

# Trottelmodellering och olinjär trottelreglering

Examensarbete utfört i Fordonssystem  
vid Tekniska Högskolan i Linköping  
av

**Fredrik Karlsson**

Reg nr: LiTH-ISY-EX-1884



# Trottelmodellering och olinjär trottelreglering

Examensarbete utfört i Fordonssystem  
vid Tekniska Högskolan i Linköping  
av

**Fredrik Karlsson**

Reg nr: LiTH-isy-ex-1884

Handledare: **Lars Eriksson**

Examinator: **Lars Nielsen**

Linköping, 20 mars 1998.



# Sammanfattning

Elektronisk trottelsestyrning, ETS, integreras i allt fler bensinmotorer. ETS förbättrar framkomligheten och ökar säkerheten vid bilkörning samt motverkar att miljöfarliga ämnen bildas vid förbränningen. För att lyckas med detta krävs god trottelse reglering.

En olinjär tillståndsmodell, bestående av två tillstånd och fem parametrar, tas fram för trotteln. Modellen implementeras i SIMULINK för att möjliggöra identifiering och validering samt simulering och utvärdering av olika reglerstrategier.

En olinjär regulator designas, bestående av två olinjära kompensatorer samt en PI-regulator, intrimmad enligt Ziegler-Nichols metod. De olinjära kompensatorerna kompenserar för Coulombfriktionen och förspänningsmomentet hos systemet. Kompensatorerna konstrueras under förutsättning att vinkelhastigheten och förspänningen är kända, detta infrias genom att observera dem ur systemets tillståndsmodell.

Den olinjära regulatorn implementeras i tillgänglig hårdvara i laboratoriet och verifieras genom ett antal trottelsesteg. Prestanda hos den olinjära regulatorn jämförs, genom stegsvarsanalys, med prestanda hos en linjär PI-regulator. Resultatet av verifieringen är att trottelse regleringen fungerar bra över bensinmotorns hela arbetsområde.

**Nyckelord:** Coulombfriktion, dithering, tillståndsobservatör, olinjär kompensator.

# Tackord

Jag vill tacka alla på Fordonsystem för en stimulerande och trevlig tid, samt tillhandahållandet av ett väldigt intressant examensarbete. Speciellt vill jag tacka min handledare Lars Eriksson för värdefull hjälp och uppmuntrande tillrop i svåra stunder. Ett tack riktas också till Andrej Perkovic för hans tålamod och assistans vid motorkörningarna samt för hans programskelett för implementering av regulatorn. Erik Frisk förtjänar också ett omnämmande för sin hjälp med L<sup>A</sup>T<sub>E</sub>X. Till sist men inte minst vill tacka min flickvän och sambo Sofia Hedby för att hon gett mig mat och kärlek under denna arbetsamma period.

Linköping den 20 mars 1998

Fredrik Karlsson

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
1.1	Varför ETS? . . . . .	1
1.2	Rapporten i sammandrag . . . . .	2
<b>2</b>	<b>Modellering</b>	<b>3</b>
2.1	Beskrivning av systemet . . . . .	3
2.2	Modellering av systemet . . . . .	5
2.3	Tillståndsbeskrivning av modellen . . . . .	8
2.4	Implementering av modellen . . . . .	9
2.5	Mätningar inför identifieringen . . . . .	9
2.6	Identifiering av modellen . . . . .	11
2.7	Modellvalidering . . . . .	11
2.8	Dithering . . . . .	13
<b>3</b>	<b>Reglering</b>	<b>15</b>
3.1	Svårigheter med regleringen . . . . .	15
3.2	Kompensering av olinjäriteter . . . . .	15
3.3	Konstruktion av observatör . . . . .	17
3.4	PID-regulator . . . . .	22
3.5	Implementering av trottelregulator . . . . .	22
3.6	Verifiering av trottelregulator . . . . .	23
3.7	Analys av störningar vid trottelreglering . . . . .	28
<b>4</b>	<b>Utvidgningar</b>	<b>29</b>
<b>5</b>	<b>Slutsatser</b>	<b>30</b>
	<b>Referenser</b>	<b>31</b>
	<b>Bilaga A: Implementering av trottelmodell</b>	<b>32</b>
	<b>Bilaga B: Programkod för trottelregulatorn</b>	<b>39</b>





# 1 Inledning

Trotteln, eller spjällets, uppgift i en bensinmotor är att styra luftflödet in till cylindrarna. Detta görs genom att variera trottelvinkeln, det vill säga trottelskivans läge, så att rätt mängd luft släpps igenom och att rätt effektutveckling uppnås. I äldre bilar så styrs trotteln mekaniskt på så sätt att föraren, med hjälp av gaspedalen, direkt ställer ut trottelvinkeln. I modernare motorer är det vanligare att ställa ut förarens önskade trottelvinkel via elektronisk trottelsestyrning. Elektronisk trottelsestyrning (ETS) bidrar till att minska bränsleförbrukningen och utsläppet av miljöfarliga ämnen hos motorn samt att förbättra framkomligheten i trafiken och att öka säkerheten på vinterunderlag. För komfort hos föraren bidrar ETS till en mjukare övergång mellan olika körstadier, till exempel vid växling med en automatlåda.

I Fordonssystemets laboratorium finns en SAAB 2.3L bensinmotor med ETS. Reglersystemet består av en linjär parameterstyrd PI-regulator, där regulatorparametrarna är framtagna enligt *Ziegler-Nichols* metod. Parameteruppsättningen hos regulatorn styrs av varvtalet hos bensinmotorn. Implementeringen av regulatorn är utförd i det styrdon som finns i laboratoriet. I styrdonet, som är tillverkat av Mecel, utförs all signalberäkning. Reglersystemet motsvarar inte de krav som ställs på utrustning som används i en forskningsmiljö, främst med avseende på snabbheten hos regulatorn. Trottelsegulatorn används ofta, i experiment, som en inre regulator och då är just snabbheten hos regleringen viktig.

## 1.1 Varför ETS?

Elektronisk trottelsestyrning (ETS) innebär att istället för att mekaniskt ställa ut trottelvinkeln med hjälp av gaspedalen, så görs detta via elektronisk styrning, till exempel med en likströmsmotor. Anledningarna till denna ersättning är att ibland är det inte lämpligt eller nödvändigt att låta föraren styra luftflödet med sin gaspedal. Det kan gälla vid följande tillämpningar [4].

- Luft/bränsle reglering
- Katalysatoruppvärmning
- Momentreglering vid mager körning
- Farthållning
- Antispinnsystem

Luft/bränsle reglering innebär en reglering av blandningsförhållandet mellan luft och bränslemängd. För att minska utsläppet av miljöfarliga ämnen och för att minska bränsleförbrukningen hos bensinmotorn så bör förhållandet mellan luft och bränsle vara olika för olika körstadier.

För att nå bästa effekt med katalysatorn, så att utsläppet av miljöfarliga ämnen minimeras, ska uppvärmningen av den ske snabbt då motorn startas. Detta uppnås

genom att styra luft/bränsleblandningen och tändvinkeln hos motorn. Motorns effektutveckling kommer dock att påverkas, men genom att införa ETS kan detta kompenseras så att föraren inte märker någon skillnad mellan körning i uppvärmningsläge och normal körning. En reduktion av utsläppet av miljöfarliga ämnen fås också genom att låta bensinmotorn gå på en mager luft/bränsleblandning, det vill säga mycket luft och lite bränsle. För att få maximal effektutveckling i motorn krävs dock en fetare blandning. Övergången mellan dessa två fall bör vara momentan så att motorn slipper gå på en "halvmager" blandning, eftersom det ger höga kväveoxidhalter i avgaserna. Detta kan uppnås med hjälp av ETS.

Vid farthållning och i antispinnsystem är användningen av ETS naturlig. Vid farthållning så bibehålls angiven hastighet oavsett vägbanans lutning, genom att tillföra rätt luft/bränsleblandning till cylindrarna så att rätt effektutveckling hos bensinmotorn uppnås. I ett antispinnsystem så minskar trottelloppningen och därmed motorns drivande moment när hjulen slirar mot underlaget. Detta förbättrar framkomligheten i trafiken och ökar säkerheten vid körning på vinterunderlag.

Just detta att utsläppet av miljöfarliga ämnen reduceras och att bränsleförbrukningen minskas samt en förbättrad säkerhet vid bilkörning kommer att göra att allt fler bilmotorer utrustas med elektronisk trottelsestyrning. Framtiden för ETS ser ljus ut.

## 1.2 Rapporten i sammandrag

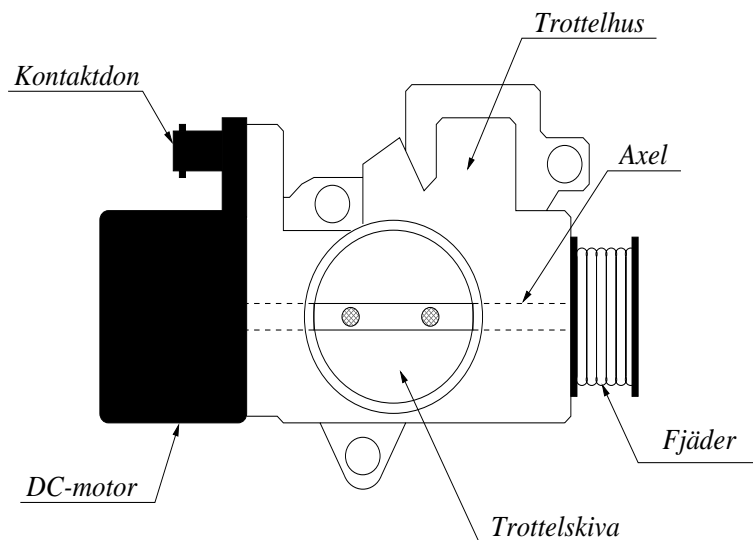
I inledningen ges en bakgrund till varför allt fler motorer utrustas med elektronisk trottelsestyrning. Kapitel 2 lägger grunden för regulatordesignen, då en fysikalisk modell tas fram för systemet. Modelleringen består av en uppdelning av systemet i delsystem och en ingående beskrivning av delsystemen ges. Den framtagna modellen implementeras i SIMULINK för att möjliggöra identifiering och slutligen validering. I kapitel 3 beskrivs själva reglerdesignen. Regleringen består av en olinjär regulator, som ges av kompensering av olinjäriteter och en PI-regulator. För att kunna kompensera olinjäriteterna så konstrueras två tillståndsobservatörer. Avslutningsvis genomförs en analys av störningar vid trottelse regleringen. Rapporten avslutas med förslag på fortsatt arbete och slutsatser från detta examensarbete.

## 2 Modellering

Syftet med att ta fram en matematisk modell av systemet är att den skall öka förståelsen för systemet samt användas för regulatordesign. Modellen implementeras i SIMULINK för att möjliggöra identifiering och validering av modellen samt simulering och utvärdering av olika reglerstrategier. Ett sätt att modellera är att ta fram en skräddarsydd fysikalisk modell. För detta krävs en fysikalisk förståelse för systemet så modelleringen börjar med en genomgång av de komponenter som ingår i systemet.

### 2.1 Beskrivning av systemet

Systemet som skall modelleras består av en likströmsmotor (DC-motor), ett trottellhus med trottelskiva och axel samt en återföringsfjäder. En bild av systemet visas i figur 2.1.

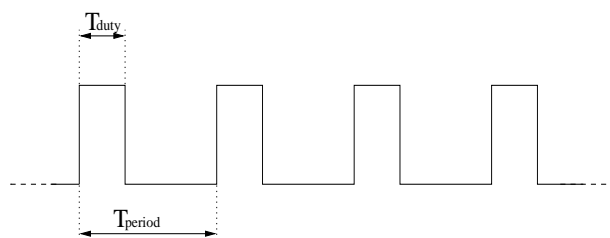


**Figur 2.1.** Bilden visar systemet ovanifrån, med DC-motorn till vänster, trottelluset med trottelskivan i mitten och återföringsfjädern till höger. Den genomgående axeln syns streckad, och längst upp till vänster finns ett kontaktdon för styrdonet.

#### DC-motorn

DC-motorn är en separatmagnetiserad likströmsmotor, och står för det drivande momentet i systemet. Insignalen till DC-motorn är en pulsbreddsmodulerad signal (PWM-signal), se figur 2.2. Det innebär kortfattat att signalen består av ett pulståg där bredden på pulserna avgör storleken på signalen och anges ofta i procent

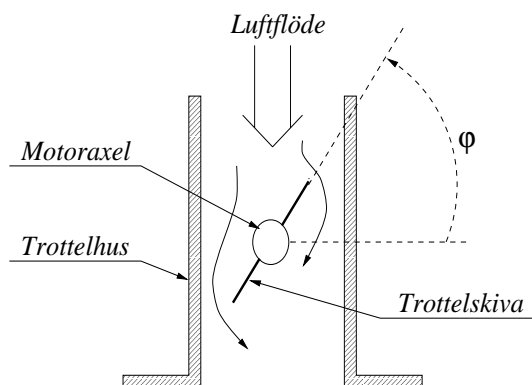
av maximal styrsignal. Att insignalen är vald till en PWM-signal beror på flera anledningar: det är billigt, enkelt och denna teknik har visats sig linjärisera system som har vissa olinjäriteter inbyggda [1].



**Figur 2.2.** Insignalen till DC-motorn består av ett pulståg, där bredden på pulserna,  $T_{\text{duty}}$ , i förhållande till periodtiden,  $T_{\text{period}}$ , avgör storleken. Här är det bredden på pulserna som styrs av insignalen,  $u$ , och förhållandet mellan  $T_{\text{duty}}$  och  $T_{\text{period}}$  är samma som mellan insignalen och maximal styrsignal.

### Trotteln

Trottelskivan, även kallad spjäll, och dess vinkel i förhållande till trottelhuset avgör hur mycket luft som kommer in till bensenmotorn. En principskiss på luftflödet visas i figur 2.3. Trottelvinkeln kan variera mellan noll och nittio grader, vilket motsvarar stängd respektive helt öppen trottel. Förhållandet mellan trottelvinkeln och trottelhuset visas också i figur 2.3. Den axel som trottelskivan sitter fast på är samma axel som DC-motorn verkar på. Det är alltså en direkt överföring mellan DC-motorn och trottelskivan.



**Figur 2.3.** Principskiss på luftflödet genom trotteln. Här visas också trottelvinkeln,  $\varphi$ , och dess förhållande med trottelhuset.

## Fjädern

Fjäders funktion i systemet är att den skall dra tillbaka trottelskivan och hindra bensinmotorn från att rusa iväg om DC-motorn skulle bli strömlös. Den fungerar alltså som en återföringsfjäder. Fjädern är också förspännd med ett antal grader, för att alltid garantera ett återföringsmoment.

## 2.2 Modellering av systemet

I detta avsnitt kommer en matematisk modell av systemet att tas fram med utgångspunkt från Newtons andra lag för roterande massor. Två grundläggande frågor som är avgörande för hur modellen kommer att se ut är. Hur komplex måste modellen vara för att en bra reglerdesign ska kunna göras? Hur komplex modell går att identifiera med de fysiska mätningar som går att utföra? Erfarenhetsmässigt så går det att designa bra regulatorer med relativt enkla modeller, speciellt då återkoppling används [5]. Strävan i detta modellbygge är att knyta ihop dessa trådar gällande komplexitet, så att modellen är enkel men fortfarande fångar systemets dynamik.

### Blockschema

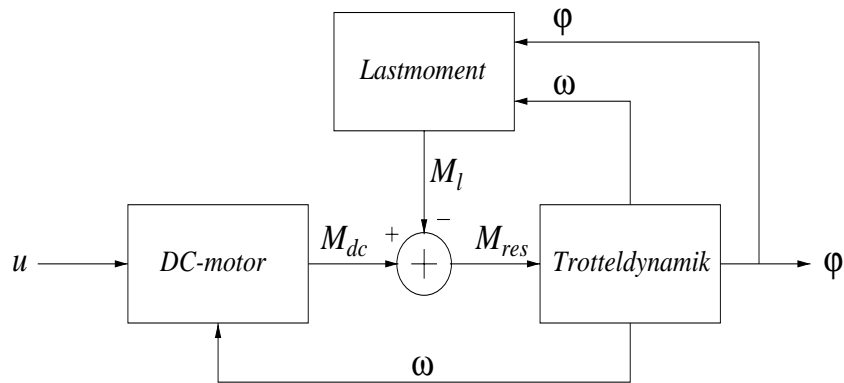
För att få en bättre överblick över systemet så delas det upp i följande tre delsystem.

Delsystem	Beskrivning
DC-motor	Likströmsmotor som ger det elektriska momentet
Lastmoment	Motverkande moment i systemet
Trotteldynamik	Skiv- och axeldynamik hos systemet

Det elektriska momentet,  $M_{dc}$ , som DC-motorn producerar beror på pulsbredden hos insignalen,  $u$ , och vinkelhastigheten,  $\omega$ , hos trotteln. Lastmomentet,  $M_l$ , beror på trottelvinkeln,  $\varphi$ , och vinkelhastigheten,  $\omega$ . Trottelskivans vinkel,  $\varphi$ , beror på det resulterande momentet,  $M_{res}$ , som verkar på trottelaxeln. Sambandet mellan in- och ut signaler för delsystemen beskrivs nu av följande tabell.

Variabel	Beskrivning	Insignal till	Utsignal från
$u$	Spänning	DC-motorn	-
$M_{dc}$	Elektriskt moment	-	DC-motorn
$\varphi$	Trottelvinkel	Lastm./DC-mot.	Trotteldynamiken
$\omega$	Vinkelhastighet	Lastmomentet	Trotteldynamiken
$M_l$	Motverkande moment	-	Lastmomentet
$M_{res}$	Resulterande moment	Trotteldynamiken	-

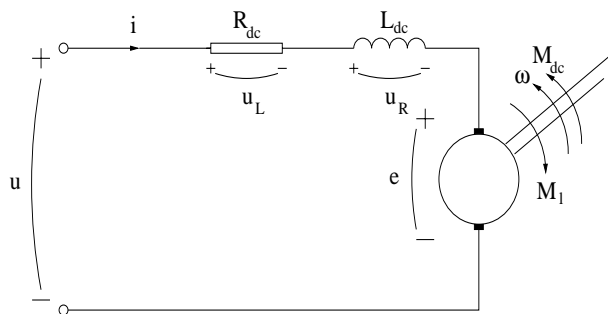
För att underlätta modelleringen och för att få en bättre överblick på delsystemen och sambanden mellan dem så representeras systemet med ett blockschema, se figur 2.4. I blockschemat fås ett naturligt flöde av signaler från insignal, via interna variabler, som till slut resulterar i en utsignal. Uppdelningen gör att det går att ställa upp matematiska samband för de enskilda delsystemen.



**Figur 2.4.** Blockschemabeskrivning av systemet, som delats upp i tre delsystem. Här visas också in och utsignalsambanden mellan de olika delsystemen.

## DC-motorn

DC-motorn är en separatmagnetiserad likströmsmotor, vilket innebär att den kan modelleras som en seriekoppling av en resistans, en induktans och en motverkande emk. Denna ekvivalenta krets för DC-motorn visas i figur 2.5.



**Figur 2.5.** Ekvivalent krets för en separatmagnetiserad DC-motor. DC-motorn kan ses som en resistans i serie med en induktans och en motemk. Detta svarar mot rotorkretsen hos DC-motorn.

Sambanden för DC-motorn ges av följande ekvationer, som innefattar *Ohms lag*, *Kirchoffs spänningslag* och proportionaliteten mellan motemk och rotationshastigheten.

$$M_{dc} = K_{dc} i \quad (2.1)$$

$$e = K_e \omega \quad (2.2)$$

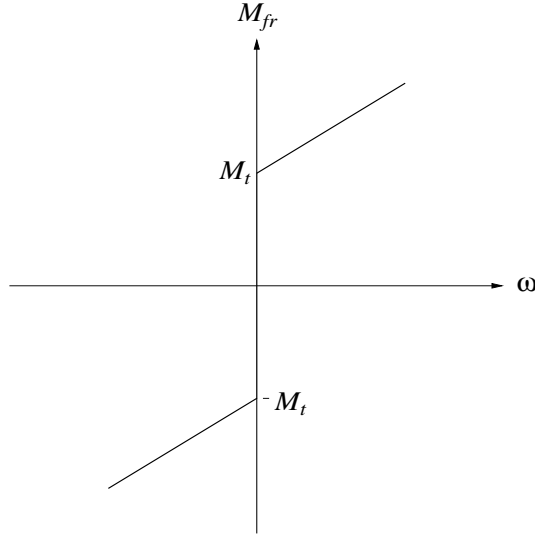
$$\frac{di}{dt} = \frac{u - R_{dc} i - e}{L_{dc}} \quad (2.3)$$

## Lastmomentet

Lastmomentet är det delsystem som innehåller alla motverkande moment på systemet. De motverkande momenten är fjädermomentet och friktionsmomentet. Fjädersystem antas vara linjär med en viss förspänning,  $\varphi_0$ , vilket ger.

$$M_{fj} = K_{fj} (\varphi + \varphi_0) \quad (2.4)$$

Friktionsmomentet modelleras som *stiction*, vilket är en statisk friktionsmodell i motsats till en dynamisk [10]. Det är en kombination av *Coulombfriktion*, *viskös friktion* samt en beskrivning av friktionen vid vila. En principskiss av Coulombfriktionen och den viskösa friktionen visas i figur 2.6.



**Figur 2.6.** Friktionsmodell för Coulombfriktionen och den viskösa friktionen hos systemet. Offsetnivån,  $M_t$ , ges av Coulombfriktionen och lutningen ges av den viskösa friktionen.

Vid vila innebär Coulombfriktion och stiction att det resulterande momentet som verkar på trotteln,  $M_{res}$ , måste komma över en viss gräns,  $M_t$ , för att trotteln skall börja röra på sig. Vid rörelse innebär det att om det resulterande momentet understiger gränsen samtidigt som vinkelhastigheten blir noll så fastnar trotteln. Viskös friktion modelleras proportionellt mot vinkelhastigheten, med en konstant  $K_{fr}$ . Detta innebär att friktionen för systemet inte kan beskrivas som en funktion av enbart vinkelhastigheten, utan den beror också på det externa moment som verkar på systemet. Nu kan följande ekvation för friktionen ställas upp.

$$M_{fr} = \begin{cases} M_{res} & \omega = 0, |M_{res}| \leq M_t \\ K_{fr}\omega + M_t \operatorname{sign}(\omega) & \text{annars} \end{cases} \quad (2.5)$$

Det totala lastmomentet beskrivs av följande ekvation

$$M_l = M_{fj} + M_{fr} \quad (2.6)$$

där fjädermomentet,  $M_{fj}$ , och friktionsmomentet,  $M_{fr}$ , ges av ekvation 2.4 respektive 2.5.

### Trotteldynamik

Enligt Newtons andra lag så bestäms vinkelhastigheten  $\omega$  för en roterande massa med tröghetsmomentet  $J$  av följande differentialekvation

$$\frac{d}{dt}(J\omega) = M_d - M_l \quad (2.7)$$

där  $M_d$  är det drivande momentet och  $M_l$  är lastmomentet. I tröghetsmomentet  $J$  så ingår trottelskivans, trottelaaxelns och motoraxelns tröghetsmoment. Förhållandet mellan trottelvinkeln och vinkelhastigheten ges av följande enkla samband.

$$\frac{d\varphi}{dt} = \omega \quad (2.8)$$

### 2.3 Tillståndsbeskrivning av modellen

Genom att skärskåda ovanstående ekvationerna så framgår att ett naturligt val av tillståndsekvationer är 2.3, 2.7 och 2.8. Dessa ekvationer är de som i någon form innehåller en tidsderivata. Med tillstånden  $\varphi$ ,  $\omega$  och  $i$  kan nu följande tillståndsmodell ställas upp.

$$\frac{d\varphi}{dt} = \omega \quad (2.9)$$

$$\frac{d\omega}{dt} = \begin{cases} 0 & , \omega = 0, |M_{res}| \leq M_t \\ -\frac{K_{fj}}{J}\varphi - \frac{K_{fr}}{J}\omega + \frac{K_{dc}}{J}i & , annars \\ -\frac{K_{fj}}{J}\varphi_0 - \frac{M_t}{J}\text{sign}(\omega) & \end{cases} \quad (2.10)$$

$$\frac{di}{dt} = -\frac{K_e}{L_{dc}}\omega - \frac{R_{dc}}{L_{dc}}i + \frac{1}{L_{dc}}u \quad (2.11)$$

Tillståndsmodellen för systemet består av tre tillstånd och totalt nio okända konstanter. Med den begränsning som finns i antalet mätbara signaler, som kan användas till identifiering av konstanter, måste en förenkling av modellen göras.

### Förenkling av tillståndsmodellen

Vid antagande att induktansen,  $L_{dc}$ , hos DC-motorn är liten och att uppladdningsförloppet, den elektriska tidskonstanten, är snabbt, i förhållande till andra



tidskonstanter i systemet, så kan spänningsförlusten  $u_L$  (se figur 2.5) försummas [6]. Tillståndsmodellen kan då skrivas som

$$\frac{d\varphi}{dt} = \omega \quad (2.12)$$

$$\frac{d\omega}{dt} = \begin{cases} 0 & , \omega = 0, |M| \leq c_2 \\ -a_1\varphi - a_2\omega + bu & , \text{annars} \\ -c_1 - c_2 \text{sign}(\omega) & \end{cases} \quad (2.13)$$

där

$$M = bu - a_1\varphi - c_1 \quad (2.14)$$

Resultatet av modelleringen och förenklingen är en tillståndsmodell med två tillstånd och fem okända konstanter.

## 2.4 Implementering av modellen

Modellen implementeras i SIMULINK [3], som är ett mjukvarupaket för modellering, simulering och analys av dynamiska system. SIMULINK ingår som en toolbox i MATLAB [2]. Både linjära och olinjära system kan implementeras. Detta kan göras i kontinuerlig tid, diskret tid eller en kombination av dessa. För modelleringen används ett grafiskt användargränssnitt (GUI) så att modellen byggs som ett blockdiagram. Modellen byggs upp hierarkiskt, först av större block som sedan blir mer och mer detaljerade. Implementeringen åskådliggörs i bilaga A.

## 2.5 Mätningar inför identifieringen

Modellen innehåller fem okända parametrar som måste identifieras. Dessa identifieras genom att anpassa modellen till mätningar på det verkliga systemet. Innan några mätningar utförs är det viktigt att bestämma modellens giltighetsområde, samplingstid och vad som mätningarna ska innehålla. Alla mätningar är utförda på den trotteln som sitter på SAAB 2.3L bensinmotorn i Fordonsystems laboratorium. Till bensinmotorn finns en tillhörande elektrisk motorbroms, som ej bör belastas av moment högre än ca 150 Nm. De mätbara storheterna är insignalen,  $u$ , till DC-motorn (innan den omvandlas till en PWM-signal) samt trottelvinkeln,  $\varphi$ .

### Modellens giltighetsområde

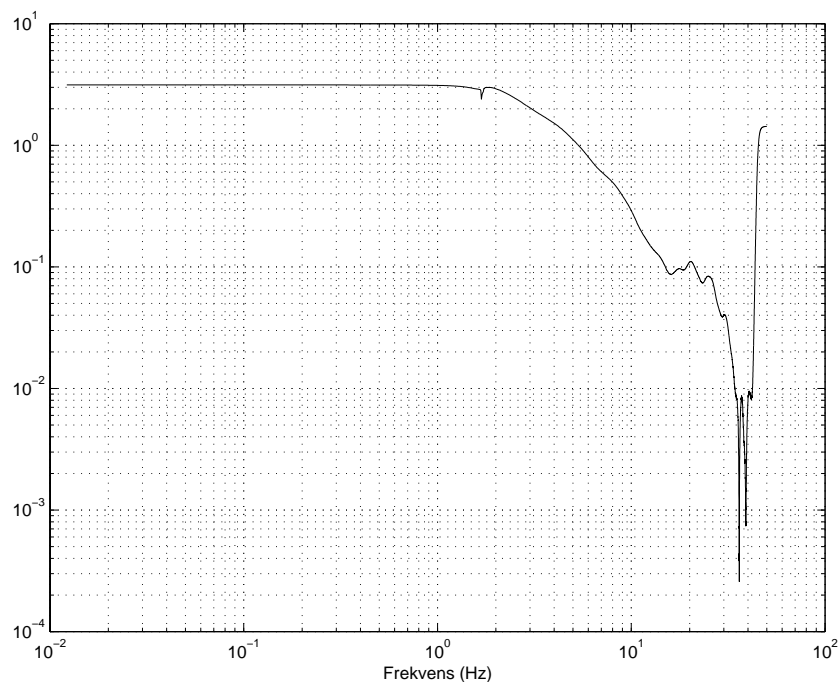
Största trottelvinkel som är tillåten vid en motorkörning utan att överbelasta motorbromsen är 30 % av maximal trottelvinkel. Detta ger att modellens giltighetsområde minst bör täcka detta område, men även fånga beteendet i det övriga området. Minsta området är.

$$0 \leq \varphi \leq 300$$

Giltighetsområdet är uttryckt i promille, där noll promille motsvarar noll grader och tusen promille motsvarar nittio grader, 300 promille motsvarar då 27 grader.

## Samplingstid

Valet av samplingsfrekvens är kopplat till systemets tidkonstanter. Det är viktigt att varken sampla för fort eller för sakt [7]. Sampling som är för snabb leder till dataredundans och sampling som är för långsam leder till en dålig beskrivning av systemets dynamik. En tumregel är att välja samplingsfrekvensen till minst 10 gånger systemets bandbredd. Systemets bandbredd kan uppskattas genom att göra en spektralskattning av utsignalen då en telegrafsignal används som insignal. En spektralskattning av systemets överföringsfunktion [7] beskrivs i figur 2.7 och bandbredden för systemets uppskattas till cirka 6 Hz. En lämplig samplingsfrekvens blir då omkring 60 Hz. Då processorn i styrdonet redan arbetar med en samplingsfrekvens på 100 Hz, väljs den eftersom den är tillräcklig.



**Figur 2.7.** Spektralskattning av systemets överföringsfunktion. Figuren beskriver den empiriska överföringsfunktionsskattningen. Systemets bandbredd kan här avläsas till cirka 6 Hz.

## Val av insignal för identifiering

De mätdata som samlas in för identifieringen måste excitera all dynamik hos systemet. Insignalen väljs då så att den har sin energi i samma frekvensband som systemet. Experimenten utförs i form av stegsvarstester där insignalen till DC-

motorn varierar i form av en fyrkantvåg med slumpmässigt varierande pulsbredd, och den resulterande trottelvinkeln mäts som utsignal. Storleken på stegen varierar över hela giltighetsområdet och längden på dem ligger mellan 0.2-0.5 sekunder. Snabbare trottelseg än 0.2 sekunder begränsas av bandbredden hos systemet och långsammare trottelseg än 0.5 sekunder leder till att olinjäriteten som härör från att insignalen till systemet är en PWM-signal slår igenom. PWM-signalens puls-beteende i samband med friktionen gör att trotteln inte riktigt fastnar utan ligger hela tiden på gränsen. Detta kallas för *dithering* och diskuteras mer ingående i avsnitt 2.8.

Mätningarna utförs i två omgångar, där den första uppsättningen mätdata används till identifiering och den andra uppsättningen används till validering.

## 2.6 Identifiering av modellen

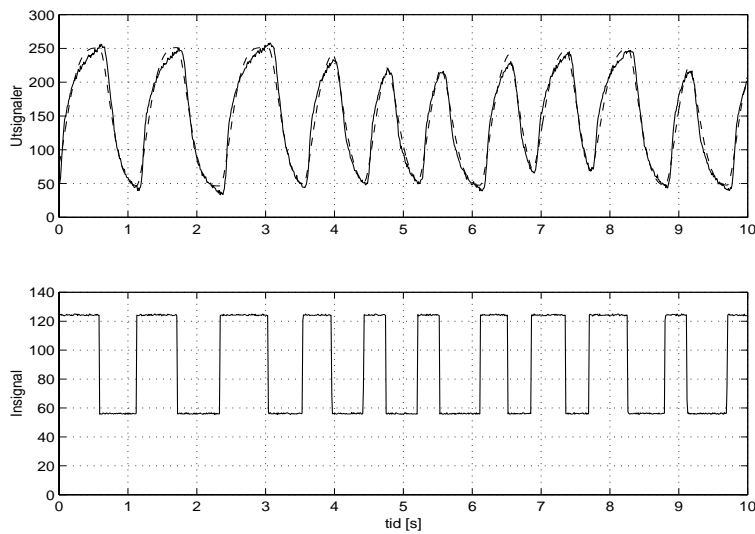
Modellparametrarna bestäms genom att anpassa modellen till ett antal trottelseg [7]. Viktigt är då att dynamiken hos systemet fångas, så regulatordesignen kan anpassas till systemets dynamik. Detta uppnås genom att använda den insignalsekvens som beskrivs i avsnitt 2.5. En kortfattad beskrivning av hur identifieringen går till är att vald insignalsekvens skickas in till det verkliga systemet där utsignalen mäts. Samma insignalsekvens skickas sedan in till modellen och därefter minimeras kvadraten på felet mellan de båda utsignalerna genom att succesivt ändra på parametrarna i modellen. Parameteruppsättningen till modellen som beskrivs av ekvation 2.12 och 2.13 skattas då till följande värden.

$$\begin{aligned}a_1 &= 66 \\a_2 &= 12 \\b &= 239 \\c_1 &= 11800 \\c_2 &= 1900\end{aligned}$$

Denna parameteruppsättning beskriver systemet bra vid simulering för olika trottelseg. Vid experiment på systemet i stationära förlopp så är överensstämmelsen inte lika bra. Detta beror på att parametrarna är framtagna med hjälp av transientanalys och då har inte olinjäriteten som härör från att insignalen är en PWM-signal lika stor inverkan.

## 2.7 Modellvalidering

Utvärderingen av modellen görs på "färska" mätdata. De mätdata som används till valideringen innehåller trottelseg som exciterar systemets dynamik. Resultatet av valideringen presenteras i figur 2.8. Figuren visar att dynamiken hos systemet är väl beskriven, medan nivåerna inte stämmer helt, det vill säga de stationära förloppen som är beroende av PWM-signalen. Detta är dock inte så allvarligt då det är relativt enkelt att reglera bort med integrerande regulatorer.



**Figur 2.8.** Validering av trottellmodellen. I den övre figuren så beskriver den heldragna kurvan utsignalen från det verkliga systemet och den streckade kurvan svarar mot modellens utsignal. I den undre figuren visas den insignal som skickades in, både till systemet och till modellen. Figuren visar att modellen beskriver dynamiken väl.

Det går inte att beräkna några poler eller nollställen för den olinjära modellen. Men genom att försumma Coulombfriktionen och fjäderförspänningen kan systemet beskrivas av en andra ordningens överföringsfunktion. Idén är att ta fram sambanden mellan systemets tidsvar och systemets poler och nollställen [5]. Detta bör ge en uppskattning om den skattade bandbredden för systemet är rimlig. Överföringsfunktionen för den förenklade modellen ges av ekvation 2.15.

$$G(s) = \frac{239}{s^2 + 12s + 66} \quad (2.15)$$

Polerna för ovanstående system ligger i

$$-6 \pm 5.48i$$

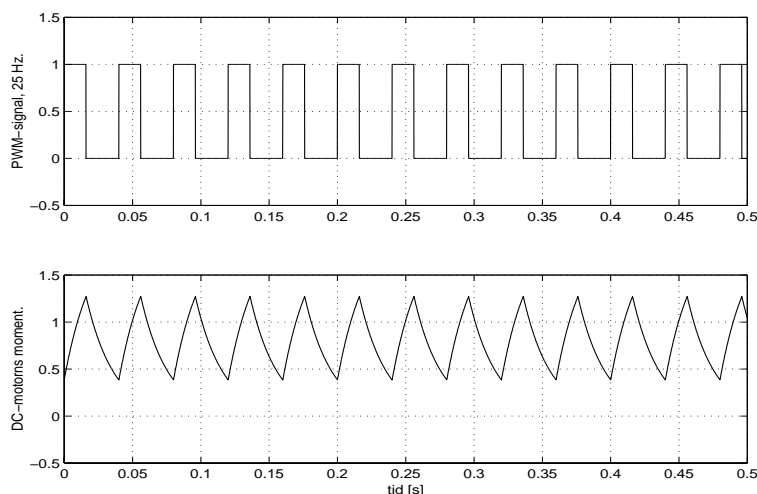
och utifrån dem kan bandbredden beräknas till

$$\omega_0 = \sqrt{66} \text{ rad/s} \Rightarrow f_0 = \frac{\omega_0}{2\pi} = \frac{\sqrt{66}}{2\pi} \text{ Hz} = 1.293 \text{ Hz}$$

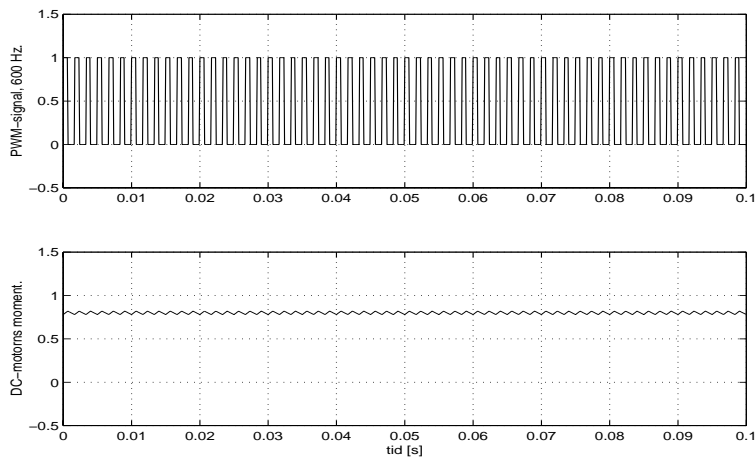
vilket nästan överensstämmer med vad bandbredden för systemet tidigare skattats till. Att skattningen av bandbredden blev lite för liten beror på att fjäderförspänningen hos systemet försumrats, med den blir systemet snabbare. Uppskattningen av systemets bandbredd till 6 Hz bör stämma bra.

## 2.8 Dithering

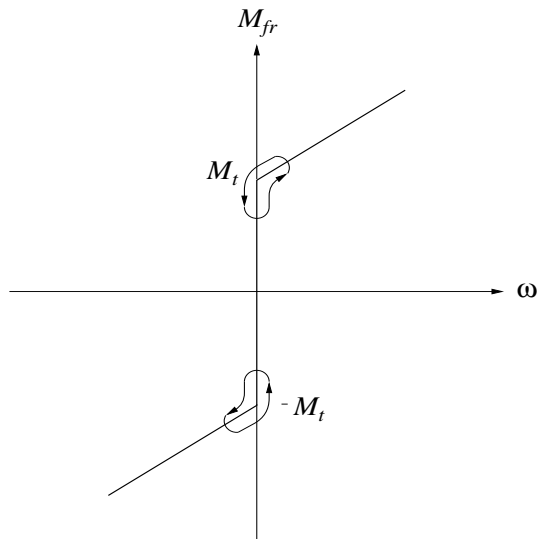
Dithering betyder darrar på engelska. Darrar är just vad trotteln gör fast “bara åt ena hållet”. Kombinationen av att insignalen till DC-motorn är en PWM-signal och att systemets fysikaliska egenskaper är av lågpasskaraktär så kommer, beroende på PWM-signalens frekvensen, pulserna att slå igenom hos det drivande momentet hos systemet. Detta gör att även trottelvinkeln kommer att pulsera. I figur 2.9 och 2.10 visas det drivande momentet hos DC-motorn för två olika frekvenser hos PWM-signalen. (Detta går ej att mäta på det verkliga systemet då det inte finns någon möjlighet att mäta DC-motorns moment, utan istället så implementeras och simuleras detta i SIMULINK. Det är dock principutseendet som eftersöks.) Frekvenserna hos PWM-signalerna är 25 Hz och 600 Hz. För den lägre frekvensen så pulserar momentet kraftigt och för den högre frekvensen är pulserna nästan helt bortfiltrerade. Att trottelvinkeln ökar eller minskar i stationärt läge beror på att det drivande momentet ligger och pulserar precis på gränsen till att trotteln ska fastna. I figur 2.11 så förtydligas detta. Momentet kommer att ligga på antingen den övre eller undre Coulombfriktionsgränsen. För att pulserandet skulle kompensera för den statiska friktionen så skulle pulserandet behöva ske runt origo. Då Coulombfriktionen för systemet är stor så skulle en relativt kraftig pulsering, det vill säga en låg frekvens, behövas, men då skulle trotteln få ett “oroligt” beteende. Pulserandet skulle slå igenom för mycket, vilket inte är acceptabelt.



**Figur 2.9.** Beskrivning av DC-motorns moment då insignalen till DC-motorn är en PWM-signal med 25 Hz. I den övre figuren visas insignalen i form av en pulståg. I den undre figuren så visas det drivande momentet som DC-motorn anbringar på trottelaxeln. Då PWM-signalens frekvens är 25 Hz så dämpas inte pulståget så mycket utan momentet pulserar kraftigt.



**Figur 2.10.** Beskrivning av DC-motorns moment då insignalen till DC-motorn är en PWM-signal med 600 Hz. I den övre figuren visas insignalen i form av ett pulståg. I den undre figuren så visas det drivande moment som DC-motorn anbringar på trottellaxeln. Vid en PWM-frekvens på 600 Hz så har nästan pulserna dämpats helts och det är bara medelvärdet av pulståget som släpps igenom.



**Figur 2.11.** Beskrivning av det drivande momentet hos systemet i stationärt läge. Det drivande momentet ligger och pulserar precis på gränsen till att trotteln skulle fastna. Detta gör att trottelvinkel kommer att antingen öka eller minska sakta beroende på om momentet befinner sig på den övre eller undre gränsen. Trottelvinkeln kommer att öka respektive minska tills pulserandet fångas upp av återföringsfjädrarna.

## 3 Reglering

Regulatordesign för olinjära system är svårt. Teorin är begränsad och i de flesta fall ingår någon form av linjärisering. I detta avsnitt kommer ett försök till linjärisering av systemet, med hjälp av olinjära kompensatorer, att göras. För att kunna kompensera bort olinjäriteterna måste tillstånden vara kända. Alla tillstånd är inte mätbara så en del av dem kommer att rekonstrueras med hjälp av observatörer. Resultatet av linjäriseringen blir en linjär tillståndsmodell med två tillstånd som sedan används för linjär reglerdesign. Reglerdesignen består av en parameterinställning av en PID-regulator med hjälp av Ziegler-Nichols inställningsregler. Kapitlet inleds med en diskussion om svårigheterna med regleringen.

### 3.1 Svårigheter med regleringen

Svårigheterna vid reglering av detta system grundar sig i att systemet är olinjärt, och inte tillåter sig att ställas upp på någon enkel form samt är svårt att modellera. De flesta metoder som finns att tillgå för olinjär regulatordesign förutsätter att systemet går att ställa upp på affin form, vilket för trottellmodellen, ekvation 2.12 och 2.13, inte är fallet. Linjäriseringsmetoder som till exempel *Feedback linearization* och *Sliding mode* [9], förutsätter också att systemet går att skriva på affin form. Tillvägagångssättet för detta system blir istället att försöka kompensera bort olinjäriteterna genom att titta på deras fysikaliska egenskaper och då försöka styra ut dem. Det resulterande systemet kommer då att kunna representeras av ett linjärt system.

### 3.2 Kompensering av olinjäriteter

Olinjäriteterna hos systemet består av friktionen, ekvation 2.5, och fjäderförspänningen, ekvation 2.4. De fysikaliska beskrivningarna av olinjäriteterna gör att det är möjligt att styra ut dem oberoende av varandra, det vill säga att en friktionskompensator och en förspänningskompensator går att konstruera.

#### Friktionskompensator

Friktionen modelleras, som tidigare diskuterats, vid rörelse som en funktion av vinkelhastigheten och vid vila som en funktion av det externa momentet som verkar på systemet, se avsnitt 2.2. Detta gör att för att veta värdet på friktionen så måste både vinkelhastigheten och momentet mätas, vilket inte är möjligt med de mätsignaler som finns tillgängliga. Antag nu att det går att mäta vinkelhastigheten eller åtminstone beräkna den, då är det möjligt att styra insignalen till DC-motorn så att det resulterande drivande momentet kompenserar för Coulombfriktionen (se figur 2.6). Då kommer uppförandet av friktionen uppfattas som om ingen Coulombfriktion existerade. Detta kan göras genom att alltid lägga till eller dra ifrån en styrsignal till DC-motorns insignal, beroende på positiv eller negativ rotation, så att "glappet" upphävs. Storleken på denna styrsignal, som kallas  $u_{fr}$ , kommer att ges av förhållandet mellan Coulombfriktionens storlek,  $c_2$ , och

DC-motorns förstärkning,  $b$ , samt rotationsriktningen,  $sign(\omega)$ . Styrsignalen som kompenserar Coulombfriktionen ges av ekvation 3.1.

$$u_{fr} = \frac{c_2}{b} sign(\omega) \quad (3.1)$$

Med denna kompensation hos styrsignalen kommer nu systemet att uppfattas som om endast den viskösa friktionen fanns, beroende på att så fort trotteln ska börja röra sig så kommer styrsignalen  $u_{fr}$  att styra ut Coulombfriktionen.

### Förspänningskompensator

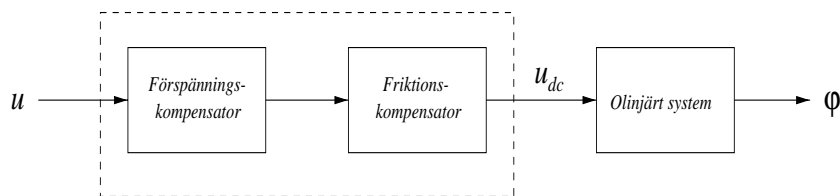
Fjäderförspänningen modelleras som ett konstant externt verkande moment på systemet. Antag att förspänningsmomentet,  $c_1$  i ekvation 2.13, går att mäta eller rekonstruera, då kan förspänningen kompenseras på samma sätt som Coulombfriktionen. Genom att styra insignalen till DC-motorn så att DC-motorns drivande moment ökas med samma moment som motsvarar förspänningen så kommer förspänningen att kompenseras. Storleken på denna styrsignal, som kallas  $u_{fj}$ , kommer att vara konstant och ges av förhållandet mellan DC-motorns förstärkning,  $b$ , och förspänningsmomentet,  $c_1$ . Styrsignalen som kompenserar för förspänningsmomentet ges av ekvation 3.2.

$$u_{fj} = \frac{c_1}{b} \quad (3.2)$$

Systemet kan nu uppfattas som om varken Coulombfriktionen eller förspänningen existerar. Insignalen till DC-motorn, som härnäst benämns  $u_{dc}$ , kommer nu att vara summan av insignalen till det kompenserade systemet, styrsignalen som kompenserar för Coulombfriktionen samt styrsignalen som kompenserar för förspänningen. Insignalen till DC-motorn beskrivs nu i sin helhet av ekvation 3.3.

$$u_{dc} = u + u_{fr} + u_{fj} = u + \frac{c_2}{b} sign(\omega) + \frac{c_1}{b} \quad (3.3)$$

Strukturen för kompensationen beskrivs av blockschemat i figur 3.1.



**Figur 3.1.** Blockschemabeskrivning av den olinjära kompensationen av systemet. Ordningen av kompensatorerna visar sig vara viktig för konstruktionen av observatörerna för vinkelhastigheten och förspänningsmomentet.

För att kunna utföra dessa kompensationer så måste vinkelhastigheten och förspänningen vara kända. Då det inte är möjligt att mäta någon av dem så är ett



alternativ att beräkna dem. Rekonstruktionen sker genom att observera dem ur systemets tillståndsmodell.

### 3.3 Konstruktion av observatör

Kompenseringen av olinjäriteterna hos systemet fungerar under förutsättning att vinkelhastigheten och förspänningens motverkande moment är kända. Eftersom de inte är mätbara så är ett alternativ att rekonstruera dem genom att observera dem ur systemets tillståndsmodell, som ges av ekvation 2.12 och 2.13. Observatören konstrueras genom att återkoppla felet mellan systemets utsignal och modellens utsignal [5]. Detta beskrivs i ekvation 3.4.

$$\dot{\hat{x}} = f(\hat{x}, u) + K(y - \hat{y}) \quad (3.4)$$

Systemets egenskaper, speciellt att det inte går att ställa upp på affin form, gör att det är svårt att bestämma återkopplingsmatrisen,  $K$ , på ett systematiskt sätt. Återkopplingsmatrisen bestämmer rekonstruktionssnabbheten och störningskänsligheten hos observatören. Genom att anta att friktionskompensatorn, som konstruerades i avsnitt 3.2, fungerar perfekt så kan det kompenserade systemet betraktas som ett linjärt system och genom att betrakta förspänningen som ett konstant tillstånd i tillståndsmodellen så fås följande linjära tillståndsmodell.

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\omega} \\ \dot{c}_1 \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 \\ -a_1 & -a_2 & -1 \\ 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \varphi \\ \omega \\ c_1 \end{pmatrix} + \begin{pmatrix} 0 \\ b \\ 0 \end{pmatrix} u \quad (3.5)$$

$$\varphi = \begin{pmatrix} 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} \varphi \\ \omega \\ c_1 \end{pmatrix} \quad (3.6)$$

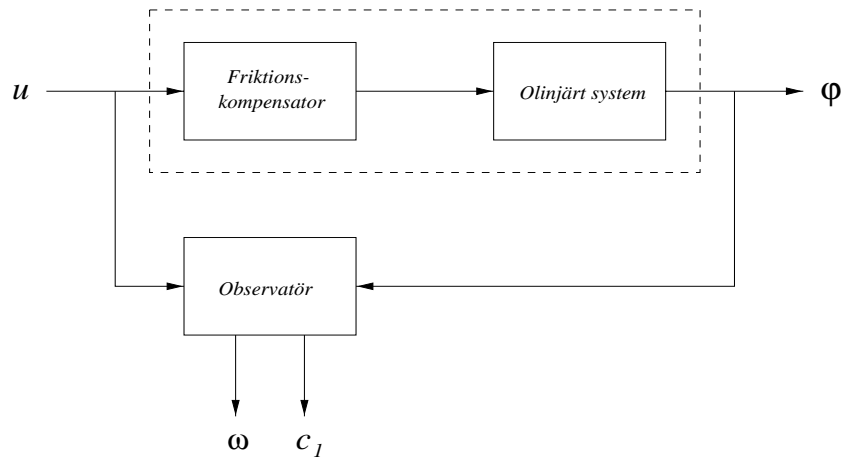
Villkoret för att tillstånden skall vara observerbara ges av att systemets observerbarhetsmatris  $O$  skall ha full radrang [8]. Då systemet är ett SISO-system (Single Input Single Output) så är detta lika med att  $\det(O) \neq 0$ . Observerbarhetsmatrisen för systemet ges av

$$O = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -a_1 & -a_2 & -1 \end{pmatrix} \quad (3.7)$$

med determinanten

$$\det(O) = -1$$

vilket visar att systemet är observerbart, det vill säga att samtliga tillstånd,  $\varphi$ ,  $\omega$  och  $c_1$ , går att rekonstruera. Struktureringen av samspelet mellan kompensatorn och observatören förtydligas av blockschemat i figur 3.2.



**Figur 3.2.** Blockschemarepresentation av friktionskompensatorns samspel med observatören. Då friktionen kompenseras i det olinjära systemet så fås en observerbar tillståndsmodell bestående av tre tillstånd. Här observeras vinkelhastigheten och förspänningsmomentet.

### Vinkelhastighetsobservatör

De tillstånd som observatörerna skall skatta är av helt olika natur. Vinkelhastighetsobservatören skall skatta snabba förlopp, som uppstår vid till exempel trottelsteg, medan förspänningsobservatören skall skatta en konstant. Detta medför att observatörerna bör konstrueras olika gällande rekonstruktionssnabbhet samt inverkan av störningar och modellfel. Vinkelhastighetsobservatörens poler placeras, med hjälp av  $K$ , så att dynamiken hos observatören blir snabbare än systemets. Det vill säga att polerna ska placeras längre in i vänster halvplan. En tumregel säger att cirka en och en halv gånger systemets poler är tillräckligt. Då systemets poler, som ges av egenvärdena till första matrisen i ekvation 3.5, ligger i

$$0 \text{ och } -6 \pm 5.48i$$

så är det lämpligt att placera vinkelhastighetsobservatörens poler på följande ställen.

$$-15 \text{ och } -15 \pm 15i$$

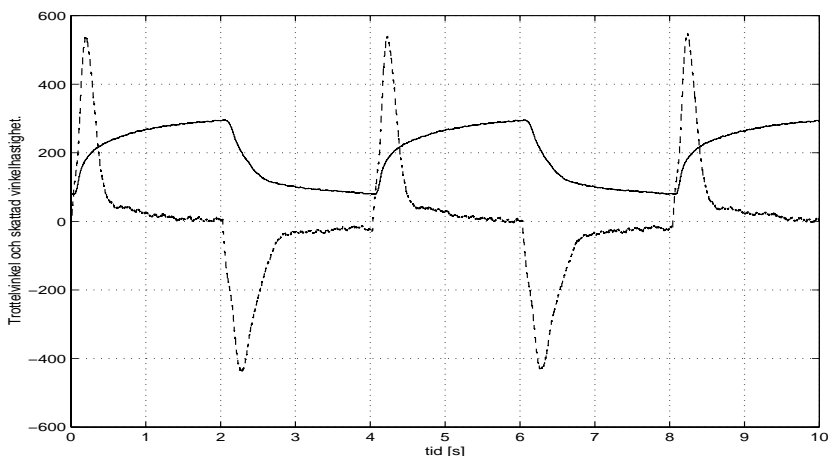
Polplaceringen sker genom att designa återkopplingsvektorn,  $K$ . För vinkelhastighetsobservatören blir  $K$ .

$$K = \begin{pmatrix} 33 \\ 438 \\ -6750 \end{pmatrix}$$

Detta resulterar i en observatör som beskrivs på överföringsform i ekvation 3.8.

$$\hat{\omega} = \frac{239s^2 + 7887s}{s^3 + 45s^2 + 900s + 6750}u + \frac{438s^2 + 4572s}{s^3 + 45s^2 + 900s + 6750}\varphi \quad (3.8)$$

I figur 3.3 så visas vinkelhastighetsobservatören för ett antal trottelsteg. Ur figuren konstateras att observatören rekonstruerar vinkelhastigheten bra, både gällande snabbhet och störningsokänslighet. Vad som är viktigt är att nollgenomgångarna stämmer, då det är tecknet på vinkelhastigheten som används vid friktionskompenseringen.



**Figur 3.3.** Vinkelhastighetsobservatören för ett antal trottelsteg. Den heldragna kurvan är trottelvinkeln och den streckade kurvan är vinkelhastigheten. Observatören rekonstruerar vinkelhastigheten bra både i avseende på snabbhet och störningsokänslighet.

### Förspänningsobservatör

Vid konstruktion av förspänningsobservatören, skattning av en konstant, så bör dynamiken hos observatören vara betydligt långsammare än systemets. Då reduceras inverkan av störningar och modellfel kraftigt. Tillståndsmodellen, som ges av ekvation 3.5 och 3.6, beskriver, som tidigare diskuterats, systemet bra för transienta förlopp men inte under stationära. En ny parameteruppsättning tas fram som beskriver systemet bra under stationära förhållanden. Detta för att möjliggöra konstruktionen av förspänningsobservatören. Parameteruppsättningen tas fram på systemet som ges av systemet för observatörkonstruktionen, ekvation 3.5 och 3.6. Parametrarna blir nu.

$$\begin{aligned} a_1 &= 29 \\ a_2 &= 10 \\ b &= 300 \end{aligned}$$

Förspänningsobservatören skall vara betydligt långsammare än systemets dynamik och ett bra val av observatörspoler blir då följande poler.

$$-0.5 \text{ och } -0.5 \pm 0.5i$$

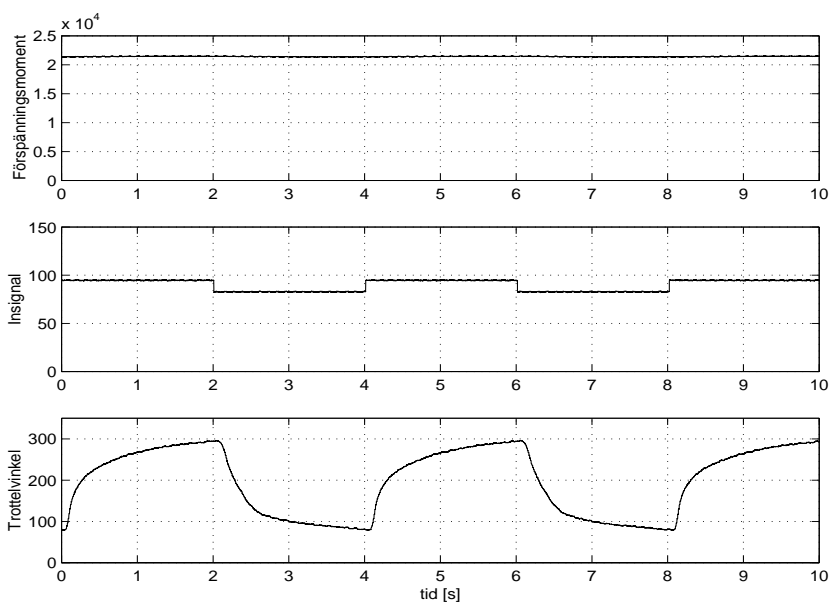
Även här så uppfylls polvalet genom att designa återkopplingsvektorn  $K$ . För förspänningsobservatören fås följande  $K$ -vektor.

$$K = \begin{pmatrix} -8.5 \\ 57 \\ -0.25 \end{pmatrix}$$

Detta resulterar i en förspänningsobservatör som beskrivs på överföringsform av ekvation 3.9.

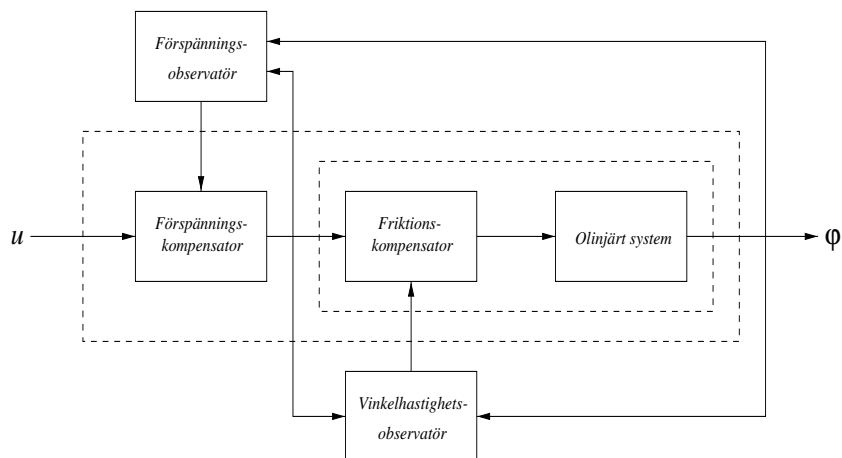
$$\hat{c}_I = \frac{75}{s^3 + 1.5s^2 + s + 0.25}u - \frac{0.25s^2 + 2.5s + 7.25}{s^3 + 1.5s^2 + s + 0.25}\varphi \quad (3.9)$$

I figur 3.4 så visas resultatet av skattningen av förspänningsmomentet. Även vid trottelsteg så ligger skattningen konstant. Värdet på det skattade förspänningsmomentet används vid förspänningskompensationen.



**Figur 3.4.** Skattning av förspänningsmomentet med hjälp av en observatör. I den översta figuren visas det skattade förspänningsmomentet, i den mitten visas insignalen till DC-motorn och i den understa visas trottelvinkeln. Dessa signaler utgör insignaler till observatören. Skattningen uppför sig bra, det vill säga värdet är konstant, även vid trottelsteg.

Strukturen av kompensatorerna och observatörerna beskrivs med ett blockschema i figur 3.5.



**Figur 3.5.** Blockschemarepresentation för de olinjära compensationerna, observatörerna och det olinjära systemet.

Resultatet av friktionskompensationen och förspänningskompensationen blir nu ett system som kan beskrivas av en linjär tillståndsmodell bestående två tillstånd.

$$\begin{pmatrix} \dot{\varphi} \\ \dot{\omega} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_2 \end{pmatrix} \begin{pmatrix} \varphi \\ \omega \end{pmatrix} + \begin{pmatrix} 0 \\ b \end{pmatrix} u \quad (3.10)$$

$$\varphi = \begin{pmatrix} 1 & 0 \end{pmatrix} \begin{pmatrix} \varphi \\ \omega \end{pmatrix} \quad (3.11)$$

### Styrbarhet

För att det ska vara möjligt att reglera ovanstående system så måste det vara styrbart [5]. Villkoret för detta är att styrbarhetsmatrisen  $S$  för systemet skall ha full kolonnrang [8]. För SISO-system så är detta lika med att  $\det(S) \neq 0$ . Styrbarhetsmatrisen för det linjära systemet är

$$S = \begin{pmatrix} 0 & b \\ b & -a_2 b \end{pmatrix} \quad (3.12)$$

med determinanten

$$\det(S) = -b^2 \neq 0 \text{ för } b \neq 0$$

vilket visar att systemet är styrbart och linjär regulatordesign går att utföra.

### 3.4 PID-regulator

De vanligaste regulatorstrukturerna ute i industrin är PID-regulatorer [5]. Styrsignalen till systemet är baserad på felet mellan önskad och verklig signal. Detta genom en proportionell ( $P$ ), integrerande ( $I$ ) och en deriverande term ( $D$ ), enligt ekvation 3.13,

$$u(t) = K \left( e(t) + \frac{1}{T_I} \int_{t_0}^t e(\tau) d\tau + T_D \frac{d}{dt} e(t) \right) \quad (3.13)$$

där parametrarna  $K$ ,  $T_I$  och  $T_D$  bestämmer regulatorns egenskaper. För att undersöka om regleringen av systemet förbättras med inkopplingen av de olinjära kompensatorerna så designas en PID-regulator enligt *Ziegler-Nichols* metod [5], som innebär följande. Parametrarna  $T_I$  och  $T_D$  sätts till  $\infty$  respektive  $0$ . Därefter ökas  $K$  tills systemet fås att självsvänga med konstant amplitud. Detta  $K$ -värde,  $K_0$ , noteras och även periodtiden hos självsvängningen,  $T_0$ . Sedan beräknas  $K$ ,  $T_I$  och  $T_D$  enligt följande tabell.

Regulator	$K$	$T_I$	$T_D$
P	$0.5K_0$		
PI	$0.45K_0$	$T_0/1.2$	
PID	$0.6K_0$	$T_0/2$	$T_0/8$

En nackdel med regulatorn i ekvation 3.13 är att det inte går att få en ren integratorterm. Detta åtgärdas genom en omparametrisering av regulatorn. Här studeras enbart PI-regulatorn då det är en sådan regulator som kommer att designas. PI-regulatorn ges nu av ekvation 3.14.

$$u(t) = Ke(t) + \frac{1}{T_I} \int_{t_0}^t e(\tau) d\tau \quad (3.14)$$

Parametrarna  $K$  och  $T_I$  beräknas nu enligt följande tabell.

Regulator	$K$	$T_I$
PI	$0.45K_0$	$T_0/0.54K_0$

Med dessa inställningsregler så fås följande värden på  $K$  och  $T_I$ .

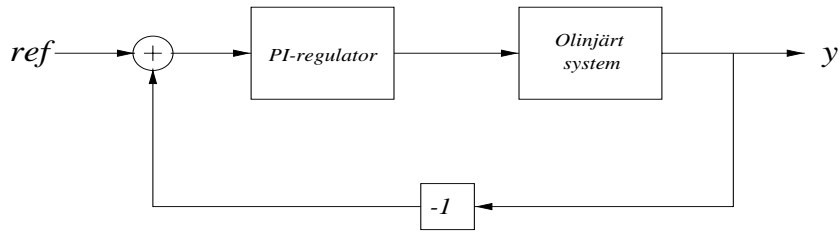
$$K = 0.27 \text{ och } T_I = 0.58$$

### 3.5 Implementering av trottregulator

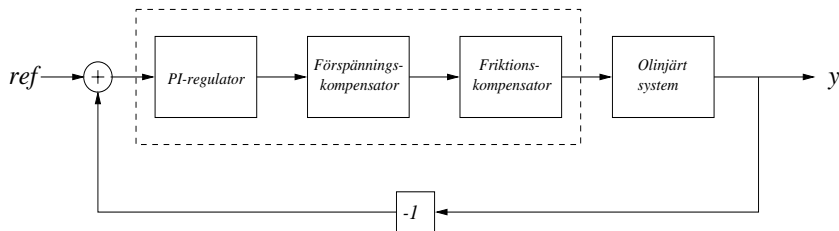
Regulatorn implementeras i den hårdvaran som finns tillgänglig i laboratoriet. Alla processer programmeras i C och beräkningarna utförs med en heltalsprocessor, vilket medför att alla signaler representeras med heltal. Detta gör att insignalen till DC-motorn kan ses som små enhetsteg. För att behålla noggrannheten i beräkningarna så skalas variablerna, och för att systematisera avrundningen så införs *floor*, som alltid avrundar nedåt. Observatörerna implementeras på balanserad tillståndsform, där koefficienterna är uppskalade på grund av heltalsoperationerna. Detta för att öka den numeriska precisionen. Programkoden för regulatorn visas i bilaga B.

### 3.6 Verifiering av trotteltregulator

Den olinjära regulator, PI-regulatorn samt de olinjära kompensatorerna, kommer att jämföras med den linjära PI-regulatorn från kapitel 1, som är designad för det olinjära systemet. Den linjära regulatorn beskrivs av blockschemat i figur 3.6 och den olinjära regulatorn ges av blockschemat i figur 3.7.

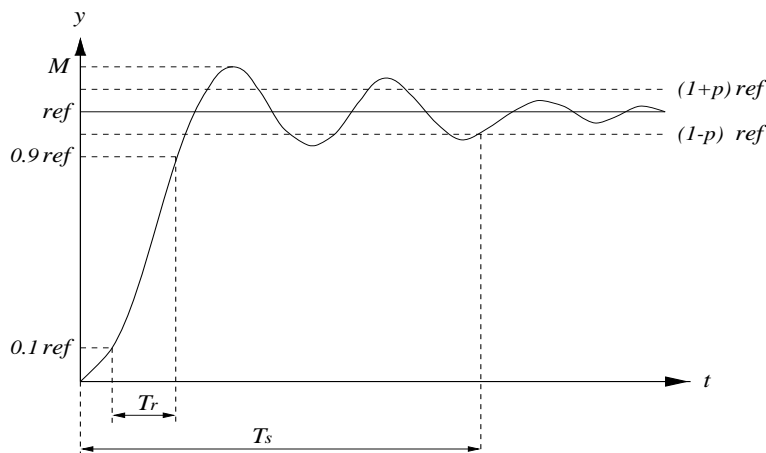


**Figur 3.6.** Linjär regulatorstruktur. Regulatorn är en PI-regulator som är inställd enligt Ziegler-Nichols metod.



**Figur 3.7.** Olinjär regulatorstruktur. Regulatorn består av en PI-regulator, inställd enligt Ziegler-Nichols metod, en friktionskompensator och en förspänningskompensator.

Det som jämförs är stigtid,  $T_r$ , insvängningstid,  $T_s$ , samt översläng,  $M$ , för ett trotteltsteg, se figur 3.8. Stigtiden,  $T_r$ , anger hur lång tid det tar för utsignalen att gå från 0.1 till 0.9 gånger slutvärdet (referensvärdet). Insvängningstiden,  $T_s$ , anger hur lång tid det tar för systemet att svänga in till det nya jämviktsläget. Insvängningstiden definieras som det minsta  $t$ -värdet sådant att  $(1 - p)ref \leq y \leq (1 + p)ref$  för alla  $t$  större än  $T_s$ . Värdet på  $p$  sätts här till 5%. Överslängen,  $M$ , anger hur mycket utsignalen tenderar att skjuta över referensvärdet. Överslängen anges i %. Trotteltstegen utförs både som öppnande och som stängande och för tre olika varvtal, dessa jämnt utspridda på motorns arbetsområde. Första varvtalet ligger på 1200 varv/min, det andra på 2400 varv/min och det tredje på 3600 varv/min. Detta beskrivs av figurerna 3.9, 3.10, 3.11, 3.12, 3.13 och 3.14. Ur dessa figurer utläses och sammanställs prestanda för de olika regulatorerna. I följande tabell beskrivs dessa prestanda och tabellen är sammanställd för öppnande trotteltsteg.



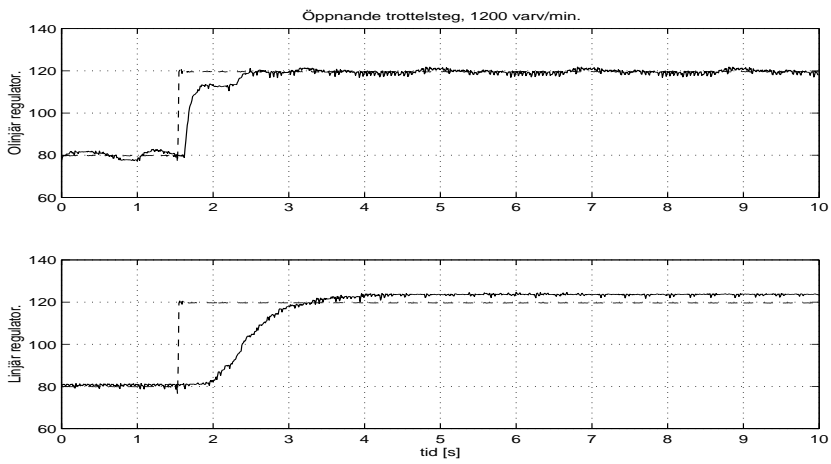
**Figur 3.8.** Stegsvartsspecifikationer. Här visas stigtiden,  $T_r$ , insvängningstiden,  $T_s$ , och överslängen,  $M$ , för systemet. Dessa tre värden är ett sätt att ange hur bra regulatorns prestanda är.

Varvtal	Regulator	$T_r$	$T_s$	$M$
1200	Olinjär regulator	0.21s	1.95s	4%
	Linjär regulator	0.86s	2.40s	5%
2400	Olinjär regulator	0.39s	0.90s	4%
	Linjär regulator	0.48s	0.95s	3%
3600	Olinjär regulator	0.53s	0.99s	3%
	Linjär regulator	0.67s	1.04s	2%

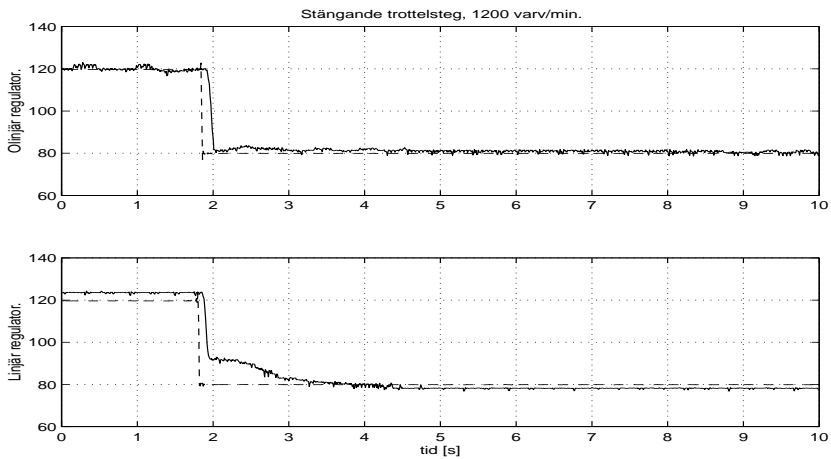
Ur ovanstående tabell fås att snabbheten hos regleringen ökat avsevärt, det vill säga att stigtiden minskat. Dessvärre så betar den sig lite svängigare, på grund av att Coulombfriktionen hela tiden kompenseras. Den del av styrsignalen som utgörs av kompensationen är stor i förhållande till den del som verkligen förflyttar trotteln. Det finns en konflikt i regleringen med avseende på snabbhet och svängighet. Det kan tänkas att systemet skulle uppföra sig lugnare om vid små fel och låg vinkelhastighet så låses den olinjära kompensationen och endast PI-regulatorn sköter regleringen. Ett annat angreppssätt är att parameterstyra PI-regulatorn på samma sätt som den linjära parameterstyrda regulatorn. Vid större trottelseg är förbättringen inte lika stor, men oftast så är det endast små steg som skall regleras. Hela tiden uppstår det små tryckvariationer i insugningsröret hos bensenmotorn och då är det nödvändigt med en snabb inre trottelloop, där det sker en reglering av små steg, och en långsammare yttre loop.

Slutsatsen är att de olinjära kompensatorerna bidrar till att snabbheten hos regleringen ökar. Detta har undersökts med en enkel PI-regulator och resultatet blev gott.

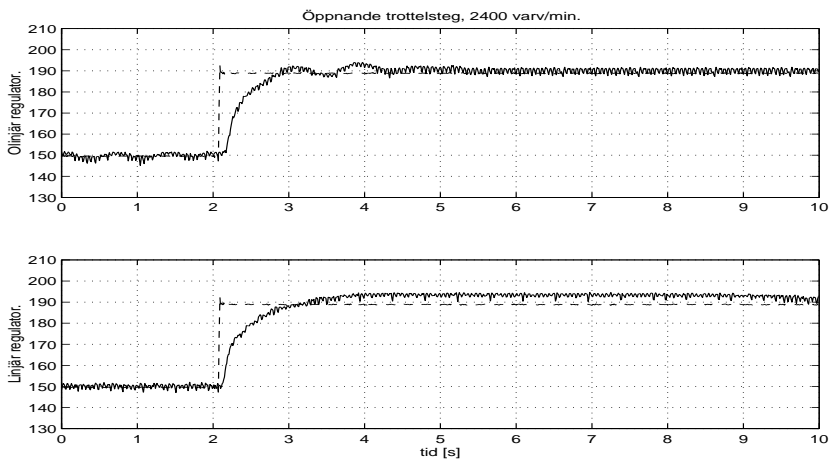




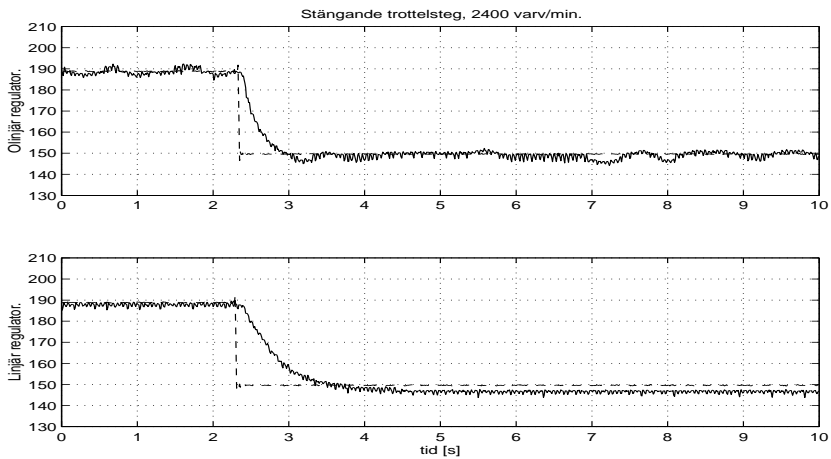
**Figur 3.9.** Öppnande trottelseg för den linjära och den olinjära regulatorn då varvtalet hos bensinmotorn är 1200 varv/min. I den övre figuren visas referenssignal (streckad) och trottelvinkel (heldragen) för den olinjära regulatorn, och i den nedre figuren visas samma signaler för den linjära regulatorn.



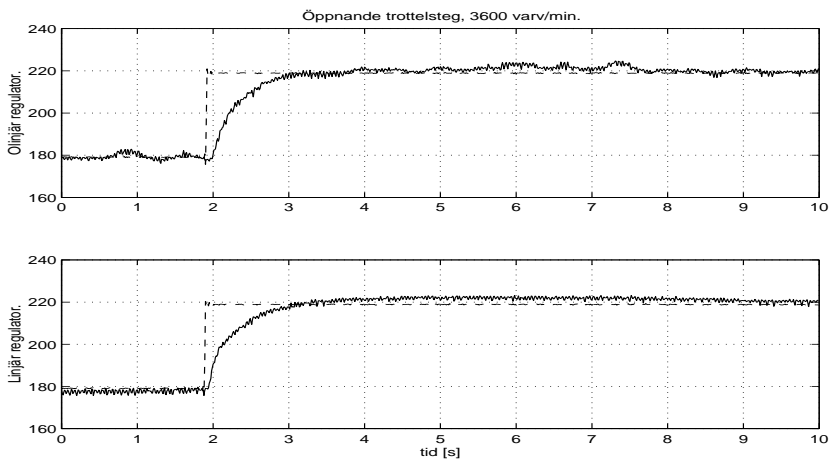
**Figur 3.10.** Stängande trottelseg för den linjära och den olinjära regulatorn då varvtalet hos bensinmotorn är 1200 varv/min. I den övre figuren visas referenssignal (streckad) och trottelvinkel (heldragen) för den olinjära regulatorn, och i den nedre figuren visas samma signaler för den linjära regulatorn.



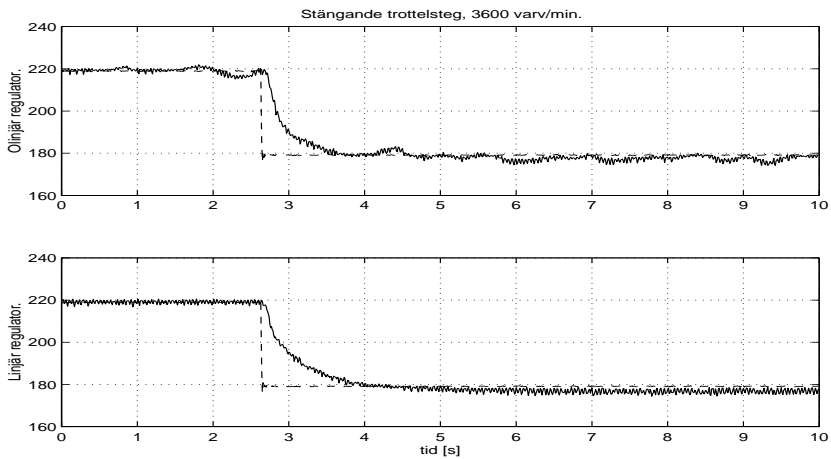
**Figur 3.11.** Öppnande trottelseg för den linjära och den olinjära regulatorn då varvtalet hos bensinmotorn är 2400 varv/min. I den övre figuren visas referenssignal (streckad) och trottelvinkel (heldragen) för den olinjära regulatorn, och i den nedre figuren visas samma signaler för den linjära regulatorn.



**Figur 3.12.** Stängande trottelseg för den linjära och den olinjära regulatorn då varvtalet hos bensinmotorn är 2400 varv/min. I den övre figuren visas referenssignal (streckad) och trottelvinkel (heldragen) för den olinjära regulatorn, och i den nedre figuren visas samma signaler för den linjära regulatorn.



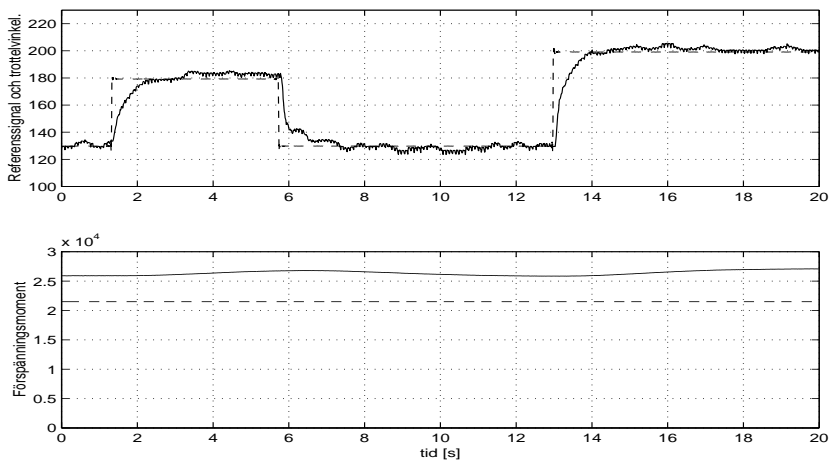
**Figur 3.13.** Öppnande trottelseg för den linjära och den olinjära regulatorn då varvtalet hos bensinmotorn är 3600 varv/min. I den övre figuren visas referenssignal (streckad) och trottelvinkel (heldragen) för den olinjära regulatorn, och i den nedre figuren visas samma signaler för den linjära regulatorn.



**Figur 3.14.** Stängande trottelseg för den linjära och den olinjära regulatorn då varvtalet hos bensinmotorn är 3600 varv/min. I den övre figuren visas referenssignal (streckad) och trottelvinkel (heldragen) för den olinjära regulatorn, och i den nedre figuren visas samma signaler för den linjära regulatorn.

### 3.7 Analys av störningar vid trotteltreglering

En regulatorverifiering blir inte komplett utan att analysera inverkan av störningar vid trotteltregleringen. Då modelleringen av systemet inte behandlar luftströmmen genom trotteln, som borde modelleras som ett motverkande moment, så undersöks hur regleringen tar hand om denna "störning". Genom att titta på skattningen av förspänningsmomentet så fås en uppfattning av vad som händer. Detta åskådliggörs i figur 3.15.



**Figur 3.15.** Analys av störningar vid regleringen. I den övre figuren visas referenssignal (streckad) och trottelvinkel (heldragen) under ett antal trotteltsteg. Bensinmotorns varvtal ligger konstant på 2400 varv/min. I den nedre figuren visas det skattade förspänningsmomentet (heldragen kurva). Den streckade kurvan motsvarar en skattning av förspänningen utan någon luftgenomströmning hos trotteln.

I ovanstående figur ses att förspänningsskattningen hjälper till att fånga upp störningarna från luftströmmen. Där ses att luftströmmen är ett motverkande moment hos systemet eftersom förspänningsskattningen ökar. (Den streckade kurvan motsvarar förspänningsskattningen utan luftström.) Även vid trotteltstegen så ökar skattningen och det beror på att luftströmmen ökar med trottelvinkeln och därmed så ökar även det motverkande momentet. Resultatet blir att regleringen tar bra hand om denna störning.

## 4 Utvidgningar

Det finns många intressanta utvidgningar till arbetet. En är att förbättra trottelmodellen, exempelvis genom att även modellera det motmoment som uppstår av luftströmmen genom trotteln. Vid tillgång till mätdata så bör det enkelt kunna modelleras med hjälp av black-boxidentifiering, där även en brusmodell kan inkluderas. Svårigheten med att förbättra modellen är att tillgången av mätbara signaler är begränsad. Detta förhindrar identifieringen av mer komplexa modeller. Inköp eller konstruktion av nya sensorer eller givare kan hjälpa till vid identifieringen av modellen.

En fortsättning på arbetet finns också hos regulatordesignen, bland annat bör fler reglerstrategier prövas och jämföras. Med en kompletterad brusmodellen så kan exempelvis *LQG*-design (Linear Quadratic Gaussian) tillämpas, där konstruktionen av observatörer kan göras med Kalman-filter. En annan intressant reglerstrategi är tidsoptimal reglering. Idén är att ta fram en styrlag (bang-bang styrning) som på minsta möjliga tid styr trottelvinkeln till önskat läge. För att styrlagen enkelt skall kunna realiserats måste delsystemet som består utav fjädern kompenseras, detta för att systemets poler skall bli reella. Tidsoptimal styrning är speciellt intressant i och med att styrsignalen antingen är maximalt positiv eller maximalt negativ. I den olinjära regulator som förnärvarande är implementerad så används inga negativa styrsignaler, då felen inte blir så stora, vilket gör att med tidsoptimal styrning så bör prestanda hos regulatorn öka avsevärt för stängande steg. Om tidsoptimal styrning ska vara realiserbart så måste modellsäkerheten vara stor, vilket medför att modellbeskrivningen av systemet bör förbättras.

Implementeringen av regulatorn sker förnärvarande i en heltalsprocessor. Detta medför att alla beräkningar görs med heltal. Även styrsignalen ställs ut med heltal, vilket innebär att styrsignalen kan ses som små enhetssteg. Med en flyttalsprocessor så bör uppförandet av systemet bli "mjukare". Då det inte är någon fråga om massproduktion så kan uppdatering av hårdvaran i laboratoriet vara av intresse, även om styrsystem är dyra.

Intressant kunde också vara att undersöka vilka olika elektriska motorer som finns. Då det i laboratoriet inte finns några begränsningar så kan en starkare och snabbare elmotor vara förtjänstfullt vid förbättring av trotteltreglering.

## 5 Slutsatser

En matematisk modell för trotteln tas fram för regulatordesign. Modellen är en tillståndsmo-  
dell och innehåller två tillstånd samt två olinjäriteter. Modellen förenklas  
för att möjliggöra identifiering, vilket resulterar i en modell med fem paramet-  
rar. Dessa parametrar identifieras genom att anpassa modellen till mätningar som  
utförts på systemet, mätningar som exciterar all dess dynamik. För att möjliggöra  
identifiering och validering av modellen så implementeras modellen i SIMULINK.  
Trottelmodellen valideras genom att jämföra den med "färska data".

Identifiering av modellen är svårt, detta på grund av att de mätbara signalerna  
är få och att de olinjäriteter som finns i systemet är komplicerade. Valideringen  
av modellen ger att modellen väl beskriver dynamiken hos systemet vid transienta  
förlopp, men vid stationära förlopp är den sämre, vilket beror på *dithering* som  
beskrivs mer ingående i rapporten. Detta gör att två parameteruppsättningar måste  
tas fram för systemet, en uppsättning för transienta förlopp och en för stationära.

Regulatordesignen för systemet består av en linjärisering av systemet, i form av  
två olinjära kompensatorer, samt en PI-regulator. Den ena kompensatorn kompen-  
serar för Coulombfriktionen och den andra kompenserar för fjäderförspänningen.  
För att konstruera kompensatorerna så antas att samtliga tillstånd i modellen samt  
fjäderförspänningen är kända. Genom friktionskompensationen så kan systemet be-  
traktas som ett linjärt system med tre tillstånd, där förspänningen lagts till som  
ett tredje konstant tillstånd. Då tillstånden inte är mätbara så rekonstrueras de  
genom att de observeras ur den linjära tillståndsmo-  
dellen.

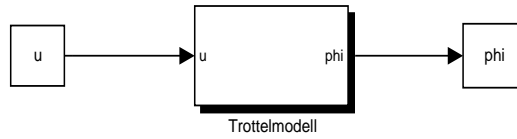
Linjäriseringen av systemet fungerar tillräckligt bra för att linjär regulatorde-  
sign skall gå att utföra. Detta verifieras genom att undersöka de rekonstruerade till-  
stånden. Den linjära regulatordesignen består av en PI-regulator, där parametrarna  
ställts in enligt *Ziegler-Nichols* metod. Som verifiering av den olinjära regulatorn så  
görs ett antal trottelseg, jämnt spridda över motorns arbetsområde. Trottelsegen  
jämförs sedan med den gamla parameterstyrda PI-regulatorn. Resultatet visar att  
regleringens snabbhet har ökat avsevärt med den olinjära regulatorn, detta både för  
öppnande som för stängande trottelseg. Förbättringen blev störst för stängande  
steg.

## Referenser

- [1] T. Glad. Nonlinear control theory. Technical Report LiTH-ISY-R-1686, Institutionen för Systemteknik, Linköpings Universitet, Linköping, Sverige, 1994.
- [2] The MathWorks, Inc., Natick, MA. USA. *MATLAB: High-Performance Numeric Computation and Visualization Software*, 1992.
- [3] The MathWorks, Inc., Natick, MA. USA. *SIMULINK: Dynamic System Simulation for MATLAB*, December 1996.
- [4] H-M. Streib och H. Bischof. Electronic throttle control (etc): A cost effective system for improved emissions, fuel economy, and driveability. *SAE paper (960338)*, 1996.
- [5] T. Glad och L. Ljung. *Reglerteknik. Grundläggande teori*. Studentlitteratur, Lund, Sverige, 1989.
- [6] G. Olsson och M. Alaküla. *Elmaskinsystem*. Institutionen för Industriell Elektronik och Automation, Lunds Tekniska Högskola, Lund, Sverige, 1997.
- [7] L. Ljung och T. Glad. *Modellbygge och simulering*. Studentlitteratur, Lund, Sverige, 1991.
- [8] L. Ljung och T. Glad. *Kompendium i Reglerteori*. Institutionen för Systemteknik, Tekniska Högskolan i Linköping, Linköping, Sverige, 1996.
- [9] J-J E. Slotine och W. Li. *Applied Nonlinear Control*. Prentice-Hall, Englewood Cliffs, N.J. USA, 1991.
- [10] H. Olsson. *Control Systems with Friction*. Phd thesis ISRN LUTFD2/TFRT-1045-SE, Institutionen för Reglerteknik, Lunds Tekniska Högskola, Lund, Sverige, April 1996.

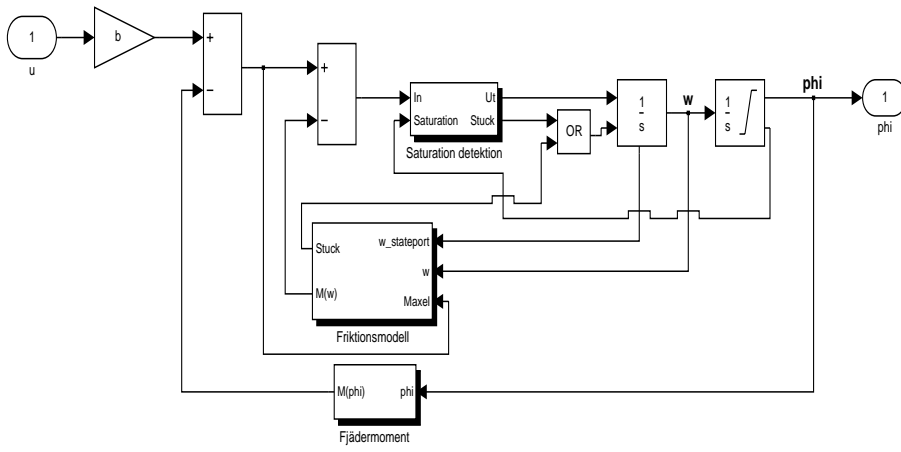
## Bilaga A: Implementering av trottellmodell

Trottellmodellen implementeras i SIMULINK. Modellen är hierarkiskt uppbyggd, först med större block som blir mer och mer detaljerade. Största djupet är fyra block. Nedan visas nu hur modellen byggts upp.

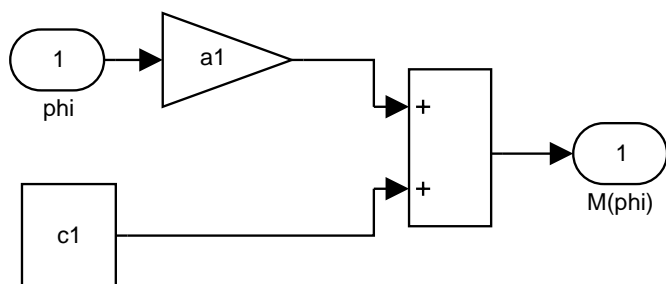


**Figur A.1.** Förenklad trottellmodell i SIMULINK.

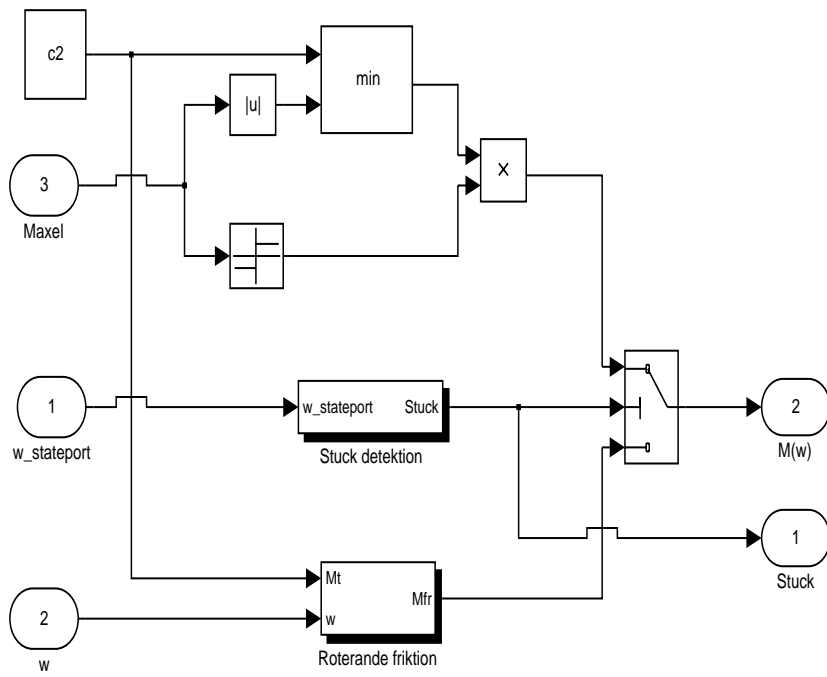




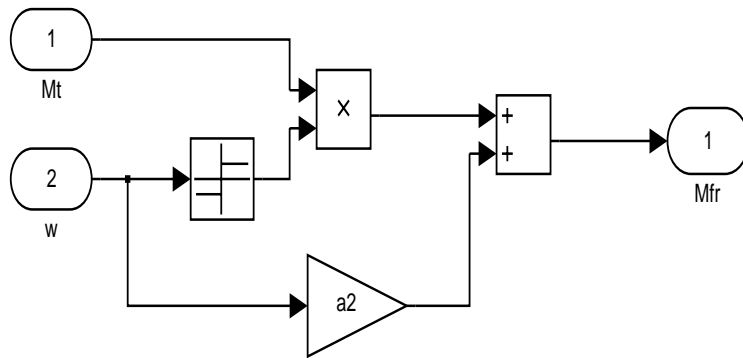
Figur A.2. Trottelmodell.



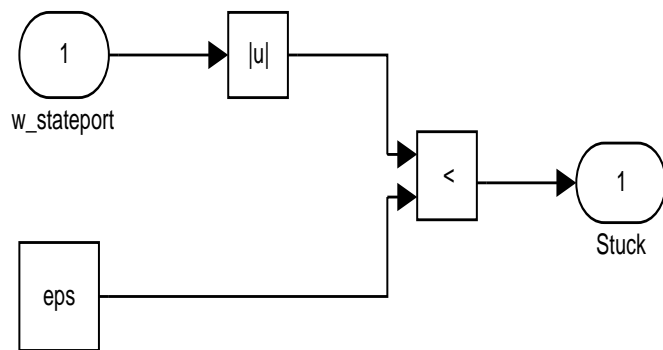
**Figur A.3.** Fjädermoment med förspänningsmoment  $c_1$ .



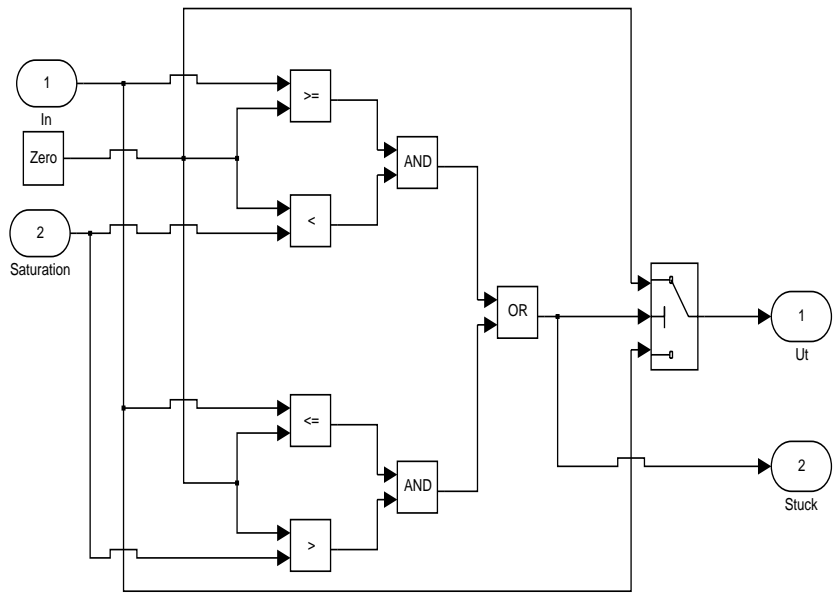
Figur A.4. Stick-slip friktionsmodell med Coulombfriktionen  $c_2$ .



**Figur A.5.** Roterande friktion, innehållande viskös friktion samt Coulombfriktion.



**Figur A.6.** Stuck detektion, returnerar en etta om trottskivan "fastnar".



**Figur A.7.** Saturation detektion, ser till så att trottelvinkeln håller sig inom tillåtna gränser. Returnerar en etta om vinkeln når övre eller undre begränsningen.

## Bilaga B: Programkod för trottelsregulatorn

Nedan redovisas programkoden för implementering av trottelsregulatorn. Den börjar med en initiering av gränserna för trottelsvinkeln. Därefter kommer observatörerna, kompensatorerna och till sist PI-regulatorn.

```
#include "datatype.h"
#include "vardec.h"
#include "throttle.h"
#include "bits.h"
#include "macros.h"

void AdaptThrottle(void)
{
    switch(throttle_mode)
    {
        case (not_adapted) :
            THROTTLE_SIGN(1);
            THROTTLE_SIGN(0);
            D01_DUTY(THROTTLE_PWM_PERIOD_D01*90/100);
            throttle_mode = read_min;

            /* Set throttlePosMin and throttlePosMax
               to a value in the */
            /* range [0,1023] and not near the possible
               max and min values */
            throttlePosMin = 511L;
            throttlePosMax = 512L;
            break;

        case (read_min) :
            if(throttlePosFilt < throttlePosMin)
                /* Store new min position */
                throttlePosMin = throttlePosFilt;
            else
                throttle_mode = open_throttle;
            /* To make sure that the throttle is
               able to reach WOT */
            throttlePosMax = throttlePosMin - 1;
            break;

        case (open_throttle) :
            THROTTLE_SIGN(1);
            D01_DUTY(THROTTLE_PWM_PERIOD_D01*90/100);
            throttle_mode = read_max;
            break;

        case (read_max) :
            if(throttlePosFilt > throttlePosMax)
                /* Store new max position */
                throttlePosMax = throttlePosFilt;
            else
                throttle_mode = adapted;
            break;

        default : break;
    }
}
```

```

    }
}

void AngularVelObserver(void)
{
    static s32 Fa[3][3] = { { 7570 , -2211 , -1566 } ,
                            { 1444 , 9688 , -498 } ,
                            { -149 , -327 , 8369 }
                          };
    static s32 Ga[3][2] = { { -12798 , -18155 } ,
                            { 1206 , -6400 } ,
                            { 6216 , -6825 }
                          };
    static s32 Ha[1][3] = { { -23078 , -2503 , -7208 } ,
                            };
    static s32 xa[3] = { { 1 } ,
                        { 1 } ,
                        { 1 }
                      };
    s32 y, u[2], tmpX[3];
    u8 i = 0;

    /* Create input-signal: u1 = control-signal,
       u2 = throttleangle. */

    u[0] = (s32)controlSignalAngularVel;
    u[1] = (s32)throttlePosNorm;

    /* Calculate output. */
    /* y(t) = H*x(t) */

    y = (Ha[0][0]*xa[0]+Ha[0][1]*xa[1]+Ha[0][2]*xa[2])/10000L;

    /* Update states. */
    /* x(t+1) = F*x(t)+G*u(t) */

    tmpX[0] = Fa[0][0]*xa[0]+Fa[0][1]*xa[1]+Fa[0][2]*xa[2]
              +Ga[0][0]*u[0]+Ga[0][1]*u[1];
    tmpX[1] = Fa[1][0]*xa[0]+Fa[1][1]*xa[1]+Fa[1][2]*xa[2]
              +Ga[1][0]*u[0]+Ga[1][1]*u[1];
    tmpX[2] = Fa[2][0]*xa[0]+Fa[2][1]*xa[1]+Fa[2][2]*xa[2]
              +Ga[2][0]*u[0]+Ga[2][1]*u[1];

    /* Ensure that the integer rounds towards floor. */

    for(i = 0; i < 3; i++)
    {
        if(tmpX[i] < 0)
        {
            xa[i] = tmpX[i]/10000L - 1;
        }
        else
        {
            xa[i] = tmpX[i]/10000L;
        }
    }
}

```



```

/* Calculate the Angularvelocity for the throttle. */

angularVel=(s32)y;
}

void PreTensionObserver(void)
{
    static s32 Fp[3][3] = { { 9990 , 37 , -13 } ,
                          { -37 , 9955 , 56 } ,
                          { -13 , -56 , 9905 }
                        };
    static s32 Gp[3][2] = { { 6586 , -691 } ,
                          { 7810 , -622 } ,
                          { 4193 , -134 }
                        };
    static s32 Hp[1][3] = { { 6622 , -7835 , 4195 } ,
                          };
    static s32 xp[3] = { { 1 } ,
                       { 1 } ,
                       { 1 }
                     };

    s32 y, u[2], tmpX[3];
    u8 i = 0;

    /* Create input-signal: u1 = control-signal,
       u2 = throttleangle. */

    u[0] = (s32)controlSignalPreTension;
    u[1] = (s32)throttlePosNorm;

    /* Calculate output. */
    /* y(t) = H*x(t) */

    y = (Hp[0][0]*xp[0]+Hp[0][1]*xp[1]+Hp[0][2]*xp[2])/10000L;

    /* Update states. */
    /* x(t+1) = F*x(t)+G*u(t) */

    tmpX[0] = Fp[0][0]*xp[0]+Fp[0][1]*xp[1]+Fp[0][2]*xp[2]
            +Gp[0][0]*u[0]+Gp[0][1]*u[1];
    tmpX[1] = Fp[1][0]*xp[0]+Fp[1][1]*xp[1]+Fp[1][2]*xp[2]
            +Gp[1][0]*u[0]+Gp[1][1]*u[1];
    tmpX[2] = Fp[2][0]*xp[0]+Fp[2][1]*xp[1]+Fp[2][2]*xp[2]
            +Gp[2][0]*u[0]+Gp[2][1]*u[1];

    /* Ensure that the integer rounds towards floor. */

    for(i = 0; i < 3; i++)
    {
        if(tmpX[i] < 0)
        {
            xp[i] = tmpX[i]/10000L - 1;
        }
        else
        {

```

```

    xp[i] = tmpX[i]/10000L;
}
}

/* Calculate the Pretension of the spring. */

preTension=(s32)y;
}

void CoulombComp(void)
{
    s32 sgn = 0, mt = 18;

    if (angularVel >= velMax)
    {
        sgn = velMax;
    }
    else if (angularVel <= (-velMax))
    {
        sgn = -velMax;
    }
    else
    {
        sgn = angularVel;
    }

    mt = (s32)coulombComp;
    uCoulombComp = (s32)( sgn*mt/velMax );
}

void PreTensionComp(void)
{
    s32 b = 300; /* Estimated amplification. */

    uPreTensionComp = (s32)(preTension)/b;
}

void ThrottlePID(void)
{
    s32 e, u, v, y, ref;
    static s32 I = 0L;
    static s32 eOld = 0L;
    s16 tmp = 0;

    /* sampling frequency is 100 Hz, must be changed if
       the priority is changed */

    u16 fs = 100;

    /* Check if throttlePosSet exceeds 1000 or is below 0 */

    if (throttlePosSet < 0)
        ref = 0L;
    else if (throttlePosSet > 1000)
        ref = 1000L;
}

```

```

else
    ref = (s32)(throttlePosSet);

/* Check if throttlePosFilt exceeds throttlePosMax or
   is below throttlePosMin */

if (throttlePosFilt < throttlePosMin)
    y = (s32)(throttlePosMin);
else if (throttlePosFilt > throttlePosMax)
    y = (s32)(throttlePosMax);
else
    y = (s32)(throttlePosFilt);

/* Only used for showing normalized throttle
   position in ADT */

throttlePosNorm = (u16)((y - (s32)(throttlePosMin))
    *1000L/((s32)(throttlePosMax)
    - (s32)(throttlePosMin)));

/* Choose right parameters */
/* if no parameter tuning, use parameters below */

if (throttleTuning != 1)
{
    throttleK = 27;
    throttleKi = 58;
}

/* Compute Observers and Compensators. */

AngularVelObserver();
PreTensionObserver();
CoulombComp();
PreTensionComp();

/* Create e=ref-measured */
/* Convert y from filtered, 16 bit AD value to [0,1000] */

e = ref - (y - (s32)(throttlePosMin))
    *1000L/((s32)(throttlePosMax)
    - (s32)(throttlePosMin));

I = I + e*(s32)(throttleKi) / (s32)(fs);
v = (s32)(throttleK)*e + I + uCoulombComp*100L
    + uPreTensionComp*100L;
v = v / 100L;
if (v < -(s32)THROTTLE_PWM_PERIOD_D01*90L/100L)
/* Note: duty cycle is limited to 90 percent
   of maximum duty cycle*/
    u = -(s32)THROTTLE_PWM_PERIOD_D01*90L/100L;
else if (v > (s32)THROTTLE_PWM_PERIOD_D01*90L/100L)
    u = (s32)THROTTLE_PWM_PERIOD_D01*90L/100L;
else
    u = v;

```

```
if (u >= 0)
{
    D01_DUTY( (u16)(u) );
    THROTTLE_SIGN(1);
}
else
{
    D01_DUTY( (u16)(-u) );
    THROTTLE_SIGN(0);
}

/* I-part correction */
I = I + (u - v) * (s32)(throttleKi) / (s32)(fs);
eOld = e; /* Update state */

/* Calculate control-signals for the observers. */

controlSignalAngularVel = (s32)(u - uCoulombComp);
controlSignalPreTension = (s32)(u - uCoulombComp);
}
```

Nedan visas *h*-filen som tillhör föregående programkod.

```
#ifndef throttle_h
#define throttle_h

/* Ticks, PWM frequency is 16MHz/64/THROTTLE_PWM_PERIOD */

#define THROTTLE_PWM_PERIOD          2500L
#define THROTTLE_DUTY_LIMIT          90L /* Percent */

/* Ticks, PWM frequency is 16MHz/64/THROTTLE_PWM_PERIOD_D01,
   used for external H-bridge with Selma */

#define THROTTLE_PWM_PERIOD_D01      420L

/***** Throttle variables *****/

enum ThrottleType {not_adapted, read_min,
                  open_throttle, read_max, adapted};

#ifdef common
    u16 throttleTestBit;
#else
    extern u16 throttleTestBit;
#endif

#ifdef common
    u16 throttleKd = 0;
#else
    extern u16 throttleKd;
#endif

#ifdef common
    enum ThrottleType throttle_mode;
#else
    extern enum ThrottleType throttle_mode;
#endif

#ifdef common
    u16 throttlePosMax = 0;
#else
    extern u16 throttlePosMax;
#endif

#ifdef common
    u16 throttlePosFilt = 0;
#else
    extern u16 throttlePosFilt;
#endif

#ifdef common
    u16 throttlePosMin = 0;
#else
    extern u16 throttlePosMin;
#endif

#ifdef common
```

```
    u16 throttlePosRange = 0;
#else
    extern u16 throttlePosRange;
#endif

#ifdef common
    /* 8.5 percent open throttle upon start */
    u16 throttlePosSet = 85;
#else
    extern u16 throttlePosSet;
#endif

#ifdef common
    u16 throttleK = 0;
#else
    extern u16 throttleK;
#endif

#ifdef common
    u16 throttleKi = 0;
#else
    extern u16 throttleKi;
#endif

#ifdef common
    u16 throttlePosNorm;
#else
    extern u16 throttlePosNorm;
#endif

#ifdef common
    u8 throttleTuning = 0;
#else
    extern u8 throttleTuning;
#endif

#ifdef common
    s32 angularVel = 0;
#else
    extern s32 angularVel;
#endif

#ifdef common
    s32 controlSignalAngularVel = 0;
#else
    extern s32 controlSignalAngularVel;
#endif

#ifdef common
    s32 controlSignalPreTension = 0;
#else
    extern s32 controlSignalPreTension;
#endif

#ifdef common
    s32 coulombComp = 18;
```

```
#else
    extern s32 coulombComp;
#endif

#ifdef common
    s32 uCoulombComp = 0;
#else
    extern s32 uCoulombComp;
#endif

#ifdef common
    s32 uPreTensionComp = 0;
#else
    extern s32 uPreTensionComp;
#endif

#ifdef common
    s32 preTension = 0;
#else
    extern s32 preTension;
#endif

#ifdef common
    s32 velMax = 1; /* To avoid division by zero. */
#else
    extern s32 velMax;
#endif

#endif /* throttle_h */
```