av ett förinställt akter roder. Akterut finns även vertikala fenor som stabiliserar farkostens kurs. Man kan reglera djupet via ett höjdroder i fören, vilket är målet för vårt arbete.



## 2 Beskrivning av undervattensfarkosten

### 2.1 Tekniska data

Farkosten är baserad på en fackverkskonstruktion av stänger i rostfritt stål med  $\varnothing$  5 mm. På detta är diverse utrustningar, såsom sensorer och kameror och motorservo, monterade. Rodret är även detta tillverkat av rostfritt stål, genom en enkel böckning på dess undersida har det fått en profil som skall ge jämnare ändring av lyftkraften då rodret ändrar utslag. Man kan mekaniskt justera rodrets nollvinkel för att välja ett lämpligt djup som nollnivå. Vajern har en infästningspunkt som kan flyttas fram och tillbaka och ger farkostens dynamik ändrad komplexitet. Detta är en frihet som dock begränsas av styrmöjligheterna. Till farkosten är även en kabel,  $\varnothing$ 16 mm, ansluten vilken är fäst vid vajern och överför alla signaler samt elförsörjning åt roderservot och utrustning.

Till sensorerna hör: ekolod, djupmätare, sonar, rodervinkelgivare. Ekolodet ger information om farkostens höjd över havsbotten, sonaren ger farkostens position (djup, bäring och avstånd) relativt moderfartyget, djupmätaren som utgörs av en trycksensor ger information om farkostens djup.

Farkostens motorservo består av likströmsmotor och mekanisk utväxling. Servot styrs med konstanta spänningpulser under variabel tid och gör att rodret ändras med konstant hastighet antingen upp eller ned.

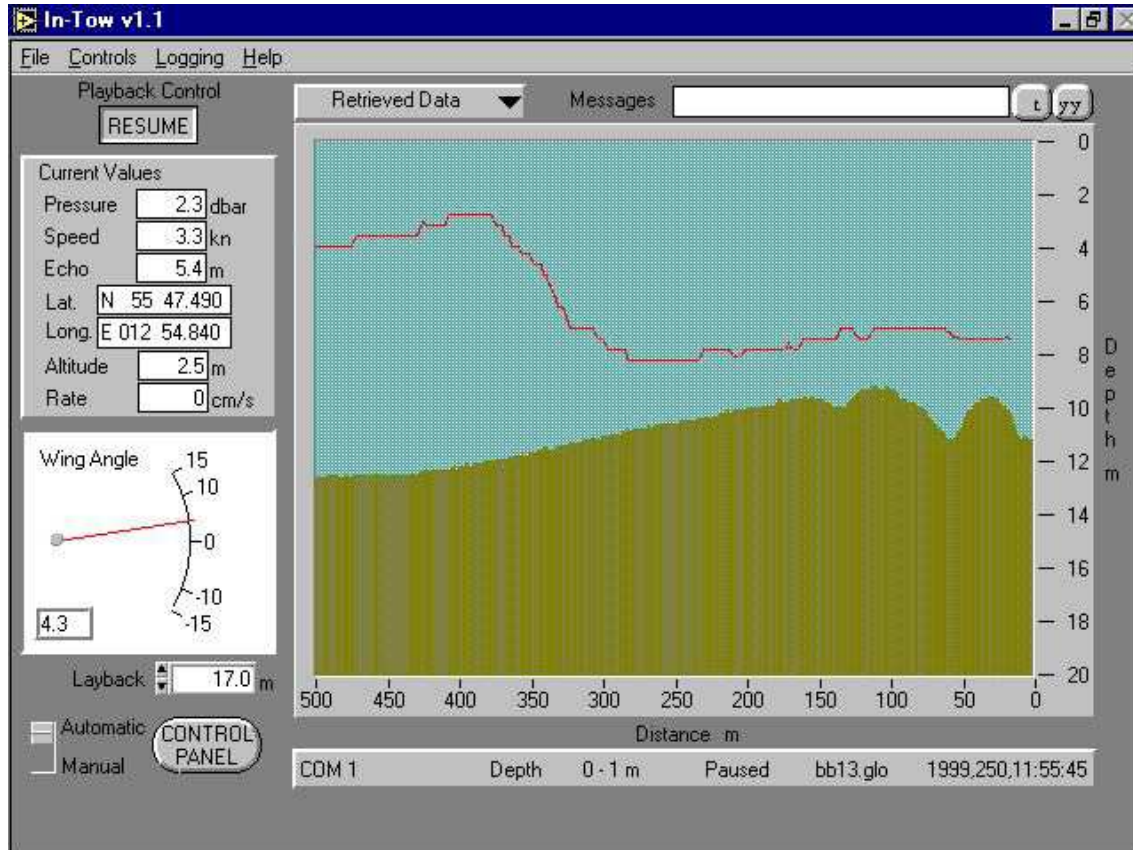
Teknisk data:	Sensorer/givare:
Längd: 1m.	Ekolod i Moderfartyg, Undervattensfarkost
Vikt: 30kg	Sonar: avstånd, bäring samt djup
Kabelvikt: 15kg	GPS i moderfartyg
Kabellängd: 50m	Rodervinkelgivare
Fart: 1 m/s	Djupmätare
Maxdjup: 30m (vid 50 m kabel)	



Figur 2.1 Visar farkosten som den ser ut utan utrustning.

## 2.2 Styrprogrammet från GuildLine

Farkostens ursprungliga styrprogram är gjort i LabView. Programmet är tyvärr en kompilerad version så man har ingen möjlighet att utifrån detta identifiera COM-portsinställningarna eller hur programmet var uppbyggt. Ett exempel hur programfönstret ser ut finns i figur 2.2.



Figur 2.2 Visar leverantörens LabView-program .

### 3 Fysikalisk modell

Olika teoretiska modeller kan sättas upp med varierande komplexitet, där farkostens dynamik även beror på kabelns dynamik.

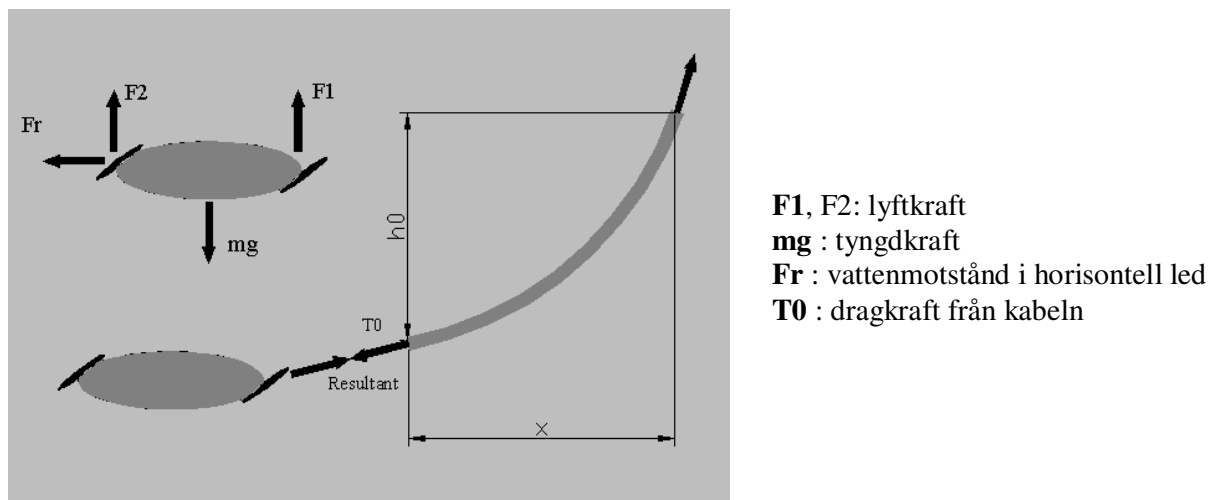
Kabeln och farkosten lyfts av vattnets motstånd som är proportionellt mot vattnets relativa hastighet. I havet strömmar vattnet ibland lika fort som farkosten rör sig, då står farkosten stilla relativt vattnet och förlorar lyftkraft. På ytan ger vågorna störningar som direkt påverkar farkosten och ger den ryckvis acceleration. Kabeln slackar då först och spänns sedan upp.

Det enklaste modell man kan tänka sig är att farkosten planflyger med konstant hastighet och regleras upp och ned med hjälp av rodret. Denna modellen gäller åtminstone vid små rörelser.

I verkligheten kan farkosten dock uppvisa stallningseffekter vid uppstigning, och acceleration vid nersjunkning. Då måste fler sensorer monteras för att mäta t ex farkostens lutning och vattnets strömning genom farkosten. Sensorer som är extra monterade på farkosten är ekolod och vattenrelativhastighetsmätare. Dessa sensorer arbetar asynkront med genomsnitt väntetid kring en sekund.

#### 3.1 Kabelns egenskap

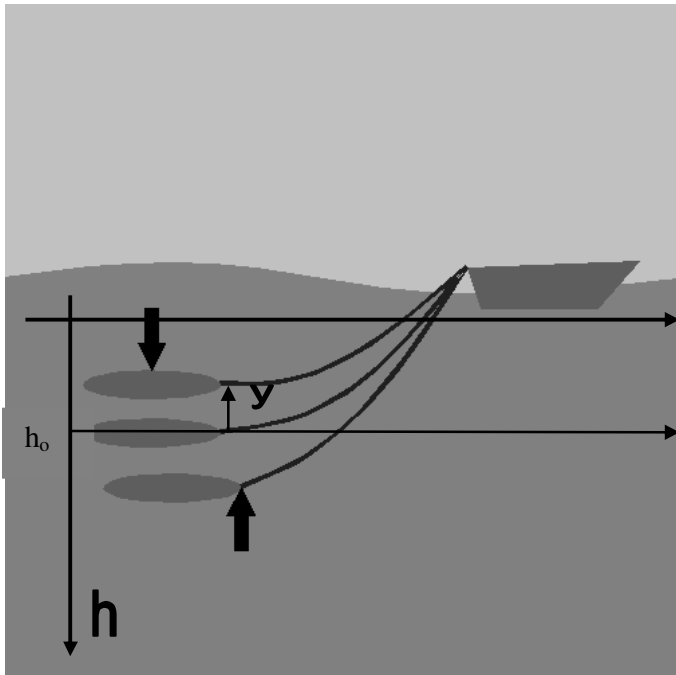
Kabeln har en stabiliserande effekt på farkosten med jämviktspunkt där kabelns uppspänning motsvarar resultanten av farkostens krafter se figur 3.1.



Figur 3.1 Visar uppdelning av krafter på farkost och kabel, samt resultant.

På farkosten finns följande krafter: Tyngdkraft, vattenmotstånd, lyftkraft från de båda rodren. Vid jämvikten motsvarar resultanten den kraft som kabeln måste bidra till farkosten.

Figur 3.2 visar att kabeln drar upp farkosten om farkosten sjunker, men farkosten får samtidigt en minskad lyftkraft när den är på väg upp.



**Figur 3.2** Visar hur kabeln stabiliserar farkosten autonomt. Det är troligt att kabeln inte påverkar farkosten linjärt.

Sambandet kan tecknas som ett autonomt system enligt (3.1) där  $h$  är verkligt djup,  $y$  är avvikelse från jämviktsdjupet  $h_0$  då rodervinkeln är noll.

$$h = h_0 - y \quad (3.1)$$

$$\dot{y} = f(y) \quad (3.2)$$

$$f(y) < 0, y > 0 \quad (3.3)$$

$$f(y) > 0, y < 0 \quad (3.4)$$

$$\lim_{t \rightarrow \infty} y = 0 \quad (3.5)$$

$$\Rightarrow \lim_{t \rightarrow \infty} h = h_0 \quad (3.6)$$

$f$  är en funktion som är negativ för positiva  $y$ -värden och positiv för negativa värden. Fixpunkten fås då tidsderivatan är noll. Funktionen  $f$  kan uppskattas på olika sätt, där polynomansättning är ett, vilket kan tecknas enligt:

$$f(y) = -a_1 y - a_2 y^2 - a_3 y^3 \quad (3.7)$$

Den enklaste approximationen av  $f$  är den linjära termen, där parameter  $a_1$  väljs positiv. Se (3.8).

$$f(y) = -a_1 y \quad (3.8)$$

Då farkosten används måste man släppa ut så mycket kabel att referensdjup och jämviktsdjup överensstämmer med varandra.

Kabeln kan modelleras som en inspänd kabel, enl. figur 3.1, med differentialekvationen (3.9). Verklighetens kabel är mer komplex då vattnets motstånd ger en lyftkraft som inte är jämt fördelad längs hela kabeln.

$$\frac{d^2 y}{dx^2} = \frac{\rho}{T_0} \cdot \frac{dl}{dx} \quad (3.9)$$

där

$T_0$  är den kabelspänning som farkostens vattenmotstånd ger vid aktuell hastighet.  
 $\rho$  är kabelns vikt per längdenhet kompenserat med lyftkraften.  
 $l$  är kabelns längd.

Differentialekvationen har, om resultanten är parallell med farkosten, jämviktsdjup  $h_0$  (räknat från ytan) enligt (3.10), där  $x$  är avståndet till båten:

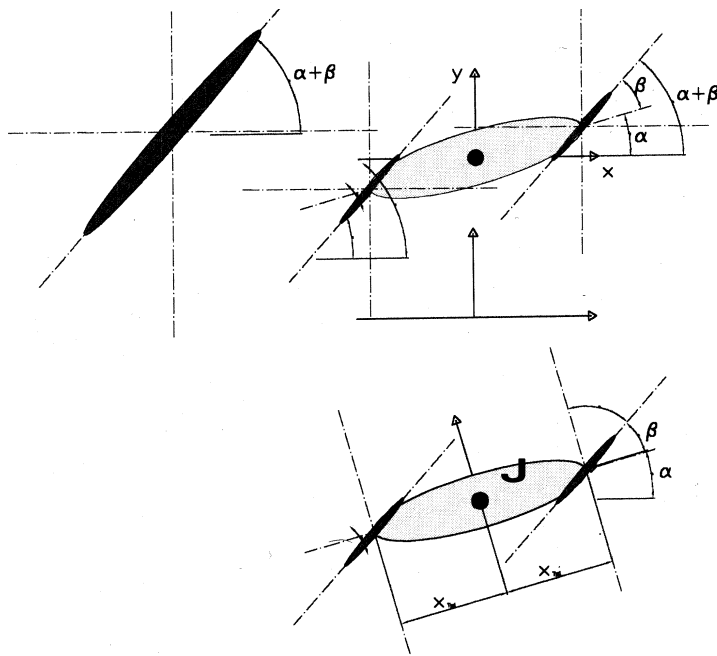
$$h_0 = T_0 / \rho \cdot (\cosh(\rho x / T_0) - 1) \quad (3.10)$$

$$l = T_0 / \rho \cdot (\sinh(\rho x / T_0)) \quad (3.11)$$

### 3.2 Frekvensberoende processmodell

Ett sätt att välja modell är att beskriva farkosten från spänningsmatning till djup ( $G_{yu}$ ). Då hade motorservots dynamik ingått i modellen. Process som valdes i arbetet är betraktad från roder till djup ( $G_{y\beta}$ ). Se figur 3.3. På så sätt kunde man lättare förstå processen samtidigt som processdynamiken förenklades. Man får dock ytterligare en process mellan rodervinkel och spänningsmatning som måste behandlas för sig, det gjordes med kaskadreglering.

Man kan tänka sig att processen inte är så komplicerad för låga roderfrekvenser, varför kommer endast modeller av högst andra ordningen att undersökas.



$\alpha$  : farkostens lutning  
 $\beta$  : rodervinkel

Figur 3.3 Visar Variabelbeteckningar. Det är den totala vinkeln som avgör lyftkraften, inte enbart roderutslaget i sig.

- **Första ordningens system**

Vattnets motstånd kan anses linjärt med proportionalkonstanten  $\zeta$  i y-led. Rodrets lyftkraft kan anses proportionell mot hastigheten  $v$ . Kabeln påverkar farkosten med en faktor proportionell mot djupet skalad med konstanten  $a$ .

$$\uparrow F = -\zeta \cdot \dot{y} - \lambda \cdot y + \gamma \cdot v \cdot \beta(t) = 0 \quad (3.12)$$

$$\dot{y} = -ay + k_1 \cdot v \cdot \beta(t) \quad (3.13) \quad \text{där } a = \frac{\lambda}{\zeta} \text{ och } k_1 = \frac{\gamma}{\zeta}$$

$$y(s) = \frac{k_1 \cdot v}{s + a} \cdot \beta(s) \quad (3.14)$$

där

$y$  - farkostens djup

$\beta$  - rodervinkel

$v$  - är hastigheten igenom vattnet

$\lambda$  - är en parameter som bestäms av kabelns inverkan på farkosten.

$\zeta$  - är en hydrodynamisk faktor.

$\gamma$  - är en faktor som beror på rodrets lyftförmåga

- **Andra ordningens system**

Farkostens masströghet och vattnets motstånd bidrar till farkostens dynamik

$$\uparrow F = m \cdot \ddot{y} = -\zeta \cdot \dot{y} - \lambda \cdot y + \gamma \cdot v \cdot \beta(t) \quad (3.15) \quad m \text{ är farkostens massa}$$

$$\ddot{y} = -a_1 \cdot \dot{y} - a_2 \cdot y + k_2 \cdot v \cdot \beta(t) \quad (3.16) \quad \text{där} \quad a_1 = \frac{\zeta}{m}; \quad a_2 = \frac{\lambda}{m}; \quad k_2 = \frac{\gamma}{m}$$

$$y(s) = \frac{k_2 \cdot v}{s^2 + a_1 \cdot s + a_2} \cdot \beta(s) \quad (3.17)$$

Fasvridningen mellan  $\beta$  och  $y$  för denna modell går från 0 till  $-180$  grader.

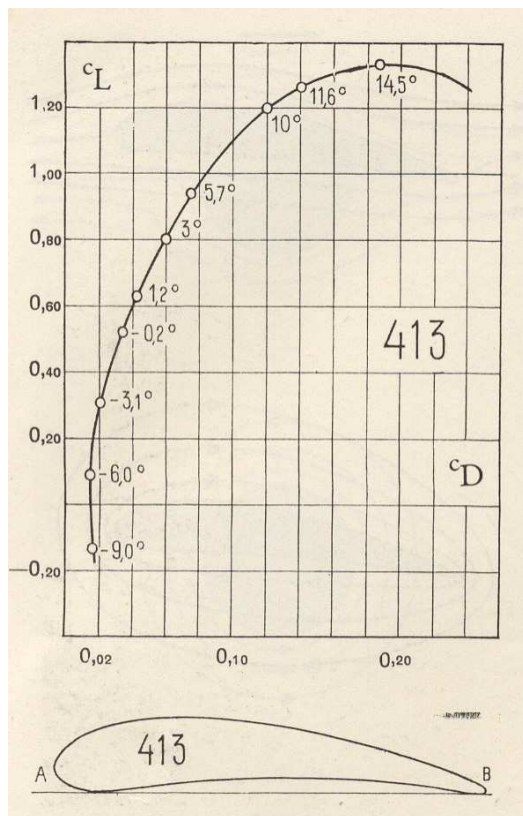
Man kan tänka sig att farkosten har vridtröghetsmoment och vrider därför till den riktningen som är parallell med farkostens utbredningsriktning för att få minst vattenmotstånd. Modellen blir i så fall av tredje ordningen. Men eftersom vridningen stabiliserar sig mycket snabbt kan effekten försummas.

Vid den slutliga identifieringen visades det att med nuvarande motorservo endast en fasförskjutning ner till  $-90$  grader kunde uppnås. Med snabbare roder hade säkert ytterligare komplexitet uppvisats.

### 3.3 Roderdynamik

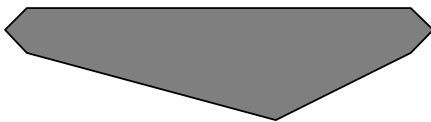
Lyftkraften från rodet är inte exakt proportionell mot roderutslaget utan av ett olinjärt förhållande som bestäms av rodrets profil. Att bestämma detta samband skulle vara möjligt i en bassäng med strömmande vatten för olika hastigheter och roderutslag. Vidare har rodet olika motstånd genom vattnet beroende av dess lutning.

I figur 3.4 ses ett roder för flygplan i profil och dess lyftkraft resp. luftmotstånd för olika lutningar. Ett motsvarande diagram skulle fås för ett roder i vatten. Exakt hur detta beroende ser ut kunde inte bestämmas då erforderlig utrustning inte fanns tillgänglig.



Figur 3.4 visar lyftkraft resp. luftmotstånd för ett aerodynamiskt roder, för olika lutning

Verklig roderprofil för vår farkost visas i figur 3.5:



Figur 3.5 Rodret är gjort i rostfritt stål med bockad kant på undersidan.

### 3.4 Linjäriserande variabelbyte

Då det visades i inledningen att den förväntade roder- och kabel- dynamiken skulle vara olinjär togs lämpliga metoder fram för att åtgärda problemet.



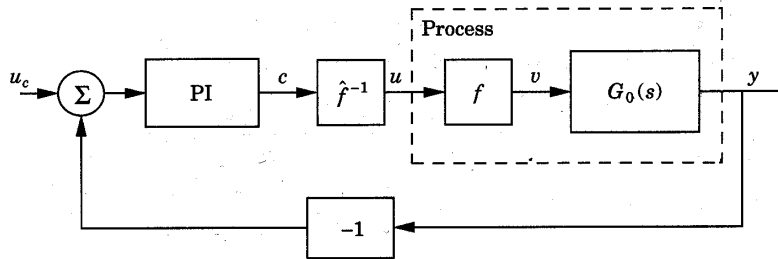
Om man skattar den funktion  $f$  som framkallar olinjäriteten hos processen kan systemet linjäriseras enligt figur 3.6. Om man multiplicerar en skattad invers funktion  $\hat{f}^{-1}$  med insignalen  $c$  (enlig 3.21) kan ett system (3.18) linjäriseras. Insignalen  $u$  motsvarar då det roderutslag som ger den önskade linjära styrsignalen  $c$ . Approximationen av den olinjära processen visas i figur 3.7.

$$\dot{y} = -ay + b_0 f(u) \quad (3.18)$$

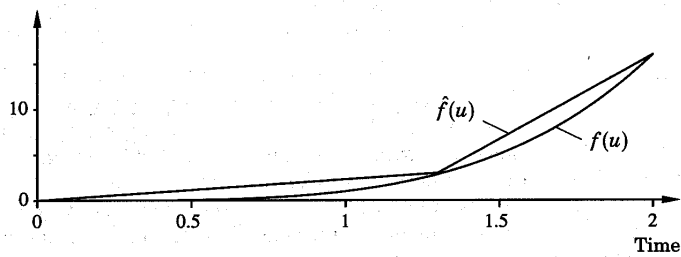
$$\dot{y} = -ay + b_0 v \quad (3.19)$$

$$v = f(u) \quad (3.20)$$

$$u = f^{-1}(c) \quad (3.21)$$



Figur 3.6 Visar linjäriserande kompensering.



Figur 3.7 Approximation av den olinjära funktionen  $f(u)$ .

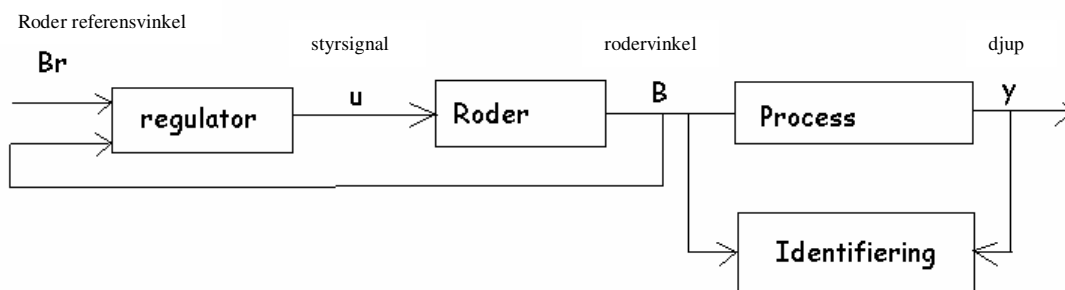
## 4 Framtagande av matematiska modeller

### 4.1 Identifieringsexperiment

Identifieringsexperimentet utfördes utanför hamnen i Landskrona. Vid experimentet samlades data för att kunna skatta farkostens egenskaper. Resultatet visade en kraftigt olinjär process, och farkosten var obalanserad. Den olinjära processen skattades parametriskt med olinjära metoder i kontinuerlig tid.

Viktbalansen för farkosten visade sig vara förskjuten bakåt. I samband med regulatortestet (utanför Limhamn) utfördes en balansering genom att öka flytkraften akterut. Farkosten blev därmed mer linjär och denna process identifierades parametriskt med linjära metoder i diskret tid (regression).

### 4.2 Identifieringsuppbyggnad



**Figur 4.1** Visar skiss av identifieringsuppställning. Det var förhållandet mellan djup och rodervinkel som identifierades.

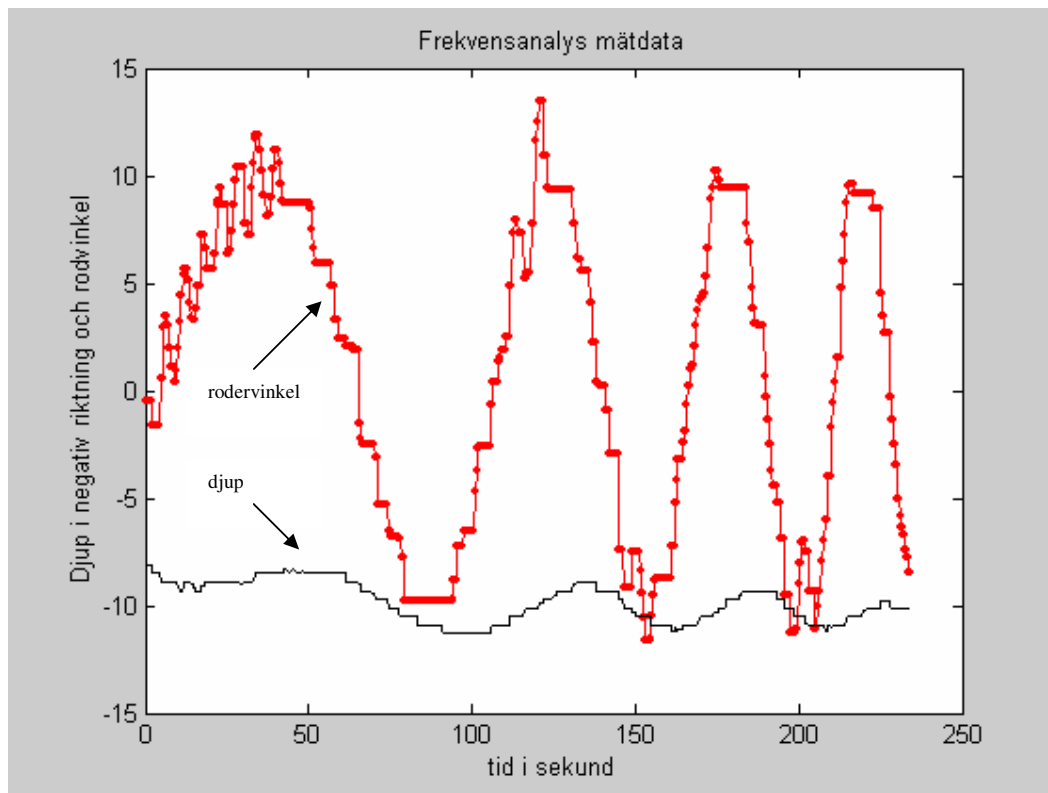
I figur 4.1 visas hur identifieringsuppbyggnaden såg ut. Regulatorn reglerade rodret till rätt referensvärde med hjälp av on/off reglering. Utsignalen från roderservot och farkostens utsignal mättes och användes vid identifieringen.

En kabellängd på 20m valdes vid experimentet. Farkostens dynamik i den kabeln undersöktes (farkosten tillsammans med kabeln betraktas i fortsättningen som en process). Insignalen var rodervinkel. Utsignalen var farkostens djup.

### 4.3 Data från den trimmade farkostens process

I figur 4.2 är mätdata av spektralanalys från experimentet utritad. Djupet, räknat från havsytan, visas där med negativa värden. När sedan processen skall identifieras drogs jämviktsdjupet (ca 10m) från djupdata.

Man ser att rodervinkeln inte var en perfekt sinussignal på grund av roderregulatorns begränsade styrmöjligheter. Detta är inte något problem för identifieringen, däremot kanske för den slutliga regleringen.



Figur 4.2 Djupet för farkosten och motsvarande rodervinkel

#### 4.4 Skattade Bodediagram

Två typer av experiment gjordes. De är anpassade för: *frekvensvarsanalys* respektive *spektralanalys*. Vattnets relativhastighet var cirka 1 m/s under hela identifieringsexperiment.

##### **Frekvensvarsanalys:**

Vid experimentet mättes svaret för olika frekvenser genom att styra rodret med sinussignaler med olika frekvens (4.2), där den teoretiska utsignalen från det linjära systemet ges av (4.1).

$$y(t) = |H(j\omega)| \cdot u \cdot \sin(\omega t - \arg(H(j\omega))) \quad (4.1)$$

$$u(t) = u \sin(\omega t) \quad (4.2)$$

$$s_T = \int_0^T y(t) \cdot \sin(\omega t) dt \quad (4.3)$$

$$c_T = \int_0^T y(t) \cdot \cos(\omega t) dt \quad (4.4)$$

Med (4.3) och (4.4) ger skattningen av överföringsfunktionen  $H(j\omega)$ :

$$|H(j\omega)| = \sqrt{s_T^2 + c_T^2} \quad (4.5)$$

$$\arg(H(j\omega)) = \arctan\left(\frac{c_T}{s_T}\right) \quad (4.6)$$

På grund av tidsbrist kunde inte alla frekvenser testas i frekvensvarsanalysen. I frekvenssvaret testades 6 perioder för varje frekvens med följande periodtider: 10s, 20s, 30s, 40s, 50s, 60s, 70s samt 90s. Dessa var för få till antalet (ger för smalt frekvensband) för att bestämma ett bodediagram.

### **Spektralanalys:**

Genom att excitera systemet med ett nedåt gående frekvenssväv, sk Chirp-signal, kunde ett bodediagram bestämmas med hjälp av spektralanalys från kovariansfunktioner och DFT (diskret fouriertransform). Kovariansfunktionerna fouriertransformeras först enligt (4.7.1) och (4.7.2).

$$S_{yu}(j\omega) = F(C_{yu}(\tau)) \quad (4.7.1)$$

$$S_{uu}(j\omega) = F(C_{uu}(\tau)) \quad (4.7.2)$$

$$\hat{Y} = F_{DFT}(y_k) \quad (4.8.1)$$

$$\hat{U} = F_{DFT}(u_k) \quad (4.8.2)$$

Överföringsfunktionen fås i princip genom (4.9.1) och (4.9.2):

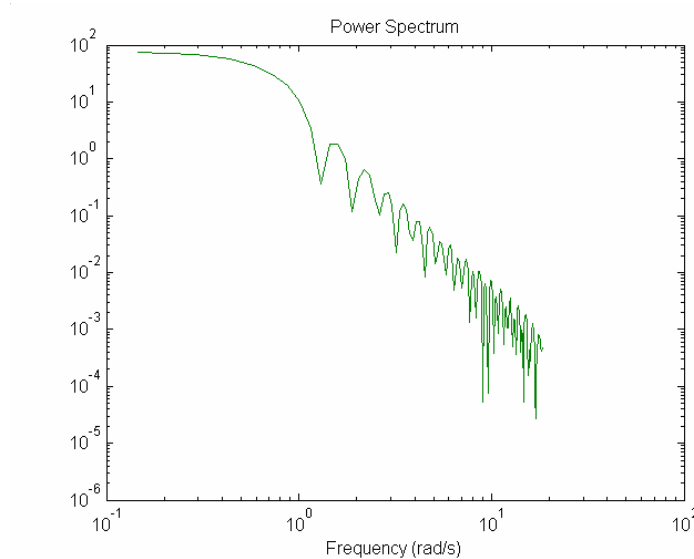
$$H(j\omega) = \frac{S_{yu}(j\omega)}{S_{uu}(j\omega)} \quad (4.9.1)$$

$$H(j\omega) = \frac{\hat{Y}(j\omega)}{\hat{U}(j\omega)} \quad (4.9.2)$$

Det som gjordes innan estimering var att dra bort mätatamedelvärde och ekvidistant omsampling från mätdata.

För att veta om det fanns tillräcklig information i data undersöktes kvadratiska koherensspektrumet för detta (figur 4.4), där ser man att ungefär upp till 0.5 rad/s skulle gå att identifieras. Dock kunde rodet maximalt styras med 30 sekunders periodtid vid fullt utslag som motsvarar 0.2 rad/s. Man hade kunnat skapa högre frekvenser men med lägre excitering.

Lägre frekvenser än 0.1 rad/s ansågs kunna orsaka olinjäritet hos utsignalen och blev därför inte undersökta.



**Figur 4.4** Visar att upp till ca 0.5 Rad/s bör processen vara identifierbar

## 4.5 Parametriska linjära modeller

Vid identifiering provades olika modeller av olika gradtal. För samtliga utfördes nedanstående punkter (se Rolf Johansson: System Modeling and Identification för noggrannare beskrivning):

- Överföringsfunktion
- Akaike final prediction error
- Bodediagram för modellen
- Autokorrelation  $r_{\epsilon\epsilon}(\tau)$
- Korskorrelation  $r_{\epsilon u}(\tau)$
- Omvandling till kontinuerlig tid med *zero order hold*
- Validering med verklig data

där Akaike är en metod att fastställa minimal modellordning. Auto- och Korskorrelationstestet skall helst ligga inom 98 % konfidensintervall. Nya modeller skapades genom att sänka modellordningen vilka sedan jämfördes med de förra.

### Slutlig linjär modell av lägsta ordning:

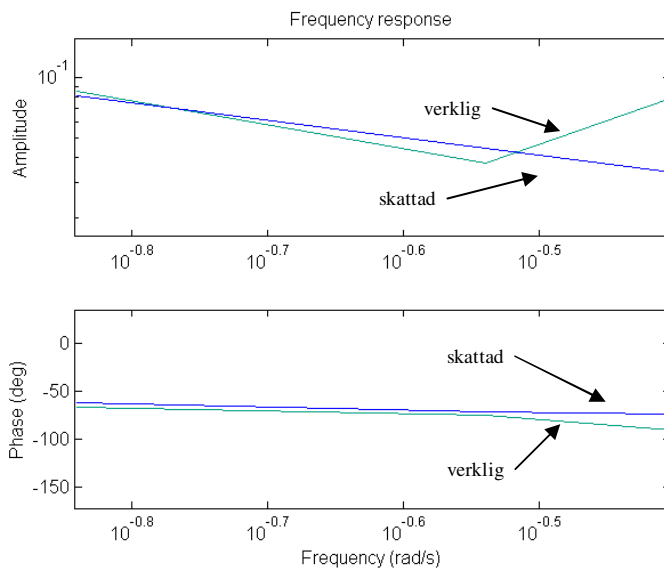
Genom matlabprogrammet: *ident* kunde processmodellen bestämmas. Den slutliga diskreta modellen av lägsta ordning bestämdes till:

$$G(z) = \frac{0.0022}{z - 0.9886} \quad (4.9.3) \quad \text{med samplingstid } 0.17 \text{ s}$$

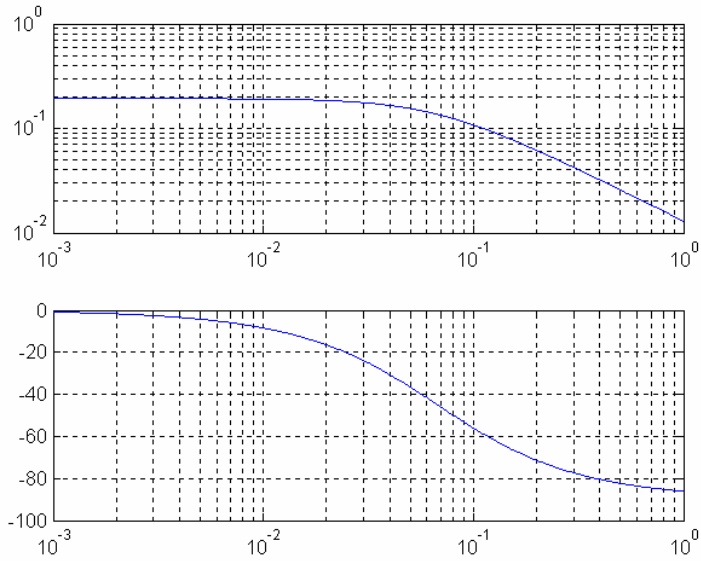
Motsvarande tidskontinuerliga modell av lägsta ordning med "zero order hold" kunde fås fram med matlab kommandot: *d2c*. Resultatet är:

$$G(s) = \frac{0.013}{s + 0.06744} \quad (4.9.4)$$

En jämförelse mellan modellen och verklig process i det identifierade frekvensbandet visas i figur 4.5. Det fullständiga bodediagrammet för modellen visas i figur 4.6.

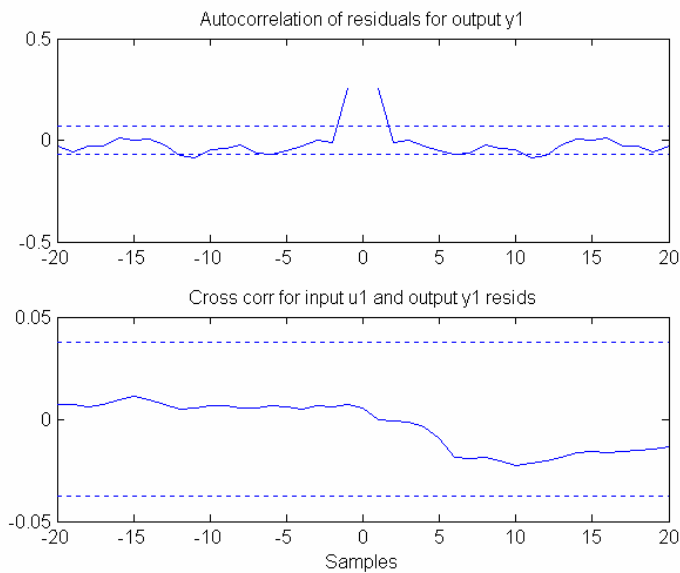


**Figur 4.5** Bodediagram för skattad process från spektralanalys och verklig process. Frekvensbandet är från 0.14rad/s till 0.28rad/s.

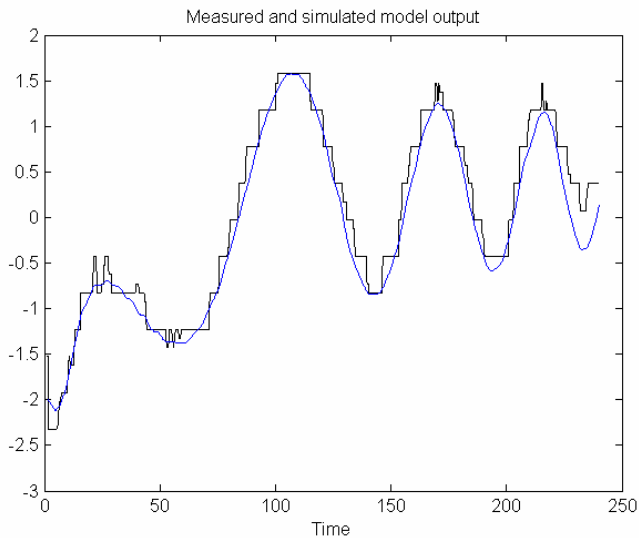


**Figur 4.6** Bodediagram för skattad linjär modell i hela frekvensområdet i rad/s.

Korrelationerna kom dock inte att helt att ligga innanför 98 % intervallet för den enklaste modellen. Se figur 4.7.



**Figur 4.7** Autokorrelation ( $r_{\hat{e}\hat{e}}$ ) och korskorrelation ( $r_{\hat{e}u}$ ).

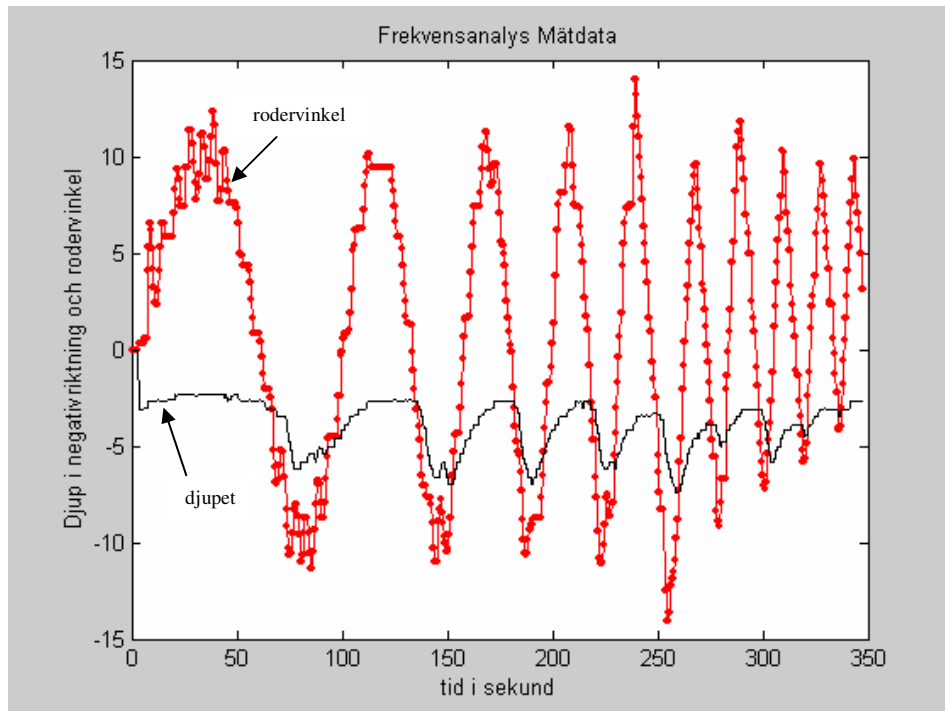


**Figur 4.8** Visar simulerat (kontinuerlig linje) och verkligt data samplat med ca 1 sekund. Tidsaxeln anger sekunder.

Genom *valideringen* kunde den slutliga ordningen fastställas. I figur 4.8 ser man att simulerad modell följer verklig process. Eftersom resultaten av olika modeller inte ändrades så mycket föredrogs en minimerad modell istället för en av högre gradtal.

#### 4.6 Identifiering av olinjära processer

Figur 4.9 visar data från den otrimmade processen och man ser tydligt att farkosten har större benägenhet att sjunka mot större djup än att stiga mot ytan.



**Figur 4.9** Visar rodervinkel & djup, där rodret går från -10 till 10 grader.



## 4.7 Modell med växlande förstärkning

Då processen var kraftigt olinjär kan det innebära problem för regleringen. För att kunna testa regulator i simulering skapades först en olinjär processmodell av första ordningen.

Genom att låta förstärkningen variera beroende av insignalens tecken får man en olinjär processmodell. Processen har då två olika förstärkningar (4.10), här kallas det för binär förstärkning. En sådan modell är mer nära verkligheten och enkel att skapa. I (4.11, 4.10) visas linjäriserande term till regulatorn, där  $c$  (enligt figur 3.6) är den linjära signalen från regulatorn och  $u$  är insignal till processen.

$$\dot{y} = -ay + \begin{cases} b = 2b_0, u > 0 \\ b = b_0, u < 0 \end{cases} \cdot u \quad (4.10)$$

$$u = \begin{cases} \left( \frac{1}{2}, c > 0 \right) \\ \left( 1, c < 0 \right) \end{cases} \cdot c \quad (4.11)$$

## 4.8 Olinjära modeller skattade med polynomansättning

Det finns ytterligare dynamik som framkom av rodervinkelns rippel som inte skulle kunna förklaras enbart med binär förstärkning. Det skulle därför vara önskvärt att finna en fullständig modell som kunde svara även för dessa effekter. Man kan tänka sig en processmodell med komplicerad struktur som kan approximeras med polynom.

Olinjära modeller skattade med minsta kvadrat enligt kap.12 i Rolf Johanssons: System Modeling and Identification. Först filtreras de olika olinjära signaler med ett lågpasfilter  $\lambda$  (4.13) som har brytfrekvensen  $1/\tau$ , sedan genom regression skattas parametrar (enlig 4.21) får man polynomen. Filtret har egenskapen att släcka ut högfrekventa störningar. Parametrarna anges av (4.17) och transformeras genom (4.22) till (4.37).

$$\dot{y} = -a_1 \cdot y - a_2 \cdot y^2 - a_3 \cdot y^3 + b_0 \cdot u + b_1 \cdot u^2 + b_2 \cdot u^3 \quad (4.12)$$

$$\lambda = \frac{1}{\tau \cdot s + 1} \quad (4.13)$$

$$s = \frac{1}{\lambda \cdot \tau} \cdot (1 - \lambda) \quad (4.14)$$

$$\frac{1}{\lambda \cdot \tau} \cdot (1 - \lambda) \cdot y = -a_1 \cdot y - a_2 \cdot y^2 - a_3 \cdot y^3 + b_0 \cdot u + b_1 \cdot u^2 + b_2 \cdot u^3 \quad (4.15)$$

$$y = (1 - a_1 \cdot \tau)[\lambda y] - a_2 \cdot \tau[\lambda y^2] - a_3 \cdot \tau[\lambda y^3] + b_0 \cdot \tau[\lambda u] + b_1 \cdot \tau[\lambda u^2] + b_2 \cdot \tau[\lambda u^3] \quad (4.16)$$

$$y = \alpha_1[\lambda y] + \alpha_2[\lambda y^2] + \alpha_3[\lambda y^3] + \beta_0[\lambda u] + \beta_1[\lambda u^2] + \beta_2[\lambda u^3] \quad (4.17)$$

$$y(n) = \varphi^T(n) \cdot \theta + e(n) \quad (4.18)$$

$$\theta = \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \beta_0 \\ \beta_1 \\ \beta_2 \end{pmatrix} \quad (3.19) \quad , \varphi^T = ([\lambda y] \quad [\lambda y^2] \quad [\lambda y^3] \quad [\lambda u] \quad [\lambda u^2] \quad [\lambda u^3]) \quad (4.19)$$

$$\Phi = \begin{pmatrix} \varphi^T(0) \\ \vdots \\ \varphi^T(N) \end{pmatrix} \quad (4.20)$$

$$\Rightarrow \hat{\theta} = (\Phi^T \cdot \Phi)^{-1} \cdot \Phi^T \cdot Y \quad (4.21)$$

$$\varepsilon_k = y_k - \varphi_k^T \cdot \hat{\theta}$$

$$a_1 = \frac{1}{\tau}(1 - \alpha_1) \quad (4.22)$$

$$a_2 = \frac{1}{\tau}(-\alpha_2) \quad (4.23)$$

$$a_3 = \frac{1}{\tau}(-\alpha_3) \quad (4.24)$$

$$b_0 = \frac{1}{\tau}(\beta_0) \quad (4.25)$$

$$b_1 = \frac{1}{\tau}(\beta_1) \quad (4.26)$$

$$b_2 = \frac{1}{\tau}(\beta_2) \quad (4.27)$$

Ett fiktivt olinjärt roder ansätts för att svara för alla olinjäriteter som t.ex. kabeln orsakade (4.28). Detta ger en enklare modell som begränsar antalet parametrar, men kan naturligtvis inte skatta processen fullt ut. Eventuella fel i kalibreringen av y-skalan skulle annars ge fel som växer kubiskt vid polynom av gradtal 3.

$$\dot{y} = -a_1 \cdot y + f(u) \quad (4.28)$$

$$f(u) = b_0 \cdot u + b_1 \cdot u^2 + b_2 \cdot u^3 \quad (4.29)$$

Observera att det är endast  $b_1$  termen som ger högre förstärkning för positiva  $u$  och lägre för negativa  $u$  (4.29).

MATLAB gav för denna olinjära modell av första ordningen parametrarna:

$$a_1 = 0.104$$

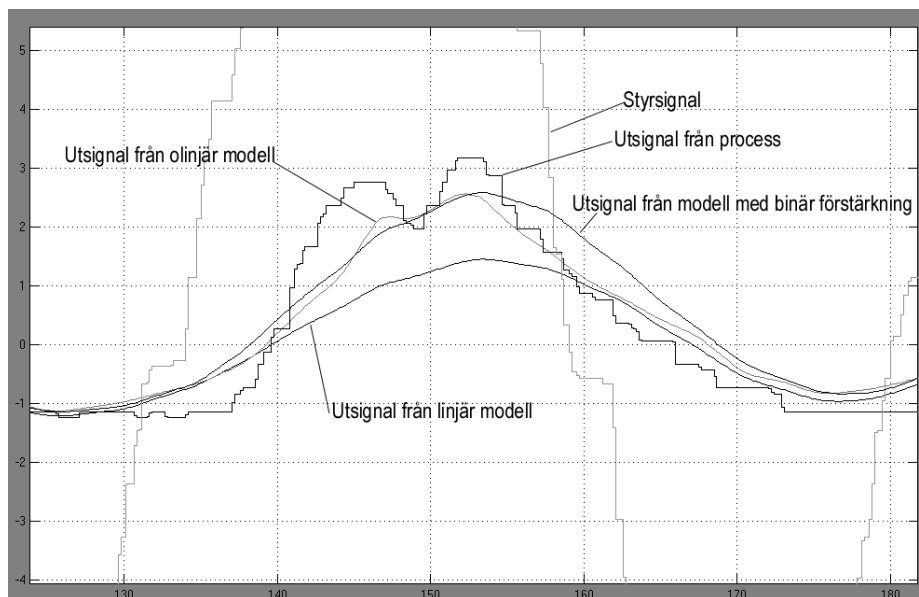
$$b_0 = 0.0014$$

$$b_1 = 2.47 \cdot 10^{-4}$$

$$b_2 = 2.07 \cdot 10^{-4}$$

## 4.9 Validering av Olinjära modeller

I figur 4.10 visas en modell med växlande förstärkning (förstärkningskvot: 2), en linjär modell och en modell skapad med polynomansättning validerade. Ingen av modellerna beskriver fullt ut processen. Man ser att den enkla binära linjära modellen är ganska bra trots sin enkelhet. Observera y-skalan är i meter och felet är därför ganska stort för alla modellerna.



Figur 4.10 Visar olinjär utsignal från process, olinjär utsignal från polynomansatt modell, binär modell, linjär modell, styrsignal. Tiden är angiven i sekunder.

## 5 Regulatorstrukturer

### 5.1 Tillståndsestimering

Då rodret kunde styras med en samplings-tid (genomsnitt) av 0.17 sekunder medan svar från givarna inte kom fortare än 1 sekund är det lämpligt att estimeras tillstånd med hjälp av ett Kalmanfilter. Tillstånden uppdateras varje gång mätsignaler kommer och estimeras annars vid varje sampel. Man kan i så fall styra farkosten utifrån dessa estimerade tillstånd. Kraven på denna metod är att modellen måste följa processen någorlunda. Detta gjordes dock inte vid framtagandet av regulatorn (eftersom linjär process saknades vid den tidpunkten). Men vid ytterligare förbättringar kan detta införas. I avsnittet förslag till fortsatt arbete kan man se det simulerade resultatet av PID reglering med observerare till den linjära processen.

### 5.2 PID-regulator

De krav som vid en första anblick kunde krävas av regulatorn var följande:

Om referensdjupet ändrades från ett visst värde skulle farkosten återgå till jämviktsdjupet så fort roderutslaget blivit noll (kabelns inverkan på farkosten) med enbart P-reglering, därför krävdes någon form av offset för rodervinkeln för olika referensdjup.

Om farkosten plötsligt skulle börja stiga av olika störningar som vågor eller strömmar, krävdes det att rodret snabbt skulle svara och styra farkosten till sitt referensdjup.

Om farkosten närmade sig referensdjupet med hög hastighet skulle ett roderutslag i motsatt riktning krävas (tillståndsåterkoppling av stighastigheten).

I princip skulle alla ovanstående krav vara uppfyllda med en PID-regulator:

Proportionaldelen skulle ge roderutslag beroende av hur stort felet i djup till referensdjupet. Integratorn skulle på rimlig tid kunna ge behövlig offset i roderutslaget. Deriveringsdelen skulle svara som en tillståndsåterkoppling av stighastigheten.

$$u = K \cdot \left( 1 + \frac{1}{T_I s} + T_D s \right) \cdot (u_c - y) \quad (5.1) \quad \text{ref: Glad \& Ljung: Reglerteknik}$$

### PI regulator med polplacering

Polplacera PI regulatorn enligt den karaktäristiska ekvationen:

$$s^2 + 2 \xi \omega_0 s + \omega_0^2, \text{ med } \xi = 0.8 \text{ för ett robust system.} \quad (5.2).$$

Det slutna systemet har överföringsfunktionen:

$$H(s) = \frac{G(s) * P(s)}{1 + G(s) * P(s)} \text{ med } G(s) = \frac{0.0013}{s + 0.067} = \frac{b_0}{s + a_1} \text{ och } P(s) = K \left( 1 + \frac{1}{T_I * s} \right) \quad (5.3).$$

Efter förenkling fås:

$$H(s) = \frac{Kb_0s + \frac{Kb_0}{Ti}}{s^2 + (a_1 + Kb_0)s + \frac{Kb_0}{Ti}} \quad (5.4).$$

Identifiera:

$$\frac{Kb_0}{Ti} = \omega_0^2, \quad a_1 + Kb_0 = 2\xi\omega_0 \text{ vilket ger:}$$

$$K = \frac{2\xi\omega_0 - a_1}{b_0} = \frac{1.6\omega_0 - 0.067}{0.0013} \text{ och } Ti = \frac{Kb_0}{\omega_0^2} = \frac{0.0013K}{\omega_0^2} \quad (5.5)$$

På så sätt fås ett robust system. Genom att ändra värde på  $\omega_0$  ställs in en önskad snabbhet i systemet. Observera att  $\omega_0$  måste vara större än 0.042 för ett positivt K.

### 5.3 MRAS baserad på Lyapunovfunktion.

Ett alternativ till PID-reglering är Model Referens Adaptiv System. Den har förmåga att anpassa regulatorn för olika processändringar beroende av kabellängd och farkostens hastighet genom vattnet.

Eftersom teorin är baserad på Lyapunovstabilitet för linjär process bör en linjäriserande kompensering införas då olinjäriteter uppträder. Den s k MRAS har två regleringsparametrar  $\theta_{1,2}$  som uppdateras adaptivt med hjälp av parametern  $\gamma$ . Modellen som man vill att processen skall följa får för första ordningens system formen enligt (5.6).

$$\dot{y}_m = -a_m y_m + b_m u_c \quad (5.6)$$

där  $u_c$  är referensvärdet och  $y_m$  det utsignalvärde som processen skall följa. Processen har motsvarande system enligt (5.7) och (5.8).

$$\dot{y} = -ay + bv \quad (5.7) \quad \text{jämför med (3.18)}$$

$$v = f(u) \quad (5.8)$$

Nu ska en Lyapunovfunktion som ger asymptotiskt stabilt system finnas. Ett vanligt sätt att finna funktionen är genom att utgå från ett systems totala upplagrade energi. Har denna energifunktion egenskapen att ständigt avta i riktning mot jämviktspunkten utan bli negativ är systemet asymptotiskt stabilt.

I Karl J. Åström och Björn Wittenmark: *Adaptive Control* ges Lyapunovfunktionen eller ”energifunktionen”  $V$  som (5.9):

$$V(\theta_1, \theta_2) = \frac{1}{2}(e^2 + \frac{1}{b\gamma}(b\theta_2 + a - a_m)^2 + \frac{1}{b\gamma}(b\theta_1 - b_m)^2) \quad (5.9)$$

$$\dot{V} < 0 \quad (5.10)$$

Lyapunovstabilitet innebär att ”energifunktionen” går mot noll, vilket svarar mot att systemets tillståndsvariabler går mot sina jämviktsvärden.

Där:

$$e = y - y_m \quad (5.11)$$

$$v = \theta_1 u_c - \theta_2 y \quad (5.12)$$

$$u = f^{-1}(v) \quad (5.13)$$

Lösningen av Lyapunovfunktionen ger uppdateringen av parametrarna enligt (5.14), (5.15). Lyapunovfunktionen garanteras då ständigt minska om  $a_m > 0$  enligt (5.16).

$$\dot{\theta}_1 = -\gamma_c e \quad (5.14)$$

$$\dot{\theta}_2 = \gamma e \quad (5.15)$$

$$\dot{V} = -a_m e^2 < 0 \quad (5.16)$$

## 5.4 MRAS baserad på MIT struktur

Teorin för denna struktur ges i Karl J. Åström och Björn Wittenmark: *Adaptive Control*.

MRAS baserad på MIT strukturen har endast en adaptiv parameter vars hastighet styrs av adaptionsförstärkningen  $\gamma$ . Processen styrs då endast genom framkoppling (5.17) av referensnivån förstärkt med  $\theta$ . Parametern  $\theta$  uppdateras enligt (5.20), förstärkt med parametern  $\gamma$ . Utsignalen som processen skall följa ges av (5.19). Felet i verklig utsignal ges av (5.21). Den enkla modellen är alltid stabil eftersom fasen inte underskrider  $-90$  grader.

$$u = \theta u_c \quad (5.17)$$

$$y = G(s)u \quad (5.18)$$

$$y_m = G_m u_c \quad (5.19)$$

$$\dot{\theta} = -\gamma_m e \quad (5.20)$$

$$e = y - y_m \quad (5.21)$$

## 6 Simulering i Simulink

Vid simuleringen som utfördes i Simulink användes de olinjära processmodellerna som identifierades från första identifieringstillfället. Man kan då testa de styrlagar som beräknats och hur de fungerar i praktiken.

### 6.1 Maximal roderhastighet

Då rodret steg 15 grader på 30 sekunder var detta en begränsning av hur snabbt farkosten kunde reagera. Denna begränsning infördes i simulering med första blocket i figur 6.1.

### 6.2 Maximalt Roderutslag

För att inte hamna helt utanför linjäriserbart område valdes i simuleringen max roderutslag till 6 grader. Denna begränsning bestämde hur snabbt systemet kunde svara på referenssignalen.

För att införa ovanstående begränsning vid simulering användes andra blocket i figur 6.1.



Figur 6.1

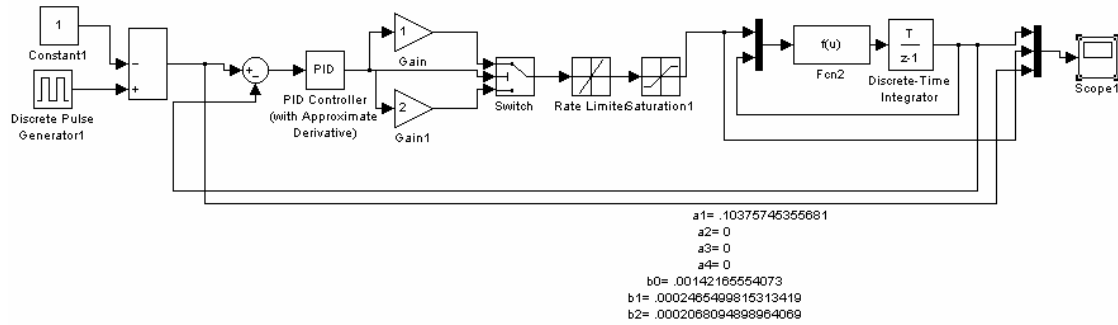
### 6.3 Maximalt avstånd från jämviktpunkt

I identifieringsförsöken visades att farkosten inte kunde stiga mot ytan mer än 1 meter från jämviktdjupet medan den kunde sjunka lägre. I simuleringen valdes max 1 meters ändring från jämviktsläget upp som ned.

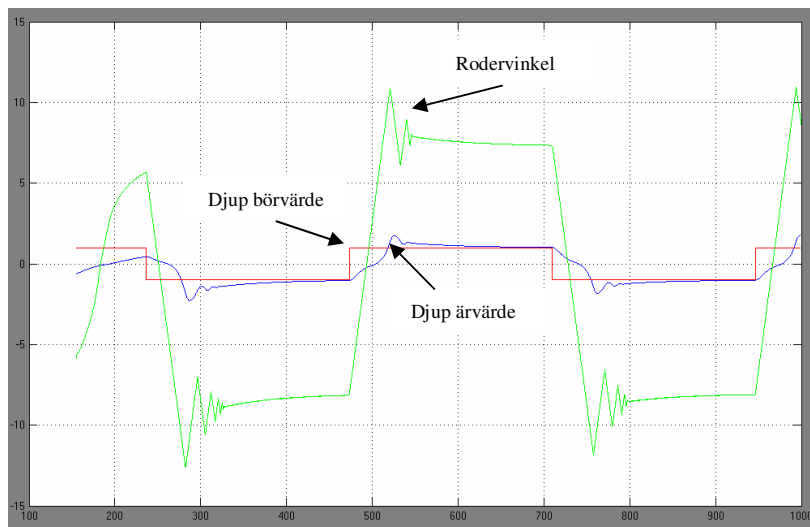
### 6.4 Simulering med PID regulator och binär förstärkning

I figur 6.3 visas simulering med binär förstärkning simulerat på skattad olinjär process. PID regulatorns uppkoppling visas i figur 6.2. Simulering visar att referenshöjning utan överslängar begränsas till 0.2 meter, vilket är alldeles för lite. Det räcker bara till att reglera farkosten om den skall följa en långsamt föränderlig botten.





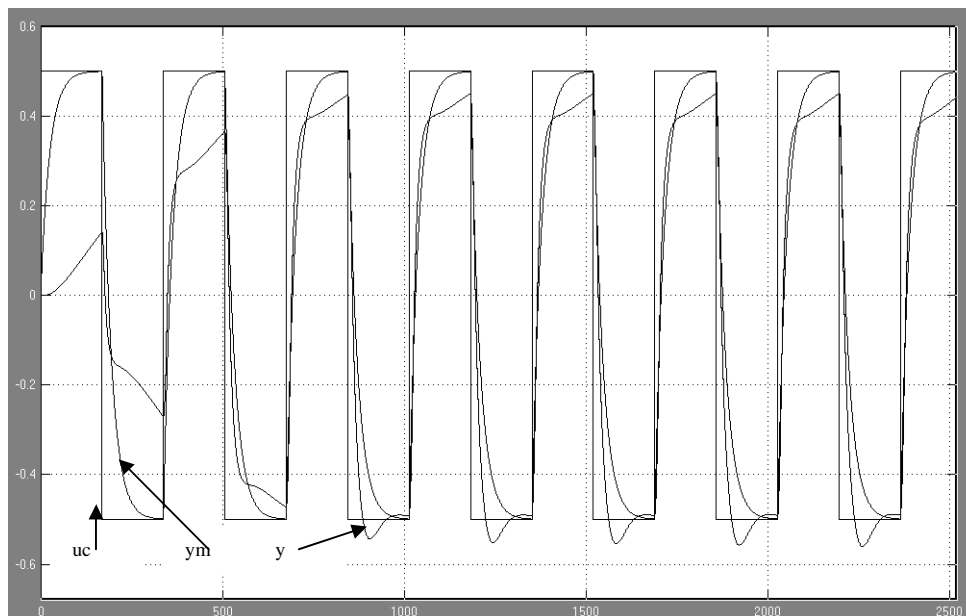
Figur 6.2 Visar simuleringsuppsättning med PID regulator och binär kompensering för olinjära effekter.



Figur 6.3 Visar insignal  $-10$  till  $10$  grader, utsignal, referensnivå  $1$  till  $-1$  meter. vid PID-reglering.

## 6.5 Simulering med adaptivt Liapunobaserad MRAS-regulator

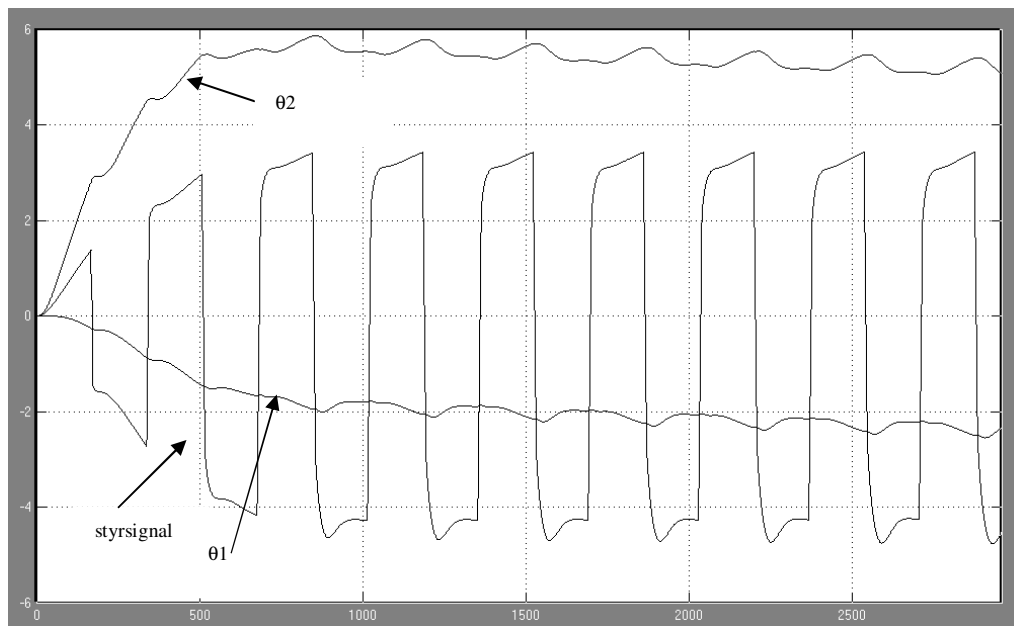
Simulink struktur av Liapunobaserad MRAS-regulator hittas i bilagan. I figur 6.4 visas simuleringsresultat. Trots att processen är olinjär svarar systemet bra.



**Figur 6.4** Visar simulering med Liaponov MRAS. Kurvorna är: referenssignal  $u_c$ , utsignal  $y$ , önskad utsignal  $y_m$ .  $y_m$  kommer från första ordningens modell, utsignalen  $y$  från olinjär process.

I figur 6.5 visas hur adaptationsparametrarna ändras med tiden. Parametrarna konvergerar ganska snabbt medan roderetshastighet begränsas till 0.2 rad/s. Den önskade processen  $G_m$  är vald så att roderutslaget begränsas av max- och min- utslag.

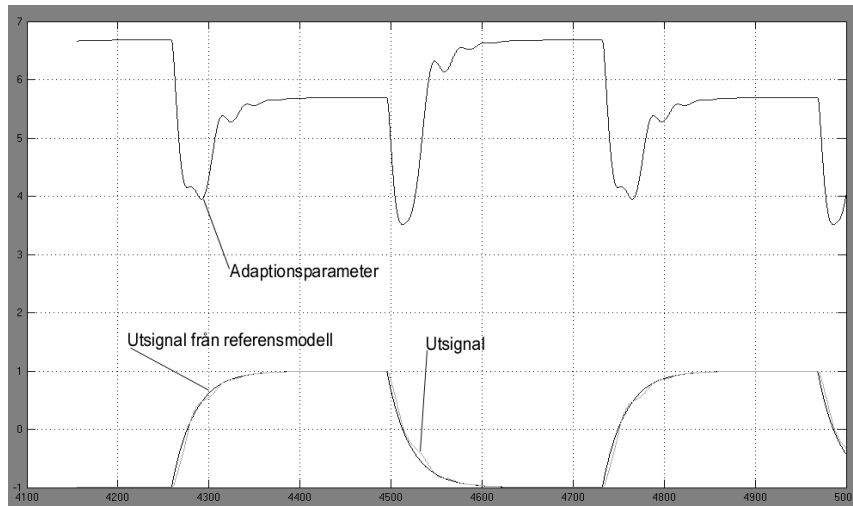
Helt perfekt följning av  $y_m$  kunde dock inte uppnås då den linjäriserande kompensationen endast var inversen till ett andragradspolynom. Processen blir då fortfarande olinjär men inte lika mycket.



**Figur 6.5** Visar styrsignal och adaptationsparametrar. Tiden är angiven i sekunder.

## 6.6 Simulering med MRAS enligt MIT-rule

Simuleringsuppställning av MIT-regulator hittas i bilaga. I figur 6.6 visas simuleringsresultat: utsignal från referensmodell, utsignal och adaptionsparameter. Man ser att utsignalen följer referensutsignalen så bra att de ligger nästan intill varandra.



Figur 6.6 Visar adaptionsparameter,  $y_m$  och  $y$ . Tiden är angiven i sekunder.

I Liapunovregulatorn finns två parametrar som garanteras konvergens för linjärt system. I det olinjära fallet görs inte detta och regulatorn är instabil och parametrarna kan i värsta fall växa mot oändligheten. I regulatorn enligt MIT-rule finns endast en parameter som adapterar. Denna parameter växer/minskar då alltid i rätt riktning. Nackdelen skulle vara att systemet endast regleras genom framkopplingen, detta gör dock inte så mycket i vårt fall eftersom ett stabilt olinjärt system av första ordningen alltid är stabilt. Parametern adapterar då endast för att ge en korrektare styrsignal till systemet.

## 7 Regulatortest i Öresund

Under arbetets gång har olika version av regulatorprogram skapats i LabView. Innan farkostens processmodell var framtagen designades en PD regulator som första försök. Anledningen till att den valdes framför andra metoder är dess förmåga att estimerar systemuppförandet utan att ha stark kännedom om processen, därmed med lämpliga parametrar begränsar eventuella överslängar. Eftersom havets botten är ganska ojämn och kraven om höjd inte är så hårda ( $\pm 20$  cm fel är godtagbart), ansågs en PI regulator var onödig. Och en PID regulator har för många parametrar att hålla reda på.

Tyvärr testades PD regulator inte vid första identifieringstillfället på grund av tidsbrist. När processen identifierades senare kunde en adaptiv regulator, baserad på MIT-rules skapas. Algoritmen och simuleringen kan läsas i kapitel 5 och 6. Under tiden då processen identifierades, skapades, enligt handledaren Tore Hägglunds rekommendation, även en PI regulator baserad på polplaceringsmetoden.

Den dag när regulatorerna skulle testas skede en konflikt mellan tidssynkronisering och inläsning av COM portarna. När tidssynkroniseringen togs bort fungerade (förstås) inte MIT regulatorn. Anledningen till att MIT regulatorn inte fortsatte förbättras vid senare tillfälle är huvudsakligen att det inte finns några mätdata om hur mycket farkosten har rört sig kring sin "jämviktspunkt". En "jämviktspunkt" som skulle vara den höjden då rodervinkeln står på noll. Men på grund av starka och relativt snabba vattenvågor skiljer det 60 cm (vid 20 meters kabel), vilket är nästan 1/3 av styrbarhetsområdet. Vid första versionen av MIT regulatorn skattades "jämviktspunkten" genom att ändra 10 % av värdet med nya höjdvärde varje gång rodervinkel passerar noll grader. Metoden förkastades dock senare då uppdateringen ansågs vara opålitlig.

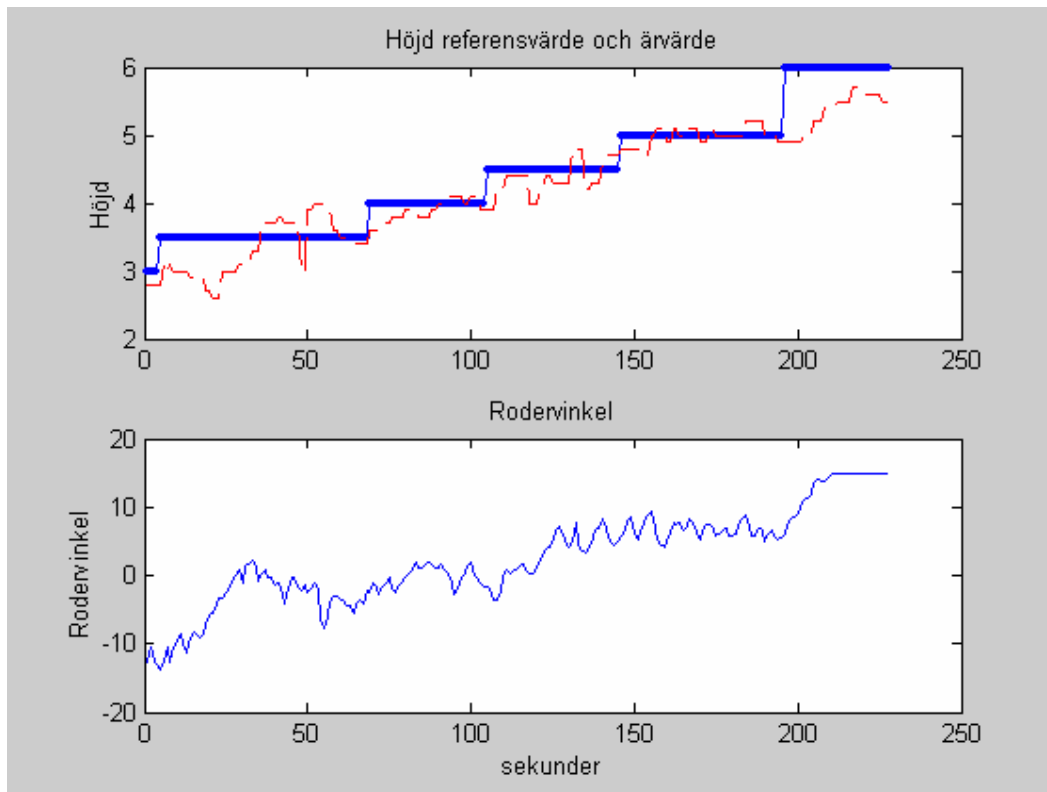
Vid testet mättes stegsvar där varje steg motsvarade en referenshöjning med 0.5 m. Stegändringen bedömdes lämplig i det begränsade arbetsområdet.

### Resultat

Testerna visade att både PD regulatorn och PI regulatorn fungerar. Utsignalen från farkosten svarade då med viss översläng men efter en insvängning kunde referensnivån följas. Dock är PI regulatorn mycket stabilare än PD regulatorn. När PI regulatorns parametrar:  $\omega_0 = 0.071$  (och  $\xi=0.8$ ) visade processen nästan ingen översläng alls. Resultaten var bra med tanke på alla begränsningar vår utrustning hade. Den kunde inte motverka "snabba" störningar från vattenvågor men regleringen är mycket robust. Det tar tiotals sekunder tills farkosten ställer in sig till det nya höjdbörvärdet och det maximala stationära felet är en halv meter.

Hela styrbarhetsområdet har utökad från 2 meter till nästan 3.5 meter efter trimningen av farkosten. Det kanske kan ökas ytterligare något om man flyttar kabelns infästning något bakåt för balansering.

Del av körningsresultatet presenteras i figur 7.1:



**Figur 7.1** Resultat av PI reglering, då kabellängd är 20 m,  $\omega$  är 0.071 och vattnetsrelativhastighet är 1 m/s. Kurvan visar Höjdreferens, Höjdärvärde och roderärvärde

Man kan se att rodervinkeln svänger en hel del kring börvärdet. Detta beror på det låga toleransvärdet på felet mellan roder referensvinkel och roder ärvinkel. När toleransvärdet i programmet ökas från 0.5 grader till 2 grader betar rodervinkeln sig mycket lugnare.

## 8 Hårdvara

### 8.1 Datorkraft

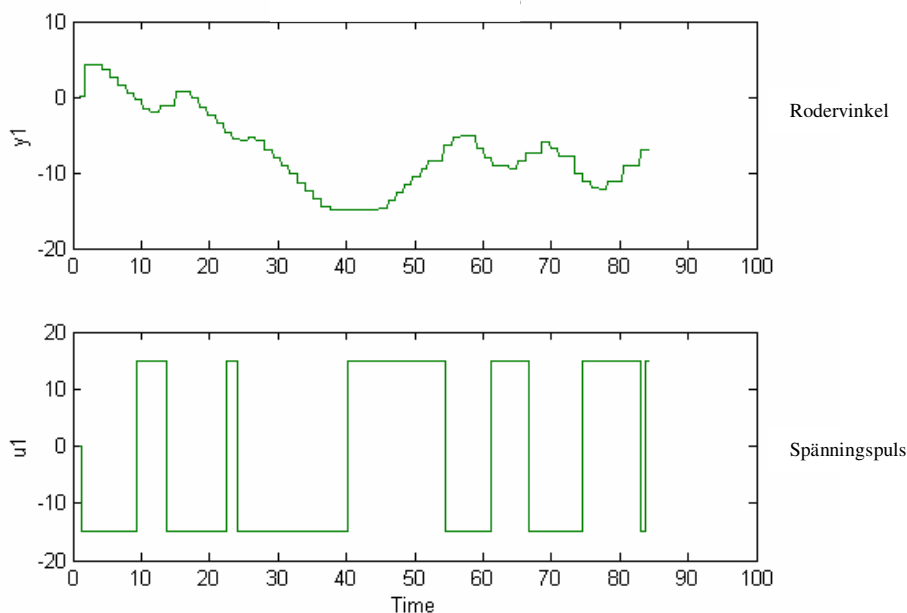
Datorn som används är en bärbar pc Toshiba 350 Mhz med 64 MB RAM. Som operativsystem används Windows 98. Datorn har tre COM-portar, varav 2 är externt anslutna.

### 8.2 Motorservo

Att styra farkosten från vinkelreferens var en rimlig strategi då motorspänningen till rodret inte var steglös och dessutom inte synkron i tiden. Rodervinkeln kunde mätas och därmed också en bra mätsignal vid identifieringen.

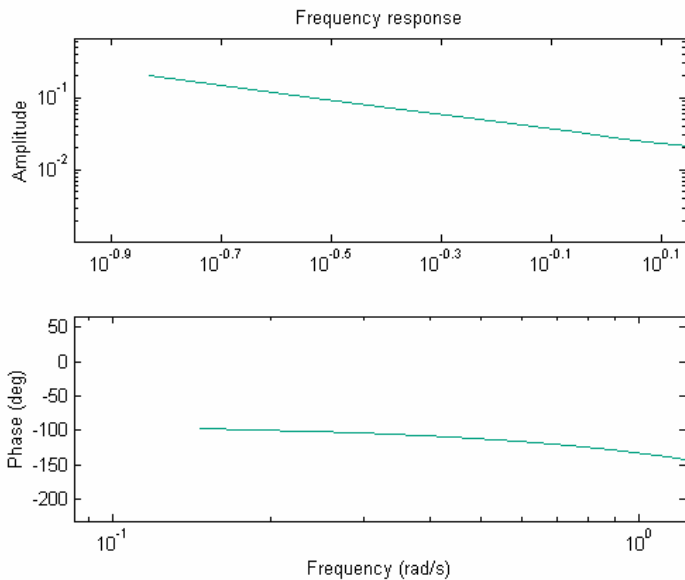
Rodret kaskadregleras för sig. Alternativt skulle pulsbreddsmodulerad styrning (s.k. PWM-styrning) vara möjligt, men skulle kräva annan utrustning än den som var given.

I figur 8.1 visas beroendet mellan rodervinkel och pulsbredden. Rodret svarar lika snabbt nedåt som uppåt med hastigheten 0.5 grader/s (dvs cirka 0.2 rad/s).



**Figur 8.1** Visar pulsbredd och resulterande rodervinkel. Lutningen på rodervinkeln är lika stor upp som ned. Tiden angiven i sekunder.

Man kan se botten rodret i  $-15$  grader vid 40 sampel och identifieringen gav därmed ett något felaktigt resultat. Identifieringen gjordes med hjälp av PRBS-signaler och gav bodediagrammet i figur 8.2. Man kunde anta att processen var en integrator med viss fördröjning av utsignalen. En aktiveringstid kunde i så fall beräknas för att ge önskat roderutslag. Det visade sig i praktiken att denna aktiveringstiden var svår att använda då överföringen av styrdata skedde asynkront. Signalen stannade efter 0.5 s trots aktiveringstiden satts till 2s.



Figur 8.2 Visar bodediagram för roderservot och uppträder som integrator.

### 8.3 Signaler från GPS och styrboxen

GPS skickar signaler till datorn via seriella COM portar. Detta sker asynkront. Det finns olika slags signaltyper som innehåller information om t ex farkostens djup och båtens relativhastighet genom vattnet. Enligt GPS handbok skickas varje typ av signal i snitt en i sekunden, men tyvärr i praktiken kunde signalerna tas emot mer sällan.

Signalen till datorns COM port 1 via styrboxen innehåller data om farkostens djup, rodervinkel och farkosten höjd över botten. Signalen till datorns COM port 2 innehåller data om båtens relativa hastighet genom vattnet och position. Karaktären för dessa signaler beskrivs i bilagan: "Marina Signaler".

### 8.3 Uppkoppling

Starta datorn som vanligt. Kabeln med vit märkning (innehåller data om farkostens djup, farkostens höjd över botten) från GPS ska in till datorns standard COM port som finns på datorns baksida. Kabel med blå märkning (innehåller data om relativ vattenhastighet) från GPS ska in till en av datorns extern COM portar: CH1. som finns på vänstersidan av datorn. Kabel med röd märkning används inte.

## 9 Identifieringsprogram

När arbetet började fanns det flera tänkbara program att arbeta med. Alternativ var bland annat: MATLAB i Unix/Pc miljö, realtidsprogrammering med Java, respektive programmeringsspråket LabView. Efter en del undersökning och övervägning bestämdes slutligen att använda LabView. Huvudanledningen till att det valdes var att det var enkelt att identifiera dataportarna.

### 9.1 Identifieringsprogram: "ident(selffrekvt).vi"

Programmet "ident(selffrekvt).vi" ger styrsignaler för framtagning av modeller av processen med hjälp av spektralanalys. Programmet samlar och sparar mätdata till en txt-fil då den sinusformade insignalen har en exponentiellt minskande periodtid sk chirpsignal. Periodtiden valdes att variera mellan 148.4 s (0.0067Hz) och 30 s (0.0286Hz). Periodtider som är längre än 148.4 s ansågs skulle kunna orsaka olinjäriteter på utsignal. Periodtider som är kortare än 30 s anses vara ointressanta vid identifieringen eftersom rodervinkeln inte kan röra sig med högre hastighet än 0.5 grader/s.

#### Att användning programmet

I programfönstret (se bilaga: Ident(selffrekvt) huvudfönster) ser man några sensorer och styrknappar. Bland sensorer finns det:

- GPS signaler (från COM portarna)
- ReferensSignal: rodrets referensvinkel, kommer från en sinus generator.
- Sinus Looptid & program Looptid: detektera tidsdelay (i ms) i programmet.
- Roder vinkel, Djup (farkost), Fart genom vattnet
- T: den längsta periodtiden för referenssignalen. är inställt vid 148.41s.
- Köretid: hur länge program har kört i sekunder.
- Styrsignal: exakta kommando som skickas till styrboxen.
- Grafen: ritar Bör/är värden och djupet

Bland styrknappar finns:

- MaxVinkel: maximal rodervinkel. Är inställd i  $\pm 10$  grader
- Stop(styrSignal): skickar stopp kommando till styrboxen.
- Manuell/Auto och Up/ner: används vid manuell styrning. Manuell/Auto av =auto modell
- Vinkel offset: används inte i vår fall.
- Start: starta sinus-generator och samla data
- Stop: stoppa sinus-generator, därmed programmet

Egentligen är knapparna Start och Stop de enda som behövs vid datainsamlingen. Andra styrsignalen fungerar vid sina defaulta värden. För säkerhet kontrolleras: alla styrknappar (Start, Stop(StyrSignal), Manuell/auto) är avslagna.

När man klickar på kör-pilen kommer ett fönster upp och frågar vilket filnamn som ska sparas. Här kan man till exempel skriva in ett filnamn som anger vilken kabellängd man har



använt. Om filen redan existerar öppnas ett nytt fönster, då klickar man på "replace". När det är klart klicka man på knappen "start" för att börja datainsamlingen.

Som sagt tidigare börjar periodtiden för chirpsignalen på  $T = 148.41$  s och minskar exponentiellt. Då  $T$  minskat till omkring 30s kan man trycka "STOP" för att avsluta mätningen. För att man skall få en bra start till nästa omgång är det klokt att klicka "STOP" då rodervinkelns ärvärde har ett värde nära noll. Sen är det bara att börja nästa mätning med en ny kabellängd. Glöm inte avmarkera styrknappen "start", annars blir det fel.

## Programmets uppbyggnad

Programmet byggdes upp av huvudsakligen två parallella processer. Den ena kallades för "generatorprocess" och har funktioner att generera en sinus referenssignal och bestämma maximal vinkel och vinkeloffset. Den maximala vinkeln begränsade rodrets vinkelutslag för att undvika att utsignalen når det mättade området och blir olinjärt. Vinkeloffset är till för att justera eventuellt asymmetrisk utsignal genom att flytta nollnivå av insignalen. Den andra processen, s k "huvudprocess", har funktioner bland annat att läsa och avkoda signaler från GPS, kaskadreglera rodervinkeln via styrboxen, spara mätdata till en txt-fil och ta hand om tidssynkronisering. När "Stop" knappen trycks ner stannas programmet och processerna termineras.

De två processerna kommunicerar med varandra med hjälp av LabViews standard sub-vi: create-, delete- och sent notification. Tyvärr då LabView inte är uppbyggt för att köra parallella processer, lyckades det inte göra processerna helt oberoende. T ex "huvudprocess" blockeras så länge den inte har fått en ny vinkelreferens från "generatorprocess". Av denna anledning kunde processerna inte köras som "realtidprocesser" som man hoppades på, utan mer sekventiellt. Med tanke på att farkosten är en långsam process kan man kaskadreglera den utan större problem.

### Generatorprocess

Jämför "ident(selffrekvt).vi programstruktur" i bilaga med beskrivning nedan.

Processen körs inom en While-loop, den körs så länge knappen "stop" inte blir nertryckt. En sinus värde skapades på nedanstående sätt:

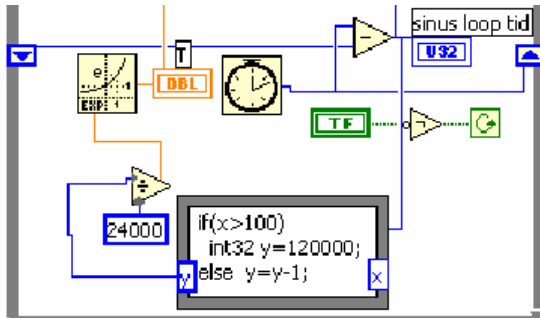
$$\text{sinus värde} = \sin \frac{2\pi * i * \text{tidsdelay}}{1000 * T} * (\text{maxVinkel} - \text{VinkelOffset}) \quad (9.1)$$

här är  $i$  = antal loopar som While-satsen har kört, tidsdelay = 10 ms och  $T$  = periodtid för sinussignalen och fås ur sambandet:

$$T = \exp\left(\frac{y}{24000}\right) \quad (9.2)$$

här  $y = 120\,000$  i begynnelsevärdet och minskas med 1 vid varje loop efter knappen *start* blir true. Räkningen kan ses i undre delen i figur 9.1. Värdena 24 000 och 120 000 är inte några bestämda värden. De var bara en kombination som testades fram för att anpassa periodtidsintervall och hur fort det skulle minskas, samtidigt hölls alla värde som heltal. Observera att i

och med "generatorprocess" kör snabbare än "huvudprocess" är tidsdelay avgörande på hur fort periodtiden för roderreferensvinkeln minskas.



Figur 9.1

Inom while-loopen finns några enkla komponenter för att räkna ut looptiden i ms, detta fås genom att ta tidsskillnad mellan två loopar (ser överst delen i figur 9.1). Resultaten används både för att ge begynnelse värde till y och körtidsberäkning. Det är nämligen så att looptiden blir ca 1000 ms om *Start* är false, men omkring 10 ms då *Start* är true.

Sensorn *körtid* är inte så nödvändig vid spektralanalys, men för frekvensanalys. Beräkningen gjordes i ett funktionsblock på ett liknande sätt som T: Då "Start" är true ökas körtiden med looptiden. Resultaten divideras med 1000 för att få tiden i sekunder.

If-satsen har funktionen att skicka sinus-värden vidare då *Start* är true, annars fördröjer det hela processen 1000 ms och skickar värdet 0 vidare. Det hela är för att minimera fel vid start då man håller på att skriva filnamn osv. Då hålls processen någorlunda i standby läge tills *Start* blir nedtryckt, då börjar programmet sina datainsamlingar.

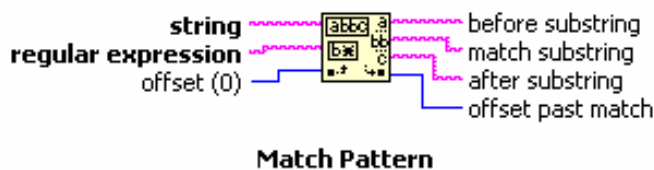
### Huvudprocess

"Huvudprocess" är en While-sats och den består av två sidor. Den första sidan används enbart för inläsning och avkodning av signaler från GPS. Den andra sidan svarar huvudsakligen för Kaskadregleringen av rodevinkeln.

### Sidan 0

I denna del läser processen först in COM porten 1 och 3. GPS signaler har karaktär enligt tidigare beskrivning "Signaler från GPS". För att hämta nyttiga värden från dem har även två sub-vi: SträngHandler.vi och RMBspeed.vi skapats för att maskar ut informationen.

Grafen för de två sub-vi:n finns i bilagan. Principen för dem är enkel: Eftersom GPS signalerna har ett standardutseende, används "Match Pattern" (figur 9.2) för att "klippa" dem



Figur 9.2

så att önskad sträng erhålls. Vidare omvandlas strängen till reella tal. Observera problemet här är att GPS inte skickar signaler kontinuerligt, vilket leder till att man får värdet 0.00 då den avkodade strängen inte finns. För att undvika detta skickas även det gamla mätvärdet, från skiftregister, in till sub-vi och ersätter det felaktiga värdet med det gamla.

Det finns även här en tidsräknare av samma utseende som den i "generatorprocess" för att användaren skall kunna hålla reda på processens periodtid.

### *Sidan 1*

I sidan, utanför while-satsen, 1 skapas först, med hjälp av "Open/Create/Replace file", en txt-fil där alla mättdata ska ligga. Mättdata som ska sparas till filen är: rodervinkelns är- resp. börvärde, båtens relativa fart genom vattnet, farkostens djup och tiden.

I while-satsen hämtar processen rodrets vinkelreferens som "generatorprocess" har skickat. För att synkronisera looptiden, dvs. att få en konstant samlingstid ur regler perspektiv, sattes det in en klocka (wait until) som används för att blockera processen tills 160 ms har gått.

Vid Kaskadreglering jämförs rodrets är- resp. bör- värde. Toleransen är 1°. Om vinkel-skillnaden är större än toleransvärdet skickas ett styrkommando till "farkostens" styrbox. Styrkommandot har nedanstående utseende:

Upp x-sekunder:	>UP x
Ner x-sekunder:	>DOWN x
Stanna:	>STOP

Här x är aktiveringstid som har satts till 0.3 sekunder.

Det finns också några andra if-satser i denna sida. Dem är till för att man ska kunna köra rodret manuellt om man så vill.

Anledningen till "huvudprocess" måste köra långsammare än "generatorprocess" är inte bara att den är blockerad då den inte kan hämta ett värde som "generatorprocess" har skickat. Den största orsaken är överföringshastighetsbegränsning hos COM portarna: GPS signaler skickas ofta ett par gånger per sekund och det tar också en viss tid för COM porten att hämta data, som består av 20-tal tecken. Detta orsakar en enorm tidsförlust. Efter många tester valdes att portarna öppnas högst 150 ms vid signalinläsning, det är en tid som är acceptabel för Kaskadregleringen i systemet.

## **9.2 Identifieringsprogram: "ident.vi"**

"ident.vi" används vid frekvensanalys. Den är nästan identisk med "ident(selffrekvt)" som har beskrivits tidigare. Här ska bara skillnaden beskrivas. Generellt kan man säga att det "ident.vi" skiljer sig från "ident(selffrekvt).vi" är att sinusbörvärdet här har en konstant periodtid.

### **Att använda programmet**

Användargränssnittet i "ident.vi" ser likadant ut som "ident(selfrekvt).vi". Skillnaden här är att man ska titta på *Körtid* för att veta hur många perioder den sinusformade referenssignalen har kört. För processidentifiering måste data samlas från minst 6 perioder för att få ett väl resultat.

En annan skillnad som medförs från analysmetoden är att  $T$  här inte längre är en sensor, utan en styrsignal, som berättar för programmet vilken periodtid referenssignalen ska ha.

### **Programmets uppbyggnad**

I "ident.vi" saknas den exponentiella periodtidsminskningen hos referenssignalen. I "generatorprocess" matas signalen  $T$  direkt in från användaren och är konstant under hela körningen.

## 10 Regulatorprogrammet

LabView programmet ”reglerPI.vi” används för att styra farkosten till önskad höjd över havets botten. Det ursprungliga styrbarhetsområdet var cirka 2 meter vid 20 meters kabel, men utökades till nästan 3.5 m efter 4 stycken lyftblock akterut lades dit för att balansera vikten.

Man kan tänka sig här att lägga till två wave filer för att varna om kabeln är för lång eller för kort, på så vis behöver användaren inte hela tiden sitta framför datorn vid körningen. Men försöket misslyckade eftersom LabView inte klarade att spela upp ljudfiler när programmet körs cirka 170 ms vid varje loop.

### 10.1 Att använda programmet

Ungefär som ident-programmen finns det några sensorer och styrknappar i huvudfönster. Se bilaga: ”reglerPI.vi- huvudfönster”.

Bland sensorer finns

- Roderreferens: rodrets referensvinkel, kommer från PI regulator
- Rodervinkel är/bör värde, Djup (farkost), Höjd (över botten) ärvärde
- Looptid: hur lång tid program kör en loop, i ms
- Styrsignal: exakt kommando som skickas till styrboxen.
- Grafen: ritar Höjdens bör/är värden och rodets ärvärde
- k, i: värdena på K och Ti med valde  $\omega$
- kortare/längre kabel: indikerar om kortare/längre kabel behövs

Bland styrknappar finns:

- Stop(styrSignal): skickar stopp kommando till styrboxen.
- Manuell/Auto och Upp/Ner: vid manuell styrning kan man välja Upp/Ner för att styra rodret
- Höjd börvärde
- $\omega$ : väljer lämpliga  $\omega$ , som bestämmer regulatorns snabbhet
- Avsluta: Avsluta hela körningen

Programmet är designat huvudsakligen för att köras i automatiskt läge, därför lyser styrknapparna: Stop; Manuell/Auto; Upp/Ner; Avsluta, grönt då läget valts.

Vid normal körning visar även sensorerna: kortare- resp. längre kabel grönt för att indikera allting fungerar rätt. När kortare eller längre kabel behövs, ändras motsvarande sensor till röd färg. Då bör man släppa ut eller dra in kabel långsamt.

Via sensorerna: Roderreferens; Styrsignal; Djup; Höjd; har man koll på om regulatorn arbetar som den ska, och hur fort GPS signalerna uppdaterar sig. Dock har sensorerna: Looptid; k; i; inte särskild betydelse för den vanliga användaren.

Styrknappen  $\omega$  är viktig vid reglering. Värdet minskas eller ökas med ca 0.001, vilket insetts vara ett lämpligt värde. Som sagts bestäms systemets snabbhet av  $\omega$  och man bör minska  $\omega$  vid höga överslängar (rodervinkel) och öka  $\omega$  vid långt stegsvar (höjdärvärde). Både överslängar och stegsvar kan synas i grafen. Generellt kan man säga att  $\omega$  ska ökas vid längre kabel (eller mindre fart) då styrbarhetsområdet minskar. Det gäller förstås omvänt vid kortare kabel och högre fart.

## 10.2 Programmets uppbyggnad

Jämför med ”reglerPI.vi programstruktur” i bilaga med beskrivningen nedan.

Till skillnad från ident programmet har PI-regulatorn endast en process. Den körs sekventiellt. Efter att tids synkronisering togs bort, körs det allt mellan 80 ms till 150 ms beror på när GPS signaler kommer fram. Programmets uppbyggnad påminner om ”huvudprocess” hos ident.vi.

Programmet består av en While-loop med två sidor: sidan 0 och sidan 1. Loopen körs så länge *Avsluta* är false.

I sidan 0 läses och dekodas GPS signaler och Looptiden beräknas. Som det nämndes tidigare kommer GPS signalerna asynkront från COM port 1, tidsbegränsning av att läsa porten är satts till 150 ms. Eftersom det inte längre behöver den relativa vattenhastigheten tas signaler från COM 3 inte längre in till datorn som ident programmet gör.

I sidan 1 räknas ut roderreferensvinkel, kaskadreglerar rodervinkeln, skicka styrsignal till styrboxen, rita grafen och kontrollera om kortare/längre kabel behövs.

I funktionsblocket beräknas de nya PI parametrarna enligt PI regulatoralgoritmen (polplaceringsmetod) som beskrevs i kapitel 5.2. Insignal till blocken är  $\omega$  och utsignaler är P och I. Parametrarna P och I skickas vidare till en PID regulatorn (subvi: simpelregler.vi) och roderreferensvinkeln räknas ut. Referensvinkeln används vidare vid kaskadregleringen, och är identisk med identifieringsprogrammets. För att kontrollera om längre/kortare kabellängd behövs kan man kolla om roderärvärdet har uppnått sitt ändläge.

Eftersom programmets uppbyggnad är så lik *ident*-filerna beskrivs den inte närmare här.

# 11 Förslag till fortsatt arbete

## 11.1 Hårdvaror förslag

Ett av de största problemen med regleringen är att mätsignaler från sensorerna kommer för sällan. Mätdata från rodervinkel kommer cirka en gång per sekund, medan data om höjden kommer ännu mer sällan. Det andra problemet är att rodret rör sig för långsamt. De två begränsningarna omöjliggör att kompensera vågstörningar oavsett hur bra regulatören är.

Styrboxen med COM port kommunikation begränsar mätsignal uppdateringshastigheten. Om man vill byta ut det med en parallell port kan hastigheten ökas. Samtidigt ökas även precisionen för styrsignalerna. Då måste man bygga om hårdvarorna.

För att öka roderhastigheten måste servot modifieras. Periodtiden för rodret bör åtminstone vara hälften så lång som periodtiden hos vågorna. En högre nivåns regulator är rekommenderad eftersom rodervinkel måste ökas mycket fort utan att orsaka överslängar. Vid regulatordesign måste man även ta hänsyn till: nollnivån som inte går att mäta (även svår att skatta) och processmodellen måste identifieras mycket väl.

Genom att balansera lasten ligger farkosten mer horisontellt i vattnet, vilket utökar styrbarhetsområdet. Man kan sätta in fler lyftblock eller flytta kabels infästning bakåt för att åstadkomma detta. Var försiktigt med balanseringen så att farkosten inte blir framtung istället. Ett annat förslag för att utöka styrbarhetsområde är att ge farkosten större roder.

## 11.2 Mjukvaror förslag

När en linjär process identifierades efter det andra identifieringstillfället var det en bra idé att skapa en observerare (t ex Kalmanfilter enl. Kapitel 5.1). I simulink har en sådan modell skapats.

### **PID simulering med observerare och asynkron tillståndsinläsning**

Med en linjär process kan tillstånden estimeras tillförlitligt. Om de skattade tillstånden överstämmer med de verkliga kan reglering förbättras avsevärt. Tillstånden uppdateras estimerande om ingen ny signal läses in från porten. Men så snart givarna skickar värden ersätts det estimerade tillståndet. Regulatorn kan i så fall beräkna styrsignalen i samplingshastighet baserad på tillstånden.

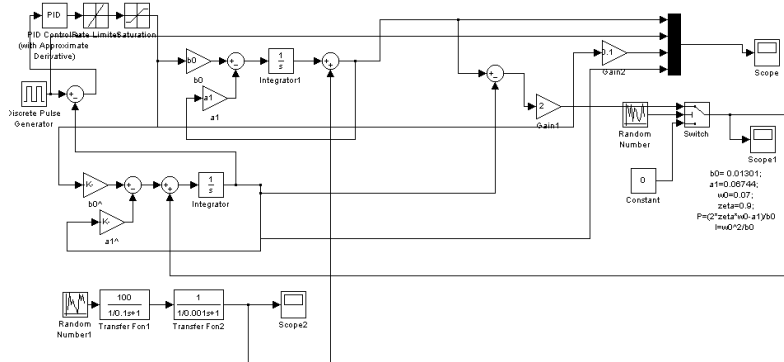
Systemet beskrivs av ekvation (11.1), där  $a_1$  och  $b_0$  är identifierade process parametrar.

Observeraren beskrivs av ekvationen (11.2) där  $\hat{a}_1$  och  $\hat{b}_0$  är skattade processparametrar och  $k$  är konstant.

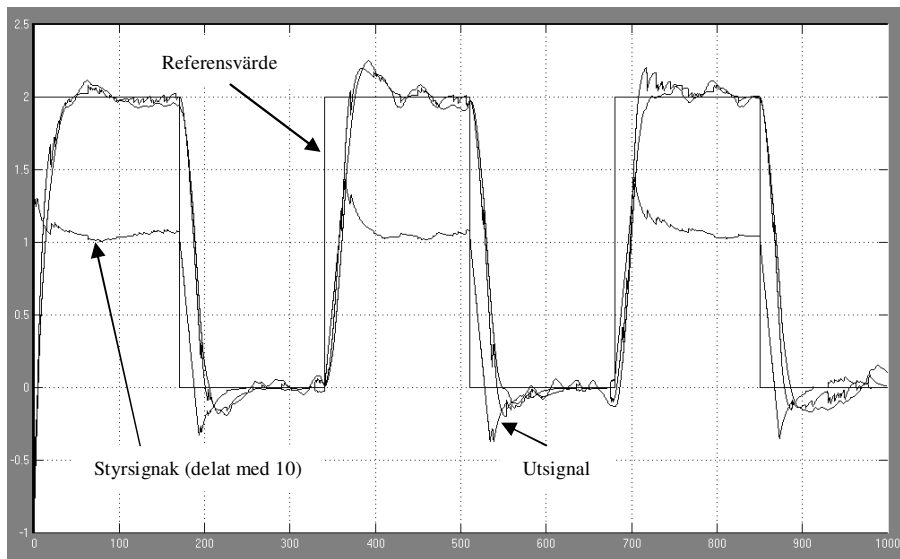
$$\dot{y} = -a_1 y + b_0 u \quad (11.1)$$

$$\dot{\hat{y}} = -\hat{a}_1 \hat{y} + \hat{b}_0 u + k(y - \hat{y}) \quad (11.2)$$

Figur 11.1 visar simuleringsuppställning av PID reglering med observerare. Vid simuleringen var även vågorstörningar och asynkront datainläsning införd. Figur 11.2 visar simulerings resultat.



Figur 11.1 Visar PID simuleringsuppställning med observerare.

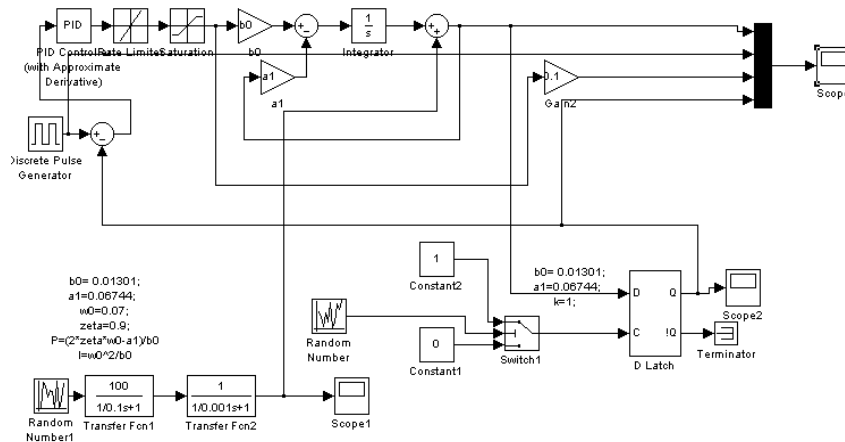


Figur 11.2 Simulerat resultat av PID regulator med observerare. Man ser referensvärde, utsignal, styrsignal (delat med 10) samt skattad utsignal. Observera att styrsignalen minskas vid överslängarna även då aktuell utsignal inte mottagits, jämför med simulering utan observerare.

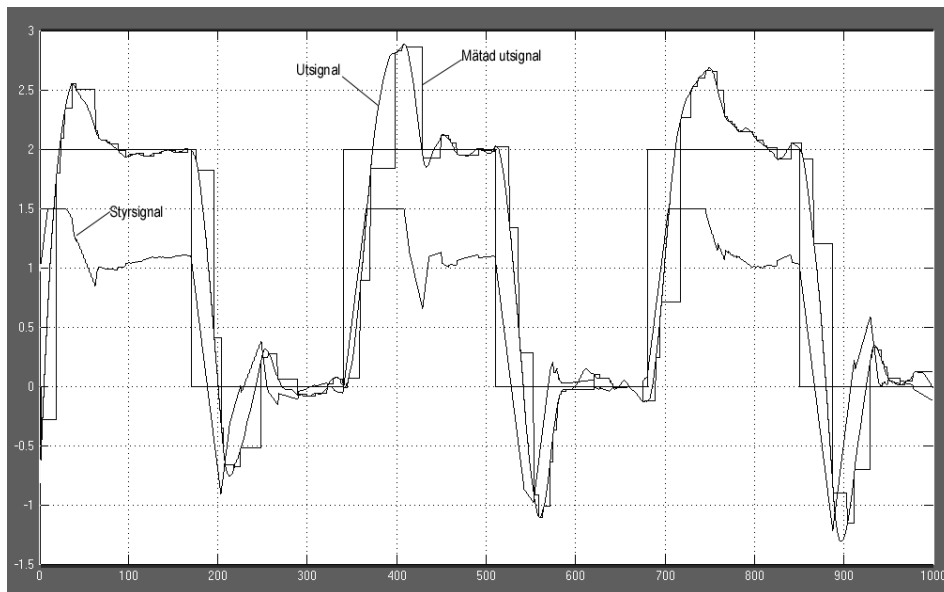


## PID simulering och asynkron tillståndsinläsning utan observerare

För att se hur bra observeraren var har simulering utan observerare utförts. Uppställningen visas i figur 11.3 och simuleringsresultat visas i figur 11.4. Man kan se kraftiga översvängar hos utsignalen.



Figur 11.3 Visar PID simuleringssupplättning utan observerare. Man ser hur inläst utsignal sparas undan med D-latch tills nytt värde inlästs. Gammalt utsignalvärde används då nytt värde saknas.



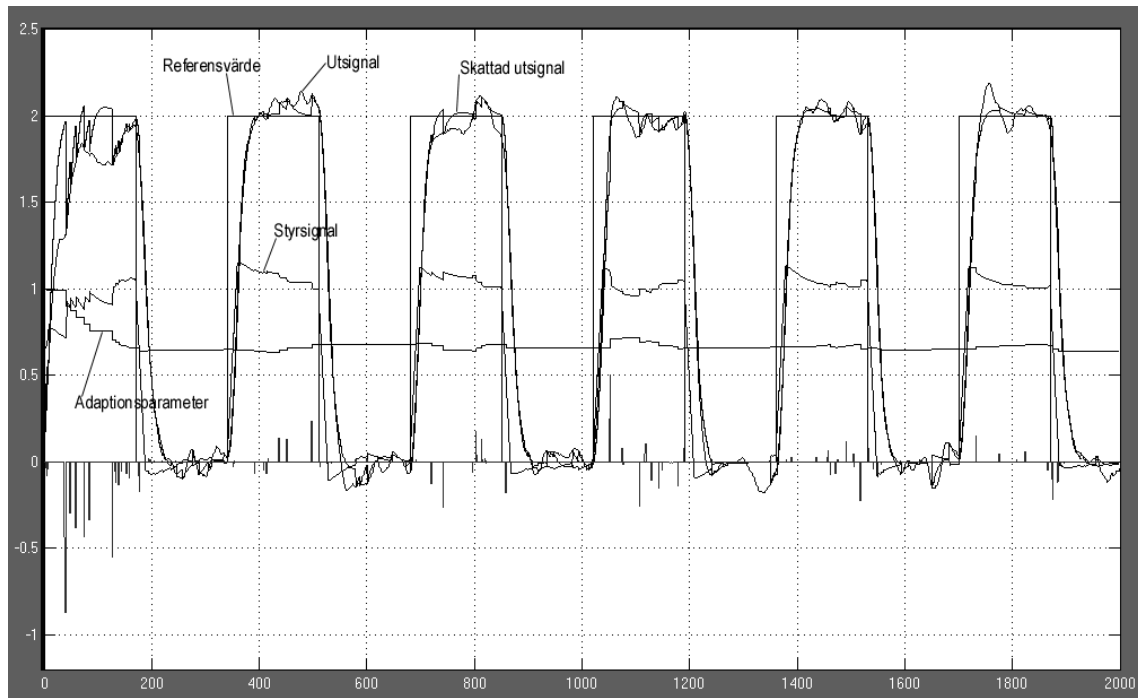
Figur 11.4 PID simulering utan observerare. Man ser referensvärde, utsignal, styrsignal (delat med 10) samt inläst utsignal. Observera att regulatorn försöker öka styrsignal (bottning) vid överslängarna (trots att verklig utsignal överstiger referensvärdet) på grund av för långsam mätdatauppdatering.

Även om det verkliga systemet störs av vågor, med amplitud upp till en meter och kortare periodtid än vad rodet har, ger tillståndsestimering en förbättrad reglering.

Utan observerare kan problemen uppstå på grund av insignal inte kommer tillräckligt ofta. Då har det gamla värdet blivit inaktuellt och regulatorn reglerar mot felriktning. Risken för integratoruppvridning ökar också. Simulering visar att ett estimerat värde är bättre än ett alltför gammalt.

### MIT regulator simulering med observerare

Största nackdelen med observerare är att processen inte tillåts ändras. Av den anledningen har även adaptivregulator (MIT regulator) med observerare simulerats. Simulerings resultat visas i figur 11.5. Resultatet är bra.



Figur 11.5 Visar resultat av adaptiv regulator med observerare. Adaptionparameter avtar från initialvärdet 1 mot ca 0.75. Styrsignalen är delad med 10.

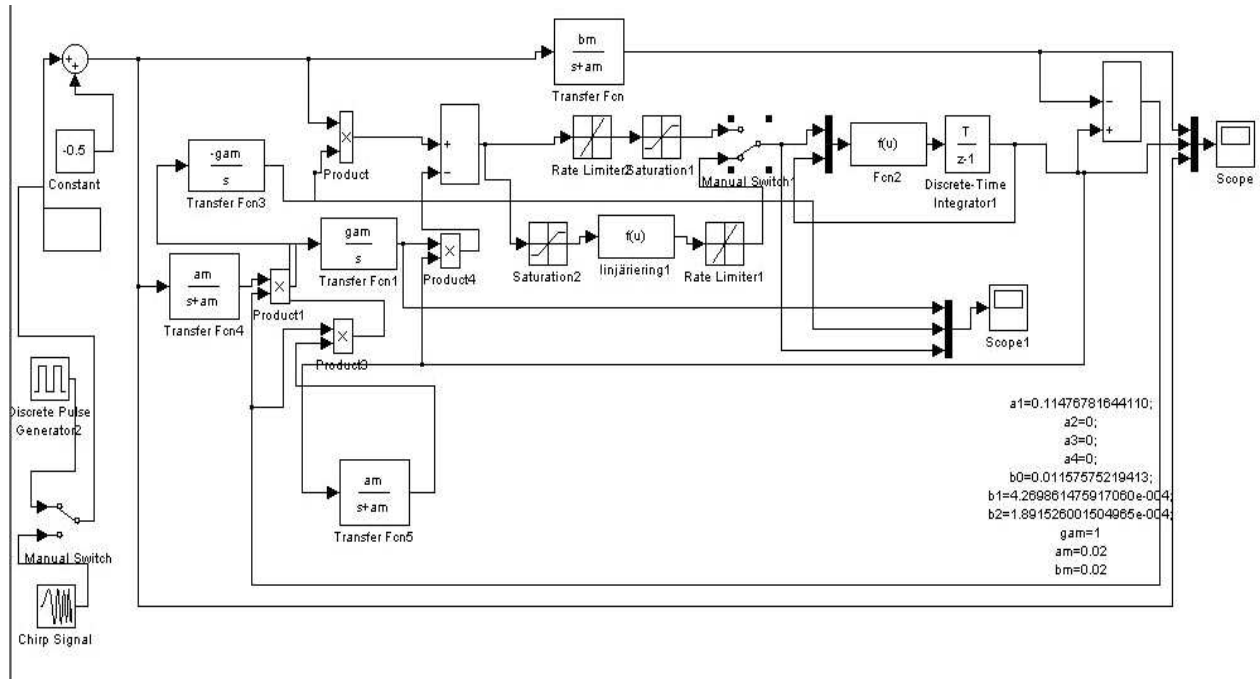
## **Referens**

- Karl J. Åström och Björn Wittenmark: *Computer-Controlled Systems*.
- Karl J. Åström och Björn Wittenmark: *Adaptive Control*.
- Rolf Johansson: *System Modeling and Identification*.
- T Glad och L Ljung: *Reglerteknik - Grundläggande teori*
- J.L Meriam L.g. Krage *Engineering mechanics* John Wiley & Sons, Inc.
- GPS handbok
- GuildLine handbok

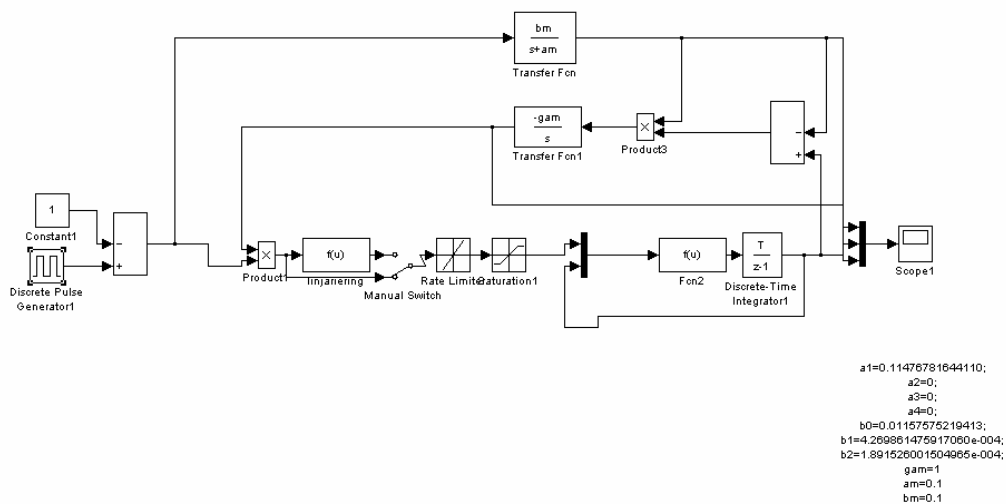
## ***Bilaga***

Simulink modell av Liapunovbaserat MRAS-regulator .....	49
MRAS baserad på MIT struktur .....	49
ident(selffrekvt).vi. huvudfönster .....	50
subvi SträngHandler.vi programstruktur .....	51
subvi RMBspeed.vi programstruktur .....	51
ident(selffrekvt).vi programstruktur .....	52
reglerPI.vi huvudfönster .....	53
subvi simplePID programstruktur .....	54
reglerPI.vi programstruktur .....	55
Marina signaler .....	56

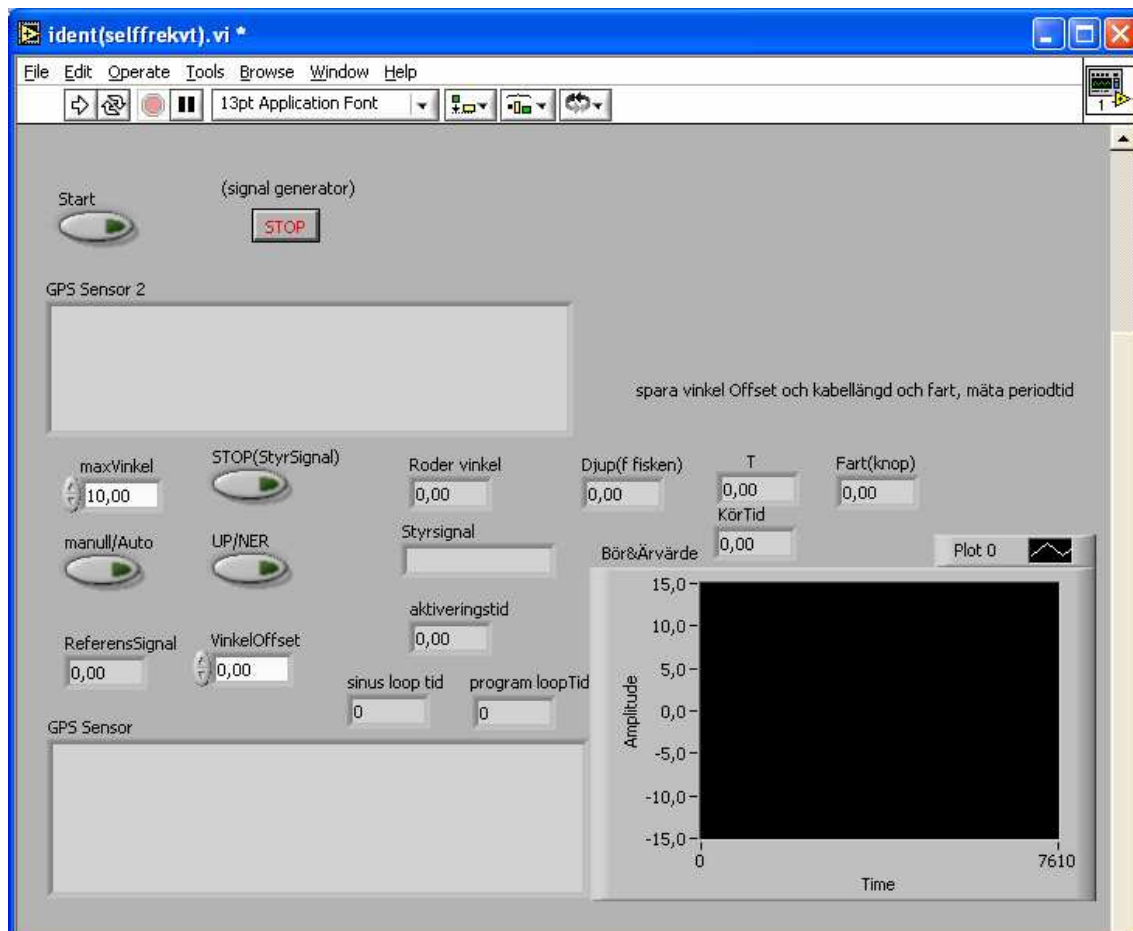
## Simulink modell av Liaponovbaserat MRAS-regulator



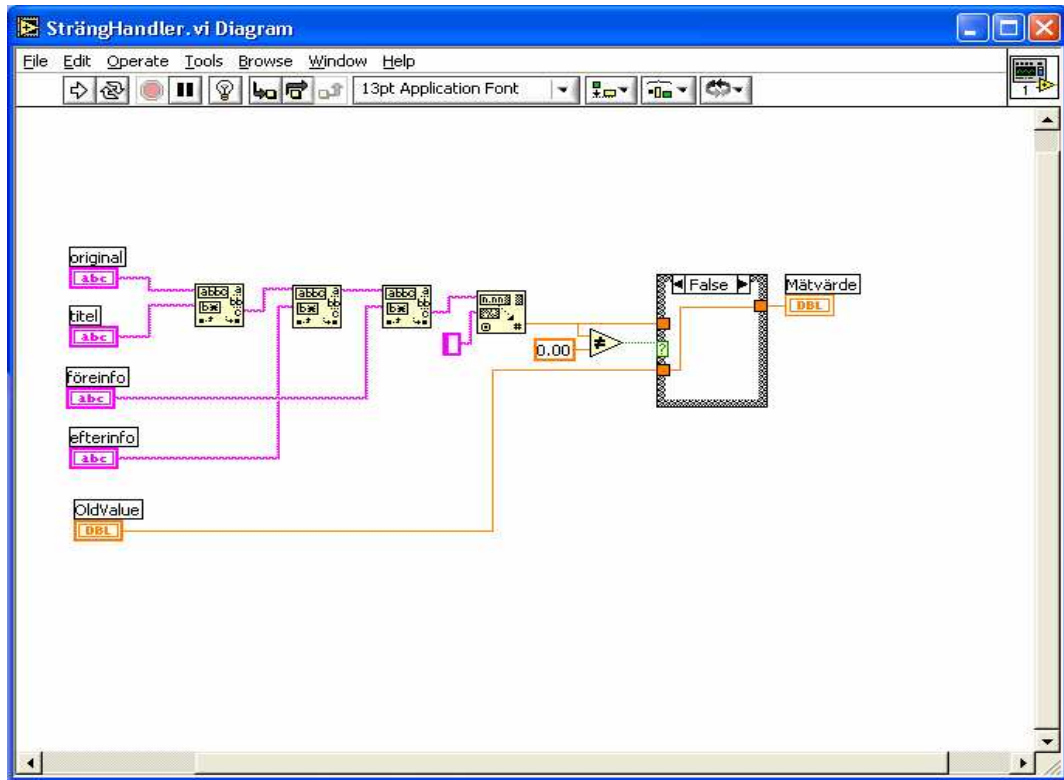
MRAS baserad på MIT struktur. Med linjäriserande block och kan väljas till eller från.



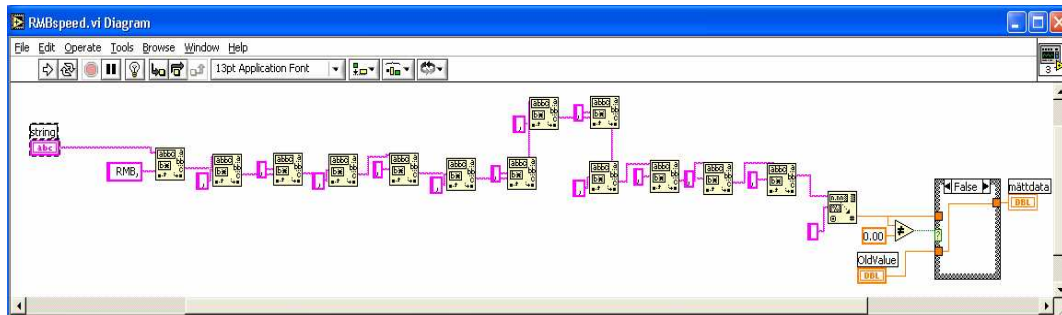
ident(selffrekvt).vi. huvudfönster. Huvudfönster för ident.vi ser liknande ut.



subvi SträngHandler.vi programstruktur



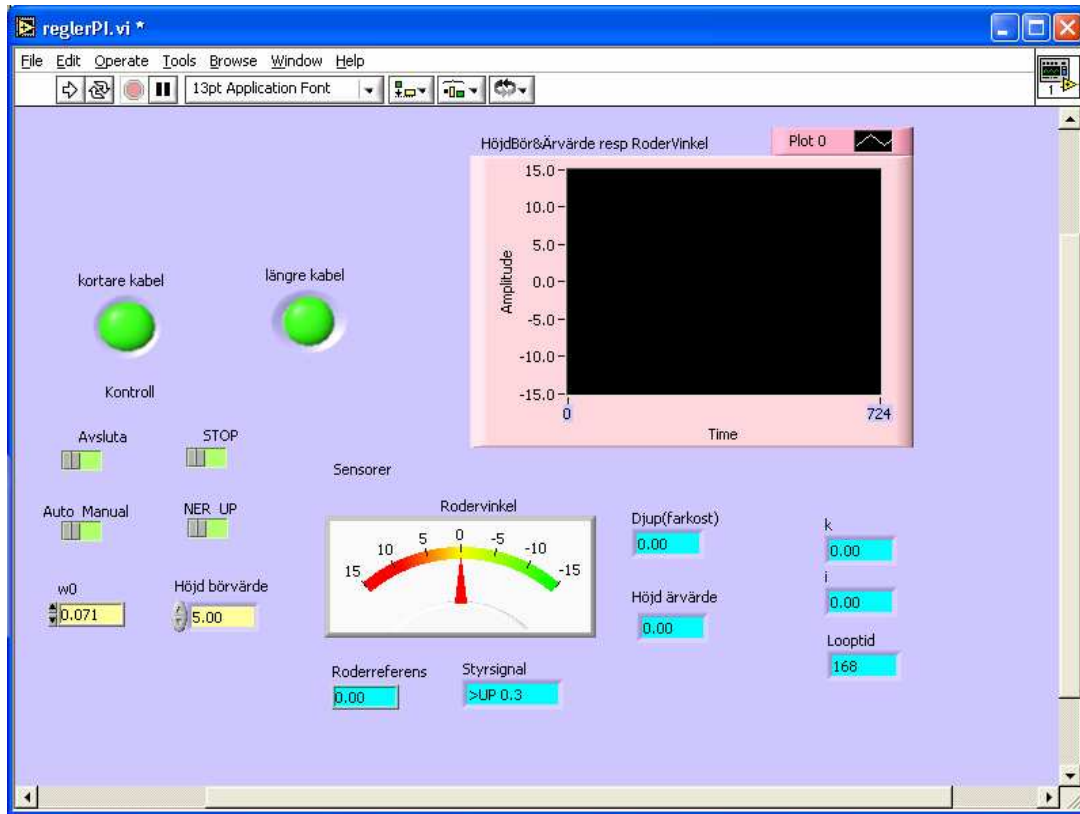
subvi RMBspeed.vi programstruktur





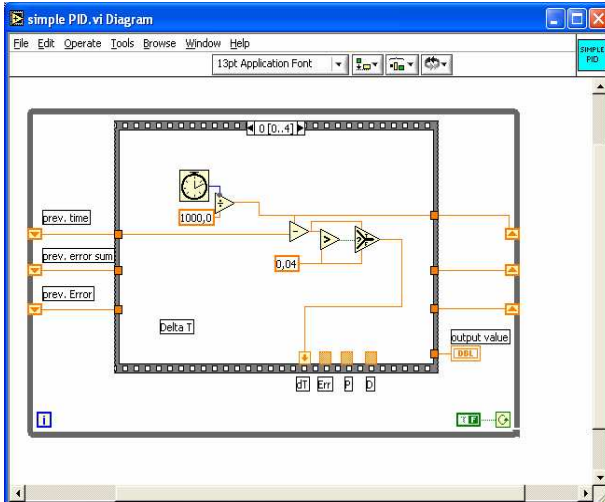


# reglerPI.vi huvudfönster

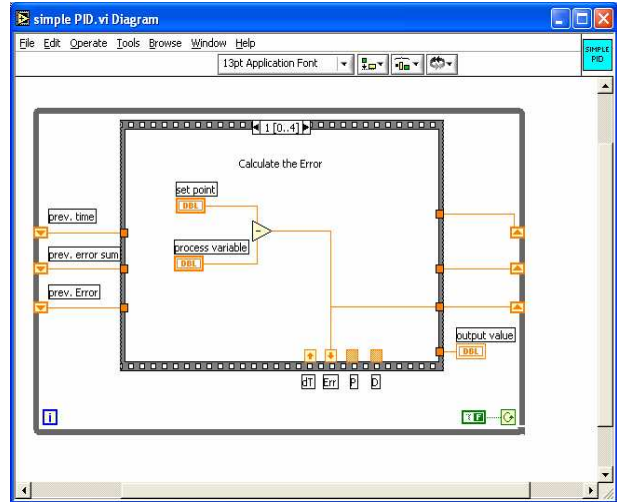


# subvi simplePID programstruktur

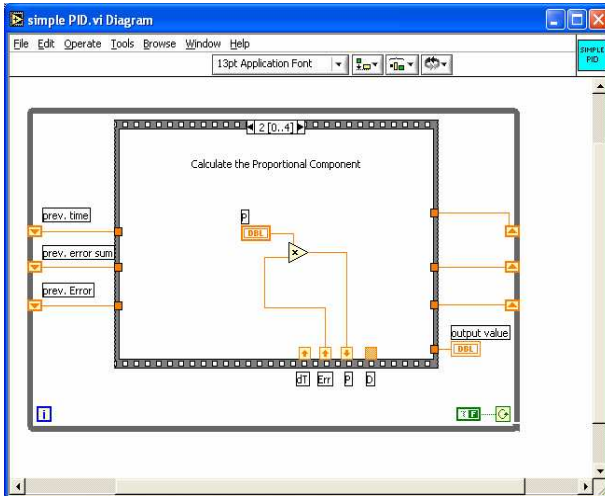
## Sidan 0



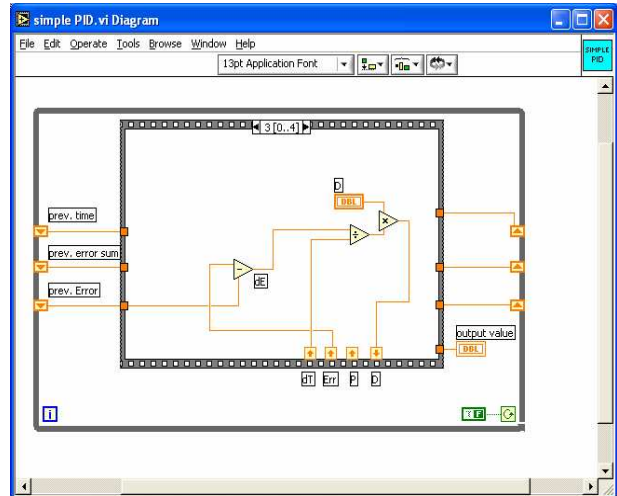
## Sidan 1



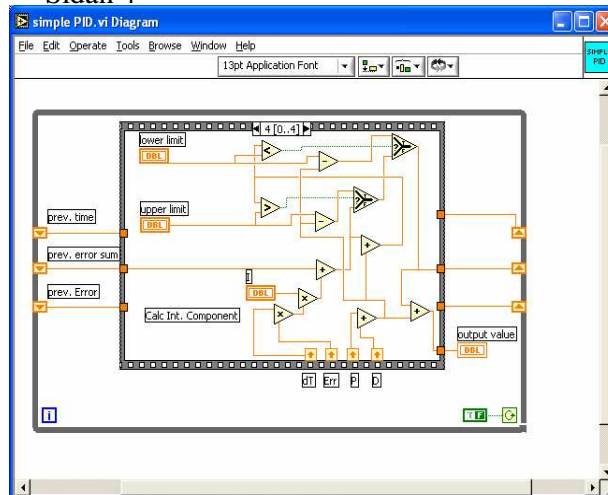
## Sidan 2



## Sidan 3

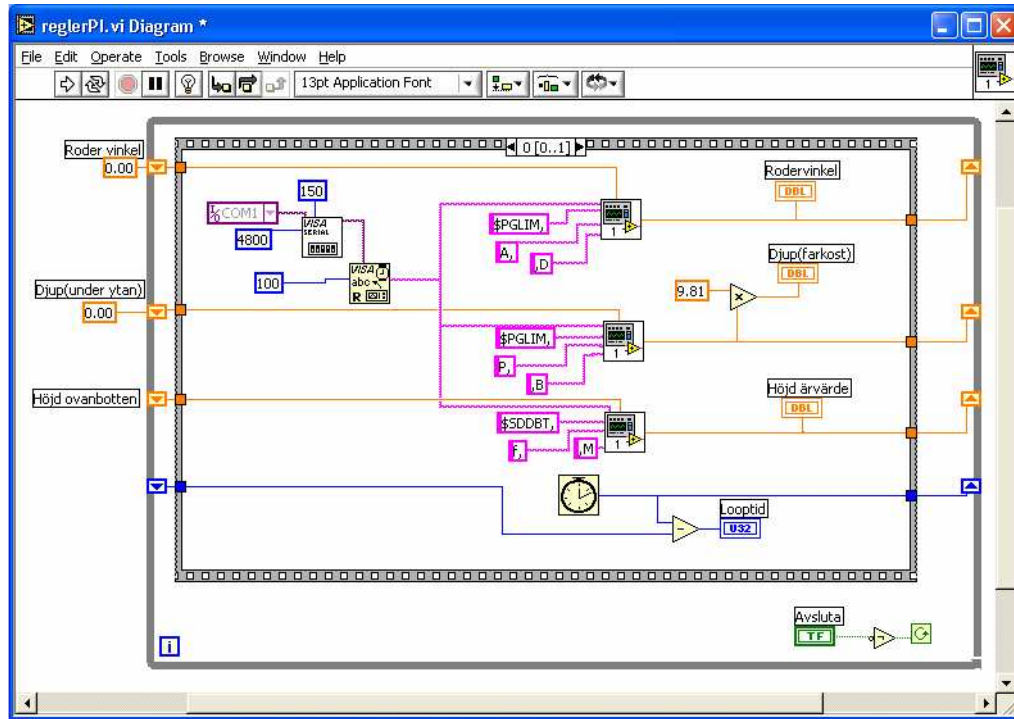


## Sidan 4

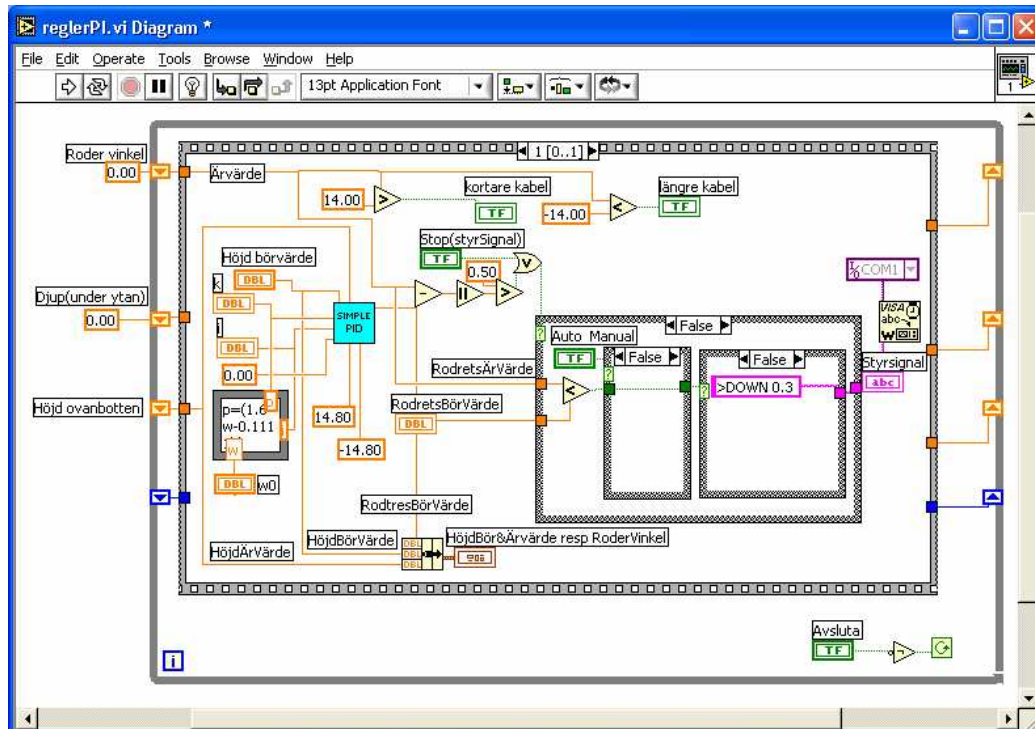


# reglerPI.vi programstruktur

## Sidan 0



## Sidan 1



Marina signaler:

Från COM1

- \$SDDBT,x.x,f,y.y,M,z.z,F\*hh

where

x.x	depth in feet
y.y	depth in meters
z.z	depth in fathoms
hh	checksum

- \$PGLIM,A,x.x,D,...P,y.y,B,...\*hh

where

x.x	wing angle in degrees
y.y	pressure in bars
hh	checksum

Från COM2

- RMB,A,0.66,L003,004,4917.24,N,12309.57,W,001.3,052.5,000.5,V\*0B

where

A	Data status A=OK, V=warning
0.66,L	Cross-track error, steer Left to correct (or R=right)
003	Origin waypoint ID
004	Destination waypoint ID
4917.24,N	Destination waypoint latitude 49 deg. 17.24 min. N
12309.57,W	Destination waypoint longitude 123 deg. 09.57 min.W
001.3	Range to destination, nautical miles
052.5	True bearing to destination
000.5	Velocity towards destination, knots
V	Arrival alarm A=arrived, V=not arrived
*0B	mandatory checksum