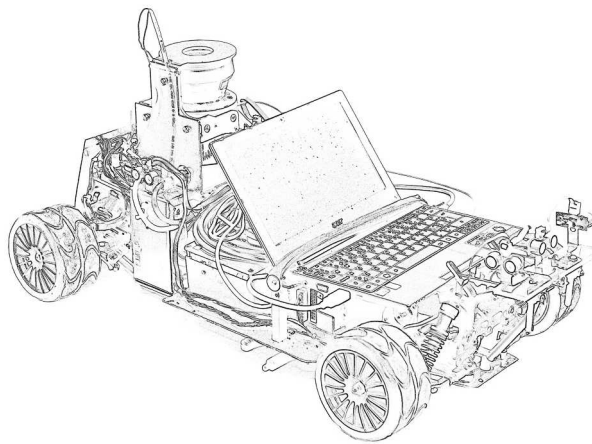


CHALMERS



Kandidatarbetesrapport



Design och implementering av ett autonomt parkeringssystem - Autonomous Parking Assistance, APA

Oskar Danielsson
Jens Samuelsson
Henrik Barsk
Viktor Nilsson
Mario Majdandzic
Emanuel Mosquera

Institutionen för signaler och system
Chalmers tekniska högskola
Göteborg, 2012

SSYX02-12-22

Förord

Rapporten avser kandidatarbetet Autonomous Parking Assistance, APA. APA syftar till att implementera en självstyrande parkeringsfunktion på en radiostyrd bil. Bilen har sedan ett antal år ingått i ett flertal projekt. Förra årets projekt (Autonomous Collision Avoidance, ACA) hade som mål att låta bilen detektera ett mobilt objekt av vilket den ledsagades genom en bana utan att kollidera med utplacerade hinder. Rapporten inleds med en bakgrundsbeskrivning. Därefter redogör rapporten för syftet med projektet och en problemställning tas upp, varpå definierade delproblem presenteras. Projektets målsättning definieras utifrån problem och avgränsningar. Avslutningsvis diskuteras resultaten utifrån valda metoder och teorier vilket i kombination med inledningen leder till en slutsats.

Tack till

Inledningsvis vill vi tacka alla berörda parter för all hjälp vi fått med projektet. Först och främst, tack alla föregående projektgrupper som passionerat ägnat tid till att utveckla bilen. Tack Chalmers Robotförening som bidragit med betydande stöd i form av expertis och material. Vi önskar framföra ett extra tack till föreningens medlemmar Erik Sternå och Victor Påsse. Vi vill också passa på att rikta ett stort tack till Claes Ohlsson för handledning av rapportskrivandet samt till vår sponsor Alina Systems som försåg oss med en bärbar dator till projektet. Slutligen, ett hjärtligt tack till vår handledare Erik Coelingh för sin rådgivning.

Sammanfattning

Kandidatarbetet APA (Autonomous Parking Assistance) har som syfte att implementera en autonom parkeringsfunktion i en radiostyrd bil. Parkeringen som utförs är av typen vinkelrät parkering. Som utgångspunkt för projektet finns en radiostyrd bil som varit del i ett antal tidigare projekt.

Bilen söker efter ledig parkeringslucka längs en rad med parkerade bilar och bedömer därefter om luckan är lämplig att parkera i. Därpå positionerar sig bilen framför luckan och backar in i den. Inne i luckan rätar bilen upp sig och parkeringen är fullbordad.

En laserscanner utgör medel för att avgöra såväl kollisionsrisk med främre sidan av bilen som att detektera lämplig parkeringslucka. Ultraljudssensorn används i syfte att bedöma kollisionsrisken bakåt samt bilens position i luckan. Bilen orienterar sig utifrån en funktion som omvandlar laserscannerdata till sammanhängande linjer. För att lösa uppgiften har funktioner och modeller simulerats i **MATLAB-Simulink**.

En jämförelse över de viktigaste resultaten mellan simuleringsmodellen och den verkliga modellen delas upp i: detektering och mätning av parkeringslucka, lämplig utgångsposition, och reglering. Med utgångsposition menas den position bilen ska ha före insvängningen i luckan.

Trots pålagt brus lyckas simuleringsmodellen med 5 % marginal, prestera korrekta mätvärden av parkeringsluckan i 9 fall av 10 till skillnad från endast 6 fall av 10 för den verkliga modellen. Simuleringsmodellen och den praktiska tillämpningen skiljer sig markant åt vad gäller utgångspositionen, då det i praktiken existerar tröghet i det verkliga systemets vilket inte inkluderas av simuleringsmodellen. Den simulerade positionsregleringen skiljer sig avsevärt från regleringen av den verkliga bilen. Denna differens förklaras med att simuleringsmodellen inte inbegriper mätdata från laserscannern vilket gör att mätstörningar inte påverkar resultatet från simuleringen.

Med nuvarande begränsningar av prestanda i det verkliga systemet, lyckas tillämpningen APA endast utföra en lyckad fullständig parkering i 3 fall av 10.

Abstract

The bachelor thesis APA (Autonomous Parking Assistance) aims to implement an autonomous parking function in a radio controlled car. The type of parking implemented is of the perpendicular type. The radio controlled car on which APA is applied has already been part of several previous projects. The car searches for an unoccupied parking space along a row of parked cars and decides thereafter whether the car could fit in the space or not. Then the car positions itself in front of the empty parking space and reverses towards it. When the car is positioned inside the parking space it straightens out the wheels and the parking sequence is completed. Furthermore, the car is able to avoid collision while parking. A laser scanner provides the means for determining both the risk of collision with the front of the car and finding a suitable parking space. Ultrasonic sensors are used with the purpose to decide the risk of collision when the car is reversing as well as for estimating the position of the car inside the parking space. The car is orientated by a function that translates the laser scanner data and returns connected lines. In order to solve the task, several functions have been simulated in **MATLAB-Simulink**.

The comparison of the most important results between the simulation model and the real model, is regarded in: the detection and measuring of the parking space, the adequate starting position, and the controlling. The starting position is referring to the position the car should have just before being conducted into the parking space.

Even though distortion is being added to the simulation model, it succeeds in acquiring the correct parking space measurements, with 5 % margin, in 9 cases out of 10, which should be compared to the real model that only succeeds in 6 out of 10 cases. In terms of estimating the starting position, the simulation model and the implementation differ markedly due to the friction of the real system. This friction is not considered in the simulation. Regarding the controlling of the car's position, there is a substantial distinction observed between the position of the car in the simulation model and the position of the real car. This distinction occurs because of that the simulation model does not include measured data from the laser scanner. As a result, distortion from the measuring is not being taken into consideration. Having the existing limitations in the performance of the real system, the implementation of APA does not fulfill the parking successfully in more than 3 cases out of 10.

Nomenklatur

α	spridningsvinkel för ultraljudssensorer
β	<i>slip-vinkel</i> , vinkel mellan hastighets och riktningsvektor för bilen
b	bredd
δ_f	styrvinkel framhjul
δ_r	styrvinkel bakhjul
δ	styrvinkel
d	avstånd
G	överföringsfunktion från styrvinkel till Y-position
h	höjd
L_x	namngivning av linje x
l_f	avstånd mellan TP och framhjul
l_r	avstånd mellan TP och bakhjul
l	avstånd mellan fram- och bakhjul
R	svängradie
R_{min}	minsta möjliga svängradie
pos_{laser}	avståndet mellan bilens kontur och laserscannerns mittpunkt
T	avstånd i godtyckliga längdenheter
TP	tyngdpunkt
V	bilens hastighet
v	hastighet i lokalt koordinatsystem
X	x-koordinat i ett globalt koordinatsystem
$\dot{X}$	x-hastighet i ett globalt koordinatsystem
$\dot{x}$	x-hastighet i ett lokalt koordinatsystem
ψ	bilens riktning i ett globalt koordinatsystem
Ψ	Laplacetransform av bilens riktning i ett globalt koordinatsystem
Y	y-koordinat i ett globalt koordinatsystem
$\dot{Y}$	y-hastighet i ett globalt koordinatsystem
$\dot{y}$	y-hastighet i ett lokalt koordinatsystem

Innehåll

1	Inledning	1
1.1	Bakgrund	1
1.1.1	Parkeringsassistans för fordon	1
1.1.2	Hållbar utveckling	1
1.1.3	Beskrivning av ursprunglig projektbil	2
1.2	Syfte	3
1.3	Problem	3
1.4	Avgränsningar	4
1.5	Mål	5
2	Teori	6
2.1	Fordonsdynamik	6
2.2	Positionsmodell	8
2.3	Detektering och sensorplacering	9
2.4	Line tracking-algoritmen	10
3	Metod	11
3.1	Design	11
3.1.1	Detektering och sensorplacering	11
3.1.2	Konstruktion av styrsystem	13
3.2	Algoritmer	14
3.2.1	Detektering av parkeringslucka	14
3.2.2	Parkeringsutförandet	14
3.3	Uppbyggnad av simuleringsmodellen	15
3.3.1	Sensorinformation	16
3.3.2	Tester av simuleringsmodellen	18
3.4	Tester av bilens funktioner	19
4	Resultat	21
4.1	Simulering	21
4.1.1	Utgångsposition vid parkering	21
4.1.2	Detektering av parkeringsluckan	22
4.1.3	Positionsreglering	23
4.2	Bilen	24
4.2.1	Detektering av parkeringslucka	24
4.2.2	Positionering i förhållande till lucka	25
4.2.3	Undvika kollision	26
4.2.4	Fullständig parkering	27

5	Diskussion	28
5.1	Simulering	28
5.2	Bilen	29
5.3	Data från ultraljudssensorerna	29
5.4	Diskussion av skillnader och likheter mellan bilen och simuleringsmodellen	32
6	Slutsatser	34
6.1	Rekommendationer för framtida projekt	35
A	Appendix	37
A.1	Ursprunglig hårdvara	37
A.2	Flödesscheman algoritmer	38
A.3	Mätdata simuleringar	40
A.4	Mätdata bilen	41
A.5	Blockscheman simuleringsmodellen	44

1 Inledning

1.1 Bakgrund

I allt större utsträckning utvecklas fordonsindustrin genom att mer teknologi samordnas med föraren. Med teknologi menas hjälpmedel och säkerhetsfunktioner för att assistera föraren med att bevara en trygghet gentemot sig själv och sin omgivning under körningen. Målsättningen med att integrera mer teknologi och datorkraft i fordon är att underlätta för föraren. Fordonet tilldelas kapacitet att på egen hand medverka till en säkrare körning genom att uppfatta och analysera situationer av olika slag och bedöma huruvida ett ingripande från fordonet själv krävs eller inte. Ett sådant autonomt system resulterar i en ansenlig förbättring av förarens komfort och en minskning av antalet moment som kan utgöra faror i trafiken.

1.1.1 Parkeringsassistans för fordon

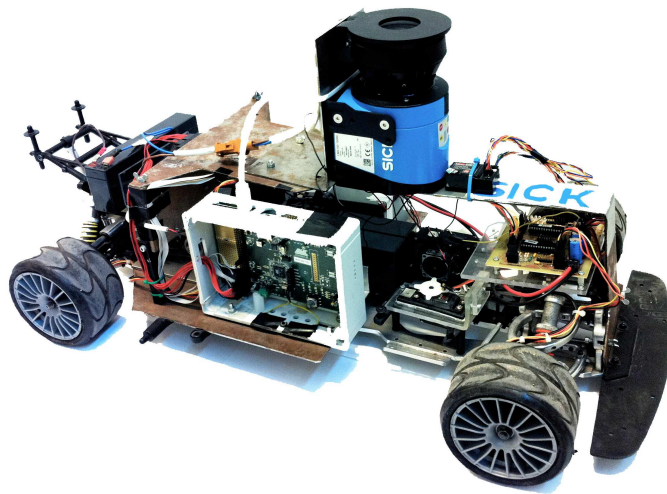
Det finns möjlighet att förbättra varierande parkeringsförhållanden. Bland annat innebär parkeringssituationer att föraren måste vara medveten om en mängd olika objekt framför, vid sidan och bakom fordonet. Samtidigt innebär förfarandet ofta svårigheten att fordonet behöver backas in i parkeringsluckan. En autonom parkeringsfunktion skulle underlätta för föraren genom att avlasta delar av dennes uppmärksamhet. Dessutom innebär det en effektivisering då parkeringen skulle kunna ske på ett försök.

1.1.2 Hållbar utveckling

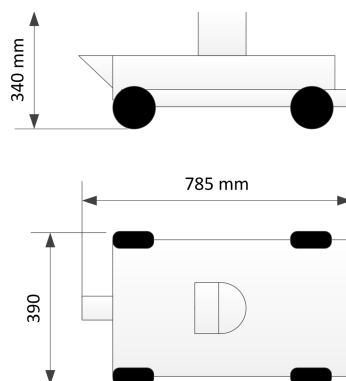
Då antalet parkeringsskador ökar innebär det ett problem för många bilägare, både för de som orsakar skador och de som får sin bil skadad [1]. En autonom parkeringsassistans som kan undvika kollision skulle kunna avhjälpa detta problem. Minskade karosskador ger en minskad förbrukning av metaller och plast. Det leder till att koldioxidutsläppen vid produktion av metaller och plaster minskar, vilket är fördelaktigt för miljön. Det ger även en ekonomisk fördel för bilägarna. Därmed kan en autonom parkeringsassistans ge både en ekonomisk och ekologisk fördel.

1.1.3 Beskrivning av ursprunglig projektbil

Bilen som används under projektet är konstruerad och utvecklad av tidigare kandidatprojekt vid Chalmers tekniska högskola och visas i Figur 1. Bilen är en fullt fungerande radiostyrd bil, vilket innebär att motor, servon för styrning och bromsar samt radiomottagare finns monterade. En handkontroll finns för radiostyrning av bilen, vilken dock inte används i den färdiga implementeringen. Avläsning av omgivningen sköts av en laserscanner. Alla komponenter drivs av tre 12 V batterier. Bilens styrsystem består av tre microcontrollerkort, AVR1 för styrning av bilen, ARM för datahantering och Arduino BT för dataloggning. Korten och deras samband beskrivs närmare i Appendix A.1. Den radiostyrda bilen finns sedan tidigare modellerad i MATLAB-Simulink. Bilens dimensioner illustreras i Figur 2.



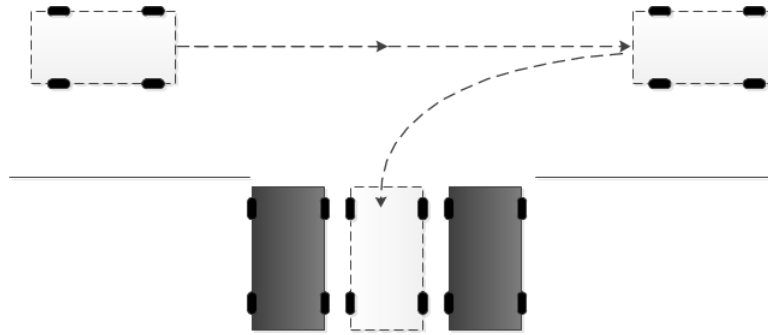
Figur 1: Den ursprungliga bilen



Figur 2: Bilens dimensioner

1.2 Syfte

Syftet med projektet Autonomous Parking Assistance, APA, är att utarbeta en funktion för autonom parkering och implementera funktionen i ett fordon. Projektet kommer att fokusera på en så kallad vinkelrät parkering, enligt Figur 3, härnäst syftar ordet parkering på just denna typ. Den radiostyrda bilen kunna parkera autonomt utan att krocka med omkringliggande hinder.



Figur 3: Parkeringstyp för bilen

1.3 Problem

Huvuduppgiften är att få den radiostyrda bilen att autonomt identifiera en parkeringslucka och sedan parkera. Detta skall utföras utan att bilen krockar med något föremål under manövern. Mer detaljerade delproblem/deluppgifter presenteras i listan nedan.

- Köra längs en given raksträcka

Fordonet behöver kunna köras längs en given raksträcka med parkerade bilar. Avståndet till dessa bilar måste mätas för att kunna följa raksträckan utan att krocka.

- Undvika kollision

Om bilen skall kunna stanna för hinder måste det finnas sensorer som kan detektera objekt på så långt avstånd att bilen hinner stanna utan att krocka.

- Hitta en lucka och beräkna dess storlek

Samtidigt som bilen kör måste den kunna avläsa var ett fast objekt slutar och ett annat börjar, för att kunna detektera en parkeringslucka där emellan. Detta kräver sensorer som kan detektera objekt tillräckligt fort så att alla objekt hinner detekteras. Vidare måste tillräcklig mängd information sparas för att hela luckan skall kunna uppfattas.

- Parkera

Bilen måste kunna placera sig på en viss position i förhållande till parkeringsluckan. Den ursprungliga hårdvaran på bilen har ingen möjlighet att se bakåt då laserscannern bara sveper i 270° framför bilen. Det gör att bilen inte kan manövrera bakåt på ett säkert sätt. Möjligheten för bilen att skaffa sig information om vad som finns bakom den måste därmed implementeras för att parkering skall kunna utföras.

- Begränsningar i gammal hårdvara

Enligt Wareborn¹ besitter den gamla hårdvaran en del begränsningar som skapar problem. Dessa gör att endast en liten del av informationen från laserscannern kan utnyttjas och mängden programkod som kan exekveras begränsas avsevärt. ARM-processorn har begränsade möjligheter att skicka ut information till en PC för felsökning. AVR1-processorn startar om sig slumpmässigt. Enligt Sternå² kan det bero på problem med signalnivåer.

1.4 Avgränsningar

Utifrån bilens egenskaper och begränsningar samt för att förtydliga parkeringsutförandets ramar har avgränsningar utformats.

- Hädanefter motsvaras alla hinder och parkerade bilar av rektangulära lådor och skall vara minst 300 mm höga, vilket är den lägsta nivå som laserscannern kan detektera.
- Bilen skall endast kunna utföra en parkering i en lucka med bilar på varje sida, där bilarna står idealt parkerade.
- Parkeringsluckans dimensioner förändras ej efter att den blivit detekterad, vilket innebär att de omkringliggande parkerade bilarna inte förflyttar sig under parkeringens gång.
- Om parkeringsluckan är tom när den detekteras förutsätts den vara det även vid parkeringsutförandet.
- Bilen skall endast kunna detektera en parkeringslucka på sin högra sida, sett till färdriktningen.

¹Victor Wareborn (Kandidatarbete ACA 2011) Intervjuad 25 jan 2012

²Erik Sternå (Kandidatarbete ACA 2010) Intervjuad 15 feb 2012

1.5 Mål

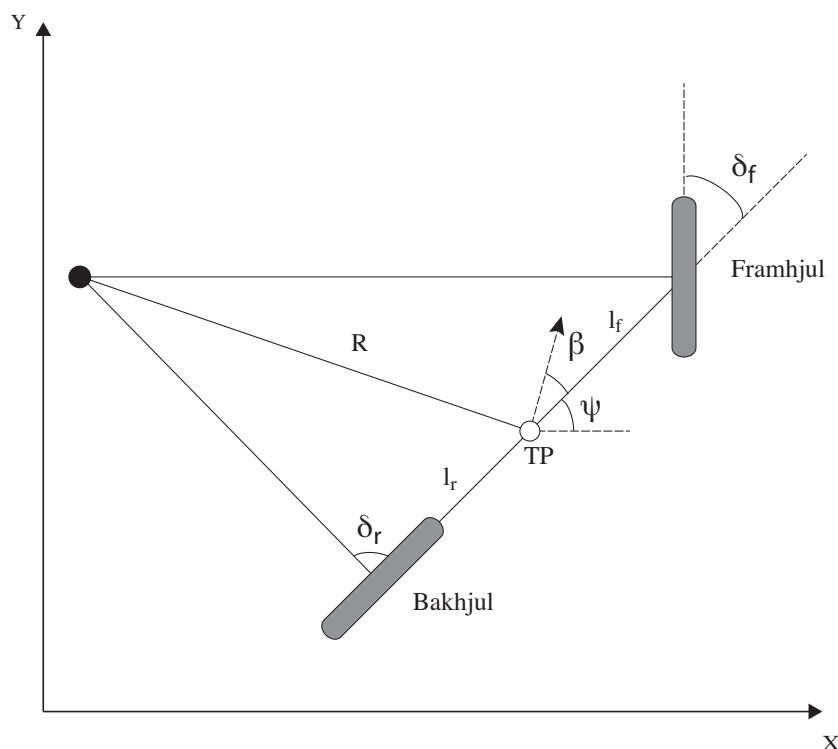
Målen omfattar såväl simuleringsmodellen som den fysiska bilen och skall uppnås i 9 fall av 10. För att kunna identifiera målen under arbetets gång har de numrerats.

1. Bilen skall kunna detektera en parkeringslucka.
 - (a) Skall körandes med en hastighet < 5 km/h kunna detektera luckan.
 - (b) Skall kunna detektera luckan ståendes på ett avstånd av 500 mm mitt framför luckan.
 - (c) Skall kunna beräkna luckans bredd och djup med 5 % marginal.
 - (d) Skall bedöma om luckans dimensioner gör den lämplig att parkera i.
 - (e) Om ingen lucka påträffas skall bilen stanna efter en given tid.
2. Bilen skall kunna parkera autonomt i den detekterade parkeringsluckan.
 - (a) Skall kunna följa en linje på ett konstant avstånd till de parkerade bilarna med en felmarginal på 10 % då den söker efter en parkeringslucka.
 - (b) Skall kunna stanna på en lämplig utgångsposition längs linjen med en felmarginal på ± 50 mm i både x- och y-led, då lämplig parkeringslucka har detekterats.
 - (c) Skall kunna parkera korrekt i mitten av parkeringsluckan, på så sätt att bilens mittpunkt ej avviker mer än 100 mm i sidled från parkeringsluckans centrumlinje.
 - (d) Skall stanna på ett avstånd ≥ 50 mm och ≤ 100 mm från bakre delen av luckan.
 - (e) Skall ha en vinkel $< 10^\circ$ i förhållande till parkeringsluckans centrumlinje.
3. Bilen skall klara av att undvika kollision.
 - (a) Skall kunna upptäcka framförvarande föremål vid detektering av luckan och stanna 800 mm ± 10 % från upptäckt föremål.
 - (b) Skall kunna backa in i luckan utan att kollidera med bakomvarande föremål.
4. Bilen skall klara av att genomföra en fullständig parkering
 - (a) Skall uppfylla huvudmål 1, 2, 3 och utföra en korrekt parkering.

2 Teori

2.1 Fordonsdynamik

Utifrån bilens uppmätta egenskaper, enligt förra årets kandidatarbete [2], kombinerat med teorier inom fordonsdynamik kunde en modell för bilens dynamik tas fram och flera storheter beräknas. Den modell som skapats utgår från en så kallad cykelmodell [3], vilken säger att ett fordonns två framhjul alltid har samma vinkel i förhållande till varandra. För att illustrera teorin och beräkningarna krävs viss nomenklatur, vilken visas och definieras i Figur 4.



Figur 4: Definition av vinklar i cykelmodellen

Enligt [3] gäller (1) och (2), där β är den så kallade *slip-vinkeln*, vilket är vinkeln mellan bilens riktungs- och dess hastighetsvektor V . Enligt Rajamani kan β antas vara lika med noll vid låga hastigheter ($< 5 \text{ km/h}$), vilket beslutats att ha vid parkering.

$$\tan(\delta_f) \cos(\beta) - \sin(\beta) = \frac{l_f}{R} \quad (1)$$

$$\sin(\beta) - \tan(\delta_f) \cos(\beta) = \frac{l_r}{R} \quad (2)$$

Genom att kombinera (1) med den uppmätta svänggradien R (1500 mm) kan den maximala hjulvinkeln beräknas:

$$\begin{aligned}\tan(\delta_f) \cos(\beta) - \sin(\beta) &= \frac{l_f}{R} \implies \{\beta = 0\} \\ \implies \tan(\delta_{f_{max}}) &= \frac{l_f}{R_{min}} \implies \delta_{f_{max}} = \arctan\left(\frac{l_f}{R_{min}}\right) = \\ &= \arctan\left(\frac{423}{1500}\right) \approx 15,7^\circ\end{aligned}\quad (3)$$

Utgående från det beräknade maximala utslaget på hjulen kan en minimal svängradie beräknas, då bilen backas. Det görs genom att samma modell som ovan används, men att vinkeln antas vara på bakhjulet istället för på framhjulet. Då bilen backar kommer de hjul som vanligtvis sitter fram att hamna bak, vilket i modellbilen kan jämföras med att bilens bakhjul vinklas, samtidigt som dess framhjul hålls fixerade. Den minimala svänggradien beräknas med hjälp av (2), vilket ger:

$$\begin{aligned}\sin(\beta) - \tan(\delta_f) \cos(\beta) &= \frac{l_r}{R} \implies \{\beta = 0\} \\ \implies -\tan(\delta_r) &= \frac{l_r}{R} \iff R = \left| \frac{l_r}{\tan(\delta_r)} \right| = \\ &= \frac{112}{\tan(15,7^\circ)} = 398,5 \text{ mm} \approx 400 \text{ mm}\end{aligned}\quad (4)$$

Alltså kommer bilens tyngdpunkt att ha en teoretisk svängradie på 400 mm, när bilen backar. Detta bör jämföras med den uppmätta svängradie på 1500 mm då bilen körs framåt.

Bilens absoluta rörelser i ett XY-koordinatsystem ges enligt:

$$\dot{X} = V \cos(\psi + \beta) \quad (5)$$

$$\dot{Y} = V \sin(\psi + \beta) \quad (6)$$

$$\dot{\psi} = \frac{V \cos(\beta)}{l_f + l_r} (\tan(\delta_f) - \tan(\delta_r)) \quad (7)$$

Genom att ange bilens hörn som positioner relativt tyngdpunkten, i en matris, kan bilen plottas i ett XY-plan. Vridningen på bilen ges av vinkeln ψ . Vilka koordinater hörnen på bilen får i det absoluta XY-planet erhålls genom att hörnmatrisen (en matris med bilens hörn) multipliceras med en rotationsmatris:

$$R = \begin{bmatrix} \cos \psi & -\sin \psi \\ \sin \psi & \cos \psi \end{bmatrix} \quad (8)$$

2.2 Positionsmodell

Modellen avser bilens justering i sidled till de parkerade bilarna, då bilen söker efter en parkeringlucka. Baserat på (6) och (7) ($\beta = 0$) ovan där

$$l = l_f + l_r \quad (9)$$

och

$$\tan(\delta) = \tan(\delta_f) - \tan(\delta_r) \quad (10)$$

ges ekvationerna:

$$\dot{y} = v \sin(\psi) \quad (11)$$

$$\dot{\psi} = \frac{v}{l} \tan(\delta) \quad (12)$$

Då maximal styrvinkel är $15,7^\circ$, ges felavvikelser på 2,58 % för sin-funktionen respektive 0,13 % för tan-funktionen. Dessa avvikelser anses vara så pass marginella att approximationer för små vinklar går att tillämpa. Vilket således ger:

$$\dot{y} = v\psi \quad (13)$$

$$\dot{\psi} = \frac{v}{l}\delta \quad (14)$$

Laplacetransform

$$Ys = v \cdot \Psi \quad (15)$$

$$\Psi s = \frac{v}{l}\delta \quad (16)$$

(15) och (16) $\implies$

$$\frac{Y}{\delta} = \frac{v^2}{ls^2} = G \quad (17)$$

2.3 Detektering och sensorplacering

Parkeringsutförandet kan delas in i två moment, där det första är detektering och positionering för parkering och det andra är genomförandet av parkeringen. Vid det första momentet bör följande kunna detekteras:

- En parkeringsluckas dimensioner.
- Position i förhållande till parkeringsluckan.
- Framförvarande hinder för att undvika kollision då bilen kör framåt.

Vid det andra momentet bör följande kunna detekteras:

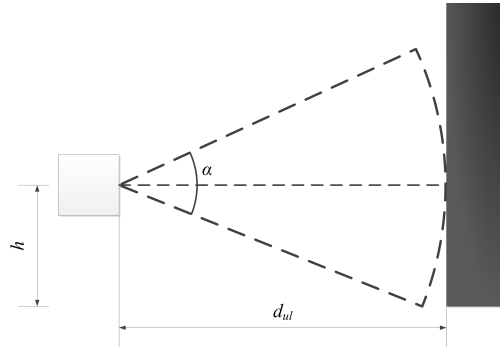
- Avstånd bakåt för att kunna slutföra parkeringen tillräckligt långt in i luckan.
- Avstånd i sidled vid två punkter längs långsidan för att visa att bilen står parallellt med övriga parkerade bilar och är placerad centrerat i parkeringsfickan.
- Närliggande föremål för att undvika kollision vid inbackning.

För att bestämma antalet sensorer och deras placering måste hänsyn tas till de yttre punkter på bilen som kan kollidera med parkerade bilar under parkeringsutförandet. De yttre punkterna är:

- Bakre hörn: Kan kollidera om bilen avviker från den planerade färdvägen.
- Bakre sida: Kan kollidera om bilen avviker betydligt från den planerade färdvägen samt om föremål finns rakt bakom.
- Längsidor och främre hörn: Kan kollidera om bilen svänger för mycket på väg in eller inne i luckan och vid korrigering av parkeringsinrättningen då bilen kör ut ur luckan.

Sensorernas placering i höjddled bestäms med hänsyn till det längsta avstånd som skall kunna detekteras korrekt. Det kritiska ögonblicket uppkommer när avståndet till objektet och underlaget är detsamma, vilket illustreras i Figur 5. Det detekterade avståndet d_{ul} visar avståndet till objektet. Om avståndet till objektet ökas kommer det detekterade avståndet förbli detsamma, eftersom underlaget detekteras istället för objektet. Den lägsta höjden, h , för sensorplacering, uttryckt i d_{ul} och spridningsvinkeln α , ges enligt (18):

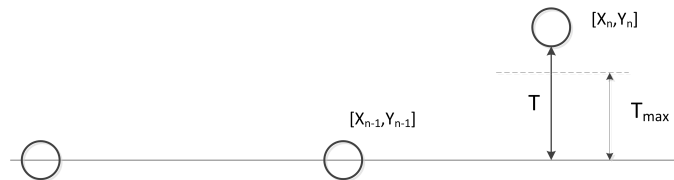
$$h = d_{ul} \cdot \tan \frac{\alpha}{2} \quad (18)$$



Figur 5: Ultraljudssensorns höjd betraktad från sidan

2.4 Line tracking-algoritmen

För att kunna tolka data i form av punkter, till användbar information om miljön runt bilen, används en algoritm som gör om dessa punkter till sammanhängande linjer. Som utgångspunkt används metoden beskriven i [4]. Algoritmen skapar en tänkt linje genom de två första punkterna från laserscannern och beräknar sedan avståndet, T , mellan nästa punkt och denna linje, se Figur 6. Om avståndet är mindre än ett bestämt avstånd T_{max} antas punkten höra till linjen och linjens slutpunkt uppdateras till den aktuella punkten. Om avståndet är större än T_{max} påbörjas en ny linje mellan slutpunkten för den förra linjen $[X_{n-1}, Y_{n-1}]$ och den aktuella punkten $[X_n, Y_n]$. När alla punkter är genomarbetade ökas T_{max} och algoritmen körs ytterligare gånger med de detekterade linjernas start- och slutpunkter som indata.



Figur 6: Avstånd mellan punkt och linje skapad av Line tracking-algoritmen

3 Metod

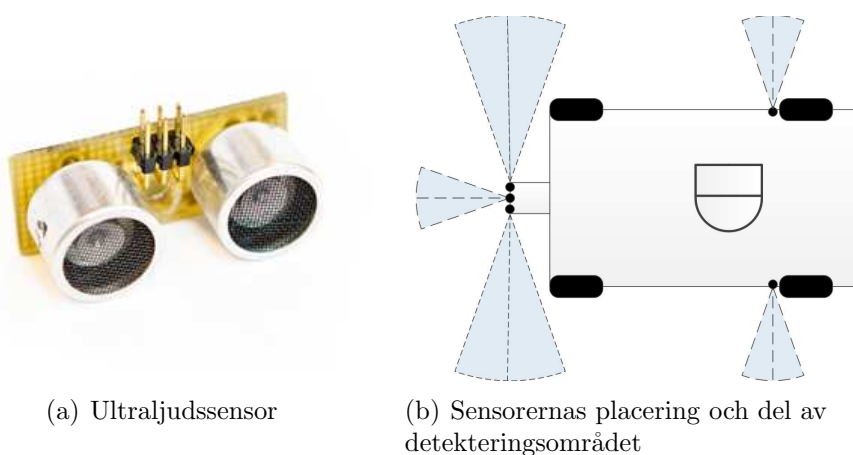
Nedan beskrivs projektets huvudsakliga metoder, designförfaranden och tester huvudsakligen indelade i två kategorier; simuleringsmodellen och den praktiska tillämpningen.

3.1 Design

3.1.1 Detektering och sensorplacering

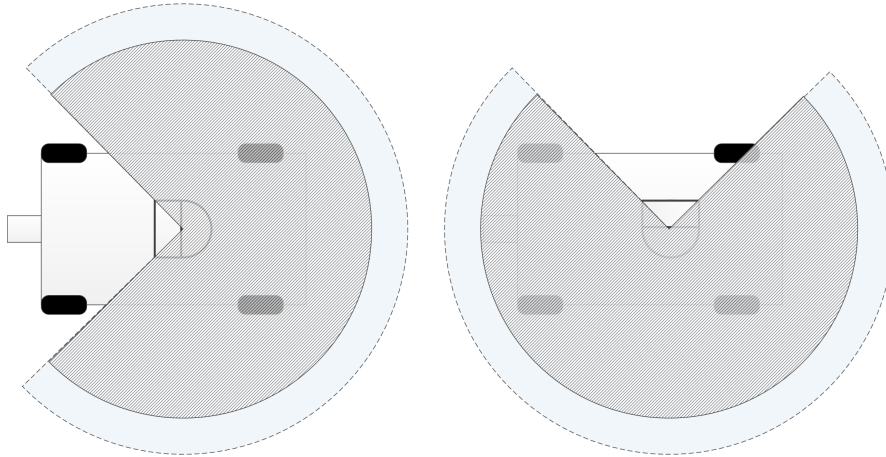
Bilen har utrustats med två typer av sensorer, den från start monterade laser-scannern och nymonterade ultraljudssensorer. Ultraljudssensorerna kompenserar för laser-scannerns begränsade synfält som inte innefattar avstånd $< 0,5$ m. Ultraljudssensorerna mäter avstånd till den närmsta punkten inom det egna synfältet. Sensorerna returnerar ett heltal mellan 0 och 255, vilket representerar ett avstånd i cm. Alla avstånd större än 255 cm anges med värdet 255. När en mätning begärs av en ultraljudssensor gör den ett antal mätningar och räknar därefter med hjälp av ett medianfilter ut ett värde.

Ultraljudssensorerna används under andra delen av parkeringsförfarandet och är placerade för att detektera omgivningen enligt stycke 2.3. Figur 7(a) visar en av ultraljudssensorerna. Fem ultraljudssensorerna är placerade enligt Figur 7(b). Då ultraljudssensorernas avståndsmätning större än 0,5 m är oväsentlig, ger ekvation (18), tillsammans med $d_{ul} = 0,5$ m och $\alpha = 40^\circ$ att ultraljudssensorerna måste placeras på en höjd $> 0,17$ m.



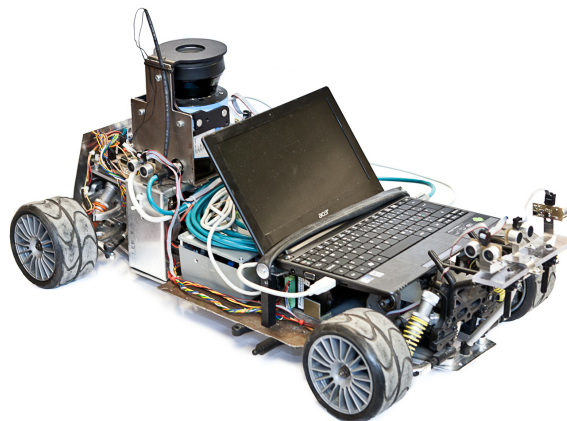
Figur 7: Fotografi av en ultraljudssensor (a) och illustration av sensorernas placering (b)

Laserscannern genomför svepningar i ett horisontellt plan och genererar ett flertal avståndsmätningar i olika riktningar. Antalet mätningar beror på vinkelupplösningen. Laserscannern används under första delen av parkeringsförfarandet och detekterar delar ur omgivningen enligt stycke 2.3. Laserscannerns position har ändrats enligt Figur 8, för att kunna mäta avstånd till parkeringsluckan då den passerats. Till vänster ses laserscannerns ursprungliga synfält och till höger det nya.



Figur 8: Laserscannerns synfält före (vänster) och efter rotation (höger), det streckade området representerar avstånd innanför laserscannerns närgräns, 0,5 m

Figur 9 visar bilen som den ser ut vid avslutat projekt, med bärbar dator, ultraljudssensorer och laserscanner monterade.



Figur 9: Bilen i aktuellt utförande

3.1.2 Konstruktion av styrsystem

Utifrån problemen med bilens ursprungliga hårdvara, se stycke 1.3, togs ett nytt styrsystem fram där ARM-processorn byttes ut. Vid konstruktion av detta utnyttjades redan implementerade gränssnitt. En bärbar dator av modell Acer Aspire D260 används för att utföra beräkningar och ta in data från sensorer. En USB-SPI-modul³ kombinerat med 2 stycken nivåkonverterare⁴ som omvandlar signalnivåer mellan 3.3 V och 5 V, används för att kommunicera med AVR1 och skicka styrkommandon. För att kommunicera med ultraljudssensorerna används en USB-UART-modul⁵ till vilken sensorerna är kopplade i en slinga, på grund av hårdvarubegränsningar. Figur 10 visar USB-SPI- och USB-UART-modulen.



Figur 10: USB-SPI-modulen (vänster) och USB-UART-modulen (höger)

³Sparkfun SPI Shortcut

⁴Sparkfun Logic Level Converter

⁵Sparkfun Breakout Board for FT232RL USB to Serial

3.2 Algoritmer

3.2.1 Detektering av parkeringslucka

Momenten detektering av parkeringslucka samt placering av bilen före backning sker med hjälp av en algoritm, för vilken ett övergripande flödesschema presenteras i Appendix A.2. Varje underfunktion i algoritmen använder en delmängd av datan från laserscannern.

Funktionen som detekterar objekt framför bilen utgår från X-koordinaterna i ett 50° synfält framför bilen. Utifrån scanners position på bilen, det minsta tillåtna avståndet till framförvarande objekt samt bromssträcka beslutar algoritmen om risk för kollision föreligger.

För att bilen skall kunna hålla ett bestämt avstånd i sidled till de parkerade bilarna mäts avståndet genom att först skapa linjer med Line tracking-algoritmen, vilken finns beskriven i stycke 2.4, och sedan filtrera bort oönskade linjer. Därefter beräknas avståndet till de parkerade bilarna utifrån det minsta ortogonala avståndet till de kvarvarande linjerna. Det uppmätta avståndet används sedan som insignal till en PID-regulator som styr bilens hjul.

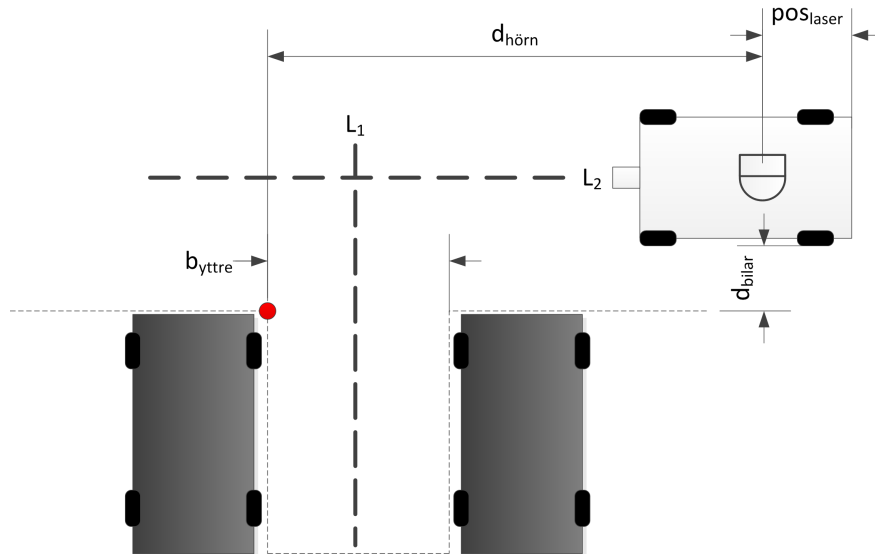
Algoritmen som detekterar och mäter parkeringsluckan använder sig av den tidigare presenterade varianten av Line tracking-algoritmen. För att mäta parkeringsluckans dimensioner används hörnen, vilka detekteras genom att beräkna vinklarna mellan alla sammanhängande linjer och därefter spara de punkter där vinkeln finns i ett intervall runt 90°. Luckans bredd beräknas sedan som det minsta av avstånden mellan de inre respektive yttre hörnen. Djupet beräknas som det största avståndet mellan det högra respektive vänstra hörnet.

Avståndet från bilen till parkeringsluckans centrumlinje (linje L_1 i Figur 11) beräknas utifrån parkeringsluckans yttre högra hörn (markerat i Figur 11). Avståndet till detta hörn, avståndet till de parkerade bilarna, scanners position på bilen samt parkeringsluckans yttre bredd ger avståndet till linje L_1 enligt:

$$d = \sqrt{d_{\text{hörn}}^2 - d_{\text{bilar}}^2} - \frac{b_{\text{yttre}}}{2} + pos_{\text{laser}} \quad (19)$$

3.2.2 Parkeringsutförandet

När bilen hittat en lämplig lucka och stannat på startpositionen kommer den att svänga mot parkeringsluckan och börja backa. Under denna procedur mäter varje ultraljudssensor avståndet till dess framförvarande objekt. När avståndet till ett objekt blir kritiskt stannar bilen i väntan på att objektet flyttar på sig så att den kan fortsätta sekvensen. Parkeringsluckans dimensioner har bilen sedan innan och kan med hjälp av sensorerna på sidan identifiera när den är parallell med parkeringsluckan. När detta sker rätas hjulen ut och bilen backar fram tills dess att en kritiskt avståndspunkt nås och den måste stanna. Flödesschema som beskriver parkeringsutförandet återfinns i Appendix A.2.



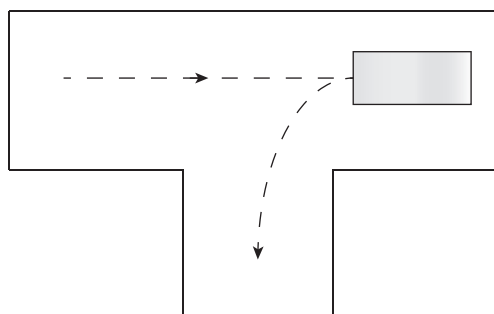
Figur 11: Definition av linjerna L_1 och L_2 . Parkeringsluckans konturer är markerade med streckad linje och det högra yttre hörnet (sett från bilen) är markerat med röd prick

3.3 Uppbyggnad av simuleringsmodellen

Parkeringsfunktionen har simulerats genom en teoretisk modell i **MATLAB-Simulink**, i vilken algoritmer för luckdetektering och parkering testats. **Simulink**-modellen är byggd för att likna en bil, men med skillnaden att styrvinkeln på båda framhjulen alltid är samma. När bilen kör framåt, ligger styrningen på bilens framhjul och när bilen backas, ligger styrningen istället på bakhjulen. Fördelen med detta synsätt är att samma modell då alltid kan användas. Teorin förklarar bilens svängradie som funktion av styrvinkeln. Från ACA 2011 [2] hämtades parametrar såsom vikt och mått i form av längd, axelavstånd och svängradie.

Genom att utgå från modellen i [2] och modifiera den efter de nya funktionaliteter som krävs för parkering, har bilen och dess funktioner simulerats. Funktioner för laserscannern och ultraljudssensorer har byggts till i modellen, samtidigt som en virtuell miljö har skapats i **MATLAB**, se Figur 12. Laserscannern ser och detekterar avstånd till väggarna i den i miljön. Ultraljudssensorerna detekterar avstånd till väggarna i **MATLAB**-miljön under backning in i parkeringsluckan. Modellen är byggd så att bilen kör rakt fram, stannar och sedan backar in i parkeringsluckan. Detta görs utan återkoppling från sensorerna.

Då det är många olika funktioner och delar som simulerats har dessa främst testats i enskilda funktioner, men också testats i en större huvudmodell, se Figur 31 och 32 i Appendix A.5.



Figur 12: Visualisering av MATLAB-miljön

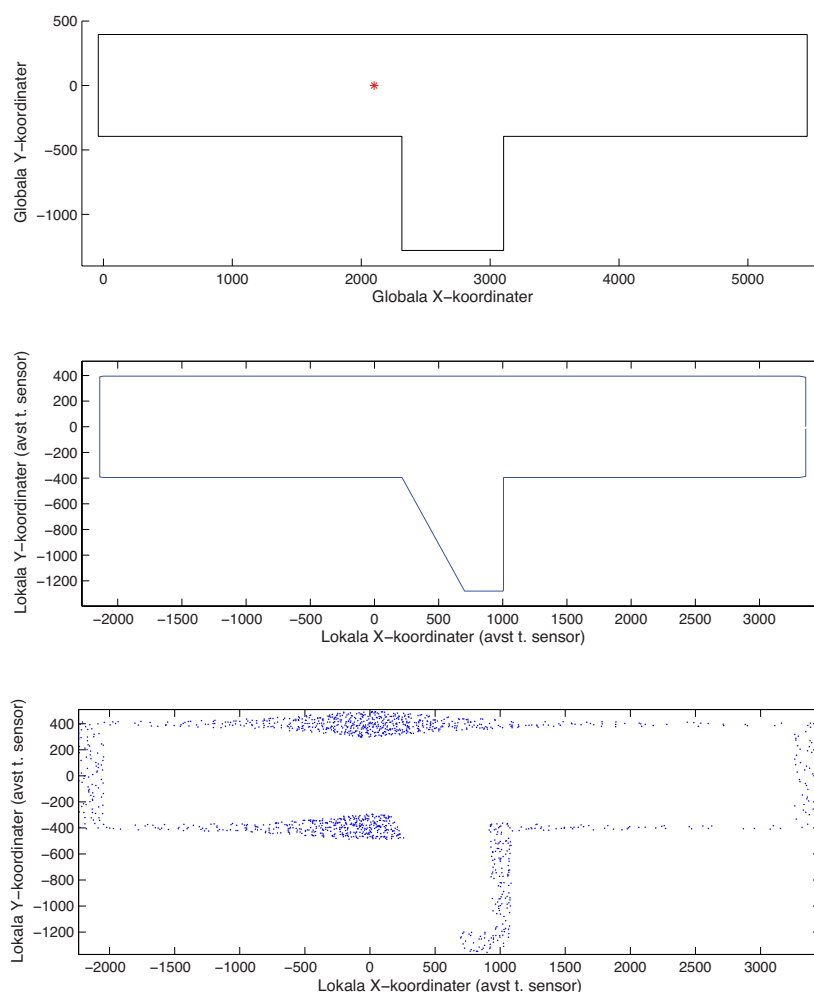
Simuleringen har till stor del inriktats på att ta reda på hur olika sensorer uppfattar verkligheten och vilken data som kan erhållas ifrån dem, samt hur denna skall behandlas. För detta har huvudmodellen i **Simulink** modifierats med **MATLAB**-funktioner som skapar simulerad sensordata så att denna kan plottas utan att alla olika körfall måste testas i verkligheten. I modellen har sensorernas data kunnat modifieras så att de uppmätta signalerna innehåller en önskad mängd brus och därmed kan brusreduceringsförmågan testas.

3.3.1 Sensorinformation

De olika typerna av sensorer använder sig i grunden av samma **MATLAB**-funktion för simulering och generering av mätdata, denna funktion kan anropas separat eller via huvudmodellen. För att visa vilken sensordata som erhålls vid användning av den ovan nämnda funktionen är det lämpligt att göra en 360° svepning runt den punkt där sensorn befinner sig. Notera dock att laserscannern inte gör ett helt varvs svepning, utan endast 270°. Den data som erhålls vid en sådan svepning presenteras i Figur 13. Genom att anropa denna funktion kontinuerligt kan värden för laserscannern erhållas genom hela simuleringen.

Resultatet från simuleringen av laserscannern visas i Figur 13. Den översta visar var sensorn (den röda asterixen) är placerad i den globala miljön och den mellersta plotten visar vad som erhålls då den uppmätta sensordatan omvandlas tillbaka till XY-positioner. Den understa plotten visar samma sak som den mellersta, men med skillnaden att datan störs genom att brus lagts till den. Värt att notera är att de understa plottarna visar avstånden i bilens lokala koordinatsystem, vilket överensstämmer med en verklig implementation av laserscannern.

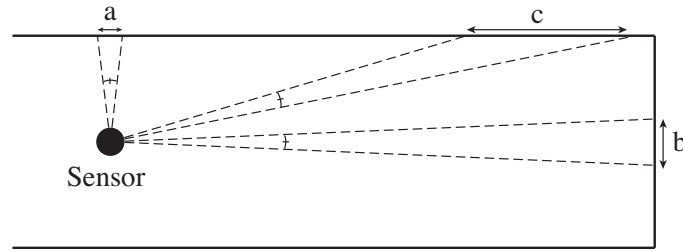
Som synes i Figur 13 är det väldigt många fler mätpunkter nära sensorn än längre ifrån den. Detta är inte att modellen ger felaktiga mätvärden, utan det är snarare så att antalet mätpunkter på väggen beror på vinkeln av väggen relativt sensorn, samt placeringen av väggen relativt sensorn.



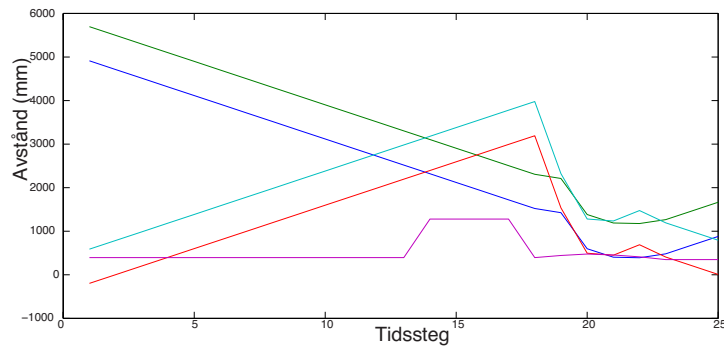
Figur 13: Sensordata från Laserscannern

Detta åskådliggörs tydligt i Figur 14, där avstånden a, b, c är i storleksordningen $a < b < c$, vid en konstant vinkelskillnad mellan första och sista punkten i intervallen.

Eftersom ultraljudssensorerna endast erhåller ett värde till närmsta vägg kan inte en plot likt den för laserscannern skapas. Här är det istället lämpligt att använda huvudsimuleringsmodellen för att visa vilken data sensorerna kontinuerligt generera då de åker genom miljön. I den praktiska implementationen används endast ultraljudssensorerna under parkeringsförfarandet. I modellen genereras istället data från ultraljudssensorerna kontinuerligt under körningen, vilket underlättar implementeringen av dem samt utformningen av programvara som skall tolka utdata från dem. Den data som erhålls vid en sådan körning av simuleringsmodellen finns att se i Figur 15.



Figur 14: Illustrationsskiss för att visa hur avståndet mellan två punkter på väggen vid konstant vinkel varierar beroende på väggens placering och riktning relativt sensorn

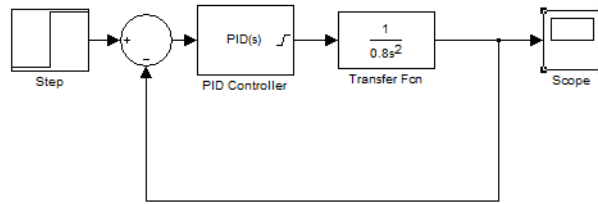


Figur 15: Sensordata från ultraljudssensorerna

3.3.2 Tester av simuleringsmodellen

En möjlig utgångsposition för inbackning testats fram. Genom att applicera teorin från fordonsdynamik, se stycke 2.1, på den erhållna datan kunde ett enklare **MATLAB**-program skapas vilket testade olika svängradier. Med hjälp av detta program kunde sedan en möjlig position testas fram. Dessutom kunde skillnader mellan att köra framåt, respektive backa in i parkeringsrutan testas. Utgående ifrån dessa tester fattades sedan beslut i hur bilen skulle köras vid parkeringsförfarandet.

Test av positionsreglering utfördes genom att simulera systemet, beskrivet i stycke 2.2. Blockschemat över den använda reglermodellen ses i Figur 16. Referensvärdet sattes till 0,5 m och startläge till 0,47 m. Parametrar sattes till $v = 1$ m/s och $l = 0,785$ m. De framtagna regulatorparametrarna används som utgångspunkt för testning av regleringen i verkligheten.



Figur 16: Blockschema över systemet G med PID-regulator

3.4 Tester av bilens funktioner

Tester har genomförts för att visa hur väl projektets mål och delmål har kunnat uppnås. Tester och mål korresponderar till varandra enligt ett nummersystem. Alla tester utfördes 10 gånger om inget annat anges.

1. Detektera parkeringslucka

- (a) Bilen kördes längs en rad med parkerade bilar på ett givet avstånd och sökte efter en tom parkeringslucka. Bilen kördes framåt med maximal hastighet och manövrerades med handkontroll.
- (b) Bilen placerades 500 mm mitt framför luckan och sökningar efter luckan utfördes då bilen var stillastående.
- (c) Bilen placerades 500 mm mitt framför parkeringsluckan. Som indata till varje mätning användes ett medelvärde av tio stycken laserscanningar, till skillnad från test 1b där enbart en mätning användes. Luckans dimensioner beräknades.
- (d) Bilen backades enligt test 2b och olika dimensioner på parkeringsluckor testades tills att parkeringen gick att utföra utan risk för kollision. En minsta möjlig storlek på parkeringsluckan kunde därmed bestämmas.
- (e) Bilen kördes autonomt framåt längs med en rad med parkerade bilar. Ingen parkeringslucka fanns utsatt. Bilen hade som indata att stanna efter en given tid. Om ingen lämplig parkeringslucka hittats vid den givna tidens slut stannar bilen.

2. Parkera autonomt i parkeringslucka

- (a) Testet utfördes i två delar. Den första delen mätte hur noggrant laser-scannern mäter avstånd till de parkerade bilarna och den andra delen hur väl bilen håller avståndet till bilarna. Bilen placerades 500 mm från raden med parkerade bilar. Avståndet till bilarna mättes. I den andra delen kördes bilen fram med hjälp av algoritmen för att hålla avstånd, se 3.2.1. Bilen körde en given tid och stannade.
- (b) Testet utfördes i två delar. I den första delen placerades bilen på en bestämd punkt i rummet. För att styra bilen användes handkontrollen. Hjulen svängdes maximalt och därefter backades bilen långsamt bakåt tills dess att den befann sig vinkelrät mot sin utgångsposition. En möjlig position i förhållande till parkeringsluckan kunde bestämmas. I den andra delen kördes bilen autonomt mot utgångspositionen och avvikelser från denna mättes då bilen stannat.
- (c) Bilen placerades på utgångspositionen och backade autonomt in i luckan. Avståndet från bilens mittpunkt till parkeringsluckans centrumlinje mättes.
- (d) Utfördes på samma sätt som test 2c, avståndet till bakre del av parkeringsluckan beräknades.
- (e) Utfördes på samma sätt som test 2c, avståndet från bilens hjul till parkeringsluckans sidor mättes och bilens vinkel beräknades.

3. Undvika kollision

- (a) Bilen kördes autonomt mot ett givet objekt. Avstånd till framförvarande objekt mättes då bilen bromsat in och stannat.
- (b) Bilen placerades på den tänkta utgångspositionen och backades in med hjälp av backfunktionen. Avstånd till bakomvarande föremål mättes.

4. Fullständig parkering

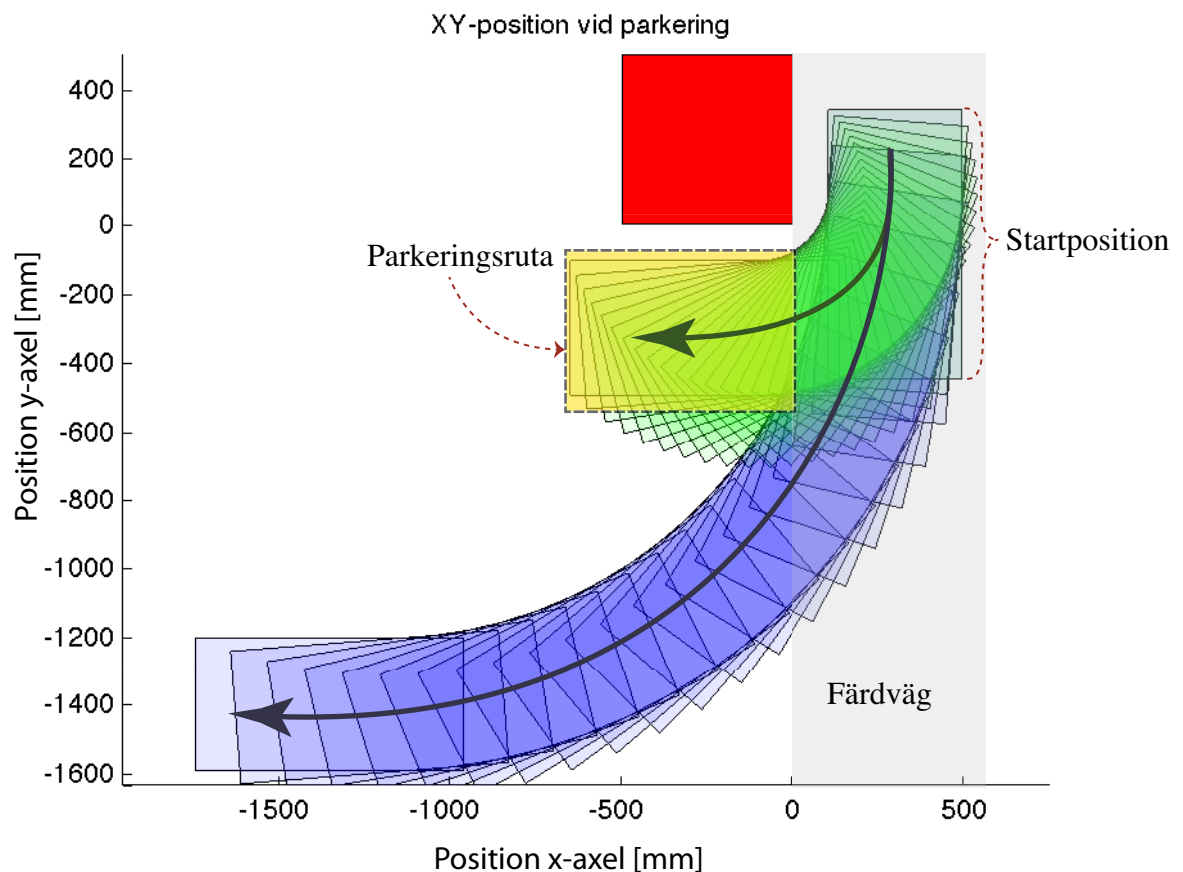
- (a) Bilen placerades 500 mm från de parkerade bilarna och 1 m innan parkeringsluckans högra hörn. Parkeringsfunktionen startades och ett uppehåll gjordes i programmet när bilen stannat på utgångspositionen för inbackningen och avvikelser från den bestämda positionen mättes. Därefter fortsattes programmet och bilen backades mot parkeringsluckan. Om parkeringen lyckades mättes avståndet från luckans centrumlinje och vinkeln i förhållande till luckan.

4 Resultat

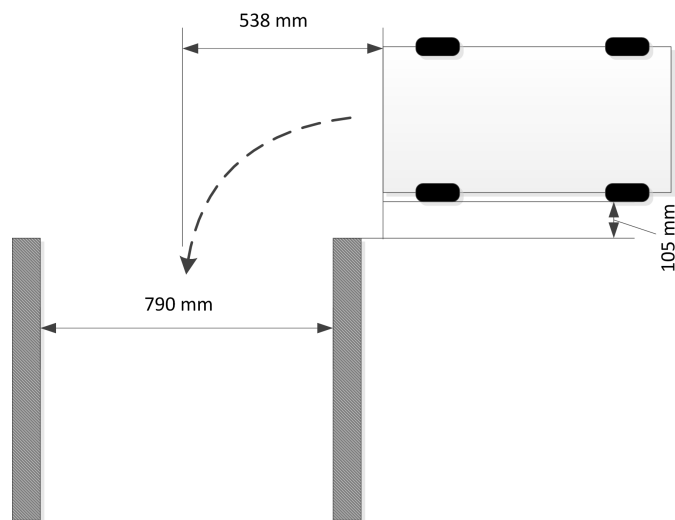
4.1 Simulering

4.1.1 Utgångsposition vid parkering

Figur 17 illustrerar hur stor skillnaden i svängradie är mellan en bil som backar och en som kör framåt. Figuren, som är framtagen i MATLAB, illustrerar också bilens optimala utgångsposition vid backning. Teorin för att beräkna utgångspositionen beskrivs närmare i stycke 2.1. Illustrationen visar skillnad mellan att backa och att köra framåt. I figuren är den röda kvadraten ett hinder, den blå bilen kör framåt och den gröna backar. Som Figur 17 visar är det fördelaktigt att backa in i luckan då svängradien i det fallet är mindre. Den simulerade optimala startpositionen är 105 mm ut från luckan och 538 mm bort från luckan, se Figur 18.



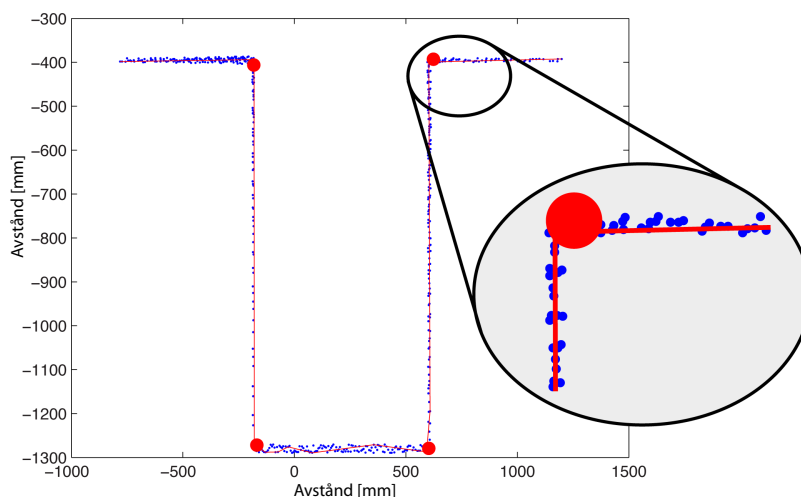
Figur 17: Utgångsposition vid parkering



Figur 18: Ideal position vid parkering i simuleringen

4.1.2 Detektering av parkeringsluckan

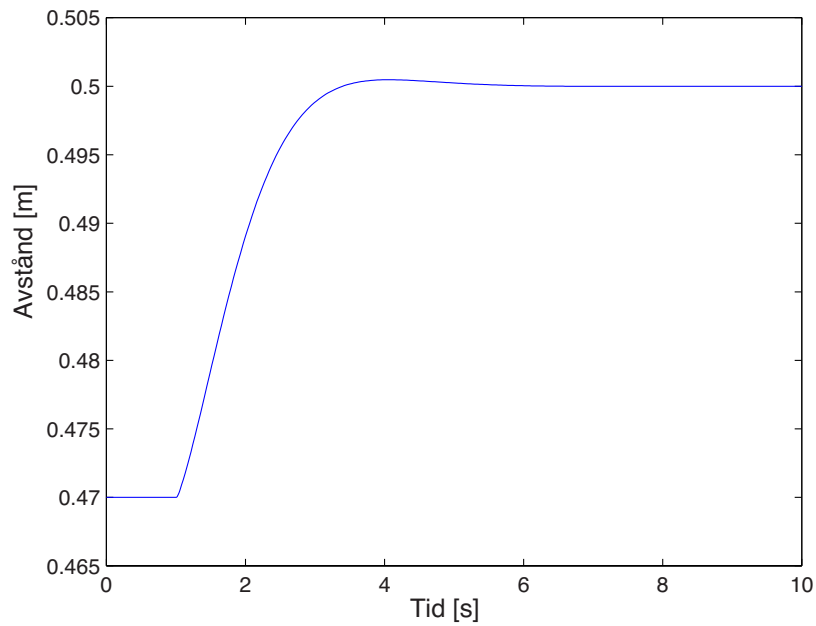
Figur 19 visar hur data som kommer från laserscannern kan tolkas. Figur 19 illustrerar linjer från Line tracking-algoritmen och detekterade hörn. Luckans bredd kunde mätas korrekt i 9 fall av 10 och med ett största mätfel på 8,3 %. Mätningarna av luckans djup var korrekt i 10 fall av 10 och det största mätfelet för luckans djup var 2,3 %. Resultatet av mätningarna presenteras i Appendix A.3.



Figur 19: Plottad utdata från Line tracking-algoritmen

4.1.3 Positionsreglering

Insvängningskurvan i Figur 20 visar resultatet från positionsregleringen i sidled. Efter 1 s påbörjas stegsvaret och efter ytterligare 5 s har bilen reglerat sin position från 0,47 m till 0,5 m i sidled till de parkerade bilarna. De mest gynnsamma parametrarna med avseende på stabilitet för PD-regulatorn var, $P = 2$, $D = 2$



Figur 20: Insvängningskurva för positionering i sidled

4.2 Bilen

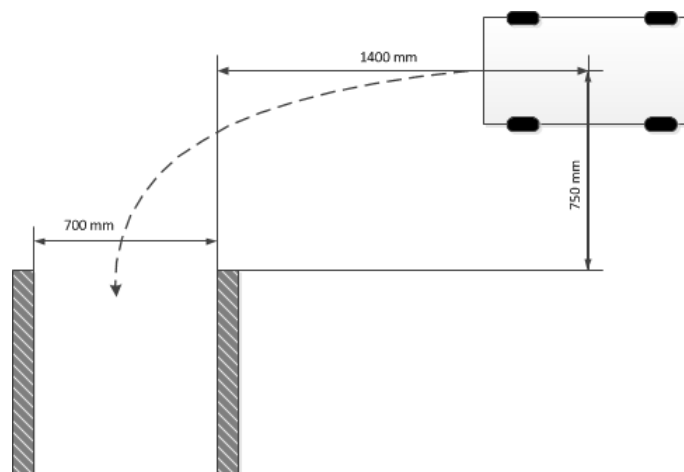
Tabell 1, 2 och 4 visar en översikt av resultaten från tester av respektive huvud- och delmål. Avvikelse presenteras i de fall det är möjligt. All mätdata presenteras i Appendix A.4.

4.2.1 Detektering av parkeringslucka

Tabell 1: Resultat från tester av huvudmål 1-*Detektera parkeringslucka*

Test:	Målvärde avvikelse	Måluppfyllnad	Största avvikelse
1a - Luckdetektering (Hastighet)	< 5 km/h	10/10	-
1b - Luckdetektering (Avstånd)	500 mm	10/10	-
1c - Luckans dimensioner	5 %	6/10	9,6 %
1d - Luckans lämplighet	-	10/10	-
1e - Stopptid	-	10/10	-

Test 1c resulterade i att parkeringsluckans dimensioner i 6 fall av 10 kunde mätas korrekt med 5 % marginal. I övriga fall var det maximala mätfelet 9,6 % av luckans bredd. I samtliga fall kunde luckans djup mätas med maximalt 2,2 % fel. Test 1d visade på en minsta lämplig storlek på parkeringsluckan; 0,7 m bred och 0,8 m djup.



Figur 21: Lämplig position i förhållande till parkeringslucka

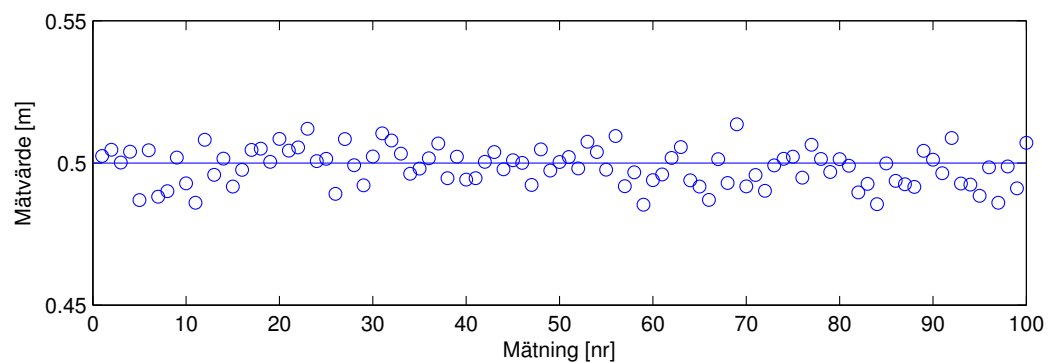
4.2.2 Positionering i förhållande till lucka

Tabell 2: Resultat från tester av huvudmål 2-*Parkera autonomt i parkeringslucka*

Test:	Målvärde avvikelse	Måluppfyllnad	Största avvikelse
2a - Hålla avstånd	10 %	0/10	17 %
2b - Utgångsposition	± 50 mm	4/10	Längsled: +670 mm Sidled: +200 mm
2c - Avstånd centrumlinje	100 mm	10/10	13 mm
2d - Avstånd bakåt	$50 \leq d_{bak} \leq 100$ mm	10/10	-
2e - Vinkel	$\leq 10^\circ$	10/10	1,89°

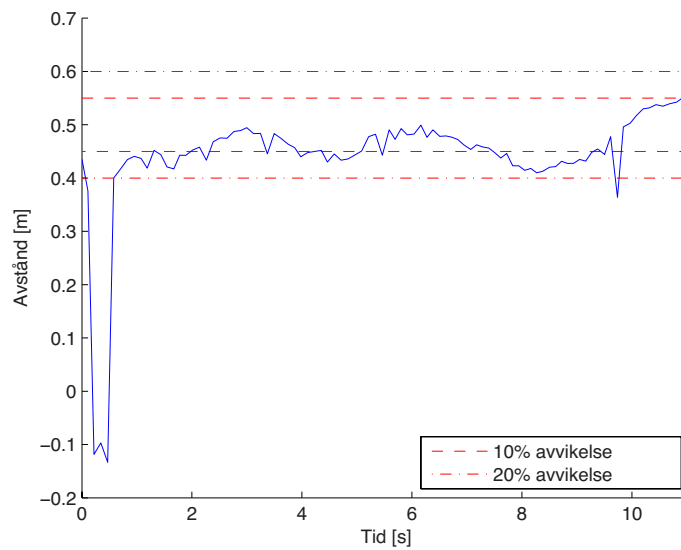
Den första delen av test 2a resulterade i ett medelavstånd till de parkerade bilarna. Medelavståndet och de 100 mätvärdena visas i Figur 22.

Första delen av test 2b resulterade i en möjlig placering av bilen för att kunna utföra parkeringen på ett försök, se Figur 21. Andra delen av testet resulterade i att bilen i 7 fall av 10 stannade för tidigt.



Figur 22: Mätning av avstånd till parkerade bilar

Utifrån andra delen av test 2a erhöles avståndsmätningar gjorda av bilen under testförfarandet, samtliga dessa mätningar presenteras i Appendix A.4. Testerna uppfyllde inte målet om att hålla avståndet med 10 % marginal i något av fallen. I 4 fall av 10 kunde bilen hålla avståndet med 20 % marginal, då uppenbart felaktiga mätningar bortsågs ifrån. I Figur 23 visas resultatet från det test där lägsta avvikelse från referensvärde erhöles. De negativa mätvärdena beror på att ett föremål snett bakom bilen detekterades istället för de parkerade bilarna.



Figur 23: Mätning av avstånd till parkerade bilar under körning med avståndsreglering

4.2.3 Undvika kollision

Tabell 3: Resultat från tester av huvudmål 3-*Undvika kollision*

Test:	Målvärde avvikelse	Måluppfyllnad	Största avvikelse
3a - Upptäcka framför	10 %	10/10	8,75 %
3b - Inbackning	-	10/10	-

Test 3b visade att bilen kunde parkera utan att kollidera när den placerades på utgångspositionen.

4.2.4 Fullständig parkering

Tabell 4: Resultat från tester av huvudmål 4-*Fullständig parkering*

Test:	Målvärde avvikelse	Måluppfyllnad	Största avvikelse
4a - Fullständig parkering	-	3/10	-

Test 4a lyckades i 3 fall av 10 slutföra hela parkeringsförfarandet och undvika kollision, i dessa fall uppfylls även målen om bilens slutposition. I 2 fall misslyckades parkeringsförfarandet totalt och i de övriga fallen stannade bilen i närheten av utgångspositionen men misslyckades med att backa in i parkeringsluckan. Fullständig mätdata finns i tabell 10 i Appendix A.4.

5 Diskussion

5.1 Simulering

Den optimala startpositionen beräknades med utgång från Rajamanis cykelmodell [3] i MATLAB. Denna beräkning visade den startposition som erhålls vid optimala förhållanden och då den reella bilen helt uppför sig enligt teorin. Det verkliga systemet har en viss tröghet, vilken inte tagits hänsyn till då en optimal startposition beräknats.

Den, i simuleringsmodellen, beräknade svänggradien är också optimal och antas inte förändras med yttre omständigheter. I verkligheten kommer den att variera beroende på friktionen mellan hjulen och underlaget. Detta antas dock inte påverka i de låga hastigheter som bilen färdas i. Enligt Rajamani [3] är denna skillnad i styrvinkel och riktningsvektor för bilen såpass liten i hastigheter under 5 km/h att den kan bortses ifrån. Då bilen har programmerats att färdas i hastigheter betydligt lägre än 5 km/h kan den nämnda friktionsvariationen göras.

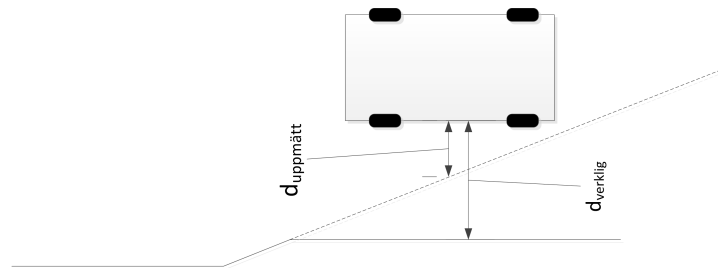
I simuleringsmodellen är alla sensorer idealt modellerade som standard. Dock kan deras signaler störas genom att brus läggs till deras signaler. Följden av att brus läggs till sensorernas signaler är att de mer efterliknar de verkliga sensorerna och deras utsignaler. Att brus kan läggas till och tas bort i modellen är viktigt eftersom det medför att flera olika aspekter av projektet kan testas. Om brus adderas kan filtrering och liknade testas. Om brus däremot inte läggs till kan grundläggande styrning och positionering testas utan implementation av filtreringsalgoritmer. Alltså kan flera osäkra delar av parkeringsutförandet testas innan alla algoritmer utvecklats. Dessutom kan plottar från system utan brus vara betydligt enklare att förstå då kritiska aspekter syns tydligt.

Ett tydligt exempel för hur brus påverkar modellen är detektering av parkeringsluckan. I en modell utan brus hittas alltid luckan, men genom att lägga på olika mängder brus kan algoritmen fås att missa att detektera luckan. På detta sätt har detekteringsalgoritmen kunnat testas.

5.2 Bilen

Den största anledningen till att parkeringsluckans bredd inte beräknas tillräckligt noggrant är med största sannolikhet Line tracking-algoritmens begränsningar i att skapa linjer som sammanfaller till hörn. Eftersom punkterna närmast hörnen ibland kan ligga på ett mindre avstånd än vad bruset vid mätning av en rak linje är, kan hörnpunkterna flyttas.

Regleringen som skall hålla bilen på ett avstånd till parkerade bilar missar att uppfylla målen av flera orsaker. För att kunna beräkna avståndet även när bilen befinner sig mitt framför luckan, måste ett brett område av laserscanningen användas så att inte luckans gränser missas. Om det i detta område finns linjer som är endast lite vinklade i förhållande till den linje som verkligen följer de parkerade bilarna finns risken att dessa inte filtreras bort. När en sådan linje är förhållandevis långt ifrån och vinklad mot bilen kommer det minsta avståndet från denna linje att vara avsevärt mindre än det verkligt sökta avståndet, vilket resulterar i en felmätning. Detta fenomen illustreras i Figur 24. Utöver detta skulle förmodligen ett bättre resultat kunna erhållas om ytterligare tid lades ner på att ställa in regulatören.



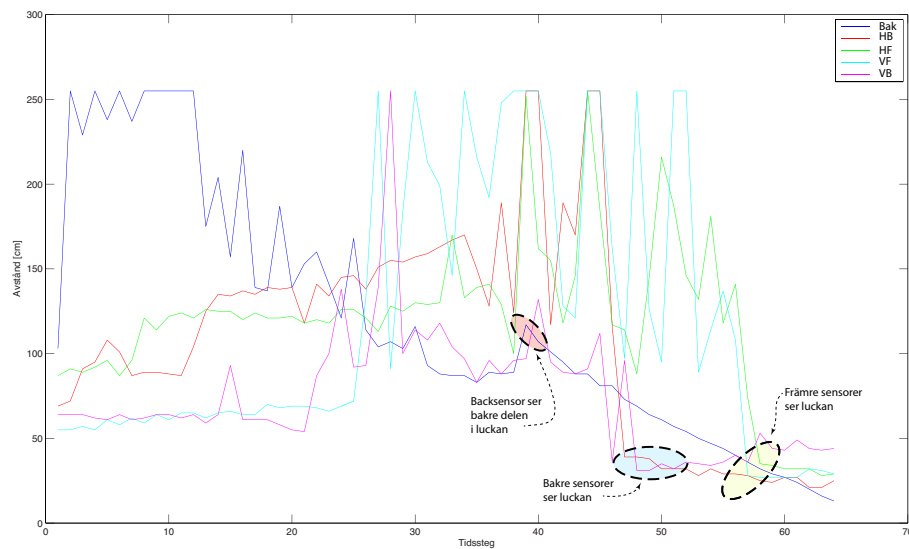
Figur 24: Felmätning av avstånd till parkerade bilar

Regulatören som reglerar avstånd till väggen och mäter parkeringsluckan kräver stor mängd indata. Den ursprungliga hårdvaran hade problem med dessa stora datamängder, vilket tidigare beskrivits i stycke 1.3. Därför ansågs det vara nödvändigt att ersätta delar av den gamla hårdvaran med ny, så att de uppsatta målen kunde nås. Valet föll på att ersätta den gamla hårdvaran med en dator så att de existerande begränsningarna byggdes bort.

5.3 Data från ultraljudssensorerna

I Figur 25 finns signaler från bilens ultraljudssensorer för backningsutförandet. Signalerna kommer från fyra identiska backningar som sedan medelvärdesbildats i MATLAB, för att ge en så rättvisande bild som möjligt. Som synes finns det en hel del brus i signalerna, men genom att studera figurerna väl kan ändå en hel del data utläsas ur dem. Bruset som syns i början kommer sig av att mätningarna är på

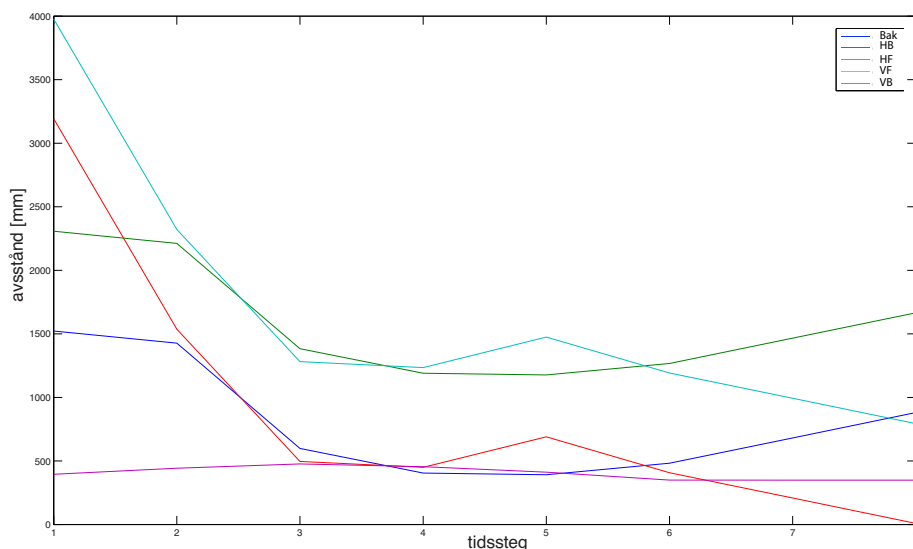
stora avstånd, för vilka ultraljudssensorerna har problem att generera tillförlitliga mätvärden. Dock kan det i slutet av parkeringsförfarandet ses att ju närmare sin slutposition bilen kommer desto mindre fluktuationer syns i signalerna från ultraljudssensorerna. Fluktuationerna beror på att sensorerna kommer närmare väggarna och därmed genererar säkrare mätvärden. Att signalerna från sensorerna är mindre brusiga då bilen är nära yttre objekt är viktigt, eftersom det är vid dessa avstånd som styrningen är baserad på signalerna från ultraljudssensorerna.



Figur 25: Sensordata för ultraljudssensorerna på bilen med märkning av när sensorerna identifierar väggarna i luckan och bruset minskar

Den brusiga data som erhålls från sensorerna kan jämföras med den simulerade, vilken visas i Figur 26. Här existerar inget brus. Dock är detta inte den enskilt viktigaste slutsatsen som kan dras vid jämförelse mellan dessa båda figurer. Istället är det viktigare att studera hur långt bilen har förflyttat sig mellan de individuella mätningarna under en mätningssykel.

I simuleringsmodellen registreras alla värden för de olika sensorerna på en gång, medan sensorerna i verkligheten är seriekopplade, så när ett mätvärde för en sensor har registrerats kan det inte processas med en gång, alla sensorer måste först mäta klart. Detta gör att bilen hinner flytta sig lite mellan att mätningen är genomförd tills dess att bilen kan styras baserat på värdet. Det här fenomenet är vitalt för den bakre sensorn på bilen. Programmeringsmässigt skall bilen stanna så fort denna registrerar ett avstånd < 150 mm. På 10 av 10 mätningar gjordes detta, men efter det att sensorn hade mätt flyttade bilen sig i snitt ytterligare 30-40 mm innan den stannade, vilket delvis beror på ultraljudssensorerna.



Figur 26: Sensordata för ultraljudssensorerna i simuleringsmodellen när bilen parkerar i fickan

Genom att öka antalet mätningar kan bruset från sensorerna reduceras, tyvärr leder detta i slutändan till att den genomsnittliga samplingstiden för varje sensor ökas. En för stor samplingstid kan leda till att sensorerna missar kritiska mätvärden i huruvida styrsystemet skall regleras eller inte. Följden av denna fördröjning kan bli att bilen för sent uppfattar om den är parallell med parkeringsluckan eller om den behöver stanna för att inte krocka med ett hinder. För att styrning av bilen baserat på information från ultraljudssensorer, skall vara så säkert och pålitlig som möjligt är det viktigt att sensorerna samplar så ofta och noggrant att det inte utgör en säkerhetsrisk.

En parallell uppkoppling eller en seriell uppkoppling med buffert skulle ha varit att föredra. Följden av detta är att man då inte behöver ha en så kallad dödtid då datorn väntar på data som skall behandlas från ultraljudssensorerna, utan kan då istället begära mätningar från samtliga sensorer på en och samma gång och minska samplingstiden avsevärt.

Den förflyttning av bilen som finns vid inläsningen från ultraljudssensorerna minimeras genom att backa in bilen så långsamt som möjligt i parkeringsrutan och genom att sköta processandet av data så effektivt som möjligt. I de genomförda fysiska testerna visade det sig att de vidtagna åtgärderna var tillräckliga.

5.4 Diskussion av skillnader och likheter mellan bilen och simuleringsmodellen

- Startposition

Den verkliga och den simulerade startpositionen för parkeringen skiljer sig markant åt. I simuleringsmodellen är den optimala startpositionen 538 mm från mitten på parkeringsluckan och 105 mm ut från de parkerade bilarna. Detta bör jämföras med värden för det verkliga parkeringsutförandet där den framtestade optimala positionen där bilen skall placeras 1900 mm från mitten på parkeringsluckan och 500 mm från de parkerade bilarna. Den stora skillnaden mellan simulerad och uppmätta är intressant, men om än något olycklig. Det bästa vore om denna skillnad inte existerade och slutsatser i simuleringen kunde litas på fullt ut. Nu krävs det istället att modeller och teorier testas i simuleringen för att se att det fungerar och sedan får de direkta värdena räknas om.

Faktorer som påverkar modellen och utfallet av den är framförallt indatan, men även vilken teori som simuleringen är baserad på. Den teori som valts att basera modellen på är en enkel cykelmodell. För ett mer korrekt resultat är det möjligt att det hade varit att föredra att använda den så kallade Ackermans styrmodell [3], vilken har olika styrvinklar på vänster och högerhjul, då dessa har olika svängradier. I detta arbete valdes det ändå att modellen skulle baseras på förra årets modell. Då det gjorts med gott resultat förra året drogs slutsatsen att det kunde göras även i år.

Andra källor till den stora skillnaden skulle kunna vara svårigheterna i att mäta svängradierna för projektbilen. Den svängradie som använts i simuleringen har erhållits från förra årets arbete och har bedömts vara rimlig.

Slutligen finns en viss tröghet i det verkliga systemet som inte finns med i simuleringsmodellen. Den verkliga bilen börjar inte svänga med fullt utslag direkt, till skillnad från den simulerade bilen, vilken direkt ger maximalt utslag i svängförfarandet då det önskas.

- Parkeringsalgoritm

Då simuleringen kördes visade det sig att bästa sättet att parkera bilen var att hitta den optimala startpositionen och därefter backa in i luckan med maximalt svängutslag. Detta visade sig även stämma i det verkliga parkeringsutförandet. I Figur 17 visas skillnaderna mellan att köra och backa in i parkeringsrutan. Denna skillnad fanns även i det verkliga systemet, dock med andra siffror på avstånden (även om relationerna mellan att köra framåt och bakåt fanns kvar).

- Detektering av parkeringslucka

Mätresultat från mätningar av parkeringsluckans dimensioner presenteras i Appendix A.3 för simuleringsmodellen och Appendix A.4 för bilen. Resultaten visar på en skillnad i mätresultat. Simuleringsmodellen presterade då den kunde i 9 fall av 10 mäta luckan korrekt och bilen klarade av att mäta luckan korrekt i 6 fall av 10. Anledningen till denna skillnaden kan bero på att de pålagda bruset i simuleringsmodellen inte motsvarar det verkliga mätbruset.

- Reglering

Den simulerade positionsregleringen skiljer sig avsevärt från regleringen av den verkliga bilen. I simuleringen används inte mätdata från laserscannerfunktionen vilket gör att mätstörningar ej påverkar resultatet. Funktionen som beräknar avståndet används inte heller i simuleringen och då denna ibland genererar mycket felaktiga data kommer denna skillnad göra att den simulerade regleringen skiljer sig från bilens verkliga beteende.

6 Slutsatser

Att tillämpa en autonom parkeringsfunktion för vinkelrät parkering på en bil har visat sig ge delvis tillfredsställande resultat, då lyckad parkering endast kunde utföras i 3 fall av 10. Bilen har under upprepade försök kunnat påträffa lämplig parkeringslucka och med god precision parkerat sig själv samtidigt som den har undvikit kollision med hinder såväl framtill som baktill. Bilen uppfyllde emellertid inte kravet på reglering vilket i första hand beror på bristande avståndsmätningar, i andra hand intrimning av regulatoren. Utvecklingen av parkeringsfunktionen har visat prov på betydelsen av robust hårdvara, med marginal för oförutsett behov av ytterligare kapacitet. Då de flesta delmomenten i parkeringsfunktionen utförs efter varandra kan felaktigheter fortplantas från en delfunktion till en annan, vilket ställer krav på hög noggrannhet och robusthet för att uppnå goda resultat genom hela parkeringsutförandet. För lyckat utfall av den autonoma parkeringen måste sensorernas mätsäkerhet och regleringen samverka på ett tillfredsställande sätt. För bästa möjliga resultat är det viktigt att lägga mycket tid på beräkningar och tester vid konstruktion av parkeringsfunktionen. Om inte riskerar parkeringen att misslyckas fatalt.

Med grund i stycke 5.3 kan slutsatsen dras att valet av seriell koppling av sensorer inte är ideal inom fordonsautomation, eftersom ökningen av samplingstiden för en ultraljudsensor leder till en totalökning av tidsåtgångens för samtliga sensorer inom slingan.

Resultat från både simulering av funktionen och tillämpning av densamma i verkligheten påvisar fördelar med förarlös parkering. Funktionen satisfierar delvis förväntningarna på att vara en säker lösning och att effektivisera parkeringen. Därmed bidrar funktionen till hållbar utveckling inom fordonsindustrin genom att undvika kollisioner och indirekt minska förbrukningen av såväl metaller som plaster.

6.1 Rekommendationer för framtida projekt

Den huvudmodell och de funktioner som användes för simulering i år kan även användas i framtiden. Framdrivningen är byggd på flera färdiga **Simulink**-block, dessa innehåller många onödiga delar som inte används och det gör modellen rörig. Rekommendationen är att ta bort framdrivningen och bygger en helt ny.

Framtida projekt rekommenderas att simulera alla ingående delar individuellt med möjligheten att slå samman dessa till en stor modell, vilket ger fördelar gentemot att direkt bygga en enda stor modell. I modellen är det också viktigt att matcha variabler och styrsignaler med mjukvaran i bilen, detta för att underlätta implementering av funktioner.

Microcontrollern som bland annat styr motor och servo av bilen bör bytas på grund av att kortet börjar bli föråldrat och är svårt att felsöka, samt att kortet har en begränsning av hur många I/O-portar som finns tillgängliga.

Kommunikationen mellan ultraljudsensorerna kan förbättras för att minska samplingstiden för en mätning. Detta kan lösas genom att koppla dem parallellt. Det kan dock inte göras utan att byta vissa delar av den befintliga hårdvaran. Eftersom datorn har en begränsning på antalet USB-portar, bör microcontrollern bytas för att få tillgång till fler UART-portar på samma controller.

Ett förslag till nästa års grupp är att implementera Kinectkameror vilket kan ge möjlighet att parkera efter linjer istället för lådor.

En möjlighet är också att lägga till funktionalitet för att kunna utföra en vanlig fickparkering. En viktig förutsättning för att kunna genomföra det är att förbättra svänggraden på bilen.

Referenser

- [1] Svenska Dagbladet. Många smiter från parkeringsskador (2008-03-27) Tillgänglig: http://www.svd.se/nyheter/inrikes/manga-smiter-fran-parkeringsskador_77193.svd. 2012-03-01
- [2] Algar, D., Lundbäck, F., Olaisson, F., Persson, J. och Wareborn, V. Autonomous Collision Avoidance. Göteborg: Chalmers tekniska högskola. 2011
- [3] Rajamani, Rajesh. Vehicle Dynamics and Control. New York: Springer. 2006
- [4] Borges, G.A och Aldon, M. Line Extraction in 2D Range Images for Mobile Robotics. Journal of Intelligent & Robotic Systems. 2004; 40(3): 267-297.

A Appendix

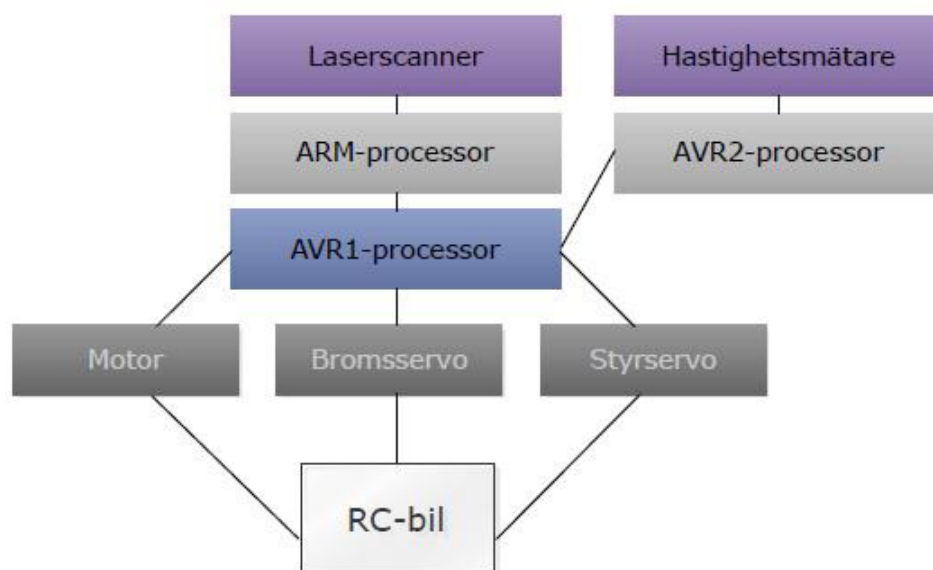
A.1 Ursprunglig hårdvara

Atmel Atmega168 Består av två 8-bit AVR-processorer, (AVR1 och AVR2) som gör kommunikationen mellan komponenterna möjlig. AVR1 är den processor som styr över fordonets viktigaste komponenter, styrservo, bromsservo och motor. AVR2 sköter beräkningen av fordonets hastighet med hjälp av pulsgivare.

Atmel AT91SAM7X Består av en 32-bit ARM-processor som ständigt sköter all beräkning och beslut som tas under driften med hjälp av de algoritmer som angivits till microprocessorn. ARM-processor sköter även mottagningen av data från lasersensorn via ett Ethernet protokoll (TCP/IP).

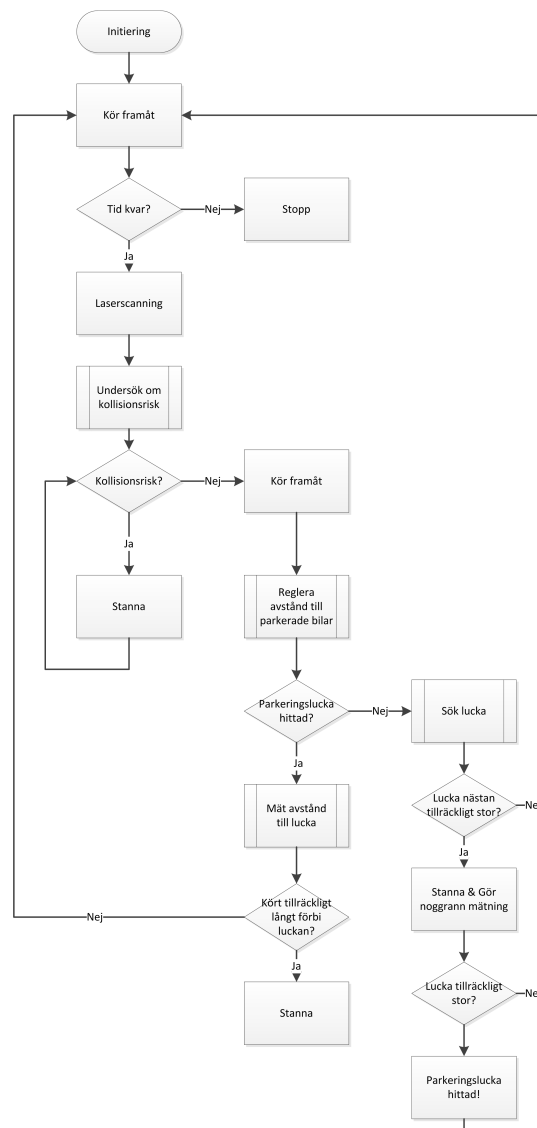
Arduino BT Består av en 8-bit AVR-processor och ett blåtandskort som gör det möjligt för processorn att skicka data via trådlös kommunikation. Kortet tar emot information från Atmel AT91SAM7X via en 10-bitars buss och kan sedan skicka vidare informationen trådlöst till en dator så att informationen kan presenteras visuellt genom cmd.

Figur 27 visar sambanden mellan de olika komponenterna.

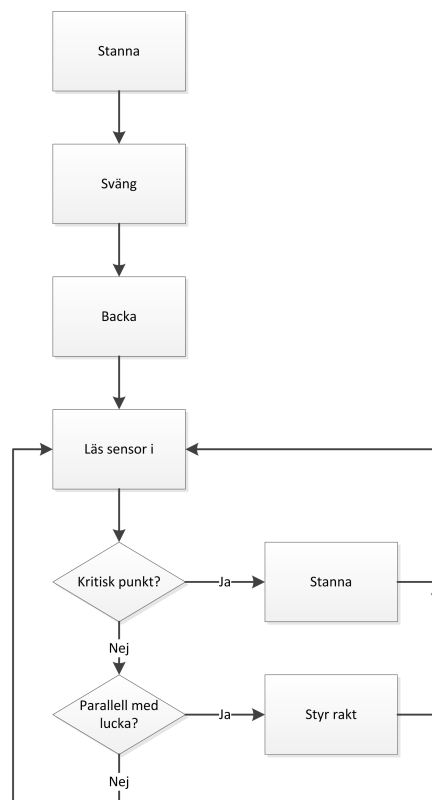


Figur 27: Samband mellan bilens hårdvara enligt [2]

A.2 Flödesscheman algoritmer



Figur 28: Flödesschema över algoritmen som hanterar parkeringsförfarandets första del



Figur 29: Flödesschema över algoritmen som hanterar parkeringsförfarandets andra del

A.3 Mätdata simuleringar

Tabell 5: Mätningar av parkeringsluckans bredd och djup i simulerings-modellen

	Bredd (m):	Fel(%):	Djup (m):	Fel(%):
Målvärde:	0,790	-	0,885	-
Uppmätt värde:	0,771	2,4	0,865	2,3
	0,773	2,1	0,887	0,2
	0,782	1,0	0,898	1,4
	0,780	1,2	0,892	0,7
	0,776	1,8	0,890	0,5
	0,729	8,3	0,897	1,3
	0,762	3,6	0,896	1,2
	0,782	1,0	0,886	0,1
	0,789	0,1	0,875	1,1
	0,771	2,4	0,896	1,2

A.4 Mätdata bilen

Tabell 6: Mätningar av parkeringsluckans bredd och djup

Målvärde:	Bredd (m):	Fel(%):	Djup (m):	Fel(%):
	0,700	-	0,810	-
Uppmätt värde:	0,674	3,7	0,810	0,0
	0,653	6,7	0,792	2,2
	0,680	2,9	0,818	1,0
	0,663	5,3	0,810	0,0
	0,682	2,6	0,820	1,2
	0,676	3,4	0,818	1,0
	0,672	4,0	0,798	1,5
	0,633	9,6	0,803	0,9
	0,674	3,7	0,810	0,0
	0,644	8,0	0,801	1,1

Tabell 7: Utgångsposition inför inbackning

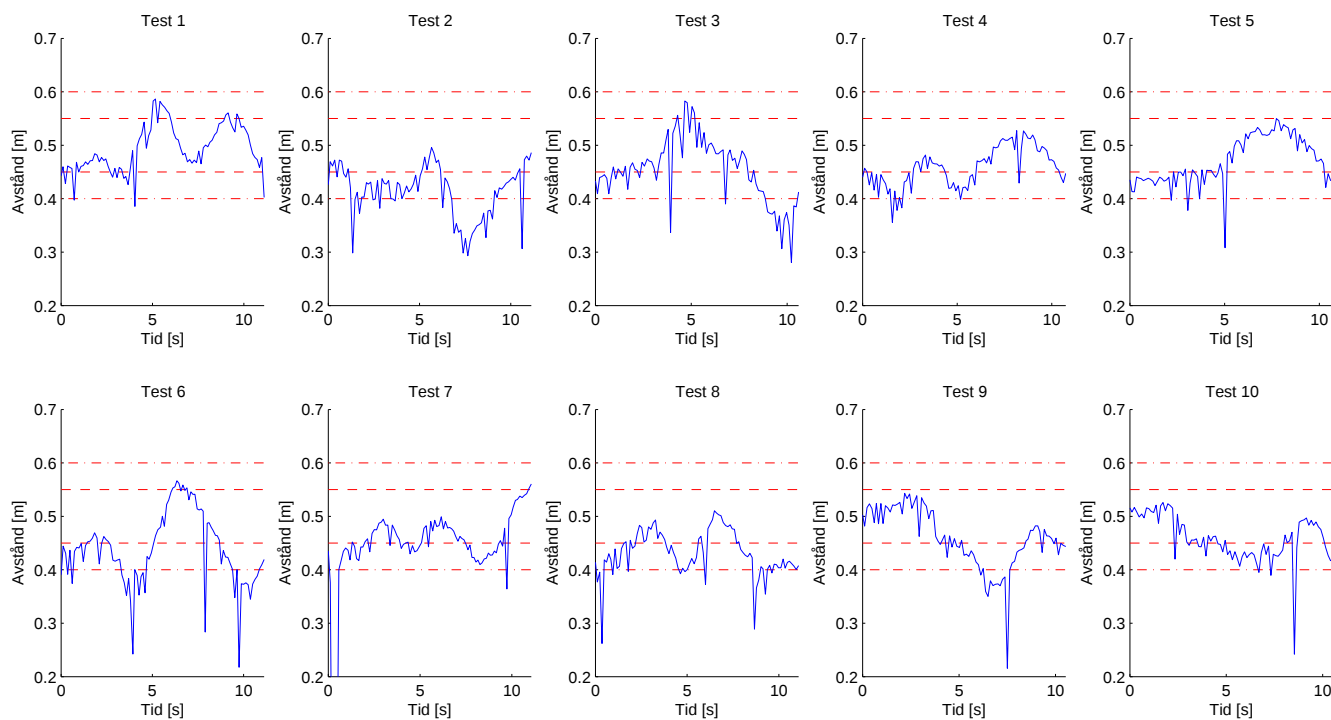
Målvärde:	Från centrumlinje (mm):	Fel (mm):	Från bilar (mm):	Fel (mm):
	1900	-	500	-
Uppmätt värde:	2000	100	480	20
	2300	400	510	10
	1930	30	460	40
	1960	60	560	60
	1930	30	510	10
	1940	40	400	100
	1890	10	600	200
	2570	670	560	60
	1910	10	500	0
	1930	30	480	20

Tabell 8: Avvikelse från optimal parkering

Parkering:	Avstånd centrumlinje (mm)	Avstånd bakre vägg (mm)	Grader (°)
1	13	50	1,89
2	11	90	1,62
3	4	60	0,54
4	4	70	0,54
5	4	50	0,54
6	5	70	0,81
7	6	60	0,81
8	0	60	0
9	13	70	1,89
10	6	80	0,81

Tabell 9: Avstånd till framförvarande hinder

Avstånd från hinder (m):	Avvikelse från målvärde 0,8 m (%)
0,84	5,00
0,86	7,50
0,83	3,75
0,85	6,25
0,87	8,75
0,78	2,50
0,81	1,25
0,82	2,50
0,77	3,75
0,85	6,25

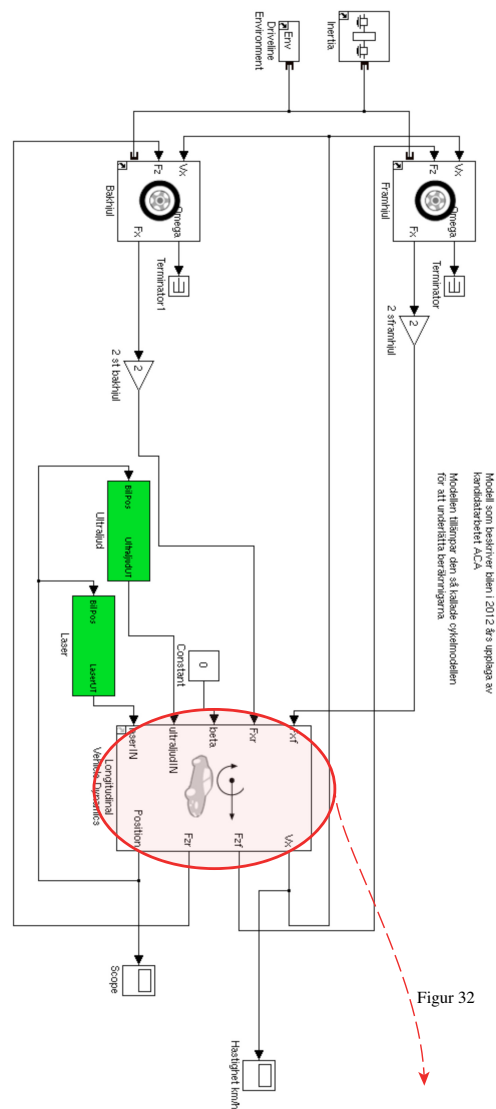


Figur 30: Avståndsmätningar under test av reglering

Tabell 10: Test av fullständig parkeringsfunktion

Försök	Avvikelse från utgångsposition		Mittavstånd (mm)	Vinkel (°)	Bakavstånd (mm)
	Från centrumlinje (mm)	Från bilar (mm)			
1	80	-200	100	3.24	90
2	-50	150	-	-	-
3	-	-	-	-	-
4	100	-80	30	0	80
5	-	-	-	-	-
6	70	70	-	-	-
7	-	-	-	-	-
8	45	100	-	-	-
9	50	30	25	1.62	60
10	35	75	-	-	-

A.5 Blockscheman simuleringsmodellen



Figur 31: Huvudvy för simuleringsmodellens blockschema

