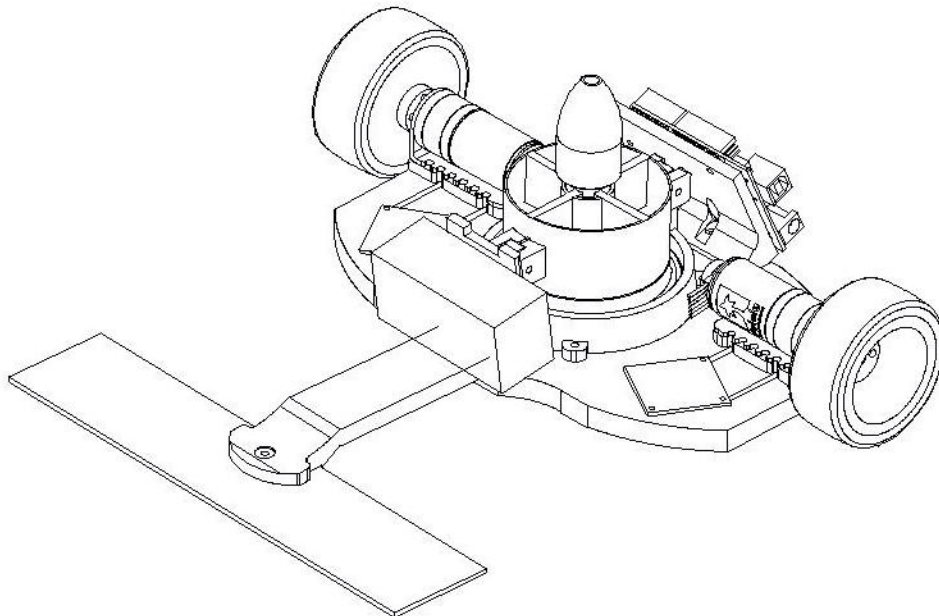




CHALMERS



Designoptimering av en Linjeföljande robot

Utveckling av en linjeföljande robot med avsikt att optimera dess förmåga att på kort tid tillryggalägga en linjerad sträcka

Kandidatarbete vid Signaler och system

VIKTOR ANDERSSON
DAVID DEURELL
ROBIN HAFSTRÖM

EMELIE JOHANSSON
KRISTOFFER LINDH
SAMUEL WILHELMSSON

KANDIDATARBETE 2017

Designoptimering av en Linjeföljande robot

Utveckling av en linjeföljande robot med avsikt att optimera dess förmåga på kort tid tillryggalägga en linjerad sträcka

Viktor Andersson
David Deurell
Robin Hafström
Emelie Johansson
Kristoffer Lindh
Samuel Wilhelmsson



CHALMERS

Institutionen för Signaler och system

SSYX02-17-06

CHALMERS TEKNISKA HÖGSKOLA

Göteborg, Sverige 2017

Designoptimering av en Linjeföljande robot

Utveckling av en linjeföljande robot med avsikt att optimera dess förmåga att på kort tid tillryggalägga en linjerad sträcka

©

Viktor Andersson

David Deurell

Robin Hafström

Emelie Johansson

Kristoffer Lindh

Samuel Wilhelmsson, 2017.

Handledare: Paolo Falcone, Signaler och system

Examinator: Yiannis Karayiannidis, Signaler och system

Kandidatarbete 2017

Institutionen för Signaler och system

Chalmers tekniska högskola

412 96 Göteborg

Telefon +46 31 772 1000

Framsida: 3D-modelerad bild av den färdigframställda linjeföljanderoboten

Typsatt i L^AT_EX

Göteborg, Sverige 2017

Sammanfattning

Rapporten behandlar ett kandidatarbete på institutionen för Signaler och system på Chalmers tekniska högskola där en autonom linjeföljande robot togs fram. I projektet lades fokus på simulering i syfte att färdigställa och verifiera designen tidigt i projektets gång. Simuleringen underlättar även kravsättning av komponenter och optimering av reglerparametrar. Ett par olika regleringsalgoritmer undersöktes och utvärderades. Av de undersökta algoritmerna valdes Pure Pursuit. Algoritmen användes vid framtagning av simuleringsmodell och vid programmeringen av roboten för att se till att det verkliga systemet avviker minimalt från simuleringen. En del saker tar simuleringsmodellen ej hänsyn till, som till exempel effektförluster och transportfördröjningar detta kompenseras för under konstruktionen av den verkliga roboten. Under konstruktion användes metoder som 3D-printing och etsning för att ta fram robotkomponenter. Den färdiga roboten skall dessutom tävla i Robot SM 2017 och uppfylla de krav och regler som tävlingen ställer.

Abstract

This project has been carried out at Chalmers University of Technology with the goal of developing an autonomous line following robot. The project focus has been on making a rigorous simulation to verify the design early on in the project. The simulation facilitates quick specification of components and easy optimization of control system parameters. A number of path-tracking algorithms were examined but only one was used. The algorithm chosen was the Pure Pursuit algorithm. The algorithm was used in the simulation and in the robot programming to ensure that the robot behavior won't deviate from the simulation. The simulation does not consider inefficiencies nor delays inherent in the system, this was however considered during the realization of the actual robot. During construction methods like 3D-printing, etching were used for robot components. The finished robot is going to enter the 2017 Swedish Robotics Championship and pass the rules and requirements required to enter the competition.

Nyckelord: Reglerteknik, Linjeföljande robot, Simulering, Produktutveckling, Autonom

Förord

Vi vill att tacka Institutionen för Signaler och system, Jonas Fredriksson och vår handledare Paolo Falcone för den hjälp och kunskap som de gett oss under projektets gång. Vår examinator Yiannis Karayiannidis för givande feedback på våra rapporter och för att han tagit sig tiden att läsa igenom projektet. Vi vill också passa på att tacka Rebecka Andersson för hjälp vid beställningar av varor. Även ett tack till E-sektionens Teletekniska Avdelning (ETA) för att vi har fått vara där och bygga samt ta del av föreningens kunskap och idéer. Dessutom riktas ett särskilt tack till *Lawicel* för deras hjälpsamhet både vid beställningar och i stundens hetta.

Tack!

Viktor Andersson
David Deurell
Robin Hafström
Emelie Johansson
Kristoffer Lindh
Samuel Wilhelmsson

Maj 2017, Göteborg

Ordlista

autonom självständig, självstyrande, oberoende. 5

Catia V5 Program för 3D-ritningar. 15

centripetalkraft är den yttre kraft som får ett föremål att följa en cirkulär bana med en konstant rotationshastighet och är riktad mot den cirkulära banans centrum. 7, 14

DC-motor Likströmsmotorn (likspänningsmotor, Direct current) är genom sin enkla princip en av de vanligaste typen av elektrisk motor. Motorn matas med likriktad elektrisk ström vars energi omvandlas till rörelseenergi.. 11

downforce Nedåtrikad kraft. 14, 38

encoder I denna rapport syftar detta till en enhet som används för att beräkna motoraxelns rotation. 25, 37

friläggning Att skapa en bild av hur yttre krafter angriper en partikel eller ett objekt. 7-9

KiCad är ett programpaket med öppen källkod för elektronisk designautomation, design av kopplingsschema och mönsterkort. 15

oscillering pendla, svänga, vibrerar, darra. 5, 37, 38

rotationscentrum Den punkt systemet roterar runt. 6, 30, 38

simulink är ett kommersiellt verktyg för modellering, simulering och analysering av dynamiska system. 13

simulinkmodell matematiskmodell gjort i programvaran simulink. 13

translationssystem System vars syfte är att förflytta, skapa translation. xiii, 3

unicycle model En slags modell vanlig för robotar där två hjul placeras på var sida av roboten. 9

Akronymer

ETA E-sektionens Teletekniska Avdelning. vii, 16

FTC Follow the Carrot. 5

LA Look-ahead. 5, 21

PP Pure Pursuit. v, 5, 6, 13

Innehåll

Figurer	xvii
Tabeller	xix
1 Inledning	1
1.1 Syfte	1
1.2 Problemformulering	1
1.3 Delproblem	1
1.3.1 Hitta linjen	2
1.3.2 Utforma chassi	2
1.3.3 Design av styrsystem	2
1.3.4 Styrning	3
1.3.5 Translationssystem	3
1.4 Avgränsningar	3
2 Teoribakgrund till robotdynamik och reglering	5
2.1 Reglering	5
2.1.1 Follow the carrot	5
2.1.2 Pure pursuit	6
2.2 Dynamik	6
2.2.1 Krafter vid acceleration	6
2.2.2 Krafter vid kurvtagning	7
2.2.3 Ökning av normalkraft	8
2.2.4 Beräkning av friktionskoefficientet	8
2.2.5 Stegning	9
2.2.6 Differentialstyrning	9
2.2.7 DC-motorer	11
3 Metod & Genomförande	13
3.1 Simulering	13
3.1.1 Simulinkmodell	13
3.1.1.1 DC-motorer	13
3.1.1.2 Sensorrigg	13
3.1.1.3 Roboten	14
3.1.2 Tester av lösningar	14
3.1.2.1 Optimeringsvariabler	14
3.1.2.2 Reglering	14

3.2	Konstruktion	15
3.2.1	Motivering av inköp	15
3.2.2	Tillverkning av komponenter	15
3.2.3	Sensorrigg	15
3.2.4	Montering av komponenter	16
3.3	Programmering	16
3.3.1	Arduino	16
3.3.2	Motorkontroller	16
3.3.3	Blåtandskommunikation	16
3.4	Utvärdering och tester	17
3.4.1	Testbana	17
3.4.2	Mätning av friktion	17
3.4.3	Mätning av ökad kraft på bakdäck med olika fläkthastigheter	17
3.4.4	Jämförelse av simulinkmodell och verklighet	18
4	Resultat	19
4.1	Simulering	19
4.1.1	Val av algoritm	21
4.1.2	Look-aheadavstånd	21
4.1.3	Antal sensorer	22
4.1.4	Reglering	22
4.1.4.1	P-regulator	23
4.1.4.2	Motorkontroller	24
4.2	Konstruktion	25
4.2.1	Färdiga komponenter	25
4.2.1.1	Motorer	25
4.2.1.2	Motorkontroll	26
4.2.1.3	Batteri	26
4.2.1.4	Fläkt	26
4.2.1.5	Däck	26
4.2.1.6	Mikrokontroller	26
4.2.1.7	IR-sensorer	26
4.2.1.8	Monteringskomponenter	26
4.2.2	Tillverkning	27
4.2.2.1	Bottenplatta	27
4.2.2.2	Mikrokontrollerfäste	27
4.2.2.3	Fläkthållare	28
4.2.2.4	Sensorarm	28
4.2.2.5	Batteri-hållare	29
4.2.2.6	Fälgar	29
4.2.2.7	Sensorrigg	30
4.2.3	Montering av komponenter	31
4.3	Arduinoprogram	32
4.4	Utvärdering och tester	34
4.4.1	Körning på Testbana	34
4.4.2	Mätning av friktion	34

4.4.3	Mätning av ökad kraft på bakdäck med olika fläkthastigheter	35
4.4.4	Jämförelse av simulinkmodell och verklighet	35
5	Diskussion	37
5.1	Simulering	37
5.2	Komponentval	37
5.3	Konstruktionsbeslut	37
5.4	Utvärdering och tester	38
6	Slutsats	41
	Litteraturförteckning	43
A	Kravspecifikation	I
B	Regler för Robot SM	III
C	Ritning för robot	VII

Figurer

1.1	Uppdelning av delproblem	2
2.1	Demonstration av Follow the Carrot	5
2.2	Demonstration av Pure Pursuit	6
2.3	Partikelrörelse i kurva	7
2.4	Friktionskraft	8
2.5	Friktionskraft i vinkel	9
2.6	Roboten sett som en balk	9
2.7	Vinkelhastighet	10
3.1	Testtrigg för mätning av ökad normalkraft på däck	17
4.1	Testkörning	20
4.2	Körning med varierande Look-ahead	21
4.3	Körning med varierande antal sensorer	22
4.4	Körning med varierande K_p	23
4.5	Grepp och effektkurvor med varierande PID-parametrar	24
4.6	Signalaktivitet med varierande PID-parametrar	25
4.7	Bottenplatta	27
4.8	Mikrokontrollerfäste	28
4.9	Kon för fläkt	28
4.10	Klämfäste för kon	28
4.11	Sensorarm	29
4.12	Batterihållare	29
4.13	Fälgar	30
4.14	KiCad design och färdigt kretskort	30
4.15	Sprängskiss av hjul- upphängning	31
4.16	Monterad hjulupphängning	31
4.17	Sprängskiss av fläkt med hållare	31
4.18	Monterad fläkt med hållare	31
4.19	Sprängskiss av monterad sensorarm	32
4.20	Monterad sensorarm	32
4.21	Sprängskiss av robot	32
4.22	Monterad robot	32
4.23	Testbana	34
4.24	Ökad kraft på bakdäck	35
4.25	Testkörning med simulering och robot	36

Tabeller

4.1	Tabell över köpta komponenter till konstruktionen	25
4.2	Tabell över uppmätta friktionskoefficienter	35

1

Inledning

Den här rapporten beskriver arbetet och resultaten av utveckling av en linjeföljande robot som kandidatarbete på Chalmers tekniska högskola. Projektet går ut på att konstruera och programmera en linjeföljande robot enligt de regler som ges av Robot-SM 2017, se appendix B, som anordnar tävlingen linjeföljning där en robot ska kunna köra en bana på så kort tid som möjligt och hela tiden överlappa linjen.

Minst ett tidigare projekt har gjorts om linjeföljande robot på Chalmers Tekniska Högskola 2015 där man avsett att ställa upp i robot-SM. [1].

1.1 Syfte

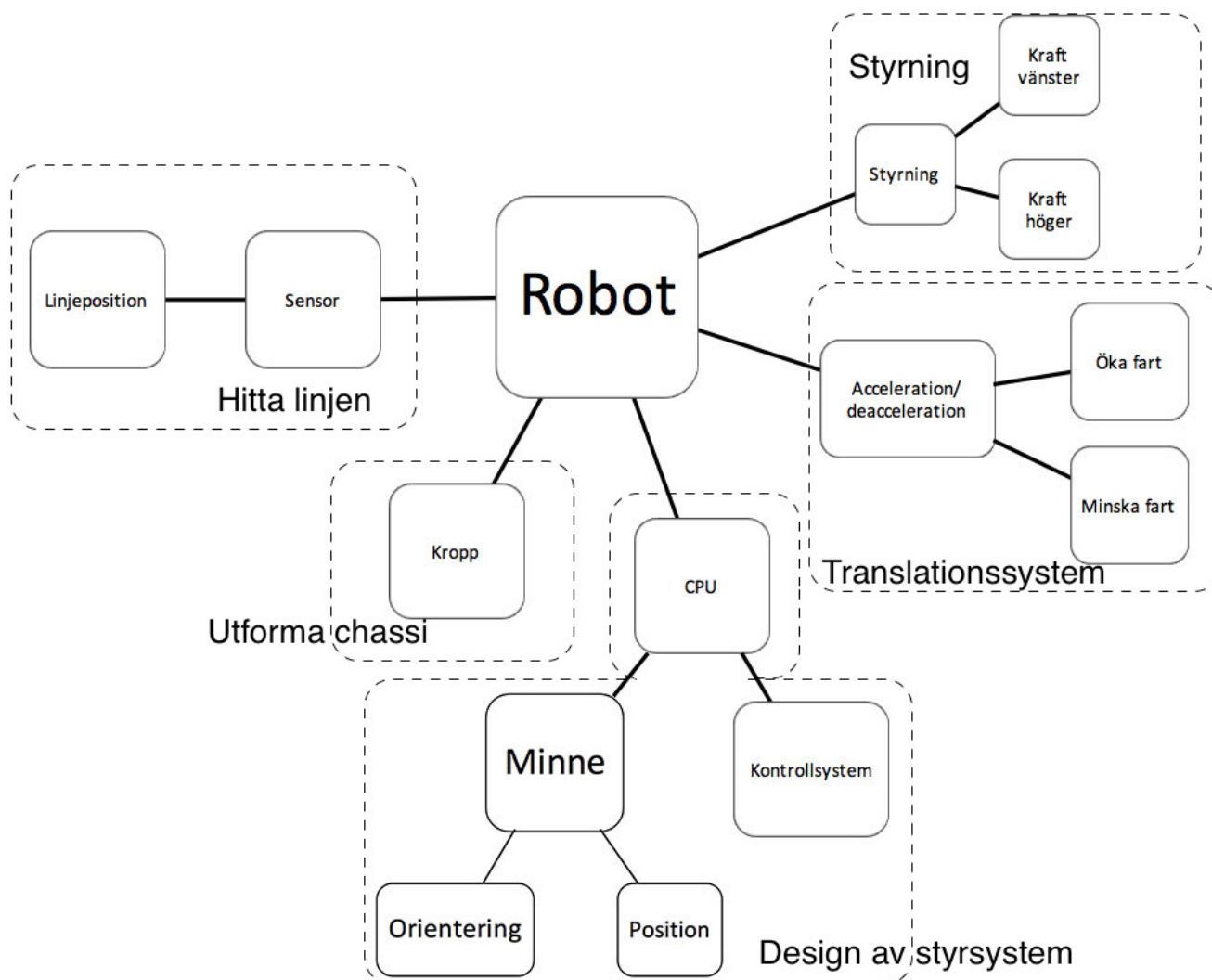
Det här projektet går ut på att bygga en robot som ska ställa upp i robot-SM i linjeföljning. Tävlingen går ut på att en robot ska kunna ta sig runt en bana på en svart linje autonomt på en så kort tid som möjligt. Den robot som klarar banan på kortast tid vinner. Banan består av en sammanhängande svart linje på ett vitt underlag där mål och start är i samma punkt. Tävlingen tillhandahåller olika regler som måste följas för ett godkänt bidrag. Mer om tävlingen går att läsa i appendix B

1.2 Problemformulering

En robot ska designas, tillverkas och programmeras för att kunna identifiera en linje och följa den. Roboten ska därefter välja fart och riktning beroende på robotens läge i förhållande till linjen. Dessutom ska roboten kunna ändra sin hastighet beroende på linjens utformning, allt för att minimera varvtiden.

1.3 Delproblem

För att göra huvudproblemet mer tydligt har detta delats in i mindre delproblem som ska lösas. Dessa illustreras i figur 1.1. Krav som delproblemen ska uppfylla finns att hitta i kravspecifikationen i appendix A.



Figur 1.1: Uppdelning av delproblem

1.3.1 Hitta linjen

Roboten ska kunna identifiera linjen och marken med hjälp av sensorer. Sensordata ska sedan skickas vidare till styrsystemet. Problem som kommer lösas är val av sensorer och sensorplacering.

1.3.2 Utforma chassi

En robot skall skapas och utvecklas med de maximala dimensionerna 300x400mm och väga max 3kg. Detta inklusive batterier, motorer och styrsystem. Chassit ska binda samma all komponenter samt möjliggöra för kablage.

1.3.3 Design av styrsystem

Roboten ska använda sensorinput, nuvarande position och riktning för att bestämma motorkraften som krävs för att minimera avståndet till linjen. I syfte att maximera

snitthastighet under bankörning ska en regulator utvecklas. Robotens reglersystem skall även kunna hantera störningar och felmarginaler.

1.3.4 Styrning

Styrningen ska ta input från regulatorn i syfte att justera robotens riktning, vilket ska ske genom att applicera den kraft som krävs för att minimera avståndet till linjen.

1.3.5 Translationssystem

Ska ta input från regulatorn för att applicera en kraft framåt eller bakåt för att hålla den givna farten.

1.4 Avgränsningar

Då projektet endast har en allokerad tid på 20 veckor samt en given budget på 5000kr kommer en del avgränsningar behövas. Dessa görs för att hålla projektet inom ramarna för vad som är rimligt att förvänta sig av ett arbete av denna omfattning.

- RobotSM arrangerar två olika linjeföljande tävlingar, en kallad Linjeföljning samt en kallad Avancerad linjeföljning. Av dessa två kommer deltagande endast ske i Linjeföljning.

- I den mån möjligt kommer redan färdiga komponenter köpas in. Detta för att inte behöva lägga tid på att tillverka egna komponenter. Dock kommer vitala komponenter som skulle prestera bättre än köpta konstrueras och byggas om tid ges.

- Translationen av roboten kommer skapas med en eller flera elmotorer i kombination med batterier. Andra alternativa källor kommer ej undersökas. Detta med argumentet att elmotorer är billiga, lätta att reglera samt att det skulle minska variationen hos produkten genom att kunna driva dessa komponenter med elektricitet.

- Regler som ställs i RobotSM begränsar robotens dimensioner till en bredd på 300mm samt en längd på 400mm. Dessa regler sätter även en begränsning av att robotens vikt ej får överskrida 3kg.

2

Teoribakgrund till robotdynamik och reglering

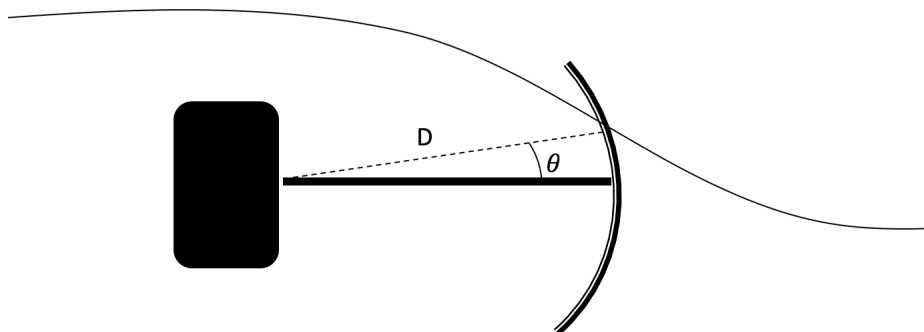
Detta kapitel beskriver bakomliggande teori och matematiska beräkningar som ligger till grund för robotens matematiska modell samt antaganden och beslut som gjorts under designprocessen av roboten.

2.1 Reglering

Look-ahead (LA), eller preview-reglering innebär att regleringen för ett system sker innan systemet nått mätpunkten. För en autonom linjeföljare kan detta utnyttjas på ett antal sätt. I detta projekt har två av dessa undersökts, "Follow the Carrot"(FTC) och "Pure Pursuit"(PP).

2.1.1 Follow the carrot

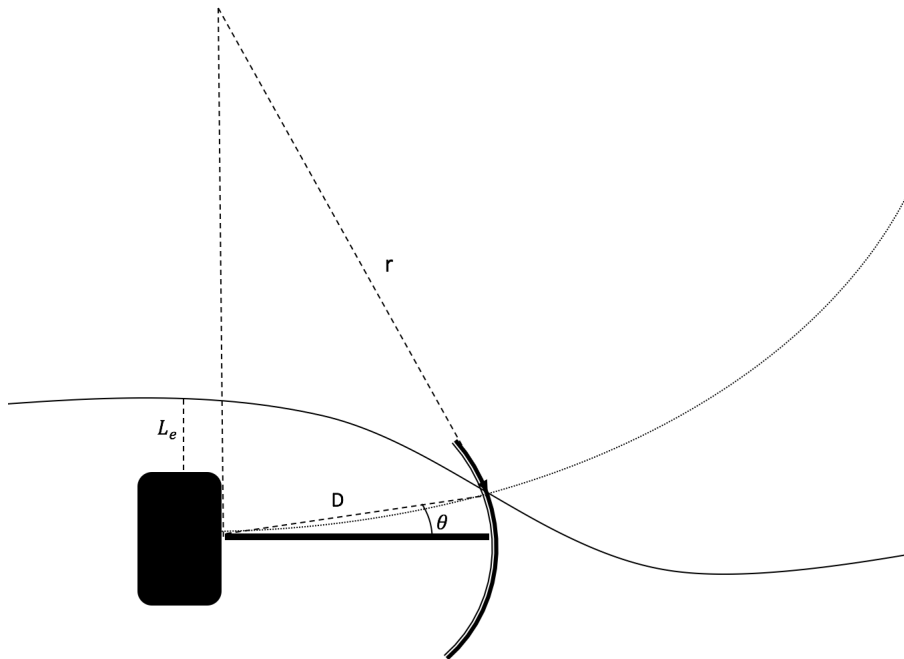
"Follow the Carrot", är ett enkelt sätt att följa en linje med hjälp av Look-ahead (LA) reglering. FTC går ut på att mäta upp en punkt på körbanan och vinkla enhetens körriktning mot punkten. FTC demonstreras i figur 2.1 Signalen till reglersystemet blir då vinkeln mellan enhetens körriktning och mätpunkten på körbanan ($f_{e_{\theta}} = e_{\theta} = \theta$). Detta är en matematiskt enkel lösning men kräver ett snabbt och hårt reglerat system för att hantera överslängar och de oscilleringar som uppstår [2].



Figur 2.1: Demonstration av Follow the Carrot

2.1.2 Pure pursuit

Vid användning av "Pure Pursuit" mäts en punkt på körbanan och utifrån detta skapas en cirkelbåge mellan enhetens rotationscentrum och mätpunkten. Radien på cirkelbågen beskrivs enligt $r = \frac{D}{2\sin\theta}$. Utifrån radien på cirkelbågen skapas en kurvatur som enheten ska följa enligt $\kappa = \frac{1}{r} = \frac{2\sin\theta}{D}$. I figur 2.2 demonstreras PP. På så sätt skapas en mjukare körbana för enheten. I denna metod regleras felet till kurvaturen, inte till linjen, detta ger en mjuk och enkel regulator [2].



Figur 2.2: Demonstration av Pure Pursuit

2.2 Dynamik

Dynamik är läran om kroppars rörelse och därför är den viktig då roboten kommer att vara i rörelse. För att förstå hur olika krafter påverkar roboten under körning finns ett antal samband som är viktiga att beräkna.

2.2.1 Krafter vid acceleration

Acceleration kan beskrivas med hjälp av tidsderivatan av hastigheten

$$a = \frac{dv}{dt} = v \frac{dv}{ds} \quad (2.1)$$

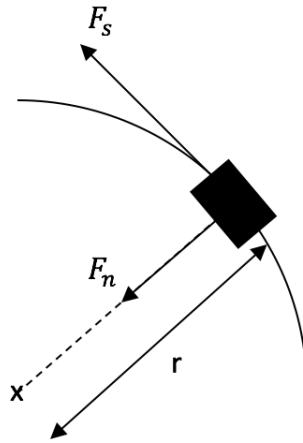
och med hjälp av newtons andra lag fås förhållandet

$$F = ma = m \frac{dv}{dt} = mv \frac{dv}{ds} \quad (2.2)$$

där kraften F är kraften som uppstår på massan vid acceleration.

2.2.2 Krafter vid kurvtagning

Då en partikel/kropp accelererar i en kurva med radien r påverkas denna av kraften \mathbb{F} . Kraften \mathbb{F} kan delas upp i två komponenter, F_s och F_n , där $\mathbb{F} = (F_s, F_n)$. En friläggning av systemet illustreras i figur 2.3.



Figur 2.3: Partikelrörelse i kurva

Kraften F_s är i detta fall kraften som uppkommer då partikeln accelererar eller retarderar. Genom att anta konstant acceleration fås ett uttryck för kraften med avseende på in- och uthastighet ur kurvan.

$$F_s = ma = m v \frac{dv}{ds} \quad (2.3)$$

$$\int_{s_1}^{s_2} F_s ds = \int_{v_1}^{v_2} m v dv \quad (2.4)$$

där $s_2 - s_1 = s$

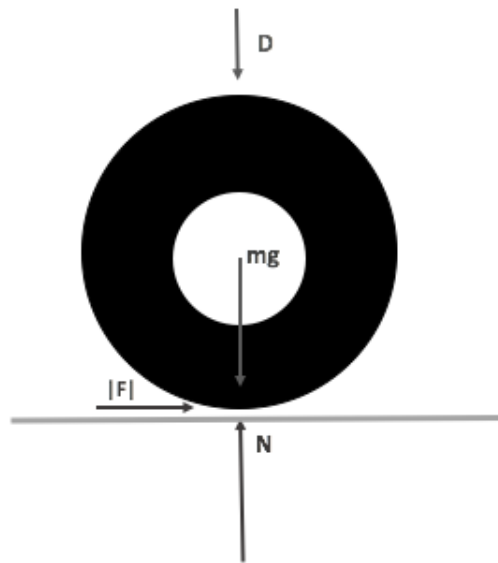
$$F_s s = \frac{v_2^2 - v_1^2}{2} m \quad (2.5)$$

Vidare fås centripetalkraften F_n genom

$$a_n = \frac{v^2}{r} \quad (2.6)$$

där $v = \frac{v_2 + v_1}{2}$

$$\rightarrow F_n = ma_n = m \frac{v^2}{r} \quad (2.7)$$



Figur 2.4: Friktionskraft

Friktionskraften $|\mathbf{F}|$, fås ur följande

$$\mathbf{F} = \mu N \quad (2.8)$$

där μ är friktionskoefficienten vid statiskt tillstånd då däcken kommer att snurra och man vill undvika glid.

2.2.3 Ökning av normalkraft

Genom att minska trycket under roboten och luftflödet underifrån kan normalkraften på roboten öka, detta medför att roboten har mindre risk till glid.

$$\mathbf{F} = \mu N \mu (mg + D) \quad (2.9)$$

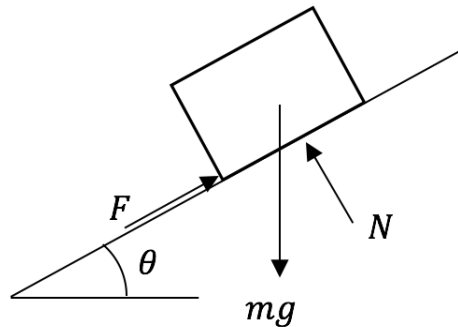
där D är ökningen av normalkraften, se figur 2.4 vilket ger oss vid omskrivning att

$$D = \frac{\mathbf{F}}{\mu} - mg \quad (2.10)$$

Med hjälp av detta kan man beräkna den extra normalkraft som behövs vid ökad kraft vid till exempel acceleration eller kurvtagning. Normalkraften som verkar är konstant då gravitationskoefficienten, g , och massan, m , är konstanta över tiden. μ kan variera och kraften F är både hastighets- och accelerationsberoende.

2.2.4 Beräkning av friktionskoefficienten

För att beräkna μ görs följande friläggning.



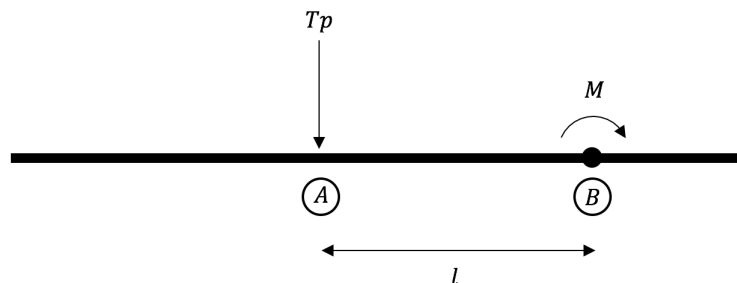
Figur 2.5: Friktionskraft i vinkel

Genom att utgå från ekvation 2.9 fås följande samband. Ur figur 2.5 fås att $F = mg \sin \theta$ och $N = mg \cos \theta$.

$$\mu = \frac{F}{N} = \frac{mg \sin \theta}{mg \cos \theta} = \tan \theta \quad (2.11)$$

2.2.5 Stegring

Om robotens kropp ses som en balk kan friläggnen i figur 2.6 göras. I punkt B angriper moment M som genereras av hjulen. I punkt A ligger robotens tyngdpunkt Tp . Avståndet mellan dessa punkter är l .



Figur 2.6: Roboten sett som en balk

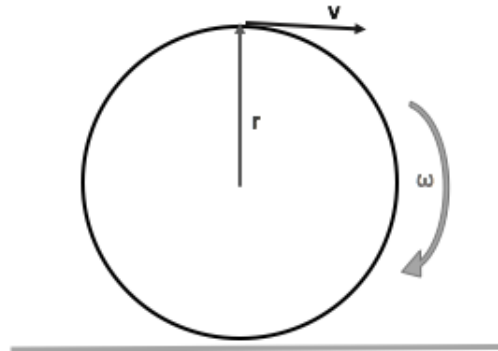
För att undvika stegring krävs att följande samband gäller:

$$Tp \cdot l \geq M \quad (2.12)$$

2.2.6 Differentialstyrning

Differentialstyrning är när två hjul placeras på sidorna av roboten och hjulens hastighet regleras för att styra hastighet och rotation. Robotar med differentialstyrning via två hjul är av typen "unicycle model". Nedan följer viktiga ekvationer som beskriver robotens hastighet V_r och robotens rotationshastighet ω_r beroende av hastigheten i hjulens tangent $V_{hjul} = \omega_{hjul} * r_{hjul}$ där ω_{hjul} är hjulets

vinkelhastighet och r_{hjul} är hjulets radie. $hjul$ kan vara *vänster* eller *höger*. ω_r beror även av avståndet mellan hjulen l .



Figur 2.7: Vinkelhastighet

$$\omega = \frac{v}{r} \quad (2.13)$$

$$V_r = \frac{V_{vänster} + V_{höger}}{2} = \frac{\omega_{vänster} * r_{hjul} + \omega_{höger} * r_{hjul}}{2} \quad (2.14)$$

$$\omega_r = \frac{V_{vänster} - V_{höger}}{l} \quad (2.15)$$

Ur dessa ekvationer kan hjulens vinkelhastighet bestämmas.

$$\omega_{vänster} = \frac{2V_r + \omega_r l}{2r} \quad (2.16)$$

$$\omega_{höger} = \frac{2V_r - \omega_r l}{2r} \quad (2.17)$$

2.2.7 DC-motorer

DC-motorer har kända differentialekvationer som beskriver deras beteende gällande spänning, ström, moment, vinkelhastighet och vinkelacceleration.

$$J\dot{\omega}_m + b\omega_m = K_t i_m \quad (2.18)$$

$$L \frac{di_m}{dt} + Ri_m = u_m - K_e \omega_m \quad (2.19)$$

Där

- ω_m är motorns vinkelhastighet
- $\dot{\omega}_m$ motorns vinkelacceleration
- K_t är motorns momentkonstant
- K_e är motorns bakåtspänningskonstant
- u_m är spänningen över motorn
- i_m är strömmen genom motorn
- L är motorns induktans
- R är motorns resistans
- b är motorns viskösa friktionskonstant
- J är motorns masströgets moment

3

Metod & Genomförande

3.1 Simulering

En omfattande simulering av lösningen har varit av stor vikt i utvecklingen av roboten. Detta tillåter enkel och billig testning av lösningskombinationer och ger möjlighet att ändra parametrar på exempelvis robotens storlek och testa motorer utifrån specifikationer.

3.1.1 Simulinkmodell

En simulinkmodell skapades med modeller av robotens viktiga komponenter inkluderade. Roboten i sig bestod av "unicycle model" med fysiska trögheter och en sensorrigg framför roboten. En uppritad bana importerades till simulink i ett koordinatsystem där en etta mostvarade vit bakgrund och en nolla mosvarade den svarta linjen som roboten skulle följa. Detta möjliggjorde att olika banor enkelt kunde ritas upp för att testa hur roboten skulle bete sig vid olika kurvor.

3.1.1.1 DC-motorer

En modell av en DC-motor ställdes upp med hjälp av ekvationerna 2.18 och 2.19. Motorerna belastades med robotens och hjulens tröghetsmoment och friktioner från framförel, andra förluster som tex friktioner i växellådor försumrades då den är liten i förhållande till effekten som kommer användas, dessutom är all friktion uppskattad till en början. Efter att motorn modellerats togs hänsyn till robotens mål hastighet vilket gav en mer rättvis bild av hur motorerna kommer prestera i verkligheten. Denna modell tillåter begränsningar av spänning och ström samt mätning av fysiska signaler i motorn som moment och acceleration. Motorns spänning och förbrukade ström gav effekten som motorerna avgav och kunde därför användas till att få fram hur starka motorer vi behöver köpa för att uppnå liknande resultat i verkligheten som i simuleringen. Modellen använder motorparametrar som kan tas från verkliga motorer.

3.1.1.2 Sensorrigg

Sensorriggen på roboten modellerades som en cirkelbåge placerad ett avstånd framför roboten, detta avstånd bestämmer systemets look-ahead. Denna plockar data ur pixlarna i den uppritade banan, svart/vitt, och ger ut en vinkel likt beskrivningen av Pure Pursuit. Vinkeln är diskret och har en upplösning beroende

av antalet sensorer som placerade på sensorriggen, högre antal sensorer ger lägre intervall mellan data. Om flera sensorer samtidigt skulle ge utslag snittas deras vinkelbidrag, något som behövs då avståndet mellan sensorerna blir lägre än bredden på linjen. Sensorriggens programmering skapades med avseende att emulera hur programmeringen i den verkliga roboten kommer att fungera.

3.1.1.3 Roboten

Robotens fysiska modell är som kropp: en tunn platta med ett masströghetsmoment. Denna påverkas av krafter från hjulen som sitter på motorerna, med möjlighet till utväxling. Roboten har en hastighet framåt och en vinkelhastighet runt sin mittpunkt som tillsammans låter roboten förflyttas. I modellen används även beräkningar på centripetalkraften tillsammans med momenten och friktionen i hjulen för att bestämma vid vilka situationer robotens friktion mot marken ej räcker till. Simuleringen tog ej hänsyn till glid om roboten skulle tappa grepp eller om däcken skulle spinna mot marken vid acceleration. Istället jämfördes dessa krafter med det maximala greppet som kan fås från robotens egentyngd tillsammans med eventuell downforce från fläkt eller liknande.

3.1.2 Tester av lösningar

När modellerna av den fysiska roboten var färdiga kunde olika parametrar på motorerna, sensorriggen, batterier och däckfriktion testas. Batteriet testas för att de måste klara av att avge den energi som motorerna kräver för att klara åtminstone ett varv av banan men helst flera då flera försök tillåts enligt reglerna. Roboten tillsammans med olika regulatorer testas för att utvärdera prestanda och olika lösningsförslag.

3.1.2.1 Optimeringsvariabler

Vid simulering uppmättes: ström genom motorer, motoreffekt, friktion/glidning, hastighet och avvikelse från bana. Datan ur motorerna användes för att bestämma om en given motor klarar av att leverera effekten som behövs samt har termisk avledningskapacitet att klara körningen. Friktionsmätningen användes för att uppskatta vilken ökning av kraft nedåt som krävdes på roboten för att den inte ska tappa grepp, samt se hur en adaptiv hastighetskontroll påverkar kraven på friktion. Avvikelsen från banan är direkt kopplad till tävlingen som kräver att roboten alltid är över banan, i simulering ansågs roboten vara över banan om antingen robotens centrum är en halv robotbredd från banan eller sensorriggen ligger över banan.

3.1.2.2 Reglering

Till en början sattes en fast hastighet på 2 m/s och perfekt grepp mot marken antogs. Hastigheten valdes sådant att roboten kör snabbare en förra årets vinnare (ca 1 m/s) med stor marginal. Med dessa premisser togs en regulatoruppsättning fram som gav en bra balans mellan genvägar och maximalt avstånd från linje. Sedan lades effektbegränsningar på motor fram och dess påverkan på systemets

prestanda noterades och justeringar på regleringen gjordes. Sedan lades beräkningar för hjulens friktion mot marken till för att uppskatta vid vilka lägen roboten skulle tappa grepp, och regleringen justerades ytterligare för att minimera glid/sladd. Efter detta lades en variabel hastighetshållning till som justerar hastigheten för att hålla krafterna i kurvor inom ett hanterbart intervall. Regleringen på motorerna gjordes även mycket mer avslappnad för att minska stora moment i hjulen och höga toppar i styrsignalerna.

3.2 Konstruktion

Enligt robot-SM finns regler angående dimensioner av roboten som nämnts tidigare. Olika sätt att tillverka komponenter av roboten diskuterades, bland annat om komponenterna skulle köpas färdiga eller konstrueras själva. Eftersom 3D-printers finns tillgängliga på Chalmers tekniska högskola valde gruppen att 3D-printa de delar som var möjliga och köpa de elektroniska delarna så som motorer och eventuell fläkt.

3.2.1 Motivering av inköp

De komponenter som inte skrevs ut i plast, tillhandahölls av Chalmers eller köptes in. Vid val av de olika komponenterna togs hänsyn till hurvida de klarade av våra krav och önskemål men framförallt grundades valen av komponenter på hur de presterade i simuleringen. (Se kravspecifikationen i appendix A) Budgeten gav stor frihet i val av komponenter som klarar de satta kraven. Utöver de satta kraven krävdes att mått och omfattande specifikationer fanns givna innan komponenten inhandlades. Utifrån de givna specifikationerna testades komponenterna i simuleringen för att verifiera att komponenterna klarar de satta kraven.

3.2.2 Tillverkning av komponenter

Vid konstruktion och tillverkning av roboten diskuterades olika former och storlekar med hänsyn till de komponenterna som hade oföränderliga mått, alltså de redan tillhandahållna eller köpta komponenterna. *Catia V5* användes för att skapa 3D-modeller och ritningar. För 3D utskrifter i plast användes en *MakerBot Replicator 2* skrivare. Skrivaren har inte möjlighet att skriva ut större komponenter än 250x115mm vilket blev en restriktion i hur stora komponenter som kunde designas.

3.2.3 Sensorrigg

Det beslutades att sensorriggen skulle tillverkas av gruppen. Dels för att kunskapen fanns och dels för att det skulle vara smidigare om något behövdes ändras eller repareras. Kretskorts-layout designades i *KiCad* med vinklar och mått från simuleringen. Efter etsning och kontroll av funktionalitet så monterades komponenterna på kortet.

3.2.4 Montering av komponenter

Då alla komponenter färdigställts monterades dessa i E-sektionens Teletekniska Avdelning (ETA) lokaler. ETA tillhandahöll komponenter som skruvförband, kablage och krympslangar. Förutom detta fanns det även de nödvändiga verktygen att tillgå.

3.3 Programmering

Programmeringen av robotens funktionalitet är uppdelad i tre enheter; en Arduino som styr Look-Aheadregleringen och bestämmer hastigheten för varje motor; Motorkontrollerna som reglerar i motorerna och ser till att de håller de hastigheter Arduino skickar; en blåtandapplikation som startar och stoppar roboten. I applikationen sätts även reglerparametrar och robothastighet vilket underlättar testning av robotens beteende vid olika reglerparametrar.

3.3.1 Arduino

För att programmera Arduinon användes Arduino IDE som tillåter programmering i en förenklad version av programmeringsspråket C. Koden består av en "setup" funktion som körs när Arduinon först kopplas in eller när reset-knappen aktiveras, och en "loop" funktion som upprepas så länge Arduinon är igång. I Arduinokoden skall roboten läsa av sensorriggen, undersöka vilken vinkelhastighet som skall sättas och sedan skriva motsvarande de hastigheter på motorerna. Algoritmen för bestämning av vinkelhastigheten baseras på algoritmen i simuleringen för att minimera skillnaden mellan simulering och verklig robot.

3.3.2 Motorkontroller

Motorkontrollerna har inbyggda PID-regulatorer med syfte att reglera bort olinjäriteter i motor och hjul. Bestämning av dess parametrar skedde genom bantester med värdet på simuleringens parametrar som utgångsvärden. Motorkontrollerna har en egen PID-period vilket ger att PID-parameterarna på den verkliga roboten skiljer sig från simuleringens parametrar.

3.3.3 Blåtandskommunikation

Ett enkelt program skrevs i Python som tillåter kommunikation med en blåtandsmottagare som kan kommunicera seriellt med Arduinon på roboten. Enkla kommandon kan skickas till Arduinon som startar, stoppar och modifierar parametrar i styrningen under körning så att Arduinon ej behöver programmeras om vid varje test.

3.4 Utvärdering och tester

För att utvärdera robotens prestanda konstruerades ett flertal tester. Detta gjorde att det gick att jämföra simuleringsdata med verklig data men också så att bankörningen kunde optimeras.

3.4.1 Testbana

Till tester av robotens bankörning designades en kort testbana med svart tejp på vitt underlag. Här kördes roboten iterativt för att hitta optimala inställningar på reglering, hastighet och fläkthastighet.

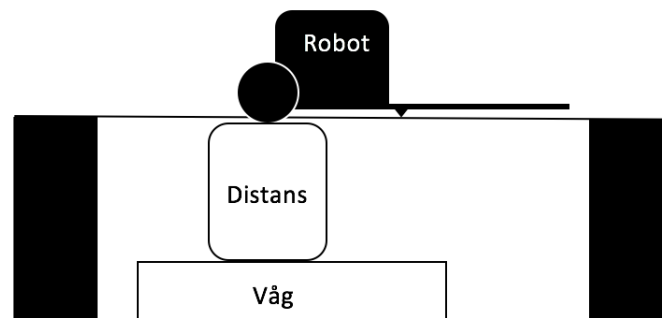
3.4.2 Mätning av friktion

För att mäta friktionen ställdes roboten på en horisontellt placerad skiva med det önskade underlaget. Genom att sedan öka lutningen på skiva nåddes ett läge där roboten tappade grepp och började glida. Här kontrollerades vinkeln och med denna kunde sedan en friktionskoefficient beräknas med ekvation 2.11.

3.4.3 Mätning av ökad kraft på bakdäck med olika fläkthastigheter

För att mäta hur mycket normalkraften på däcken ökar på grund av fläkten gjordes en enkel testrigg. Testriggen bestod av en våg. Över vågen hängdes en skiva för att ge roboten något att skapa undertryck mot. På vågen placerades distanser för att ge däcken kontakt med vågen och skapa rätt markfrigång. Figur 3.1 illustrerar testriggen.

Testet genomfördes genom att successivt öka fläktens hastighet och se hur normalkraften på bakdäcken förändras. Detta dokumenterades sedan och datan sammanställdes i en graf.



Figur 3.1: Testrigg för mätning av ökad normalkraft på däck

3.4.4 Jämförelse av simulinkmodell och verklighet

När testning och optimering av roboten var gjord så gjordes jämförelse med simuleringsresultaten. Detta gjordes genom att banan som simuleringen testats på skapades i verkligheten och sedan kördes med samma parametrar. Den verkliga bankörningen filmades och robotens centrum plottades upp mot banan. För att underlätta plottningen valdes fläktens centrum som centrum för hela roboten.

4

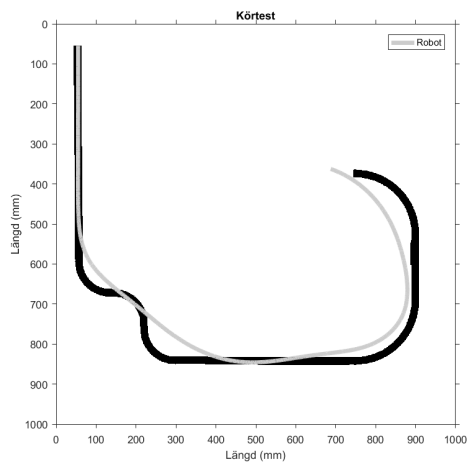
Resultat

Utifrån de metoder som var givna i kapitel 3 togs följande resultat fram i design av roboten.

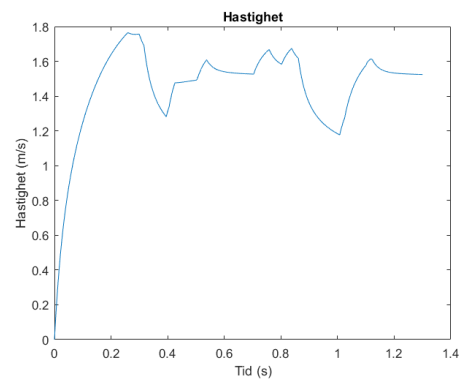
4.1 Simulering

Figur 4.1 (a) visar en exempelsimulering vid körning runt en bana. Simuleringen modellerar endast hur roboten kör med perfekt grepp men visar om grepp tappas. Denna sorts simuleringsresultat användes för att avgöra hurvida roboten kan uppfylla kravet att hålla sig över banan. Figur 4.1 (a) visar på ett önskvärt beteende, dvs roboten följer inte linjen perfekt utan tar genvägar utan avvikningar större än 280mm från linjen. Alla simuleringar i kapitlet gjordes med samma målhastighet 2 m/s och under samma tid 1.3 s. I figuren kan även hastighet, RPM, grepp, effekt och styrsignal på den högra motorn utläsas då det är den motorn som belastas hårdast under körning vid vänsterkurvor. Figur 4.1 (c) visar rotationshastighet i motorn under körning med utväxling 9.87:1. Detta ger att den verkliga robotens motorer måste kunna leverera ca 500 RPM till hjulen. Figur 4.1 (d) visar robotens grepp under körning, den horisontella linjen är vid vilken kraft roboten tappar grepp från banan. Den simulerade kraften visar att roboten kommer tappa grepp under hårda svängar och acceleration om inte mer downforce kan appliceras med hjälp av en fläkt eller förbättra friktionskoefficienten mellan däck och bana. Figur 4.1(e) visar effekten som krävs för att accelerera motorerna. Eftersom inga förluster är modellerade antas att högre effektkrav i den verkliga roboten.

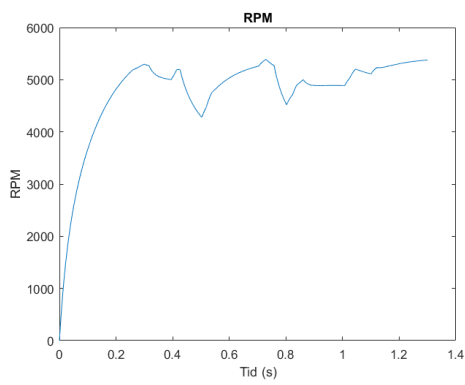
4. Resultat



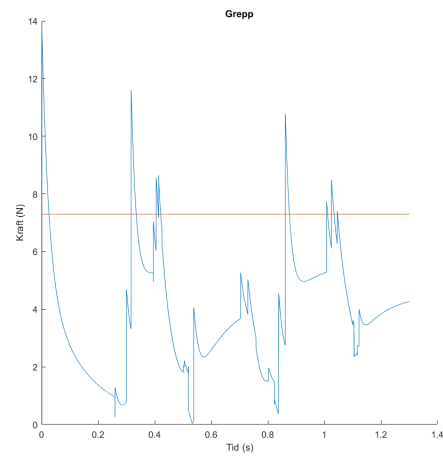
(a) Testkörning



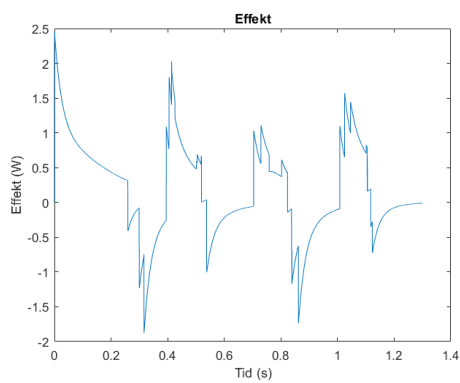
(b) Hastighet relativt banan



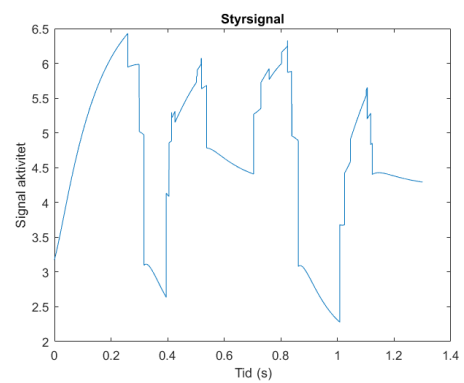
(c) RPM



(d) Grepp



(e) Effekt



(f) Styrsignal till motorerna

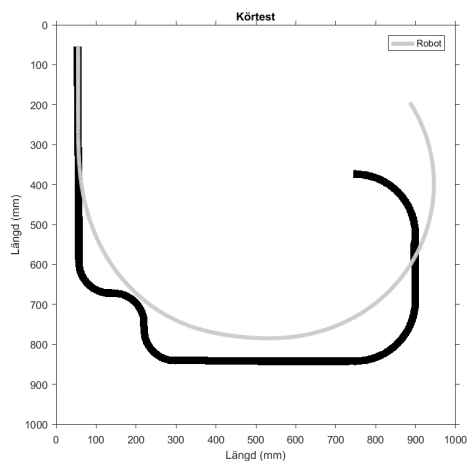
Figur 4.1: Testkörning

4.1.1 Val av algoritm

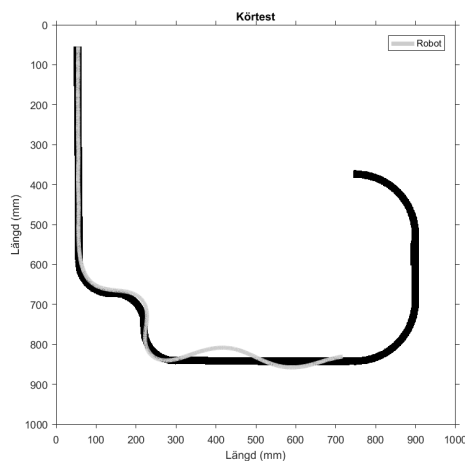
Look-ahead (LA) reglering med Pure Pursuit algoritm valdes med antagandet att oscilleringarna som uppstår av Follow the Carrot kräver ett hårdare reglerat system, vilket leder till större toppar i styrsignalen vilket sätter större krav på robotens motoreffekt och kräver bättre grepp. Cirkelbågen som skapas vid Pure Pursuit kan även regleras med en P-regulator för att kompensera överslängar som uppstår på grund av transportfördröjningar i systemet.

4.1.2 Look-aheadavstånd

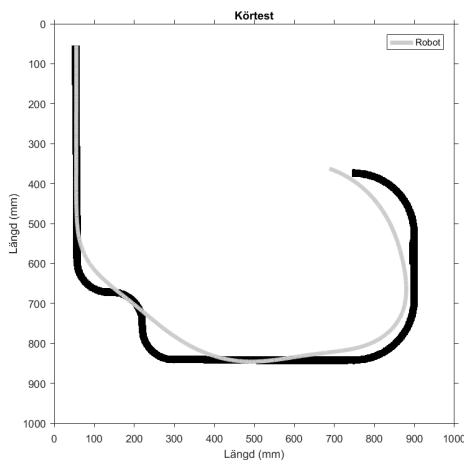
Av alternativen i figur 4.2 ansågs alt (c), med 20 cm LA vara det bästa valet då roboten kan ta genvägar utan att kompromissa med kurvskarphet och håller sig nära linjen utan att oscillera.



(a) 40cm Look-ahead



(b) 10cm Look-ahead



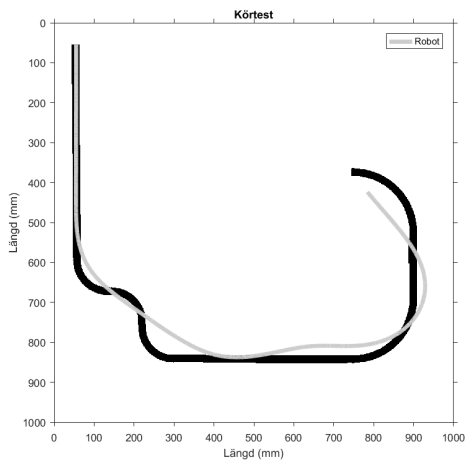
(c) 20cm Look-ahead

Figur 4.2: Körning med varierande Look-ahead

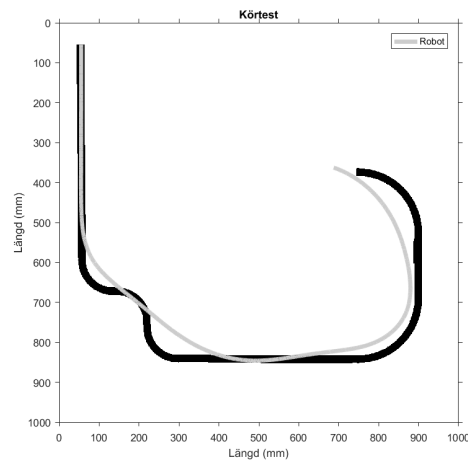
4.1.3 Antal sensorer

I figur 4.3 visas ett antal simuleringar med olika antal sensorer på sensorriggen. Figuren visar tydligt att prestandan planar ut med ett ökande sensorantal. Med ett sensorantal större än 8 ökar prestandan marginellt. Upplösningen θ_{res} , vinkeln mellan två sensorer i arrayen, kan enkelt beskrivas beroende av LA-avståndet d_{LA} , bredden på sensorarrayen b_{sens} och antalet sensorer N :

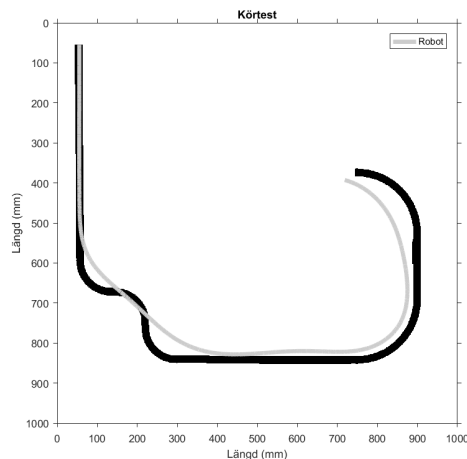
$$\theta_{res} = \frac{\arctan\left(\frac{b_{sens}}{2d_{LA}}\right)}{N} \quad (4.1)$$



(a) 4 sensorer



(b) 8 sensorer



(c) 12 sensorer

Figur 4.3: Körning med varierande antal sensorer

4.1.4 Reglering

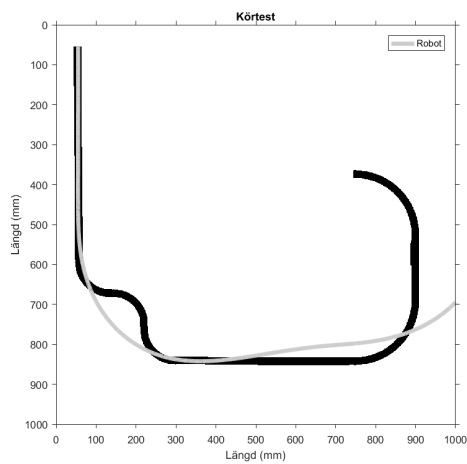
Reglersystemet består av en kaskadreglering. Ytterst sitter en P-regulator som styr cirkelbågen roboten skall följa. I motorkontrollerna sitter PID-regulatorer som utgör

den inre återkopplingen.

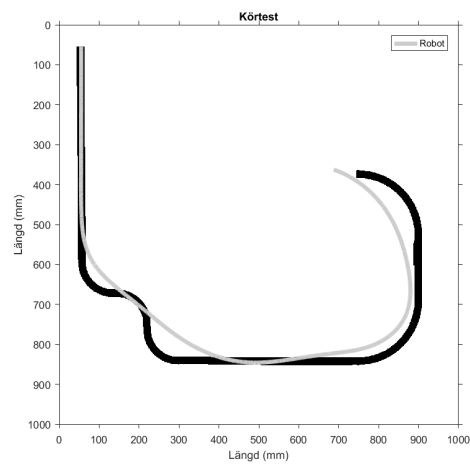
4.1.4.1 P-regulator

I simuleringen undersöktes tre olika K_p som värde på P-regulatorn. Figur 4.4 visar att ett för högt K_p skapar oscilleringar och roboten kör en längre väg utan att förbättra avståndet till linjen. För lågt K_p ger att systemet inte hinner svänga snabbt nog.

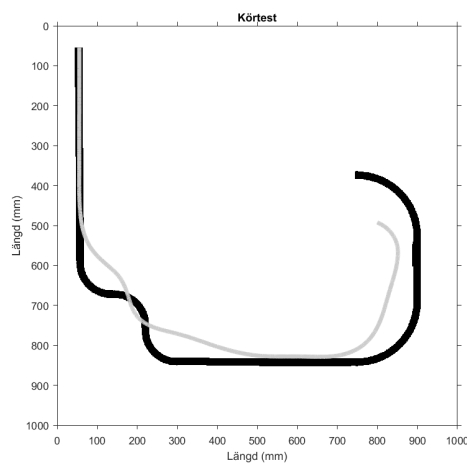
Den svarta linjen i plotten motsvarar linjen roboten önskar att följa och den grå är robotens faktiska väg.



(a) $K_p = 2$



(b) $K_p = 5$

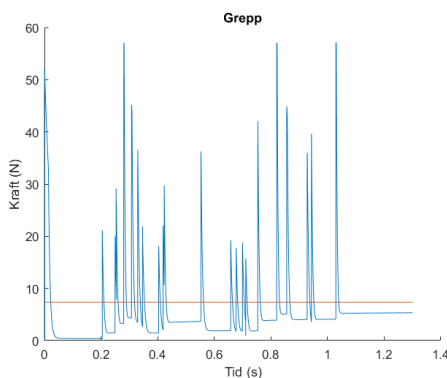


(c) $K_p = 15$

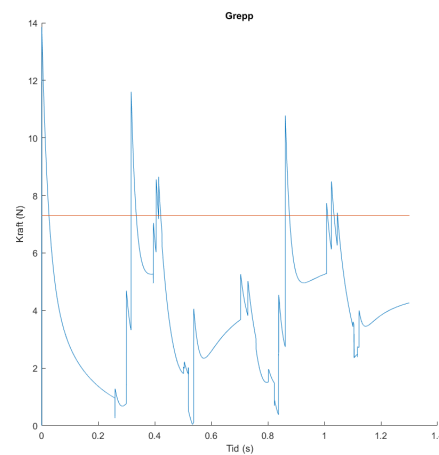
Figur 4.4: Körning med varierande K_p

4.1.4.2 Motorkontroller

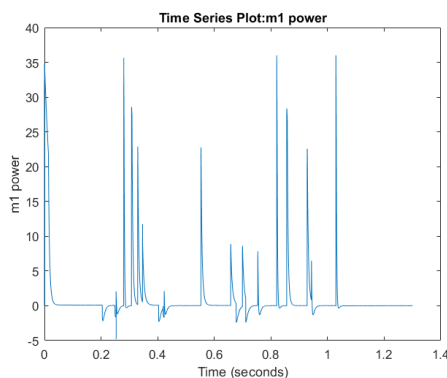
PID-regulatorerna i motorkontrollerna påverkar robotens körbanor minimalt men ger större toppar på styrsignalen vilket kan ses i figur 4.6. Detta sätter högre krav på grepp och effekt vilket kan ses i figur 4.5. Den horisontella linjen på greppkurvorna är den kraft som krävs för att roboten skall tappa grepp.



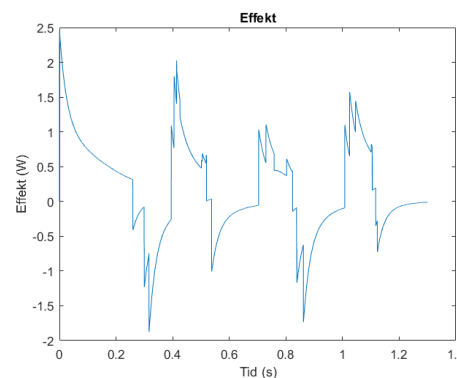
(a) Grepp med hård reglering ($P=0.5$, $I=1$, $D=0$)



(b) Grepp med mjuk reglering ($P=0.05$, $I=0.8$, $D=0$)

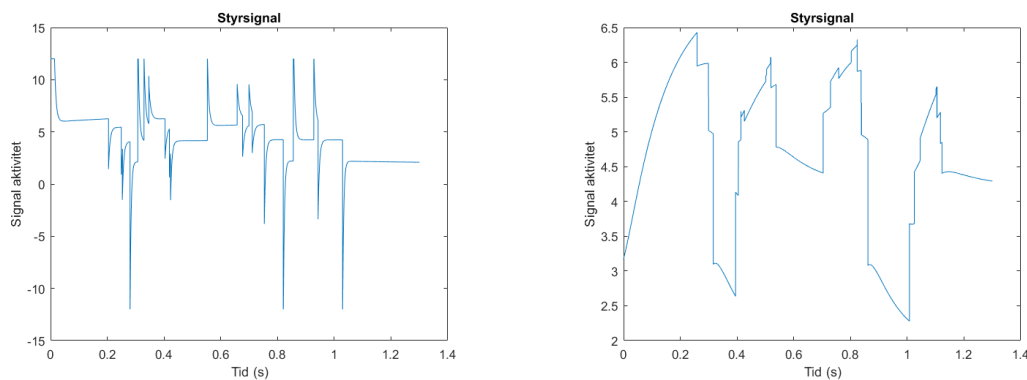


(c) Effekt med hård reglering ($P=0.5$, $I=1$, $D=0$)



(d) Effekt med mjuk reglering ($P=0.05$, $I=0.8$, $D=0$)

Figur 4.5: Grepp och effektkurvor med varierande PID-parametrar



(a) Styrsignal med hård reglering
($P=0.5$, $I=1$, $D=0$)

(b) Styrsignal med mjuk reglering
($P=0.05$, $I=0.8$, $D=0$)

Figur 4.6: Signalaktivitet med varierande PID-parametrar

4.2 Konstruktion

Vid val av komponenterna som både konstruerades och köptes togs hänsyn till simuleringsresultatet. Detta medförde följande resultat på val av komponenter.

4.2.1 Färdiga komponenter

De komponenter som inhandlades var följande:

Tabell 4.1: Tabell över köpta komponenter till konstruktionen

<i>Komponent</i>	<i>Fullständigt namn</i>
Motorer	Pololu 9.7:1 Metal Gearmotor 25Dx48L, 48 CPR encoder
Motorhållare	Pololu 25D Metal Gearmotor Bracket Pair
Hjulnav	Pololu Universal Aluminum 4mm Mounting Hub
Motorkontroll	Jrk 21v3 USB Motor Controller with Feedback
Batteri	GensAce Tattu 1300 mAh 4S 45-90 C LiPo
IR-sensorer	Fairchild semiconductor QRE1113GR
Fläkt	EDF 64 mm ink.l motor Tiger AT2010 5000 kv
Rullkula	Polulo metal Ball Caster 9.5mm
Däck	HPI Racing x-pattern radial tier 26mm D compound
Mikrokontroller	Arduino Uno Rev. 3
Bluetooth-modul	HC-06, Bluetooth-modul

4.2.1.1 Motorer

Motorerna som valdes var två *Pololu 9.7:1 Metal Gearmotor 25Dx48L med en 48 CPR Encoder*. Motorn har en maximal rotationshastighet på 10000 rpm. Denna motor har en inbyggd planetväxellåda med en utväxling på 9.7 vilket ger en

tomgångshastighet på 1030 rpm och ett vridmoment på 0,31 Nm.

4.2.1.2 Motorkontroll

Två motorkontroller krävdes, en till varje motor. De som valdes var *Jrk 21v3 USB Motor Controller with Feedback* som har inbyggd programmerbar PID-regulator.

4.2.1.3 Batteri

Batteriet som användes var ett *GensAce Tattu 1300 mAh 4S 45-90 C LiPo*. Batteriet har en kapacitet på 1300 mAh, en vikt på 163 g. Vid leverans erhöles fel batteri och gruppen hade egentligen önskat ett *GensAce Tattu 1300 mAh 3S 45-90 C LiPo* som är ett 3-cellsbatteri.

4.2.1.4 Fläkt

Fläkten köptes för att kunna öka normalkraften på roboten och den som valdes var en kanaliserad fläkt med 64mm blad. S.k. *EDF 64 mm inkl motor Tiger AT2010 5000 kv*.

4.2.1.5 Däck

För att få så hög friktionskoefficient som möjligt togs beslutet att köpa gummidäck. Däcken som införskaffades var ett par *HPI Racing x-pattern radial tier 26mm D compound for 1/10th on-road cars*.

4.2.1.6 Mikrokontroller

Som mikrokontroller valdes en Arduino UNO Rev 3. Denna kommer bearbeta sensordata och skicka ut signaler till motorkontrollerna.

4.2.1.7 IR-sensorer

IR-sensorerna mäter reflektiviteten av underlaget och kan avgöra hurvida underlaget under en sensor är mörkt eller ljust.

4.2.1.8 Monteringskomponenter

Till motorn köptes motorfästen, *Pololu 25D Metal Gearmotor Bracket Pair*, vilka användes till att fästa motorena i bottenplattan. Två nav, *Polulu universal Aluminium Hub 4mm, M3*, köptes för att kunna fästa fälgen på motoraxeln. Rullkulan som inhandlades var en *Polulu metal Ball Caster 9.5mm*. Denna

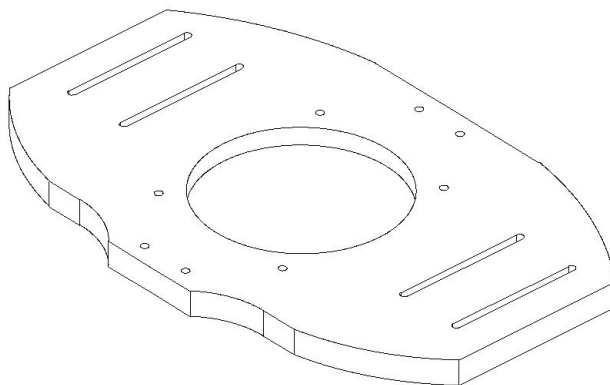
användes för att få en tredje kontaktpunkt till marken.

4.2.2 Tillverkning

De komponenter som kom att tillverkas var en fläkthållare, bottenplatta, arduino-hållare, batteri-hållare, sensorarm, sensorrigg och fälgar. En komplett ritning för hur montering sker och dimensioner för de egen tillverkade komponenter återfinns i appendix C.

4.2.2.1 Bottenplatta

Bottenplattan fick ett mått på 220x138mm. Bottenplattans undersida konstruerades med en urfasning vilket skapade en kjol med höjden 3mm. Dessutom gjordes ett hål i bottenplattans mitt för att kunna skapa ett luftflöde till fläkten. Runt detta hål placerades fyra hål för att kunna fästa fläkthållaren. På bottenplattans sidor togs beslutet att använda två skåror vari motorfästen monteras. Detta skapar en frihet i att tyngdpunkten i förhållande till däckan kan förflyttas bak enligt ekvation 2.12. Fram på bottenplattans mitt gjordes två hål avsedda att fästa sensorarmen. Baktill på bottenplattan finns ytterligare två hål för att fästa mikrokontrollern.

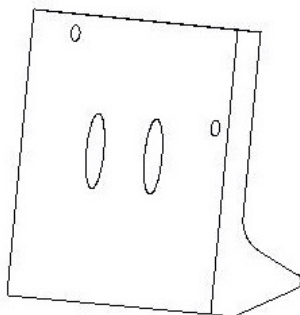


Figur 4.7: Bottenplatta

4.2.2.2 Mikrokontrollerfäste

Ett mikrokontrollerfäste tillverkades med bredden 70mm och höjden 54mm för att kunna fästa en Arduino Uno på roboten. Med hjälp av två hål kunde mikrokontrollern skruvas fast i fästet och med ytterligare två skruvar fästs

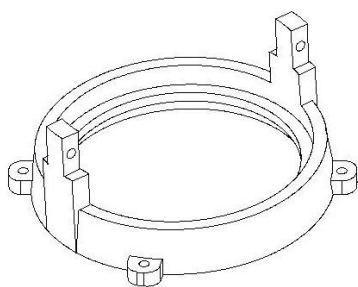
mikrokontrollerfästet på bottenplattan.



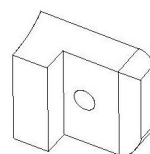
Figur 4.8: Mikrokontrollerfäste

4.2.2.3 Fläkthållare

Hållaren till fläkten är anpassad efter de mått från den inköpta fläkten. Fläkthållaren är konstruerad med en upphöjd kant för att hålla fast fläkten över det urfasade hålet i bottenplattan. För att förhindra rotation av fläkten har fläkthållaren ett stag där ytterligare två fästen kan klämma fast fläkten. Se figur 4.9 och 4.10. Fläkthållaren konstruerades med fyra hål för att kunna monteras på bottenplattan.



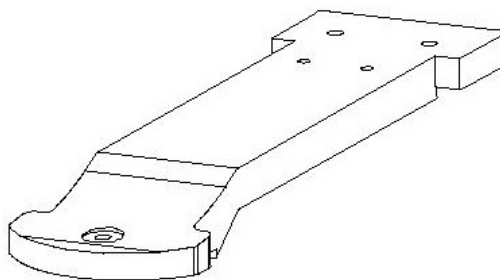
Figur 4.9: Kon för fläkt



Figur 4.10: Klämfäste för kon

4.2.2.4 Sensorarm

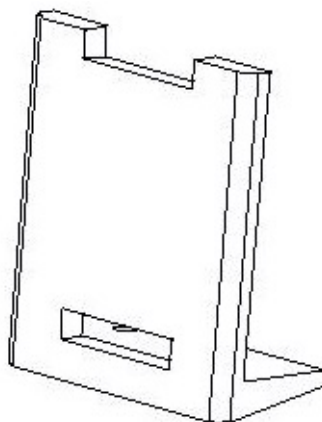
Sensorarmen fick en längd på 100 mm för att ge det önskade pure-pursuit avståndet. Infästningarna på sensorarmen anpassades efter bottenplattans och sensorriggens mått. Dessutom gjorde en nedböjning av sensorarmen så att sensorriggen hamnade närmare golvet för att sensorerna lättare ska se linjen.



Figur 4.11: Sensorarm

4.2.2.5 Batteri-hållare

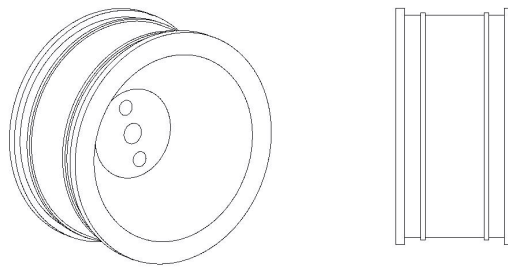
Batterihållaren konstruerades så att den kunde monteras ovanpå sensorriggen och bottenplattan. Den påverkades av batteriets storlek och hamnade på en längd av 47mm och en bredd på 40mm. För att fästa batteriet på hållaren användes karborreband och i hållaren gjorde två spår av måtten 20x4mm på under- samt ovansidan.



Figur 4.12: Batterihållare

4.2.2.6 Fälgar

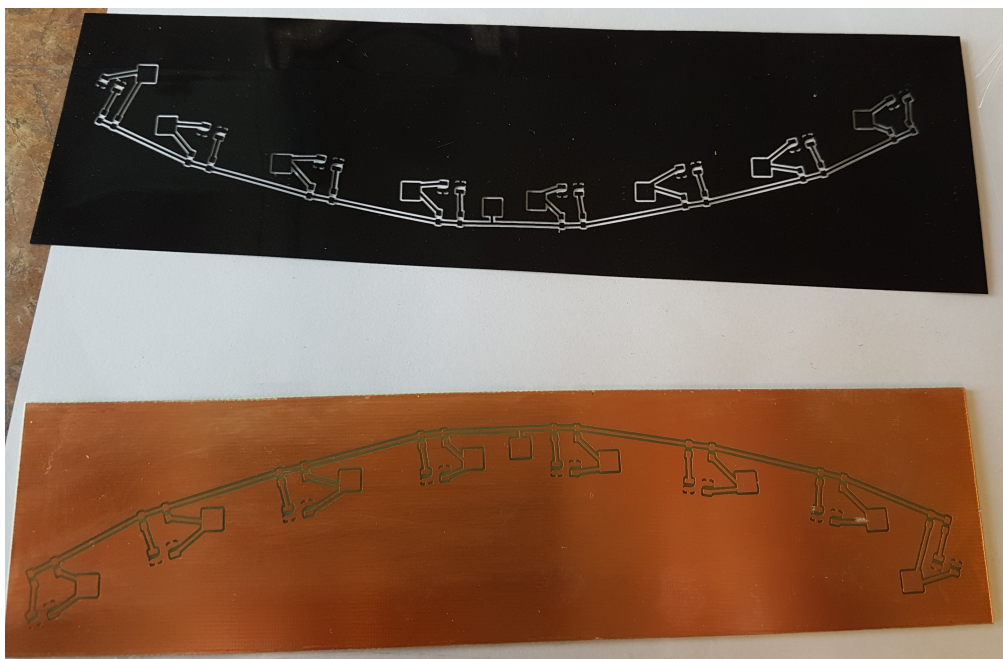
Fälgarnas dimensioner anpassades för att passa de däck som inhandlades. Däcken fästes sedan i motorfästen som sedan skruvas direkt på motoraxeln. Fälgarna fick en radie på 31mm.



Figur 4.13: Fölgar

4.2.2.7 Sensorrigg

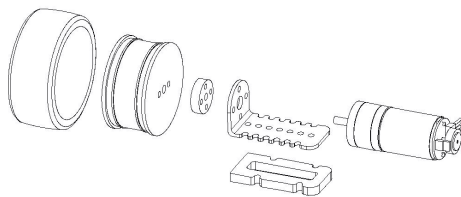
Sensorriggen tillverkades så att sensorerna sitter i en cirkelbåge 200mm från robotens rotationscentrum. Sensorerna är placerade så att de sprider ut sig 45 grader sett från rotationscentrum. Enligt simuleringen nås bäst resultat vid dessa mått. Figur 4.14 nedan visar layouten från KiCad samt det etsade färdiga kretskortet.



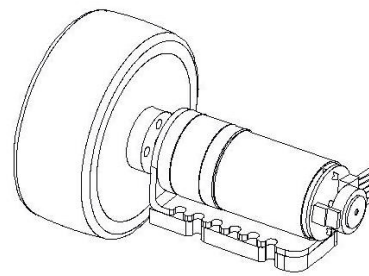
Figur 4.14: KiCaddesign och färdigt kretskort

4.2.3 Montering av komponenter

Då alla robotens komponenter var inköpta och tillverkade började monteringen. Först trädde däcken på fälgarna. På fälgarna fästes hjulnaven. Efter detta skruvades motorerna fast i motorfästena. Därefter fästes hjulnaven på motoraxeln. (Se figur 4.15).

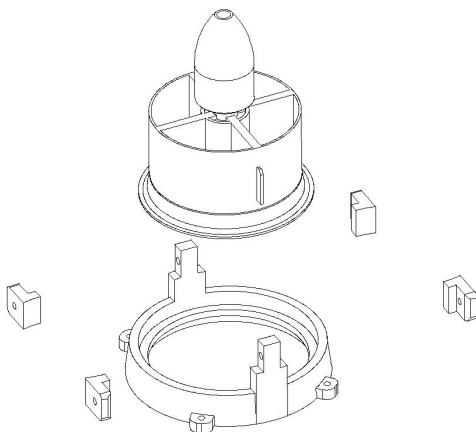


Figur 4.15: Sprängskiss av hjulupphängning

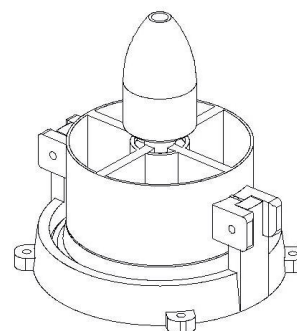


Figur 4.16: Monterad hjulupphängning

Fläkten placerades i fläkthållaren och fästes sedan med klämfästena för att hållas på plats. Fläkten kommer vid körning pressas ner mot konen. För att säkerställa att konen inte roterar kring sin axel eller rör sig i vertikalled kläms den även fast med klämfästen. (Se figur 4.17).



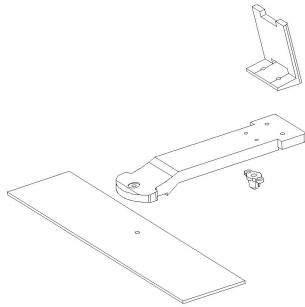
Figur 4.17: Sprängskiss av fläkt med hållare



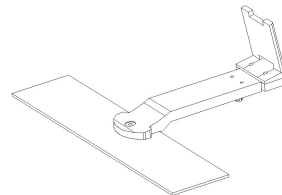
Figur 4.18: Monterad fläkt med hållare

På sensorarmen skruvades kretskortet och styrkulan fast. Styrkulan kommer fungera som en tredje stödpunkt, där däcken är de andra två. Sensorarmen är

utformad för att skapa rätt avstånd mellan IR-sensorerna och marken. För att även minska användningen av skruvar monteras batterihållaren i samma hål som sensorarmen och bottenplattan. (Se figur 4.19)

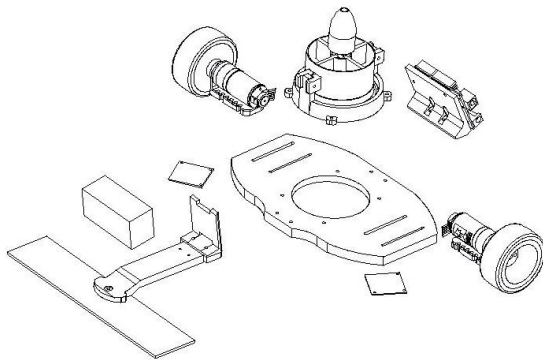


Figur 4.19: Sprängskiss av monterad sensorarm

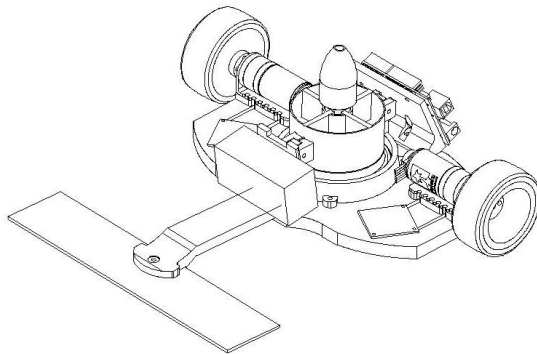


Figur 4.20: Monterad sensorarm

På bottenplattan fästs sedan mikrokontrollerfästet baktill på plattan och framtill fästs sensorarmen. Hjulupphängning fästs i bottenplattans skåror på vardera sida. Motorkontrollerna till DC-motorerna skruvas även de fast i dessa skåror. I mitten av bottenplattans skruvas fläkthållaren fast. (Se figur 4.21) De slutgiltiga dimensionerna på roboten blev 293x283mm och en vikt på 918gram.



Figur 4.21: Sprängskiss av robot



Figur 4.22: Monterad robot

4.3 Arduinoprogram

En förenklade version av arduino koden för roboten beskrivs nedan i C kod. Koden beskriver algoritmen som använts men är ej hela använda koden.

```
|| void setup() {
```

```

lCPin = 3;
rCPin = 2;
pinMode(lCPin, INPUT_PULLUP);
attachInterrupt(0, lTick, RISING);
pinMode(rCPin, INPUT_PULLUP);
attachInterrupt(1, rTick, RISING);

//global variables initialization and motor+fan setup

time = millis(); //initialize timers

countTime = millis();
}

void loop() {
  if(run){
    int sensArray[noSens] = {analogRead(A2), 0, analogRead(A1),
      analogRead(A0), analogRead(A3), analogRead(A4), 0,
      analogRead(A5)};
    angleDes = getDesiredAngle(sensArray); //calculate seen angle
      with sensor data
    double omega = cOmega(cKp(), robVel); //calculate Kp and use it
      to calculate omega
    runMotors(MotorSpeed(targetSpeed, omega), MotorSpeed(
      targetSpeed, -omega)); //run the motors given a target speed
      and a target omega according to the equations for wheel
      speed given in theory chapter
    if(millis() - time > runTime){
      run = false;
    }
  }else{
    //stop the robot
  }

  if(Serial.available()){
    btVal=Serial.read();
    switch(btVal){
      //manage commands read from bluetooth and set parameters
      accordingly
    }
  }

  //update the current speed of the robot 50 times per sec
  if(millis() - countTime > 20){
    robVel = (((lCount + rCount)/2)/CPR)*2*PI*r*50;
    lCount = 0; //reset the counts
    rCount = 0;
    countTime += 20; //update time
  }
}

//the following functions are run on rising edges on pins set in
  setup
//keeps track of the velocity of the motors
void lTick(){
  lCount++;

```

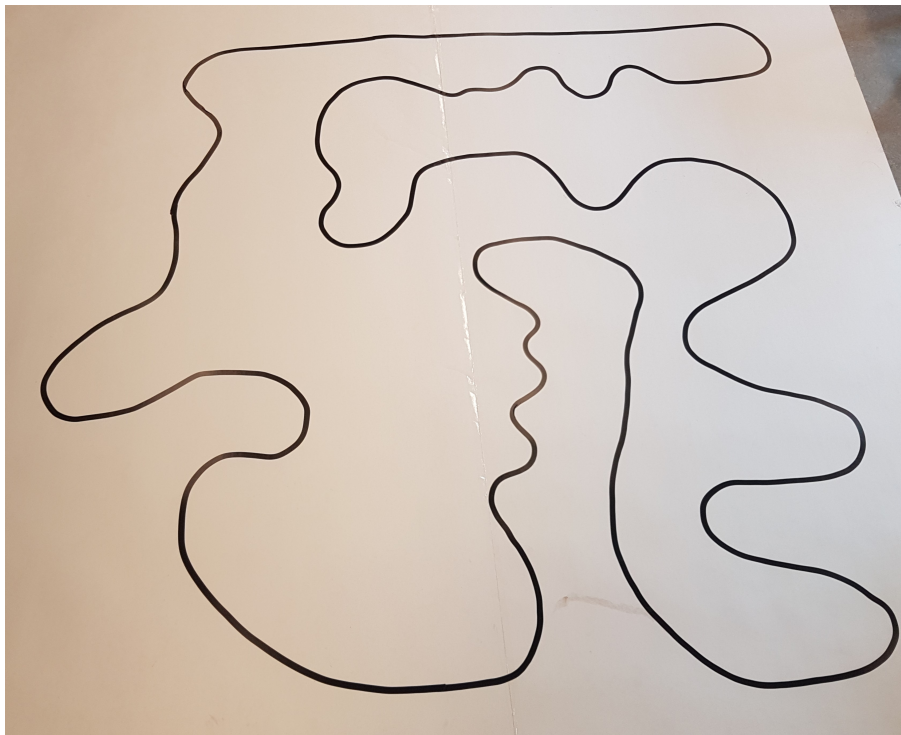
```
}  
void rTick(){  
    rCount++;  
}  
}
```

4.4 Utvärdering och tester

Ett antal tester utformades enligt metoderna beskrivna i kapitel 3. Följande är testresultaten för dessa tester.

4.4.1 Körning på Testbana

Figur 4.23 visar den bana som roboten testats på. Banan testar alla ytterligheterna som banreglerna tillåter i robotSM. Roboten klarar denna ban med en målhastighet på 1.2 m/s. Utöver 1.2 m/s krävs justering av PID-parametrar specifikt för hur banan ser ut.



Figur 4.23: Testbana

4.4.2 Mätning av friktion

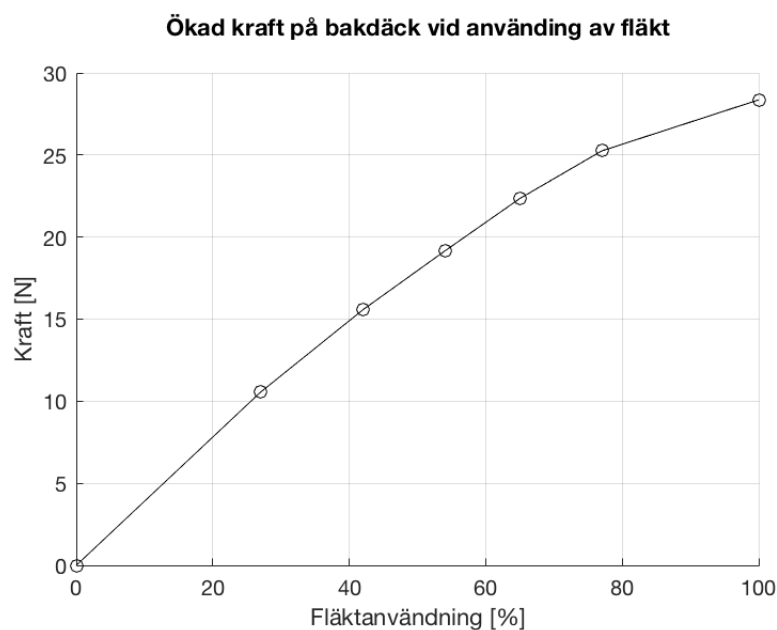
Följande friktioner mättes upp vid friktionstestet. I de två testen som gjordes observerades att roboten hade bättre grepp på en träskiva än vad den hade på ett pappmaterial.

Tabell 4.2: Tabell över uppmätta friktionskoefficienter

Underlag	Friktionskoefficienten, μ
<i>Papper</i>	0.70
<i>Trä</i>	0.83

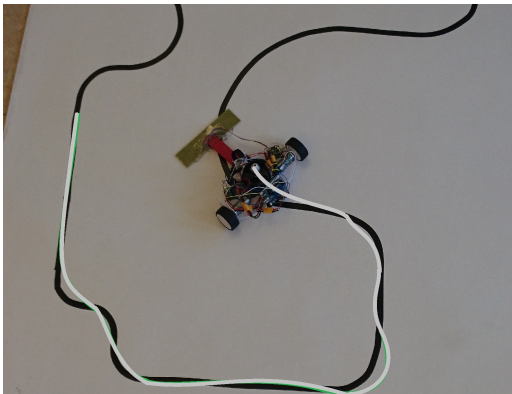
4.4.3 Mätning av ökad kraft på bakdäck med olika fläkthastigheter

Vid test av fläkten gjordes sex mätningar där fläkten successivt ökades. Resultaten för dessa illustreras i figur 4.24. Cirkelar motsvarar mätpunkter. Då fläkten kördes på 100 % mättes en ökad normalkraft på 28.4 N.

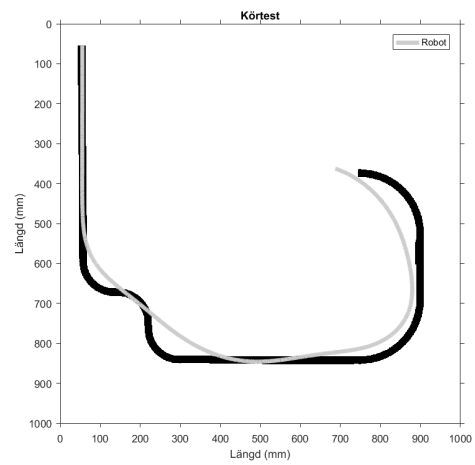
**Figur 4.24:** Ökad kraft på bakdäck

4.4.4 Jämförelse av simulinkmodell och verklighet

I figur 4.25 visas hur roboten kör och en simulerad körning. Den verkliga roboten har en mål hastighet på 1.3 m/s medan simuleringens mål hastighet är 2 m/s. Det kan ses att roboten kör en liknande bana jämfört med simuleringen men vid en lägre hastighet. När roboten ökar hastigheten ökar oscilleringarna och avviker mer från simuleringen.



(a) Testkörning med Robot



(b) Testkörning i simulering

Figur 4.25: Testkörning med simulering och robot

5

Diskussion

5.1 Simulering

Den verkliga roboten har betydligt mer oscilleringar än vad den simulerade roboten visade. Detta beror delvis på att simuleringen inte modellerar fördröjningar i systemet. Fördröjningar uppskattades vara för små för att ha någon inverkan på systemet under simuleringen men detta är en av anledningarna till att robotens oscilleringar är större än i simuleringen. När hastigheten ökar ökar också oscilleringarna, det sker eftersom med högre hastighet krävs ett snabbare system och transportfördröjningarnas inverkan ökar. Simuleringen modellerar ej heller de förluster som uppstår i robotens motorer, växellåda och hjul. Detta togs hänsyn till vid konstruktionen och motorer som klarade simuleringskraven med marginal valdes.

5.2 Komponentval

Besluten av vilka komponenter som skulle användas motiverades utav simuleringsresultaten. Med avseende på komponentval var det rimligt att lägga stor vikt på simulering och förarbete istället för att impulsköpa och besluta.

Motorerna som används är överdimensionerade för roboten, men då effektkraven antas vara större än enligt simuleringen är detta nödvändigt för att efterlikna simuleringsresultaten. Motorerna har både encoders, vilket underlättade återkopplingen till motorkontrollerna, och växellådor som levererade precis det varvtal som hjulen kräver för att nå upp till 2 m/s. Detta eliminerade stora delar av extraarbete som att skaffa encoders med rätt specifikationer och konstruera en egen växellåda.

Motorkontrollerna som används valdes på grund av dess inbyggda PID-regulatorer vilket förenklar programmering av roboten och sänker de transportfördröjningar som uppstår när arduinon måste göra mer avancerade beräkningar.

5.3 Konstruktionsbeslut

Beslutet i att göra chassit modulärt grundas delvis i de begränsningar som existerar i storleken på objekt som kan skrivas ut ur 3D-skrivaren. Ett modulärt

chassi innebär även en ökad frihet i att ändra design av de olika komponenterna utan att behöva konstruera om hela roboten.

På bottenplattan gjordes skenor för att kunna förflytta motorerna framåt och bakåt längs platta. Detta gav en möjlighet att flytta robotens masscentrum i förhållande till hjulen. Genom att flytta masscentrum kan man förhindra att roboten stegrar (se ekvation 2.12). Dock sker detta på bekostnad av att mer kraft går ner i styrkulan och mindre i bakdäcken. Att kunna flytta motorerna medförde även att rotationscentrum förflyttades i förhållande till sensorerna. Detta kräver mindre kalibreingar i styrkoden men påverkade inte beteendet något nämnvärt. En annan effekt av detta var att luft sögs ner i dessa skåror och kan ha minskat den downforce som fläkten genererade.

Val av fläkt grundar sig i beräkningar med ökning av normalkraften vid kurvtagning samt simulinkmodellen för att klara banan utan sladd eller glidning. Detta kan då undvikas med pålagd kraft neråt. Valet av den fläkttyp som valdes var inte självklart. Diskussion kom upp om det verkligen behövs eller om man istället för fläkt skulle montera på propellrar som pressar roboten nedåt. Enligt beräkningarna behövdes en ytterligare downforce på nästan 30 N om roboten inte skulle glida vid körning i en kurva med radien 75 mm och en hastighet på 2 m/s vilket är den önskade hastigheten. Se ekvation 2.10 Vad som märktes ganska tidigt vid körning av roboten var att vid höga hastigheter slirar roboten utan pålagd tryck/downforce, därav ansågs det vara ett rimligt val att köpa till en fläkt. Utöver fläkten gjordes kjolen på bottenplattan för att begränsa inflödet av luft och öka trycket under bottenplattan av samma anledning som fläkten.

I den verkliga roboten användes endast sex sensorer eftersom varianten av arduinon som använts endast har sex portar för analoga signaler. Enligt simuleringen ska antal sensorer mellan 8 och 6 ej ha någon större inverkan på robotens hastighet och oscillering. Men detta tillsammans med att simuleringen ej tar hänsyn till transportfördröjningar ger att roboten oscillerar mycket mer än i simuleringen.

5.4 Utvärdering och tester

Testerna som utfördes var i vissa fall inte optimala och återspeglar inte helt samma miljö som roboten befinner sig i vid drift.

Testet som gjordes för att undersöka den ökade normalkraften på bakdäcken (kapitel 3.4.3 och 4.4.3) gjordes med en för liten platta under roboten. Vilket medförde ett större luftflöde än vad roboten vanligen upplever när den köra på platt mark. Detta kan definitivt ha bidragit till fel i mätvärden. Dock gav detta test ett bra riktvärde för att avgöra i vilken magnitud kraften var i. Ytterligare saker som borde gjorts i detta test är att undersöka hur markfrigången påverkar fläktens förmåga att skapa en nedåtriktad kraft.

Testet för friktion att beräkna koefficientent (kapitel 3.4.2 och 4.4.2) gav rimliga resultat. Dock kan resultaten komma att ha påverkats av en dammig miljö, då testet skedde i en verkstad. Noggrannheten på det vinkelbara skruvstället är även något som är okänt. Trots detta var resultatet 0.83 något som även står i formelsamlingen *Engineering Toolbox*[3]. Därav anses resultatet vara på en rimlig magnitud.

6

Slutsats

Anledningen till att roboten inte klarar samma hastigheter som i simuleringen är att simuleringsmodellen inte tar hänsyn till några förluster, transportfördröjningar och antar perfekt grepp. Simuleringens vikt vid framtagningen av roboten var av stor relevans, både för testning av designval och optimering av PID-parametrar. Simuleringen gav även en bra förståelse för systemets beteende med avseende på hur kaskadregleringen och dess parametrar påverkar grepp och effektkrav. Simuleringen underlättade även vid komponentval och kravsättning.

Vid design av en robot kan simulering hjälpa till mycket i designprocessen av styrsystem och mekanik då kombinationer av olika delar och lösningar kan testas effektivt. Designen av reglersystemet kan även optimeras för att tillåta en enkel implementation. Tidsfördröjningar som ej tagits hänsyn till i simulering påverkar prestandan av roboten negativt då oscillationer uppstår. Förluster och sladdning tillkommer vilket, om ej inräknat i simulering, påverkar prestandan negativt.

Om projektet skulle fortgå i framtiden skulle fokus främst läggas på inköp av bättre mikrokontroller som bland annat är snabbare och har fler analogportar vilket skulle möjliggöra användandet av fler sensorer. Dessutom hade optimering av regleringen varit att prioriterat då de inte fanns mycket tid till optimering efter konstruktion. Det hade även varit intressant att undersöka flera olika reglermetoder och se hur dessa presterar.

Vid vidare arbete bör konstruktionen ses över och en analys bör göras över designen på roboten med avseende på bland annat mekanik-beräkningar så som vart man ska lägga tyngdpunkten för optimal körning.

Sammanfattningsvis kan man se att projektet nådde de förväntningar som gruppen hade och att det blev ett givande resultat.

Litteraturförteckning

- [1] Bäckström, L. Pettersson, Tomas. Tillberg, Hanna. Åkerlund, Fredrik. (2015) Linjeföljare med STM32. <http://publications.lib.chalmers.se/records/fulltext/219008/219008.pdf>
- [2] Wit, C. Crane and D. Armstrong, "Autonomous ground vehicle path tracking", Journal of Robotic Systems, vol. 21, no. 8, pp. 439-449, 2004. DOI:10.1002/rob.20031
- [3] "Friction and Friction Coefficients, "The Engineering ToolBox, maj 2017 [Online] Tillgänglig: http://www.engineeringtoolbox.com/friction-coefficients-d_778.html. [Hämtad: 12 maj, 2017].

A

Kravspezifikation

Kravspecifikation - Linjeföljande robot

1 - lågt
10 - högt

Kriterier	Målvärde	Enhet	K/Ö	Vikt	Referens
-----------	----------	-------	-----	------	----------

1. Prestanda

1.1	Arbetshastighet	> 1	m/s	K		Kandidatgruppen
1.2	Önskad arbetshastighet	> 2	m/s	Ö	8	Kandidatgruppen
1.3	Stillastående till arbetshastighet	0.1	s	K		Kandidatgruppen
1.4	Dataöverföring			K		Kandidatgruppen
1.5	Trådlös dataöverföring			Ö	5	Regler i Robot SM
1.6	Del av robot måste hållas över linje			K		Regler i Robot SM
1.7	Ingen glidning mot marken			Ö	8	Kandidatgruppen
1.8	Bantid	180	s	K		Kandidatgruppen
1.9	Önskad bantid	< 30	s	Ö	10	Kandidatgruppen
1.10	Batteritid	>180	s	K		Kandidatgruppen
1.11	Batteri spänning					Kandidatgruppen
1.12	Uppdateringsfrekvens					Kandidatgruppen
1.13	Störningar					Kandidatgruppen
1.14	Fel (mellan bör och ärvärde)					Kandidatgruppen
DC motor						
1.14	Spänning	12	V	K		Komponentdimensioner
1.15	Moment	0.075	Nm	K		Komponentdimensioner
1.16	Max RPM	1500	RPM	Ö	8	Önskad maxhastighet
1.17	Arbets RPM	1200	RPM	Ö	8	Önskad hastighet

2. Konstruktion & tillverkning

2.1	3D printa chassi			Ö	2	Kandidatgruppen
2.2	Möjligt att konstruera med resurser på Chalmers			Ö	6	Kandidatgruppen
2.3	3D printa hjul			Ö	5	Kandidatgruppen

3. Material och komponenter

3.1	Använda komponenter som finns tillgängliga via chalmers			K		Kandidatgruppen
3.2	Använda material som finns tillgängliga via chalmers			K		Kandidatgruppen
3.3	Använda komponenter som redan finns på chalmers			Ö	5	Kandidatgruppen
3.4	Använda material som redan finns på chalmers			Ö	5	Kandidatgruppen
3.5	Robotkroppen tillverkat i polymer			K		Kandidatgruppen

4. Underhåll

4.1	Lätt att rengöra			Ö	4	Kandidatgruppen
4.2	Lätt att reparera			Ö	6	Kandidatgruppen

5. Dimensioner

5.1	Längd	< 0.4	m	K		Regler Robot-SM
5.2	Bredd	< 0.3	m	K		Regler Robot-SM
5.3	Massa	< 3.0	kg	K		Regler Robot-SM
5.4	Önskad massa	< 0.5	kg	Ö	8	Kandidatgruppen

6. Kostnader

8.1	Totalkostnad	< 5000	SEK	Ö	10	Budget
8.2	Testplattform	< 2000	SEK	Ö	1	Kandidatgruppen
8.3	Prototyp 1	< 1500	SEK	Ö	1	Kandidatgruppen
8.4	Prototyp 2	< 1000	SEK	Ö	1	Kandidatgruppen
8.5	Slutprodukt	< 500	SEK	Ö	1	Kandidatgruppen

7. Säkerhet

9.1	Ej skada människor			K		Regler Robot-SM
9.2	Ej skada materiell omgivning			K		Kandidatgruppen

B

Regler för Robot SM

Regler för Linjeföljning under Robot-SM

Uppdateringar

- 2014-04-27 - *Specifikation på hur långt ifrån hindret roboten får släppa samt återta linjen i avancerad linjeföljning (15 cm före och efter).*
- 2014-02-16 - *Punkt 4.2.3 (90°-svängar) är borttaget, eftersom det var öppet för tolkningar. Skarpa vinklar är alltså inte tillåtet på planen. Detta kan dock vara tillåtet i Avancerad linjeföljning.*
- 2014-02-16 - *Robotens begränsningar är minskade till 30cm bred och 40cm lång, samt max 3 kg massa.*
- 2014-02-16 - *Definition av när en robot anses följa linjen.*

Sammanfattning av reglerna

Linjeföljning går helt enkelt ut på att följa en bana utmärkt av en linje. Reglerna för Linjeföljning bygger delvis på de regler som används i Robotchallenge, Europas största robottävling. Nedan följer reglerna i sin helhet. Använd ert sunda förnuft när ni tolkar reglerna. Om ni är tveksamma på hur ni ska tolka dem rekommenderar vi att ni hör av er snarast så att vi kan räta ut alla frågetecken i god tid. Ni kontaktar oss enklast på e-postadressen info@robotism.se.

Regler för Linjeföljning samt Avancerad linjeföljning

1. Mål

1. Målet är att roboten ska följa en bana utmärkt av en linje. Roboten måste hela tiden följa linjen. Hela roboten måste börja bakom startmarkeringen och räknas som i mål när någon del av roboten har passerat startmarkeringen efter ett komplett varv runt hela banan.

2. Roboten

1. Roboten skall vara autonom (ingen yttre styrning är tillåten).
2. Roboten får anpassas och ändras under tävlingen så länge reglerna efterföljs. All fysisk förändring av roboten ska rapporteras till tävlingsledningen för godkännande. Tävlingsledningen behöver dock ej underrättas om det endast gäller förändring av mjukvara
3. Roboten får inte vara bredare än 30 cm eller längre än 40 cm.
4. Robotens massa får inte överstiga 3 kg.
5. Roboten måste kunna ses av ett automatiskt tidtagningssystem. Den måste därför på minst ett ställe vara mer 5 cm hög (förslagsvis genom en flagga eller liknande ifall roboten är lågt byggd).
6. Roboten får inte vara byggd för att skada robot, människa eller omgivning.
7. Roboten får inte skrapa eller på annat sätt göra åverkan på eller skada arenan (t.ex. lämna efter klistriga hinnor, färg och liknande).

3. Inspektion

1. Tävlingsledningen har rätt att när som helst kontrollera att roboten uppfyller kraven för tävlingen.

2. Tävlingsledningen har rätt att kräva att roboten modifieras för att uppfylla tävlingskraven.
3. Om problemen inte åtgärdats kan roboten diskvalificeras från tävlingen.

4. Arenan

1. Arenan består av ett vitt underlag med en svart linje ovanpå. Linjen kan antingen vara målad eller tejpad. Arenan ska i bästa fall bestå av målat trä, men kan även bestå av annat material (såsom papper eller plast).
2. Linjen uppfyller följande restriktioner:
 1. Linjen är 19 ± 2 mm bred (linjens bredd kan variera inom dessa gränser på olika delar av arenan).
 2. Inga segment av linjen är närmare varandra än 15 cm (mätt från centrum på linjerna).
 3. Arenans kant är aldrig närmare linjen än 15 cm (mätt från centrum på linjen).
 4. Kurvradien är minst 7.5 cm.
3. I början av banan finns ett s.k. startområde. Runt startområdet står portar för tidsmätning. Avstånden mellan dessa portar är minst 30 cm.
4. Det kan förekomma skarvar, ojämnheter och håligheter i arenan. Även linjen kan vara skarvad, med ojämnheter och håligheter som följd. Dessa effekter kommer att minimeras, men kan förkomma.

5. Tävlingen

1. Roboten har 3 minuter på sig att ta sig runt banan.
2. Tävlingen går till så att den tävlande placerar ut roboten bakom startlinjen. Därefter aktiverar den tävlande roboten på domarens signal.
3. Tiden startar när roboten kör förbi porten första gången och stannar när någon del av roboten går förbi porten andra gången efter att ha kört ett helt varv (varje del av linjen har varit överlappad). Tiden ska om möjligt mätas med ett automatiskt system, men kan (ifall ett sådant inte finns tillgängligt) mätas av en domare med tidtagarur.
4. Roboten måste ha markkontakt hela tiden.
5. Roboten måste följa linjen hela tiden. En robot anses följa linjen ifall någon del av roboten överlappar linjen. Om roboten lämnar linjen måste den återta linjen igen på det stället där linjen tappades eller tidigare.
6. Tävlingsformatet kommer att bero på antalet deltagare, så detta kommer att bestämmas närmare tävlingen.

6. Avancerad linjeföljning

1. Avancerad linjeföljning går till på precis samma sätt som vanlig linjeföljning förutom att det finns en del hinder på banan. Hindren är potentiella 90° -hörn, en tunnel, ett avbrott och ett objekt i form av en tegelsten eller motsvarande. Dessa beskrivs mer ingående nedan:
2. Det kan förekomma skarpa vinklar på linjen ned till 90° . Vid dessa områden kan linjen överlappa.
3. Linjen kommer att gå genom en tunnel. Tunneln är en låda som är minst 30 cm bred och 30 cm hög. Roboten måste fortfarande följa linjen trots den ändrade ljussättningen.
4. På ett ställe på linjen finns ett avbrott. Avbrottet är 10 cm långt och realiserad med vit tejp.

Avbrottet är placerat på en raksträcka, vilket innebär att det kommer att vara rak linje innan och efter avbrottet.

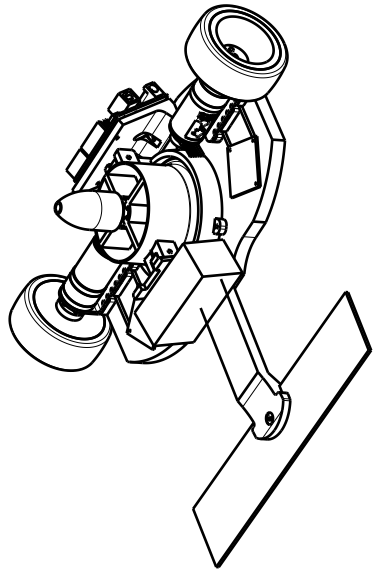
5. På banan finns ett objekt utplacerat. Objektet är en tegelsten (eller motsvarande) med ungefärliga dimensioner 25 cm x 12 cm x 6 cm. objektet är placerad mitt på linjen, och roboten måste vika av från linjen, åka runt objektet och hitta tillbaka till linjen igen för att fortsätta följa den. Roboten får vika av från linjen senast 15 cm innan objektet och återta linjen senast 15 cm efter objektet. Roboten får inte knuffa bort objektet.

7. Tveksamheter

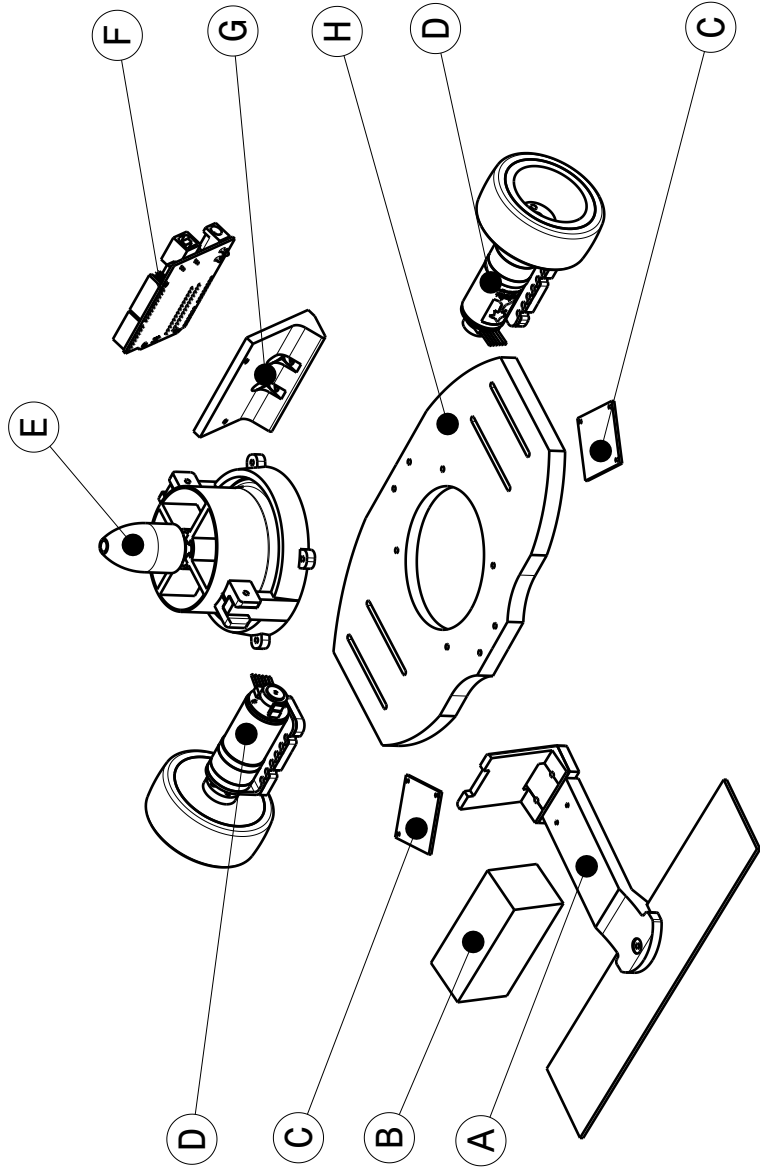
1. Använd ert sunda förnuft när ni tolkar reglerna. Vid tveksamheter och tvister gällande regeltolkningar har huvuddomaren beslutsrätt att avgöra vad som är rätt och fel.

C

Ritning för robot



3D vy
Skala: 1:5

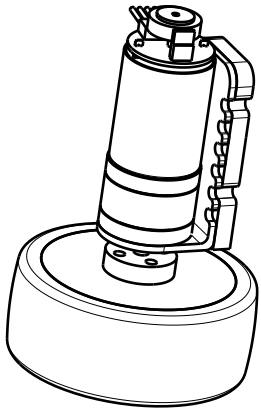


Bill of Material

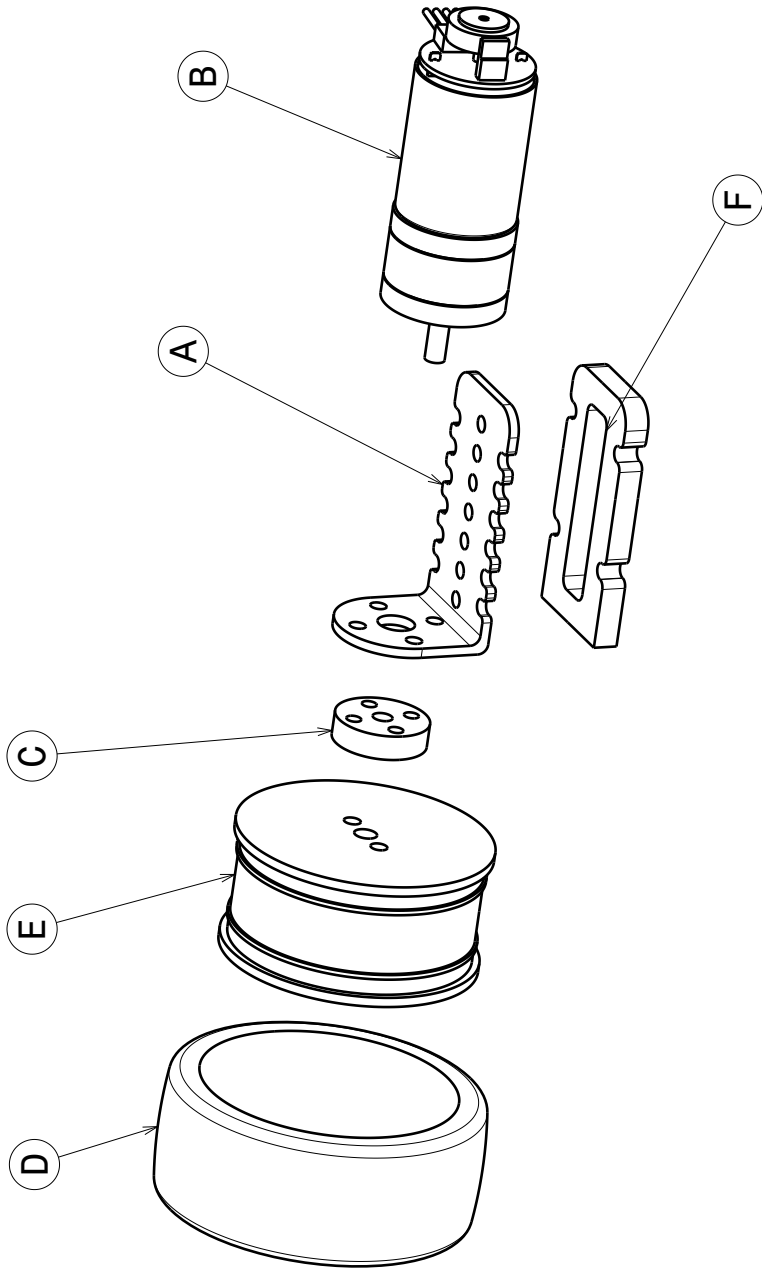
Number	Quantity	Part Number
A	1	Monterad sensorarm
B	1	Batteri
C	2	Motorkontroller (DC-motor)
D	2	Hjulupphängning
E	1	Monterad fläkt
F	1	Arduino UNO
G	1	Mikrokontrollerfäste
H	1	Bottenplatta

DESIGNED BY: Robin Hafström Emelie Johansson		NAME
DATE: 2017-05-11		
SCALE 1:4	PAGE 1/11	

Sprängskiss av hela roboten



Isometric view
Scale: 1:2



Bill of Material

Number	Quantity	Part Number
A	1	Motorfäste
B	1	Pololu 9.7:1 Metal Gearmotor 25Dx48L
C	1	Hjulnav
D	1	HPI Racing radial tier 26mm
E	1	Fälg
F	1	Motorfästedistanser

DESIGNED BY:

Robin Hafström
Emelie Johansson

DATE:

2017-05-11

SCALE

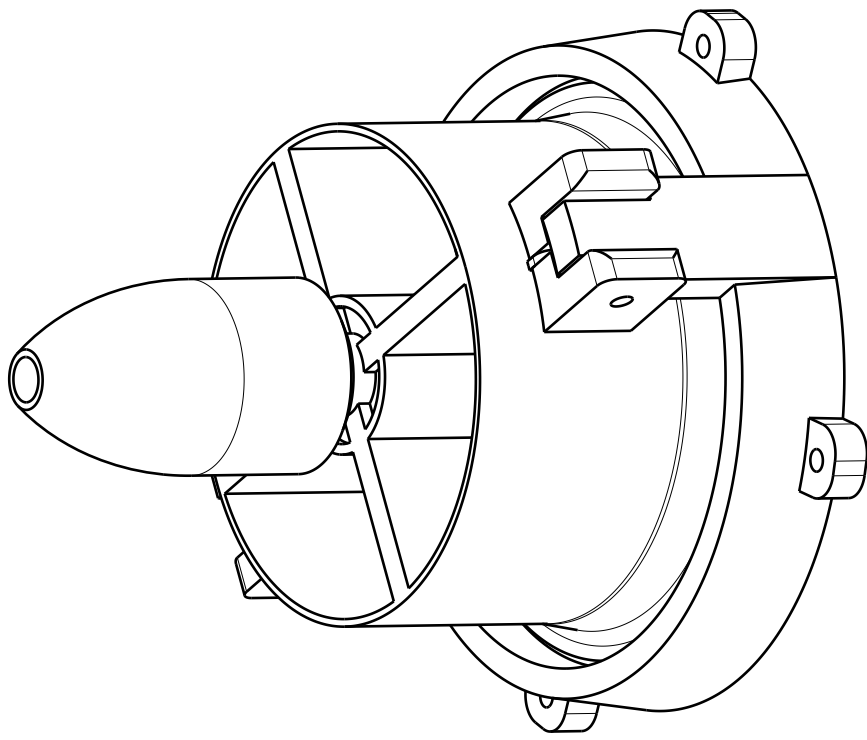
2:3

PAGE

2/11

NAME

Sprängskiss av moterad motor



Isometric view

Scale: 1:1

Bill of Material

Number	Quantity	Part Number
A	1	Kon
B	1	EDF 64mm med Tiger AT2010 5000 kv motor
C	4	Klämfäste

DESIGNED BY:

Robin hafström

Emelie Johansson

DATE:

2017-05-11

NAME

Sprängskiss av monterad fläkt

SCALE

1:2

PAGE

3/11

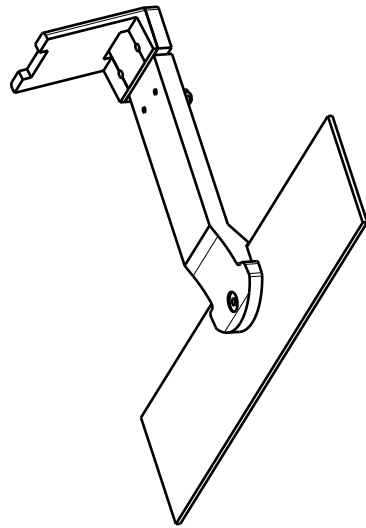
4 3 2 1

A

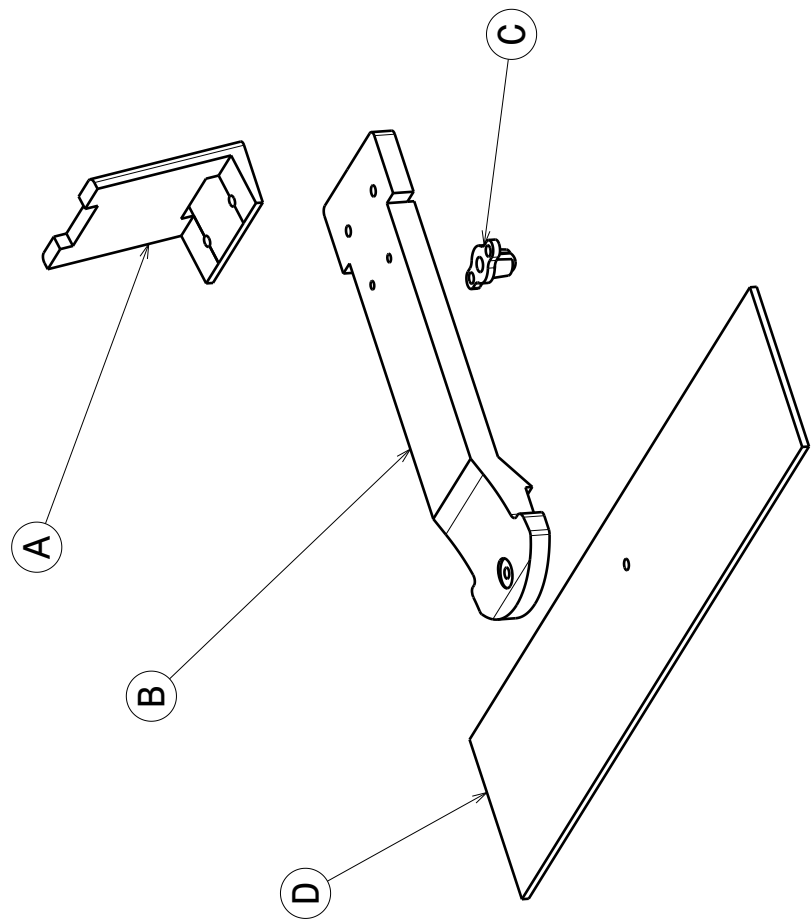
m

C

D



Isometric view
Scale: 1:3



Bill of Material

Number	Quantity	Part Number
A	1	Batterifäste
B	1	Sensorarm
C	1	Pololu Metal Ball
D	1	Caster 9.5mm
	1	Sensorrigg

DESIGNED BY:		NAME	
Robin Hafström		Emelie Johansson	
DATE:		2017-05-11	
SCALE	PAGE		
1:2	4/11		

Sprängskiss av monterad
sensorrigg

A

D

4 3 2 1

4

B

C

D

A

4

3

2

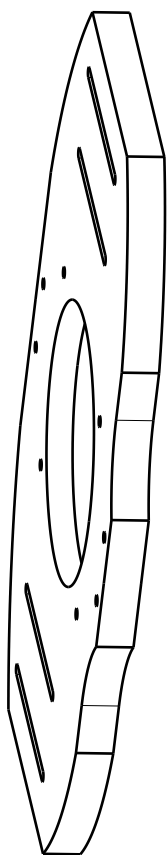
1

4

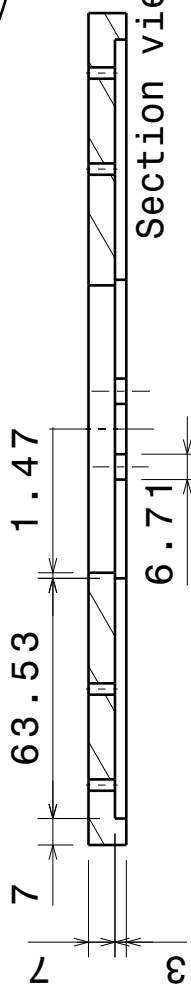
3

2

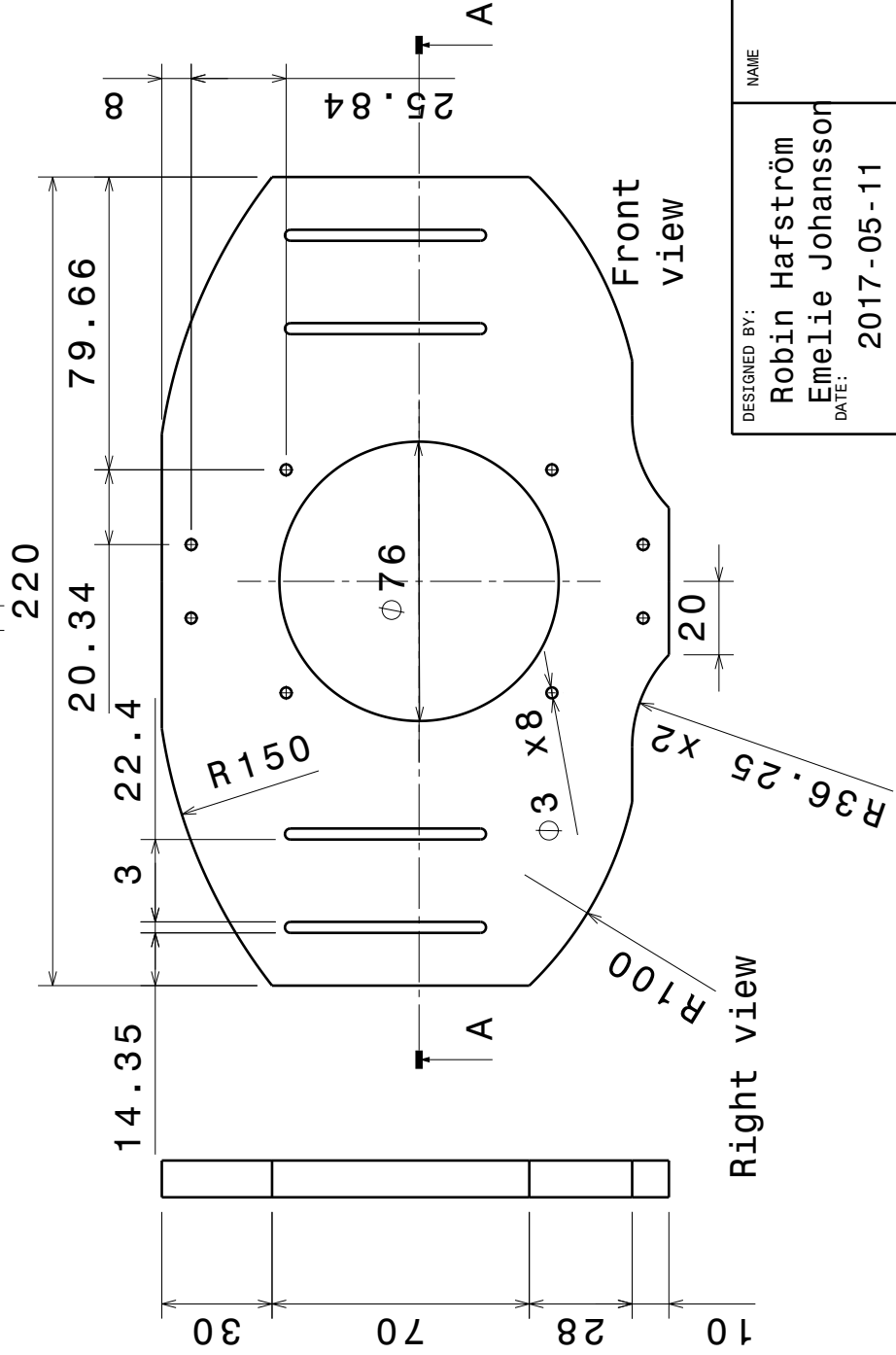
1



Isometric view



Section view A-A



Front view

Right view

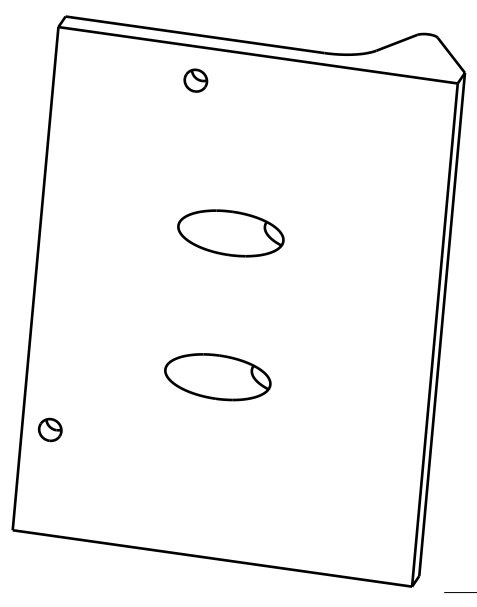
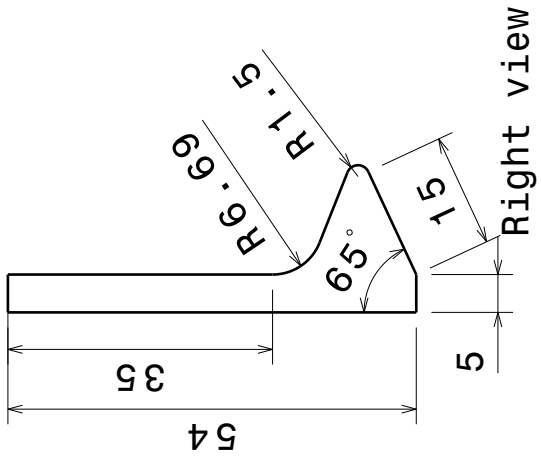
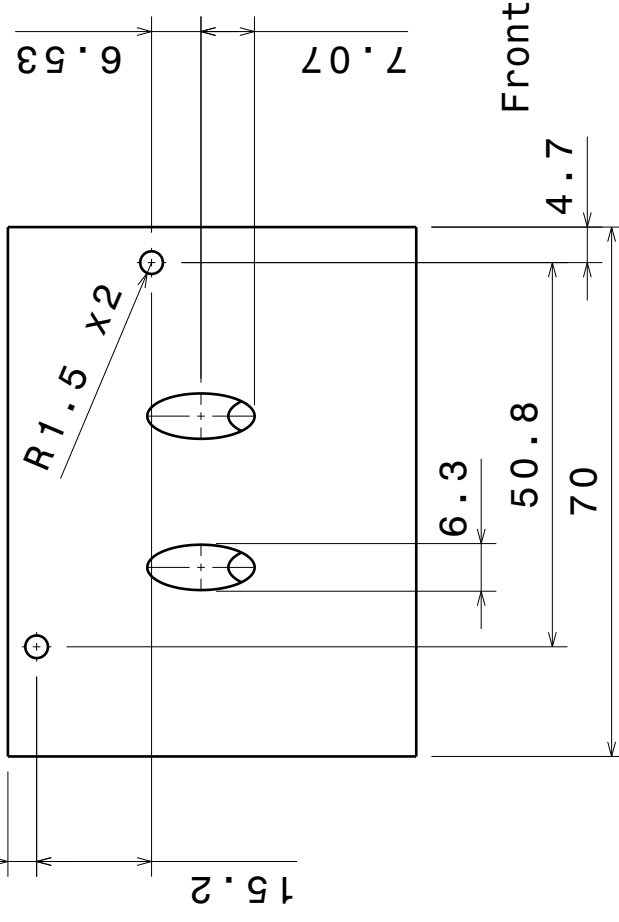
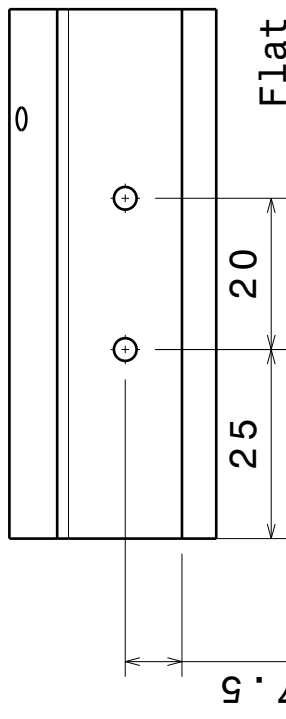
Back view
Scale: 1:4

DESIGNED BY:		NAME	
Robin Hafström		Emelie Johansson	
DATE:		2017-05-11	
SCALE	PAGE		
1:2	5/11		

Bottenplatta

A

D



DESIGNED BY:		NAME	
Robin Hafström		Microkontrollerfäste	
Emelie Johansson		alla mått är i mm	
DATE: 2017-05-11		SCALE	PAGE
		1:1	6/11

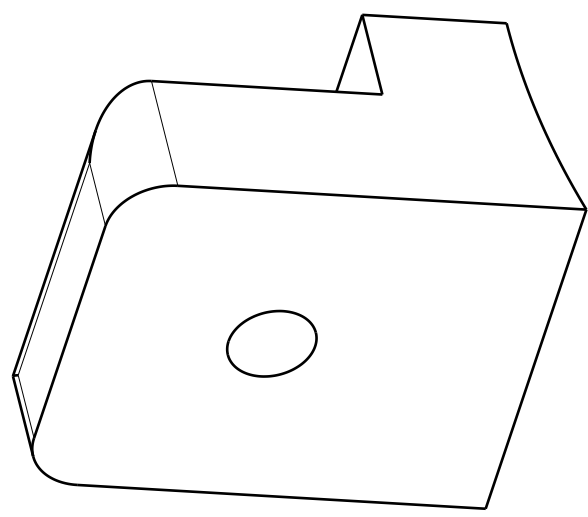
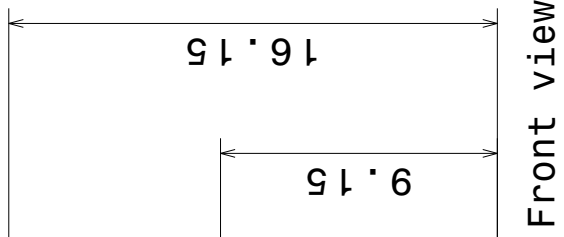
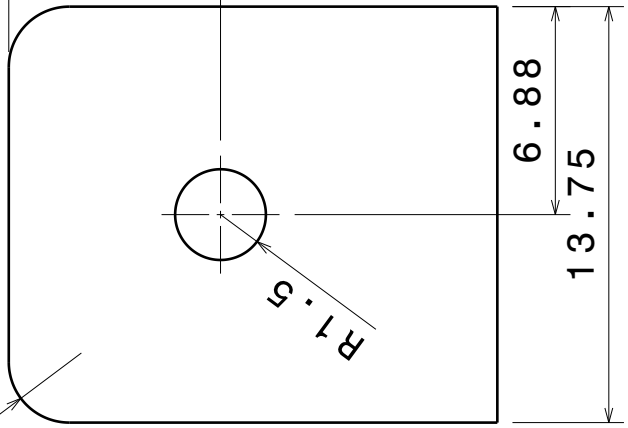
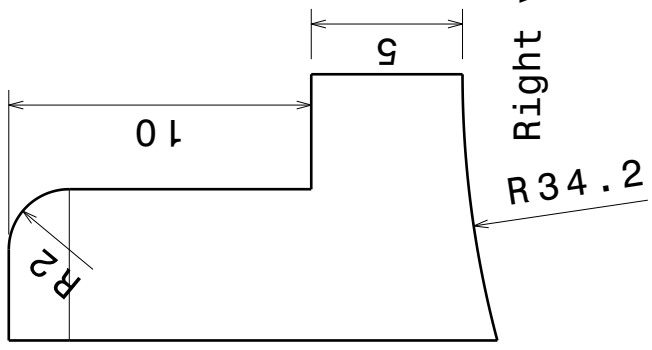
4 3 2 1

A

B

C

D



DESIGNED BY: Robin Hafström Emelie Johansson		NAME
DATE: 2017-05-11		
SCALE 4:1	PAGE 8/11	

Klämfäste

alla mått är i mm

Flat top view

A

D

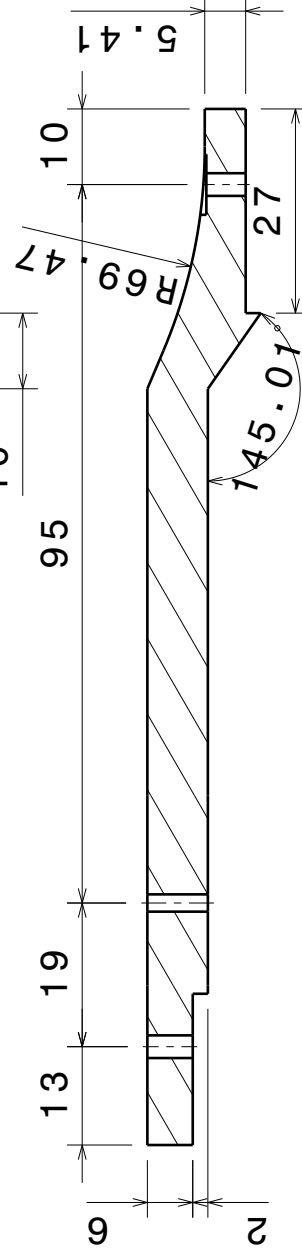
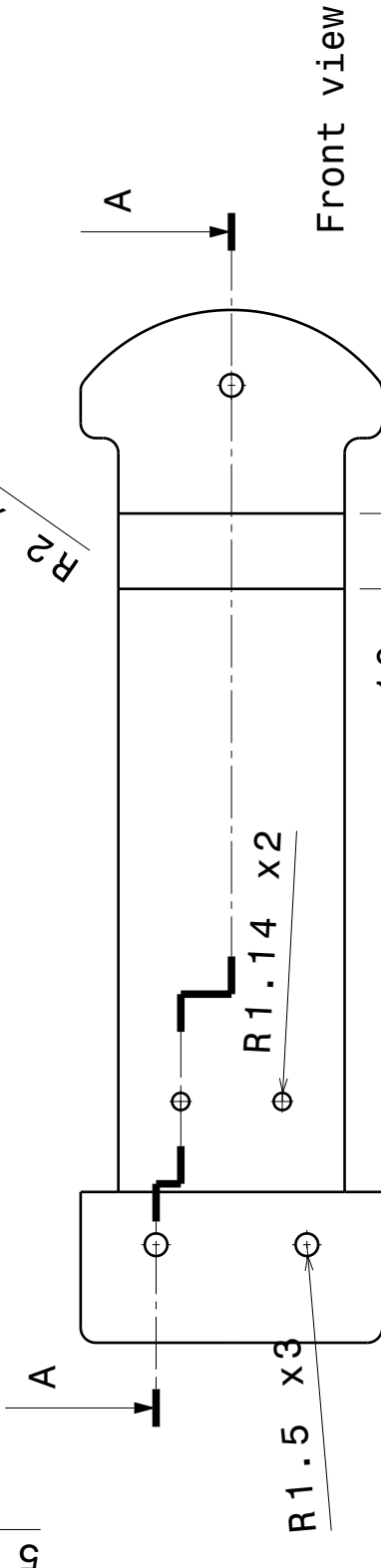
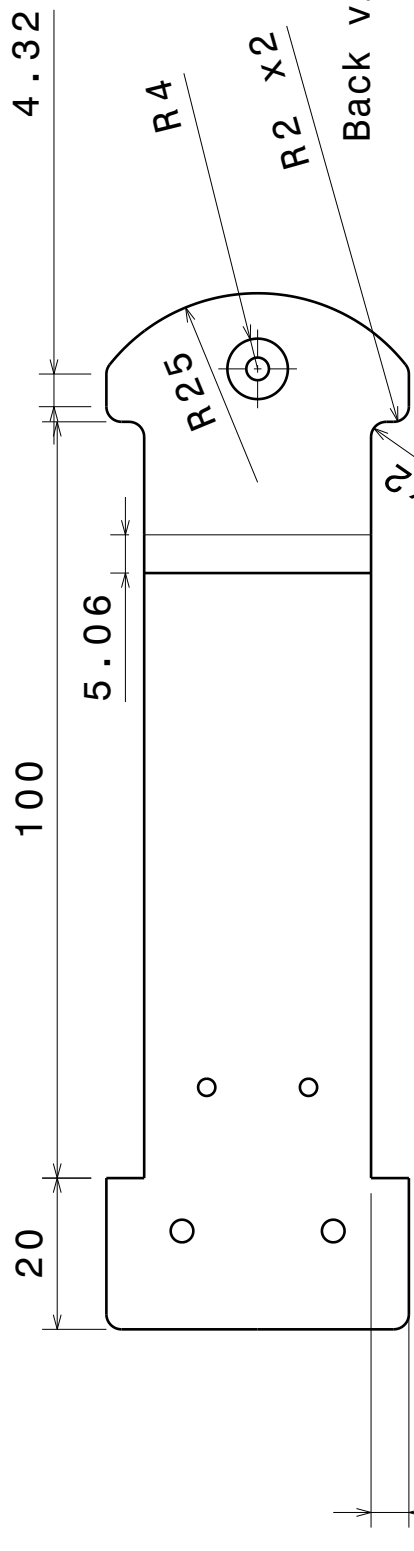
4 3 2 1

4 3 2 1

mm

0

0



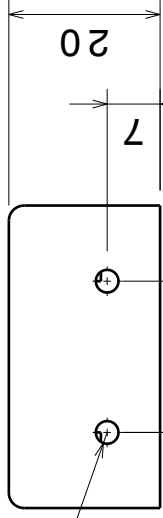
DESIGNED BY:		NAME	
Robin Hafström		Sensorarm	
Emelie Johansson		alla mått är i mm	
DATE:		1	
2017-05-11			
SCALE	PAGE		
1:1	9/11		

4 3 2 1

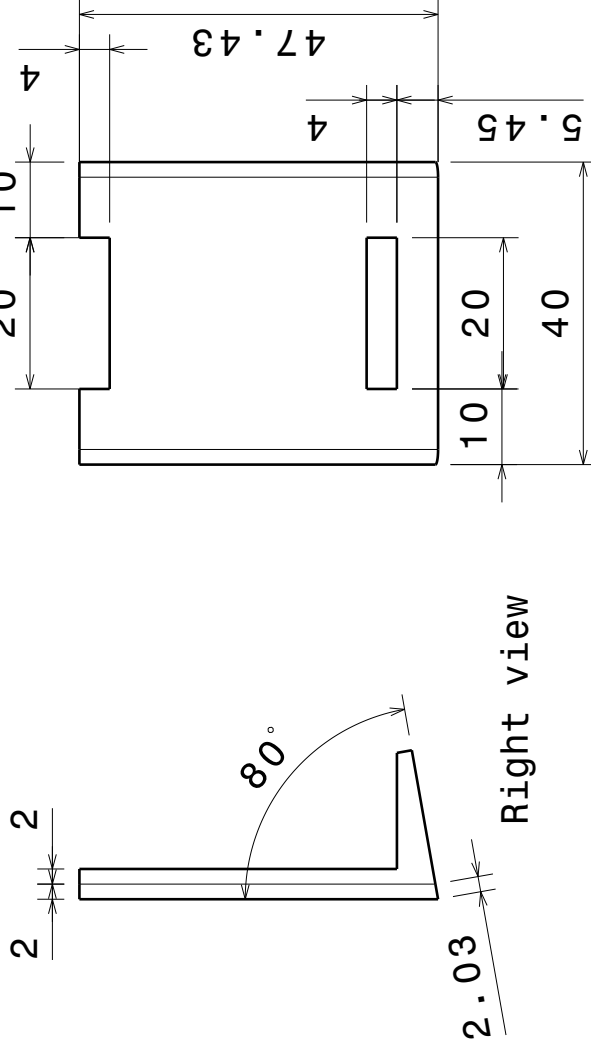
A

D

$R1.5 \times 2$

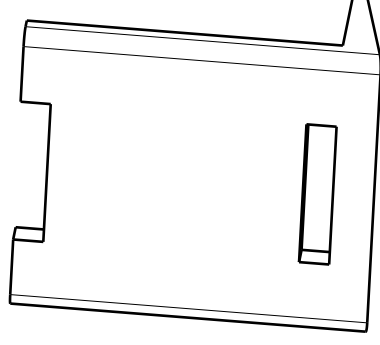


Bottom view without angle



Right view

Front view



Isometric view

NAME

DESIGNED BY:
Robin Hafström
Emelie Johansson

DATE:
2017-05-11

SCALE

1:1

PAGE

10/11

Batterihållare

alla mått är i mm

D

A

4

3

2

1

4

3

2

1

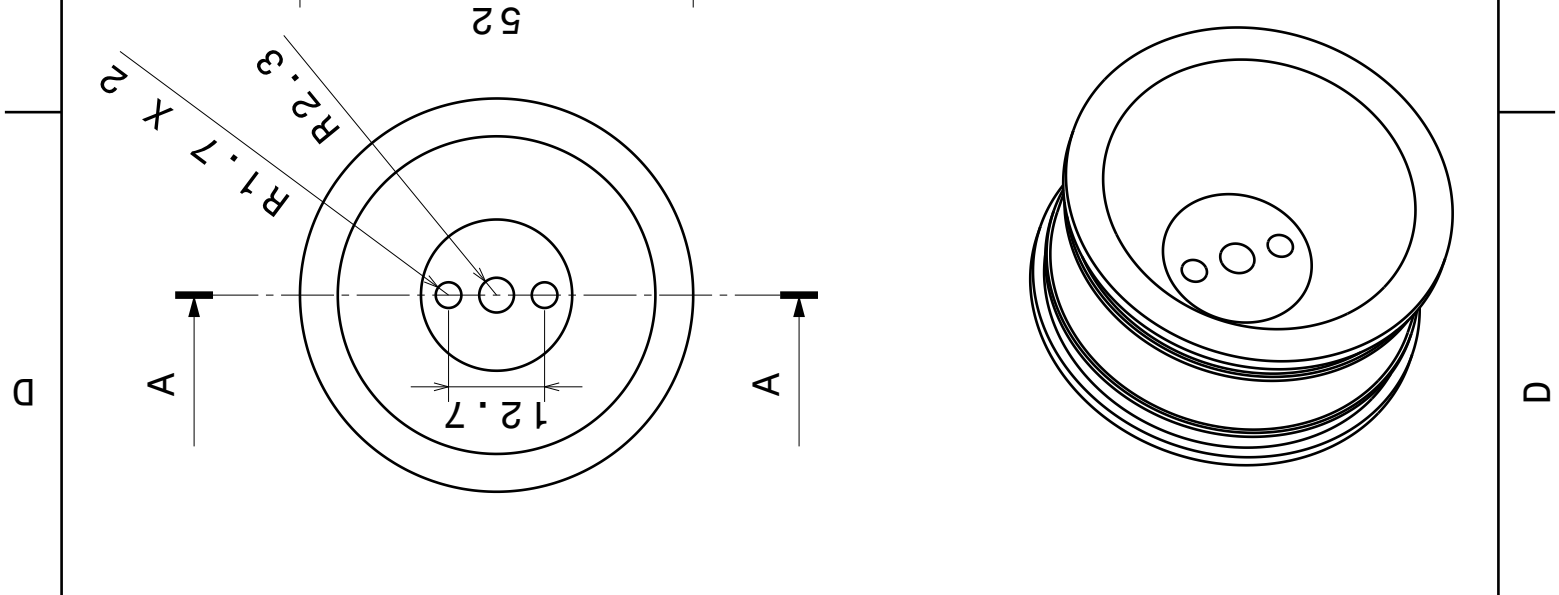
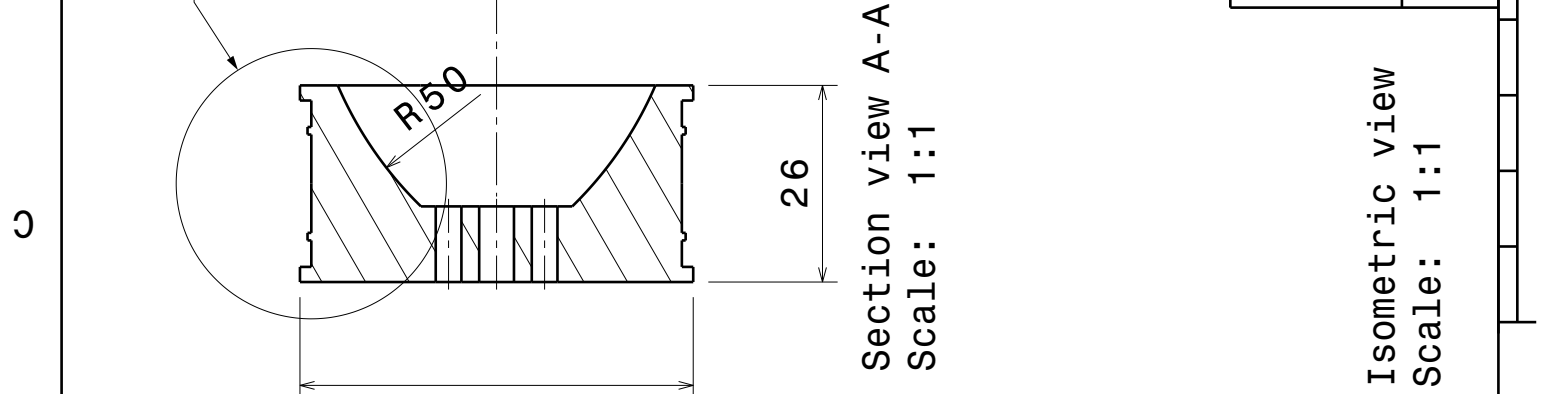
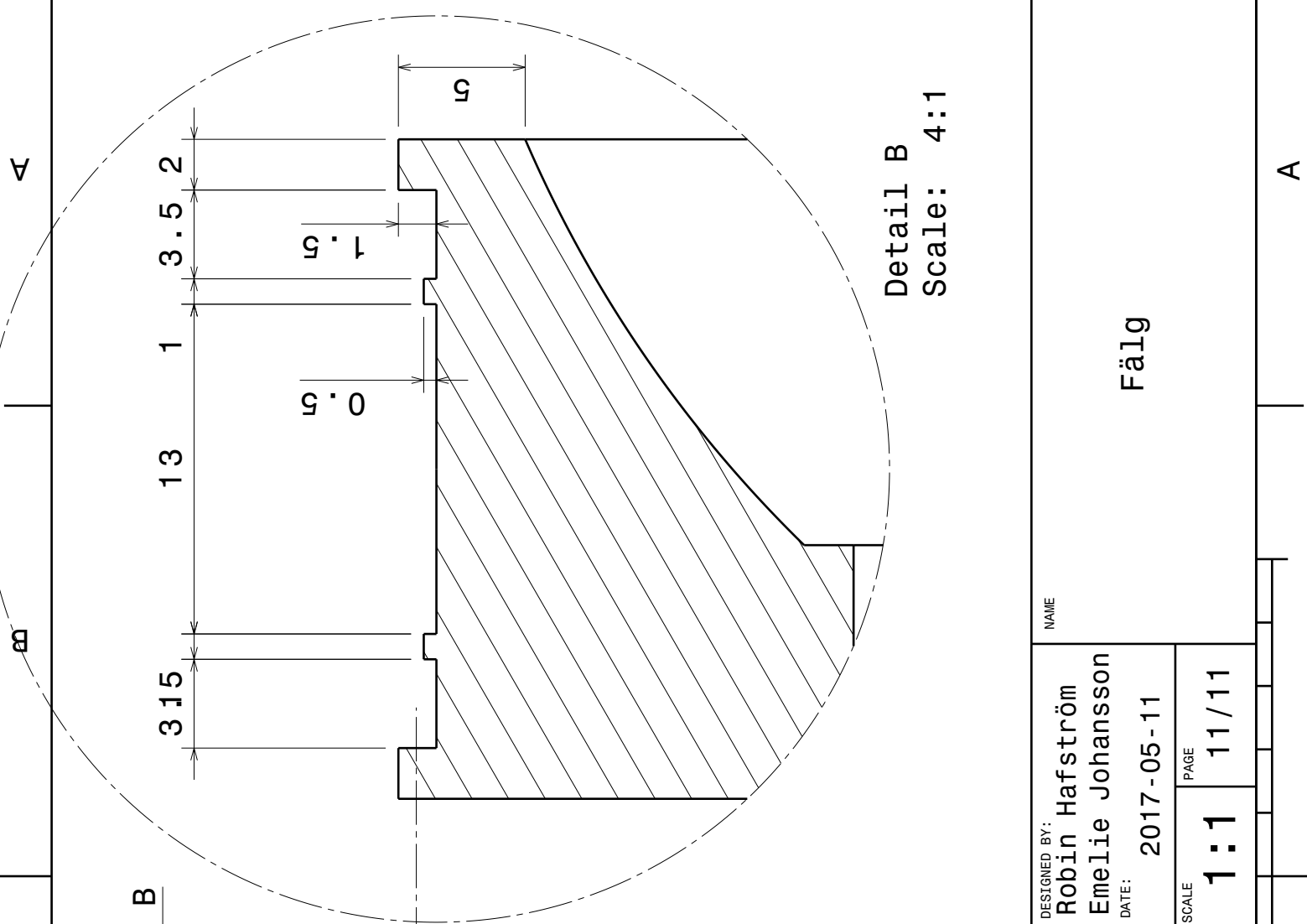
A

B

C

D

4 3 2 1



4 3 2 1

DESIGNED BY:		NAME	
Robin Hafström		Fälg	
Emelie Johansson			
DATE:		11/11	
2017-05-11		PAGE	
SCALE		1:1	

