



# CHALMERS

CHALMERS TEKNISKA HÖGSKOLA

INSTITUTIONEN FÖR SIGNALER OCH SYSTEM

SLUTRAPPORT SSYX02-15-19

---

## Fordonssimuleringsmiljö för vägplanering och manöverering

---

*Författare:*

Muhamed FARAJ  
Sanna ÄLEGÅRD  
Pontus KIELEN

*Handledare:*

Nikolce MURGOVSKI

Göteborg, Sverige 2015

## Sammanfattning

I denna rapport beskrivs utvecklingen av ett regelsystem, där syftet är att autonomt reglera ett fordon till att följa en given referenslinje med en given referenshastighet. I detta projekt genomförs endast simuleringar av regelsystemet men syftet är att produkten ska kunna vidareutvecklas och implementeras i verkliga fordon i framtiden. Många liknande projekt har utförts tidigare ur vilka det funnits mycket inspiration och information att hämta. Reglersystemet i detta projekt har dock byggts från grunden och visat sig ge goda resultat med tanke på den grundläggande nivån. Fordonsmodellen som har utvecklats kan följa en rad varierande referenslinjer utan att avvika avsevärt. I rapporten presenteras en bra grund för förståelse om utmaningar vid utveckling av ett regelsystem och en verklighetsbeskrivande modell. Det är också en bra grund för vidare utveckling och implementering i verkliga regelsystem för autonoma fordon.

# Innehåll

<b>1</b>	<b>Inledning</b>	<b>1</b>
1.1	Syfte . . . . .	2
1.2	Avgränsningar . . . . .	2
1.3	Metod . . . . .	2
<b>2</b>	<b>Koordinatsystem</b>	<b>4</b>
<b>3</b>	<b>Referenslinje</b>	<b>5</b>
3.1	Design av referenslinje . . . . .	5
3.2	Verifiering av referenslinje . . . . .	7
3.2.1	Verifiering av sträcka . . . . .	8
3.2.2	Verifiering av kurvatur . . . . .	9
<b>4</b>	<b>Fordonsmodell</b>	<b>9</b>
4.1	Punktmassmodell . . . . .	9
4.2	Verifiering av fordonsmodell . . . . .	11
<b>5</b>	<b>Förarmodell</b>	<b>12</b>
5.1	Krav och önskemål på förarmodell . . . . .	13
5.2	Design av förarmodell . . . . .	13
5.2.1	Skalfaktor och beräkning av sträcka . . . . .	13
5.2.2	Förhandsgranskning . . . . .	16
5.2.3	Positionsreglering . . . . .	17
5.2.4	Hastighetsreglering . . . . .	20
<b>6</b>	<b>Simulering och uppbyggnad av systemet</b>	<b>21</b>
6.1	Översikt över reglersystemet i Simulink . . . . .	21
6.2	Fordonsmodell i Simulink . . . . .	22
6.3	Förarmodell i Simulink . . . . .	23
<b>7</b>	<b>Visualisering i grafiskt användargränssnitt</b>	<b>25</b>
<b>8</b>	<b>Resultat och diskussion</b>	<b>27</b>
8.1	Simulering . . . . .	27
8.1.1	Friktionens och massans påverkan på simuleringsresultatet . . . . .	28

8.1.2	Analys av data vid olika inställningar på friktion och massa . . . . .	30
8.2	Simulering med brus . . . . .	31
8.3	Diskussion om reglersystemet . . . . .	34
<b>9</b>	<b>Slutsatser och vidare arbete</b>	<b>35</b>
<b>Bilagor</b>		
<b>A</b>	<b>Fordonsmodell i Simulink</b>	<b>38</b>
<b>B</b>	<b>Förarmodell i Simulink</b>	<b>39</b>
B.1	Beräkning av positions- och vinkelfel i Simulink . . . . .	39
B.2	Beräkning av referenshastighet i Simulink . . . . .	40
B.3	PID-kontroller i Simulink . . . . .	41
<b>C</b>	<b>Simuleringsresultat</b>	<b>42</b>
<b>D</b>	<b>Hastighet vid olika friktion</b>	<b>44</b>

# 1 Inledning

Runtom i världen ökar användningen av och intresset för autonoma fordon, det vill säga fordon som kör och manövrerar utan en människas hjälp. Dessa fordon kan öka människors säkerhet och bekvämlighet. Det blir till exempel möjligt att under färden arbeta, vila, läsa eller göra andra sysslor som tidigare var för osäkert. Ett helt nytt beteendemönster och livsstil kan alltså komma att utvecklas som följd av dessa fordon.

Det finns potential att genom autonoma fordon optimera energiförbrukning, minska åktiden samt få en säkrare trafik (1). Säkerheten ökas till exempel genom en minskad inverkan av mänskliga faktorer såsom trötthet och reaktionstid. En minskad energiförbrukning inom trafiksektorn leder till mindre utsläpp vilket är intressant ur både miljöperspektiv och hälsoperspektiv. Delvis autonoma fordon finns ute på marknaden. Dessa fordon assisterar föraren och kan under vissa förutsättningar utföra bestämda kontrolluppgifter. Exempel på system som finns på marknaden idag är: nödbromssystem, parkeringshjälp, system för att hindra fordonet från att köra över på fel fil och adaptiv farthållare (2). I framtiden kommer mer avancerade uppgifter kunna utföras och framförallt en kombination av kontrolluppgifter samtidigt.

Den snabba utvecklingen av autonoma fordon har också negativa sidor och många säkerhetsfrågor måste lösas innan tekniken kan börja användas kommersiellt. I alla system uppstår fel och det är något som måste hanteras på ett smidigt sätt, antingen genom att systemet själv utför någon sorts säkerhetsåtgärd eller genom att uppmärksamma passagerare om att det har uppstått ett problem och vad som måste göras (3). Man måste också ha en väldigt hög säkerhet på systemet så att ingen obehörig kan tränga sig in. Trots detta kommer det med nuvarande utvecklingstakt inte ta lång tid innan autonoma fordon är säkrare än förarstyrda fordon (4).

System utvecklade för att reglera autonoma fordon måste testas på olika sätt innan det släpps på marknaden. Testerna måste utföras under många olika förhållanden, såsom olika väglag, luftmotstånd och fordonsmodeller, för att ge ett så givande resultat som möjligt. Den mest kostnads- och tidseffektiva strategin är att simulera dessa tester, innan fysiska tester görs. Simuleringar i en välutvecklad och användarvänlig simuleringsmiljö gör det även enkelt att testa olika funktioner, ändra parametrar samt jämföra olika resultat.

## 1.1 Syfte

Syftet med projektet är att utveckla en punktmassmodell av ett fordon samt reglera denna till att följa en förutbestämd bana samtidigt som den håller en given referenshastighet. Målet är att fordonet ska kunna följa banan i så hög hastighet som möjligt utan att glida med hjälp av ett reglersystem. För testning av modellerna och reglersystemet samt visualisering av resultaten ska även en enkel simuleringsmiljö med tillhörande användargränssnitt utvecklas. Där ska modellen simuleras på olika typer av vägbanor och användaren ska kunna ställa in parametrar som beskriver fordonet och dess omgivning.

## 1.2 Avgränsningar

En modell av verkligheten är alltid mer eller mindre förenklad. I detta projekt används en enkel modell av ett fordon; en punktmassmodell. Denna tar ej hänsyn till alla krafter som påverkar alla fyra hjul. Användning av enkla modeller som denna kan dock fortfarande ge goda resultat och är mindre tidskrävande än mer avancerade modeller.

Även förenklingar av verkligheten runtom fordonet har gjorts. I verkligheten ska ett fordon kunna köra under många olika förhållanden, med många olika parametrar som påverkar. Inom ramarna för projektet anses det dock inte rimligt att i modellen ha med alla dessa parametrar. En avvägning har därför gjorts, angående vilka parametrar som är mest relevanta för projektet. Ett exempel är att inga höjdskillnader på vägen antas.

Endast simuleringar av modellen har utförts, det vill säga inga fysiska tester. Fysiska tester anses var ett eget delprojekt som kräver mer tid än detta projekt avser. Målet är dock att modellen i framtiden ska kunna implementeras i ett verkligt fordon.

## 1.3 Metod

En stor del av projektet har bestått av att ställa upp ekvationer som beskriver reglersystemet som består av tre huvudsakliga delar. Dessa är en fordonsmodell, en förarmodell och en referenslinje. Fordonsmodellen består av differentialekvationer som beskriver ett fordonens rörelse, beroende på vilka krafter som påverkar fordonet. Förarmodellen ska agera på sätt och vis som en riktig förare i ett fordon och undersöka vägen framför fordonet samt reglera så att fordonet håller sig på referenslinjen med given referenshastighet.

Datorprogrammet Matlab, som är ett välkänt program för att utföra matematiska beräkningar, har använts vid uppbyggnad av referenslinjen medan Simulink har använts vid uppbyggnad fordonsmodellen och förarmodellen. Simulink är ett användarvänligt program innehållande verktyg för simulering av olika slag. Det går ut på att man skapar en modell, eller ett system, som byggs upp av block. Dessa hämtas från Simulink Library och innehåller en funktion av något slag (till exempel en matematisk funktion). Till varje block skickas en eller flera insignaler, som appliceras på funktionen i blocket. Ut ur blocket kommer sedan önskad utsignal. Simulink kan även integreras med Matlab-funktioner vilket har visat sig vara praktiskt i detta projekt. Utöver detta har ett enkelt användargränssnitt skapats för att visualisera resultaten. Detta har skapats med hjälp av Matlabs verktyg GUIDE. Analys av resultat har skett genom att studera insamlad data från simuleringar. Exempel på data att analysera har varit plottar av fordonets färdväg, maximalt reglerfel och medelvärde av reglerfelet.

För att förenkla arbetsprocessen och tydliggöra vad som måste uppfyllas ställdes i ett tidigt stadie av projektet ett antal krav på systemet. Utöver dessaa ställdes ett antal delproblem upp för att lösa kraven.

### **Kraven som systemet ska uppfylla är**

- Fordonsmodellen ska kunna följa en given referenslinje.
- Fordonsmodellen ska kunna följa en given referenshastighet.
- Simuleringsmiljön ska visa fordonets färdväg jämfört med referenslinjen.

### **De olika delproblemen som har lösts för att uppfylla kraven är**

- Beskriva lokalt och globalt koordinatsystem
- Skapa referenslinje
- Utveckla fordonsmodell
- Utveckla förarmodell och regulatorer
- Integrera modellerna samt simulera
- Visualisera i grafiskt användargränssnitt

## 2 Koordinatsystem

Fordonet är implementerat som en punktmassmodell vilket beskrivs i detalj i Kapitel 4.1. Därför behövs två koordinatsystem, ett globalt för hela systemet och ett lokalt för punktmassan. Det globala har positiva x-axeln uppåt och positiva y-axeln åt höger. Det lokala systemet har origo i punkten där fordonet befinner sig, x-axel längs med och y-axel vinkelrätt mot fordons riktning (se Figur 1). Detta innebär att det lokala koordinatsystemet är transformerat först genom translation och sedan genom rotation. Systemet är roterat med vinkeln  $\psi$  i medurs riktning.

Translation innebär att koordinatsystemet är förflyttat i x- och y-led jämfört med det globala systemet. De globala koordinaterna  $x$  och  $y$  omvandlas till lokala koordinater  $\tilde{x}$  och  $\tilde{y}$  enligt

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} x_{\text{bil}} \\ y_{\text{bil}} \end{bmatrix}, \quad (1)$$

där de globala koordinaterna  $(x_{\text{bil}}, y_{\text{bil}})$  är origo för det lokala systemet.

Punkten  $(\tilde{x}, \tilde{y})$  roteras i sin tur med vinkel  $\psi$  i det lokala koordinatsystemet genom att använda rotationsmatrisen

$$R(-\psi) = \begin{bmatrix} \cos(\psi) & \sin(\psi) \\ -\sin(\psi) & \cos(\psi) \end{bmatrix}. \quad (2)$$

En kombination av (1) och (2) ger

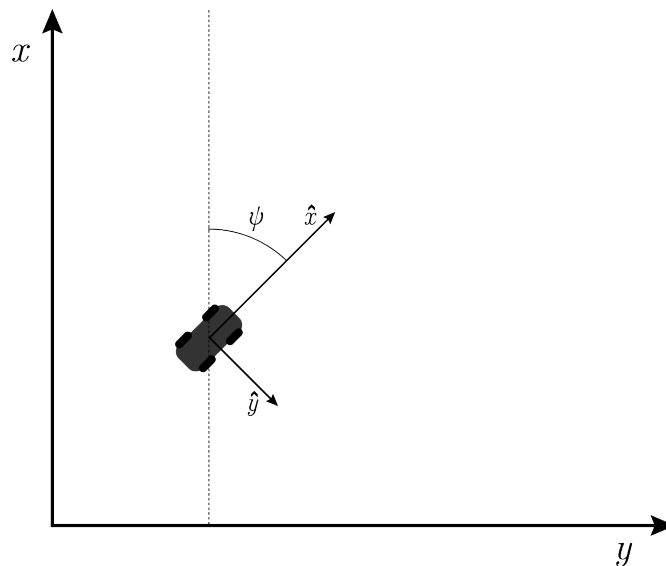
$$\begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = R(-\psi) \begin{bmatrix} \tilde{x} \\ \tilde{y} \end{bmatrix}. \quad (3)$$

Det direkta sambandet mellan koordinater från det globala systemet  $(x, y)$  till det lokala systemet  $(\hat{x}, \hat{y})$  fås då enligt

$$\hat{x} = (x - x_{\text{bil}})\cos(\psi) + (y - y_{\text{bil}})\sin(\psi) \quad (4)$$

$$\hat{y} = (y - y_{\text{bil}})\cos(\psi) - (x - x_{\text{bil}})\sin(\psi). \quad (5)$$





Figur 1: Illustration av globalt samt lokalt koordinatsystem.

### 3 Referenslinje

Referenslinjen anger vilken väg fordonet ska följa. Linjen beskrivs av en matris innehållande nödvändiga parametrar för att beskriva vägen samt information nödvändig för att fordonet ska kunna följa den.

#### 3.1 Design av referenslinje

Referenslinjen skapas med hjälp av funktioner i Matlab och sparas som en matris

$$T = [ s \ k \ \psi_{\text{ref}} \ x \ y ]. \quad (6)$$

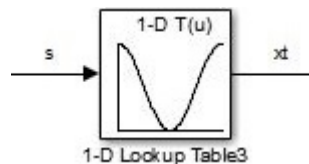
Matrisen innehåller fem tillståndsvariabler: Sträcka, kurvatur, tangentvinkel samt position i x- och y-led. Variablerna sparas i kolumner och varierande antal rader beroende på hur många punkter användaren väljer att beskriva referenslinjen med.

#### Sträcka på referenslinjen och interpolation

Sträckan som färdats längs referenslinjen är benämnd  $s$  och är en kolumn i

matrisen (6). Varje element i  $s$  anger sträckan från startpunkt till alla övriga punkter på referenslinjen. Det sista värdet i kolumnen  $s$  anger alltså hela referenslinjens längd. Denna kolumn gör det möjligt att beskriva övriga parametrar i (6) som en funktion av  $s$ . Detta sker med hjälp av interpolation. Interpolation innebär att man från en mängd diskreta värden beräknar funktionsvärden mellan dessa kända värden.

I (6) finns ett antal punkter beskrivna. Dessa värden är kända för varje värde på  $s$  upp till referenslinjens längd. I Simulink används blocket LookUp-table för att interpolera en funktion. Ett värde på  $s$  skickas in som insignal och i blocket ställs det in vilken kolumn av (6) som är önskad utsignal. Om insignalen ligger mellan två kända värden utför blocket en linjär approximation för att hitta ett insignal att interpolera med (se Figur 2).



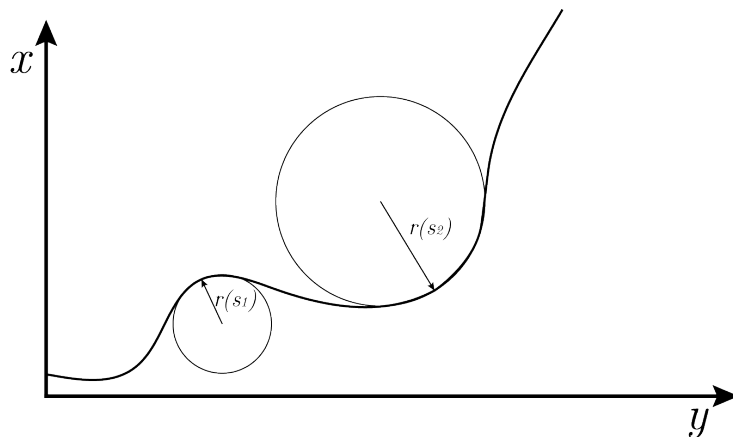
Figur 2: Exempel där insignal  $s$  interpoleras och utsignal är ett motsvarande värde på  $x$ .

### Kurvatur

Kurvatur  $k$  anger hur skarpt en kurva svänger. Kurvaturen för en rak linje bör alltså vara 0 [1/m] och för en cirkel bör kurvaturen vara konstant och bero på radien. En stor radie ger minskad kurvatur och tvärtom. Kurvaturen beror på radien, se Figur 3, enligt

$$k = \frac{1}{r} \quad (7)$$

där  $r$  är radien på den cirkel som närmast beskriver kurvans krökning i varje punkt (5). Tecknet på kurvaturen beror på i vilken riktning tangentvektorn för kurvan roterar längs med kurvan. Om vektorn roterar moturs så är  $k$  positiv och om den roterar medurs så är  $k$  negativ. Detta innebär att  $k$  för en kurva har samma tecken som andraderivatet av kurvan. Tecknet kommer dock bero på hur användaren väljer att parametrisera kurvan. Val av parametrisering innebär val av riktning på kurvan.



Figur 3: Exempel på hur radien  $r$  beror på kurvans krökning.

För godtyckliga kurvor parametriserade med kartesiska koordinater  $(x(t), y(t))$  definieras kurvans kurvatur enligt

$$k = \frac{\dot{x}\ddot{y} - \dot{y}\ddot{x}}{(\dot{x}^2 + \dot{y}^2)^{3/2}} \quad (8)$$

där derivator är med avseende på tid (5). Ekvation (8) används i Matlab för att räkna ut kurvaturen för varje punkt i matrisen  $T$ .

### Tangentvinkel

Tangentvinkel  $\psi_{\text{ref}}$  är vinkeln mellan x-axeln och referenslinjens tangent och beräknas trigonometriskt enligt

$$\psi_{\text{ref}} = \arctan \frac{\Delta Y}{\Delta X}, \quad (9)$$

där  $\Delta X = x_2 - x_1$  och  $\Delta Y = y_2 - y_1$ . Punkterna  $(x_1, y_1)$  och  $(x_2, y_2)$  är två efterföljande punkter på referenslinjen.

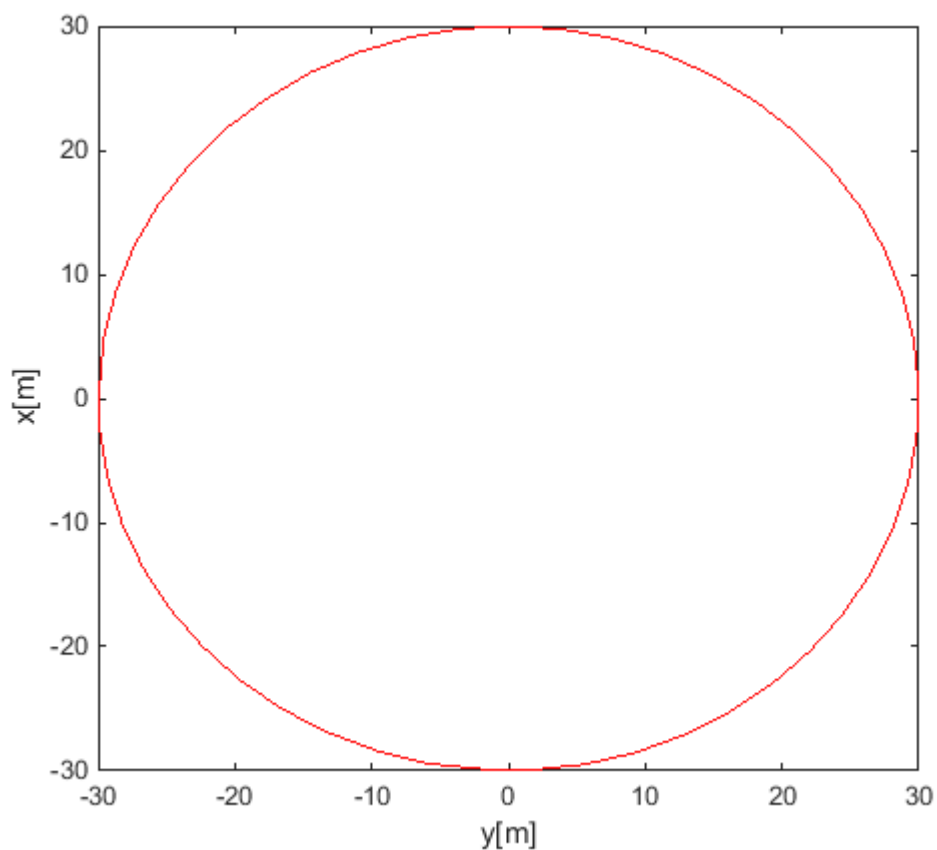
### Koordinater

Koordinaterna beskriver referenslinjens position i varje punkt. Punkterna längs referenslinjen är jämt fördelade över linjen med ett bestämt antal.

## 3.2 Verifiering av referenslinje

För enkla referenslinjer går värdena i matrisen  $T$  att räkna ut manuellt och på så sätt kan vi verifiera att Matlab-funktionerna stämmer. Exempel på en

enkel referenslinje är en cirkelformad sådan. Där förväntas en konstant kurvatur enligt (7) och en slutgiltig sträcka  $s$  som är lika med cirkelns omkrets.



Figur 4: Figuren visar punkterna utsatta på en cirkelformad referenslinje med radien  $r = 30$  [m].

### 3.2.1 Verifiering av sträcka

Omkretsen  $O$  för en cirkel ges av

$$O = 2\pi r, \quad (10)$$

där  $r$  är radien för cirkeln. För  $r = 30$  [m] (se Figur 4) fås  $O = 188.5$  [m]. Detta ska alltså motsvara sträckan  $s$  för sista punkten på referenslinjen. Detta

erhålls från matrisen  $T$  och där hittas  $s = 188.42655$  [m], vilket anses stämma tillräckligt bra överens med teorin. Detta verifierar att Matlab-funktionen för att räkna ut sträckan stämmer.

### 3.2.2 Verifiering av kurvatur

Kurvaturen för en cirkel ges enligt (7). Med exemplet i Figur 4 förväntas  $k = 0.033$  [1/m].

Med exemplet i Figur 4 plottad med 210 punkter fås medelvärdet av vektorn  $k$  i Matlab till 0.0328 [m], vilket anses stämma bra överens med teorin. Även för färre och fler punkter plottade stämmer  $k$  väl överens med teorin. Detta anses verifiera att funktionen för att beräkna kurvaturen stämmer.

## 4 Fordonsmodell

I detta projekt används endast en enkel punktmassmodell av ett fordon. En punktmassmodell kan ses som en enda punkt som är positionerad i fordonets tyngdpunkt. Punkten kan röra sig framåt och färdriktningen kan endast påverkas med en vinkelhastighet. Man kan beskriva rörelsen som ett hjuls rörelse. Utifrån detta kan en enkel modell ställas upp som beskriver hur fordonet beter sig när den utsätts för olika krafter. Regulatorn i systemet styr dels hur mycket fordonet ska svänga men också hastigheten. Insignalerna till fordonsmodellen består av en accelerationskraft samt förändring av svängningsradie (storhet kurvatur/tidsenhet).

### 4.1 Punktmassmodell

Punktmassmodellen innehåller fem tillståndsvariabler: hastighet och position i x- och y-riktning samt fordonets riktningsvinkel. Dessa benämns:  $V_{xbil}$ ,  $V_{ybil}$ ,  $\psi$ ,  $x_{bil}$ ,  $y_{bil}$ . Hastigheten  $V(t)$  i bilens riktning ges av Newtons andra lag,

$$V(t) = \frac{1}{m} \int_0^t F dt. \quad (11)$$

där  $F$  är summan av alla krafter som påverkar bilen longitudinellt och ges av

$$F = F_{acc} - F_{luft}. \quad (12)$$

$F_{\text{acc}}$  är accelerationskraften, som skickas som insignal till modellen, och  $F_{\text{luft}}$  är en bakåtdrivande kraft som uppstår på grund av luftmotstånd.  $F_{\text{luft}}$  beräknas enligt

$$F_{\text{luft}} = C_d V(t)^2, \quad (13)$$

där  $C_d$  är en luftmotståndskoefficienten som beror på fordonets form. Hastigheten delas upp i två komponenter i det globala koordinatsystemet, hastigheten i x-riktning i det globala koordinatsystemet,  $V_{\text{xbil}}$ , och hastigheten i y-riktning,  $V_{\text{ybil}}$ , där förhållandena

$$V(t) = [ V_{\text{xbil}}(t) \ V_{\text{ybil}}(t) ]^T \quad (14)$$

$$V_{\text{xbil}}(t) = |V(t)| \cos(\psi) \quad (15)$$

$$V_{\text{ybil}}(t) = |V(t)| \sin(\psi), \quad (16)$$

gäller. Fordonets position erhålls då från

$$x_{\text{bil}} = \int_0^T V_{\text{xbil}} dt \quad (17)$$

$$y_{\text{bil}} = \int_0^T V_{\text{ybil}} dt. \quad (18)$$

Vinkeln  $\psi$ , som motvarar vinkeln mellan hastighetsvektorn och den globala x-axeln, påverkas av den ena av fordonsmodellens insignaler. Denna insignal är i storheten kurvatur per tidsenhet och motsvarar hastigheten med vilken vi ändrar svängradien hos fordonet. För att fordonet inte ska svänga för skarpt har begränsningar satts på denna insignal. Begränsningarna är satta utifrån kravet att fordonet inte ska glida.

Då insignalen har storheten kurvatur per tidsenhet fås fordonets kurvatur  $k_{\text{bil}}$  genom att integrera insignalen

$$k_{\text{bil}} = \int_0^T k_{\text{in}} dt. \quad (19)$$

Fordonets riktningsvinkel  $\psi$  erhålls genom att först multiplicera kurvaturen  $k_{\text{bil}}$  med hastigheten  $V$  för att få ut vinkelhastigheten hos fordonet för att sedan integrera denne. Detta ger

$$\omega = V k_{\text{bil}} \quad (20)$$

och

$$\psi = \int_0^T \omega dt. \quad (21)$$

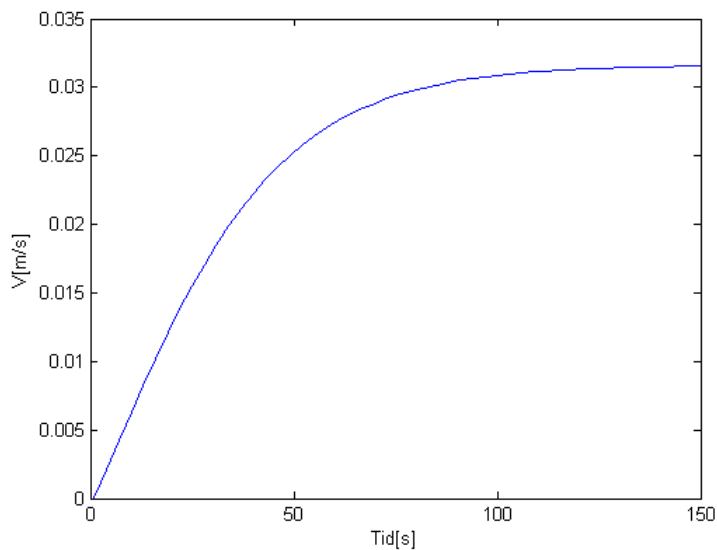
där  $\omega$  är fordonets vinkelhastighet.

Ekvationerna beskrivna i detta kapitel har implementerats i en Simulink-modell vilken i sin tur beskrivs i kapitel 6.2.

## 4.2 Verifiering av fordonsmodell

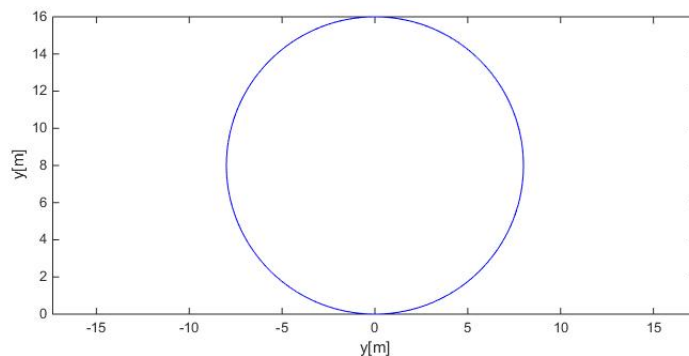
Verifiering av fordonsmodellens funktion har utförts genom en serie enkla tester där utgången går att förutspå. Resultatet på testerna jämförs sedan med förväntat resultat. Om resultatet överensstämmer med vad som är förväntat anses modellen uppfylla sitt syfte.

I Figur 5 visas resultatet från ett av testerna. Här har en konstant ökning av fordonets acceleration skickats in till modellen. Detta har till slut lett till att accelerationskraften och kraften från luftmotståndet tagit ut varandra och en konstant hastighet fås. Detta är också vad som förväntas hos ett verkligt fordon och alltså anses testet lyckat.



Figur 5: Resultatet från ett av testerna av fordonsmodellen.

För att verifiera en del av fordonmodellens funktionalitet kan man testa ifall följande samband för en punktmassa även gäller fordonmodellen: En punktmassa med vinkelhastigheten  $\omega$  och hastigheten  $v$  ger cirkelradien  $r$  på körbanan enligt  $r = v/\omega$  [m]. Med  $v = \pi$  [m/s] och  $\omega = \pi/8$  [rad/s] förväntas en cirkelformad körbana med radien  $r = \pi/(\pi/8) = 8$  [m], vilket överensstämmer med resultat som presenteras i Figur 6.



Figur 6: Fordonets färdväg då  $v = \pi$  [m/s] och  $\omega = \pi/8$  [rad/s] ger cirkelradien 8

Det är omöjligt att utvärdera hur bra en modell stämmer med verkligheten utan att jämföra med verkligheten. I detta projekt har ingen implementering i ett verkligt fordon skett. De enda tester som utförts är jämförelser mellan simulerat resultat och förväntat resultat, det kommer därför alltid finnas en osäkerhet kring hur bra modellen verkligen är.

## 5 Förarmodell

Uppgiften för förarmodellen är att hålla fordonet på en given referenslinje samt hålla fordonet så nära referenshastigheten som möjligt. Detta innebär att fordonets position och hastighet ska regleras så att dessa matchar referenslinjens. Referenshastigheten varierar längs banan och beräknas utifrån kravet att fordonet inte får glida. Vi är intresserade av positionsfelet längs fordonets lokala  $y$ -axel,  $\hat{y}$ , det vill säga positionsfel i sidled.



## 5.1 Krav och önskemål på förarmodell

Förarmodellen har flera olika uppgifter den ska utföra. Den ska dels agera som en regulator och alltså minska felet mellan fordonets tillstånd och referenslinjen. Detta innebär att fordonet ska hålla sig på både rätt position och hålla rätt hastighet. Förarmodellen ska också fungera på sätt och vis som en riktig förare i ett fordon genom att förhandsgranska vägen en bit framför fordonet och reglera därefter. En viktig uppgift för förarmodellen är också att för varje punkt fordonet befinner sig i, hitta motsvarande punkt på referenslinjen

## 5.2 Design av förarmodell

Ekvationerna som beskriver förarmodellen i detta kapitel togs fram i arbetet On minimum time vehicle manoeuvring: the theoretical optimal lap (6), men beskrivs i detalj för en ökad förståelse för hur modellen ser ut och hur den har framtagits.

Differentialekvationerna för en allmän fordonsmodell, som beskriver hastigheten med vilken modellens tillståndsvariabler förändras (med avseende på tid) definieras som

$$\frac{d\mathbf{x}}{dt} = \mathbf{g}(\mathbf{x}(t), \mathbf{u}(t), t) \quad \mathbf{x}(0) = \mathbf{x}_0 \quad t \in [0, T] \quad (22)$$

Där  $\mathbf{x}(t)$  är vektorn innehållandes fordonsmodellens tillståndsvariabler,  $\mathbf{u}(t)$  är en vektor med två element som returnerar kontrollsignaler i sidled (kontroll av styrvinkel) och i fordonets riktning (kontroll av hastighet).

Utöver dessa tillståndsvariabler har förarmodellen tillgång till information om referenslinjen. Denna information finns i matrisen  $T$  (6).

### 5.2.1 Skalfaktor och beräkning av sträcka

En skalfaktor  $Scf$  (23) är nödvändig för att kunna beskriva förändringshastigheten av systemets variabler med avseende på sträcka istället för med avseende på tid.

$$Scf = \frac{dt}{ds} \quad (23)$$

Detta görs för att lättare kunna veta hur långt fordonet kört på referenslinjen och vilka värden på referenslinjens tillståndsvariablerna som motsvarar denna

sträcka. Systemet kan då beskrivas med

$$\frac{dx}{ds} = Scf \cdot a(x(s), u(s), s) = \hat{a}(x(s), u(s), s), \quad (24)$$

$$x(0) = x_0, \quad s \in [0, S], \quad (25)$$

där  $u(s)=f(x(s),T(s),s)$ ,  $s$  är sträckan fordonet har kört längs referenslinjen och  $S$  är hela referenslinjens längd.

Skalfaktorn  $Scf$  beräknas med hjälp av information från referenslinjens tillståndsvariabler samt tillståndsvariabler från fordonsmodellen (6) enligt

$$Scf = \frac{(1 - d/r_{\text{ref}})}{V_{\text{xbil}}\cos(\psi - \psi_{\text{ref}}) - V_{\text{ybil}}\sin(\psi - \psi_{\text{ref}})}, \quad (26)$$

där  $\psi_{\text{ref}}$  är referenslinjens tangentvinkel.

I ekvation (26) finns två okända variabler  $d$  och  $r_{\text{ref}}$  som beräknas med hjälp av referenslinjens tillståndsvariabler enligt

$$d = (y_{\text{bil}} - y_{\text{ref}}) \cos(\psi_{\text{ref}}) - (x_{\text{bil}} - x_{\text{ref}}) \sin(\psi_{\text{ref}}) \quad (27)$$

och

$$r_t = \frac{1}{k} \quad (28)$$

där  $k$  är referenslinjens kurvatur för varje punkt och erhålls från matrisen  $T$  (6).

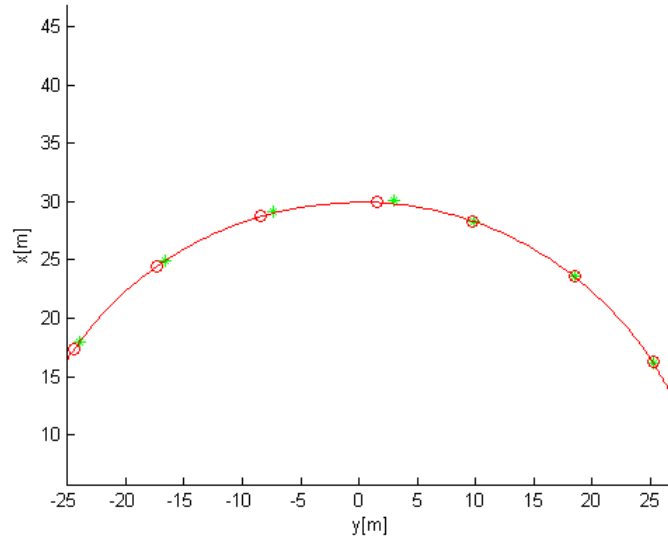
Sträckan  $s$  kan nu beräknas från  $Scf$  genom att modifiera ekvation (23) och sedan integrera ekvation (29).

$$ds = \frac{1}{Scf} dt \quad (29)$$

$$s = \int_0^T \frac{1}{Scf} dt \quad (30)$$

### Lokalisering av fordonet

Ett ackumulerande fel i modellen upptäcktes i ett tidigt stadiet av projektet, där sträckan  $s$  inte motsvarade den punkt på referenslinjen närmast fordonet, vilket den borde. Detta fel kunde observeras genom att för ett antal punkter plotta fordonets position tillsammans med positionen motsvarande  $s$  då fordonet åker runt en bana, se Figur 7.

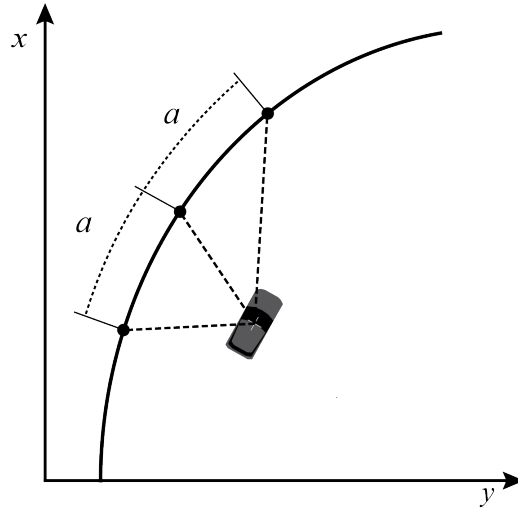


Figur 7: Figuren visar fordonets position (röda ringar) och punkter motsvarande sträckan  $s$  (gröna stjärnor) då fordonsmodellen kört runt en cirkelbana. Simuleringen börjar i tredje punkten från höger och ett litet fel kan observeras. I slutet av simuleringen har felet ackumulerat.

För korta banor var detta inget problem, men på längre banor observerades instabilitet. Felet beror troligtvis på en osäkerhet vid beräkningen av skalfaktorn  $Scf$ . En funktion för att rätta till detta fel har därför skapats. Funktionen jämför hela tiden fordonets position med tre punkter på referenslinjen. Punkterna som jämförs är punkten motsvarande sträckan  $s$  samt en punkt framför och en bakom.  $s$  uppdateras automatiskt till den sträcka som motsvarar punkten närmast fordonet. Eftersom felet ackumulerar fortare vid högre hastigheter har avståndet  $a$  mellan de tre punkterna satts till att variera enligt. Avståndet  $a$  mellan punkterna skalas med hastigheten enligt

$$a = 0.5V \quad (31)$$

och visas i Figur 8.



Figur 8: Punkterna som sträckan  $s$  justeras i samt avstånden mellan dessa.

Detta görs för att funktionen ska hinna justera  $s$  även vid höga hastigheter. Metoden har visat sig fungera bra och är lätt att verifiera genom att stega fram i simuleringen och se om  $s$  justeras som önskat. Eftersom felet är litet i början och ökar förhållandevis långsamt räcker det att jämföra med tre punkter. Hade felet varit större kanske fler punkter längs banan hade varit nödvändigt att ha med i beräkningen.

### 5.2.2 Förhandsgranskning

Förarmodellen använder sig av fordonets nuvarande position och riktningsvinkel för att räkna ut framtida positioner som ligger i linje med fordonets riktning. Hur långt fram förarmodellen förhandsgranskar beror på fordonets hastighet samt en förbestämd tidskonstant enligt

$$\hat{L} = V_{\text{xbil}} \cdot T_p \quad (32)$$

Sträckan  $\hat{L}$  delas upp i tre delsträckorna  $L = [L_0, L_1, L_2]^T$ , där  $L_0 = 0$ . Avståndet mellan delsträckorna ökar desto längre från fordonet de ligger, så att  $L_1 = 0.2\hat{L}$  och  $L_2 = 1\hat{L}$ , vilket illustreras i Figur 9. Detta ger bättre kontroll över regleringen och förarmodellen simuleras mer likt en riktig förare än om endast en punkt framför fordonet hade analyserats.  $L$  använder fordonets nuvarande koordinater samt riktningsvinkel på fordonet för att få

ut koordinaterna på slutet av varje delsträcka. Detta ger

$$x_{L,i} = x_{\text{bil}}(s) + L_i \cos(\psi) \quad (33)$$

och

$$y_{L,i} = y_{\text{bil}}(s) + L_i \sin(\psi), \quad (34)$$

där index  $i$  antar värden 0, 1 och 2. Positionerna läggs i vektorn  $(x_L, y_L) = [(x_{L,0}, y_{L,0}), (x_{L,1}, y_{L,1}), (x_{L,2}, y_{L,2})]$ , där  $(x_{L,0}, y_{L,0})$  är fordonets position.

### 5.2.3 Positionsreglering

För att fordonet ska följa referenslinjen måste dess position regleras i varje sampling. Detta görs genom att beräkna positionsfelet och vinkelfelet hos fordonet och med hjälp av en PID-regulator skicka en styrsignal till fordonsmodellen som ändrar dess kurvatur. Punkterna  $(x_{L,i}, y_{L,i})$  är de punkter som positionsfelet och vinkelfelet beräknas i.

#### Positionsfel

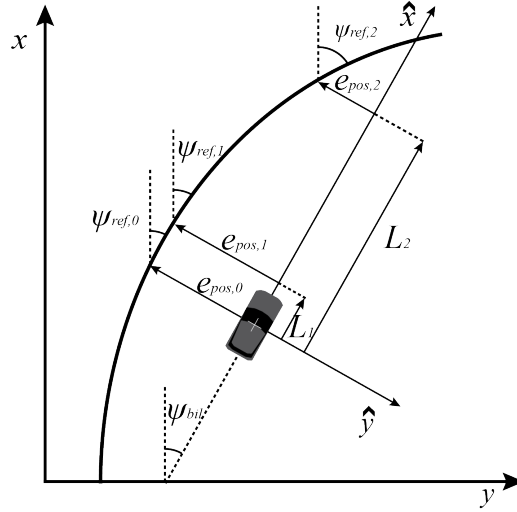
Positionsfelet i denna rapport definieras som vektorer vinkelräta mot fordonets riktning, det vill säga positionsfel i sidled. Vektorerna utgår från punkterna  $(x_{L,i}, y_{L,i})$  och slutar i punkterna där vektorerna skär referenslinjen, se Figur 9. För att beräkna var skärning med referenslinjen sker så adderas  $L_i$  till sträckan  $s$  som fordonet kört längs banan, detta ger referenslinjens motsvarighet till  $L$  och ges av

$$s_{\text{ref},i} = s + L_i, \quad (35)$$

så att  $s_{\text{ref}} = [s_{\text{ref},0}, s_{\text{ref},1}, s_{\text{ref},2}]$ . Beräkningen görs med antagandet att banan inte hinner svänga märkbart inom sträckan  $L$ . Felvektorerna kan alltså ses som vinkelräta bara då krökningen antas vara minimal under den givna sträckan. Hade banan hunnit svänga mycket så hade problem uppstått och dessa antaganden samt ekvation (35) hade ej gett ett fungerande system. Approximationen innebär att punkterna motsvarande  $s_{\text{ref}}$ ,  $(x_{\text{ref}}, y_{\text{ref}})$ , antas ligga på  $y$ -axeln i ett lokalt koordinatsystem med samma riktningar på axlarna som fordonets lokala koordinatsystem (Figur 9) och reglerfelet är detta  $y$ -värde. Origo för de lokala koordinatsystemen ligger i punkterna

$(x_L, y_L)$ . Denna metod för att hitta positionsfelen har visat sig fungera väl och förenklar beräkningarna.

Koordinaterna  $(x_{\text{ref}}, y_{\text{ref}})$ , tas fram genom interpolation av koordinatvektorerna i referenslinjen  $T$  (6) med variabeln  $s_{\text{ref}}$  som insignal. Detta ger punkterna  $(x_{\text{ref}}, y_{\text{ref}}) = [(x_{\text{ref},0}, y_{\text{ref},0}), (x_{\text{ref},1}, y_{\text{ref},1}), (x_{\text{ref},2}, y_{\text{ref},2})]$



Figur 9: Figuren visar fordonet och dess riktning, sträckan som förarmodellen förhandsgranskar med dess delsträckor  $L_1$  och  $L_2$  samt hur felvektorerna  $e_{\text{pos},0}$ ,  $e_{\text{pos},1}$  och  $e_{\text{pos},2}$  ser ut. Dessutom visas fordonets riktningsvinkel och banans tangentvinklar i punkterna där regleringen sker.

Positionsfelet beräknas med hjälp av ekvationerna i kapitel 1. Då felvektorn är definierad som ett värde på  $y$ -axeln i ett lokalt koordinatsystem fås positionsfelet  $e_{\text{pos}}$  från ekvation (5). Felet beräknas för positionerna  $(x_L, y_L)$  och läggs i en vektor enligt

$$e_{\text{pos}} = (y_{\text{ref}} - y_L)\cos(\psi) - (x_{\text{ref}} - x_L)\sin(\psi). \quad (36)$$

där  $e_{\text{pos}} = [e_{\text{pos},0}, e_{\text{pos},1}, e_{\text{pos},2}]$

### Vinkelfel

För punkterna i  $(x_L, y_L)$  räknas utöver positionsfelet även tre vinkelfel ut. Detta ges av sambandet

$$e_{\psi,i} = \psi_{\text{ref},i} - \psi \quad (37)$$

där  $\psi$  är riktningsvinkeln för fordonet och  $\psi_{\text{ref},i}$  antar värdet för de tre tangentvinklarna  $\psi_{\text{ref},0}$ ,  $\psi_{\text{ref},1}$  och  $\psi_{\text{ref},2}$ . Då fås tre vinkelfel som sammanfattas i vektorn  $e_{\psi} = (e_{\psi,0}, e_{\psi,1}, e_{\psi,2})$ . Se figur 9. Vinkelfelet används för att se till att fordonet håller samma riktningsvinkel som banan och ger en stabilare reglering.

### Regulator för att reglera position

Elementen i vektorn  $e_{\text{pos}}$  har förstärkningar 2, 1 och 0.1 så att förstärkningen blir lägre ju längre bort från fordonet punkterna ligger. Hela vektorn med vinkelfelen,  $e_{\psi}$ , har en förstärkning på 0.5 och ytterligare förstärkning på varje element i vektorn ligger på 1, 0.5 och 0.1. Även där blir förstärkningen lägre ju längre bort punkten som vi beräknar felet för ligger från fordonet. Elementen efter förstärkningarna adderas för både positions- och vinkelfelen och skickas sedan som en signal till en PID-regulator. Regulatorn har en proportionell del på 20, en integrerande del på 2 och en deriverande del på 16, vilket har gett ett önskat resultat för ett flertal banor av varierande karaktär. För att komma fram till inställningarna på regulatorn har metoden “trial and error” använts, det vill säga tunande och experimenterande tills önskat resultat uppnåtts. Även Simulinks inbyggda verktyg för att ställa in regulatorn har använts. Fokus låg på att hitta inställningar där fordonet följer banan bra och att undvika kraftiga svängningar i systemet.

PID-regulatorn skickar en styrsignal till fordonsmodellen som reglerar hur mycket och åt vilket håll fordonet ska svänga. Detta ges i form av storheten kurvatur per tidsenhet. Fordonets kurvatur är proportionell mot fordonets riktningsvinkel och är den storhet som används för att beskriva hur mycket fordonet svänger.

### Begränsningar på styrsignal

För att uppfylla kravet på att fordonet inte ska glida i kurvor måste styrsignalerna till fordonsmodellen begränsas. När ett fordon svänger påverkas det av en centripetalkraft som drar fordonet inåt i kurvan. På en, som i detta projekt enligt Avsnitt 1.2, platt bana och vid gränsen för glidning gäller att centripetalkraften är lika stor som friktionskraften mellan fordonet och underlaget. Detta eftersom friktionskraften är den enda kraften som påverkar

fordonet i sidled. Vid gränsfallet för glidning är friktionskraften maximal. Alltså gäller att

$$F_{\text{centripetal}} = F_{\text{friktion}}, \quad (38)$$

där centripetalkraften ges av

$$F_{\text{centripetal}} = \frac{mV(t)^2}{r} = mV(t)^2k, \quad (39)$$

med kurvaturen  $k$  och fordonets massa  $m$ .

Friktionskraften för ett fordon i rörelse utan glidning på en platt bana ges av

$$F_{\text{friktion}} = \mu F_N = \mu mg \quad (40)$$

där  $\mu$  är friktionskoefficienten för underlaget,  $F_N$  är normalkraften och  $g$  är tyngdkraften. Tillsammans ger ekvation (38), (39) och (40) att

$$mV^2k = \mu mg \quad (41)$$

och från ekvation (41) kan maximala värdet på  $k$ , det vill säga kurvaturen eller svängningsradien, hos fordonet fås:

$$k_{\text{max}} = \frac{\mu g}{V(t)^2} \quad (42)$$

Styrsignalen som reglerar fordonets position har  $k_{\text{max}}$  som övre gräns och  $-k_{\text{max}}$  som undre gräns.

#### 5.2.4 Hastighetsreglering

För att kunna reglera hastigheten utmed banan måste en referenshastighet räknas ut kontinuerligt. Referenshastigheten utgår ifrån kravet om att fordonet inte får glida. Som tidigare beskrivits i ekvation (41) uppstår glidning då centripetalkraften överstiger friktionskraften. Den maximala hastigheten utmed hela banan kan då beräknas genom att lösa ut den maximala hastigheten ur ekvation (41) enligt

$$V_{\text{max}} = \sqrt{\frac{g\mu}{k}}. \quad (43)$$



Kurvaturen  $k$  interpoleras från referenslinjen  $T$ . Man vill dock ha en säkerhetsmarginal för fordonet varvid referenshastigheten sätts till halva maximala hastigheten för att vara säker på att fordonet inte glider. Referenshastigheten är dessutom begränsad till 20 m/s för att undvika för höga hastigheter på raksträckor där referenshastigheten i teorin är oändlig. Referenshastigheten beräknas i samma punkter som används vid positionsreglering enligt (35). De tre referenshastigheterna subtraheras med fordonets momentan hastighet för att erhålla hastighetsfelen. Därefter viktas hastighetsfelen enligt 0.1, 0.5 och 2 där hastighetsfelet i punkten längst från fordonet har störst vikt. Detta har visat sig ge bäst resultat då förarmodellen hinner reagera och bromsa i god tid innan en kurva och behålla en låg hastighet genom hela kurvan.

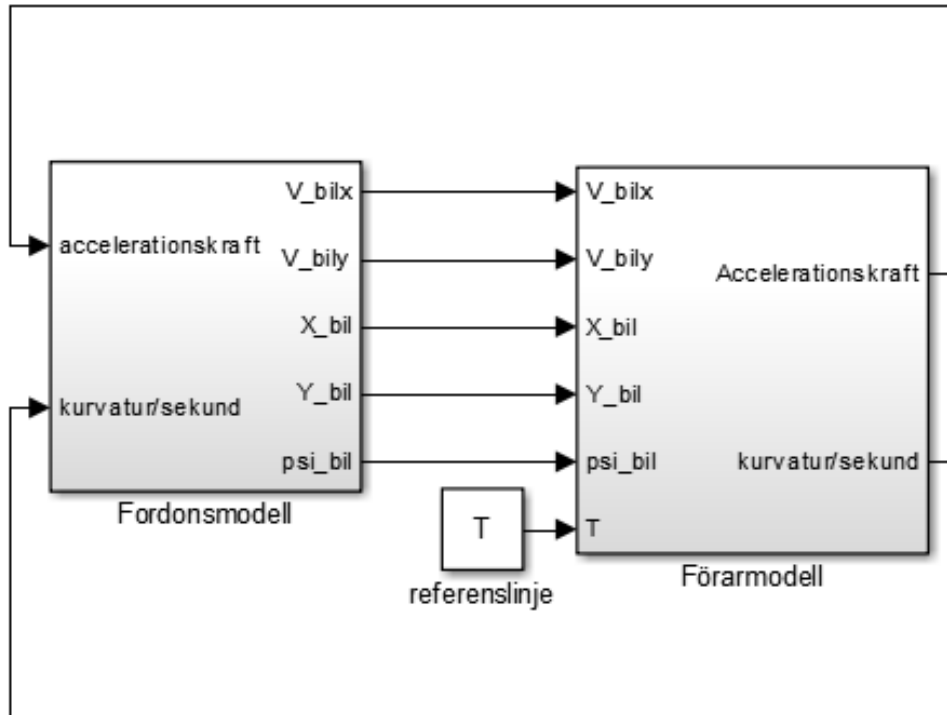
De viktade hastighetsfelen adderas och skickas in i en PID-regulator. Regulatorn har en proportionell del på 1, en integrerande del på 0.1 och en deriverande del på 0.1. Dessa värden har tagits fram genom upprepade försök med olika inställningar och har visat sig fungera bra. Regulatorn skickar som styrsignal en acceleration som kopplas till bilmodellen och på så vis reglerar dess hastighet.

## 6 Simulering och uppbyggnad av systemet

För simulering och uppbyggnad av reglersystemet har programmet Simulink använts i detta projekt. Då ekvationerna för systemet är uppställda och beskrivna i kapitel 4 och 5 är det enkelt att bygga upp Simulinkmodellen eller systemet utifrån dessa.

### 6.1 Översikt över reglersystemet i Simulink

Systemet är uppbyggt av subsystem för att ge en bra överblick. Subsystemen består av en förarmodell med dess två regulatorer och fordonsmodell. Figur 10 visar reglersystemet i sin helhet.

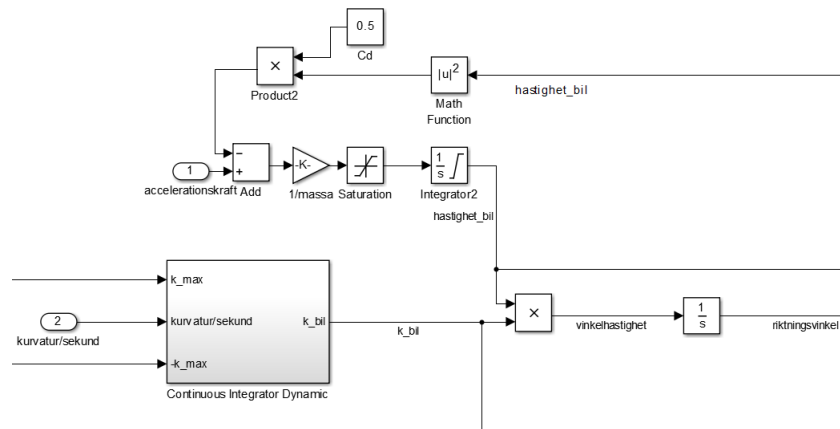


Figur 10: Översikt över systemet i Simulink.

Figur 10 visar hur förarmodellen använder sig av fordonsmodellens tillståndsvariabler tillsammans med referenslinjen  $T$  och skickar tillbaka styr signaler i form av en accelerationskraft och förändring av svängningsradie (kurvatur per tidsenhet).

## 6.2 Fordonsmodell i Simulink

Figur 11 visar en del av fordonsmodellens uppbyggnad i Simulink. Hela fordonsmodellen finns i Bilaga A. Modellen är uppbyggd utifrån ekvationerna i kapitel 4.1.

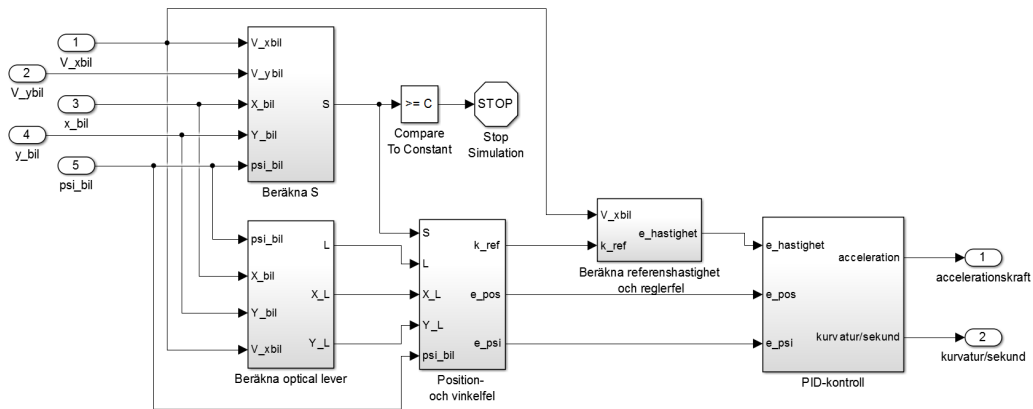


Figur 11: En del av fordonmodellen i Simulink.

Figur 11 visar den del av bilmodellen som beskriver krafterna som påverkar fordonet. Figuren visar även hur styrsignalen med storheten kurvatur per tidsenhet används för att få fram en vinkelhastighet vilket sedan integreras för att få fordonets riktningvinkel.

### 6.3 Förarmodell i Simulink

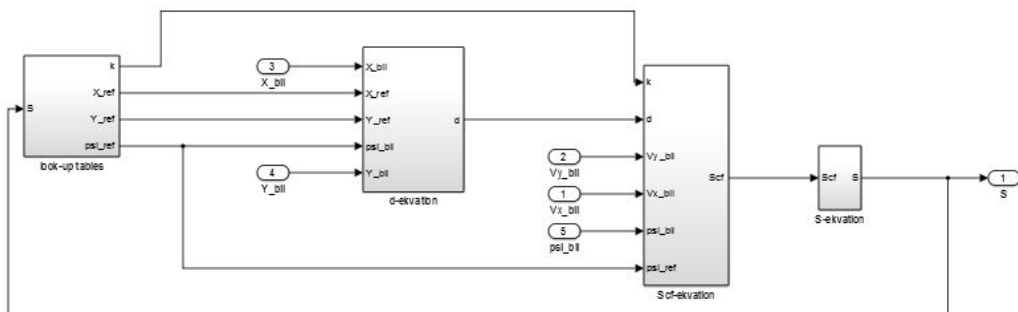
Förarmodellen är uppbyggd av subsystem för de olika uppgifterna som ska utföras. Figur 12 visar förarmodellen med subsystemen "Beräkna S", "Beräkna optical lever", "Beräkna referenshastighet och reglerfel", "Position- och vinkelfel" och "PID-kontroll".



Figur 12: Översikt över förarmodellen i Simulink.

En funktion för att stoppa simuleringen när hela banan körts finns också i förarmodellen, som jämför sträckan som fordonet kört med sista värdet i vektorn  $s$  som ligger i matrisen  $T$ . När fordonets färdade sträcka är större än sista värdet i  $s$  avbryts simuleringen.

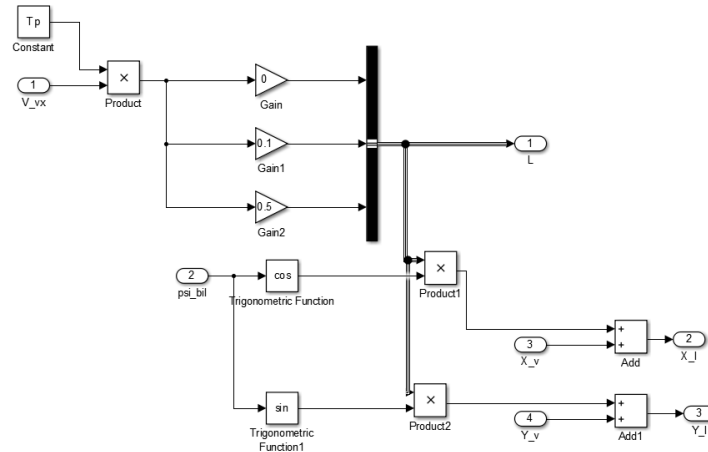
En viktig uppgift för förarmodellen är att beräkna sträckan  $s$  för varje sampling. Figur 13 visar implementationen av denna uppgift utifrån ekvationerna i kapitel 5.2.1. Sträckan  $s$  är den sträcka som fordonet har färdats längs med referenslinjen och används för att genom interpolation ta fram övriga värden ur matrisen  $T$ .



Figur 13: Beräkning av sträckan  $s$  implementerat i Simulink.

Beräkning av optical lever sker i ett tidigt stadi i förarmodellen och visas i

Figur 14. Här används bussfunktionen i Simulink för att förenkla uppbyggnaden av modellen. Bussfunktionen gör det möjligt att samla flera signaler och utföra operationer på dessa gemensamt. Här samlas de tre punkter vid vilka reglerfelen för position och hastighet beräknas.



Figur 14: Beräkning av optical lever i Simulink.

Övriga subsystem finns att studera i detalj i Bilaga B. I Bilaga B.1 finns funktioner för att beräkna position och vinkelfel, i Bilaga B.2 finns funktioner för att beräkna referenshastighet och hastighetsfel och i Bilaga B.3 finns PID-regulatorerna för att reglera hastigheten och positionen hos fordonet.

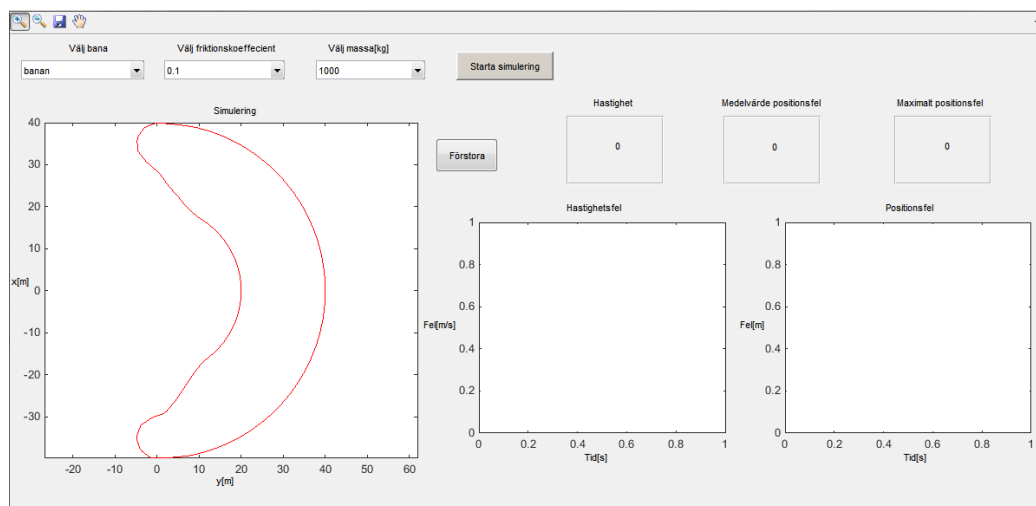
Simulink har använts under hela projektet för att testa och felsöka de olika delsystemen var för sig. Felsökningen har skett genom att analysera signalerna under simuleringen för att hitta var och när eventuella fel har skett. Verifiering har skett genom att studera plottar av intressanta signaler för att se att systemet beter sig som förväntat. Det slutliga resultatet är ett fullständigt reglersystem implementerad som en Simulink-modell, där de olika delsystemen är integrerade.

## 7 Visualisering i grafiskt användargränssnitt

Visualisering av systemet är en viktig del av projektet. Detta möjliggör för användare att själva testa systemet och studera resultatet genom ett enkelt användargränssnitt, eller GUI(Graphical User Interface) som det också kallas.

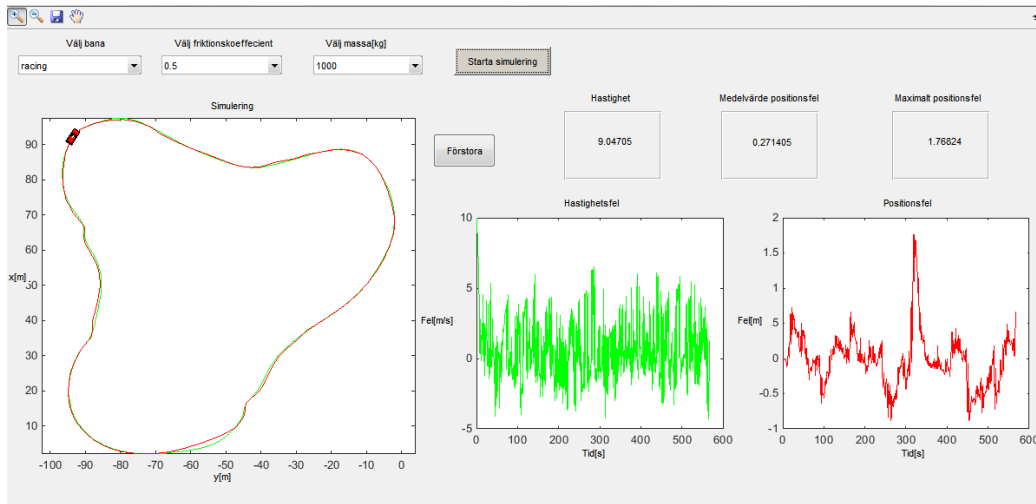
Visualiseringen av systemet sker i ett GUI skapat med hjälp av Matlabs verktyg GUIDE. GUIDE är ett enkelt verktyg där man grafiskt kan designa sitt gränssnitt. MATLAB genererar automatiskt kod där funktionaliteten för gränssnittet läggs till.

I GUI:t för detta projekt kan användaren välja bana, friktionskoefficient och massa på fordonet. Friktionskoefficienterna användaren har att välja på är 0.1, 0.2, 0.5, 0.7 och 1 medan massan på fordonet kan sättas till 1000 respektive 1500 kilogram. Därefter kan användaren köra simuleringen och studera resultatet. Under simuleringen kan användaren genom ett knapptryck välja att förstora bilden där visualiseringen sker, för att tydligare se hur fordonet åker. Förstoringen sker då i ett separat fönster som kan stängas när simuleringen är färdig.



Figur 15: Figuren visar hur användargränssnittet ser ut i sitt initialtillstånd, med friktionskoefficienten satt till 0.1, massan till 1000 kg och vald bana till "banan".

Positionsfel och hastighetsfel plottas, över samplingstiden, efter simuleringen i två grafer. Fordonets färdväg visualiseras så att användaren kan se en bil åka runt banan på skärmen. Samtidigt som bilen åker visas även dess hastighet i en ruta. Maximalt positionsfel och medelvärdet för positionsfelet visas efter simuleringen. Bilen som visualiseras är skapad i MATLAB och är ett så kallat Graphical Object.



Figur 16: GUI:t efter att simulering har körts.

Figur 16 visar GUI:t efter att en simulering har körts. Referensbanan (röd linje) samt fordonets färdväg (grön linje) visas tillsammans med bilen i den första grafen. I de två övriga graferna visas positionsfel och hastighetsfel över tid. Även det maximala positionsfelet och medelvärdet av positionsfelet kan observeras i separata rutor.

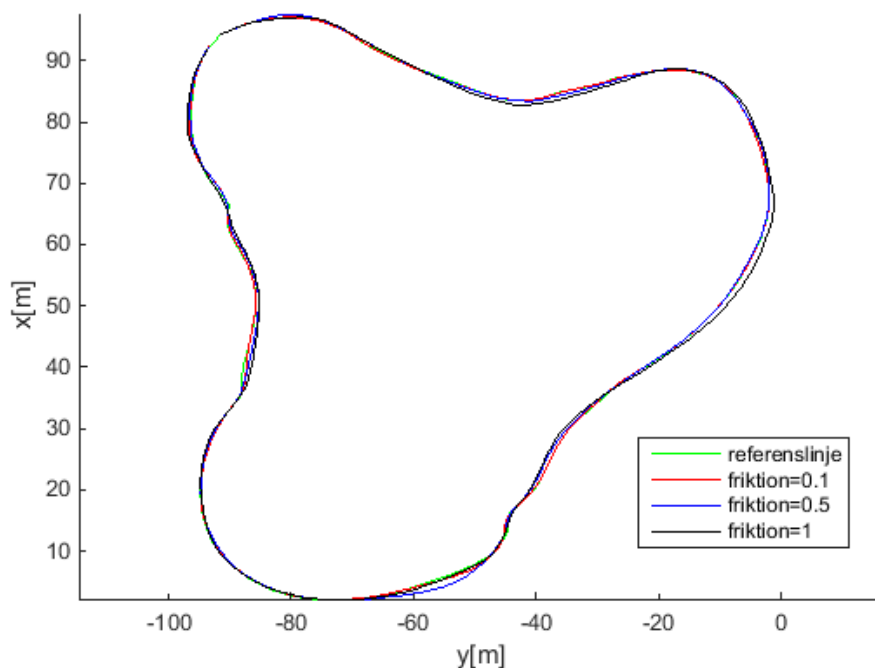
## 8 Resultat och diskussion

Alla resultat bygger på data extraherad från simulationer med olika inställningar. Först verifieras att förarmodellen och bilmodellen tillsammans visar förväntat resultat för en given referenslinje samt olika värden på massa och friktion. Vidare jämförs och diskuteras positionsfelets maximal- och medelvärde. Detta görs för olika referenslinjer och olika inställningar på massa och friktion. Därefter testas systemets robusthet genom att addera brus till fordonets position, detta görs också för att positionen i ett verkligt system ofta är svår att bestämma exakt. Andra tillståndsvariabler som hastighet och riktningvinkel är i regel lättare att bestämma exakt.

### 8.1 Simulering

En första verifikation på att systemet fungerar är att fordonet följer banor av varierande karaktär. Figur 17 visar hur fordonet följer en av banorna,

“Racingbanan”, vilket är en bana med varierande kurvatur och en hel del svängar.



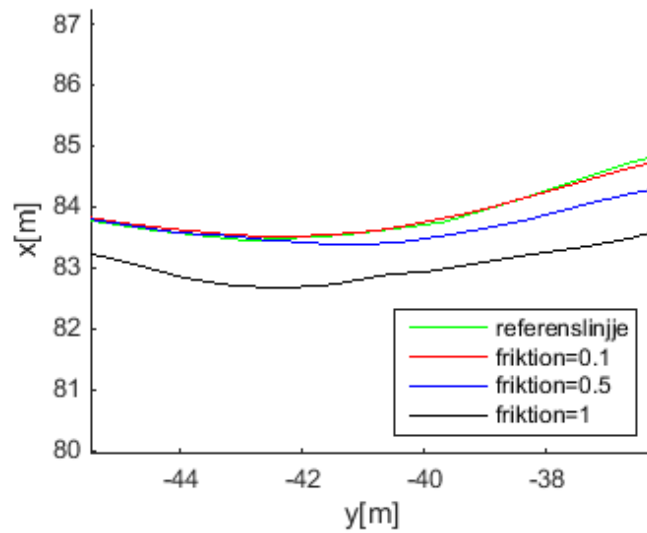
Figur 17: Simuleringsresultat på “racingbanan”.

Som Figur 17 indikerar så följer fordonet referenslinjen med hög noggrannhet för olika inställningar på friktionskoefficienten mellan fordon och vägbana. Liknande resultat kan observeras då fordonsmodellen simuleras på övriga banor, se Bilaga C. Detta är en god indikation på att reglersystemet fungerar. Ytterligare komponenter behöver dock analyseras för att säkerställa resultatet.

### 8.1.1 Friktionens och massans påverkan på simuleringsresultatet

Figur 18 visar en inzoomad del av racingbanan och fordonets färdväg vid olika inställningar på friktionskoefficienten.

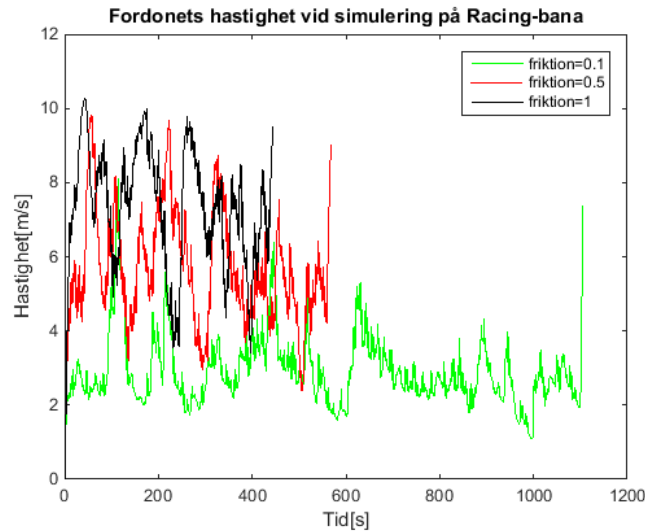




Figur 18: En del av banan och fordonets färdväg med friktionskoefficienten inställd på 0.1, 0.5 och 1.

Här ses att vid högre friktion avviker fordonet mer från banan än vid låg friktion. Detta beror på att hastigheten på fordonet ökar vid högre friktion, vilket tvingar fordonet att åka med större svängradie för att undvika glidning.

Figur 19 visar hur hastigheten varierar beroende på friktionskoefficienten.



Figur 19: hastigheten på fordonet vid simulering på Racingbaning vid olika inställningar på friktionen.

Här kan man se tydligt att hastigheten ökar till följd av friktionen då referenshastigheten är högre vid högre friktion. Liknande resultat kunde observeras för övriga banor och kan studeras i Bilaga D. Medelhastigheten är 7.2924 [m/s] då  $\mu = 1$ , 5.6863 [m/s] då  $\mu = 0.5$  och 2.9041 [m/s] då  $\mu = 0.1$ . Kravet att fordonet inte får glida samt att positionsregleringen blir svårare att få perfekt med högre hastigheter gör att fordonet åker av banan något. Den följer dock banan helt okej även vid de höga hastigheterna. Det maximala positionsfelet vid  $\mu = 1$  fås till ca 1.5 [m].

### 8.1.2 Analys av data vid olika inställningar på friktion och massa

För att undersöka effekterna av vilket värde som ställs in på parametrarna massa och friktion så jämförs färdvägen kring den aktuella banan. Den maximala avvikelserna och medelavvikelsen från banan beräknas för ett antal värden och banor och presenteras i Figur 20.

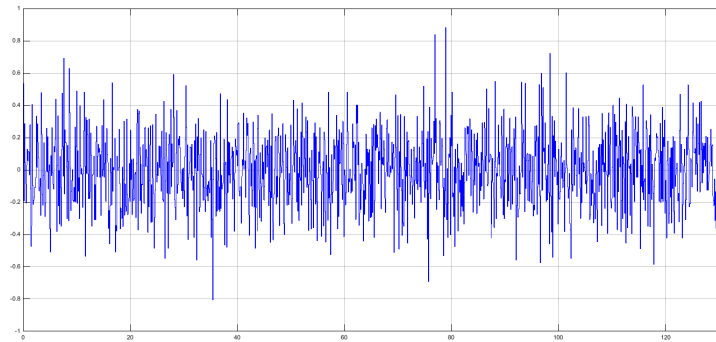
Cirkelbana				Bananbana			
Maxfel		Massa		Maxfel		Massa	
Medelfel		1000	1500	Medelfel		1000	1500
Friktion	0,1	0,1976	0,1829	Friktion	0,1	0,1607	0,1124
		0,035165	0,037988			0,018504	0,015889
	0,5	0,3354	0,1978		0,5	0,6795	0,6538
		0,060512	0,052421			0,077333	0,070414
	1	0,3441	0,2856		1	1,4383	1,3246
		0,069783	0,054430			0,103127	0,119847

Figur 20: Subfigurerna 20a och 20b visar hur friktionen och fordonets massa påverkar positionsfelet vid simulering på två olika banor.

I Figur 20 kan man se att både medel- och maxfel ökar då friktionen ökar och massan minskar. Detta beror på att hastigheten då ökar och gör regleringen svårare. Notera att det maximala felet blir som mest 1.44 [m], vilket är i överkant. Om det hade handlat om ett riktigt fordon kan en sådan felmarginal innebära att fordonet kör av vägen eller in i ett hinder. Då fordonet kör långsammare vid lägre friktioner och med större massa blir positionsfelen mindre. För att undvika större fel kan man därför begränsa hastigheten eller förbättra reglersystemet.

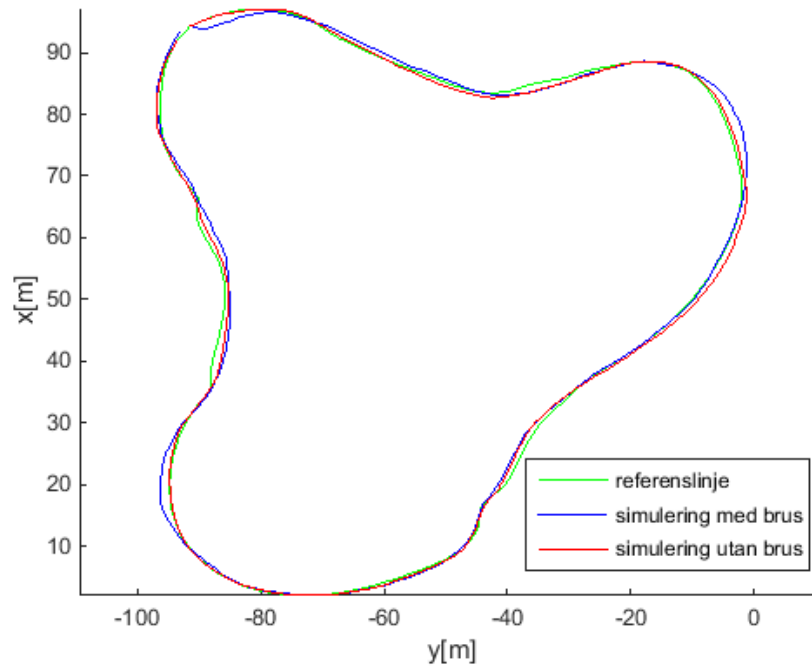
## 8.2 Simulering med brus

För att verifiera systemets robusthet har brus adderats till fordonets x- och y-koordinater. Bruset som adderats visas i Figur 21. Detta skapades med hjälp av Simulinks Band-limited White Noise-block som genererar slumpmässigt brus där amplituden beror på vad användaren ställer in för värde på "Noise power".



Figur 21: Brus som adderats tills fordonets position.

Bruset i Figur 21 ska simulera det fel som kan uppstå vid mätning av position av till exempel en GPS och man kan se att bruset, som adderas till fordonets koordinater, varierar mellan ca  $-0.8$  och  $0.8$  [m]. Förmodligen kommer GPS:er i autonoma fordon ha högre noggrannhet då det idag finns GPS-system med en noggrannhet på upp till någon centimeter (7). Fungerar reglersystemet även med högt brus adderat indikerar detta på att det implementerat i ett fysiskt system skulle klara av störningar såsom mätfel utifrån, vilket är en förutsättning för ett stabilt system. Figur 22 visar hur fordonmodellen följer en bana dels med brus och dels utan brus. Massan hos fordonet och friktionskoefficienten är densamma för båda simuleringarna.



Figur 22: Simulering med brus och utan brus

Här ses att fordonet lyckas följa banan även med brus, men avviker något mer i kurvorna vid brus tillagt. För att tydliggöra skillnaden på resultat mellan simulering med brus och utan brus har Tabell 23 sammanställts, i samma format som Tabell 20. Tabell 23 visar maximalt positionsfel och medelvärde för positionsfelet vid simulering med brus.

Cirkelbana (brus)				Bananbana (brus)			
Maxfel		Massa		Maxfel		Massa	
Medelfel		1000	1500	Medelfel		1000	1500
Friktion	0,1	2,0290	2,6131	Friktion	0,1	2,6518	2,6950
		0,328049	0,476613			0,428358	0,472943
	0,5	1,5319	2,1532		0,5	2,2398	10,533
		0,435306	0,482199			0,536631	1,12143
	1	3,6359	2,8086		1	4,519	4,0301
		0,616661	0,534467			0,707255	0,607342

Figur 23: Medelvärde och maximala positionsfelet, för olika inställningar på massa och friktion, vid simulering på Cirkelbana och Bananbana med brus adderat.

Från tabellen kan man se att vid maximal friktion och minsta massa på fordonet fås det högsta värdet på maximala positionsfelet, liksom vid simulering utan brus. För cirkelbanan uppstiger felet till 3.636 [m] och för bananbanan 4.519 [m]. Motsvarande maximala fel utan brus erhålls från Tabell 20 och fås till 0.344 [m] och 1.438 [m]. Bruset har således uppenbarligen påverkat systemet. Medelfelet med brus är 0.617 [m] för cirkelbanan och 0.707 [m] för bananbanan och motsvarande utan brus är 0.069 [m] och 0.103 [m]. Medelfelet har således också ökat mycket, proportionellt sett, i och med addition av brus, men är fortfarande acceptabelt. Ser man referenslinjen som mittlinjen på en väg så hade fordonet hålls sig kvar på vägbanan, om man ser på medelfelet. Problemet ligger således i extremvärdena. Som nämndes tidigare så är bruset som adderats dock förmodligen högre än vad motsvarande mätfel skulle vara och i verkligheten hade fordonet då inte påverkats så mycket som som ovanstående resultat visar.

### 8.3 Diskussion om reglersystemet

Fordonsmodellen som har använts i projektet är en mycket förenklad modell i form av en punktmassa, vilket inte är optimalt vid slutgiltig implementering i en fysisk bil. Fördelen med en punktmassmodell är dock att den är väldokumenterad och testad i många projekt och att den trots sin enkelhet har visat sig fungera bra i många fall, till exempel vid styrning av radiostyrda bilar (8).

Har man en fungerande punktmassmodell och god grundförståelse för denna, är steget heller inte så stort att gå vidare till mer komplicerade modeller. Nästa lämpliga steg vore att gå vidare till en cykelmodell där man ser fordonet som en cykel med ett hjul fram och ett hjul bak, eller en fyrhjulmodell där alla fyra hjul hos en bil modelleras. Dessa mer avancerade modeller hade bättre beskrivit en verklig bil och på så sätt också kunnat regleras mer exakt i ett verkligt system. Omständigheterna runtom fordonsmodellen, såsom friktion och luftmotstånd, är också förenklade.

En mycket enkel förarmodell har använts för att reglera fordonsmodellen, den kollar endast i tre punkter framför fordonet för att göra en bedömning om vilka styr signaler som ska skickas. Begränsade resurser har lagts på viktning av reglerfel och tuning av regulatorer. En bättre förarmodell hade kunnat skapas genom att använda fler punkter samt genom bättre viktning och tuning. Ett annat alternativ hade varit att använda en mer avancerad metod kallad Model Predictive Control(MPC), vilket förmodligen hade gett ett bättre resultat, då det går ut på att optimera ett givet reglerproblem. Metoden med PID-regulatorer ger dock en bättre förståelse för grundläggande reglering. I ett verkligt system måste särskild hänsyn tas till hur tidseffektivt regler systemet är. För att få så exakt reglering som möjligt måste systemet ha en låg responstid, i detta projekt har ingen hänsyn till responstid tagits.

Referenslinjen i regler systemet motsvarar data hämtad från färdvägen i ett verkligt system. Inga resurser har lagts på att undersöka vilka metoder det finns för att hämta denna data. I ett verkligt system måste även hänsyn till eventuella fel och osäkerhet i datan tas.

## 9 Slutsatser och vidare arbete

I stort anses projektet lyckat då förarmodellen klarar att reglera fordonsmodellens position och hastighet på olika typer av referenslinjer utan att avvika avsevärt. Projektet är dock mycket grundläggande. Med mer resurser kan en fordonsmodell som bättre beskriver verkligheten och en förarmodell som bättre reglerar fordonsmodellen konstrueras. Ett nästa steg hade varit att ta hänsyn till glidning hos fordonet samt höjdskillnader på vägen. Detta hade gett möjligheter att testa systemet på högre hastigheter. På grund av projektets grundläggande nivå har dock en bra grundläggande förståelse för regler system och modellering skapats och dokumenterats. Ett väl dokumenterat projekt på en grundläggande nivå kan vara användbart för andra som vill

skapa sig en bättre förståelse för modellering och reglersystem eller vill bygga vidare på ett mer avancerat reglersystem. Projektet kan även användas som grund för ett enkelt fysiskt system med verkliga bilar.

### **Projektet ur ett miljö- och hållbarhetsperspektiv**

Projekt som dessa kan användas vid testning och utformning av autonoma fordon, något som är högaktuellt och förväntas inom en snar framtid göra bilkörning både säkrare och miljövänligare med hjälp av snabbare och mer exakt styrning och ökad möjlighet för smart planerad körning (9).

Tester i form av simuleringar istället för fysiska tester är också ett sätt att minska utsläpp inte bara i bilindustrin utan i övriga industrier (10). Detta kräver välutvecklade simuleringsmiljöer med modeller som beskriver verkligheten på ett så bra sätt som möjligt. Detta projekt kan därför, med en väldokumenterad och fungerande simuleringsmiljö, ses som en grundpelare för att i framtiden öka användningen av testsimuleringar och på så sätt ge mindre påverkan på miljön

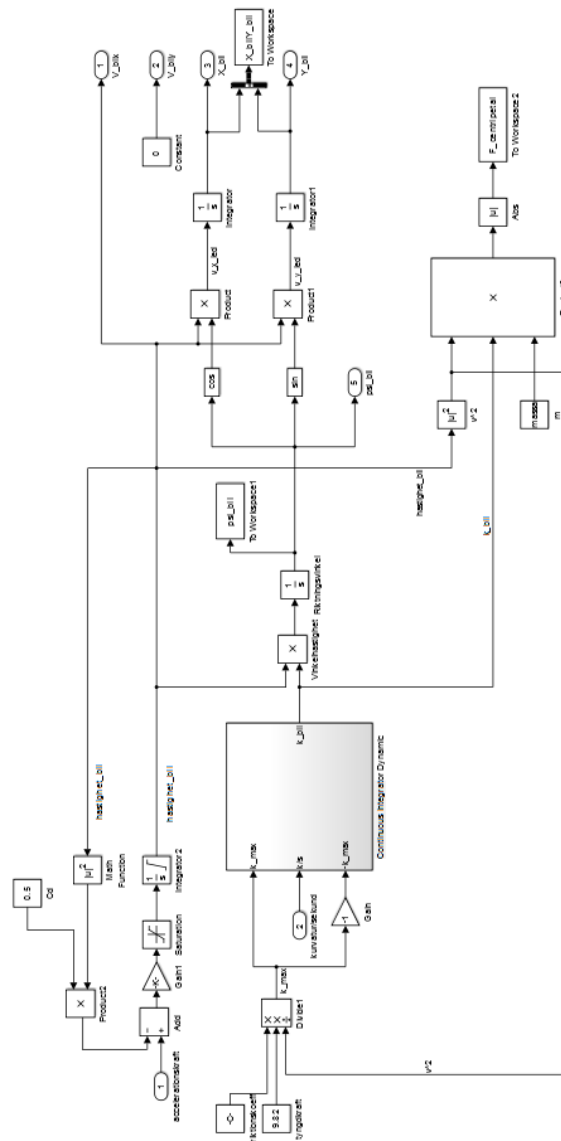


## Referenser

- [1] PiiA, "Vi vill leda utvecklingen mot förarlösa fordon.,"
- [2] B. A. et al., "Autonom körning," report, Väg- och järnvägsavdelningen, 2014.
- [3] C. de Locht, "Från hjälpsystem till autonoma fordon," 2014.
- [4] TT, "Säkerhet knäckfråga för självkörande bilar," 2015.
- [5] Y.-B. Jia, "Curvature," tech. rep., IOWA State University, 2014.
- [6] D.Casanova, "On minimum time vehicle manoeuvring: The theoretical optimal lap," phd thesis, Cranfield University, 2000.
- [7] Precisionsskolan, "Positionering,"
- [8] A. N. et al., "Cars - camera-based autonomous racing system," tech. rep., Uppsala Universitet, 2014.
- [9] Trafikverket, "Unikt pilotprojekt med självkörande bilar på allmän väg," 2013.
- [10] J. Bergh, "Simuleringar ska minska industrins miljöpåverkan," 2010.

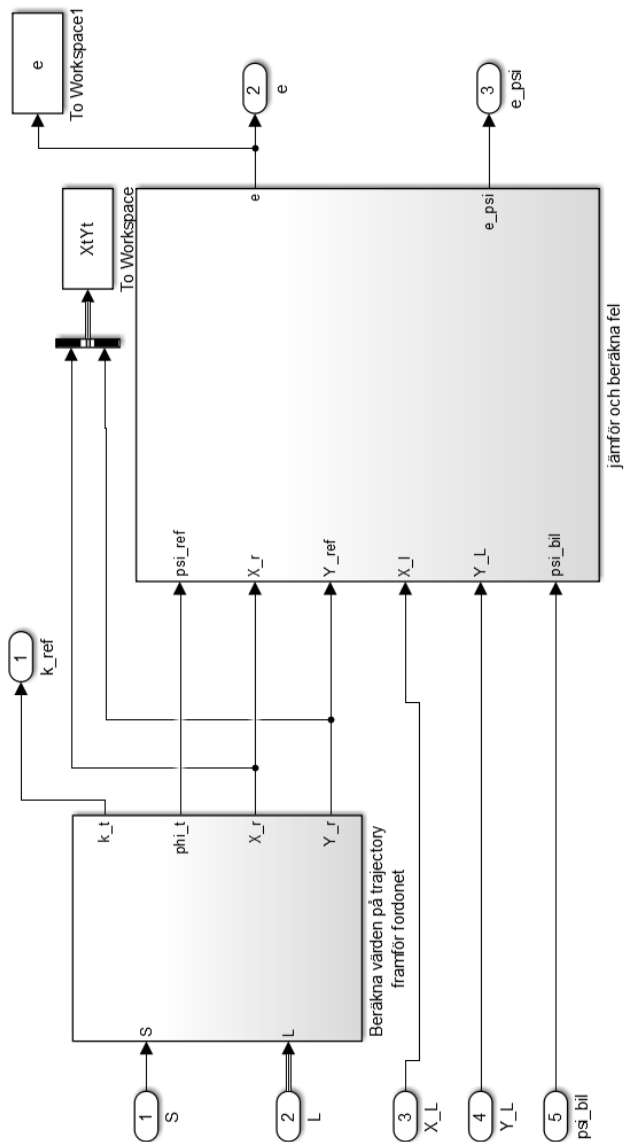
# Bilagor

## A Fordonsmodell i Simulink

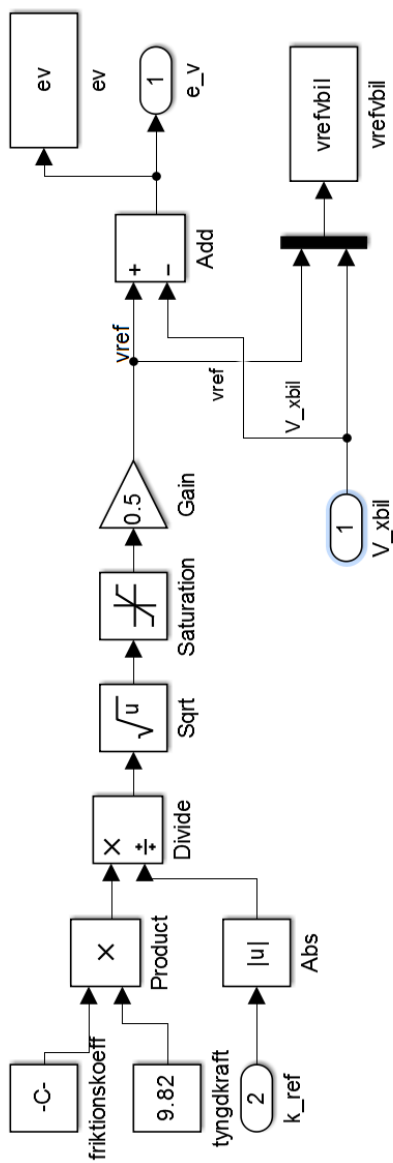


## B Förarmodell i Simulink

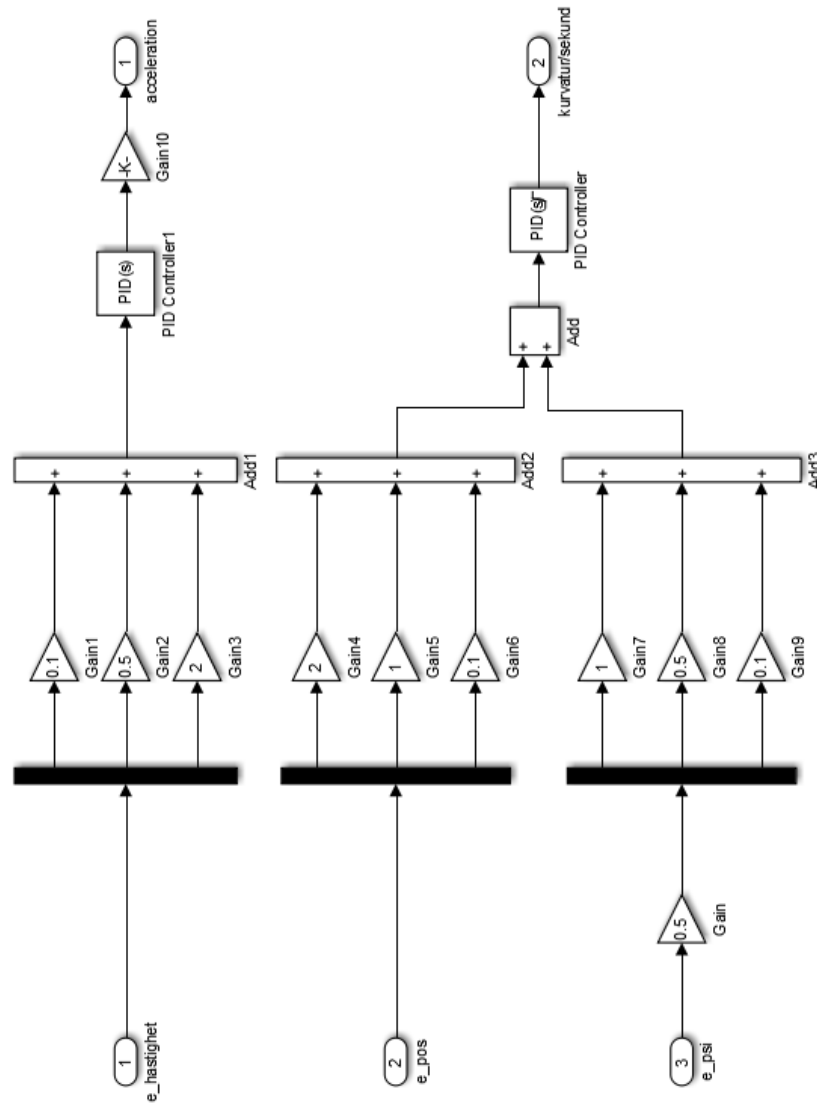
### B.1 Beräkning av positions- och vinkelfel i Simulink



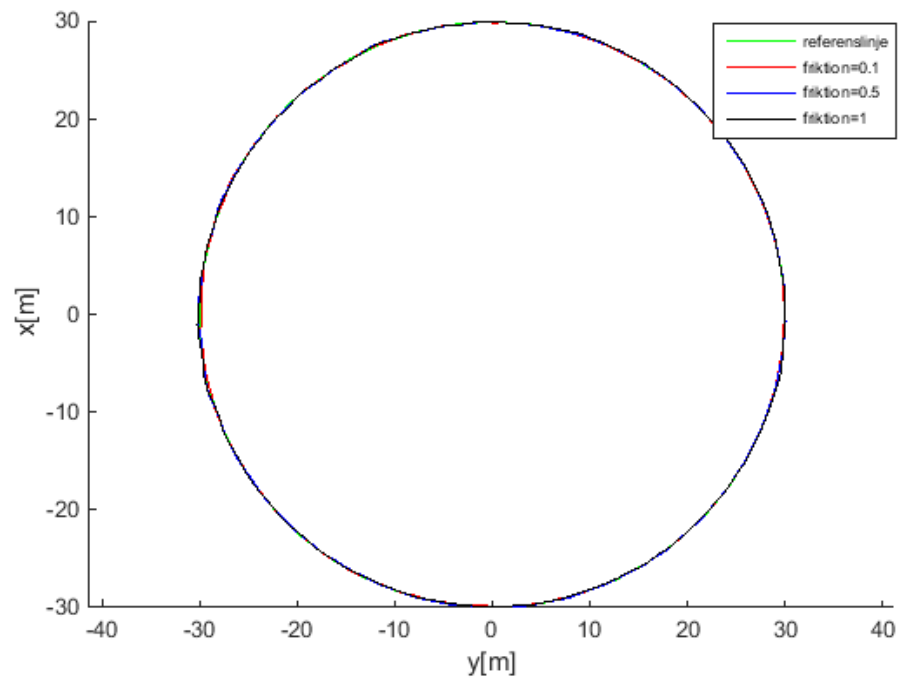
## B.2 Beräkning av referenshastighet i Simulink

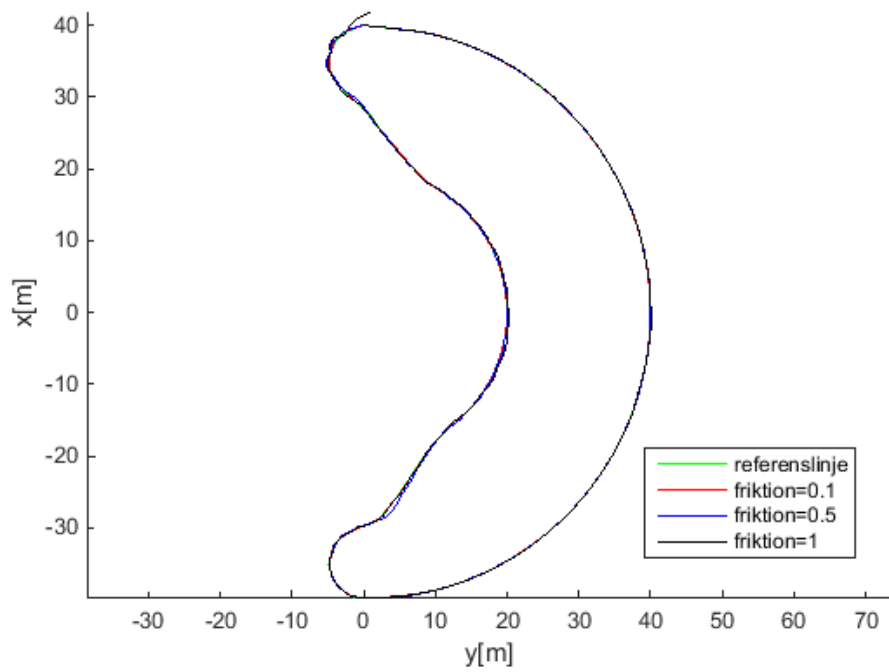


### B.3 PID-kontroller i Simulink



## C Simuleringsresultat





## D Hastighet vid olika friktion

