



**CHALMERS**



# Konstruktion av en självkörande enhjuling

Konceptvalidering

Kandidatarbete inom systemteknik och reglerteknik

Alfred Aronsson  
Andreas Karlsson  
Edvin Fredlund  
Jawed Hakimi  
Jonathan Lööv  
Pummarin Khamkhachorn

**INSTITUTIONEN FÖR ELEKTROTEKNIK  
SYSCON**

---

CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2023  
[www.chalmers.se](http://www.chalmers.se)

CHALMERS TEKNISKA HÖGSKOLA

KANDIDATARBETE VID SYSCON

---

# Konstruktion av en självkörande enhjuling

EENX16-VT23-07

## Kandidatsrapport

---

Alfred Aronsson  
Andreas Karlsson  
Edvin Fredlund  
Jawed Hakimi  
Jonathan Lööv  
Pummarin Khamkhachorn

*Examinator:*  
Jonas Sjöberg

*Handledare:*  
Anders Grauers

26 maj 2023

## Sammanfattning

I denna rapport utvecklas en prototyp för en självkörande och -balanserade motoriserad enhjuling. Enhjulingens syfte är att användas i tester för bilars ADAS (Advanced Driver Assisting Systems, förarassistanssystem), där i nuläget självkörande cyklar används. Testerna ska se till att ADAS kan undvika VRU:er (Vulnerable Road Users, sårbar trafikant) bättre, genom att bidra till en större samling testobjekt att använda inom testerna.

En enhjuling byggs i metall med ett drivhjul i längsled och ett reaktionshjul i sidled samt fästpunkter för elektronik. Motorerna drivs via två VESC (Vedder Electronic Speed Controller, motordrivare) som också ger återkoppling i motorernas varvtal. Flera filter- och reglersystem utvecklas och utvärderas. Systemen implementeras på en STM32 mikrokontroller, som via koppling till motordrivarna, samt via data från en kombinerad accelero- samt gyrometer ska balansera enhjulingen och låta den följa en väg, antingen fördefinierad eller kontrollerad via en extern kontroll.

Enhjulingen och dess tillhörande reglersystem ger ett stabilt system som kan balansera väl i stationärt tillstånd. I sidled kan enhjulingen räta upp sig från en stillastående startvinkel på cirka  $\pm 3$  deg, men i körning när reaktionshjulet rullar har vinklar stora som 4.3 deg uppmäts vilket enhjulingen klarade att balansera. Inget system för att följa en definierad väg implementerades, men ett enkelt test där en fjärrkontroll kontrollerar börvärdet på enhjulingens hastighet i längsled kunde en hastighet på  $0.2 \text{ m/s} \approx 0.7 \text{ km/h}$  accelereras till och hållas utan problem.

## Abstract

In this paper, a prototype for a self-driving and self-balancing motorized unicycle is designed and constructed. The purpose of the unicycle is to be used in tests for ADAS (Advanced Driver Assisting Systems) in the car industry, where currently only self driving bicycles are used. The tests will ensure that the ADAS can better avoid VRU's (Vulnerable Road Users), by contributing to a larger pool of test objects used in the tests.

A unicycle is constructed in metal, with a drive wheel in pitch and a reaction wheel in roll, along with attachment points for electronics. The motors are driven using two VESC (Vedder Electronic Speed Controller) motor drivers which give feedback on the rotational speed of the motors. A number of filters and control systems are created and tested. The system, which is implemented on an STM32 microcontroller and connected to the motor drivers and to a combined gyrometer and accelerometer, shall balance the unicycle and follow a path, either pre-programmed or given via an external remote control.

The unicycle and its control system results in a stable system that can balance well in stationary conditions. In roll, the unicycle can get into a balanced position after an initial stationary starting angle of  $\pm 3$  deg. While driving, as the reaction wheel is already spinning, angles as large as 4.3 deg have been measured, which the unicycle could maintain balance from. No system for following a set path was developed, but a basic test where a remote control controlling the setpoint speed of the unicycle was performed. A speed of  $0.2 \text{ m/s} \approx 0.7 \text{ km/h}$  could be accelerated to and maintained without losing balance.

# Innehåll

0.1	Ordlista	4
<b>1</b>	<b>Introduktion</b>	<b>5</b>
1.1	Bakgrund	5
1.1.1	ADAS	5
1.1.2	Enhjuling som testobjekt	5
1.2	Projektets mål	6
1.2.1	Delmål för projektet	6
1.2.2	Syfte för denna rapport	6
1.2.3	Avgränsningar	6
1.3	Samhälleliga och etiska aspekter	7
<b>2</b>	<b>Metod</b>	<b>8</b>
2.1	Simulering	8
2.2	Sensorer, filter och reglersystem	9
2.3	Elektronik	9
2.3.1	Kretskort	9
2.4	Prototyp	10
2.4.1	Pitch	10
2.4.2	Roll	10
2.4.3	Snabb prototypframtagning och iterativ konstruktion	10
2.4.4	Arbetslokaler/verkstäder	11
2.4.5	Materialval och tillverknings teknik	11
2.5	Tester	11
2.5.1	Datainsamling under test	11
2.5.2	Prototyp test	11
2.5.3	Säkerhet	11
<b>3</b>	<b>Modellering och Simulation</b>	<b>13</b>
3.1	Fysikalisk modellering	13
3.1.1	Beteckningar	13
3.1.2	Matematisk modellering för pitch	15
3.1.3	Matematisk modellering för roll	15
3.1.4	Matematisk modellering för rotation i yaw	16
3.1.5	Linjärisering i pitch-led	17
3.1.6	Linjärisering i roll-led	17
3.2	Simuleringsapplikation	18
<b>4</b>	<b>Reglersystem och filter</b>	<b>19</b>
4.1	Reglering	19
4.1.1	Optimal kontroll-reglering i pitch-led	19
4.1.2	Kaskadreglering i roll-led	21
4.1.3	Kaskadreglering i pitch-led	22
4.2	Kalmanfilter	22
4.2.1	Pitch	23
4.2.2	Roll	24
4.3	Komplementfilter	24

<b>5</b>	<b>Konstruktion</b>	<b>26</b>
5.1	Mekanik . . . . .	26
5.1.1	Prototyp 1 . . . . .	26
5.1.2	Prototyp 2 . . . . .	27
5.1.3	Prototyp 3 . . . . .	27
5.2	Elektronik . . . . .	29
5.2.1	Kompletterande komponenter . . . . .	31
5.3	Mjukvara . . . . .	31
5.3.1	Uppgiftskö och schemaläggning . . . . .	31
5.3.2	Kommunikation . . . . .	33
5.3.3	Styrsystem . . . . .	34
<b>6</b>	<b>Tester</b>	<b>35</b>
6.1	Pitch . . . . .	35
6.1.1	Simulering . . . . .	35
6.1.2	Prototyp . . . . .	36
6.2	Roll . . . . .	36
6.2.1	Simulering . . . . .	36
6.2.2	Prototyp . . . . .	37
6.3	Pitch och Roll . . . . .	38
<b>7</b>	<b>Resultat</b>	<b>39</b>
7.1	Hårdvara . . . . .	39
7.2	Balans i stationärt tillstånd . . . . .	39
7.3	Reglersystem . . . . .	39
7.3.1	Roll . . . . .	40
7.3.2	Pitch . . . . .	43
7.4	Filtrering och tillståndsuppskattning . . . . .	44
7.5	Styrbarhet med fjärrkontroll . . . . .	44
<b>8</b>	<b>Slutsats och Diskussion</b>	<b>45</b>
8.1	Konstruktion . . . . .	45
8.2	Filter . . . . .	45
8.3	Reglersystem . . . . .	45
8.4	Användning i testmiljö . . . . .	46
8.5	Rekommenderad uppföljning . . . . .	46
8.5.1	Uppföljningar för att få en manövrerbar enhjuling . . . . .	46
8.5.2	Andra rekommenderade uppföljningar . . . . .	47
<b>9</b>	<b>Tack</b>	<b>49</b>
<b>A</b>	<b>Komponentlista</b>	<b>52</b>
<b>B</b>	<b>Källkod</b>	<b>55</b>
B.1	Simulering . . . . .	55
B.2	STM32 . . . . .	55
<b>C</b>	<b>Detaljrutningar för prototyper</b>	<b>56</b>
C.1	Prototyp 1 . . . . .	56
C.2	Prototyp 2 . . . . .	57
C.3	Prototyp 3 . . . . .	62

## 0.1 Ordlista

Tabell 1: Ordlista

ADAS	Advanced Driver Assisting System (Förarassistanssystem)
Pitch	Lutningen på enhjulingen sett vinkelrätt med färdriktningen
Roll	Vinkeln som enhjulingen lutar sett från färdriktningen
Yaw	Vinkeln mot färdriktningen sett från ovan enhjulingen
VESC	Vedder Electronic Speed Controller (motordrivare)
FSESC	Flipsky Electric Speed Controller (motordrivare)
Python, C	Programmeringsspråk
Gyrometer	Sensor för avläsning av rotation
Accelerometer	Sensor för avläsning av acceleration
PCB	Printed Circuit Board (kretskort)
STM32	Mikrokontroller
VRU	Vulnerable Road User (Sårbar trafikant)
UART, I2C, SPI	Kommunikationsprotokoll mellan olika komponenter, mikrokontroller och datorer
SWD	Serial Wire Debug
(Mjukvaru-) Interrupt	Funktion som körs när en extern händelse, till exempel att data mottages på en port, händer en mikrokontroller, varpå normal funktion efteråt återupptas.
Race condition	Två eller fler processer som kör samtidigt försöker på samma gång utan synkronisering redigera samma objekt, vilket resulterar i oförutsägbara resultat.
GPIO	General purpose input output, portar på en mikrokontroller som fritt kan kontrolleras och läsas från i mjukvara utan att följa något specifikt protokoll.

# Kapitel 1

## Introduktion

I takt med att bilar de senaste åren har fått mer kapabla och kompetenta säkerhetssystem, så kallat ADAS (Advanced Driver Assisting System) [1] [2], har det uppkommit ett behov av att kunna testa dessa system på ett repeterbart och verklighetstroget sätt. För att förstå den verkliga prestandan på systemen som biltillverkare utvecklar är det viktigt att det finns ett brett spektrum av testscenarion, vilket ställer krav på att det måste finnas utrustning för att iscensätta dessa scenarion. I dagsläget testas bilar förmåga att upptäcka och avvärja kollision med cyklister. Testutrustningen som används är en cykel-atrapp med en människoliknande docka monterad på en platta. Plattan drivs av ett rep som rullas in [3].

I dagsläget pågår ett forskningsprojekt på Chalmers med att ta fram en cykel med förmåga att följa en bana och bära en människoliknande docka [4]. Det är med liknande anledning som detta projekt är aktuellt. Genom att testa hur bilar ADAS reagerar på flera typer av transportmedel, kommer framtidens ADAS bli säkrare.

Detta projekt ska utforska hur en enhjuling kan balansera och manövrera för att användas som testobjekt för att testa ADAS. För att balansera och följa en rak linje krävs reglering av två dimensioner, pitch-led (längsled) och roll-led (sidled). För att manövrera och följa andra banor än raka linjer krävs reglering i tre dimensioner, pitch, roll och yaw. Utmaningen i detta arbete ligger i att konstruktionen kräver en ständig reglering av alla plan för att upprätthålla balansen och kunna navigera. Att manövrera och följa en bana innebär ytterligare svårigheter eftersom flera motorer behöver samspela för att hålla en aktiv avvikning från det stabila tillståndet.

För det endimensionella fallet, balansera i pitch, finns det diverse liknande arbeten. En rapport från Harvard [5] beskriver hur en linjär kvadratisk regulator kombinerat med ett kalmanfilter ger ett mycket gott resultat. För balans i flera dimensioner finns det demonstrationsvideor online [6], men inte rapporter som beskriver problemet mer djupgående.

### 1.1 Bakgrund

ADAS är ett nödvändigt system för säkerheten i dagens bilar, men det är något som är under ständig utveckling för att bli mer och mer robust. I denna del kommer ADAS betydelse att förklaras samt varför en enhjuling valdes som ett testobjekt.

#### 1.1.1 ADAS

De flesta bilar numera är i någon utsträckning utrustade med ADAS [1]. ADAS innefattar bland annat automatisk nödbroms, adaptiv farthållare och dödvinkeln-varning [2]. Kraven från organisationen Euro NCAP, som testar och klassificerar alla bilar som produceras på den europeiska marknaden, blir högre och högre. Tidigare har mycket fokus legat på säkerheten för föraren. Numera har det utökats med fokus på skydd för VRU:er (vulnerable road user) [7]. I begreppet VRU ingår alla som rör sig kring vägen och kan bli skadad av en bil, exempelvis fotgängare, cyklister och enhjulingscyklister. Det vill säga, bilar ADAS ska bland annat minimera risken för kollision med VRU.

#### 1.1.2 Enhjuling som testobjekt

Enhjuling har varit tillgängligt att köpa, både i butik och online, i många år. I dagsläget har enhjulingen utvecklats och blivit mer populärt. Därav är det viktigt för moderna ADAS att kunna upptäcka och undvika kollision med enhjulingen. Att använda en enhjuling som testobjekt kan verka långsökt, men

detta anses vara relevant för att nå den nollvision som transportstyrelsen strävar efter [8]. För att nå nollvisionen krävs det att alla tänkbara transportmedel finns representerade i arsenalen av testobjekt.

Förhoppningen är att ett detektionssystem som kan känna igen både cyklar och enhjulingar också kommer känna igen andra fordon som liknar dessa, exempelvis mono-wheels, segways, hoverboards. Genom att ADAS känner igen enhjulingar kan dessa andra objekt kännas igen bättre.

## 1.2 Projektets mål

Projektet i stort är att enhjulingen ska kunna användas inom tester för ADAS, för vilket en flexibel och repeterbar vägföljning behövs, likt de cyklar som används i nuläget. Repeterbarhet behövs för att kunna utföra samma test många gånger, vilket är ett krav för säkerhetställa att ADAS alltid kan undvika trafikanter. För att kunna utföra mer utförliga tester behövs ett flexibelt system med möjlighet att programmera vägar att följa, samt markera hastigheter vid olika punkter.

### 1.2.1 Delmål för projektet

För att uppnå detta övergripande mål delas arbetet för projektet upp i flera mindre delmål. I denna rapport utförs en del av dessa delmål, i framtida arbeten baserade på detta bör resterande mål utredas. Delmålen för detta projekt är

- Balansera en axel — roll-led
- Balansera två axlar — roll-led och pitch-led
- Styra tre axlar — roll-, pitch- och yaw-led
- Styra enhjulingen med fjärrkontroll
- Följa en förprogrammerad rutt

### 1.2.2 Syfte för denna rapport

I denna rapport uppnås de två första delmålen för projektet — att först balansera endast i roll-led för att sedan balansera i pitch-led. Dessutom görs ett enkelt test för att kontrollera pitch-axelns hastighet med en fjärrkontroll.

Denna rapport försöker svara på ett antal frågor:

- Hur ska en enhjuling konstrueras för att kunna användas som testobjekt?
- Hur ska reglersystemet konstrueras och dimensioneras för att balansera en enhjuling?
- Vilka sensorer och komponenter krävs för reglering?
- Kan en autonom enhjuling styras likt en hur människa cyklar enhjuling?

### 1.2.3 Avgränsningar

Projektet kommer att avgränsas till att försöka skapa en fungerande prototyp. Enhjulingen kommer åka på plan mark där hjulen inte sladdar, och inte heller några plötsliga laststörningar uppkommer. Inget fokus kommer att läggas på att ta fram det bästa dellösningarna för varje problem, detta kan ses mer som en validering av koncept. För att kunna optimera dellösningarna kommer det behövas mer tid än vad projektet erbjuder. Detta arbete kommer dessutom inte ta hänsyn till den estetiska designen av enhjulingen. Arbetet kommer inte heller jobba efter några krav på hastighet eller krav på att bära en viss last.

## 1.3 Samhälleliga och etiska aspekter

Den huvudsakliga samhälleliga påverkan hos detta projekt kommer ifrån hur självkörande bilar och andra fordon tränas på att undvika kollision med denna enhjuling. Ett väl fungerande system där enhjulingen kör på ett realistiskt sätt, fordonen lär sig från detta och en stor arsenal av andra testfordon också utvärderas leder detta till ett säkrare samhälle där fler krockar mellan fordon och VRU kan undvikas.

Problem kan uppstå om någon del av detta system ej fungerar väl. Om enhjulingen ej emulerar en människa kan säkerhetssystemet i bilen lära sig "fel" rörelsemönster och bli förvirrat, om detta generaliseras till andra icke motorfordon kan risken ökas för cyklister och fotgängare.

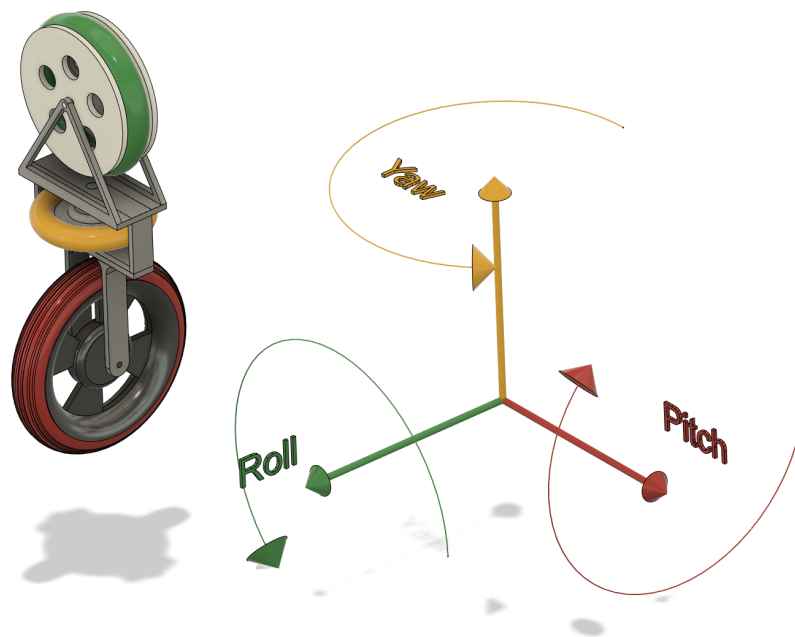
Risken finns också att igenkänningsystemet i bilen inte generaliserar bra, alltså att rörelsemönstrena den lär sig från enhjulingen inte förs över och appliceras hos andra fordon, till exempel cyklar. Att anpassa systemet för mycket på enhjulingar kan då få oproportionellt mycket av systemets "fokus", som mer effektivt kunde läggas på vanligare trafikfaror.

En risk som existerar för alla autonoma fordon är att den krockar och skadar personer i närheten. På grund av detta är det kritisk att ett nödstopp och en hastighetsbegränsning är implementerat inför tester. Då fordonet uteslutande kommer att användas i test sammanhang bedöms det att inga yttligare åtgärder behövs.

# Kapitel 2

## Metod

I detta kapitel beskrivs översiktligt metoden som har använts för att konstruera prototypen. Figur (2.1) illustrerar de olika rotationsaxlarna samt en övergripande bild på enhjulingen.



Figur 2.1: Illustration av de olika axlarna som enhjulingen ska styras i

### 2.1 Simulering

För att simulera hur enhjulingen beter sig vid olika reglersystem och systemparametrar skrivs ett Python-program som simulerar det fysikaliska systemet som tas fram. Simuleringen har både ett grafiskt gränssnitt för att visuellt se hur modellen beter sig och spela upp data från fysiska tester, samt ett programmeringsgränssnitt för att utföra automatiserade tester. Simuleringen tillåter test av reglersystem i kombination med olika fysikaliska parametrar utifrån den aktuella konstruktionen. Simuleringen bortser från friktion mellan de olika hjulaxlarna, samt antar oändlig friktion mot marken. Detta är i linje med avgränsningarna som presenterades i avsnitt 1.2.3 men innebär att simuleringen inte är helt och hållet verklighetstrogen.

Simuleringen samt den fysikaliska modellen som simuleras beskrivs mer ingående i kapitel 3.

## 2.2 Sensorer, filter och reglersystem

I detta projekt används följande sensorer:

- Kombinerad gyrometer och accelerometer (MPU6050)
- Varvtalsutläsning från motorkontrollern VESC (Vedder Electronic Speed Controller)

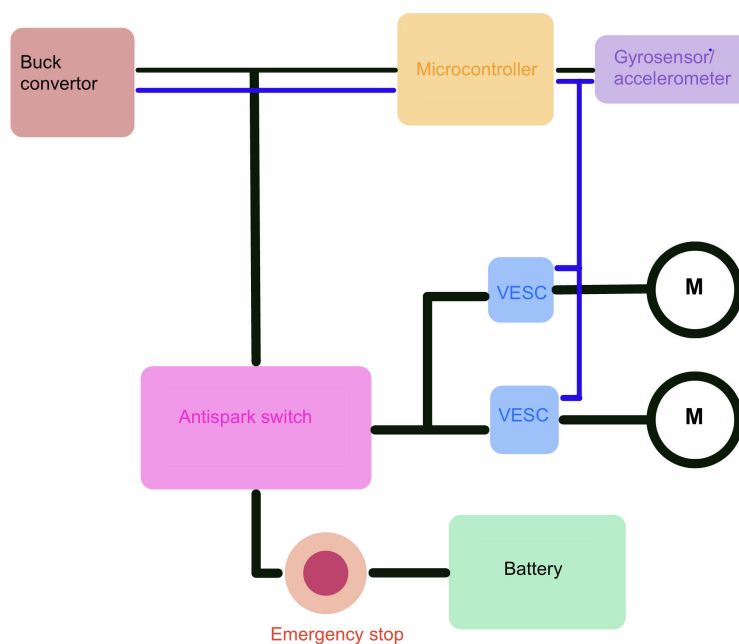
Den kombinerade accelero- och gyrometern ger acceleration och vinkelhastighet i tre axlar  $x$ ,  $y$  och  $z$ . Den är monterad på prototypen och i stående läge motsvarar dessa axlar pitch, roll och yaw. Gyrometrar är kända för hög känslighet, men att mätvärdet förskjuts med tiden eftersom integration krävs för att uppskatta vinkeln [9]. En accelerometer däremot är väldigt stabil med tiden.

För att filtrera bort brus och uppskatta tillstånd från det sensorerna mäter, testas både ett kalmanfilter och komplementfilter. Dessa beskrivs i detalj under kapitel 4.

För att balansera och styra enhjulingen krävs ett reglersystem som utifrån de filtrerade mätvärdena ger ut signaler till motorerna i form av vridmoment vilket applicerar rätt krafter på enhjulingen för att hålla den i balans. Två typer av reglersystem prövas för enhjulingen, en regulator baserad på optimal kontroll vilken beskrivs i avsnitt 4.1.1 samt en kaskadregulator som beskrivs i avsnitt 4.1.3 och 4.1.2.

## 2.3 Elektronik

Figur (2.2) visar ett blockdiagram där de elektriska komponenterna som är tänkt att användas i projektet ingår. Funktioner av delarna som ingår i diagrammet beskrivs i avsnitt 5.2.



Figur 2.2: Blockdiagram av elektriska komponenter

### 2.3.1 Kretskort

Initialt användes ett utvecklingskort med moduler för att testa att alla komponenter (mikrokontroller, sensorer och motordrivare) fungerade med varandra. När funktion var verifierad utvecklades ett kretskort som kombinerade alla komponenter. Fördelen med detta var att elsystemet blev mycket mer kompakt och minimerade risken för glapp i kontakter från lösa kablar, ett kompaktare system kan monteras på en säkrare plats för bättre skydd vid eventuella krascher. Kretskortet kan också användas av framtida studenter som kommer att fortsätta med projektet.

## 2.4 Prototyp

Vid projektets start fanns ett cykelhjul med inbyggd navmotor till godo enligt figur (2.3), detta hjulet används som grund för prototypen och har som syfte att balansera i pitch. Runt detta hjul byggs en ram som ska inrymma elektronik samt agera grund för hjulet som balanserar roll.



Figur 2.3: Pitch-hjul 20 tum i diameter

### 2.4.1 Pitch

Balanseringen i pitch fungerar på många sätt likt en inverterad pendel, för detta krävs att det ska finnas en viss massa ovanför hjulet för pitch. I en typisk inverterad pendel låses den stationära punkten till marken eller i stationär plattform av något slag. Men om, som i fallet för en enhjuling, den stationära punkten inte är fast, introduceras en extra frihetsgrad i systemet. När den inverterade pendeln får laststörningar kommer den stationära punkten att flyttas. Tanken är då att pendeln ska svara på detta genom att korrigera sin position likt hur den stationära punkten flyttades. Till exempel om pendeln börjar luta höger kommer enhjulingen köra åt höger för att upprätthålla balans. Genom att kontinuerligt låta den inverterade pendeln svara på laststörningarna den blir utsatt för, går det att uppnå stabilitet. Detta beskrivs mer i avsnitt 3.1.2.

### 2.4.2 Roll

För att balansera i roll används ett reaktionshjul. Reaktionshjulet skapar ett moment genom att snabbt rotera i motsatt riktning som momentet ska generas. Detta beskrivs mer utförligt i avsnitt 3.1.3, figur (2.4) visar bilden på roll-hjulet.



Figur 2.4: Roll-hjulet 13 tum i diameter

### 2.4.3 Snabb prototypframtagning och iterativ konstruktion

Arbetet i detta projekt jobbar efter filosofin snabb prototypframtagning, vilket siktar på att få en fungerande prototyp för varje delmål i arbetet så tidigt som möjligt. Att skapa en lösning för varje delproblem

utan att fastna i djupare analyser visar tidigt på potentiella problem med metoden, och kan ge en praktisk förståelse av målet tidigt. Vilket i sin tur bidrar till bättre förståelse för kommande steg. Därmed måste prototypen vara tillräckligt tålig för att klara arbetsbelastningar och kunna skydda slagkänsliga komponenter, vid fall.

Konceptet snabb prototypframtagning innebär att konstruktion kommer iterativt förbättras allt eftersom testerna visar på svagheter eller styrkor med konstruktionen.

#### 2.4.4 Arbetslokaler/verkstäder

Förverkligandet och byggnationen av prototyperna tog plats på Prototyplabbet och på E-sektionens Teletekniska Avdelning (ETA). Prototyplabbet är en lokal i maskinhuset där det finns möjlighet till konstruktion i metall och trä. I prototyplabbet finns det även möjlighet att få vägledning och råd av forskningsingenjörerna som arbetar där. Där tillverkades den mekaniska konstruktionen till varje prototyp. ETA är en fristående elektrisk verkstad som är en del av Chalmers studentkår. På ETA finns möjlighet till additiv tillverkning, kretskortstillverkning och lödning.

#### 2.4.5 Materialval och tillverkningsteknik

Materialen som användes för den mekaniska prototypen är plåtar av aluminium och stål med tjocklekar 2 mm och 3 mm. Aluminium användes i de fallen där låg massa var fördelaktig och stål där det ansågs fördelaktigt med hög massa. Tillverkningsteknikerna som användes var bandsågning, plåtklippning, borrar, plåtbockning och vattenskarvning.

### 2.5 Tester

Testerna kan delas upp i två stora segment: test på prototyp och test på reglersystemet. Syfte med testerna är att samla in information och använda dem för vidare utveckling och förbättring.

#### 2.5.1 Datainsamling under test

För varje tidsteg loggas alla relevanta datavärden. Detta sker i realtid. Dessa datavärden skickas över USB till en dator som antingen kan spara värdena till en fil för att processeras senare, eller kopplas direkt till simulatorns grafiska gränssnitt för visualisering.

Datainsamlingen används för många olika syften, delvis för att kalibrera sensorer och regulatorparametrar, samt för att utvärdera reglersystemets effektivitet kvantitativt. Loggdatan används också för att identifiera och diagnostisera buggar och problem i mjukvaran som driver reglersystemet och kommunikation med sensorer och motordrivare.

#### 2.5.2 Prototyptest

En mer kvalitativ typ av test kommer utföras på prototypen när reglersystemet är stabilt nog för att stå upprätt under längre perioder av tid. Här avses att svara på arbetets frågeställningar i projektets syfte. Ifall reglersystemet fungerar dåligt har detta test också som syfte att identifiera vad i prototypen som behöver ändras och förbättras, eller om en ny iteration av prototyp behöver byggas.

#### 2.5.3 Säkerhet

Eftersom prototypen är konstruerad i metall, har vassa kanter och väger cirka 20 kg behövs vissa säkerhetsåtgärder vidtas för att inte riskera skada för personer och saker i enhjulingens närhet. Vid samtliga test fästes ett rep mellan toppen av prototypen och en stabil punkt ovanför, till exempel en takbalk, i syfte att säkerställa att den inte åker in i något eller slår i marken när den faller. Utöver detta monteras en mjuk "krockudde" kring toppen av prototypen för att göra eventuella kollisioner säkrare, samt skydda den monterade elektroniken mot att slå i marken.

I roll-led, där ett tungt reaktionshjul accelereras upp till 600RPM, konstruerades en liten bur runt hjulet. Detta för att skydda både mot att oavsiktligt röra hjulet när det snurrar, samt agera skydd ifall hjulet skulle gå sönder eller lossna från ramen. Buren skyddar även hjulet från kollision mot marken vid fall.

I den elektroniska delen används inga nominella spänningar över 60 V DC. I Sverige klassas spänningar under 120 V DC som "klenspänning" [10], så inga särskilda säkerhetsåtgärder tilltas för att skydda

människor från elsystemet. Det finns en risk att motorerna genererar elektromotorisk spänning som kan klassas som farlig. Alla kablar som är kopplade till eller från motorn är helt isolerade med plasthölje, och kopplas endast till motordrivarna som är monterade på chassit.

Batteriet som användes för att strömföra enhjulingen under testerna är ej monterat på enhjulingen. Litiumjonbatterier har risk att explodera vid skada, vilket bör undvikas då enhjulingen lätt ramlar. Genom att inte montera batteriet på enhjulingen finns en risk att batterikabeln skulle dras ut om kabeln skulle sträckas ut för hastigt. XT-60- samt XT-90-kopplingsdon användes för att koppla batteriet till enhjulingen, monterat på ett sådant sätt att kopplingsdonet blir den svagaste punkten då kabeln sträcks ut. På detta sättet kopplar batteriet på ett säkert sätt ur sig ur enhjulingen istället för att de strömförande kablarna skulle skadas. För att undvika gnistor vid inkoppling användes ett XT-90-kopplingsdon designat för att ha hög initial resistans vid inkoppling.

Vidare lades stor vikt att testa på säkra områden där alla närvarande var involverade och uppmärksamma. Under vissa tester fanns en del publik, då sågs det till att endast gruppmedlemmar fick vistas inom en viss radie av enhjulingen och att alla var informerade om hur farlig enhjulingen kunde vara om den skulle ramla på en.

Ett nödstopp i form av ett "dödmansgrepp" används för att motorerna ska drivas. Nödstoppknappen på systemet är designad för att om någonting oförväntat skulle hända så ska kopplingen brytas och enhjulingens motorer ska sluta drivas. Dödmansgreppet behöver aktivt tryckas ner av en människa för att motorerna ska kunna strömföras. Detta är för att om människan ansvarig för nödstoppet skulle skadas eller på något annat sätt förlora kontrollen så ska enhjulingen slås av. För att förenkla designen implementeras nödstoppet i mjukvara, där funktionen ansvarig för att strömsätta motordrivaren alltid kollar om knappen är aktiverad, annars nollställs strömmen. Detta innebär att stoppet slutar fungera om mikrokontrollern skulle låsa sig, vilket kan ses som en säkerhetsrisk då det även borde gå att stoppa motorerna om mikrokontrollern slutar funka. Dock verkar motorkontrollern VESC märka om mikrokontrollern slutar skicka begäran om ström och stänger av motorerna kort efter.

# Kapitel 3

## Modellering och Simulation

För att effektivt kunna utveckla styrsystemet för enhjulingen krävs en förståelse för hur det fysikaliska systemet beter sig.

En fysikalisk modell är användbar för att analytiskt kunna utveckla regleringsmodeller, men för att praktiskt kunna arbeta med modellen utvecklas en simulering där den fysikaliska modellen på en dator kan beräknas och testas i olika aspekter.

### 3.1 Fysikalisk modellering

Den fysikaliska modellen utvecklas separat i pitch-led och roll-led. Då regleringens mål är att låta enhjulingen stå upprätt kommer analysen utgå från att enhjulingen nominellt står upp. Mer specifikt, i pitch-led kommer modellerna utgå från att enhjulingen står vinkelrätt med marken i både roll- och pitch-led. Detta antagande förenklar analysen genom att låta axlarna analyseras helt oberoende från varandra. Detta innebär också att styrsystemen för pitch och roll kan vara helt frikopplade från varandra.

Att alltid sikta på stående läge för enhjulingen innebär ett reglerbart system som kan köra framåt och bakåt, men det begränsar vilka avancerade manövrar som kan göras. Till exempel, att köra framåt i pitch-led samtidigt som roll-led balanserar åt sidan kan låta enhjulingen svänga i sin bana, men detta kräver en koppling i pitch- och roll-led denna modell inte reflekterar.

En modell i yaw-led tas även fram. Denna modell används inte i projektet på grund av tidsbrist, men kan finnas användbar för framtida arbeten. Vidare diskussion om kontroll i yaw-led sker i avsnitt 8.5.1.

#### 3.1.1 Beteckningar

I denna rapport används följande beteckningar för systemparametrar. Subskriptet  $p$  betecknar pitch,  $r$  betecknar roll och  $y$  betecknar yaw. Vinkel av hjul betecknas med  $\phi$ , vinkel av enhjulingen med  $\theta$ . För andra parametrar används subskriptet  $h$  för parametrar för hjul.

Beteckningarna som används för systemets parametrar såsom vikter, tröghetsmoment etc. presenteras i tabell (3.1). Systemets läge i pitch- respektive roll-led presenteras i tabell (3.2) respektive tabell (3.3).

Tabell 3.1: Beteckningar för systemets parametrar

Beteckning	Enhet	Beskrivning
$I_{tr}$	[kgm <sup>2</sup> ]	Tröghetsmoment för hela enhjulingen runt punkten den rör marken, i roll-led
$I_{cp}$	[kgm <sup>2</sup> ]	Tröghetsmoment för chassi runt punkten enhjulingen rör marken i pitch-led
$I_{hp}$	[kgm <sup>2</sup> ]	Tröghetsmoment för drivhjulet
$I_{hr}$	[kgm <sup>2</sup> ]	Tröghetsmoment för reaktionshjulet i roll
$m_t$	[kg]	Total massa
$m_c$	[kg]	Massa för chassi
$m_{hp}$	[kg]	Massa för drivhjulet i pitch
$m_{hr}$	[kg]	Massa för reaktionshjulet för roll
$m_{hy}$	[kg]	Massa för reaktionshjulet för yaw
$r_{hp}$	[m]	Radie för drivhjulet i pitch
$r_{hr}$	[m]	Radie för reaktionshjulet i roll
$r_{hy}$	[m]	Radie för reaktionshjulet i yaw
$l_c$	[m]	Avstånd från enhjulingens masscentrum till punkten enhjulingen rör marken
$l_{hp}$	[m]	Avstånd från drivhjulets centrum till punkten enhjulingen rör marken
$l_{hr}$	[m]	Avstånd från reaktionshjulet i roll centrum till punkten enhjulingen rör marken

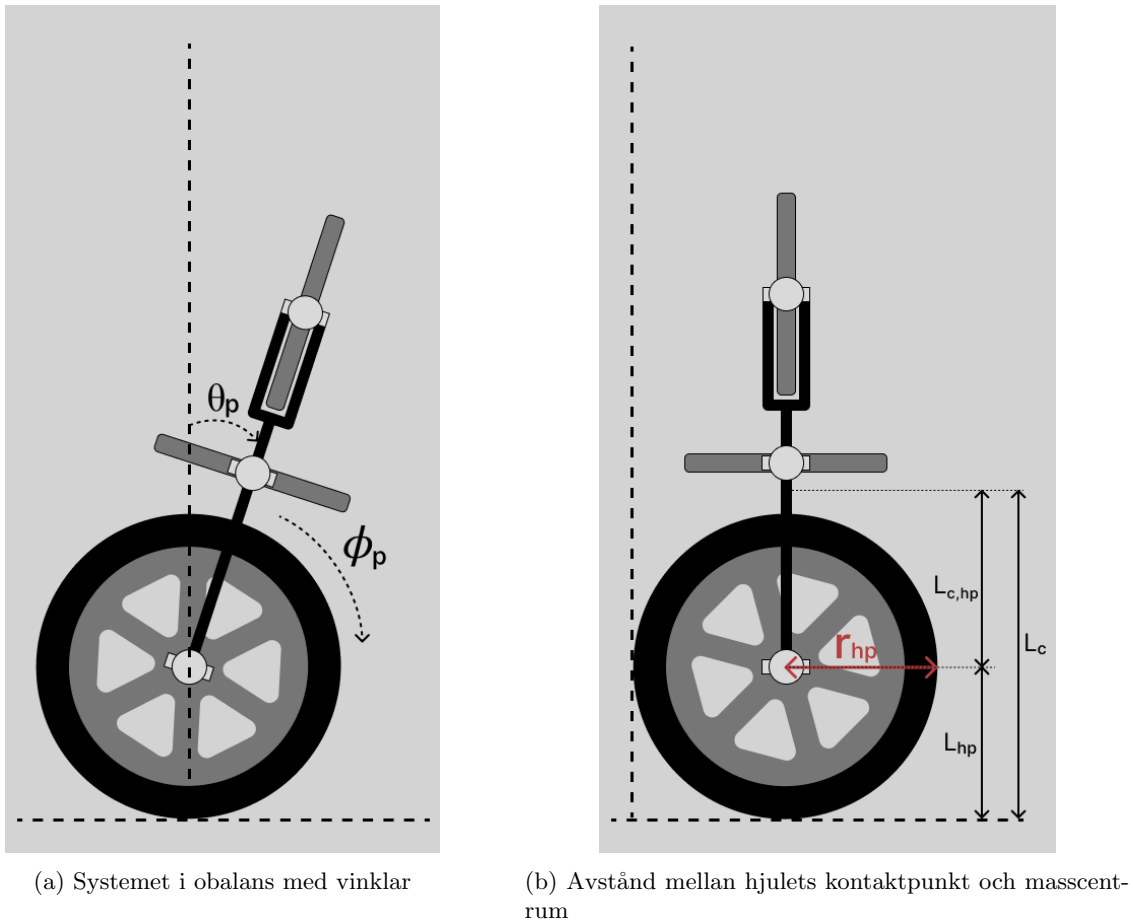
Tabell 3.2: Beteckning för systemets läge i pitch-led

Beteckning	Enhet	Beskrivning
$\theta_p$	[rad]	Pitch-vinkel för chassi relativt $\hat{z}$
$\phi_p$	[rad]	Pitch-vinkel för hjulet
$x$	[m]	Position av drivhjulscentrum
$\tau_p$	[N m]	Vridmoment av drivhjulet

Tabell 3.3: Beteckning för systemets läge i roll-led

Beteckning	Enhet	Beskrivning
$\theta_r$	[rad]	Roll-vinkel för chassi
$\phi_r$	[rad]	Vinkel av reaktionshjulet i roll
$\tau_r$	[N m]	Vridmoment av reaktionshjulet i roll
$\tau_{gr}$	[N m]	Vridmoment från gravitation i roll

### 3.1.2 Matematisk modellering för pitch



Figur 3.1: Systemparametrar och lägesbeteckningar för systemet i pitch-led

Systemet i pitch-led beskrivs av tillståndsvektorn  $q = [\phi_p, \dot{\phi}_p, \theta_p, \dot{\theta}_p]$ . Under antagandet oändlig friktion med plan mark är enhjulingens position  $x = r_{hp}\phi_p$ . Därav beskriver även vektorn  $[x, \dot{x}, \theta_p, \dot{\theta}_p]$  systemet. Då dessa endast skiljer i en konstantfaktor används båda dessa utbytbart.

Systemets lagrange-ekvationer löses likt [5] och ger följande:

$$\begin{cases} (I_{hp} + (m_{hp} + m_c)r_{hp}^2)\ddot{\phi}_p + m_c r_{ph} l_{c, hp} \cos \theta_p \ddot{\theta}_p - m_c r_{hp} l_{c, hp} \sin \theta_p \dot{\theta}_p^2 = -\tau_p \\ m_c r_{hp} l_{c, hp} \cos \theta_p \ddot{\phi}_p + (I_{cp} + m_c l_{c, hp}^2)\ddot{\theta}_p - m_c g l_{c, hp} \sin \theta_p = \tau_p \end{cases} \quad (3.1)$$

där  $l_{c, hp} = l_c - l_{hp}$  är avståndet mellan chassits masscentrum och drivhjulets centrum.

För att analysera systemet löses  $\dot{q}$  ut som en funktion av  $q$  samt  $\tau$ . Ekvation (3.1) skrivs om som en matrisekvation.

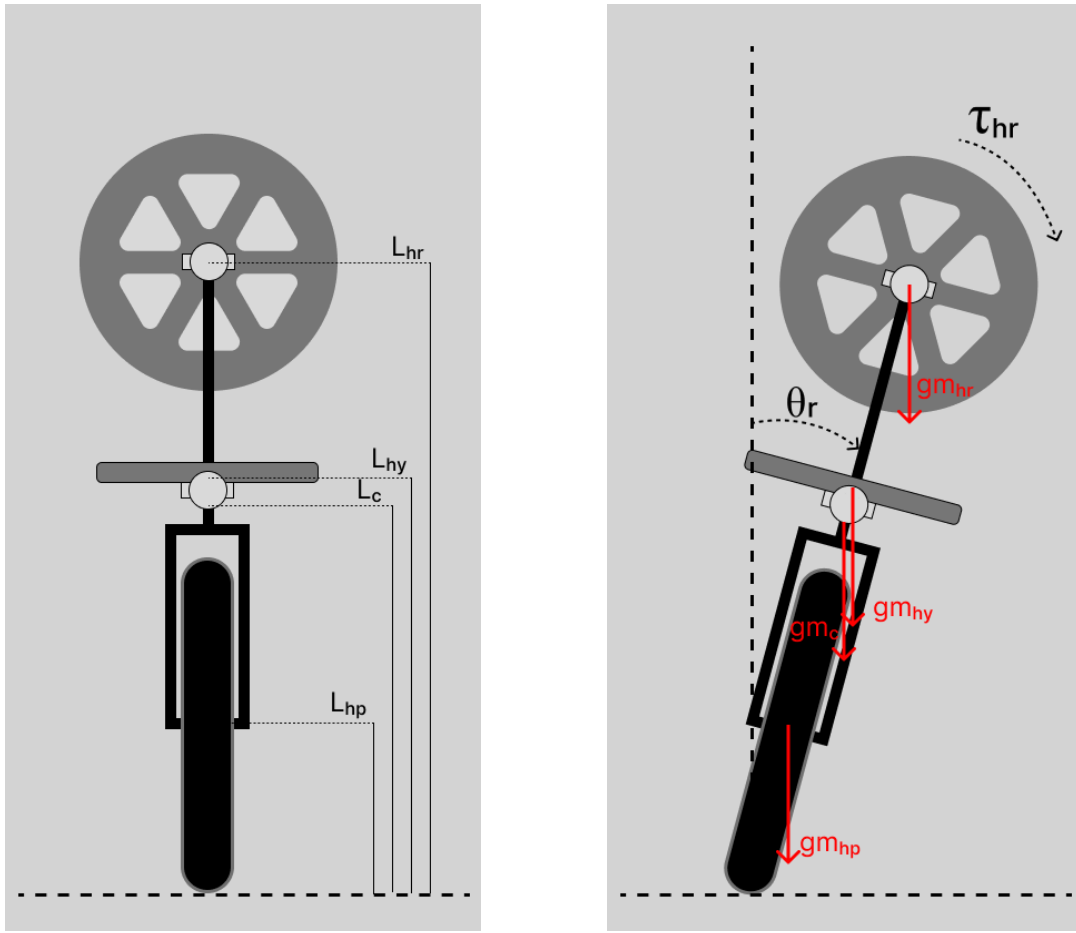
$$\begin{bmatrix} I_{hp} + (m_{hp} + m_c)r_{hp}^2 & m_c r_{ph} l_{c, hp} \cos \theta_p \\ m_c r_{hp} l_{c, hp} \cos \theta_p & I_{cp} + m_c l_{c, hp}^2 \end{bmatrix} \begin{bmatrix} \ddot{\phi}_p \\ \ddot{\theta}_p \end{bmatrix} = \begin{bmatrix} -\tau_p + m_c r_{hp} l_{c, hp} \sin \theta_p \dot{\theta}_p^2 \\ \tau_p + m_c g l_{c, hp} \sin \theta_p \end{bmatrix}$$

$\ddot{\phi}_p$  och  $\ddot{\theta}_p$  löses ut vilket ger systemmodellen:

$$\begin{bmatrix} \ddot{\phi}_p \\ \ddot{\theta}_p \end{bmatrix} = \begin{bmatrix} I_{hp} + (m_{hp} + m_c)r_{hp}^2 & m_c r_{ph} l_{c, hp} \cos \theta_p \\ m_c r_{hp} l_{c, hp} \cos \theta_p & I_{cp} + m_c l_{c, hp}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\tau_p + m_c r_{hp} l_{c, hp} \sin \theta_p \dot{\theta}_p^2 \\ \tau_p + m_c g l_{c, hp} \sin \theta_p \end{bmatrix} \quad (3.2)$$

### 3.1.3 Matematisk modellering för roll

För att balansera i roll led används ett reaktionshjul placerat vinkelrätt mot drivhjulet. När reaktionshjulet roterar genereras ett vridmoment  $\tau_r$ . Av Newtons tredje lag om rotation genererar detta ett motriktat vridmoment av samma storlek på resten av enhjulingen.



(a) Avstånd från rotationspunkt till masscentrum (b) Krafter, vinklar och vridmoments riktningar

Figur 3.2: Krafter och längder för roll

$$\tau_r = \ddot{\phi}_r (m_{hr} r_{hr}^2) = I_{hr} \ddot{\phi}_r \quad (3.3)$$

Vridmomentet från gravitationen i roll beror både på vinkeln i roll och vinkeln i pitch.

$$\tau_{gr} = (m_{hp} l_{hp} + m_c l_c + m_{hy} l_{hy} + m_{hr} l_{hr}) g \sin \theta_r \cos \theta_p$$

För att förenkla detta uttryck för  $\tau_{gr}$  införs ett vridmoment  $\tau_{gr0}$  som gravitationens påverkan i roll-led när enhjulingen "ligger ner", alltså har  $\theta_r = \pi/2, \theta_p = 0$ :

$$\begin{aligned} \tau_{gr0} &= (m_{hp} l_{hp} + m_c l_c + m_{hy} l_{hy} + m_{hr} l_{hr}) g \\ \tau_{gr} &= \tau_{gr0} \sin \theta_r \cos \theta_p \end{aligned} \quad (3.4)$$

Enhjulingens vinkelacceleration i roll blir då.

$$\ddot{\theta}_r = \frac{\tau_{gr} - \tau_r}{I_{tr}} \quad (3.5)$$

Reaktionshjulets vinkelacceleration blir.

$$\ddot{\phi}_r = \frac{\tau_r}{I_{hr}} \quad (3.6)$$

### 3.1.4 Matematisk modellering för rotation i yaw

För att kunna manövrera lateralt behöver enhjulingen kunna rotera i yaw-led. För att uppnå detta används även här ett reaktionshjul vilket acceleration skapar vridmoment. Detta vridmoment behöver övervinna både konstruktionens tröghetsmoment och framför allt friktionen mellan hjulet och marken.

För att beräkna friktionskraften behövs friktionskoefficienten mellan gummihjulet och underlaget. Då tester kommer genomföras på asfalt och ideella förhållanden antas friktionskoefficienten ligga på mellan 0,7 och 0,9. Friktionsmomentet beräknas sedan genom att multiplicera med den längsta radien på arean av hjulet i kontakt med underlaget.

$$\tau_f = (m_{tot}g\mu)r_{hk} \quad (3.7)$$

$$(I_{toty} - I_{hy})\ddot{\alpha}_y + \tau_f = I_{hy}(\ddot{\alpha}_{hy}) \quad (3.8)$$

$$\ddot{\alpha}_y = \frac{I_{hy}(\ddot{\alpha}_{hy}) - \tau_f}{(I_{toty} - I_{hy})} \quad (3.9)$$

Beteckning	Enhet	Beskrivning
$I_{toty}$	[kgm <sup>2</sup> ]	Tröghetsmoment på konstruktionen i yaw
$I_{hy}$	[kgm <sup>2</sup> ]	Tröghetsmoment på det lodräta reaktionshjulet i yaw
$\ddot{\alpha}_y$	[rad/s <sup>2</sup> ]	Vinkel acceleration på hela konstruktionen i yaw
$\ddot{\alpha}_{hy}$	[rad/s <sup>2</sup> ]	Vinkel acceleration på det lodräta reaktionshjulet i yaw
$\tau_f$	[N m]	Vridmoment från friktionen mellan hjulet och underlaget
$m_{tot}$	[kg]	Massa för hela konstruktionen
$r_{hk}$	[m]	längsta radien på arean av drivhjulet i kontakt med underlaget
$\mu$	-	Friktionskoefficienten mellan drivhjulet och underlaget

### 3.1.5 Linjärisering i pitch-led

För att kunna applicera etablerade regleringsprinciper på systemet linjäriseras ekvationerna. Ett antagande är att enhjulingen i pitch-axeln endast kommer röra sig med små avvikelser från den stabila punkten. Den stabila punkten är när enhjulingen står upprätt,  $\theta_p = 0$ . Vilket möjliggör bland annat följande förenklingar:

$$\begin{aligned} \sin \theta_p &\approx \theta_p \\ \cos \theta_p &\approx 1 \end{aligned}$$

I pitch-led linjäriseras ekvation (3.2) vilket ger följande modell

$$\begin{bmatrix} \ddot{\phi}_d \\ \ddot{\theta}_d \end{bmatrix} = \begin{bmatrix} I_{hp} + (m_{hp} + m_c)r_{hp}^2 & m_c r_p l_{c, hp} \\ m_c r_{hp} l_{c, hp} & I_{cp} + m_c l_{c, hp}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\tau \\ \tau + m_c g l_{c, hp} \theta_p \end{bmatrix} \quad (3.10)$$

Alltså är  $\ddot{\phi}_p$  samt  $\ddot{\theta}_p$  linjära funktioner i  $\tau_p$  och  $\theta_p$ . Följande systemparametrar ansätts  $A, B, C, D$  så att

$$\begin{cases} \ddot{x} &= A\theta_p + B\tau_p \\ \ddot{\theta}_p &= C\theta_p + D\tau_p \end{cases} \quad (3.11)$$

och kan reglera utifrån denna modell.

### 3.1.6 Linjärisering i roll-led

På samma sätt linjäriseras rörelsekvationerna i roll-led, nu även under antagandet att enhjulingens vinkel i roll-led är liten,  $\theta_r \approx 0$ . Med antagandena

$$\sin \theta_r \approx \theta_r \cos \theta_r \approx 1$$

vilket möjliggör omskrivning av ekvation (3.5) som

$$\begin{aligned} \ddot{\theta}_r &= \frac{\tau_{gr} - \tau_r}{I_{tr}} \\ &= \frac{\tau_{gr0} \sin \theta_r \cos \theta_p - \tau_r}{I_{tr}} \\ &= \frac{\tau_{gr0} \theta_r - \tau_r}{I_{tr}} \end{aligned} \quad (3.12)$$

ekvation (3.6) förenklas ej av linjärisering då den endast beror på  $\tau_r$ .

## 3.2 Simuleringsapplikation

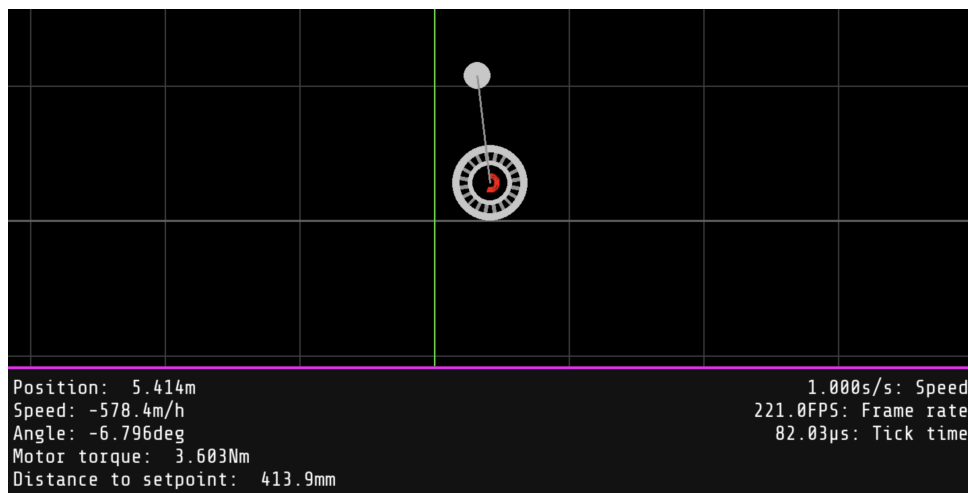
För att testa och utvärdera styrsystem och parametrar av systemets påverkan har ett simuleringsprogram utvecklats. Simulatorens dynamik i pitch- och roll-led separeras, och använder ekvationer (3.2, 3.4, 3.5 och 3.6) för att simulera systemet. Systemfunktionen integreras med hjälp av en runge-kutta-integrator av ordning 4, samt har logik för att hantera ifall enhjulingens vinkel blir för stor och den "slår i marken". Simulatorens har även två typer av simulerade sensorer på enhjulingen. En gyrometer i pitch-led implementeras genom att ge ut vinkelhastigheten  $\dot{\theta}$ . En accelerometer implementeras genom att lösa ut positionens acceleration och gravitation i tid, och transformera det till accelerometers koordinatsystem.

Ett enkelt användargränssnitt är kopplat till simuleringen, där systemet ritas upp på skärmen, intressanta parametrar skrivs ut i realtid, samt inmatning från användaren kan introducera laststörningar och ändra regulatorns referensvärde. Användargränssnittet använder Python-biblioteket Pygame [11] för att sköta grafik och interaktion med användaren. Till simulatorens kopplas ett reglersystem. Reglersystemet skrivs så att det enkelt ska gå att byta regulator utan större modifikation i koden. Detta möjliggörs genom att använda en överklass som respektive regulator är en underklass till. Vilket gör det enkelt att testa olika regulatorers prestanda.

För att göra simulatorens mer lik verkligheten finns det möjlighet att koppla ett kalmanfilter samt lägga på brus från de simulerade tillstånden. Detta för att försöka kalibrera filtrets parametrar.

Till simulatorens går det också att koppla styrsystemet och filtret som är skrivna i C för mikrokontrollern. Det möjliggör för verifiering av funktionen på det som är skrivet i C. För att låta Python kommunicera med den koden används Ctypes [12] som är Pythons egna FFI (foreign function interface).

Till simulatorens skrivs en resimulator modul. Modulen kan läsa in loggar från test och koppla värden från loggen till de olika simuleringsparametrarna, insamling av loggar beskrivs i avsnitt 2.5.1. Det gör att loggar kan spelas upp i samma visualiseringsmiljö som simulering för att i efterhand analysera resultatet. Resimulatorens består av en inläsnings modul som läser in textfilen som loggen sparas i. Inläsningsmodulen sparar datan för varje parameter i en dataframe. En dataframe är Python biblioteket pandas datastruktur för att lagra data. Resimulatorens är en underklass till simulatorens, det gör att resimulatorens enkelt ersätta simulatorens i visualiseringskoden utan större ändringar.



Figur 3.3: Skärmbild på det grafiska gränssnittet av simuleringsapplikationen under ett test i pitch-led

# Kapitel 4

## Reglersystem och filter

### 4.1 Reglering

För att reglera systemet i pitch- och roll-led kräver varje axel en regulator och ett sensorfilter. Filter används för att uppskatta vinklar och andra systemparametrar utifrån sensorerna som används, samt filtrera bort brus från sensorerna. Regulatorns syfte är att driva pitch- och rollmotorn för att hålla enhjulingen upprätt.

I arbetet utvecklades ett antal filter och regulatorer för varje axel, som sedan testades. För pitch-led utvecklades två regulatorer: en kaskad-PD-regulator beskriven i avsnitt 4.1.3 samt en optimal kontrollregulator beskriven i avsnitt 4.1.1. I roll-led utvecklades en liknande kaskad-PD-regulator som förklaras i avsnitt 4.1.2. Ett kalmanfilter och ett komplementfilter utvecklades i för både roll- och pitch-led som förklaras i avsnitt 4.2.1, 4.2.2, 4.3.

#### 4.1.1 Optimal kontroll-reglering i pitch-led

Optimal kontroll-reglering innebär att optimera styrsignalen  $\tau$  som en funktion av systemets läge  $\tau(q)$  under något optimeringskriterie [13]. Optimeringskriteriet är att ha ett visst börvärde för hastighet  $\dot{x}_{\text{ref}}$ , samt att chassit är upprätt och stillastående, alltså  $\theta_p = \dot{\theta}_p = 0$ .

Inom optimal kontroll löses denna styrsignal ut med hjälp av till exempel variationskalkyl för att hitta en globalt optimal signal för att uppfylla optimeringskriterierna. Detta är beräkningsmässigt intensivt och kan inte lösas analytiskt. För att förenkla problemet löses den optimala styrsignalen ut i varje tidssteg  $t$ , där signalen optimeras för att uppnå kriterierna efter en viss tid, som kallas stopp-tid.

Denna stopp-tid  $t_s$  är alltså en tid vald så att reglersystemet alltid optimerar för att optimeringskriterierna ska vara uppnådda vid tid  $t + t_s$ . En kort stopp-tid tvingar systemet att optimera för kraven väldigt hastigt, vilket kan leda till instabilitet och översvängningar. En lång stopp-tid ger på samma sätt ett långsammare system då det har mer tid på sig att uppnå optimeringskraven.

Alltså optimeras

$$\begin{cases} \dot{x}(t + t_s) = \dot{x}_{\text{ref}} \\ \theta_p(t + t_s) = 0 \\ \dot{\theta}_p(t + t_s) = 0 \end{cases} \quad (4.1)$$

Var optimeringskriterie optimeras separat till var sin styrsignal, och ansätter superpositionen av dessa styrsignaler som den slutgiltiga styrsignalen.

Utifrån ek. (3.11)

$$\begin{cases} \ddot{x} &= A\theta_p + B\tau_p \\ \ddot{\theta}_p &= C\theta_p + D\tau_p \end{cases}$$

ansätts  $\tau_p = -\frac{C}{D}\theta_p + \Delta\tau_p$ , samt  $E = A - \frac{CB}{D}$  vilket förenklar till

$$\begin{cases} \ddot{x} = A\theta_p - \frac{CB}{D}\theta_p + B\Delta\tau_p &= E\theta_p + B\Delta\tau_p \\ \ddot{\theta}_p = C\theta_p - \frac{CD}{D}\theta_p + D\Delta\tau_p &= D\Delta\tau_p \end{cases}$$

så att en term i högerledet elimineras för  $\ddot{\theta}_p$

För  $\dot{x}(t + t_s)$  taylorutvecklas till tredje graden ( $\dot{x}$  beror på  $\Delta\tau$  i tredje graden)

$$\begin{aligned}
\dot{x}(t+t_s) &= \dot{x}(t) + t_s \ddot{x}(t) + \frac{t_s^2}{2} \dddot{x}(t) + \frac{t_s^3}{6} \dots(t) \\
&= \dot{x}(t) + t_s(E\theta_p + B\Delta\tau_p) + \frac{t_s^2}{2}(E\dot{\theta}_p + B\Delta\dot{\tau}_p) + \frac{t_s^3}{6}(E\ddot{\theta}_p + B\Delta\ddot{\tau}_p) \\
&= \dot{x}(t) + t_s(E\theta_p + B\Delta\tau_p) + \frac{t_s^2}{2}(E\dot{\theta}_p + B\Delta\dot{\tau}_p) + \frac{t_s^3}{6}(ED\Delta\tau_p + B\Delta\ddot{\tau}_p)
\end{aligned}$$

Antar att  $\Delta\dot{\tau}_p = \Delta\ddot{\tau}_p = 0$

$$\dot{x}(t+t_s) = \dot{x}(t) + t_s(E\theta_p + B\Delta\tau_p) + \frac{t_s^2}{2}E\dot{\theta}_p + \frac{t_s^3}{6}ED\Delta\tau_p$$

Nu löses  $\Delta\tau_p$  då  $\dot{x}(t+t_s) = \dot{x}_{\text{ref}}$

$$\begin{aligned}
\dot{x}(t+t_s) = \dot{x}_{\text{ref}} &= \dot{x}(t) + t_s(E\theta_p + B\Delta\tau_p) + \frac{t_s^2}{2}E\dot{\theta}_p + \frac{t_s^3}{6}ED\Delta\tau_p \\
&= \dot{x}(t) + t_sE\theta_p + \frac{t_s^2}{2}E\dot{\theta}_p + \left[ t_sB + \frac{t_s^3}{6}ED \right] \Delta\tau_p \\
\Rightarrow \Delta\tau_p &= \left[ t_sB + \frac{t_s^3}{6}ED \right]^{-1} \left( \dot{x}_{\text{ref}} - \dot{x}(t) - t_sE\theta_p - \frac{t_s^2}{2}E\dot{\theta}_p \right)
\end{aligned}$$

vilket ger  $\Delta\tau_p$  för att optimera första kriteriet.

För att optimera nästa kriterium  $\theta(t+t_s) = 0$  taylorutvecklas  $\theta$  i andra graden

$$\begin{aligned}
\theta(t+t_s) &= \theta_p(t) + t_s\dot{\theta}_p(t) + \frac{t_s^2}{2}\ddot{\theta}_p(t) \\
&= \theta_p(t) + t_s\dot{\theta}_p(t) + \frac{t_s^2}{2}D\Delta\tau_p
\end{aligned}$$

För att uppfylla kriteriet  $\theta_p(t+t_s) = 0$  löses

$$\begin{aligned}
\theta_p(t+t_s) = 0 &= \theta_p(t) + t_s\dot{\theta}_p(t) + \frac{t_s^2}{2}D\Delta\tau_p \\
\Rightarrow \Delta\tau_p &= \left[ \frac{t_s^2}{2}D \right]^{-1} \left( -\theta_p(t) - t_s\dot{\theta}_p(t) \right)
\end{aligned}$$

Slutligen taylorutvecklas kriteriet  $\dot{\theta}(t+t_s) = 0$ , den här gång i första graden.

$$\begin{aligned}
\dot{\theta}_p(t+t_s) &= \dot{\theta}_p(t) + t_s\ddot{\theta}_p(t) \\
&= \dot{\theta}_p(t) + t_sD\Delta\tau_p
\end{aligned}$$

och löser ut  $\Delta\tau$ .

$$\begin{aligned}
\dot{\theta}_p(t+t_s) = 0 &= \dot{\theta}_p(t) + t_sD\Delta\tau_p \\
\Rightarrow \Delta\tau_p &= [t_sD]^{-1} (-\dot{\theta}_p(t))
\end{aligned}$$

Som tidigare förklarats superpositioneras dessa tre optimala styrsignaler för en slutgiltig styrsignal  $\tau(t) = \tau_{\dot{x}=0}(t) + \tau_{\theta_p=0}(t) + \tau_{\dot{\theta}_p=0}(t)$ .

För att mäta  $\dot{x}$  används följande formel för konvertering från ERPM till hastighet. Följande antar att hjulet har ständigt kontakt med marken och därmed resulterar rotation av hjulet i direkt translation av rörelsen in planet.

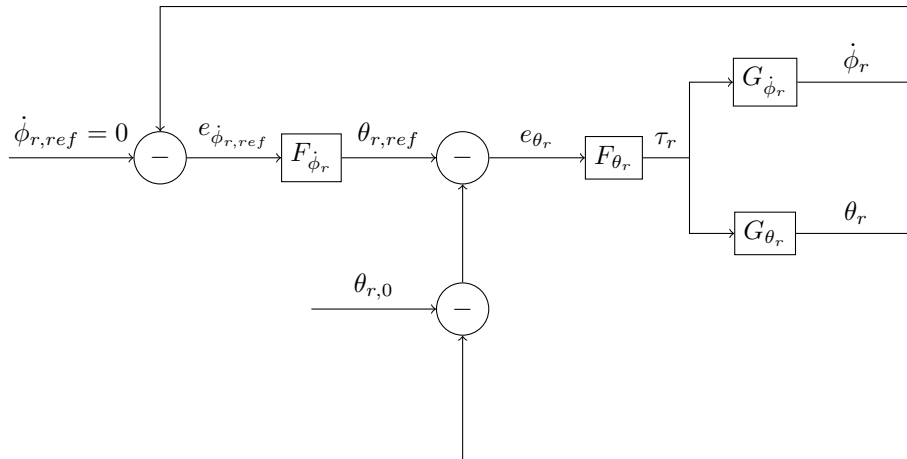
$$\begin{aligned}
ERP M &= nRPM \\
RPM &= \frac{30}{\pi r} v_x \\
\Rightarrow ERP M &= n \frac{30}{\pi r} v_x \\
\Rightarrow v_x &= \frac{ERP M}{30n} \pi r \\
n &: \text{Antal polpar}
\end{aligned}$$

#### 4.1.2 Kaskadreglering i roll-led

Roll-regulatorns jobb är att låta enhjulingen balansera uppåt i roll-led. Då masscentrum av systemet inte är exakt centrerat så är systemet i balans då  $\theta_r = \theta_{r,0}$ , där  $\theta_{r,0}$  är en referensvinkel där masscentrum ligger precis ovanför balanspunkten.

Reaktionshjulet i roll-led som drivs av denna regulator, har ett maxvarvtal som regulatorn bör underskrida. Att hålla hastigheten av roll-hjulet nära noll, alltså att  $\dot{\phi}_r(t) = 0$  lågfrekvent, är därav ett optimeringskriterium.

För detta används kaskadreglering. Som figur (4.1) visar så reglerar den inre regulatorn enhjulingens roll-vinkel  $\theta_r$  runt referensvinkeln  $\theta_{r,ref}$  som regleras runt  $\theta_{r,0}$  av en yttre regulator. Den yttre regulatorn reglerar i sin tur reglerar  $\dot{\phi}_r$  runt nollan.



Figur 4.1: Blockdiagram av kaskadregulatorn i roll-led

Analys av systemfunktionerna utgår från att referensvinkel för masscentrum är i nollan,  $\theta_{r,0} = 0$ . Denna referensvinkel kalibreras genom att manuellt ställa enhjulingen i det närmaste möjliga balanspunkt och läsa av vinkeln.

Systemfunktionerna  $G_{\dot{\phi}_r}(s)$  och  $G_{\theta_r}(s)$  härleds utifrån ekvation (3.12) och (3.6):

$$\begin{aligned}
\ddot{\theta}_r(t) &= \frac{1}{I_{tr}} (\tau_{gr0} \theta_r(t) - \tau_r(t)) && \text{Från ekvation(3.12)} \\
\Rightarrow [\text{Laplace}] \Rightarrow & \\
s^2 \theta_r(s) &= \frac{\tau_{gr0}}{I_{tr}} \theta_r(s) + \frac{-1}{I_{tr}} \tau_r(s) \\
\frac{\theta_r(s)}{\tau_r(s)} = G_{\theta_r}(s) &= \frac{-1}{I_{tr}} \frac{1}{s^2 - \frac{\tau_{gr0}}{I_{tr}}}
\end{aligned}$$

$$\begin{aligned}
\ddot{\phi}_r(t) &= \frac{\tau_r(t)}{I_{hr}} && \text{Från ekvation (3.6)} \\
&\Rightarrow [\text{Laplace}] \Rightarrow \\
s\dot{\phi}_r(s) &= \frac{\tau_r(s)}{I_{hr}} \\
\frac{\dot{\phi}_r(s)}{\tau_r(s)} &= G_{\dot{\phi}_r} = \frac{1}{I_{hr}s}
\end{aligned}$$

För den inre regulatör väljs en PD-regulator  $F_{\theta_r}(s) = k_{P1} + k_{D1}s$ , detta då systemfunktionen  $G_{\theta_r}$  är en andra ordningens integrerande process. Detta ges en total systemöverföring  $G_{\theta_r, \theta_r, ref}$  som

$$\begin{aligned}
G_{\theta_r, \theta_r, ref} &= \frac{G_{\theta_r} F_{\theta_r}}{1 + G_{\theta_r} F_{\theta_r}} \\
&= \frac{-k_{D1}}{I_{tr}} \frac{s + \frac{k_{P1}}{k_{D1}}}{s^2 - \frac{k_{D1}}{I_{tr}}s - \frac{k_{P1} + \tau_{gr0}}{I_{tr}}}
\end{aligned}$$

En nollpunkt finns i  $s$ -planet vid  $s = -\frac{k_{P1}}{k_{D1}}$ , samt två poler vid rötterna till  $s^2 - \frac{k_{D1}}{I_{tr}}s - \frac{k_{P1} + \tau_{gr0}}{I_{tr}}$ , alltså:

$$s = \frac{k_{D1}}{2I_{tr}} \pm \sqrt{\left(\frac{k_{D1}}{2I_{tr}}\right)^2 + \frac{k_{P1} + \tau_{gr0}}{I_{tr}}}$$

För att garantera ett stabilt system behöver alla poler  $s$  befinna sig på vänstra halvplanet, alltså  $\text{Re } s < 0$ . Detta ges då både första termen och termen under rottecknet båda är negativa, vilket resulterar i ett avtagande oscillerande system

$$k_{D1} < 0 \wedge k_{P1} < -\tau_{gr0} - \frac{k_{D1}^2}{4I_{tr}}$$

Alternativt går det att ha ett positivt uttryck under rottecknet, om hela uttrycket blir negativt är det ett stabilt system som ej oscillerar

$$k_{D1} < -2I_{tr} \sqrt{\left(\frac{k_{D1}}{2I_{tr}}\right)^2 + \frac{k_{P1} + \tau_{gr0}}{I_{tr}}} \wedge k_{P1} > -\tau_{gr0} - \frac{k_{D1}^2}{4I_{tr}}$$

För det yttre systemet  $F_{\dot{\phi}_r}$  ansättes en P-regulator  $F_{\dot{\phi}_r}(s) = k_{P2}$ . Parametrarna  $k_{P1}$ ,  $k_{D1}$  samt  $k_{P2}$  som uppfyller stabilitetskraven ovan kalibreras för hand inom simulatören och i verkligheten för att få ett snabbt och stabilt system.

### 4.1.3 Kaskadreglering i pitch-led

Topologin som används i roll-kaskadregulatören kan även användas i pitch-led. Här ska enhjulingen balansera runt  $\theta_p = \theta_{p,0}$  som kalibreras på samma vis. Den yttre regulatören håller pitch-hjulet runt en referenshastighet  $\dot{\phi}_p = \dot{\phi}_{p,ref}$ .

Samma analys som innan kan göras för att hitta systemöverföringsfunktionen för detta reglersystem. Parametervärdena bestäms inte analytiskt här, istället bestäms de genom simulering och kalibrering i verkligheten. Därav tas inte denna analytiska systemöverföring ut här.

## 4.2 Kalmanfilter

Kalmanfiltret för enhjulingen kommer att ha två syften, reducera brus och uppskatta tillstånd som är svåra att mäta [14]. Kalmanfiltret delas upp efter de olika axlarna för att hålla nere storleken på matriser. Ett kalmanfilter formuleras vanligtvis som en linjär tillståndsmo-  
dell enligt [14] [15]:

$$x_k = Fx_{k-1} + Bu_{k-1} + P_{k-1} \quad (4.2)$$

$$z_k = Hx_k + R_k \quad (4.3)$$

Beteckning	Beskrivning
$F$	Tillståndsövergångsmatrix
$B$	Styrningsmatrix
$P$	Felkovariansmatrix
$H$	Avläsningsmatrix
$K$	Förstärkningsvektor
$Q$	Processosäkerhetsmatrix
$R_k$	Mätosäkerhetsmatrix vid tidpunkten $k$
$x_k$	Tillståndsvektor vid tidpunkten $k$
$y_k$	Differens mellan uppskattning och avläsning

Subskript  $k$  betecknar tidsteg. Vid ekvationer som går från ett tidsteg  $k$  till  $k + 1$  används notationen  $x_{till, från}$ . Från ekvationerna (4.2), (4.3) går det att ta fram ekvationerna (4.4 - 4.9). Kalmanfiltret delas upp det approximerande stadiet och uppdaterings stadiet. Ekvationer ger en approximation av tillstånden i vektorn  $x_k$  i tidsteg  $k$  utifrån tillståndsövergångs matrisen och styrningsmatrisen.

Approximeringsstadiet:

$$\hat{x}_{k+1,k} = F\hat{x}_{k,k} + Bu_k \quad (4.4)$$

$$P_{k+1,k} = FP_{k,k}F^T + Q \quad (4.5)$$

Uppdateringsstadiet:

$$y_k = z_k + H\hat{x}_{k,k-1} \quad (4.6)$$

$$K_k = P_{k,k-1}H^T(R_k + HP_{k,k-1}H^T)^{-1} \quad (4.7)$$

$$\hat{x}_{k,k} = \hat{x}_{k,k-1} + K_k y_k \quad (4.8)$$

$$P_{k,k} = (I - K_k H)P_{k,k-1}(I - K_k H)^T + K_k R_k K_k^T \quad (4.9)$$

Ekvation (4.4, 4.5) används för att approximera den nya tillståndsvektorn och felkovariansen utifrån tidigare värden. För att detta ska fungera krävs initialiserings värden för tillståndsvektorn  $x_k$ . Ekvationer (4.7, 4.8, 4.9) uppdaterar tillståndsvektorn och felkovariansen utifrån vektorn  $K$ . Vektorn  $K$  beräknas av de tidigare felkovarianserna. Det gör att kalmanfiltret kommer vikta sensoravläsningen  $z_k$  och det approximerade tillståndet  $x_k$  olika beroende på hur  $R$ ,  $Q$  och  $P$  skiljer sig åt.

I detta projekt antas att  $R$  är ett konstant värde som uppskattas genom att mäta variansen från sensorerna. Enligt [16] är en god grundregel för  $Q$ -matrisen att låga värden ger mycket utjämningsseffekt, vilket gör att filtret inte reagerar på snabba förändringar. Höga värden på  $Q$  gör tvärtom, att lite filtrering sker och filtret kommer vara känsligare för de avlästa sensorvärdena.

### 4.2.1 Pitch

Vektorn  $x_{k_p}$  beskriver de tillstånd kalmanfiltret ska approximera i tidpunkten  $k$ . För pitch-led är det vinkeln  $\theta_p$  och vinkelhastigheten  $\dot{\theta}_p$ , samt drivhulets position  $x_p$  och hastighet  $\dot{x}_p$ .

$$x_{k_p} = [\theta_p \quad \dot{\theta}_p \quad x_p \quad \dot{x}_p]^T \quad (4.10)$$

Matrisen  $H_p$  är avläsningsmatrisen. Den multiplicerat med tillståndsvektorn  $x_k$  ger faktiskt mätbara tillstånd. För pitch-led är det  $\dot{\theta}_p$  och  $\dot{x}_p$ . Faktorn  $\frac{30}{r_{hp}\pi}$  konverterar från vartal till hastighet [ $\text{m s}^{-1}$ ].

$$H_p = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & \frac{30}{r_{hp}\pi} \end{bmatrix} \quad (4.11)$$

$F_p$  beskriver hur tillstånden övergår till ett nytt tillstånd vid approximering. Exempelvis  $\theta_{p_{k+1}}$  är lika med föregående värde  $\theta_{p_k}$  plus integrering av  $\dot{\theta}_{p_k}$  med tidsteget  $dt$ , enligt ekvation (4.4)

$$F_p = \begin{bmatrix} 1 & dt & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & dt \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.12)$$

I processosäkerhetsmatrisen  $Q$ , används två konstanter  $q_{\theta_p}$  och  $q_{h_p}$ . Dessa används för att kalibrera kalmanfiltrets snabbhet och känslighet. Dessa termer ska representera hur stor osäkerhet processen som

ska approximeras har. Hur matrisen  $Q$  ska konstrueras är en avancerat ämne som ingen i gruppen har vidare kunskaper inom. Efterforskningar visade att en diskret brusmodell enligt ekvation (4.13) verkade lämpligt [17]:

$$Q_p = \begin{bmatrix} q_{\theta p} \frac{dt^4}{4} & q_{\theta p} \frac{dt^3}{2} & 0 & 0 \\ q_{\theta p} \frac{dt^3}{2} & q_{\theta p} dt^2 & 0 & 0 \\ 0 & 0 & q_{hp} \frac{dt^4}{4} & q_{hp} \frac{dt^3}{2} \\ 0 & 0 & q_{hp} \frac{dt^3}{2} & q_{hp} dt^2 \end{bmatrix} \quad (4.13)$$

I mätosäkerhetsmatrisen  $R_p$  antas det att variansen  $\sigma$  kommer vara normalfördelad. Det antas också att variansen är konstant och approximerar den utifrån analys av de två olika sensorernas varians.

$$R_p = \begin{bmatrix} \sigma_{\theta p} & 0 \\ 0 & \sigma_{wp} \end{bmatrix} \quad (4.14)$$

Analys av gyrometers mätvärden i statistiskt tillstånd och motordrivarens varvtalsensor i låga varvtal gav följande:

$$\sigma_{\theta p} \approx 0.3 \text{ deg/s} \quad (4.15)$$

$$\sigma_{wp} \approx 20 \text{ RPM} \quad (4.16)$$

I detta projekt användes aldrig en styrningsmatris  $B$  eftersom gruppen hade svårt att tolka hur den skulle implementeras. Försök att använda en  $B$  matris gav resultat som var svåra att tolka och orimliga värden.

## 4.2.2 Roll

I roll axeln används följande matriser för kalmanfiltret. Dessa matriser är likadana som för pitchmatriserna gällande  $\theta_p$  och  $\dot{\theta}_p$ .

$$x_{k_r} = [\theta_r \quad \dot{\theta}_r]^T \quad (4.17)$$

$$F_r = \begin{bmatrix} 1 & dt \\ 0 & 1 \end{bmatrix} \quad (4.18)$$

$$H_r = [0 \quad 1] \quad (4.19)$$

$$Q_r = \begin{bmatrix} q_{\theta r} \frac{dt^4}{4} & q_{\theta r} \frac{dt^3}{2} \\ q_{\theta r} \frac{dt^3}{2} & q_{\theta r} dt^2 \end{bmatrix} \quad (4.20)$$

$$R_r = [\sigma_{\theta r} \quad 0] \quad (4.21)$$

Enligt ekvation 4.15 antas variansen densamma för roll som pitch.

$$\sigma_{\theta r} \approx 0.3 \text{ deg/s} \quad (4.22)$$

## 4.3 Komplementfilter

Komplementfilter är en metod som kombinerar två eller flera sensorers värden gemensamt för att ge en bättre uppskattning av en variabel än vad varje sensor kan ge individuellt. Specifikt jobbar komplementfilter med sensorer där vissa ger ett bra högfrequenssvar och andra ger ett bra lågfrequenssvar. I detta arbete kombineras accelerometrar och gyrometrar i tre axlar för att uppskatta pitch- och rollvinkel  $\theta_p$  respektive  $\theta_r$ .

Att uppskatta vinkel utifrån gyrometer ger en bra momentan uppskattning, men efter längre integrering av gyrometers vinkelhastighet bildas en förskjutning, ett lågfrequent vinkelfel [9]. Accelerometern som mäter jordens gravitationsriktning ger en bra lågfrequent uppskattning — i det stora hela kommer jordens gravitationskraft ge en konstant kraft neråt, medans momentana krafter på enhjulingen inte varar länge [18].

I pitch-led benämns gyrometers sensorvärde som  $\hat{\omega}_p$  samt accelerometers sensorvärde i  $x$ - samt  $z$ -led som  $\hat{a}_x$  samt  $\hat{a}_z$ . Tecknet på  $\hat{a}_z$  är positivt uppåt, tecknet på  $\hat{a}_x$  matchar  $\theta_p$ .

När enhjulingen är upprätt i roll-led,  $\theta_r = 0$ , kommer sensorvärdena utan tillagt brus beskrivas av

$$\begin{aligned}
\hat{\omega}_p &= \dot{\theta}_p \\
\hat{a}_x &= g \sin \theta_p \\
\hat{a}_z &= -g \cos \theta_p \\
&\Rightarrow [\text{Lös ut } \theta_p] \Rightarrow \\
\theta_p &= \int \hat{\omega}_p dt \\
\theta_p &= -\tan^{-1} \left( \frac{\hat{a}_x}{\hat{a}_z} \right)
\end{aligned}$$

För att konstruera ett komplementfilter bildas en lågpasfunktion  $G(s)$  samt en högpasfunktion  $\overline{G}(s)$ , där högpasfiltret appliceras på gyrometervärdena samt lågpasfiltret på accelerometervärdena. Det är viktigt att  $G(s) + \overline{G}(s) = 1$  för att filtrena ska komplettera varandra och inte ge en förstärkning eller dämpning av sensorvärdena.

$$\theta_p = \overline{G}(s) \left( \int \hat{\omega}_p dt \right) + G(s) \left( -\tan^{-1} \left( \frac{\hat{a}_x}{\hat{a}_z} \right) \right) \quad (4.23)$$

Ett vanligt val av filter är ett första ordningens låg- samt högpasfilter med samma filtertid  $\tau_f$  [18]:

$$\begin{aligned}
G(s) &= \frac{\tau_f s}{1 + \tau_f s} \\
\overline{G}(s) &= \frac{1}{1 + \tau_f s}
\end{aligned}$$

Dessa summerar till 1:

$$\begin{aligned}
G(s) + \overline{G}(s) &= \frac{\tau_f s}{1 + \tau_f s} + \frac{1}{1 + \tau_f s} = \\
&= \frac{1 + \tau_f s}{1 + \tau_f s} = 1
\end{aligned}$$

Filtret blir detsamma för båda pitch och roll, det hela leder till följande differentialekvation:

$$\dot{\theta}_p = \hat{\omega}_p + \frac{1}{\tau_f} \left( \tan^{-1} \left( \frac{\hat{a}_x}{\hat{a}_z} \right) - \theta_p \right) \quad (4.24)$$

vilket diskretiseras med ett tidssteg  $\Delta t$

$$\theta_p(t + \Delta t) = \left( 1 - \frac{\Delta t}{\tau_f} \right) \theta_p(t) + \frac{\Delta t}{\tau_f} \tan^{-1} \left( \frac{\hat{a}_x}{\hat{a}_z} \right) \quad (4.25)$$

Filtertiden  $\tau_f$  kalibreras genom att visualisera vinkeln via ett dataprogram som ritar upp den uppskattade vinkeln samtidigt som konstruktionen roteras för hand.

Exakt samma modell används i roll-led, men med gyro- och accelerometervärden i pitch  $\hat{\omega}_r$ ,  $\hat{a}_y$  och  $\hat{a}_z$ . De två komplementfiltrernas filtertider  $\tau_f$  kalibreras separat.

# Kapitel 5

## Konstruktion

I detta kapitel beskrivs hur konstruktionen av prototyperna gick till. Konstruktionen delas i tre delar: mekanik, elektronik och mjukvara.

### 5.1 Mekanik

Här förklaras genomförandet av konstruktion av de rent mekaniska delarna av prototypen, ingen elektronik eller mjukvara utan själva chassit.

#### 5.1.1 Prototyp 1

Den första prototypen tillverkades med målsättningen att undersöka hur dynamiken i prototypen fungerade i praktiken, alltså hur den mekaniska prototypen svarar på den drivande motorn. För att testa detta beslutades att alla frihetsgrader skulle låsas utom pitch. Detta resulterade i att konstruktionen skulle behöva ha stödhjul som stabilisering.

Delarna i prototyp 1 var: två sidopaneler, en topplatta och det drivande hjulet. Sidopanelerna tillverkades av aluminiumplåt på 2 mm. Först bockades plåten för att skapa stöden där stödhjulen skulle sitta. Därefter borrarades det ut hål för fästning av drivande motor samt topplatta. Kanterna på sidopanelerna bockades för att öka styvheten. Även topplattan tillverkades av aluminiumplåt på 2 mm. Plåten bockades så att den bildade en låda vars syfte var att öka hållfastheten och möjliggöra infästning i sidopanelen. Detaljer om delar i prototyp 1 erhålls i bilaga C.1.



Figur 5.1: Rending av CAD-model för prototyp 1

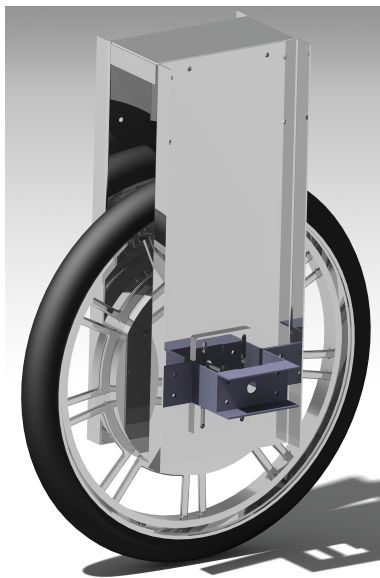
Tabell 5.1: Första prototypens parametrar

Beteckning	Beskrivning	Värde	Enhet
$m_t$	Massa totalt	11,63	[kg]
$m_{hp}$	Massa pitch hjulet	9,3	[kg]
$I_{hp}$	Tröghetsmoment pitch hjulet	0,114	[kgm <sup>2</sup> ]
$m_c$	Massa chassi	2,33	[kg]
$I_c$	Tröghetsmoment chassi	0,126	[kgm <sup>2</sup> ]

### 5.1.2 Prototyp 2

Den andra prototypen tillverkades med samma målsättning som den första. Skillnaden är att stödhjulen togs bort och, istället, konstruerades två handtag som satt fast på varsin sida av sidopanelerna.

Handtagen bestod av två vinklar i aluminium som fästes på varsin sida av sidopanelen. På vinklarna fästes sedan ett u-järn. Sidopanelerna i sig konstruerades likt sidopanelerna i prototyp 1, förutom att fästet för stödhjulen inte längre finns med. Sedan knöts ett rep fast på varje handtag. När sidopanelerna monterades ihop med hjulet sattes handtagen på samma höjd som hjulets vridpunkt. På detta sätt kunde alla frihetsgrader, utom i pitch-led, låsas när repen dras samtidigt. Därmed byggdes två tilläggspaneler (ej visade i figuren) med syftet på att kunna montera fast en vikt på dem, vilket i sin tur höjde upp masscentrum för hela prototypen. På panelerna fanns även ett fåtal håll på olika höjd som gav möjligheter att justera höjden på masscentrumet. Den lägsta höjden vikten kunde sättas på var 200 mm över hjulet och högst 500 mm. Detaljer för delar i prototyp 2 erhålls i bilaga C.2.



Figur 5.2: Rendering av CAD-model för prototyp 2

Tabell 5.2: Andra prototypens parametrar

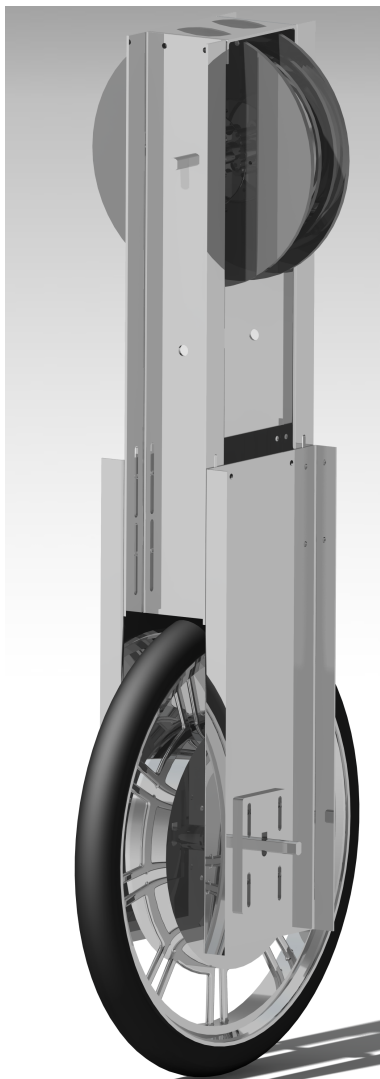
Beteckning	Beskrivning	Värde	Enhet
$m_t$	Massa totalt	11,135	[kg]
$m_c$	Massa chassi	1,835	[kg]
$I_c$	Tröghetsmoment chassi	0,088	[kgm <sup>2</sup> ]

### 5.1.3 Prototyp 3

Den tredje prototypen byggdes med den andra prototypen som bas. Efter att ha byggt prototyper med stabilitetshjälpmedel som stödhjul och rep var det sista steget att bygga en prototyp som har möjlighet att balansera i roll av utan stabilitetshjälpmedel. Utförandet av detta gjordes med prototyp 2 som grund

och med målet att fästa en motor med ett reaktionshjul som skulle utföra balanseringen i roll. För att få ut ett användbart vridmoment från motorn behövdes det även skapas tilläggsplattor som ökar motorns tröghetsmoment, och då även dess totala vikt.

Delarna i prototyp 3 var två sidopaneler, två motorlås, en topplatta, två u-järn för infästning av roll-motor, roll-motor, fyra cirkulära vikttillägg för roll-motorn. Sidopanelerna i prototyp 3 var samma som i prototyp 2. U-järnen tillverkades av 3 mm aluminiumplåt som hade vattenskurna hål för infästning av roll-motor samt för infästningsmöjligheter i sidopanelerna. Topplattan tillverkades av aluminiumplåt på 2 mm som bockats. De fyra cirkulära vikttilläggen skapades av 3 mm stålplåt. Tilläggen utnyttjade existerande skruvhål på roll-motorn för montering. För att möjliggöra denna montering vattenskars skruvhålen i tilläggen. Detaljer för delar i prototyp 3 erhålls i bilaga C.3.



Figur 5.3: Rending av CAD-modell för prototyp 3

Tabell 5.3: Tredje prototypens parametrar

Beteckning	Beskrivning	Värde	Enhet
$m_t$	Massa totalt	20,5	[kg]
$I_{tr}$	Tröghetsmoment för hela enhjulingen runt punkten den rör marken, i roll-led	9,2	[kgm <sup>2</sup> ]
$I_{hp}$	Tröghetsmoment för reaktionshjulet i roll	0,112	[kgm <sup>2</sup> ]

## 5.2 Elektronik

Figur (2.2) visar en övergripande konstruktion av elektriska komponenter som ingick i prototypen, funktioner av de olika elektriska komponenterna förklaras nedan:

- **Batteri:** batteriet är källan till energin som kommer att sätta igång hela enhjulingen. I ett scenario där båda motorerna snurrar i full fart kommer systemet behöva mycket effekt, och eftersom huvudmotorn som bär alla komponenterna har en nominell spänning på 48 V kommer den inte kunna styras tillräckligt bra med spänningar lägre än 30V. Därför behövs ett batteri som kan ge höga strömmar (>50A) och har en nominell spänning högre än 30 V. LiPo (Lithium polymer batteri) är den sorts batteri som passar bäst för projektet eftersom de kan ge höga effekter och väger mycket mindre jämfört med andra alternativ, men priset på ett LiPo batteri som uppfyller kraven är väldigt högt och ligger utanför budgeten. Under projektet används ett hemmabyggt litiumjon (Li-ion) batteripack, batteripacket har en nominell spänning på 48 V och klarar en kontinuerligt strömförsörjning på 55 A. Det innebär att batteriet kan generera en maximal uteffekt mellan 2 – 3kW, vilket är mer än enhjulingen uppskattas komma att behöva.
- **Mikrokontroller:** mikrokontrollern är själva hjärnan i systemet, all kod och algoritmer kommer att läggas upp i mikrokontrollern som senare kommer att läsa värden från sensorerna och styr motordrivarna för att skapa rörelse och balans. En mikrokontroller av märket STM32 kommer att användas i detta projekt eftersom den har flera egenskaper som liknande processorer inte har, till exempel ett integrerat USB-gränssnitt, en klockfrekvens på hundratals megahertz och ett stort processorminne.
- **Nödstoppsknapp:** systemet använder sig av en nödstoppsknapp som är kopplad till mikrokontrollern genom en cirka 2 m lång kabel. För att enhjulingen ska kunna köra behöver nödstoppsknappen vara ständigt nedtryckt så att ifall ett fel uppstår ska enhjulingen sluta köra genom att knappen släpps. Systemet märker om knappen skulle kopplas loss och slutar även då rulla. Knappen har en inbyggd LED som används för att visa att systemet är redo att starta.
- **VESC:** en mikrokontroller kan inte driva en elmotor direkt eftersom elmotorn kräver hundratals gånger större effekt än vad mikrokontrollern and ge. Därför används en annan modul som får signal av mikrokontrollern och driver motorn. Två motordrivare av märket FVESC används i detta projekt, vilka är baserade på VESC:s öppna källkod [19]. VESC-drivare är populära bland hobbyingenjörer, speciellt inom elektriska skateboards. Eftersom de är relativt enkla att operera med hög anpassningsbarhet och samtidigt effektivare och mer precisa än mer traditionella motordrivare.  
Alla motorer behöver kunna startas från fullständigt stopp, någonting som traditionella ESC:er har svårt för, speciellt vid låga varvtal. VESC-baserade motordrivare klarar detta mycket bättre. VESC-motordrivare kan även läsa ut varvtal från motorn som drivs.
- **Gnistfri koppling:** när batteriet kopplas till VESC kommer en stor mängd ström rusa in i VESC för att ladda upp ett antal kondensatorer, detta orsakar gnistor som under längre tid kommer att förstöra kopplingsdonet. Därför är det rekommenderat att använda sig av kopplingdon som inte orsakar gnistor. Under projektet kommer en XT90-koppling användas.
- **Nedspänningsomvandlare:** STM32 och andra sensorer som används i kretskortet har en ingångsspänning på 3.3V och därmed måste spänningen från batteriet som kan gå upp till 56.4V sänkas till detta. Sänkningen av spänningen sker i två steg, i steg ett är huvudkomponenten en LM5017MR som är en switchande regulator. Med hjälp av regulatorn plus några kompletterande komponenter sänks spänningen till c.a. 10-12V beroende på batterispänning och last. I steg två används en linjär spänningsregulator för att sänka spänningen vidare till 3.3 V. Anledningen till att sköta spänningsregleringen i två steg är att det första steget med en switchande regulator har väldigt hög verkningsgrad, men låg noggrannhet. Att sänka spänningen direkt till 3.3V här skulle resultera i en spänning med stor variation vilket antagligen inte kan driva mikrokontrollern. Det andra steget med en linjär regulator har väldigt dålig verkningsgrad, men hög noggrannhet. Att sänka spänningen från batterispänning till mikrokontrollerns spänning skulle resultera i stor effektutveckling i denna komponent, vilket den antagligen inte skulle klara av.
- **Kombinerad gyrometer och accelerometer:** för att kunna mäta acceleration och rotation av enhjulingen använder projektet en MPU6050 som kombinerar en elektroniskt gyrometer och accelerationssensor. En gyrometer används för att kunna mäta vinkelhastighet [20]. En accelerometer

mäter förändring av hastighet relativt sig själv [21]. MPU6050 kan mäta vinkelhastighet [deg/s] och acceleration [ $\text{m s}^{-2}$ ] i alla tre axlar. Informationen om riktningförändringar av enhjulingen från MPU6050 kommer användas som återkoppling i reglersystemet. En fördel med MPU6050 är att den har ett inbyggt analogt lågpasfilter, vilket innebär att mikrokontrollern inte behöver sampla värdena lika ofta för att filtrera själv.

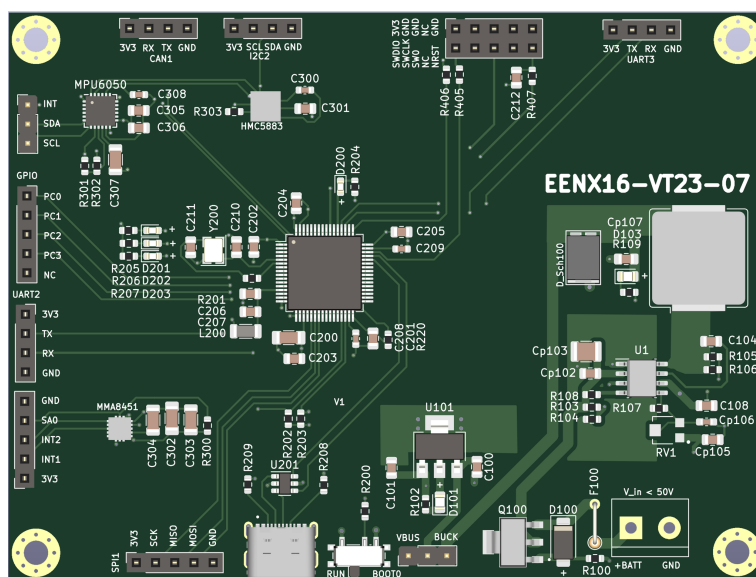
- Motorer: motorerna används för att upprätthålla enhjulingen och för förflyttningen. Motorerna är borstlösa och har inbyggda hall-sensorer för att återkoppla motorpositionen till motordrivaren. Specifikationerna av motorerna är till stor del okända grund av bristfällig information från säljaren.

Den enda modulen som behövde byggas för enhjulingen var kretskortet. Kretskortets kretschema och formgivning (PCB layout) gjordes i KiCad som är ett datorprogram med öppen källkod. Sedan beställdes kretskortet på JLCPCB som är en kretskorttillverkare baserad i Kina. De flesta passiva komponenter som ingick i inköpslistan för kretskortet fanns på redan på ETA (Elektros Teletekniska Avdelning) och resterande komponenter såsom STM32, LM5017MR och sensorer beställdes på LCSC som är en elektronikåterförsäljare baserad i Kina.

Montering av komponenterna gjordes på ETA där alla nödvändiga verktyg för att kunna löda kretskortet fanns. Figur (5.4) visar ett fyra lagers kretskort med STM32, två olika gyrometer, en kompass, nedspänningsomvandlare och andra kompletterande komponenter. Kretskortet delas generellt in i tre delar:

- Strömförsörjning:  
Består av en switchande regulator vars funktion har nämnts tidigare.
- Mikrokontroller och sensorer:  
Det finns totalt tre sensorer och en mikrokontroller monterade på kretskortet vars funktion har redan förklarats. I detta projekt kommer dock bara sensorn MPU6050 användas.
- Portar för externt kommunikation:  
På kretskortet finns USB, SPI, I2C, UART och SWD för att kunna kommunicera med mikrokontrollern, programmera den och styra enhjulingen. MPU6050 använder I2C för att kommunicera. Varje VESC använder en egen UART-kanal för kommunikation. USB används för att programmera mikrokontrollern med ny mjukvara samt för att logga insamlad data under körning till en dator. SWD används för att felsöka felfungerande mjukvara på mikrokontrollern.

Några extra portar (en port för CAN, samt en extra SPI-port) sattes även på kretskortet. Dessa används inte, men kan användas i framtiden för potentiella extra kommunikationsmetoder så kortet kan återanvändas. Även en port för GPIO (general purpose Input/Output) har satts där, några av dessa används för att koppla in nödstoppknappen.



Figur 5.4: 3D modell av kretskortet.

För första modellen används mikrokontrollern STM32H7AZIQ på en Nucleo-144 utvecklingsbräda. Utvecklingsbrädan ger enkel kontakt till en extern dator, där mikrokontrollern både kan programmeras

och kommunicera över UART med en dator med hjälp av en och samma USB-port. Mikrokontrollern kommunicerar med en FSESC75100 (en model av VESC) över en UART-koppling, samt med en gyrometer MPU6050 över I2C. Alla dessa kopplades med lösa kablar till Nucleons kontaktdon. FSESC:en drivs av ett 48 V Li-ion batteri, och STM32:an strömförs antingen av en laptop eller av en powerbank fäst på den mekaniska konstruktionen.

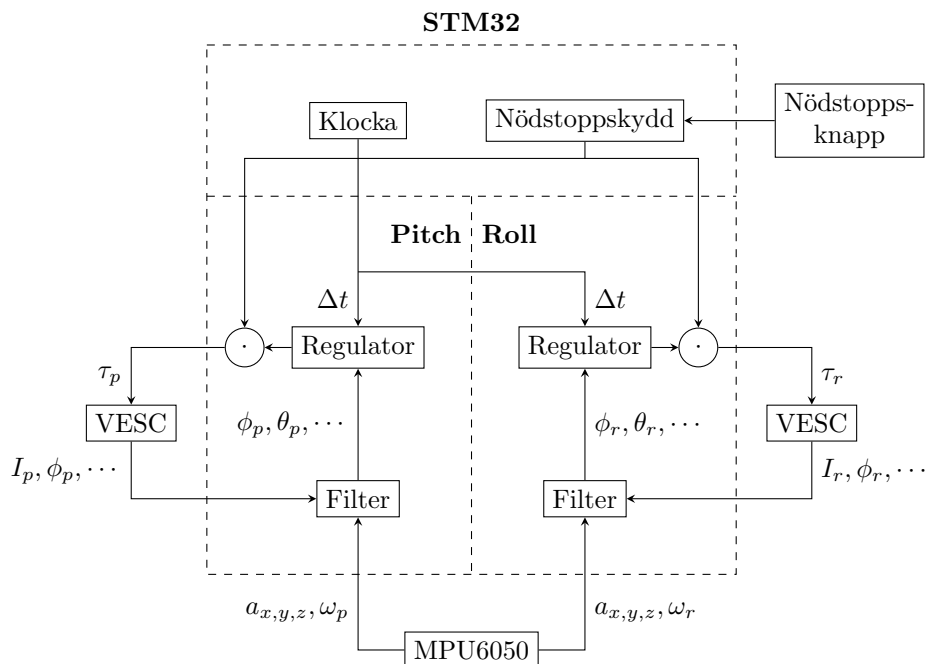
För kretskortet byttes mikrokontrollern till en STM32F405RGT. Detta är en liknande mikrokontroller till den tidigare, men den har en lägre klockfrekvens och färre funktioner. Anledningen till bytet av mikrokontroller var huvudsakligen för att många funktioner på den tidigare mikrokontrollern inte användes, samt att processorn spenderade en stor andel av tiden utan att göra något. Tiden det tog att läsa data från sensorer och att kommunicera med VESC:en var mycket högre än styrsystemet tog att stega, alltså underanvändes processorns prestanda till stort. Dessutom var den exakta modellen av mikrokontroller som användes innan slut i lager, så ett byte behövde göras. Tack vare STM32:s HAL-bibliotek (hardware abstraction layer) behövde inte mycket av koden skrivas om för att fungera på den nya mikrokontrollern.

### 5.2.1 Kompletterande komponenter

Förutom mikrokontrollern och sensorer på kretskortet, finns många andra passiva och aktiva komponenter för att få ett fullfungerande system. Tabell (A.1, A.2) i bilaga A visar alla komponenter med varsitt värde och antal som ingår i kretskortet.

## 5.3 Mjukvara

För att implementera styrsystemet och kontrollera motorerna behövs elektronisk hårdvara i form av mikrokontroller, sensorer och motordrivare, samt mjukvara på mikrokontrollern där styrsystemet, filter och andra funktioner kör.



Figur 5.5: Diagram över de huvudsakliga ansvarsområdena och dataflödet i mjukvaruimplementationen.

Mikrokontrollern har fyra ansvarsområden att sköta för att styrsystemet ska fungera: samla data från sensorer, köra styrsystemet utifrån denna data, kontrollera motordrivaren utifrån styrsystemets utsignal samt att schemalägga alla dessa delar så att de processeras med minimal fördröjning.

### 5.3.1 Uppgiftskö och schemaläggning

För schemaläggning av alla aktiviteter används ett kösystem, där mikrokontrollerns huvudtråd konstant lyssnar efter uppgifter att utföra från en kö, och där olika typer av hårdvaruavbrott (interrupt) kan lägga till uppgifter i kön.

Detta kösystem försäkrar att uppgifter processeras i rätt ordning (första tillagda uppgiften i kön utförs först, sista uppgiften sist), samt undviker problem som kan uppstå med kaskad-avbrotts (ett avbrott händer samtidigt som kod från ett tidigare avbrott redan kör).

Var uppgift som kan utföras identifieras av ett ID som representerar typen av uppgift att utföra, samt potentiell associerad data om uppgiften att utföra, vars betydelse varierar med typ av uppgift.

Tabell 5.4: Tabell över uppgifter som kan hamna i kön

ID	Betydelse	Associerad data
NONE	Standby, gör ingenting	[-]
SEND_DEBUG	Skicka debug-data över USB	[-]
TIME_STEP	Låt kontrollsystemet göra ett tidssteg	[-]
REQ_VESC	Begär data från alla VESC	[-]
FLUSH_VESC	Skicka alla köade meddelanden till VESC	[-]
REQ_ACC	Begär data från MPU6050	[-]
VESC_UART_GOT_DATA	Tolka mottagen data från VESC <sup>1</sup>	VESC-ID (pitch eller roll)
GOT_ACC_DATA	Processera svar från MPU6050	Svar från MPU6050
GOT_ESC_DATA	Processera svar från VESC	Svar från VESC, samt VESC-id

Denna kö är implementerad med en cirkulär buffer, vilket är ett minnesområde som betraktas som cirkulärt, alltså att sista elementet ligger bredvid det första. Detta realiserar genom att pekare till buffern som stegas alltid återvänder till början av buffern om de stegar ur den. Köen har två pekare, en pekare kontrollerar vart kön läser från när systemet vill plocka en uppgift, och en annan pekare kontrollerar var i buffern kön ska skriva då systemet lägger till nya uppgifter.

Innan en uppgift plockas ur kön av huvudtråden kontrollerar kön om det finns en uppgift i kön att ge ut (alltså att pekarna inte är samma). Om ingen uppgift finns i kön skapas en uppgift med ID NONE, vilket innebär att systemet inte ska göra något. Om en uppgift finns att ge ut så returneras uppgiften och kön stegar läs-pekaren ett steg framåt i buffern.

När systemet vill lägga till en ny uppgift i kön kontrolleras det först att utrymme finns i kön, det vill säga att skriv-pekaren inte är precis ett steg bakom läs-pekaren. Om det finns utrymme i kön skrivs uppgiften in och skriv-pekaren stegas ett steg framåt. Om utrymme inte finns i buffern så returneras ett felmeddelande.

Uppgifter som TIME\_STEP och liknande behöver regelbundet utföras vid ett visst tidsintervall för att kontrollsystemet ska arbeta. En 10 kHz tidtagare divideras och för att lägga olika typer av meddelanden i kön. Dessa meddelanden presenteras i tabell (5.5). Frekvenserna har valts för att inte vara höga nog att kön blir överbelastad, alltså att uppgifter läggs i kön snabbare än huvudtråden plockar och behandlar dem.

Tabell 5.5: Frekvens för schemaläggning av olika uppgifter

ID	Frekvens
REQ_ACC	100 Hz
REQ_VESC samt FLUSH_VESC	50 Hz
SEND_DEBUG	100 Hz
TIME_STEP	500 Hz

Kön innehåller några synkroniseringsfunktioner för att undvika race conditions. Ett exempel på race condition som behövde hanteras var när huvudtråden höll på att lägga till en uppgift i kön, och samtidigt utfördes ett avbrott som samtidigt också vill lägga till en uppgift i kön. Här behövs en viss försiktighet så det säkerställs att meddelandet skrivs till minnet och skriv-pekaren stegar synkront, för att inte både huvudtråden och avbrottet båda skriver till samma plats och ökar skriv-pekaren två gånger efteråt.

<sup>1</sup>Detta tolkar endast datan VESC:n skickar för att dela upp den i paket. Om ett paket är av typen COMM\_GET\_VALUES, som innehåller bl.a. data om RPM, så genereras ett MSG\_GOT\_ESC\_DATA-meddelande.

### 5.3.2 Kommunikation

För att samla data till filtret, samt driva motorerna med regulatoren behöver mjukvarosystemet kommunicera med omvärlden. Sensorn MPU6050 som ger accelerometerdata kommunicerar över I2C, båda motordrivare VESC kommunicerar över UART och för nödstoppsskyddet används en enkel GPIO-koppling.

#### MPU6050

I2C-kopplingen för MPU6050 kör på en maxfrekvens av 400 kHz. Varje data-meddelande innehåller 96 bitar information (6 kanaler med 16 bitar värde vardera), vilket betyder att var meddelande tar storleksordningen<sup>2</sup>  $\frac{96\text{bit}}{400\text{kbit/s}} = 240\ \mu\text{s}$ . Då denna tid är nära på försumbar används blockerande anrop för att skicka begäran om värdet. Ett blockerande anrop innebär att mikrokontrollern låser sig efter att ha skickat begäran och inte kan processera någonting annat förrän svar på begäran återvänt. Detta är i kontrast till ickeblockerande anrop, där mikrokontrollern återvänder till normal funktion efter att ha gjort begäran, varpå mikrokontrollern sedan får ett avbrott när svar på begäran kommit.

Vid uppstart konfigureras MPU6050 till att gå ur sömnläge (alla funktioner är normalt avstängda vid uppstart) genom att skriva till register `0x6B-POWER MANAGEMENT 1`. Det interna lågpas-filtret startas genom att skriva till register `0x1A-CONFIGURATION`, mätprecision samt mätområde på gyrometern respektive accelerometern sätts genom att skriva till registern `0x1B-GYROSCOPE CONFIGURATION` respektive `0x1B-ACCELEROMETER CONFIGURATION`. [22]

Under körning gör uppgiften `REQ_ACC` i kön att gyro- och accelerometerdata läses från MPU6050. Register `0x3B-ACCELEROMETER MEASUREMENT` samt `0x43-GYROSCOPE MEASUREMENT` läses. När datan tas emot appliceras en fördefinierad linjär kalibrering på dessa, varpå en `GOT_ACC_DATA`-uppgift generas med den kalibrerade datan och läggs i kön. När denna uppgift senare processeras matas värdena till filtret och vidare till reglersystemet.

#### Nödstoppsknapp

En nödstoppsknapp kopplar till en jord-port, samt två GPIO-pinnar. En GPIO-pinne har en intern resistor kopplad till hög logikspänning, som knappen kortsluter till jord när den trycks ner. Mjukvaran läser av denna port, och nollställer alla motorströmmar om porten visar hög spänning (då knappen alltså inte är nedtryckt och kortsluter porten till jord).

En andra GPIO-pinne används för att driva en lampa på knappen. Lampan blinkar ett mönster i mörkare färg när systemet kör men motorerna är inaktiverade, och lampan lyser starkt när knappen är nedtryckt för att signalera att systemet är aktivt. Ljustyrkan kontrolleras genom pulsbreddsmodulering, där en lägre arbetscykel på 25% ger ett svagare sken.

#### Kontroll av motordrivaren

Som tidigare nämnt kommunicerar motordrivaren VESC över UART. Detaljer om protokollet och meddelandena som skickas över det var svåra att hitta. Gruppen har utgått från [19] för hur protokollet ser ut. En start-byte skickas som bestämmer storleksordningen av datalängden (ett värde 02 betyder att längden av datan är mellan 0 till 255 bytes lång, 03 betyder en längd mellan 0 till 65535 bytes). Efter detta följer en eller två bytes (beroende på första byten) som ger den exakta längden av datan. Efter detta följer datan, följt av en CRC16-kontrollsumma, till sist följt av en slut-byte 03.

Värt att notera är att UART-protokollet arbetar utan parity-bits vilket är en extra fel-detektionsbit som vanligtvis läggs till efter en byte i UART för att representera om antalet 1-bitar i byten är udda eller jämt. Detta skapade mycket förvirring eftersom att UART vanligtvis använder parity-bits.

Formatet av datan är inte dokumenterat någonstans gruppen kunde hitta. Gruppen fick läsa mjukvarans källkod för VESC-systemet [23] samt exempelkod [24] för kommunikation över UART för att komma fram till de relevanta detaljerna av protokollet.

Datan startar med en byte som beskriver vilket kommando som skickas. Ett kommando kan antingen vara en begäran från systemet som kontrollerar VESC, eller ett svar på en sådan begäran från VESC:en. Kommando-byten är följt av ett antal bytes som representerar data tillhörande det kommandot. Dessa består oftast av heltal med tecken, representerade av antingen en, två eller fyra bytes. Data som är decimaltal skalas upp till en viss precision och avrundas till närmaste heltal.

<sup>2</sup>Detta är en uppskattning som endast utgår från att accelerometerdata skickas. I verkligheten har I2C en overhead där både vissa acknowledgement-bitar behöver sändas, samt STM som måste skicka en I2C-förfrågan om att läsa rätt ett specifikt register innan MPU6050 svarar, innebär att denna uppskattning underskattar tiden till viss grad

De till detta projekt relevanta kommandona är SET\_CURRENT (begäran) vars data är strömmen VESC:en ska driva motorn med (enhet 1 mA), GET\_VALUES (begäran) som begär många olika data från VESC:en, samt GET\_VALUES (svar) som innehåller denna data. Datan för GET\_VALUES beskrivs i tabell (5.6).

Tabell 5.6: Strukturen för svaret GET\_VALUES

Storlek i bytes	Representerar	Enhet och skala
2	Transistortemperatur	0.1 °C
2	Motortemperatur	0.1 °C
4	Motorström	10 mA
4	Inström	10 mA
8	? <sup>3</sup>	[-]
2	Arbetscykel	Antagligen i tusendelar
4	Motorhastighet	1 RPM
2	Inspänning	0.1 V
4	Ström förbrukad	1 mA h
4	Ström laddad	1 mA h
4	Effekt förbrukad	1 mW h
4	Effekt laddad	1 mW h
8	Tachometervärden	?
1	Felkod	[-]
4	PID-position	[-]
1	Hårdvaru-ID	[-]

För att driva motorn med vridmomentet  $\tau$  regleras motorerna ström  $I$ . Båda motorerna som används approximeras ha en vridmomentkonstant  $K_t = \frac{I}{\tau} \approx 0.2 \text{ Nm/A}$ , alltså bör motorerna matas med ström  $I = \frac{\tau}{K_t}$ . Motordrivaren har en viss tröghet i strömkontrolleringen, så en enkel P-regulator används för att reglera strömfelet baserat på skillnaden i motordrivarens insignal och uppmätt ström som ges tillbaka. Detta ger en snabbare och mer noggrann strömkontroll, men introducerar brus i strömmen. Det är oklart om bruset är ett riktigt fenomen, eller om det endast är en artefakt av strömmätningen i VESC:en.

### 5.3.3 Styrsystem

Styrsystemet består av ett filter samt en regulator. Varje gång en GOT\_ESC\_DATA- eller GOT\_ACC\_DATA-uppgift processeras sparas datan för att matas in i filtret. Uppgiften TIME\_STEP får mjukvaran att stega filtret och regulatorn ett  $\Delta t$  framåt i tiden. Tidsstegets storlek  $\Delta t$  räknas ut i mikrosekunder utifrån hur många steg en tidtagare tagit sedan senaste TIME\_STEP-meddelandet processerats. Filtret matas in med de senaste sensorvärdena, varpå det får göra ett steg i tid. Filtrets uppskattning av systemets läge matas in i regulatorn som ger ut en styrsignal för vridmomentet av motorn  $\tau$ .

Tidtagaren, vars uppgift är att räkna  $\Delta t$ , är en 1 MHz-timer med en inbyggd räknare som är inställd att räkna till 1000. Varje gång räknaren nått sin gräns återställs den till 0, och ett tidtagar-avbrott ökar en sekundär mjukvaruräknare. När  $\Delta t$  räknas ut kombineras både tidtagarnes interna räknare samt den sekundära mjukvaruräknaren. Tidtagaren stegar en gång var 1  $\mu\text{s}$  och den sekundära räknaren en gång var 1 ms, så den totala tiden räknas ut enligt:

$$t_{tot} = 1 \mu\text{s} \cdot ctr_{tim} + 1 \text{ ms} \cdot ctr_{sec}$$

där  $ctr_{tim}$  är tidtagarens interna räknare och  $ctr_{sec}$  är den sekundära mjukvaruräknaren.

<sup>3</sup>Osäkert vad denna data representerar

# Kapitel 6

## Tester

I detta kapitel beskrivs de olika tester som gjordes.

### 6.1 Pitch

Det bestämdes tidigt att pitch var den första och mest grundläggande axeln att testa. Balansering i pitch-led ansågs vara svårast och mest avgörande konstruktionsmässigt. Det ansågs viktigt att utseendet för pitch var mer eller mindre färdigt innan projektet gick vidare till roll, för att undvika dubbelt arbete med dimensioneringen av roll.

#### 6.1.1 Simulering

Första steget i projektet var att testa regleringsmodell i simulation. Simuleringens parametervärden för enhjulingens egenskaper baserades på de uppmätta och beräknade värdena allt eftersom prototypen tog form i CAD. Exempelvis hjulet som fanns tillgängligt vägdes och ritades upp i CAD för att ta fram tröghetmomentet, hjulets parametrar finns beskrivet i tabell (5.1).

#### PD-regulator

Först testades den mycket enkla och klassiska PD-regulatorn. För enbart balansering, dvs stå upprätt gav detta snabbt ett tillfredsställande resultat. Denna regulator återkopplade vinkeln  $\theta_p$  med en proportionell och deriverande försträckning till utsignalen  $\tau_p$  som är momentetbegäran till motorn. I simuleringen testades olika referenssingaler till regulatorn för att avgöra stabiliteten, dvs att regulatorn klarade av att hålla en vinkel som inte var helt upprätt.

#### Kaskadreglering

Den tidiga framgången med att kunna balansera enhjulingen i simulering gjorde att ambitionerna snabbt höjdes till att skapa en regulator som kunde ta en in hastighetsbegäran, eftersom detta ansågs som det mest användbara användargränssnittet för regulatorn. Det ledde till försök med kaskadreglering.

Ett problem som upptäcktes med kaskadreglering med hastighet som insignal är riktningförändring på hjulet. För att enhjulingen ska förflytta sig från stillastående måste hjulet först rotera bakåt. Rotation bakåt innebär negativ vinkelhastighet på  $\dot{\phi}_p$ , för att förflytta tyngdpunkten åt den riktningen som enhjulingen ska röra sig. Detta skapar en positiv vinkel på  $\theta_p$ . När tyngdpunkten är förflyttad måste hjulet byta riktning, positiv vinkelhastighet  $\dot{\phi}_p$ , för att rotera i samma riktning som enhjulingen ska för flytta sig.

#### Optimal kontroll-regulator

Svårigheterna med kaskadregleringen ledde till försök med optimal kontroll-regulator. Optimal kontroll-regulatorn klarade av kraftiga stegsvar i hastighetsförändring utan stabilitetsproblem, samtidigt som den klarade av att bromsa enhjulingen till stillastående stabilt. Detta sågs som en lämplig regulator att gå vidare med för test på prototyp.

## 6.1.2 Prototyp

Tre tester utfördes i pitch led på olika prototyper.

### Första test

Efter att ha kopplat samman hårdvarukomponenterna till mikrokontrollern och kod för kommunikation däremellan skrivits, genomfördes det första testet i pitch-led på prototyp 1. Resultatet gav att tyngdpunkten på denna prototypen var för låg, vilket ledde till att prototypen inte fungerade som en inverterad pendel. Detta resulterade i att den inte kunde balansera upp eftersom regleringsmodellen för pitch baserades på en modell av en inverterad pendel. Metallplattorna som stödhjulen satt på slog i marken när prototypen testkördes. När metallplattorna slog i marken introducerades en liten laststörning, vilket gjorde att chassit vred åt motsatt håll. Denna plötsliga ändring av vinkel försvårade dessutom regler-systemet för balanseringen. Stödhjulen bidrog med friktion mot marken, vilket reglermodellen inte tog något hänsyn till. Detta bidrog till försvåring av regleringen. Det resulterade till att en ny prototyp som kunde åtgärda dessa problem byggdes enligt figur (5.2).

### Andra test

Ett enklare test utfördes innan den nya prototypen hade byggts. Stödhjulen monterades bort från gamla prototypen. Två personer stod på varsin sida av prototypen för att försöka med hålla enhjulingen upprätt i roll-led, samtidigt försökte reglersystemet balansera i pitch-led. Detta test misslyckades också, det konstaterades att reglersystemet reagerade som förväntat. Men oftast antigen för långsamt eller kraftiga översvängar. Slutsatsen blev att balanserings metoden för roll inte var god nog. Trots försök till små rörelser på enhjulingen blev det lätt stora riktningförändringar.

Vid analys av datan som samlades in under test konstaterades också att reglersystemet inte exekverade i den takt som var tänkt. Dessutom skedde loggning i en för långsam takt. Dessa problem löstes genom att öka takten i schemaläggningssystemet, samt att öka takten data loggades över UART.

### Tredje test

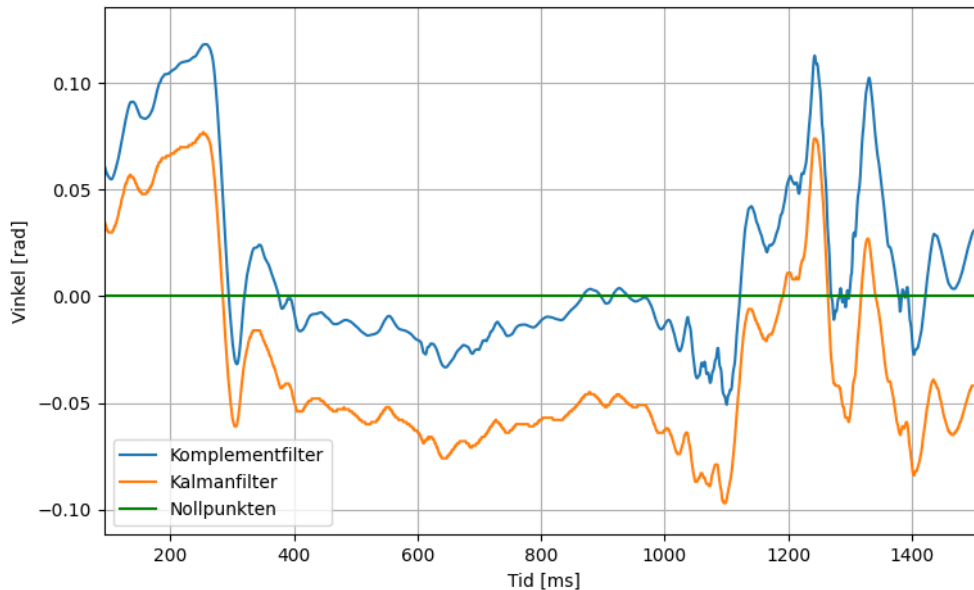
Till detta test hade en ny prototyp byggts, prototyp 2. Denna prototyp hade högre sidopaneler med hål i sig för montering av en vikt vilket ger möjlighet till att justera tyngdpunkten. Utöver detta så slopades stödhjulen, och istället fäster rep i sidan av hjulaxeln. Prototypens frihetsgrader, förutom pitch-led, kunde till viss del begränsas när två personer drog på repet i varsin sida samtidigt. Prototypen tippade åt sidan, i roll-led, vid tillfällena när personerna drog i repet med olika kraft. Tillfälligtvis kunde prototypen väl balanseras upp utan att tippa. När det kom till balanseringen i pitch-led lyckades prototypen inte genomföra arbetet. Testet genomfördes med olika höjd på tyngdpunkten, började först på samma läge som det första testet och sedan höjde upp tyngdpunkten lite åttaget. Prototypen tippade oavsett om tyngdpunkten så långt ner eller så högt upp som den kunde vara. Däremot tenderade prototypen att tippa snabbare när tyngdpunkten satts lägre ner.

## 6.2 Roll

Efter många försök att balansera enbart pitch-led togs beslutet att fokusera på att balansera roll-led. Motivering var att roll-led är lättare att isolera rörelsen i än pitch-led. Ett annat argument var att det blir lättare att balansera pitch-led om den redan kan balansera i roll.

### 6.2.1 Simulering

Simulering av roll enligt ekvationerna i avsnitt 3.1.3 gav att denna axeln var relativt lätt att kontrollera. De två viktigaste parameterarna är varvtalet motorn kan åstadkomma, samt tröghetsmomentet  $I_{hr}$ . Tester visade att motor hade ett maximalt varvtal på  $\approx 600$ RPM. Det innebar att den enda parametern som kunde påverka det moment rollhjulet skapade var tröghetsmomentet  $I_{hr}$  som i sin tur är beroende av massan  $m_{hr}$  och hjulets radie  $r_{hr}$ . Simuleringen visade att kaskadregulatorn beskriven i avsnitt 4.1.2 skulle ha svårt att reglera systemet med dessa parametrar, och att tröghetsmomentet antagligen skulle behöva ökas.



Figur 6.1: Jämförelse mellan komplement filter och kalmanfilter

## 6.2.2 Prototyp

### Test med kalmanfilter

Initialt testades roll-axeln med pitch-hjulet urkopplat men fritt rullande. Testet gjordes genom att direkt koppla motordrivaren till datorn och manuellt ge en strömbegäran för att både undersöka maxhastighet på roll-hjulet, samt vilken vinkelacceleration som kunde uppnås. Det konstaterades ganska tidigt att mer masströghetsmoment krävdes på roll-hjulet. Momentet som bildades var alldeles för lågt för att kunna påverka prototypens toppvinkel i roll axeln. Med hjälp av simuleringen uppskattades en hjulmassa på 8 kg att behövas. Två metallskivor fästes i hjulet för att öka massan och därmed masströghetsmomentet. Efter att samma test utfördes med det nu tyngre hjulet verkade det realistiskt för motorn påverka vinkeln tillräckligt för att hålla enhjulingen upprätt.

Reglersystemet utökades med kalmanfilter för roll enligt avsnitt 4.2.2 samt en kaskadregulator enligt avsnitt 4.1.2. Med det nya reglersystemet hade enhjulingen fortfarande svårt att balansera sig. Analys av loggar från testet tydde på att uppskattningen av vinkel som kalmanfiltret producerade var för långsam att reagera på förändring, samtidigt som vinkeln blev förskjuten med tiden.

Eftersom projektiden snart var slut bestämde sig gruppen att överge kalmanfiltret för att istället använda ett komplementfilter. Efterforskningar tydde på att ett komplementfilter är lättare att få tillräckligt bra prestanda, jämfört med ett kalmanfilter. I figur (6.1) jämförs kalmanfiltret med komplementfiltret. Jämförelsen är gjort i efterhand med datan som loggades under test. Det går att se att kalmanfiltret har en förskjutning från nollpunkten jämfört med komplementfiltret.

Analys av koden för accelerometern och gyrometern visade att det gick att ändra en inställning på den för att öka känsligheten. Fram tills denna tidpunkt hade den varit inställd på maximal mätområde, men på bekostnad av känsligheten [22]. Analys av datan visade på att enhjulingen i balans sällan överskred en vinkelhastighet på  $15 \text{ mrad s}^{-1}$  eller en acceleration på  $15 \text{ ms}^{-2}$  vilket är inom mätområdet på den mest känsliga inställningen på MPU6050.

### Test med komplementfilter

Ett komplementfilter implementerades enligt ekvationerna i avsnitt 4.3. Eftersom tidigare tester tydde på felaktig vinkelmätning var mycket fokus nu på att verifiera realistiska värden på vinkelmätningen. Testet visade att uppskattning av vinkel var känsligare och stabilare nu än med kalmanfiltret, samt inte visa på driften kalmanfiltret hade.

Fokus övergick nu till att försöka balansera enhjulingen enbart i roll. För att lättare testa enbart roll och eliminera ej önskade laststörningar fixerades enhjulingen med hjälp av två träbitar enligt figur (6.2). I



Figur 6.2: Prototyp fixerad för att testa enbart balansering i roll

botten på den nedre träbiten skruvade två skruvar in halvvägs för att skapa en skarp punkt att balansera på.

Genom att för hand hitta balanspunkten i roll-led och läsa av komplementfiltrets uppskattade vinkel kunde referensvinkeln i roll  $\theta_{r,0}$  uppskattas. Efter att ha uppskattat denna tillräckligt noggrant och kalibrerat parametervärdena så fick gruppen en balanserad enhjuling i roll-led.

### 6.3 Pitch och Roll

Efter att ha fått balans i roll-led togs träbitarna bort och det tidigare styrsystemet aktiverades igen. Styr-systemet i pitch, som bestod av optimal kontroll-regulatorn beskriven i avsnitt 4.1.1 samt ett kalmanfilter i avsnitt 4.2.1 lyckades inte komma nära att balansera. Analys av den loggade datan indikerade på att varvtalsuppdateringen som VESC hade dålig noggrannhet i pitch-led, speciellt i lägre varvtal. Eftersom den optimala kontrollregulatorn var beroende av hjulets longitudinella hastighet väcktes tanken om att detta kunde vara en potentiell orsak till bra prestanda på den optimala kontroll-regulatorn i simulering men dålig på prototypen. Inspirerad av framgången av komplementfiltret och kaskad-PD-regulatorn i roll-led, samt att tiden var knapp, bestämde sig gruppen för att prova samma topologi i pitch-axeln.

Kaskad-PD-regulator visade sig även fungera väl i pitch-led, på några dagar gick enhjulingen från att inte kunna balansera alls till att kunna balansera både pitch- och roll-led.

# Kapitel 7

## Resultat

I detta kapitel presenteras resultaten av arbetsgång under projektet.

### 7.1 Hårdvara

- Kretskort: Spänningen från batteriet som är på cirka 50V är sänkt till 4.5V med hjälp av nedspänningsomvandlaren, det finns inga höga värmeutvecklingar eller störande högfrekvens-ljud från spolen, det innebär att nedspänningsomvandlaren fungerar lik förväntad. Dessutom är det möjligt att strömförsörja andra moduler som inte är monterade på kretskortet, direkt från nedspänningsomvandlaren, tack vare den justerbara utspänningen.

Den första iterationen av kretskortet hade två fel, det första var att fotavtrycket för USB-kontakten blev fel. Det andra felet var; för att kunna boota mikrokontrollern behövde pin 28 vara kopplad till jord, vilket inte hade gjorts. Dessutom fanns det förbättringsförslag till andra iterationen såsom en resetknapp på kretskortet, och ett precis matchande fotavtryck till den SWD-kontakten som fanns tillgänglig. Dessa ändringar gjordes i iteration två.

- VESC: VESC:n som har använts i projektet har väldigt bra förmåga att klara höga effekter av cirka 1.5KW kontinuerlig. Att kunna driva motorer med ström är den enklaste formen av styrning, när motordrivarna inte är en begränsande faktor är det en stor fördel. Vanligtvis används VESC för motorer som har höga RPM, vanligtvis över 500. Det innebär att det är svårt att kunna mäta en motors hastighet vid låga RPM. Detta har orsakat hinder för att kunna få ett stabilare system speciellt när det gäller pitch-hjulet då den roterar bara ett par grader när den står stilla.

### 7.2 Balans i stationärt tillstånd

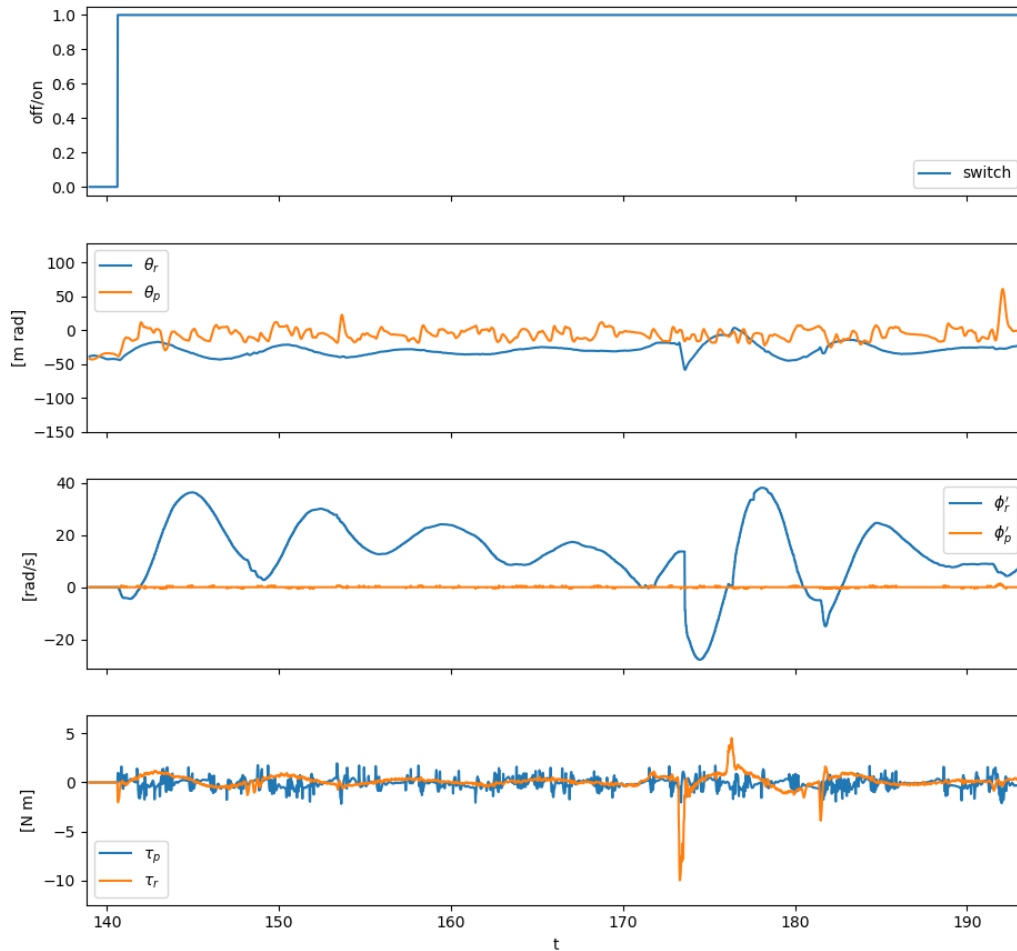
Under första försöket skruvades pitch-hjulet fast på en plank så att testet kunde utföras bara på roll-hjulet. Genom att balansera bara roll-hjulet först, kunde antalet parametrar i reglersystemet som behövde justeras nästan halveras. Navmotorn för roll-hjulet vägde ca 3.4kg, givet dess låga RPM på cirka 600, var 3.4kg inte ens tillräckligt för att bibehålla balansen i roll även om den startades i balans med noll graders vinkel. Efteråt höjdes roll-hjulets vikt till cirka 8 kg för att få mer tröghetsmoment, detta resulterade till bättre balanseringsförmåga i roll, efter ytterligare ändring av parametrar kunde enhjulingen balansera sig i roll-led. Dock är balansen i roll-led väldigt känslig mot störningar.

I steg två, justerades parametrar i pitch-regulatorn för att behålla balansen i pitch-led. Enhjulingen balanserade sig efter några små finjusteringar i pitch-led trots att Pitch-hjulen behöver rotera bara ett par grader för att balansera enhjulingen. Att kunna balansera i pitch med några grader är speciellt svårt eftersom pitch-hjulen och motordrivaren är designad för höga rpm.

### 7.3 Reglersystem

Det slutgiltiga styrsystemet, där pitch- och roll-axeln båda kör med komplementfilter och kaskadreglering, klarar att hålla enhjulingen stabil. I alla tester som gjordes på den slutgiltiga prototypen, prototyp 3, kunde den stå upp i minuter åt gången utan att ramla, såvida inga laststörningar applicerades. Vind och dåligt underlag kunde få enhjulingen att välta omkull. När enhjulingen stod stabilt hände ingen

oscillation i styrsystemet, och den kunde motverka små laststörningar som att till exempel lätt peta på den i roll- eller pitch-led.



Figur 7.1: Datasamling från ett fritt test i pitch och roll

I figur (7.1) ritas data från ett test där enhjulingen åkte fritt med reglersystemet igång. Det syns att både pitch- och rollvinkeln  $\theta_p$ ,  $\theta_r$  håller sig inom  $75 \text{ mrad} \approx 4.3 \text{ deg}$ . Roll-hjulets hastighet  $\dot{\phi}_r$  når ett maximum på c.a.  $40 \text{ rads}^{-1} \approx 380 \text{ RPM}$ , vilket är under maxhastigheten på  $600 \text{ RPM}$ .

De gångerna enhjulingen ramlat omkull har det många gånger varit just på grund av att roll-hjulet nått sin maxhastighet. När maxhastigheten uppnås kan inte något vridmoment appliceras i den riktningen enhjulingen skulle behöva vrida hjulet för att resa upp sig igen. Resultatet blir att enhjulingen ramlar åt sidan, med roll-systemet inkapabelt att alls agera.

De slutgiltiga parametervärdena som användes i reglersystemet presenteras i tabell (7.1). Referensvinklarna  $\theta_{p,0}$  samt  $\theta_{r,0}$  behöver kalibreras om var gång krets-kortet monteras av och på, då monteringen är flexibel nog att accelerometers referensram ändras med några milliradianer. Ingen vinkeldrift har observerats utan att kortet monterats av.

Notera att den inre regulatören i roll-led har negativa parametrar då att rotera hjulet i en riktning får chassit att rotera åt motsatt håll. Alltså ger en positiv styrsignal en negativ påverkan på systemets insignal.

### 7.3.1 Roll

Ett kvantitativt test utfördes på roll-axeln för att se hur stor lutning den kan kompensera för från stillastående läge. Notera att i stillastående läge är både enhjulingens vinkel och roll-hjulets hastighet stilla, när enhjulingen kör på riktigt så kommer båda dessa axlar alltid vara i rörelse, vilket innebär att

Tabell 7.1: Parametervärden i pitch- och roll-regulatorn

Parameter i pitch	Värde	Enhet
$\theta_{p,0}$	-30	[mrad]
$k_{P1}$	30	[-]
$k_{D1}$	8	[-]
$k_{P2}$	0.2	[-]
$\tau_f$	1	[s]
Parameter i roll	Värde	Enhet
$\theta_{r,0}$	-40	[mrad]
$k_{P1}$	-80	[-]
$k_{D1}$	-70	[-]
$k_{P2}$	0.0006	[-]
$\tau_f$	1	[s]

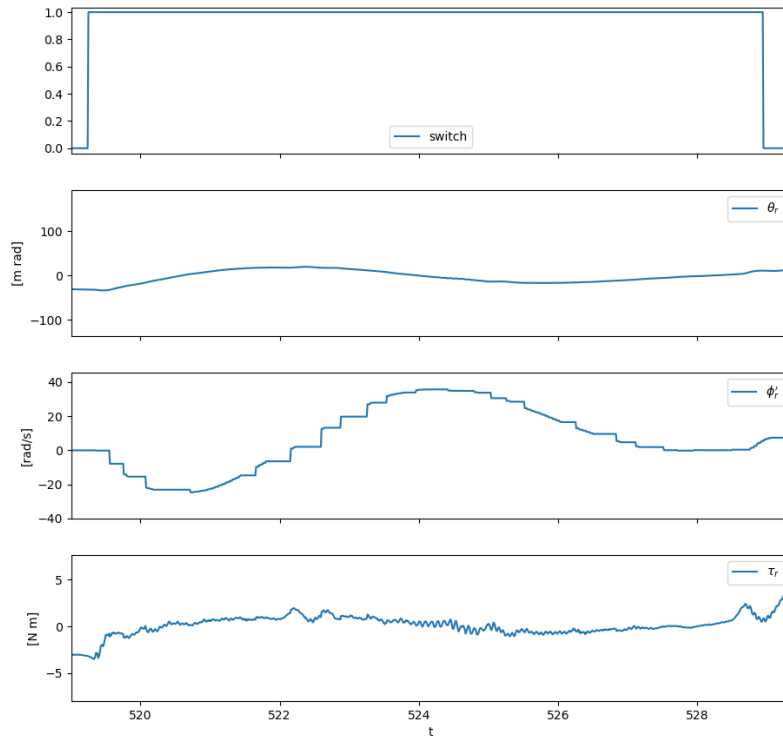
vinklarna enhjulingen kan kompensera för i rörelse inte nödvändigtvis matchar vinklarna i stillastående läge.

Tabell 7.2: Vinklar roll-regleringen klarar av att stabilisera från i stillastående läge

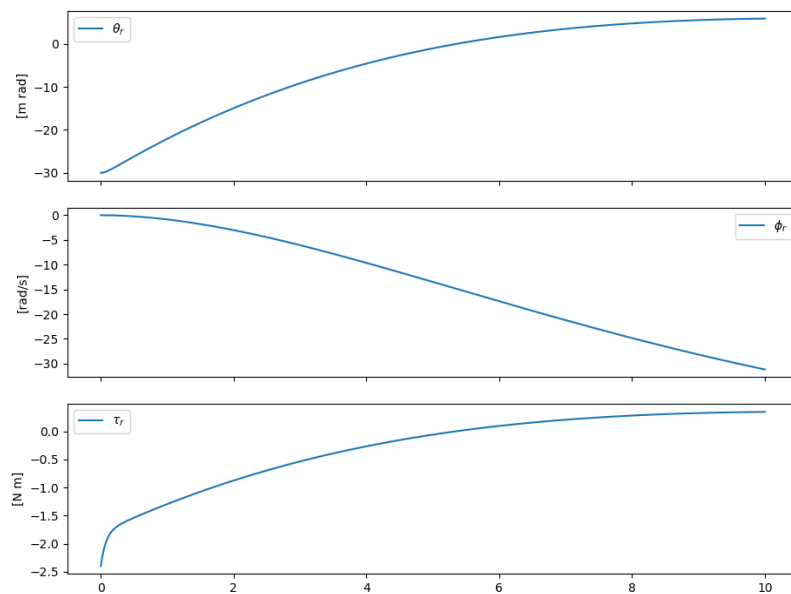
$\theta_r$ [mrad]	$\theta_r - \theta_{r,0}$ [mrad]	Klarar att stabilisera?
20	60	Nej
10	50	Nej
0	40	Nej
-25	15	Nej
-27	13	Nej
-30	10	Ja
-35	5	Ja
-40	0	Ja
-50	-10	Ja
-60	-20	Ja
-70	-30	Ja
-80	-40	Ja
-85	-45	Nej
-90	-50	Nej

Resultaten i tabell (7.2) visar att ett intervall på cirka 50 mrad  $\approx 3$  deg går att balansera med från stillastående läge. Som nämnt betyder inte det att enhjulingen i rörelse nödvändigtvis är begränsad till samma intervall, i figur (7.1) visas att styrsystemet kompenserar en vinkel på  $\theta_r \approx 4.3$  deg.

Intressant är att jämföra regulatorn i pitch från simuleringen och dess verkliga data. Då simulatoren använder en nollställd referensvinkel  $\theta_{r,0} = 0$  ritas datan upp med  $\theta_r \rightarrow \theta_r - \theta_{r,0}$  för att matcha simulering i denna aspekt.

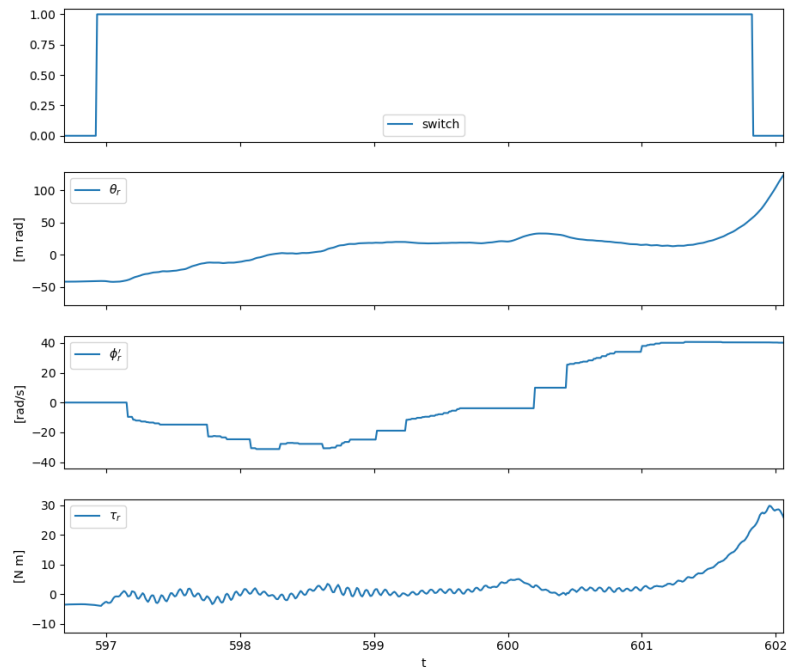


Figur 7.2: Roll-axeln rätar upp en stillastående lutning på  $-30$  mrad från referensvinkeln

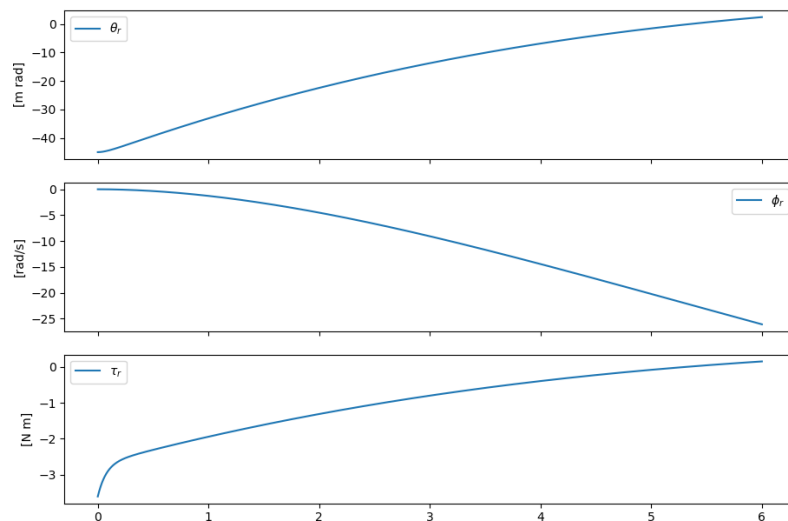


Figur 7.3: Simulering på att räta upp stillastående lutning på  $-30$  mrad

I scenariot med en initial lutning på  $-30$  mrad, med verklig data i figur (7.2) samt simulator i figur (7.3) går det att identifiera att vridmomentet i både simulator och verklighet verkar gå från cirka  $-3$  N m och snabbt avtar. Vinkeln av enhjulingen avtar direkt och närmar sig nollan på några sekunder. Vinkelhastigheten av roll-hjulet minskar mycket långsammare i simulatoren än i verkligheten, detta kan bero på effekter som friktion i hjulaxeln eller att motordrivaren inte klarar av att driva hjulet med riktigt lika mycket vridmoment som begärs. I både verkligheten och simulatoren stabiliseras enhjulingen helt och hållet.



Figur 7.4: Roll-axeln räta upp en stillastående lutning på  $-45$  mrad från referensvinkeln, men översvänger sedan



Figur 7.5: Simulering på att räta upp stillastående lutning på  $-45$  mrad

Med  $-45$  mrad lutning, med data i figur (7.4) och simulation i figur 7.5 iakttas en viss oscillation i styrsignalen. Detta kommer troligtvis från viss flexibilitet i prototypen. Här kommer enhjulingen till ett upprätt läge, med  $\theta_r = 0$ , men på grund av redan hög hastighet på roll-hjulet lyckas inte roll-regulatorn hålla den vinkeln, utan den översvänger och enhjulingen ramlar åt andra hållet. I simulationen verkar det som enhjulingen borde kunna stabilisera den vinkeln, men verkar även där börja närma sig hjulets maxhastighet.

### 7.3.2 Pitch

Inga kvantitativa tester har specifikt gjorts på pitch-axeln, delvis på grund av tidsbrist, delvis på grund av att en repeterbar testuppställning i pitch-led inte var enkelt att utföra. Som tidigare nämnt gjordes flera

försök till att hålla roll-axeln stilla utan att påverka pitch-axeln, men i slutändan kunde inte pitch-axeln isoleras utan att roll-regulatorn var med och kompenserade rörelse.

Kvalitativt så lyckas pitch-regulatorn balansera systemet utan stora problem. Den klarar av vissa laststörningar och kan överlag hålla enhjulingen nära stillastående i pitch-led.

Som nämnt i avsnitt 6.3 var ett stort problem med motordrivaren att varvtalsavläsningen inte hade tillräcklig upplösning. Varvtalsläsningen indikerade endast när motorn gick förbi en diskret pol, vilket innebar att vid väldigt låga hastigheter, som för pitch-hjulet i nära stillastående läge, det gick inte att få en bra uppdateringsfrekvens för hjulhastigheten. Detta uppenbarar sig under körningen; det går att identifiera hur den yttre P-regulatorn i kaskadregulatorn rycker till varje gång motorn går över en pol.

## 7.4 Filtrering och tillståndsuppskattning

Från testerna går det att konstatera att ett komplementfilter är ett bättre val än kalmanfilter. Kalmanfiltret visade sig vara väldigt svårt att implementera och resulterade i att projektet stannade upp. Första testet med komplementfilter gav lovande resultat direkt och efter justeringar av filtertiden  $\tau_f$  visade det sig vara tillräckligt bra för att styrsystemet skulle kunna balansera enhjulingen. En filtertid på  $\tau_f = 1$  gav bra resultat. En jämförelse mellan kalmanfiltret och komplement filtret gav att komplement filtret gav högre känslighet och framförallt lägre drift än kalmanfiltret enligt figur (6.1)

## 7.5 Styrbarhet med fjärrkontroll

Ett enkelt test där en fjärrkontroll kopplades in i kontrollsystemet för att variera  $\phi_{p,ref}$  utfördes. Enhjulingen klarade att köra framåt och bakåt över en sträcka på cirka 15 m plan asfalt utan att ramla. Hastigheten kontrollerades med fjärrkontrollen mellan  $\pm 0.7 \text{ rads}^{-1}$ , vilket motsvarar en markhastighet på cirka  $\pm 0.2 \text{ ms}^{-1}$ . Inga mätningar gjordes för att verifiera hur nära hastigheten hölls, men uppskattningsvis höll den rätt storleksordning av hastighet. Enhjulingen hade svårigheter att komma över små uppförsbackar, vilket slutligen fick den att ramla. Vid högre hastigheter för  $\phi_{p,ref}$  kunde inte regulatorn hålla systemet stabilt. Detta verkade till synes vara mer på grund av en hastigare acceleration än en högre topphastighet.

# Kapitel 8

## Slutsats och Diskussion

I detta kapitel kommer projektets arbete att diskuteras respektive vilka slutsatser som kan dras av arbetet.

### 8.1 Konstruktion

Enhjulingens konstruktion har varit relativt problemfri. Arbetet var mer fokuserat på att validera en självkörande enhjuling som koncept än att ta fram en slutgiltig produkt. Därför har prototyp tillverkningen fokuserat på att få fram prototyper som visar på att idén är genomförbar och kanske inte prototyper som är perfekta. Utifrån denna valideringsståndpunkt har det tagits fram prototyper som visar att det är genomförbart att konstruera en självkörande enhjuling.

Arbetsgången snabb prototypframtagning har fungerat relativt väl för att få fram fungerande prototyper. Det som gjort att det fungerat väl är att det har fokuserats på att testa framför att tänka sig till hur det ska bli. Att göra prototyper för testning av individuella delmål har gjort att gruppen har stött på problem tidigt som hade kommit senare i arbetsgången och kunnat lösa dessa efter hand. Det har också varit bra för att få information snabbt om hur en viss del i enhjulingen beter sig, ta lärdom och jobba vidare efter det.

Det som har fungerat mindre bra i snabb prototypframtagning är att det inte gick särskilt snabbt att ta fram en ny prototyp. Prototyperna som framtoogs var stora och materialen som användes var svårhanterliga. Detta gjorde att det krävdes mycket planering för varje prototypbygge. Detta kombinerat med att tillgängligheten i prototyplabbet var begränsad till tre stycken fyra timmars pass i veckan gjorde att prototypframtagningen var relativt långsam. Överlag var dock snabb prototypframtagning en god idé att följa och gruppen bedömer att det har varit ett bra arbetssätt för detta projekt.

### 8.2 Filter

Filtreringen av sensorvärden är troligtvis vad som varit det mest utmanande och komplext för detta projekt. Det går att konstatera att beslutet om att försöka implementera ett kalmanfilter var fel, för dåligt kompetens inom område gjorde att projektet underskattade vilken insats och tid det skulle krävas för att få ett filter som fungerade på ett tillfredsställande vis. Samtidigt var det ett beslut som gav projektet en god insikt, att inte underskatta de enklare lösningarna innan de har testas.

Filtreringen gav också goda insikter i skillnaden mellan en simulerad miljö och verkligheten, det är relativt lätt att skapa något som fungerar i simulerad miljö där allt är optimalt och avskärmat från yttre störningar. Även om gruppen tidigt var medveten om att det kommer finnas en viss mätosäkerhet i sensorernas mätvärden, så underskattade gruppen svårigheten i att förstå vad som påverkade mätosäkerheten. Gruppen underskattade också svårigheten i att skapa ett system som kunde producera samma resultat på ett repeterbart sätt, troligtvis på grund av avsaknaden i vilka bakomliggande faktorer som påverkade resultatet.

### 8.3 Reglersystem

Likt filtren är en regulator någonting som väldigt enkelt kan funka i simulering men vara mycket svårare i verkligheten. Faktorer som friktion och sensorbrus spelade definitivt roll, och var någonting gruppen från början var medvetna om skulle göra verkligheten svårare. Dock underskattades effekter av parameterfel,

som till exempel vikter, vridmoment och längder som alltid har viss osäkerhet, samt problem från och ickelinjäriteter i motordrivaren. Att ta hänsyn till alla dessa faktorer i simuleringen hade inte varit realistiskt — störningar som friktion är väldigt svårmodellerade, och problem i motordrivaren är väldigt svåra att förutsäga. Trots att simuleringen var till stor hjälp för att utveckla reglersystemet så behövde i slutändan systemet kalibreras rejält i verkligheten för att hitta fungerande parametervärden.

Likt filtrena användes till en början ett reglersystem som var bortom vår kompetens. Att använda optimal kontroll visade sig vara väldigt beroende av att sensorvärden och parametrar hade exakt värde, vilket inte systemet i verkligheten hade, och att felsöka systemet blev väldigt svårt då gruppledammarna inte hade erfarenheten att veta vad som kunde gå fel. Att förenkla systemet till en kaskadregulator kanske försämrade den optimala prestandan som teoretiskt kunde uppnås, men i utbyte så balanserade systemet och kunde kalibreras mycket lättare.

## 8.4 Användning i testmiljö

Eftersom målet med enhjulingen är att utöka mängden testfordon till ADAS tester är det relevant att diskutera hur väl prototypen skulle fungera på en testbana. För att utvärdera detta jämförs vår prototyp med Chalmers självkörande cykel med samma mål [25]. Elektroniken på prototypen är fäst på enhjulingen på ett kontrollerat och ordnat men inte skyddat vis. En krock från fel vinkel kan därför innebära att hårdvara förstörs. Enhjulingen har även nuvarande ett beroende av ett flertal fysiska sladdar kopplade till prototypen. För att vara redo för en testbana kan det vara en bra ide att samla all elektronik i en krocksäker låda likt [25], samt att göra systemet självständigt från externa sladdar med ett batteri och en trådlös sändare. Vidare är en viktig del att prototypen är enkel att interagera med, även här kan inspiration tas från cykeln där en tydlig av- och påknapp samt e-stop monterats.

Enhjulingens vikt kombinerat med vassa kanter kan även innebära en fara för testare och fordon, speciellt vid krock. Här kan prototypen istället efterlikna Chalmers vidareutveckling på cykeln där en ny cykel konstrueras för att vara så lätt och krockvänlig som möjligt [26].

En del av arbetets syfte var att besvara på huruvida enhjulingen kunde balansera likt en människa. Den huvudsakliga skillnaden i hur en människa balanserar en enhjulingen är att en människa kan skifta sitt masscentrum när hen sitter på enhjulingen genom att omfördela sin kroppsamma, någonting den prototyp inte gör alls. I pitch-led balanserar prototypen enbart genom att driva hjulet framåt och bakåt, en människa driver både hjulet med hjälp av pedaler men lutar sig också fram och bak för att bättra reglera i pitch-led. I roll-led arbetar denna prototyp helt olikt en människa, att agera reaktionshjul är ingenting en människa har förmåga att göra. Om positionen av hjulet kan hållas upprätt nominellt, går det att hävda att det agerar mer likt en människa — en människa kan inte luta sig allt för långt ut från centrum, vilket att hålla en viss position av hjulet skulle kunna emulera. Detta är ingenting detta arbete utforskat, och lämnas som uppgift till kommande arbeten.

Slutligen kan det vara relevant att kunna placera en mänskoliknande docka på enhjulingen. Detta är inte möjligt i nuläget och en utvärdering kring vad som krävs kring detta är nödvändigt.

## 8.5 Rekommenderad uppföljning

Det ursprungliga målet med projektet var en manövrerbar enhjuling som repeterbart kunde följa förprogrammerade vägar. Detta uppnåddes inte i detta projekt, vilket innebär den huvudsakliga uppföljningen med projektet skulle vara att uppfylla detta mål.

Vid sidan av detta mål finns också många mindre problem som skapade friktion i arbetssättet. Målet med en manövrerbar enhjuling skulle kunna uppnås utan att fixa dessa problem, men arbetet görs mer effektivt av att rätta dessa.

### 8.5.1 Uppföljningar för att få en manövrerbar enhjuling

Första rekommenderade steget är att få enhjulingen att kunna manövrera i en linje med hjälp av pitch-hjulet. Enhjulingen kan i nuläget nätt och jämt köra fram och bakåt i väldigt låg hastighet, och den klarar varken stora accelerationer eller stora laststörningar som vind eller uppförsbackar. Detta innebär att den i allmänhet inte är stabil nog för att återkopplas till en referensgenerator som ger en väg att följa. Att åtgärda detta kan kräva ett mer avancerat reglersystem än kaskadregulatorn som används för detta arbete, men att kalibrera det nuvarande systemets parametrar till perfektion kan också vara en möjlighet. Att utföra optimal kontroll som beskrivits i avsnitt 4.1.1, tillsammans med ett fungerande och

bättre kalmanfilter skulle antagligen leda till ett stabilare, pålitligare och mer styrbart system, men detta kräver mer avancerad förståelse för kontrollsystem och reglerteknik.

Nästa steg blir att möjliggöra svängning kring yaw-axeln, någonting som kan lösas på två radikalt olika sätt: avancerad gemensam reglering i pitch- och roll-led, alternativt ett dedikerat hjul i yaw-led.

Den förstnämnda varianten går ut på att luta enhjulingen innan och under sväng vid körning för att hitta ett jämviktsläge i svängen. Att luta åt sidan i roll bör få enhjulingen att svänga i den riktningen, och att under sväng ge rätt pitch-hastighet i relation till roll-vinkeln kan ge ett stabilt lutande system i svängen. När svängen är utförd behöver enhjulingen sedan räta på sig med hjälp av roll-hjulet. Detta kräver antagligen ett kraftigare reaktionshjul i roll-led samt en mycket mer avancerad och välkalibrerad roll-reglering. Detta skulle emulera mänsklig styrning bättre än alternativet och kräver inte särskilt stora fysiska ändringar på prototypen utan kan antagligen utföras med mindre hårdvaruändringar.

Det andra varianten fungerar genom att kraftigt accelerera eller retardera ett tredje hjul kring yaw-axeln. Detta är modulerat i avsnitt 3.1.4 och kommer kräva ett kraftigt yaw-hjul. Detta kommer innebära stora fysiska ändringar av prototypen för att kunna montera hjulet, vilket gör prototypen både tyngre och kräver antagligen i sin tur ett kraftigare roll-hjul för att kompensera denna massa i roll-led. En tredje motordrivare behövs för att reglera detta hjul. Det är oklart om VESC kommer kunna ge nog med effekt till hjulet för att övervinna friktionen mot marken. Dock krävs mindre kraft för svängning under körning.

## 8.5.2 Andra rekommenderade uppföljningar

Prototypen väger 20.5 kg. En möjlig förbättring kan därför vara att minska vikten på chassit. Detta gör prototypen säkrare, mer lämplig för användning i testmiljö och kan innebära lägre krav på tröghetsmoment för reaktionshjulet. Antingen kan tunnare aluminium och stål användas, eller kan hål placeras på strategiska platser i chassit för att minska vikten. En stressanalys skulle behöva göras för att se vart vikten kan minskas, samtidigt som chassit är starkt nog att klara vikten av enhjulingen och alla krafter som tillkommer under styrning, och potentiellt i en krock mot marken. Massan skulle också kunna omfördelas för att minska tröghetsmomentet av enhjulingen i relevanta led. Genom att sätta massan närmare enhjulingens svängpunkt minskas mängden vridmoment motorer behöver ge ut för att vrida på chassit.

Arbetsättet för att kalibrera sensorer och regulatorparametrar är väldigt långsamt. Var gång en ny mängd värden ska testas behöver enhjulingen stängas av, sättas i boot-läge (se bilaga B), ny mjukvara med de nya parametervärdena behöver kompileras och laddas upp till mikrokontrollern vilket tar tid och efter detta behöver igen enhjulingen startas om, denna gång i kör-läge. När enhjulingen startas om, eller är i boot-läge balanserar den ej, vilket innebär att den behöver balanseras av någon, till exempel en gruppkamrat som håller i enhjulingen, alternativt ett tak den hängs upp i. Här skulle processen förenklas och snabbas upp något enormt om nya parametervärden kunde skickas till systemet trådlöst, till exempel via Bluetooth. ST har en serie mikrokontrollers STM32WB som har stöd för Bluetooth Low Energy. Med detta system skulle en dator kunna ladda upp nya parametervärden under körning, alltså medans enhjulingen fortfarande balanserar.

Loggdata har alltid varit en svårighet att hantera på ett bra sätt. Då logg-data skickas över USB behöver en dator vara fysiskt inkopplad till den körande enhjulingen, vilket inte är perfekt då USB-kontakter kan vara ömtåliga och USB-sladden lätt sträcks ut när enhjulingen kör. Förutom detta är det lite jobbigt att hantera loggdatan då det bara är en ström av sensorvärden och uppskattade parametervärden. Att lista ut vilken körning en viss loggserie hörde till i efterhand är svårt, speciellt om det ska synas till en film eller liknande. Det första problemet skulle kunna lösas genom att antingen spara logg-data lokalt på mikrokontrollern, eller kanske på ett extert minneskort som ett flash-minne eller USB-sticka, så det går att senare ladda över det på en dator. Här blir minneskortets kapacitet ett problem då logg-data kan bli många megabyte stora. Det skulle också gå att använda Bluetooth för att streama data till en dator trådlöst istället för trådbundet för att också slippa en USB-sladd.

Det andra problemet skulle kunna förbättras genom att ha en realtidsklocka som daterar alla loggar i hög precision. Med hjälp av detta skulle det vara mycket lättare att synka data till film i efterhand, men också att veta vilken tidpunkt en loggserie kommer ifrån. För detta skulle ett litet batteri behövas på kretskortet för att hålla igång klockan när enhjulingen är avstängd. Realtidsklockor brukar just av denna anledningen dra väldigt lite effekt, och sekundärbatteriet skulle kunna laddas upp när huvudbatteriet driver systemet.

Ett bättre krocksäkerhetssystem skulle med fördel kunna utvecklas. Krockkudden som monterats på denna prototyp har räddat den många gånger från ett annars katastrofalt fall, men tecken som böjd metall börjar långsamt visa sig. Dessutom har enhjulingen en gång fallit på en trotoarkant vilket ledde till att elektroniken fick en smäll. Detta går i linje med att reducera enhjulingens vikt, då en lägre vikt

innebär mindre krafter under krock.

# Kapitel 9

## Tack

Vi skulle vilja Michal Dzialak, ETA:s ordförande 2022-2023, för att vi fick använda ETA:s lokal som lagringsutrymme för enhjulingen och att använda ETA:s verkstad och instrument för utveckling av arbete. Han har även varit till stor hjälp under tester av enhjulingen, samt under inspelning av material till kandidatarbetespresentationen.

Vi tackar vår handledare Anders Grauers för givande konversationer, uppmuntran och vägledning.

Ytterligare ett tack riktar vi till Reine Nohlborg och Jan Brage på prototypplabbet för hjälp och vägledning vid framtagandet av prototyper.

# Litteratur

- [1] Kyle Fortune. *Advanced Driver Assistance Systems: How adas car tech is put to the test*. Mars 2022. URL: <https://www.autoexpress.co.uk/features/354927/advanced-driver-assistance-systems-how-adas-car-tech-put-test>.
- [2] *ADAS explained: Technology to improve vehicle safety*. Okt. 2019. URL: <https://www.aa.co.nz/cars/motoring-blog/adas-explained-technology-to-improve-vehicle-safety/>.
- [3] Euro NCAP. *AEB Cyclist*. URL: <https://www.euroncap.com/en/vehicle-safety/the-ratings-explained/vulnerable-road-user-vru-protection/aeb-cyclist/>. (accessed: 05.15.2023).
- [4] Sandra Tavakoli och Björn Forsman. *Robotcykel gör trafiken säkrare för cyklister*. URL: <https://www.chalmers.se/sv/institutioner/e2/nyheter/Sidor/Robotcykel-g%C3%B6r-trafiken-s%C3%A4krare-f%C3%B6r-cyklister.aspx>. (accessed: 01.23.2023).
- [5] Ye Ding, Joshua B. Gafford och Mie Kunio. "Modeling , Simulation and Fabrication of a Balancing Robot". I: 2012.
- [6] URL: <https://www.youtube.com/watch?v=xNgPIqj5k>.
- [7] Jacob Miller. "Understanding Euro NCAP - Setting the Standard for AEB Performance Validation". I: 2020. URL: [https://www.autovision-news.com/wp-content/uploads/2020/04/EuroNCAPWhitepaper\\_VSI-Labs.pdf](https://www.autovision-news.com/wp-content/uploads/2020/04/EuroNCAPWhitepaper_VSI-Labs.pdf).
- [8] Transportstyrelsen. *Nollvisionen*. Dec. 2022. URL: <https://www.transportstyrelsen.se/sv/vagtrafik/statistik/olycksstatistik/statistik-over-vagtrafikolyckor/nollvisionen/>.
- [9] Shutong Li m. fl. "Accelerometer-Based Gyroscope Drift Compensation Approach in a Dual-Axial Stabilization Platform". I: *Electronics* 8 (maj 2019), s. 594. DOI: 10.3390/electronics8050594.
- [10] Elsäkerhetsverket. *ELSÄK-FS 2017:2 — Elsäkerhetsverkets föreskrifter och allmänna råd om elinstallationsarbete*. 2017. URL: <https://www.elsakerhetsverket.se/om-oss/lag-och-ratt/foreskrifter/elsak-fs-2017-2/>.
- [11] URL: <https://www.pygame.org/docs/>.
- [12] URL: <https://docs.python.org/3/library/ctypes.html>.
- [13] R.W.H. Sargent. "Optimal control". I: *Journal of Computational and Applied Mathematics* 124.1 (2000). Numerical Analysis 2000. Vol. IV: Optimization and Nonlinear Equations, s. 361–371. ISSN: 0377-0427. DOI: [https://doi.org/10.1016/S0377-0427\(00\)00418-0](https://doi.org/10.1016/S0377-0427(00)00418-0). URL: <https://www.sciencedirect.com/science/article/pii/S0377042700004180>.
- [14] Youngjoo Kim och Hyochoong Bang. *Introduction to kalman filter and its applications*. Nov. 2018. URL: <https://www.intechopen.com/chapters/63164>.
- [15] Alex Becker. *Online kalman filter tutorial*. URL: <https://www.kalmanfilter.net/default.aspx>.
- [16] *Tuning kalman filter to improve state estimation*. URL: <https://www.mathworks.com/help/fusion/ug/tuning-kalman-filter-to-improve-state-estimation.html>.
- [17] Alex Becker. *Online kalman filter tutorial*. URL: <https://www.kalmanfilter.net/covextrap.html>.
- [18] Parag Narkhede m. fl. "Cascaded Complementary Filter Architecture for Sensor Fusion in Attitude Estimation". I: *Sensors* 21.6 (2021). ISSN: 1424-8220. DOI: 10.3390/s21061937. URL: <https://www.mdpi.com/1424-8220/21/6/1937>.
- [19] Benjamin Vedder. *Communicating with the VESC using UART*. URL: <http://vedder.se/2015/10/communicating-with-the-vesc-using-uart/>.
- [20] *Gyroscope*. Mars 2023. URL: <https://www.britannica.com/technology/gyroscope>.

- [21] Chris Woodford. *How accelerometers work: Types of accelerometers*. Jan. 2023. URL: <https://www.explainthatstuff.com/accelerometers.html>.
- [22] *MPU-6000/MPU-6050 Register Map and Descriptions*. RM-MPU-6000A-00. Rev. 4.2. InvenSense. Aug. 2013. URL: [https://product.tdk.com/system/files/dam/doc/product/sensor/motion-inertial/imu/data\\_sheet/mpu-6000-datasheet1.pdf](https://product.tdk.com/system/files/dam/doc/product/sensor/motion-inertial/imu/data_sheet/mpu-6000-datasheet1.pdf).
- [23] Benjamin Vedder. *The VESC motor control firmware*. URL: <https://github.com/vedderb/bldc>.
- [24] Benjamin Vedder. *bldc\_uart\_comm\_stm32f4\_discovery*. URL: [https://github.com/vedderb/bldc\\_uart\\_comm\\_stm32f4\\_discovery](https://github.com/vedderb/bldc_uart_comm_stm32f4_discovery).
- [25] Guanzheng Wen och Jonas Sjöberg. "Lateral Control of a Self-driving Bike". I: *2022 IEEE International Conference on Vehicular Electronics and Safety (ICVES)*. 2022. DOI: 10.1109/ICVES56941.2022.9986548.
- [26] Leonardo Chouha och Edin Smajic. *Design of light, crash-worthy, self-driving bike*. URL: <https://odr.chalmers.se/server/api/core/bitstreams/7a990d35-8de6-49f1-8c6d-23a4c916156a/content>. (accessed: 05.15.2023).

# Bilaga A

## Komponentlista

Alla digitala filer för kretsschema och PCB layout finns tillgängligt vid <https://github.com/jawedhakimi/Kandidatarbete-Autonom-Enhjuling>. De är utvecklade i Kicad 7.

Tabell A.1: Komponentlista på PCB

Reference	Value	Qty
C100, C101	10u	2
C104	10nF	1
C108	1uF	1
C200, C301, C302	4u7	3
C201-C205, C212	100n	6
C206	1u	1
C207, C306	10n	2
C208, C209	2u2	2
C210, C211	12p	2
C300	0u22	1
C303-C305, C307	0u1	4
C308	2n2	1
Cp102	0u47F/50V	1
Cp103	2u2F/50V	1
Cp105	3u3F	1
Cp106	0u1	1
Cp107	22u/25	1
D100	D_Zener_Small	1
D101, D103	RED	2
D200	Yellow	1
D201	B	1
D202	G	1
D203	R	1
D_Sch100	D_Schottky_Small	1
F100	500mA	1
H1-H4	MountingHole_Pad	4
HMC5883	HMC5883L	1
J1, J100	Screw_Terminal_01x02	2
J2	USB_C_Receptacle_USB2.0	1
J3	Conn_01x06	1
J4, J207, J208	Conn_01x05	3
J200	Conn_02x05_Odd_Even	1
J201, J203-J205	Conn_01x04	4
JP100	Jumper_3_Open	1
L100	220u	1
L200	10u	1
MMA8451	MMA8451QR1	1
MPU6050	MPU-6050	1
Q100	BSP 171	1

Tabell A.2: Komponentlista på PCB

Reference	Value	Qty
R100, R109, R200, R220, R300	10k	5
R102, R107, R204-R207	1k	6
R103	127k	1
R104	14k	1
R105	47k	1
R106	4M7	1
R108	499k	1
R201	0	1
R202, R203	2k2	2
R208, R209	5k1	2
R301, R302	100	2
R303, R405-R407	22	4
RV1	R_Potentiometer_Small	1
SW1	SW_Push	1
SW200	BOOT0	1
U1	LM5017MR	1
U101	LVR 3.3V	1
U200	STM32F405RGT6	1
U201	USBLC6-2SC6	1
Y200	16Mhz	1

# Bilaga B

## Källkod

All källkod finns publicerad på [https://github.com/karlsandreas/autonomous\\_unicycle](https://github.com/karlsandreas/autonomous_unicycle). En översiktlig beskrivning av koden följer:

### B.1 Simulering

Simuleringen, som beskrivs i avsnitt 3.2 bor i mappen `simple-2d-sim/`. Filen `sim.py` beskriver själva simuleringen, `render.py` ritar det grafiska gränssnittet och `debugparser.py` läser logg-data från enhjulingen. För att köra koden behövs biblioteken `pygame`, `numpy` samt `pandas` (dessa är paketerade i `shell.nix` för er som använder pakethanteraren Nix).

### B.2 STM32

I mappen `STM32/` finns all kod som kör på mikrokontrollern, samt regulatorn som kan användas i simuleringen.

`UniController` är kod som körde på det äldre kontrollkortet, `Unicontroller-OnBoard` kör på kretskortet som senare designades och bör användas. Denna kod utvecklas lättast via `STM32CubeIDE` för att kunna kompilera med ST:s mjukvaru- samt HAL-bibliotek. `UniController-OnBoard/Core/Src/` innehåller all källkod vi skrivit, `UniController-OnBoard/Core/Inc/` innehåller motsvarande header-filer.

Notera att filer som `kalman_filter.c` och `regulator.c` som ligger i `STM32`-mappen laddas av simuleringen (de behöver kompileras till `shared-libraries` separat). Dessa används INTE i mjukvaran som laddas upp till STM-kortet, utan de finns i `UniController-OnBoard/Core/Src/ctrl`.

För att ladda upp koden till mikrokontrollern används lättast `STM32CubeProgrammer`, och kan antingen laddas upp över USB med mikrokontrollern i `BOOT`-läge, eller med hjälp av en `ST-Link` kopplad till debug-headern på kretskortet. Notera att pinouten på `ST-Link` inte matchar pinouten på kretskortets header, utan kablar behöver kopplas manuellt mellan relevanta portar (`GND`, `3V3`, `SWDIO`, `SWCLK` samt `NRST`).

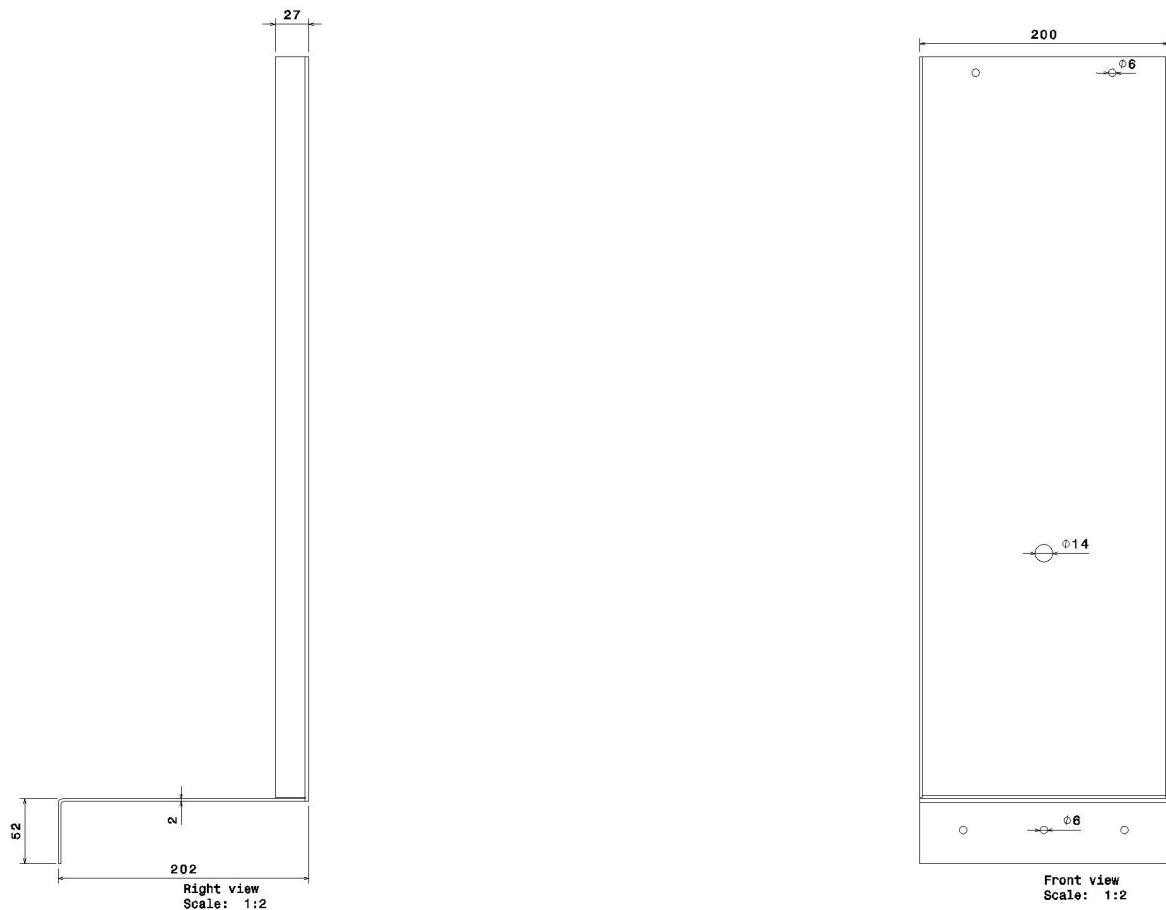
# Bilaga C

## Detaljritningar för prototyper

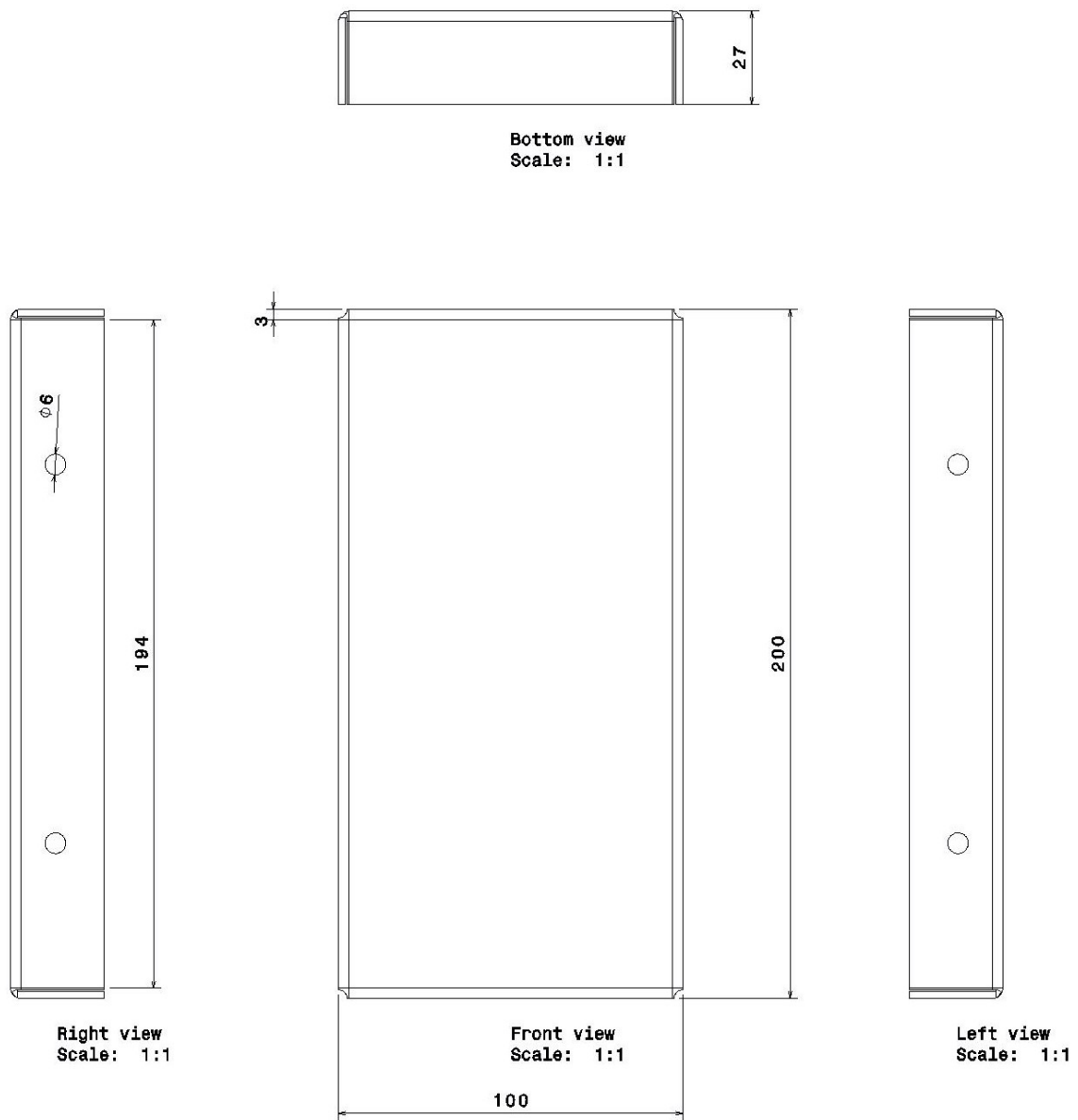
Samtliga ritningar för samtliga delar i samtliga prototyper.

### C.1 Prototyp 1

Ritningar för prototyp 1:



Figur C.1: Ritning för sidopanel i prototyp 1



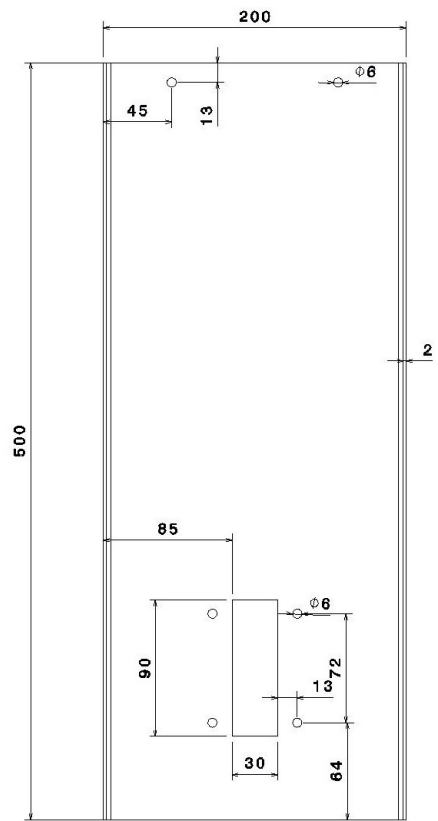
Figur C.2: Ritning för topplatta i prototyp 1

## C.2 Prototyp 2

Ritningar för prototyp 2:

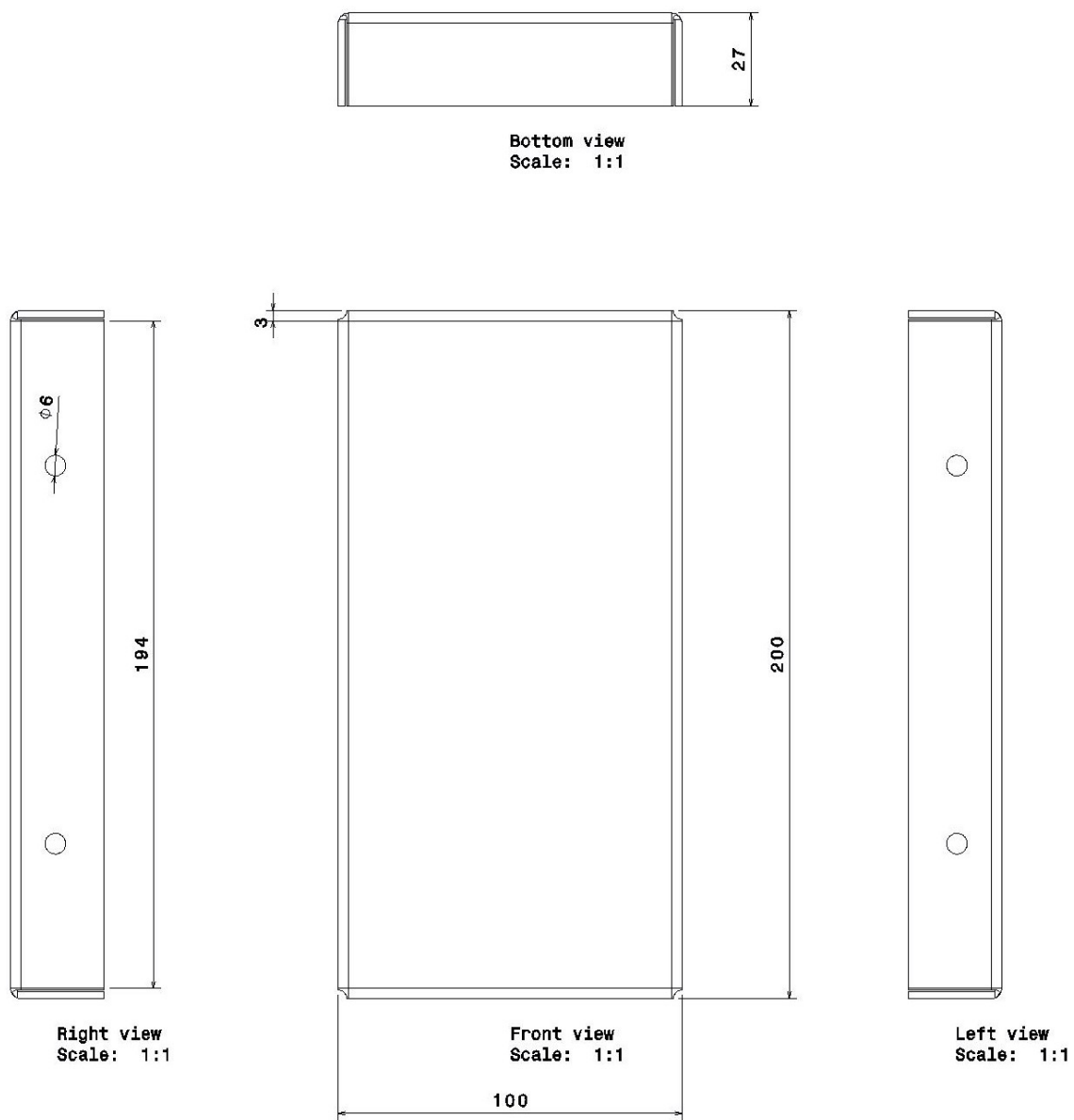


Right view  
Scale: 1:2

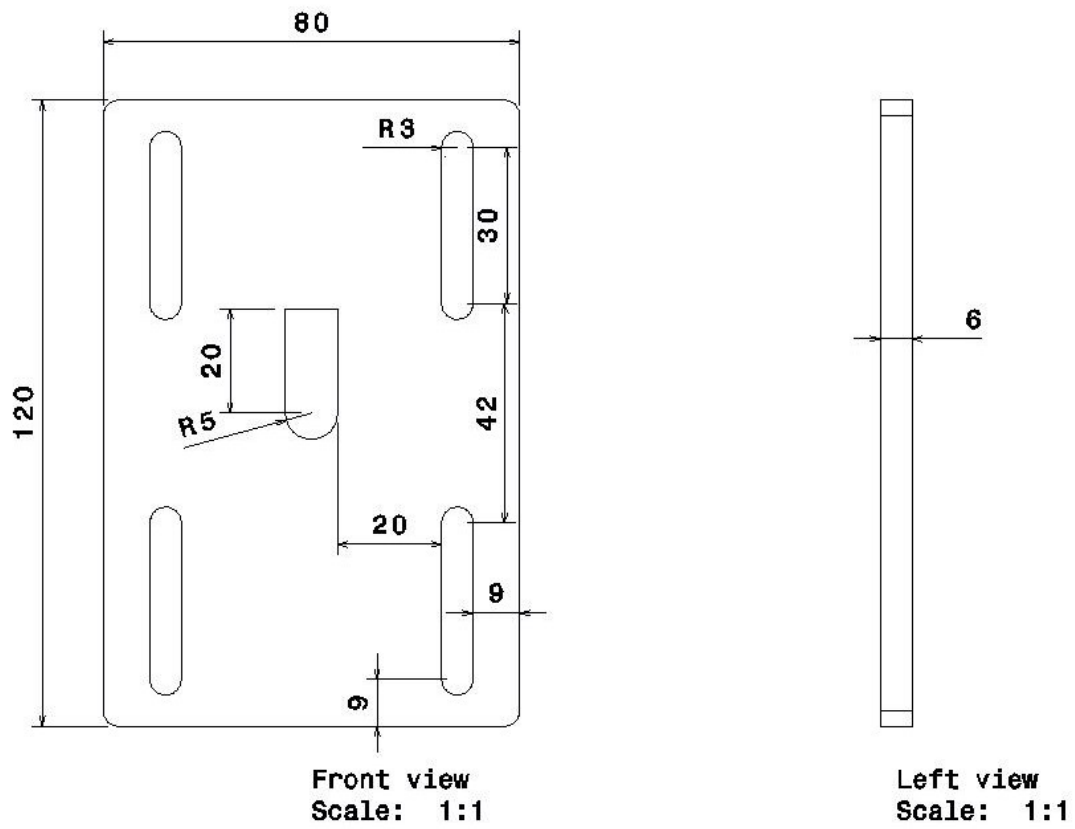


Front view  
Scale: 1:2

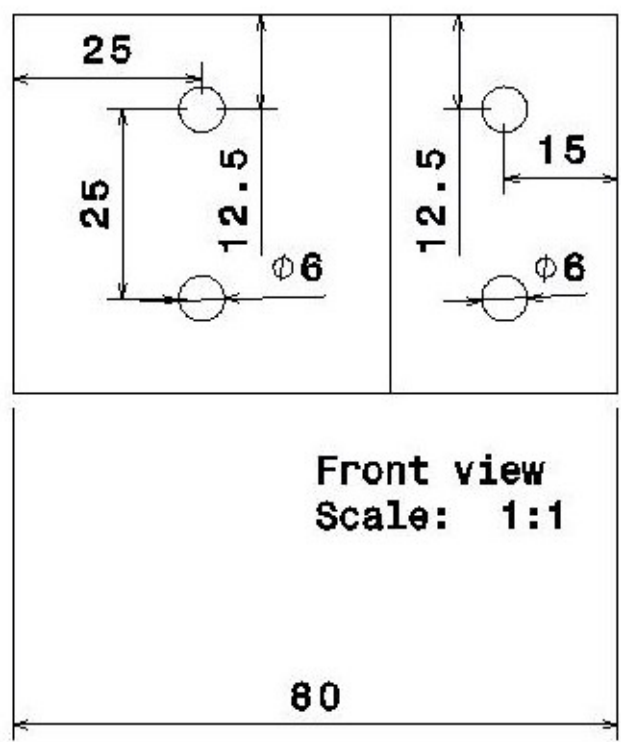
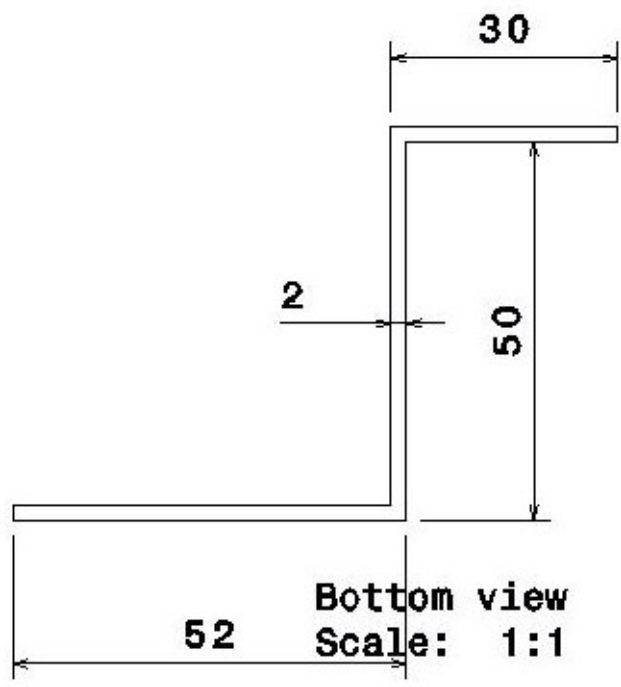
Figur C.3: Ritning för sidopanel i prototyp 2



Figur C.4: Ritning för topplatta i prototyp 2



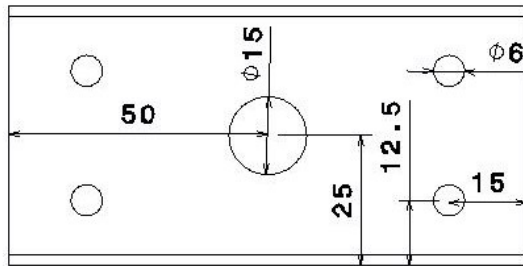
Figur C.5: Ritning för läsmkansim för drivande motor i prototyp 2



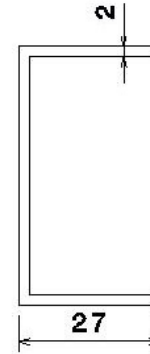
Figur C.6: Ritning för vinkeljärn i prototyp 2



**Bottom view**  
**Scale: 1:1**



**Front view**  
**Scale: 1:1**

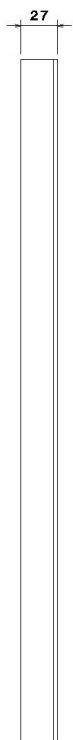


**Left view**  
**Scale: 1:1**

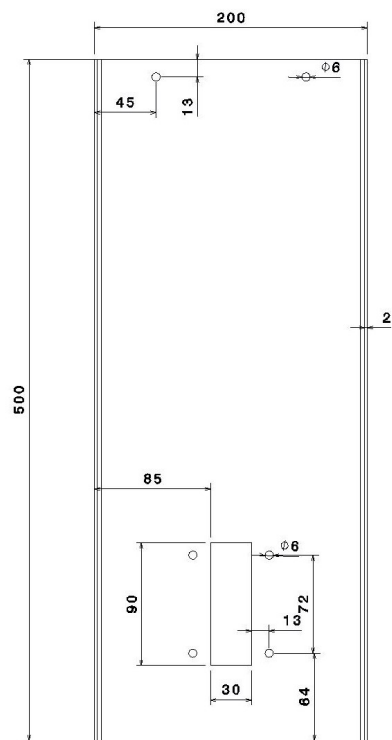
Figur C.7: Ritning för ujärn i prototyp 2

### C.3 Prototyp 3

Ritningar för prototyp 3:

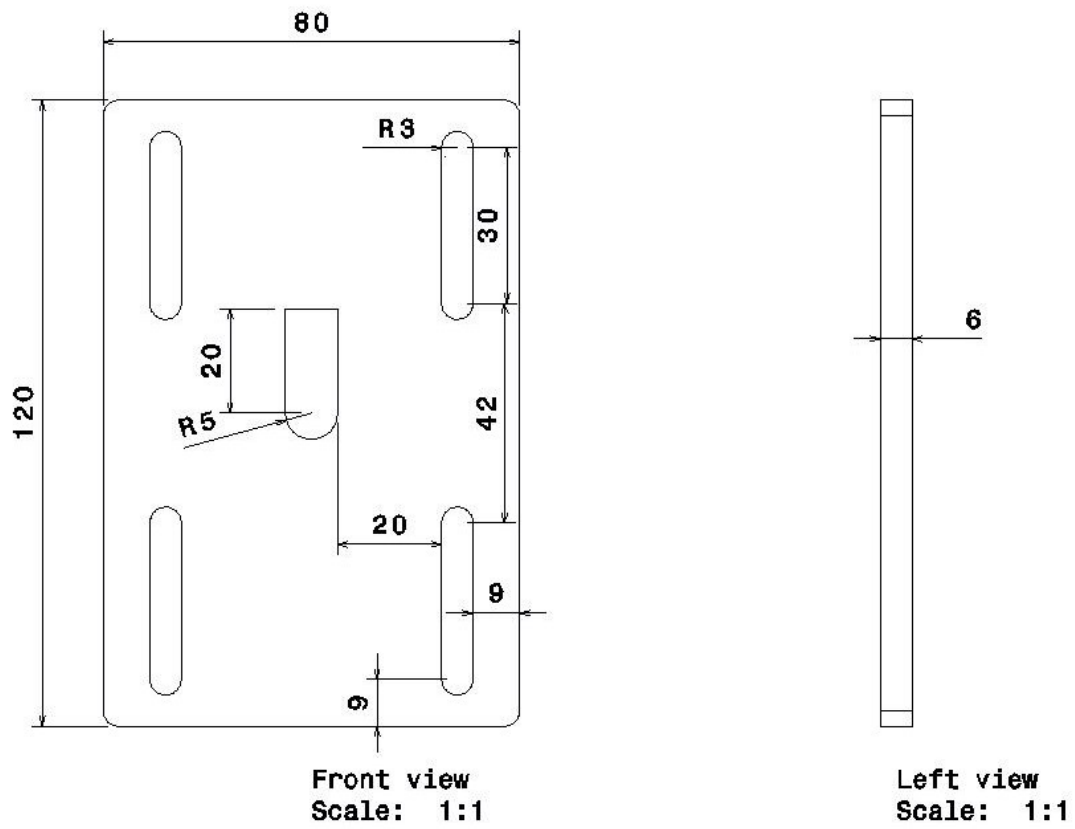


Right view  
Scale: 1:2

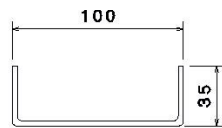


Front view  
Scale: 1:2

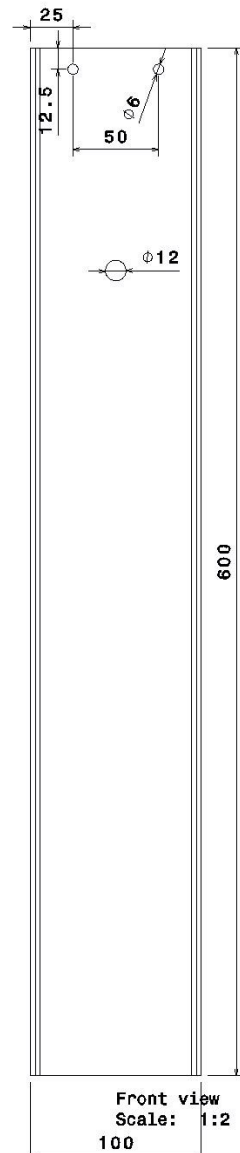
Figur C.8: Ritning för sidopanel i prototyp 3



Figur C.9: Ritning för läsmkansim för drivande motor i prototyp 3



Bottom view  
Scale: 1:2

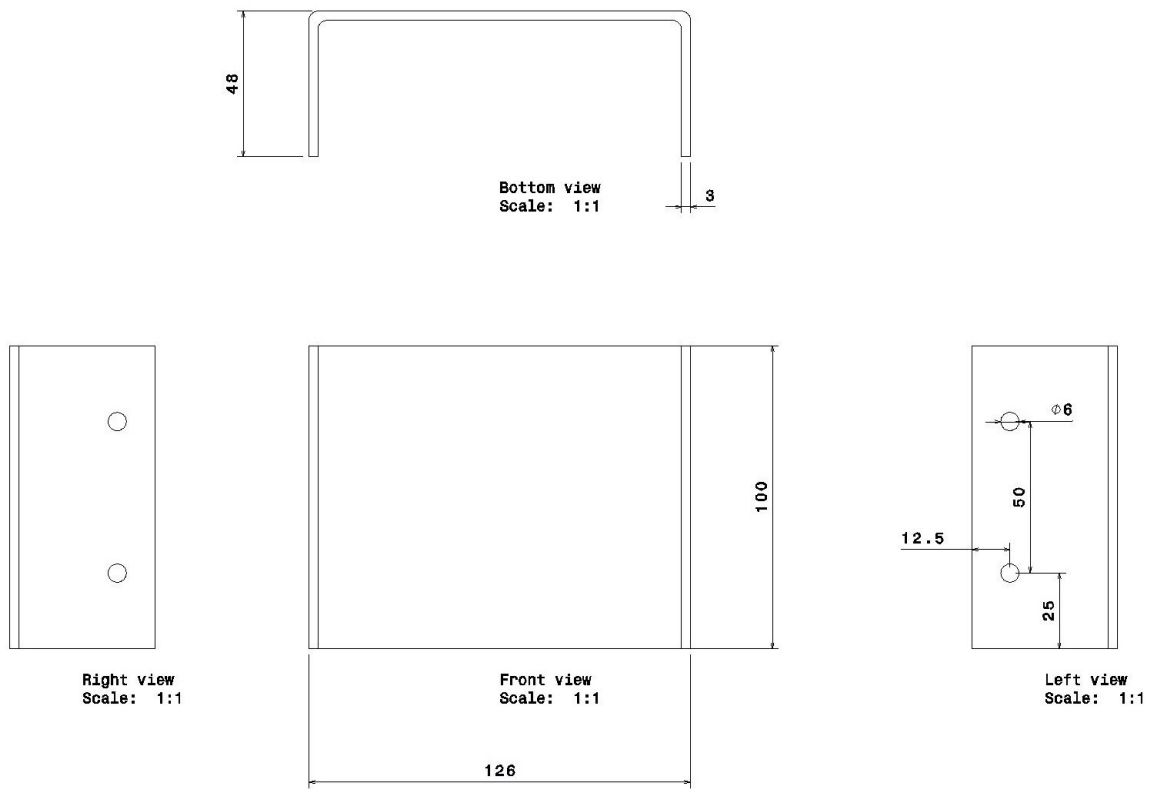


Front view  
Scale: 1:2  
100



Left view  
Scale: 1:2

Figur C.10: Ritning för motorfäste för rollmotor i prototyp 3



Figur C.11: Ritning för topplatta i prototyp 3

INSTITUTIONEN FÖR ELEKTROTEKNIK  
SYSCON  
CHALMERS TEKNISKA HÖGSKOLA  
Göteborg, Sverige 2023  
[www.chalmers.se](http://www.chalmers.se)



**CHALMERS**