

Utvärdering och implementering av automatiska farthållare i fordonssimulator

Examensarbete
utfört inom **Fordonssystem**

av
Rikard Borst

Reg nr: LiTH-ISY-EX-3791-2006

17 februari 2006

Utvärdering och implementering av automatiska farthållare i fordonssimulator

Examensarbete

utfört inom **Fordonssystem**,
Institutionen för systemvetenskap
vid **Linköpings universitet**


av **Rikard Borst**

Reg nr: LiTH-ISY-EX-3791-2006

Handledare: **PhD student Anders Fröberg**
Linköpings Universitet
PhD student Erik Hellström
Linköpings Universitet

Examinator: **Professor Lars Nielsen**
Linköpings Universitet

Linköping, 17 februari 2006

	Avdelning, Institution Division, Department Vehicular Systems, Dept. of Electrical Engineering 581 83 Linköping	Datum Date 17 februari 2006
	Språk Language <input checked="" type="checkbox"/> Svenska/Swedish <input type="checkbox"/> Engelska/English <input type="checkbox"/> _____	Rapporttyp Report category <input type="checkbox"/> Licentiatavhandling <input checked="" type="checkbox"/> Examensarbete <input type="checkbox"/> C-uppsats <input type="checkbox"/> D-uppsats <input type="checkbox"/> Övrig rapport <input type="checkbox"/> _____
URL för elektronisk version http://www.vehicular.isy.liu.se http://www.ep.liu.se/exjobb/isy/2006/3791/		
Titel Utvärdering och implementering av automatiska farthållare i fordonssimulator Title Valuation and implementation of adaptive cruise control in an vehicle simulator		
Författare Rikard Borst Author		
Sammanfattning Abstract <p>Vehicle simulators are becoming more common in vehicle industries. Company earns lot of money on simulations instead of real tests. Real tests are necessary but not made so extensively as before.</p> <p>In this thesis the building of an vehicle simulator will be described and a comparison between three different cruise controls. The three cruise controls are PI-regulator, a regulator who regulates after positions in the terrain and a MPC-regulator. The reason for choosing this three is to see the difference between simple regulation and more complex regulation with respect to fuel consumption, travel time and complexity.</p> <p>The vehicle simulator is made in Matlab/Simulink, Visual Studio and Open Scene Graph. The facilities needed for running the simulator is a relative good computer with a graphics card on at least 128 MB RAM plus a steering wheel and pedals for brake and gas to achieve best feeling. A keyboard can be used but it reduces almost all feeling.</p> <p>After several simulations a conclusion was made. The MPC-regulator was the regulator who consumed least fuel and travel time. The regulator who regulates after positions in the terrain was not too far away. It would be interesting to do more research about it. In fact it is only a PI-regulator who makes "clever" decisions when a hill with enough slope appears. With enough slope means a downhill where the vehicle can accelerate without the use of fuel and an uphill where the vehicle can not keep its speed with maximum use of fuel.</p> <p>A conclusion was stated that the friction and height profile influenced on settings for the PI-regulator and with some adjustments on this settings, fuel could be saved.</p>		
Nyckelord Keywords Simulator, cruise control, tire models, friction		

Sammanfattning

Fordonssimulatorer blir allt vanligare ute i fordonsindustrin. Företagen tjänar mycket pengar på att simulera körningar och krocktest istället för att göra det med riktiga fordon. Detta görs men inte alls i samma utsträckning som förr.

I denna rapport redovisas uppbyggandet av en fordonssimulator samt en jämförelse mellan tre olika farthållare. De tre farthållarna är en PI-regulator, en MPC-regulator samt en regulator som reglerar efter intressanta punkter i terrängen. Anledningen till valet är att se skillnaden mellan enkel reglering och mer komplex reglering med avseende på bränsleförbrukning, tidsåtgång och komplexitet.

Fordonssimulatorens är uppbyggd i Matlab/Simulink, Visual Studio och Open Scene Graph. Hjälpmedel som behövs är en bra dator med ett grafikkort på minst 128 MB Ram samt ratt och pedaler för bästa känsla. Simulatorens går att köra med tangentbord men det tar bort mycket av känslan.

Efter åtskilliga simuleringar drogs slutsatsen att MPC-regulatorens var den som drog minst bränsle och tog minst tid. Regulatorens som kör efter intressanta punkter i terrängen låg inte långt efter och med tanke på den enkla reglering som ligger till grund vore det intressant att titta närmare på den. I själva verket är det bara en PI-regulator som tar "smarta" beslut vid backar med tillräcklig lutning. Med tillräcklig lutning menas nedförsbackar där lastbilen accelererar med släpmoment, dvs ingen gas, och uppförsbackar där lastbilen tappar fart med full gas.

Det visade sig att friktionskoefficienten och höjdprofilen var faktorer som påverkade inställningarna för PI-regulatorens och med en finjustering av parametrarna P och I fanns det framförallt bränsle att tjäna.

Nyckelord: Simulator, farthållare, däckmodell, friktion

Abstract

Vehicle simulators are becoming more common in vehicle industries. Company earns lot of money on simulations instead of real tests. Real tests are necessary but not made so extensively as before.

In this thesis the building of an vehicle simulator will be described and a comparison between three different cruise controls. The three cruise controls are PI-regulator, a regulator who regulates after positions in the terrain and a MPC-regulator. The reason for choosing this three is to see the difference between simple regulation and more complex regulation with respect to fuel consumption, travel time and complexity.

The vehicle simulator is made in Matlab/Simulink, Visual Studio and Open Scene Graph. The facilities needed for running the simulator is a relative good computer with a graphics card on at least 128 MB RAM plus a steering wheel and pedals for brake and gas to achieve best feeling. A keyboard can be used but it reduces almost all feeling.

After several simulations a conclusion was made. The MPC-regulator was the regulator who consumed least fuel and travel time. The regulator who regulates after positions in the terrain was not too far away. It would be interesting to do more research about it. In fact it is only a PI-regulator who makes “clever” decisions when a hill with enough slope appears. With enough slope means a downhill where the vehicle can accelerate without the use of fuel and an uphill where the vehicle can not keep its speed with maximum use of fuel.

A conclusion was stated that the friction and height profile influenced on settings for the PI-regulator and with some adjustments on this settings, fuel could be saved.

Keywords: Simulator, cruise control, tire models, friction

Förord

Detta examensarbete har utförts mellan september 2005 och januari 2006 hos Fordonssystem på Linköpings Universitet i samarbete mellan Anders Fröberg, Erik Hellström och Rikard Borst.

Erkännande

Jag vill tacka mina handledare Anders Fröberg och Erik Hellström för givande diskussioner och hoppas ni har användning för fordonssimulatore. Jag vill även tacka mina kompisar för alla härliga fredagsgrillar som ger nya tag inför kommande vecka. HKL Make Believes ska ha ett stort tack för att ni finns till och gör det möjligt att tänka på annat än skolan. Jonas Knutsson ska ha ett stort tack som lånade ut sovplats de gånger jag varit uppe i Linköping efter flytten till Jönköping. Sist men inte minst tack till min tjej Katarina Åkersten för att hon har stått ut med mig när jag gått hemma hela dagarna.

Jönköping, januari 2006

Rikard Borst

Innehåll

Sammanfattning	v
Abstract	vi
Förord	vii
1 Introduktion	1
1.1 Mål	2
1.2 Metod	2
2 Simulatorns uppbyggnad	3
2.1 Fordonsmodulen	4
2.1.1 Motorn	5
2.1.2 Drivlinan	6
2.1.3 Orientering	7
2.2 Inputmodulen	8
2.3 Kommunikationsmodulen	10
2.4 Ljudmodulen	10
2.5 Visualiseringsmodulen	10
2.6 Loggmodulen	10
3 Däckstudie	11
3.1 Simuleringar	14
4 Automatiska farthållare	17
4.1 Implementering i simulatören	20
5 Simuleringar	21
5.1 Jämförelse mellan tre regulatorer	21
5.1.1 Linköping-Jönköping, $\mu = 0.7$	23
5.1.2 Linköping-Jönköping, $\mu = 0.5$	28
5.1.3 Linköping-Jönköping, $\mu = 0.3$	28
5.1.4 Jönköping-Linköping, $\mu = 0.7$	29
5.1.5 Jönköping-Linköping, $\mu = 0.5$	29

5.1.6	Jönköping-Linköping, $\mu = 0.3$	34
5.2	Undersökning av P- och I-delen	35
5.2.1	10 % mindre P-, I- och PI-del	35
5.2.2	10 % mer P-, I- och PI-del	37
5.2.3	Jönköping-Linköping med nya P- och I-delen	38
6	Slutsatser	39
7	Framtida förbättringar	41
	Litteraturförteckning	42
A	Fordonsparametrar	43

Kapitel 1

Introduktion

Fordonssimulatorer börjar bli allt mer vanligt ute i fordonsindustrin. Företagen sparar miljoner på att simulera krocktest, manövertest och andra dyra tester med riktiga bilar. Dessa tester är självklart nödvändiga men görs inte alls i samma utsträckning som innan simulatorns tid. En välkänd simulator som är känd över hela världen är VTI:s fordonssimulator. Den är uppbyggd av ett gammalt bilchassi som är fastmonterat på ett metallband som kan flytta bilen i sidled och tilla den framåt och bakåt med en viss acceleration och hastighet, allt för att testföraren ska få en så realistisk känsla som möjligt. Se figur 1.1.



Figur 1.1: VTI:s fordonssimulator

1.1 Mål

Målet med detta examensarbete är att bygga om en bilsimulator till en lastbilsimulator, för att sedan undersöka olika regulatorer för automatiska farthållare och implementera dessa i simulatorn. Målet med undersökningen är att fastställa vilken reglering som är mest optimal med avseende på bränsleförbrukning, tidsåtgång och komplexitet.

1.2 Metod

För att lösa dessa problem behövdes en del inläring av de olika delarna, främst grafikdelen. Grafiken är gjord med Visual Studio och Open Scene Graph. De delarna i grafiken som var intressanta var höjdkarta, underlag, visning av intressanta data på instrumentpanelen och kommunikationen mellan alla delar. Själva utseendet är lämnat till personer inom detta område.

Tanken är att skapa ett program som tar en vektor med position och höjd, för att skapa en höjdprofil i visualiseringen som fordonet ska köra i. Vägprofilen som skapas blir en raksträcka pga att det blir för mycket jobb med att implementera svängarna, samt att grafiken är gjord på ett sätt som inte tillåter för stora kartprofiler.

Den största kunskapen ligger i Matlab/Simulink och det är där som tiden ska tillbringas mest. De saker som ska göras i Matlab/Simulink är bl. a. följande:

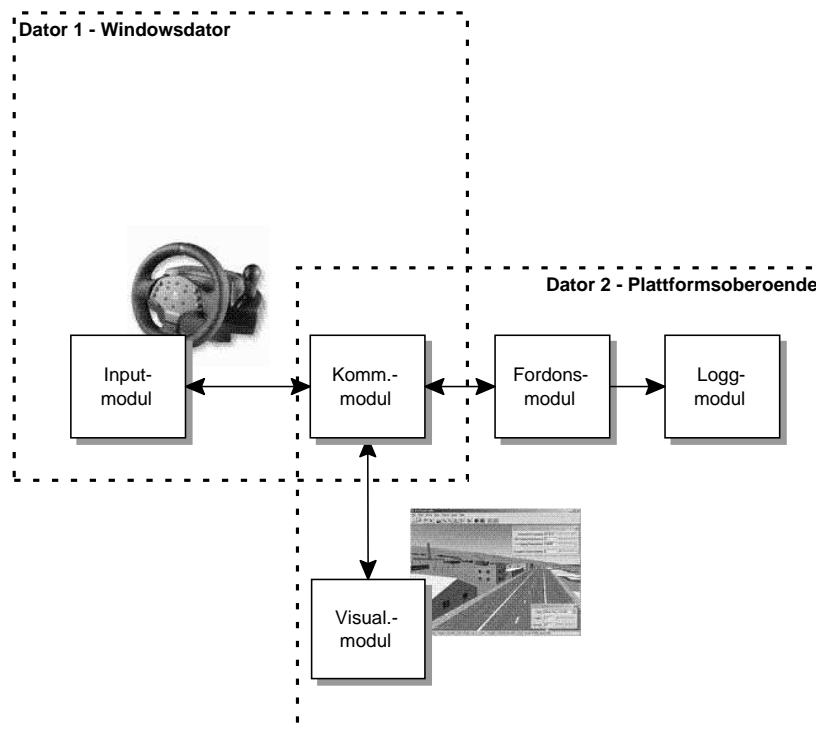
- Bygga om bilen till en lastbil, främst motorn.
- Parameterskattning för att få en så realistisk modell som möjligt.
- Hitta lämpligt gränssnitt för den automatiska farthållaren så det är lätt att byta ut en regulator mot en annan utan att behöva göra ett nytt program.
- En del undersökningar ska göras, främst på däcksidan. Det finns flera däcksmodeller att välja mellan så en studie har gjorts mellan två av modellerna.
- Jämföra olika regulatorer med varandra och med en mänsklig förare

De regulatorer som valts är en MPC-regulator, en regulator som reglerar efter utsatta punkter i terrängen och en PI-regulator. Anledningen till detta val är att se skillnaden mellan enkel, medel och svår reglering. PI-regulatorn styr enbart efter hastigheten medan MPC-regulatorn styr efter flera faktorer, bl. a. lutningen på vägen en bit längre fram.

Kapitel 2

Simulatorns uppbyggnad

Simulatorn består av sex moduler. Dessa är följande: fordonsmodul, inputmodul, kommunikationsmodul, ljudmodul, visualiseringmodul och loggmodul. Se figur 2.1. En snabb beskrivning av vad varje modul gör kommer i detta kapitel. För att få en bättre bild av modulerna se [7].

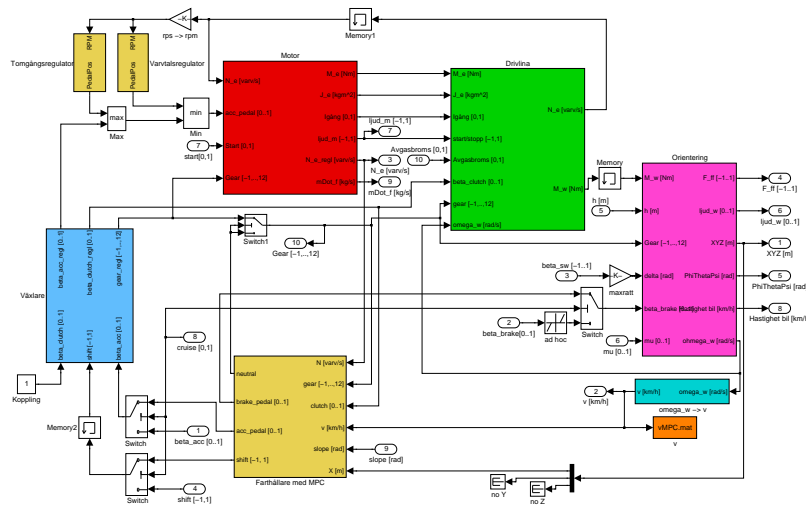


Figur 2.1: Översikt av systemet

2.1 Fordonsmodulen

Fordonsmodulen är uppbyggd i Matlab/Simulink. Den är modellerad med fem kroppar, ett chassi och fyra hjul, och kan röra sig i tre dimensioner. Modellen består av motor, drivlina, farthållare, växlare och orientering som räknar ut lokal och global position för tyngdpunkten. Modellen är moduluppbyggd, vilket gör det enkelt att byta ut en modul om så önskas. I bilaga A finns parametrarna som fordonet simulerats med.

Modulen kommunicerar med visualiserings-, ljud- och inputmodulen via kommunikationsmodulen och kommunicerar direkt med loggmodulen. Se figur 2.2.



Figur 2.2: Översikt av fordonsmodulen

Fordonsmodulen skickar information om position, rotation, hastighet, varvtal, momentan bränsleförbrukning och aktuell växel till visualiseringsmodulen och får i sin tur tillbaka information om höjd och friktion för varje hjul samt lutningen 1000 m längre fram.

Inputmodulen skickar information om rattvinkel, gaspedal- och bromspedalposition, växla upp/ner, farthållare på/av, starta fordonet och avgasbroms på/av till fordonsmodulen som skickar tillbaka information om hur mycket force-feedback som ska appliceras på ratten.

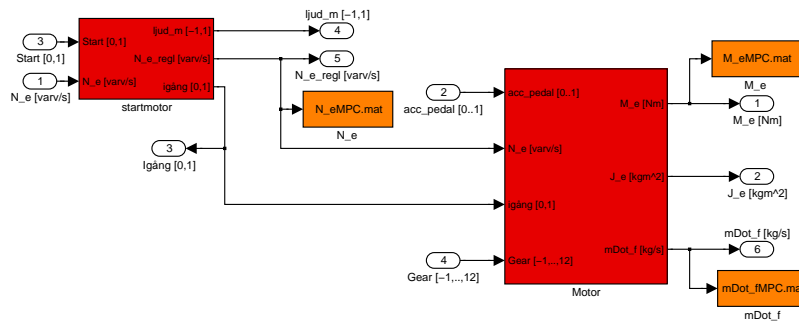
2.1.1 Motorn

Motorn är modellerad som i figur 2.3. Startmotorn har, förutom att starta motorn, i uppgift att hålla reda på om lastbilen får motorstopp. Dvs om motorn understiger ett visst varvtal stryps bränsletillförseln och föraren måste starta motorn igen.

I figur 2.2 finns det två block som heter tomgångsregulator och varvtalsregulator. Dessa kan räknas till motorn. Deras uppgifter är att reglera tomgången och skydda motorn från att varva över. Båda regulatorerna är enkla P-regulatorer med olika förstärkning.

I en förbränningsmotor blandas luft med bränsle som sedan antänds och skapar energi samt avgaser. Det som är intressant i denna studien är energin och bränsleförbrukningen. Därför används en motormapp för att få fram rätt moment och bränsleförbrukning som en funktion av varvtal och gaspedalsposition. Mappen är gjord av uppmätta punkter i stationärt tillstånd vilket medför att motorns inre friktion är inräknad. För en mer detaljerad beskrivning av motorn se [2].

Insignaler Varvtal, gaspedalsposition, starta motorn och växel
Utsignaler Moment, tröghetsmoment, motorn igång, ljud från motorn, varvtal och bränslemassflöde



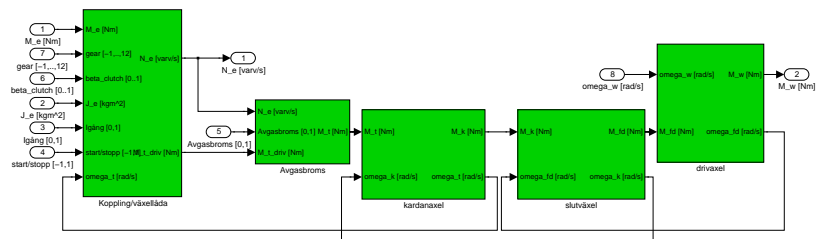
Figur 2.3: Översikt av motorn

2.1.2 Drivlinan

Drivlinan är modellerad som i figur 2.4. Drivlinan är implementerad som stel drivlina. Detta medför att inga vekheter förekommer och därmed ingen uppridning av kardanaxeln eller drivaxeln. Förluster finns i drivlinan för att göra den så realistisk som möjligt. Växellådan är av typen sekventiell med 12 växlar. Kopplingen är implementerad så att den slirar lite innan lamellerna slår ihop. Slutväxeln är en öppen differential som fördelar kraften lika på de drivande hjulen. Avgasbromsen är proportionell mot varvtalet med en faktor som genererar ett bromsande moment på 450 Nm vid 2500 varv/min. Den aktiveras med hjälp av inputmodulen. Den används inte aktivt vid reglering. Ekvationerna kan ses i [7].

Insignaler Moment, motorns tröghetsmoment, motorn igång, motorstopp, avgasbroms, kopplingsposition, växel och hjulhastighet

Utsignaler Varvtalet och drivande moment

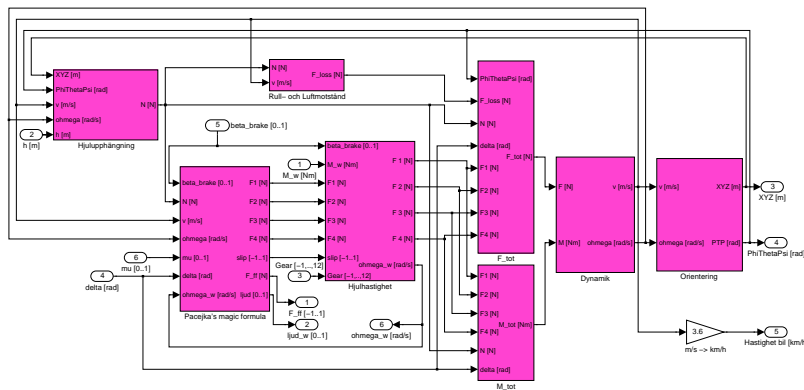


Figur 2.4: Översikt av drivlinan

2.1.3 Orientering

Orienteringsblocket är modellerat som i figur 2.5. Ekvationerna kan ses i [7]. Orienteringsblocket räknar ut de krafter och moment som verkar på fordonskroppen. Dessa beräknas i bilens respektive hjulens koordinatsystem. Därefter integreras dessa till hastigheter och rotationshastigheter för att sedan transformeras till global position och rotation för fordonet. Denna information skickas sedan till visualiseringsmodulen. Visualiseringsmodulen i sin tur skickar tillbaka höjd för respektive hjul samt friktion på underlaget fordonet färdas på.

- Insignaler** Moment på hjulen, höjden för varje hjul, växel, rattposition, broms och friktion på varje hjul
Utsignaler Varvtal och drivande moment



Figur 2.5: Översikt av orienteringsblocket

De krafter som verkar på fordonet är rull- och luftmotståndet samt gravitationen. Formeln för rull- och luftmotstånd ser ut enligt följande:

$$\begin{cases} F_a = \frac{1}{2} c_w A_a \rho_a v^2 \\ F_r = c_r N \end{cases} \quad (2.1)$$

där F_a är luftmotstånd, c_w är luftmotståndskoefficienten, A_a är maximala tvärsnittsarean av fordonet, ρ_a är luftens densitet, v är fordonets hastighet, F_r är rullmotstånd, c_r är rullmotståndskoefficienten och N är normalkraften verkandes på däckena. Se bilaga A för numeriska värden.

Däcksmodellen är av stor vikt för bränsleförbrukningen. De däcksm modeller som undersökts kommer från Pacejka [5]. Den ena modellen bygger på att slipet som uppstår mellan däck och kontaktytan beräknas med hjälp av en differens mellan däckshastigheten och fordonets hastighet. Formeln ser ut enligt följande:

$$s = \frac{r\omega - v}{r\omega} \quad (2.2)$$

Problemet med denna är att det inte går stå still eller låsa upp bromsarna för då delar vi med noll. Lösningen på detta är en switch som slår mellan dessa formler:

$$s = \begin{cases} \frac{r\omega - v}{r\omega} & r\omega > v \\ \frac{r\omega - v}{v} & r\omega < v \\ r\omega - v & v < 0.1 \end{cases} \quad (2.3)$$

Den andra modellen ser ut enligt följande:

$$\begin{aligned} \frac{du}{dt} &= r\omega - v - |v| \frac{u}{\sigma} \\ s &= \frac{u}{\sigma} \end{aligned} \quad (2.4)$$

där s är slipet, u är däckets deformation, v är fordonets hastighet, r är däcksradien, ω är hjulets hastighet och σ är deformationens längd i avslappnat läge. Fördelen med denna är att det går att stå still. Formeln fungerar så att vid låga hastigheter fungerar den som en fjäder och vid högre hastigheter som en dämpare. Problemet är att den oscillerar vid fix steglängd, vilket krävs för Real-Time Workshop. Se kapitel 3 för utförligare undersökning.

Hjulupphängningen är en olinjär funktion av hoptryckning och hoptryckningshastighet. Den fungerar så att fjädningen bottenar vid hög påfrestning samt att dämparna blir styvare vid högre hoptryckningshastigheter.

2.2 Inputmodulen

Inputmodulen kommunicerar med datorn via DirectX. Den skickar information till fordonsmodulen om gas, broms, rattutslag, starta motorn, avgasbroms på/av, farthållare på/av, tuta och växla. Fordonsmodulen skickar tillbaka kraften som ska läggas på ratten i form av force-feedback. Inputmodulen skickar även information om rattvinkeln till visualiseringsmodulen.

Knapparna på ratten används till att starta motorn, sätta på/stänga av farthållaren, sätta på/stänga av avgasbromsen, tuta och växla. Se figur 2.6 för placering. Vid styrning med hjälp av tangentbord, se tabell 2.1 för beskrivning av aktuella knappar och dess funktioner.



Figur 2.6: Knapparnas funktion och placering

Knapp	Funktion
Pil framåt	Gasa
Pil bakåt	Bromsa
Pil vänster	Sväng åt vänster
Pil höger	Sväng åt höger
A	Växla upp
Z	Växla ner
D	Starta motorn
G	Farthållare på/av
E	Avgasbroms på/av
C	Tuta

Tabell 2.1: Knapparnas funktion vid styrning med hjälp av tangentbord

2.3 Kommunikationsmodulen

Kommunikationen består av enkla textmeddelanden som skickas via en server ut till respektive klient. Varje modul har en klient. Meddelanden skickas kontinuerligt och det finns inga krav på att de måste komma fram. Meddelanden hamnar i en lista hos varje klient som sedan öppnar dessa meddelanden efter hand. För mer detaljerad beskrivning se [7].

2.4 Ljudmodulen

Ljudmodulen tar emot information från fordonsmodulen och spelar upp dessa. Ljuden består av korta ljudfiler som spelas upp med olika frekvens och volym om och om igen. De ljud som spelas upp kommer från motorn, däckskrik, fartvind, växling och motor på/av. För mer detaljerad beskrivning se [7].

2.5 Visualiseringsmodulen

Visualiseringen är programmerad i Visual Studio och gjord med hjälp av Open Scene Graph. För mer detaljerad beskrivning se [7].

Visualiseringsmodulen tar bitmap-filer som den läser av färgskalan på. Höjdkartan består av blå färg som är skalad mellan 0 och 255. 255 är högsta punkten och 0 lägsta. Bitmap-filerna görs i Matlab med ett program som tar en matris med höjd och position som inargument och skapar tre bitmap-filer. De representerar höjdkarta, vägkarta och friktionskarta.

Visualiseringsmodulen får information från fordonsmodulen om position, rotation, hastighet, varvtal, momentan bränsleförbrukning och aktuell växel. Visualiseringsmodulen skickar tillbaka höjd för varje hjul, aktuell friktionskoefficient och lutningen 1000 m längre fram. De signaler som representeras på instrumentpanelen är hastighet, varvtal, växel, momentan bränsleförbrukning, aktuell position och lutning 1000 m längre fram.

2.6 Loggmodulen

Loggmodulen lagrar intressanta data från fordonsmodulen. Alla orange block i fordonsmodulen är variabler som lagras. De intressanta data som lagras är varvtal, moment, bränslemassflöde, körd sträcka, hastighet, slip, växel, gas- och bromspedalsläge. Data lagras som .mat-filer och kan sedan användas i Matlab.

Kapitel 3

Däckstudie

En avgörande faktor vid simulering av fordon är däckmodellen. Utan en bra däckmodell fås orealistisk bränsleförbrukning och modellen kan börja oscillera vid gaspådrag eller broms. Av denna anledning antas ofta rullvillkor vid modellering av fordon för att undvika svårigheterna med att modellera däckens uppvridning, vilket sker i verkligheten. Detta kan göras med gott samvete eftersom uppvridningen är ytterst liten vid konstant farthållning. Vid gaspådrag ökar uppvridningen och rullvillkoret gäller inte för drivande hjul. MPC-regulatorn är framtagen under rullvillkor så det ska bli intressant att se hur den klarar sig då uppvridningen kommer med i beräkningarna.

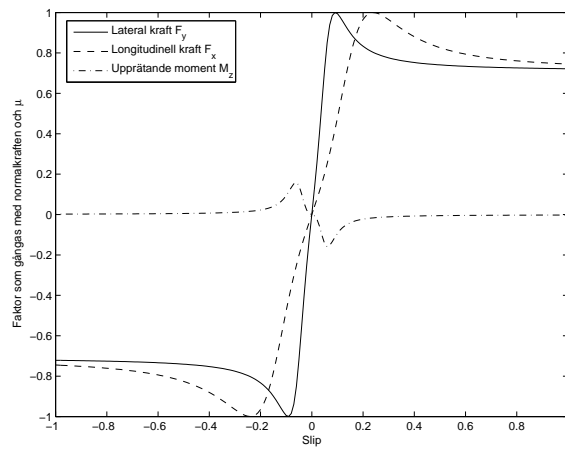
I denna studie ska vi undersöka skillnaden mellan två däckmodeller tagna ur [5]. De två däckmodellerna är följande:

$$s = \begin{cases} \frac{r\omega - v}{r\omega} & r\omega > v \\ \frac{r\omega - v}{v} & r\omega < v \\ r\omega - v & v < 0.1 \end{cases} \quad (3.1)$$

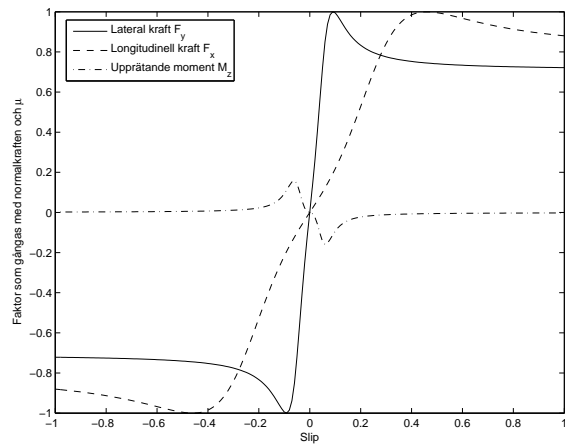
$$\begin{cases} \frac{du}{dt} = r\omega - v - |v| \frac{u}{\sigma} \\ s = \frac{u}{\sigma} \end{cases} \quad (3.2)$$

där s är slip, u är däckets deformation, v är fordonets hastighet, r är däcksradien, ω är hjulets hastighet och σ är deformationens längd i avslappnat läge.

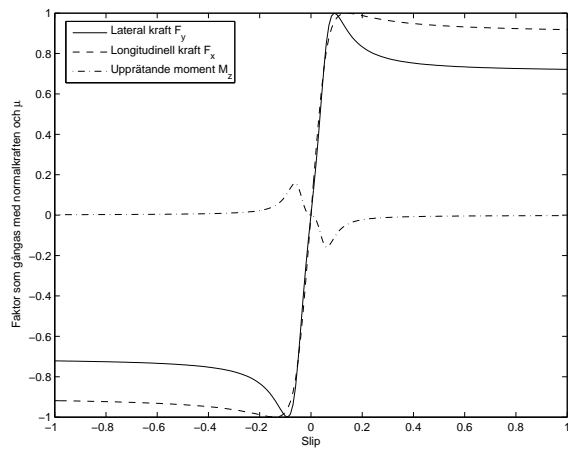
Problemet med den första ekvationen är att övergångarna motsvarar en switch i Simulink, vilket medför att vi tappar fysiken vid övergångarna. Problemet med den andra ekvationen är att tidskonstanten för systemet varierar och är för liten vid vissa tillfällen. Därmed går det inte att simulera däckmodellen med fix steglängd, vilket krävs för Real-Time Workshop som simulatören körs i. Tillsammans med följande slipkurvor får vi en däckmodell. Se figur 3.1, 3.2 och 3.3 samt appendix A för formlerna till figurerna.



Figur 3.1: Slipkurva för ekvation 1



Figur 3.2: Slipkurva för ekvation 2 med fix steglängd



Figur 3.3: Slipkurva för ekvation 2 med variabel steglängd

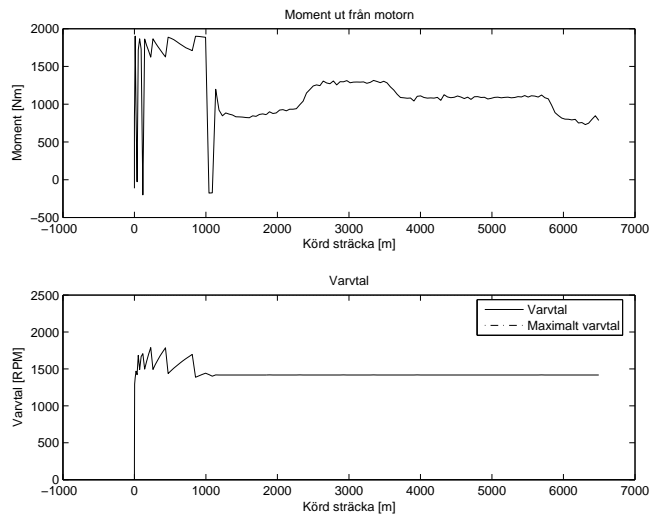
3.1 Simuleringar

Simuleringarna är gjorda på vägen mellan Linköping-Jönköping och de är simulerade i 300 s. Lastbilen startar stillastående och accelererar till 85 km/h, en hastighet som den sedan håller.

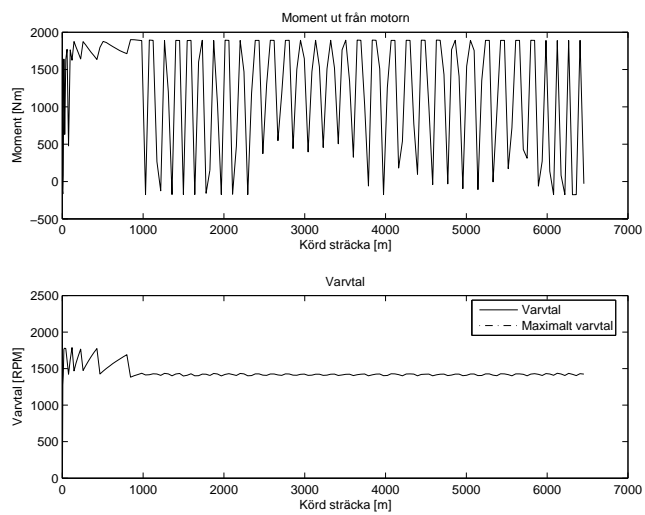
- Första simuleringen är gjord med ekv. 3.1 och fix steglängd med samplingstid 0.01 s. Resultatet kan ses i figur 3.4.
- Andra simuleringen är gjord med ekv. 3.2 och fix steglängd med samplingstid 0.01 s. Resultatet kan ses i figur 3.5.
- Tredje simuleringen är gjord med ekv. 3.2 och variabel steglängd. Resultatet kan ses i figur 3.6.

Med den variabla steglängden ser vi att momentet inte oscillerar. Se figur 3.6. Alltså beror svängningarna i momentet med fix steglängd på just steglängden. Steglängden varierar mellan $2.9110 \cdot 10^{-19}$ och 0.0559 s. Snittsteglängden är emellertid 0.0026 s men med de stora variationerna blir det svårt för Simulink att räkna ut vettiga värden med fix steglängd. När momentet oscillerar så kraftigt fås ingen vettig bränsleförbrukning. En annan observation som gjordes med ekv. 3.2 och variabel steglängd var att den hade problem då friktionen tilläts variera. Oscillationer i momentet uppstod vilket orsakade en bränsleökning med 2 dl/mil ungefär. Se figur 3.7.

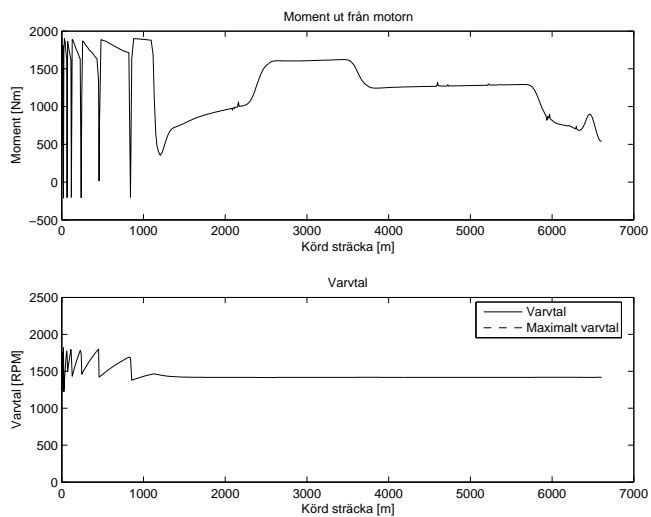
Trots att vi tappar fysiken i övergångarna för switchen fungerar ekv. 3.1 bra. Inga kraftiga oscillationer som i den andra simuleringen med ekv. 3.2 och fix steglängd. Därmed kommer ekv. 3.1 att användas vid simuleringarna senare.



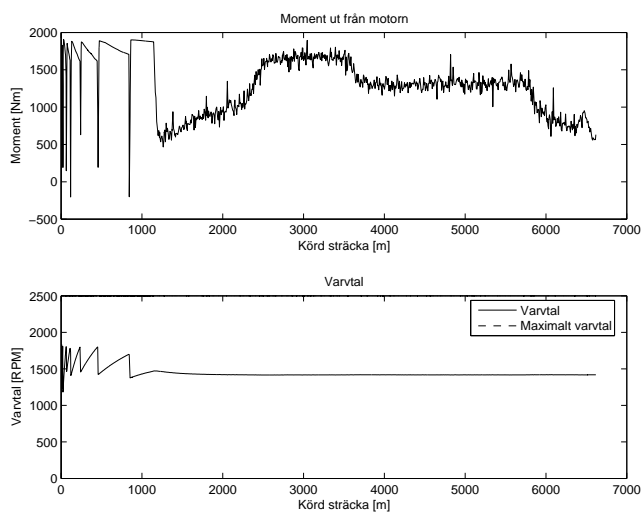
Figur 3.4: Moment och varvtal med däckmodell 1



Figur 3.5: Moment och varvtal med däckmodell 2



Figur 3.6: Moment och varvtal med däckmodell 2 och variabel steglängd



Figur 3.7: Moment och varvtal med däckmodell 2 och variabel steglängd samt varierande friktion

Kapitel 4

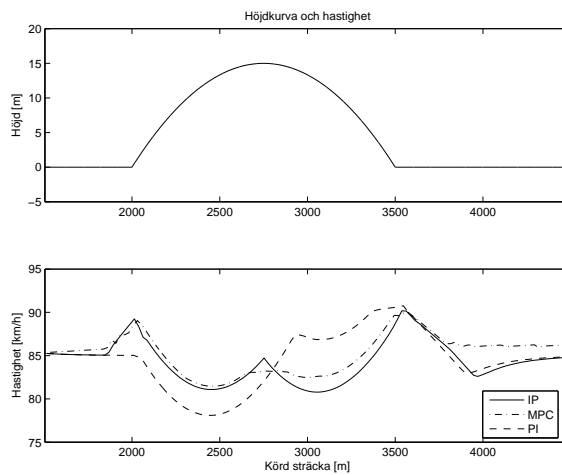
Automatiska farthållare

För att minska utsläppen från tunga transporter har företag som bland annat Scania börjat titta på automatiska farthållare. Det innebär att föraren styr raten men själva körningen tas om hand av en regulator som planerar körningen efter en topografisk karta. Föraren har självklart den övergripande kontrollen men så länge inget händer i trafiken kör lastbilen efter regulatorm. Detta medför, i vissa fall, att det går åt mindre bränsle och därmed blir det mindre avgaser.

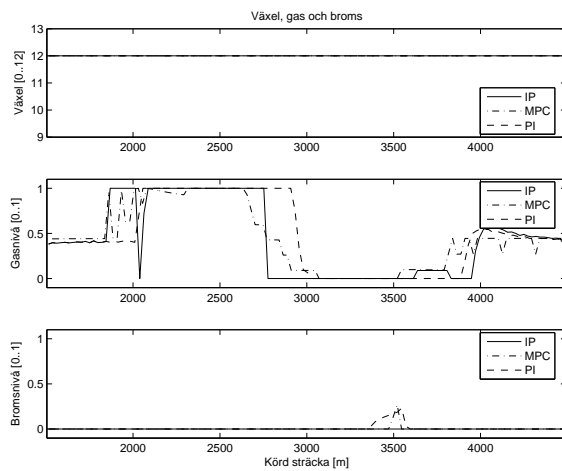
Det finns flera strategier när det gäller regleringen. En strategi som har undersökts i [6] går ut på att körsträckan simuleras innan körning och där man tittar på lutningen hos vägen. Det finns en maximal lutning för en motor där man tar sig upp utan att tappa fart på maximalt moment och en minimal lutning där lastbilens hastighet ökar med släpmoment. Dessa lutningar används på ett optimalt sätt genom att exempelvis släppa gasen innan en nedförsbacke med tillräcklig lutning för att vid backens slut uppnå maximal hastighet och därmed inte behöva använda bromsen, förutom i de allra värsta backarna. Mellan dessa backar körs lastbilen med vanlig PI-regulator. Denna körstrategi kommer förkortas IP hädanefter.

Andra strategier som görs under körning är MPC-regulatorn som beskrivs i [2]. Där ”tittar” regulatorn på lutningen en eller ett par km fram i terrängen och tar beslut därefter. Besluten grundas med avseende på en kostnadsfunktion som väger följande fem faktorer mot varandra: bränsleförbrukning, negativ avvikelse från referenshastigheten, hastighetsförändringar, växlingar och användandet av bromsarna. Denna reglering är ganska beräkningskrävande men med dagens teknologi är det inga problem att simulera den.

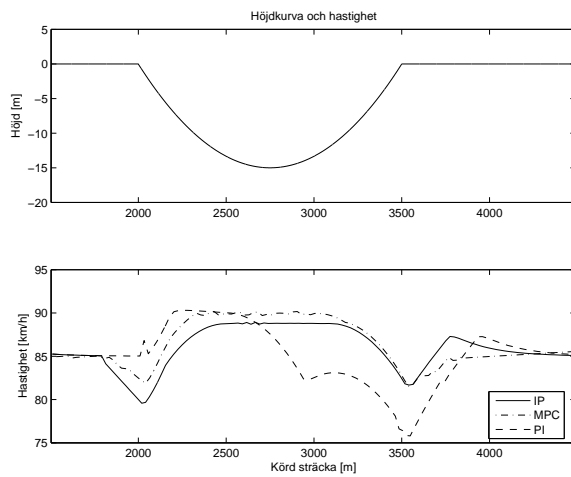
För att visa skillnaden mellan körstrategierna har två simuleringar gjorts, första med en kulle och den andra med en sänka. Resultaten kan ses i figur 4.1 - 4.4.



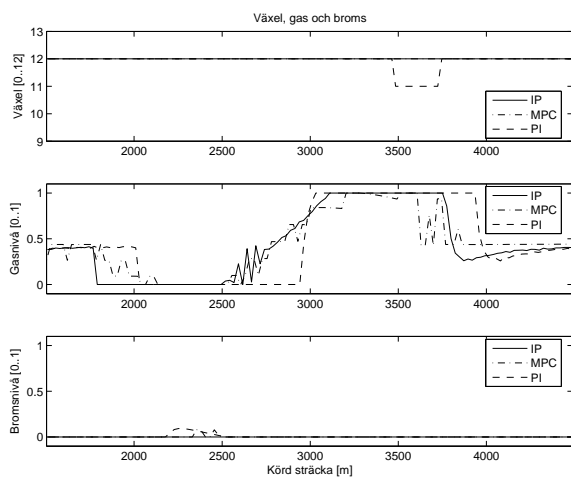
Figur 4.1: Backen börjar med en lutning på 4 % och avtar till -4 %. MPC-regulatorn och farthållaren med intressanta punkter, IP, gasar upp lastbilen till ca 89 km/h innan backen för att ha så mycket moment som möjligt i uppførsbacken. Innan nedförsbacken släpper de av på gasen och utnyttjar backen för att accelerera. Se figur 4.2.



Figur 4.2: MPC- och IP-regulatorn gasar innan backen och har därmed högre hastighet vid backens topp än PI-regulatorn. De använder även nedförsbacken på ett bättre sätt genom att använda släpmomentet för att låta lastbilen accelerera. Se figur 4.1.



Figur 4.3: Backen börjar med en lutning på -4% för att sedan öka till 4% . MPC- och IP-regulatorn låter lastbilen sakta ner innan nedförsbacken för att sedan använda släpmoment och öka lastbilens hastighet till knappt 90 km/h i slutet av nedförsbacken. Se figur 4.4.



Figur 4.4: MPC- och IP-regulatorn släpper gasen tidigare och behöver därmed inte använda lika mycket broms i nedförsbacken som PI-regulatorn, vilket medför energiförlust. De gasar på strax innan uppförsbacken för att ha maximalt moment i uppförsbacken. Se figur 4.3.

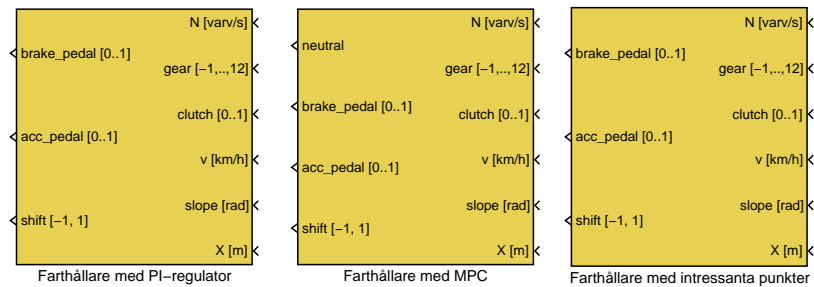
4.1 Implementering i simulatorn

Vid implementeringen av den automatiska farthållaren är det viktigt med ett bra gränssnitt så det är lätt att byta ut algoritmen som optimerar körningen. Som insignaler används varvtal, växel, kopplingsposition, hastighet, körd sträcka och väg lutning 1000 m längre fram. Utsignaler blir gas, broms och växla upp/ner samt neutral för MPC-regulatorn. I simulatorn ser blocken ut som i figur 4.5.

PI-regulatorn är implementerad som en ”Embedded Matlab Function”, vilket är ett block som man skriver sin egen Matlab-kod i. Farthållaren som reglerar körningen efter intressanta punkter är också implementerad som en ”Embedded Matlab Function”. MPC-regulatorn är implementerad som en ”S-function”, vilken man skriver C-kod i. [9]

PI- och MPC-regulatorn är tagna från [2] medan IP-regulatorn är gjord efter beskrivningar i [6]. De parametrar som används i de olika regulatorerna är hämtade ur [2] och [6]. De kan ses i bilaga A.

För att hitta punkterna i terrängen skapades ett program som simulerar en lastbilmodell på den aktuella körsträckan med olika starthastigheter/startpunkter tills lastbilen har rätt hastighet i slutet av nedförsbacken samt i början av uppförsbacken. När den har det sparas punkter i terrängen där lastbilen ska släppa gasen inför en nedförsbacke samt gasa inför en uppförsbacke. Dessa punkter används sedan av regulatorn i simulatorn.



Figur 4.5: Farthållare i Simulink med gränssnitt

Kapitel 5

Simuleringar

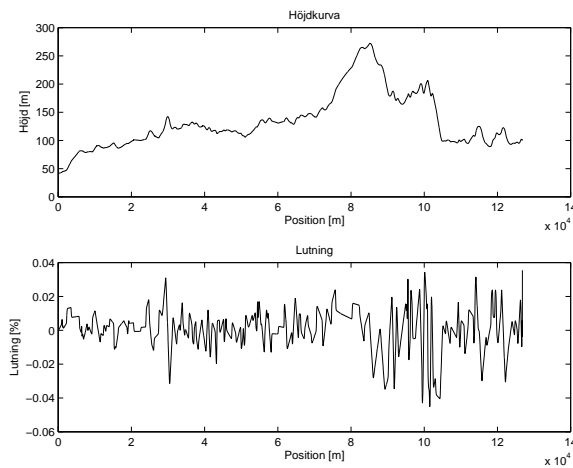
Simuleringarna är gjorda "offline", vilket innebär att de inte är simulerade i realtid. Fordonsmodulen styrs med "look-up tables" med position som insignaler och höjddata för varje hjul som utdata, resten sköts av farthållaren. Det är enbart longitudinell körning som gäller. Modellerna med PI- och IP-regulatorn går att köra i realtid men inte MPC-regulatorn. Detta pga en funktion som används för att få rätt insignaler till regulatorn. Den stöds inte av Real-Time Workshop. Själva regulatorn fungerar i realtid.

Vid framtagning av intressanta punkter i terrängen upptäcktes intressanta fenomen. Om det kommer två nedförsbackar efter varandra och den andra backen är längre än 2 km medför det svårigheter i att bestämma var lastbilen ska släppa gasen. Avståndet mellan backarna är inte tillräckligt vilket medför att man måste ta båda backarna som en. Svårigheten med detta ligger i att hastigheten mellan backarna understiger minsta tillåtna hastighet. För att komma undan detta problemet finns det två alternativ. Antingen behandlar man backarna som två och bromsar lastbilen i den andra backen eller så kör man med farthållaren inställd på minsta tillåtna hastighet vid behov. Nackdelen att köra med farthållaren är att det tar längre tid att köra sträckan Linköping - Jönköping och vinsten i energi väger knappt upp tidsförlusten. I följande kapitel följer en redogörelse av resultaten från simuleringarna.

5.1 Jämförelse mellan tre regulatorer

För att ta reda på vilken körstrategi som drar minst bränsle och tar minst tid gjordes en jämförelse mellan en vanlig PI-regulator, en farthållare som kör efter utsatta punkter (förkortas IP) och en MPC-regulator. Det finns två typer av IP-regulatorer som är undersökta. Dessa är IP70 som tillåter hastigheter ner till 70 km/h, och IP80 som tillåter hastigheter ner till 80 km/h. Däcksmodellen som användes var ekv. 3.1. Körsträckan som simuleringarna är gjorda på är Linköping-Jönköping och Jönköping-Linköping. Höjdprofilen för sträckan ser ut enligt figur 5.1.

Friktionskoefficienten, μ , är satt till 0.7, 0.5 och 0.3. Friktionen varierar från det ursprungliga värdet med maximalt ± 0.1 för att det ska vara så realistiskt som möjligt. Se tabell 5.1 för motsvarande väglag som är hämtat ur [3].



Figur 5.1: Höjdkurva och lutning mellan Linköping - Jönköping

μ	Väglag
0.7	Ny beläggning eller fräst yta
0.5	Motsvarande dagens asfaltväglag
0.3	Packad snö eller grus

Tabell 5.1: Friktionskoefficienternas motsvarighet till väglag

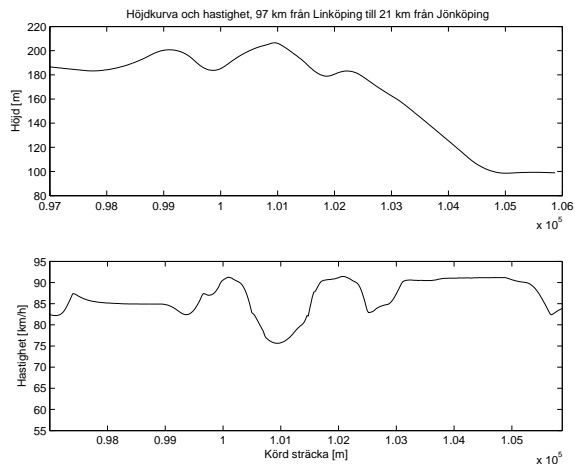
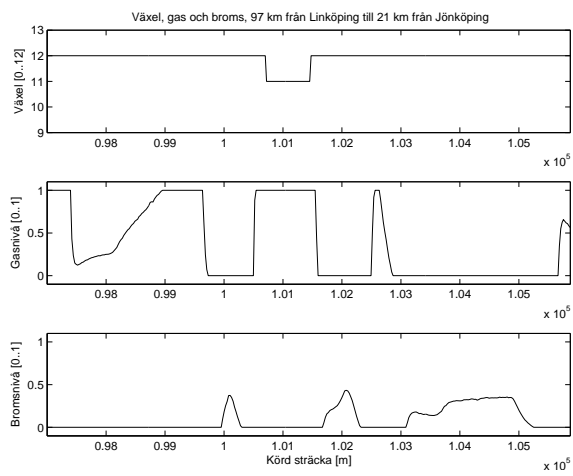
5.1.1 Linköping-Jönköping, $\mu = 0.7$

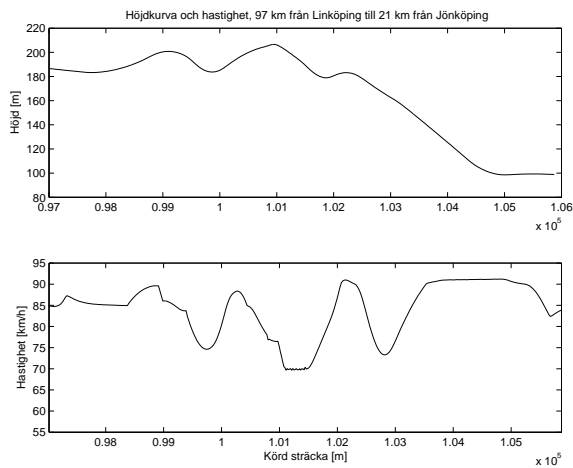
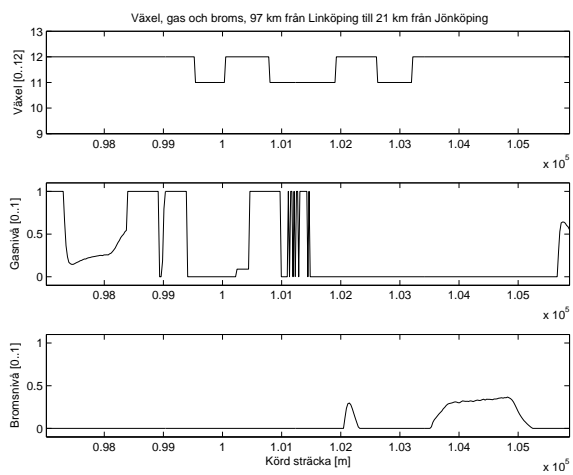
Första simuleringen gjordes från Linköping till Jönköping med friktionskoefficienten satt till 0.7. Resultatet finns i tabell 5.2.

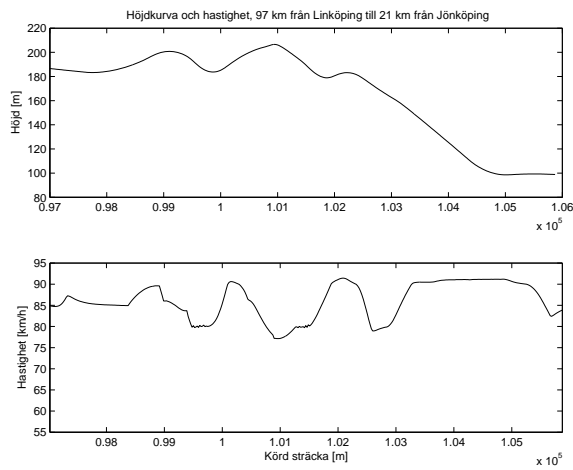
Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.6906	5405
IP70	3.6141	5425
IP80	3.6506	5405
MPC	3.6442	5404

Tabell 5.2: Resultat från simuleringarna mellan Linköping-Jönköping, $\mu = 0.7$

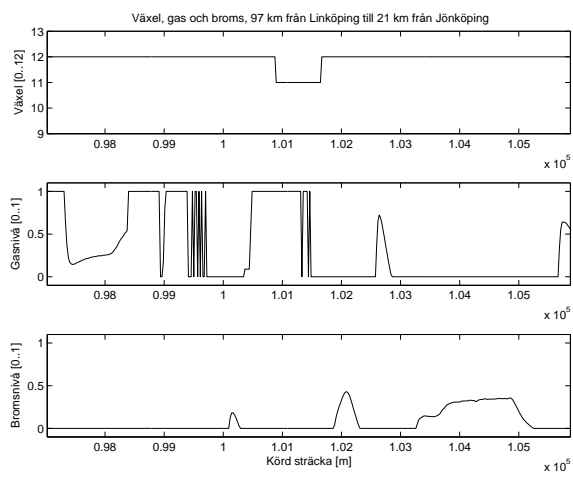
Med resultaten i bakgrund ser vi att IP70-regulatorn är den som drar minst bränsle. Den är 2 % bättre än PI-regulatorn per mil. Den tar dock lite längre tid än den vanliga farthållaren och MPC-regulatorn vilket är ett argument lastbilsåkerierna håller hårt på. IP80-regulatorn är något sämre än MPC-regulatorn men med tanke på den relativt enkla reglering som ligger bakom regulatorn fungerar den bra ändå. I figurerna 5.2 - 5.9 visas intressant data för vägsträckan 97 km från Linköping till 21 km från Jönköping. I figurerna kan vi se hur de olika regulatorerna arbetar. Det syns tydligt hur IP- och MPC-regulatorn utnyttjar backarna för att accelerera fordonet och slippa bromsa medan PI-regulatorn måste bromsa mer i nedförsbackarna, vilket resulterar i onödiga energiförluster.

Figur 5.2: PI-regulatorn, $\mu = 0.7$ Figur 5.3: PI-regulatorn, $\mu = 0.7$

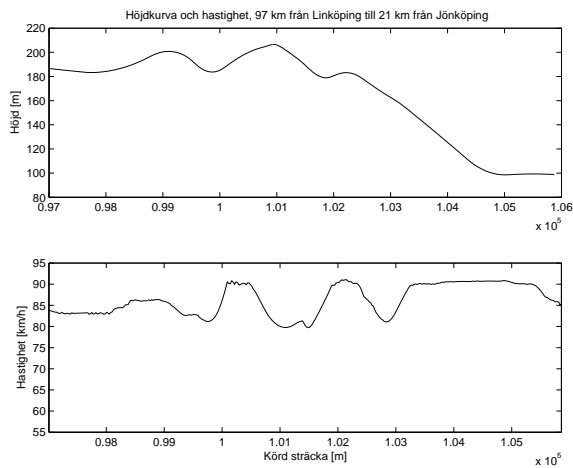
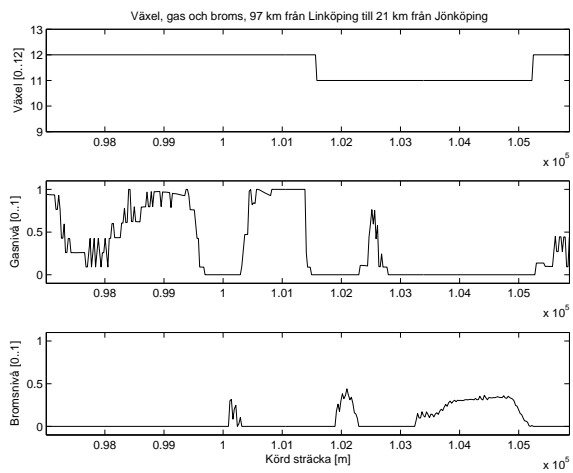
Figur 5.4: IP70-regulatorn, $\mu = 0.7$ Figur 5.5: IP70-regulatorn, $\mu = 0.7$



Figur 5.6: IP80-regulatorn, $\mu = 0.7$



Figur 5.7: IP80-regulatorn, $\mu = 0.7$

Figur 5.8: MPC-regulatorn, $\mu = 0.7$ Figur 5.9: MPC-regulatorn, $\mu = 0.7$

5.1.2 Linköping-Jönköping, $\mu = 0.5$

Andra simuleringen gjordes mellan Linköping och Jönköping fast med $\mu = 0.5$ istället. Se tabell 5.3 för resultaten.

Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.6942	5410
IP70	3.6182	5430
IP80	3.6524	5410
MPC	3.6443	5410

Tabell 5.3: Resultat från simuleringarna mellan Linköping-Jönköping, $\mu = 0.5$

Resultaten liknar dem med $\mu = 0.7$. Alla regulatorer drar marginellt mer bränsle och tar ung. 5 s längre tid. Minst skillnad från föregående simulering är det för MPC-regulatorn. Den utnyttjar tiden för att dra ner bränsleförbrukningen till samma nivå i stort sett. Här kan vi även se vikten av hur viktiga sekunderna är för bränsleförbrukningen. Trots att det bara skiljer 20 s mellan IP70 och IP80 på 12.6 mils körning så drar IP80 0.34 dl/mil mer bränsle. Med tanke på att en tung lastbil rullar ca 6-8000 mil på ett år blir sekunderna betydande.¹

5.1.3 Linköping-Jönköping, $\mu = 0.3$

Sista simuleringen gjordes med friktionskoefficienten 0.3, vilket motsvarar packad snö eller grus. Resultaten blev anmärkningsvärda. Se tabell 5.4.

Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.8636	5416
IP70	3.7749	5434
IP80	3.8190	5414
MPC	3.6469	5415

Tabell 5.4: Resultat från simuleringarna mellan Linköping-Jönköping, $\mu = 0.3$

MPC-regulatorn fungerar nu bäst men mest anmärkningsvärt är att skillnaden i bränsleförbrukning inte är mer jämfört med när μ var satt till 0.5. Ser vi på tidsåtgången är det än mer imponerande. MPC-regulatorn lyckas utnyttja den extra tidsåtgången på 5 s för att få ner bränsleförbrukningen till samma nivå som vid bättre väglag. Detta tyder på att MPC-regulatorn är robust, vilket är ett krav vid implementering i verkligheten.

Som vi ser på de andra regulatorerna drar de 1.5 dl/mil mer bränsle, vilket är en relativt stor ökning. Anledningen till den stora ökningen i bränsleförbrukningen hos IP70-, IP80- och PI-regulatorn kommer i senare kapitel.

¹Hämtat från <http://www.scb.se>

5.1.4 Jönköping-Linköping, $\mu = 0.7$

Rent intuitivt genom att bara kolla på höjdprofilen för körsträckan borde bränsleförbrukningen bli mindre. Dock finns det ett par rejäla stigningar ett par mil efter Jönköping som kan dra upp snittförbrukningen rejält. Se tabell 5.5 för resultaten.

Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.2980	5474
IP70	3.2952	5441
IP80	3.2999	5433
MPC	3.2737	5441

Tabell 5.5: Resultat från simuleringarna mellan Jönköping-Linköping, $\mu = 0.7$

Resultaten besannar påståendet innan med blandade intryck. MPC-regulatorn drar minst bränsle men IP80 tar minst tid på sig. Tidsskillnaden räcker dock inte för att kompensera bränsleförlusten, som vi kan se genom att titta på IP70-regulatorns resultat. PI-regulatorn drar relativt lite bränsle men tidsförlusten blir för stor för att den ska vara lönsam. Den största tidsförlusten beror på att PI-regulatorn har 85 km/h in i de stora backarna strax efter Jönköping medan IP- och MPC-regulatorn har 89 km/h och har därmed mer moment in i backen, vilket resulterar i en högre lägsta hastighet.

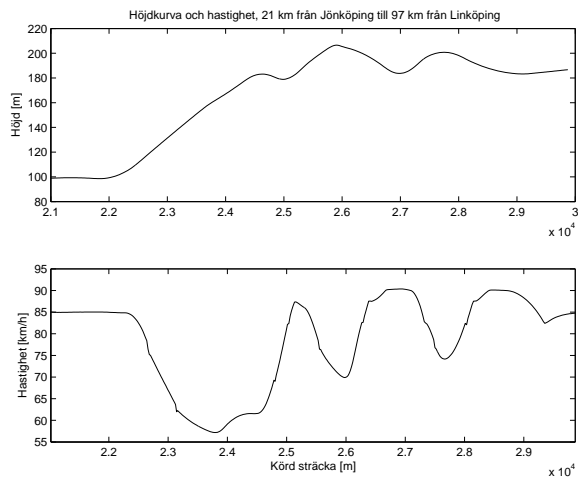
5.1.5 Jönköping-Linköping, $\mu = 0.5$

Se tabell 5.6 och figurerna 5.10 - 5.17. Resultaten liknar dem från föregående underkapitel. Skillnaden blir mindre mellan $\mu = 0.7$ och $\mu = 0.5$ för Jönköping-Linköping än mellan Linköping-Jönköping. Alltså har friktionskoefficienten större betydelse för vägsträckor med mycket uppförsbackar än för motsatta förhållandet.

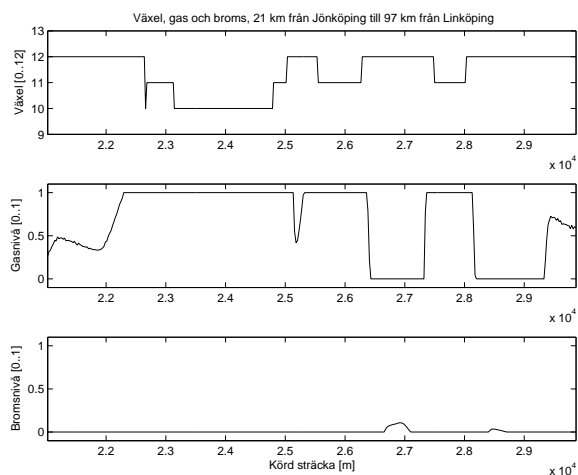
Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.3038	5478
IP70	3.2987	5442
IP80	3.3029	5436
MPC	3.2797	5443

Tabell 5.6: Resultat från simuleringarna mellan Jönköping-Linköping, $\mu = 0.5$

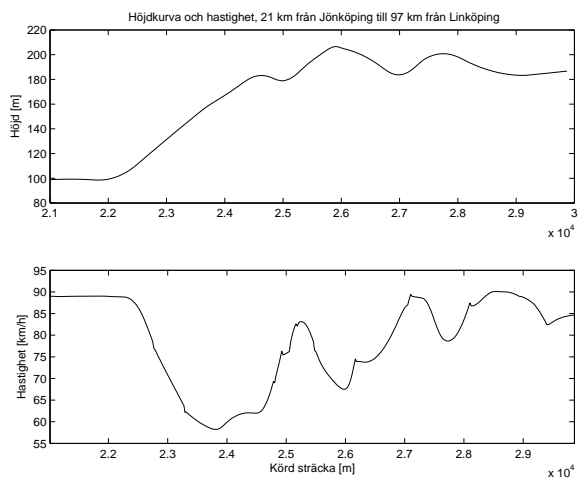
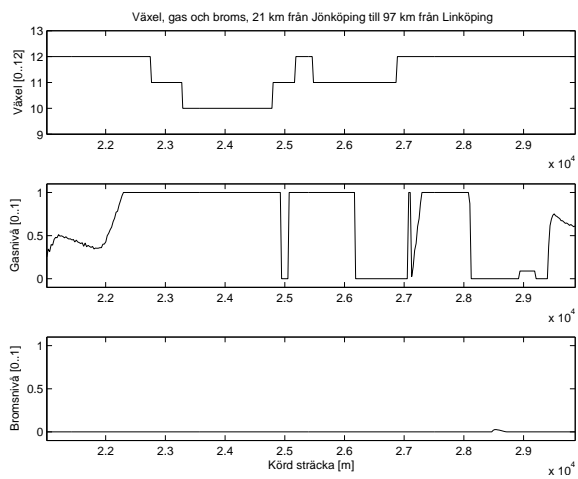
I figurerna kan vi se hur IP- och MPC-regulatorn har en hastighet uppåt 90 km/h innan lastbilen når första uppförsbacken, medan PI-regulatorn har 85 km/h in i backen. Detta resulterar i att PI-regulatorn har mindre moment in i backen och får en mindre lägsta hastighet. Därmed förlorar PI-regulatorn värdefull tid. IP- och MPC-regulatorn utnyttjar även nedförsbacken vid 2.6-2.7 mil från Jönköping medan PI-regulatorn gasar upp lastbilen till 90 km/h och får sedan bromsa för att inte överstiga 90 km/h.

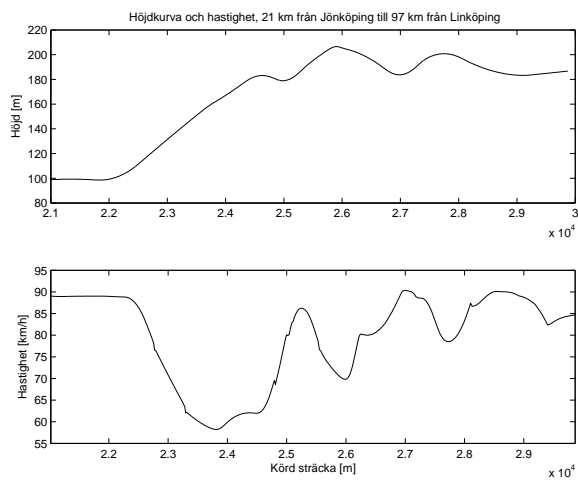
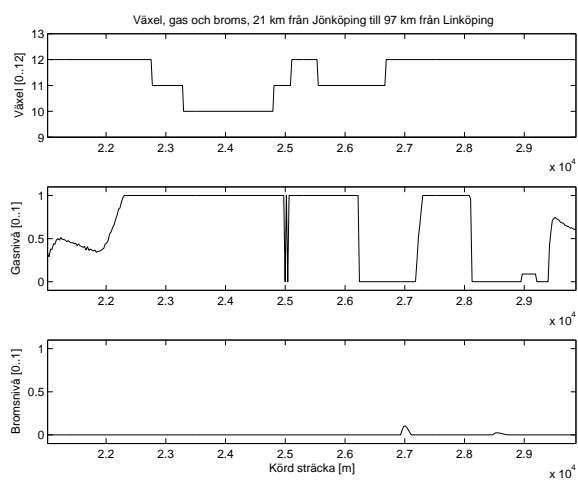


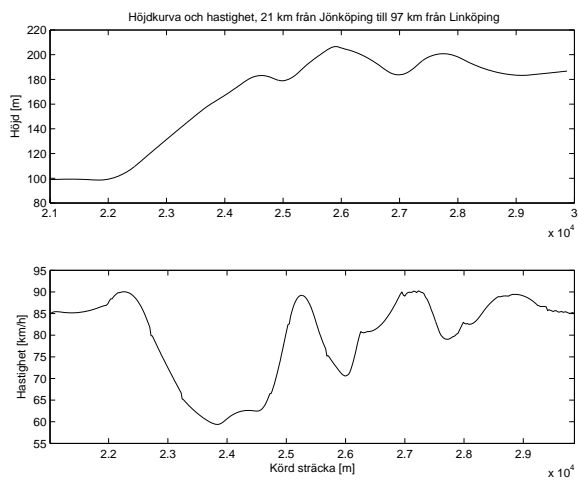
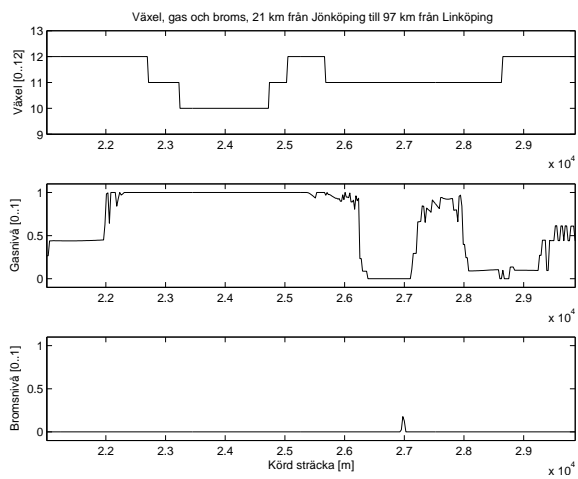
Figur 5.10: PI-regulatorn, $\mu = 0.5$



Figur 5.11: PI-regulatorn, $\mu = 0.5$

Figur 5.12: IP70-regulatorn, $\mu = 0.5$ Figur 5.13: IP70-regulatorn, $\mu = 0.5$

Figur 5.14: IP80-regulatorn, $\mu = 0.5$ Figur 5.15: IP80-regulatorn, $\mu = 0.5$

Figur 5.16: MPC-regulatorn, $\mu = 0.5$ Figur 5.17: MPC-regulatorn, $\mu = 0.5$

5.1.6 Jönköping-Linköping, $\mu = 0.3$

Vid simuleringar med $\mu = 0.3$ blev resultatet inte som förväntat. För IP- och PI-regulatorn uppstod det små svängningar vid konstant farthållning, vilket medförde att dessa drog relativt mycket mer bränsle än då $\mu = 0.5$. I senare kapitel kommer en undersökning om vad som händer då P- och I-delen ökas respektive minskas, och en förklaring till dessa fenomen. Se tabell 5.7.

Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.4814	5482
IP70	3.4810	5450
IP80	3.4914	5441
MPC	3.2791	5448

Tabell 5.7: Resultat från simuleringarna mellan Jönköping-Linköping, $\mu = 0.3$

De slutsater som kan dras är att MPC-regulatorn är robust och ligger i topp vid alla simuleringar. Den är i snitt 2.6 % bättre än PI-regulatorn. IP-regulatorn är en god idé som står sig bra jämfört med MPC-regulatorn. Med tanke på att IP-regulatorn är en PI-regulator som tar ”smarta” beslut vid upp- och nedförsbackar fungerar den riktigt bra. PI-regulatorn tar för lång tid för att den ska vara konkurrenskraftig.

5.2 Undersökning av P- och I-delen

Det uppstod underliga fenomen när friktionskoefficienten sattes till $\mu = 0.3$ i PI- och IP-regulatorn. Gaspedalen började oscillera när farthållaren försökte hålla jämn hastighet. För att undersöka detta närmare ska ett par simuleringar försöka reda ut varför. De simuleringar som ska göras är med 10 % mindre P-, I- och PI-del samt med 10 % mer P-, I- och PI-del. Friktionskoefficienten är satt till 0.3 och sträckan som simuleras är Linköping-Jönköping.

5.2.1 10 % mindre P-, I- och PI-del

Se tabell 5.8.

Som vi kan se i tabellen minskar bränsleförbrukningen för mindre P-del, dvs vi gör systemet långsammare vid förändringar. Vid minskad I-del blev förbättringen knappt märkbar, i stort sett samma resultat som för ursprungsvärdena. Om vi istället minskar I-delen till en tiondel av ursprungsvärdet minskade PI och IP80 medan IP70 ökade marginellt, mindre än ± 0.01 l/mil för varje regulator. Då vi minskar både P- och I-delen samtidigt fås en liten förbättring jämfört med om vi bara skulle öka P-delen. Alltså är det P-delen som orsakar större delen av ökningen i bränsleförbrukningen. Detta innebär att det finns ett samband mellan inställningarna i PI-regulatorn och friktionskoefficienten, vilket inte är fallet för MPC-regulatorn.

För att göra en snabb undersökning gjordes en simulering med 20 % mindre P-del. Resultatet finns att beskåda längst ner i tabell 5.8. Nu blev resultaten bättre än för $\mu = 0.5$. Alltså finns det tid och framförallt bränsle att spara genom att skruva lite på parametrarna till PI-regulatorn.

	Regulator	Bränsleförbrukning [l/mil]	Tid [s]
10 % mindre P-del	PI	3.7055	5414
	IP70	3.6378	5434
	IP80	3.6671	5414
10 % mindre I-del	PI	3.8618	5415
	IP70	3.7742	5433
	IP80	3.8168	5414
10 % mindre PI-del	PI	3.7040	5413
	IP70	3.6236	5433
	IP80	3.6606	5413
20 % mindre P-del	PI	3.6914	5414
	IP70	3.6166	5434
	IP80	3.6538	5414

Tabell 5.8: Resultat från simuleringarna med 10 % mindre P-, I- och PI-del samt 20 % mindre P-del mellan Linköping-Jönköping, $\mu = 0.3$

5.2.2 10 % mer P-, I- och PI-del

Se tabell 5.9.

Som sig bör ökar bränsleförbrukningen då vi ökar P-delen, helt enligt reglerteorins resonemang kring PI-regulatorer [1]. Det som är intressant är att då vi ökar I-delen minskas bränsleförbrukningen och med en minskning av P-delen samtidigt borde ge bra resultat. Vid simuleringar med en minskning av P-delen med 20 % och en ökning av I-delen med 20 % fick lastbilen en bränsleförbrukning på 3.6504 l/mil och det tog 5415 s för IP80, 3.6148 l/mil och det tog 5435 s för IP70 och för PI blev det 3.6952 l/mil på 5414 s. Bättre än för friktionskoefficienten $\mu = 0.5$ (!), om man ser till bränsleförbrukningen. Dock tog det längre tid vilket kostar bränsle att köra in. Men med tanke på vad lastbilen drog första simuleringstillfället med ursprungsvärdena på P och I, så är det betydligt bättre.

	Regulator	Bränsleförbrukning [l/mil]	Tid [s]
10 % mer P-del	PI	3.9374	5415
	IP70	3.8487	5434
	IP80	3.8885	5415
10 % mer I-del	PI	3.8614	5415
	IP70	3.7750	5432
	IP80	3.8162	5415
10 % mer PI-del	PI	3.9381	5416
	IP70	3.8384	5434
	IP80	3.8799	5415

Tabell 5.9: Resultat från simuleringarna med 10 % mer P-, I- och PI-del mellan Linköping-Jönköping, $\mu = 0.3$

5.2.3 Jönköping-Linköping med nya P- och I-delen

Eftersom simuleringarna mellan Jönköping-Linköping med $\mu = 0.3$ blev dåliga, gjordes en simulering med de nya parametrarna på PI-regulatorn. Resultaten kan ses i tabell 5.10. En förbättring men inte så bra som förväntat. Kan höjdprofilen ha nånting med saken att göra? Det är en stigning totalt sett mellan Linköping-Jönköping och tvärtom andra hållet.

Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.3029	5480
IP70	3.3032	5448
IP80	3.3055	5441

Tabell 5.10: Resultat från simuleringarna med 20 % mindre P-del och 20 % mer I-del mellan Jönköping-Linköping, $\mu = 0.3$

För att undersöka detta gjordes en simulering med 20 % mindre P- och I-del, alltså minskar på I-delen istället för att öka. Samma väg och friktionskoefficient. Resultatet finns i tabell 5.11. Enligt resultaten har även höjdprofilen inverkan på inställningarna för PI-regulatorn. Nu drar lastbilen mindre bränsle än vad den gjorde för de andra friktionskoefficienterna. Det finns helt klart ett samband mellan PI-inställningarna, friktionskoefficienten och höjdprofilen.

Regulator	Bränsleförbrukning [l/mil]	Tid [s]
PI	3.2917	5482
IP70	3.2915	5447
IP80	3.2959	5441

Tabell 5.11: Resultat från simuleringarna med 20 % mindre P- och I-del mellan Jönköping-Linköping, $\mu = 0.3$

Kapitel 6

Slutsatser

Vid modelleringen av lastbilsmodellen uppstod inte direkt några komplikationer. Det som saknades var en ordentlig validering av parametrarna. Motorn är den delen i modellen som kommer från en riktig lastbil. Den är uppmätt i stationära punkter från en körning med en Scania-lastbil. Det optimala hade varit att få parametrar även för andra detaljer från en riktig körning med en lastbil men med de givna förutsättningarna fanns inte detta alternativet. Med [10] och [2] löstes det mesta. Som validering kontrollerades fordonets lutning kontra vägens lutning, normalkrafter, fordonets hastighet kontra visarens hastighet, slip m.m. Som kvitto på att det fungerade gjordes körningar i simulatorm. Det som undersöktes var främst känsla, accelerationer och beteende vid inbromsning och sväng.

När allt var på sin plats undersöktes de tre regulatorerna. På sträckan Linköping-Jönköping var MPC-regulatorn bäst. Den drog i snitt 2.7 % mindre bränsle än PI-regulatorn och 1.6 % mindre bränsle än IP80-regulatorn. IP70-regulatorn ligger för långt efter tidsmässigt så den faller på den punkten. Om vi istället tittar på PI-regulatorn med andra inställningar för $\mu = 0.3$ drog MPC-regulatorn i snitt 1.3 % mindre bränsle än PI-regulatorn och bara 0.2 % (!) mindre bränsle än IP80-regulatorn. Jämför vi komplexiteten bakom reglerna är det anmärkningsvärt hur bra IP-regleringen fungerar. IP-regleringen är trots allt bara en vanlig PI-regulator som tar ”smarta” beslut vid upp- och nedförsbackar med tillräcklig lutning för att kunna utnyttja dem.

Vid simuleringar motsatta hållet, dvs Jönköping-Linköping, var MPC-regulatorn också bäst. I snitt drog den 2.4 % mindre bränsle än PI-regulatorn, 2.3 % mindre bränsle än IP70-regulatorn och 2.5 % mindre bränsle än IP80-regulatorn. PI-regulatorn ligger för långt efter tidsmässigt, vilket inte är acceptabelt. IP70-regulatorn håller samma tider men IP80-regulatorn är 7-8 s snabbare än MPC-regulatorn, trots att den inte drar mycket mer bränsle än vad IP70-regulatorn gör. Om vi istället använder PI-regulatorn med andra inställningar för $\mu = 0.3$ drog MPC-regulatorn i snitt bara 0.6 % mindre bränsle än PI- och IP80-regulatorn och 0.5 % mindre bränsle än IP70-regulatorn. PI-regulatorn tar för lång tid men IP70- och IP80-regulatorn har samma och bättre tidsåtgång.

Med dessa resultat i bakgrunden dras slutsatsen att MPC-regulatorn är bäst och mest robust. Antagandet om rullvillkor som MPC-regulatorn är uppbyggd kring står sig bra, även när däcksdynamiken och hjulupphängningen kommer med i ekvationerna. IP-regulatorn hävdar sig mycket bra. Den ligger inte långt efter MPC-regulatorn och med tanke på den enkla reglering som ligger bakom är det en intressant idé. Hade varit intressant att se hur bra den blivit om den tagit hänsyn till friktionskoefficienten och höjdprofilen med tanke på inställningar för PI-regulatorn. PI-regulatorn fungerar som bäst då det inte är för mycket branta och långa uppförsbackar, för då tappar den tidsmässigt gentemot de andra regulatorerna. I jämförelse med en mänsklig förare står sig PI-regulatorn riktigt bra. Vid körning i simulatoren mellan Linköping-Jönköping med mänsklig förare drog lastbilen 4.3841 l/mil och det tog 5079 s att köra 122 km jämfört med PI-regulatorn som drar 3.8106 l/mil på 5089 s och samma sträcka (fick sätta referenshastigheten till 87 km/h för att komma ner i samma tid). Alltså 5.7 dl/mil mindre bränsle.

Kapitel 7

Framtida förbättringar

För att underlätta skapandet av vägprofiler i simulatören och andra saker som görs innan simulering, skulle det vara användbart med någon form av GUI. GUI innebär en miljö där användaren har allting framför sig och det enda han/hon gör är att ställa in vägval, vilken regulator, vikt på fordonet, längd, m.m. Som det är nu krävs vissa program om användaren vill ändra på parametrar eller byta miljö.

För att förbättra körkänslan och det allmänna intrycket behövs en rejäl omgång med den visuella biten. Ett examensarbete som bara innefattar att skapa en simuleringsmiljö är nog vad som krävs.

Det fordon som använts i detta examensarbete är en lastbil utan släp. Det skulle vara intressant att modellera ett släp för att se hur det beter sig vid olika manövrar. Även en trailerdragare hade varit intressant.

Det skulle vara intressant att bygga vidare på IP-regulatorns idé och försöka modellera in friktionskoefficienten och höjdprofilen i regleringen för att se om det går förbättra regleringen ytterligare.

Litteraturförteckning

- [1] Glad, T. och Ljung, L. (1989) *Reglerteknik, Grundläggande teori*. Studentlitteratur, Lund
- [2] Hellström, E. (2005) *Explicit use of road topography for model predictive cruise control in heavy truck*. Examensarbete, ISY, Linköping
- [3] Jacobsson, T. (2005) *Polering av asfaltbeläggningar* Publikation hos VTI. <http://www.vti.se/EPiBrowser/Publikationer/N12-2005.pdf>
- [4] Nielsen, L. och Eriksson, L. (2004) *Course material Vehicular Systems*. Fordonssystem, ISY, Linköping
- [5] Pacejka, Hans B. (2002) *Tire and vehicle dynamics* Butterworth-Heinemann.
- [6] Wingren, A. (2005) *Fordonsreglering med framförhållning*. Examensarbete, ISY, Linköping
- [7] Fordonssimulators officiella hemsida. (2005)
<http://www.vehicular.isy.liu.se/Edu/Courses/TSFS07/Projects05/nigthrider/index.html>
- [8] Naval Postgraduate Schools Open Scene Graph Tutorials.
<http://openscenegraph.org/documentation/NPSTutorials/index.html>
- [9] The Mathworks. Matlab dokumentation.
<http://www.mathworks.com/access/helpdesk/help/helpdesk.shtml>
- [10] Scania lastbilsdokumentation.
http://www.scania.se/Images/DB4x2%20AB%20PR%20OKT%202005_tcm136-66006.pdf
- [11] Visual Studio Dokumentation.
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnanchor/html/vsnetanchor.asp>

Bilaga A

Fordonsparametrar

$T = 0.01$ Simuleringstid [s]
 $rps2rpm = 60$ Omvandlingsfaktor

Växlare

$acc_releasetime = 0.2$ Gaspedalens uppsläppningstid [s]
 $acc_presstime = 0.7$ Gaspedalens nedtryckningsstid [s]
 $clutch_presstime = 0.2$ Kopplingspedalens nedtryckningstid [s]
 $clutch_releasetime = 0.5$ Kopplingspedalens uppsläppningstid [s]
 $total_shifttime = 1.0$ Total tid för växlingsförloppet [s]
 $init_gear = 0$ Startväxel

Orientering

$g = 9.81$ Gravitation [m/s^2]

$r = 0.52$ Hjulradie [m]

$m_c = 39000$ Chassits massa [kg]

$m_m = 1000$ Motorns massa [kg]

$B = 2.5$ Bredd mellan hjulen [m]

$H = 3.5$ Plattans höjd [m]

$L = 9$ Plattans längd [m]

$$I_G_platta = \begin{pmatrix} \frac{m_c \cdot (B^2 + H^2)}{12} & 0 & 0 \\ 0 & \frac{m_c \cdot (L^2 + H^2)}{12} & 0 \\ 0 & 0 & \frac{m_c \cdot (L^2 + B^2)}{12} \end{pmatrix} \quad \begin{array}{l} \text{Tröghetsmatrix} \\ \text{karossen} \end{array}$$

$$md_2 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{m_c \cdot m_m}{(m_c + m_m) \cdot \frac{L^2}{4}} & 0 \\ 0 & 0 & \frac{m_c \cdot m_m}{(m_c + m_m) \cdot \frac{L^2}{4}} \end{pmatrix} \quad \text{Tröghetsmatrix motor}$$

$I_G_car = I_G_platta + md_2$ Tröghetsmatrix fordon

$inv_I_G = inv(I_G_car)$ Tröghetsmatrix fordon inverterad

$m = m_c + m_m$ Bilens massa [kg]

$$\begin{aligned}
 a &= \frac{m_c}{(m_c + m_m) \cdot \frac{L}{2}} && \text{Längd från tyngdpunkten till fronten [m]} \\
 b &= L - a && \text{Längd från tyngdpunkten till baken [m]} \\
 c &= \frac{B}{2} && \text{Längd från tyngdpunkten till sidan [m]} \\
 d &= \frac{H}{2} && \text{Längd från tyngdpunkten till bottenplattan [m]} \\
 Z_0 &= 0.5 && \text{Höjd från marken till fjädern [m]} \\
 ac &= \sqrt{(a - 1.5)^2 + c^2} && \text{Längd från tyngdpunkten till framhjulen [m]} \\
 bc &= \sqrt{(b - 2)^2 + c^2} && \text{Längd från tyngdpunkt till bakhjulen [m]} \\
 sr &= 10 && \text{Svängradie [m]} \\
 maxratt &= \frac{ac + bc}{sr} && \text{Maximal hjulvinkel [rad]}
 \end{aligned}$$

Pacejka

$$\begin{aligned}
 brus &= 0.001 && \text{Varians på brus i } \mu \\
 F_ff &= 1/1500 && \text{Normering av force-feedback kraften} \\
 B_c &= 8 && \text{Parametrar för slipkurvan i lateral led} \\
 B_l &= 2.5 && \text{Parametrar för slipkurvan i longitudinell led} \\
 B_M &= 10 && \text{Parametrar för slipkurvan som genererar force-feedback} \\
 C_c &= 1.5 \\
 C_l &= 1.5 \\
 C_M &= 2 \\
 E_c &= -10 \\
 E_l &= -19 \\
 E_M &= -14 \\
 magicx &= [-1:0.01:1] && \text{x-värden} \\
 magicy_c &= \sin(C_c \cdot \arctan(B_c \cdot magicx - E_c \cdot (B_c \cdot magicx - \arctan(B_c \cdot magicx)))) && \text{y-värden lateral led} \\
 magicy_l &= \sin(C_l \cdot \arctan(B_l \cdot magicx - E_l \cdot (B_l \cdot magicx - \arctan(B_l \cdot magicx)))) && \text{y-värden longitudinell led} \\
 magicy_M &= 0.2 \cdot \sin(C_M \cdot \arctan(B_M \cdot magicx - E_M \cdot (B_M \cdot magicx - \arctan(B_M \cdot magicx)))) && \text{y-värden force feedback}
 \end{aligned}$$

Hjulupphängning

$$\begin{aligned}
 k_spring &= 300000 && \text{fjäderkonstant} \\
 c_damper &= 3000000 && \text{dämpkoefficient} \\
 z_max &= 0.3 && \text{maximal ihoptryckning [m]} \\
 z_dot_max &= 15 && \text{maximal hoptryckningshastighet [m/s]}
 \end{aligned}$$

Hjulhastighet

$$\begin{aligned}
 c_w &= 0.6 && \text{Luftmotståndskoefficient} \\
 A_a &= 10 && \text{Maximal tvärsnittsarea för fordonet [m}^2\text{]} \\
 Rho_a &= 1.29 && \text{Luftdensitet [kg/m}^3\text{]} \\
 J_w &= 32.9 && \text{Hjulens tröghetsmoment [kgm}^2\text{]} \\
 c_r &= 7 \cdot 10^{-3} && \text{Rullmotståndskoefficient}
 \end{aligned}$$

Copyright

Svenska

Detta dokument hålls tillgängligt på Internet - eller dess framtida ersättare - under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för icke-kommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida: <http://www.ep.liu.se/>

English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: <http://www.ep.liu.se/>