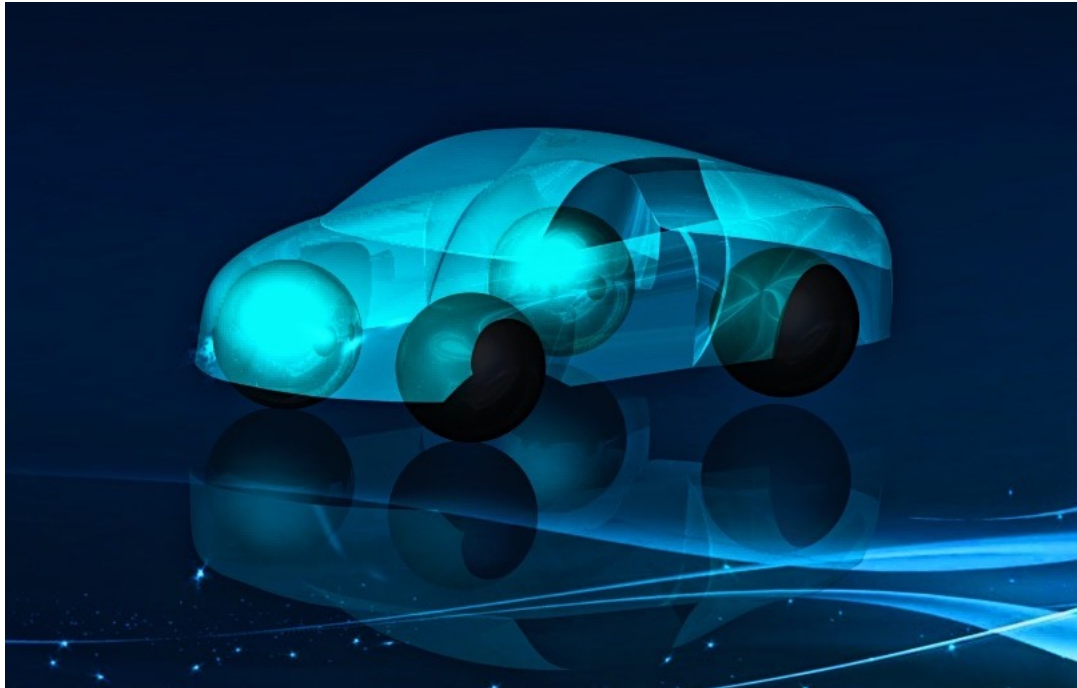




CHALMERS
UNIVERSITY OF TECHNOLOGY



SPHERiCAR

Ett kandidatarbete för konstruktion och utveckling av en småskalig bilprototyp med sfäriska hjul

Jasmine Azadani
Marcus Gideflod
Christoffer Hjalmarsson
Osvald Lindholm
Sara Roth
Lukas Wikman

Förord

Följande rapport behandlar ett kandidatarbete som utfördes våren 2019 för Elektroinstitutionen, avdelning för System- och Reglerteknik på Chalmers Tekniska Högskola i Göteborg. Projektet utfördes av sex stycken Civilingenjörstudenter inom Maskinteknik, Elektroteknik, Teknisk Fysik samt Automation och Mekatronik.

Projektgruppen vill börja med att tacka handledare Adrian Ilka för stöd under hela projektets gång samt uppmuntran till fortsatt utveckling av arbetet. Ett stort tack riktas även till Ignacio Bona som har varit till hjälp under arbetet med programmeringen av Sphero Bolt. Gruppen vill också tacka Göran Stigler, Reine Nohlborg och Jan Bragee i Chalmers prototypverkstad som har bidragit med stöd och rekommendationer under tillverkningen av prototypen. Ett tack riktas även till Rikard Karlsson för all hjälp vid inköp av komponenter till prototypen. Slutligen vill projektgruppen tacka examinatorn Nikolce Murgovski som har varit behjälplig med frågor kring rapporten.

Abstract

This report describes the work regarding the implementation of spherical wheels in cars. The purpose is partly to investigate difficulties and opportunities with this type of wheels, but also to show that today's construction with cylindrical wheels can be further developed. To investigate whether the implementation of spherical wheels in cars are possible, a small-scale prototype was constructed. This project does not include the development of spherical wheels due to the usage of four Sphero Bolts. In addition to the design of a prototype, a simulation was created in Simscape to investigate a line-following algorithm and control systems.

One of the results from this project was a solution for attaching spherical wheel to a small-scale prototype using "ball transfer units". With this construction it was also easy to disassemble the wheels from the chassis. Another result is the line-following algorithm that uses eight reflectance measuring sensors located under the car. With the help of the algorithm and the freedom of movement that spherical wheels entail, the car can follow a path while the car rotates arbitrarily around its own axis. This has probably never been developed before. The control system that has been developed gives the possibility of controlling speeds in both the x- and y-direction and specify which direction the car should be aimed at.

In summary, the project came up with several solutions that can be implemented to construct and regulate vehicles with spherical wheels. Nevertheless, to achieve a full scale implementation of the solutions there is still much work remaining.

Keywords: Spherical wheels, future car, line-following algorithm for spherical wheels, omnidirectional vehicles

Sammandrag

I följande rapport beskrivs arbetet kring implementation av sfäriska hjul i bilar. Syftet är dels att undersöka svårigheter och möjligheter med denna typ av hjul, men även att visa att dagens konstruktion med cylindriska hjul kan utvecklas. För att undersöka om implementationen av sfäriska hjul i bilar är möjligt konstruerades en småskalig prototyp. I projektet utvecklades inga sfäriska hjul utan fyra stycken Sphero Bolt användes som substitut. Utöver konstruktionen av en prototyp designades en simulation i Simscape för att undersöka implementationen av en banföljningsalgoritm samt reglersystem.

Ett av resultaten blev att det var möjligt att fästa sfäriska hjul med inre drivning på prototypen där infästningen innehåller så kallade kulrullar (ball transfer units). Med denna konstruktion var det även lätt att demontera hjulen. Ett annat resultat är banföljningsalgoritmen som använder sig av åtta stycken reflektansmätande sensorer under bilen. Med hjälp av algoritmen och den ökade friheten som sfäriska hjul medför kan bilen följa en bana samtidigt som bilen roterar godtyckligt kring sin egen axel, vilket troligtvis aldrig utvecklats innan. Reglersystemen som framtagits ger möjlighet till att styra hastigheter i både x- och y-led samt specificera i vilken riktning bilens för ska vara riktad vid körning.

Sammanfattningsvis kom projektet fram till flera lösningar som kan appliceras för att konstruera och reglera fordon med sfäriska hjul. För att åstadkomma en fullskalig implementation av lösningarna kvarstår dock mycket arbete.

Nyckelord: Sfäriska hjul, framtidens bil, banföljningsalgoritm för sfäriska hjul, omnidirektiva fordon

Innehåll

1	Introduktion	1
1.1	Mål och Syfte	2
1.2	Problembeskrivning	2
1.3	Avgränsningar	3
2	Kravspecifikation	4
3	Metodik	7
3.1	Konstruktion och Elektronik	7
3.2	Simulering	9
3.2.1	Modellering av Systemet	10
3.2.2	Dimensionering av Reglersystem	12
3.2.3	Justering av Reglersystemet	15
3.2.4	Banans Uppbyggnad i Simscape	16
3.2.5	Utveckling av Banföljningslogiken	18
4	Resultat	22
4.1	Fysisk Konstruktion och Modellering	22
4.2	Utfall av Simulering	25
4.2.1	Verifiering av Reglersystem	25
4.2.2	Prestation av Banföljningsalgoritmen	27
5	Diskussion	30
5.1	Utvärdering av Konstruktion	30
5.2	Evaluering av Simulering	31
5.2.1	Eventuella Problem med Reglersystemet	31
5.2.2	Problem och Användningsområden för Banföljningsalgoritmen	33
5.3	Samhälleliga och Etiska Aspekter	33
6	Slutsatser	35
6.1	Rekommendationer till Fortsatt Arbete	35
	Referenser	37
	Appendix	i
A	Elimineringsmatris	i
B	Undersökning av Kommunikation med Sphero Bolt	ii
B.1	Felmeddelande från sphero.js SDK	iii
B.2	Felmeddelande från SpheroBB8-python API	iv
B.3	Konsolmeddelande från Gobot API	iv
C	Modeluppbyggnad i Simscape	v
C.1	Hjul och Infästningsmodel	v
C.2	Prototypsmodel	v
C.3	Anglemeter	v
C.4	Helhetsmodel (vänster)	vi
C.5	Helhetsmodel (höger)	vii
C.6	Helhetsmodel (hel)	viii

	C.7	Superpositonering	ix
	C.8	Speedometer	ix
	C.9	Sensormodel	x
D		Detaljritningar	xi
	D.1	Infästning	xi
	D.2	Chassi	xii
E		Matematiska Beräkningar	xiii
	E.1	Matematisk Modell för Translation och Rotation	xiii
	E.2	Maximal Acceleration Beroende på Friktion	xiv
F		Övrigt	xv
	F.1	Projekttes SPHERiCAR	xv

1 Introduktion

Sedan bilen uppfanns har dess hjul roterat kring en axel som styrts med en ratt. Detta är en välbeprövad teknik som gör det lätt för föraren att manövrera sitt fordon. Efter ett sekel av teknisk utveckling går nu bilindustrin mot datorstyrda bilar vilket öppnar upp för en mer komplicerad teknik, där varje hjul kan styras individuellt [1]. Med den tekniska utvecklingen som bakgrund kan nya och bättre koncept för styrning därmed vara realiserbara.

Ett alternativ för att förbättra bilens manövreringsförmåga är att ersätta de konventionella cylindriska hjulen med sfäriska för att därmed möjliggöra körning i alla riktningar. Ökad frihet medför större möjligheter till anpassad körning under olika omständigheter. En bil med total frihet i planet skulle medföra avsevärt mindre svängradier och följaktligen skulle inte lika stort stadsutrymme behöva avsättas för bilarna. Den ökade manövreringen kommer även kunna bidra till en mjukare körning som ger en mer komfortabel resa för passagerarna. Dessutom skulle fickparkering med de sfäriska hjulen tillåta fordonet att sidledes köra in i parkeringsfickan. Manövreringen med sfäriska hjul skulle också kunna vara en stor fördel inom lagermiljöer för att på så vis kunna optimera arealen med mer utrymme för hyllor och varor.

Utmaningarna med sfäriska hjul är dock många och dessa måste övervinnas för att detta koncept ska kunna realiseras. De största tekniska utmaningarna för implementation på en storskalig bil är infästningen och drivandet av hjulen samt en styralgorithm. Innan sfäriska hjul kan kommersialiseras i bilindustrin behövs ett nytt styrsätt utvecklas då den traditionella ratten inte kommer klara av styrningen i tre frihetsgrader. För att utnyttja de sfäriska hjulens fulla potential skulle bilen behöva vara autonom, med tanke på svårigheterna för en mänsklig förare att manövrera en bil som inte är bunden att ha fronten fram.

Idén med sfäriska hjul är inte helt ny utan har bland annat varit med i populärkulturen. Exempelvis hade alla fordon i filmen "I, Robot" (2004) sfäriska hjul [2]. Bortsett från filmer så har några studenter på San Jose State University försökt utveckla en motorcykel med sfäriska hjul, dock tycks de inte ha slutfört projektet [3]. Även däckföretaget Goodyear har insett de möjligheter som sfäriska hjul medför och har därför utvecklat ett sfäriskt koncepthjul kallat Eagle 360 Urban Concept som presenterades i Geneve Motorshow 2017 [4]. Det är härifrån inspirationen till detta kandidatarbete kommer.

1.1 Mål och Syfte

Målet med projektet är att konstruera en småskalig prototyp av en bil med sfäriska hjul. Till prototypen ska en hastighetsregulator samt en rotationsregulator utvecklas som möjliggör körning av en användare. Till sist ska även prototypen autonomt kunna följa en tydligt utmärkt bana oberoende av hur prototypen roterar kring z-axeln. Speciellt ska prototypen kunna följa banan utan att rotera alls, vilket är omöjligt för en vanlig bil med cylindriska hjul.

Syftet är att dels undersöka svårigheterna och möjligheterna med sfäriska hjul på fordon och dels att utmana uppfattningen om hur ett fordon är uppbyggt. Genom att bygga en prototyp med sfäriska hjul är förhoppningen att fler får upp ögonen för konceptet och att detta leder till fortsatt utveckling inom området. Projektet önskar också att kunna bidra med lösningar och idéer på vissa problem som sfäriska hjul medför.

1.2 Problembeskrivning

För att utveckla en småskalig prototyp med sfäriska hjul som dessutom kan följa en bana oberoende av hur prototypen roterar kring sin egen z-axel, behövs ett antal delproblem lösas. Dessa listas nedan.

- Ett av de essentiella problemen som ska lösas är hur hjulen ska fästas i prototypens chassi. Detta på grund av att den konventionella lösningen inte tillåter sfäriska hjul att rotera fritt eftersom ett fäste med en axel endast tillåter rotation kring axeln. Målen för infästningen är så låg friktion som möjligt samt att hjulen ska vara avtagbara.
- Ett annat centralt problem är kontakten mellan en användare som kör prototypen fjärrstyrt och hjulen på prototypen. För detta behövs ett kommunikationssystem utvecklas där användaren skickar en signal om hur prototypen ska köra till en lokal dator på prototypen. Den lokala datorn ska därefter skicka unika styrsignaler till vardera hjul för att prototypen som helhet ska köra så som användaren preciserat. I detta problem ingår dels att signaler ska kunna skickas mellan de olika komponenterna och dels att de skickade signalerna ska ge rätt beteende för prototypen.
- Vidare ska prototypen kunna följa en tydligt utmärkt bana då prototypen roterar godtyckligt kring z-axeln. Detta färdssätt skapar nya problem som behöver lösas som berör antal och placering av sensorerna. Dessutom ska en algoritm som utnyttjar sensordata till att följa banan utvecklas. Målet är att banföljningen ska vara så allmän som möjligt.
- För att förenkla framtagningen av en banföljningsalgoritm samt reglersystem ska även en simulering av prototypen göras.

1.3 Avgränsningar

Begränsningarna för projektet är följande:

- Budget på 5 000 kronor samt fyra stycken Sphero Bolts
- Arbetet pågår under 14 veckor på halvtid

Begränsningarna leder till att följande avgränsningar sätts:

- Prototypen ska ha fyra stycken Sphero Bolts som hjul då vi begränsar oss ifrån att konstruera egna sfäriska hjul. Sphero Bolts ger en bra efterliknelse till hur ett Sfäriskt hjul skulle bete sig.
- Utformningen av prototypen görs enbart utefter funktionalitet och för de krav som ställts på prototypen, se kravspecifikationen 1.
- Tillräcklig friktion mellan hjul och mark förutsetts då projektet inte kommer att innefatta omfattande undersökningar kring friktion.
- Vid banföljning avgränsas projektet från att använda sensorer mer avancerade än reflektansmätande sensorer som monteras under bilen, på grund av begränsning i budget och tiden det skulle kräva att implementera.
- Banan vid banföljning ska vara tydligt utmärkt och inte korsa sig själv för ökad simplicitet och för att undvika problematiska specialfall vid utvecklandet av banföljningsalgoritmen.

2 Kravspecifikation

För att få en tydlig bild över vad projektet eftersträvar gjordes, med det givna projektförslaget som grund, se bilaga F.1, en kravspecifikation för prototypen, se figur 1. I kravspecifikationen gjordes uppdelningen krav och önskan, förkortat K och Ö, på prototypens egenskaper utefter vad som prioriteras i projektet. Ett krav är något som är av yttersta vikt för prototypen medan en önskan är något som ska genomföras i mån av tid och resurser. Eftersom målet med projektet endast är att bygga en småskalig prototyp av bilen görs en del förenklingar gällande kraven på prototypen. Vissa krav som vanligtvis ställs på en färdig produkt så som lagkrav, livslängd och effektivitet för storskalig produktion tas ej med i denna kravspecifikation.

Kraven fördelades mellan prototypens funktioner, prestanda, dimensioner och vikt på de olika delarna av prototypen. Ett av kraven för prototypens funktioner är att prototypen ska kunna färdas med fronten i godtycklig orientering, det vill säga prototypen ska inte vara bunden till att ha fronten i körriktningen. Ett annat funktionskrav är att prototypen dels ska kunna rotera kring z-axeln med prototypens masscentrum stilla samt när dess masscentrum färdas. Det är dessa funktioner som särskiljer en bil med sfäriska hjul gentemot en vanlig bil och är således av högsta prioritet.

Vidare sätts utifrån projekttesen, se appendix F.1, ett krav på att prototypens hastighet och körriktning ska kunna fjärrstyras av en användare under färd. Ytterligare ett funktionskrav är att prototypens hjul ska sitta fast under färd men ska kunna tas av vid behov. Att hjulen sitter fast i prototypen innebär att hjulen inte kan trilla av oplanerat, vilket är en förutsättning för säker körning. Möjlighet till hjulbyte är också en viktig aspekt för underhåll. Till sist sätts även ett funktionskrav att prototypen endast med hjälp av sensorer ska kunna följa en bana samtidigt som den godtyckligt roterar kring sin egen z-axeln. Detta kan exempelvis vara fördelaktigt för autonoma linjeföljande truckar (Automated Guided Vehicles) i lagermiljö om dessa skulle utrustas med sfäriska hjul.

De första prestandakrav som ställs på prototypen är att den ska klara att köra i 0.6 m/s , vid rotation ska masscentrum förflyttas mindre än 20 mm per varv och prototypens färdriktning ska inte avvika mer än 5% vid translation. Att precis 0.6 m/s sattes som kravhastighet var lite slumpartat men bygger på att topphastigheten för en Sphero Bolt är 2 m/s och de väger cirka 200 g [5]. Med fyra bollar och en karossvikt på cirka 1.5 kg (inklusive elektronik) tycks 0.6 m/s vara görbart. Förflyttningen av masscentrum per rotationsvarv sattes till 20 mm eftersom detta är ungefär $1/10$ av prototypens bredd och noggrannare än så tycktes överflödigt. Avvikelsen vid translation får maximalt vara 5% och denna gräns sattes eftersom det verkade vara rimligt och en större avvikelse skulle göra prototypen svårstyrd.

För hastighetshållning sätts ett prestandakrav att överslängen ska vara mindre än 10 % och att insvängningstiden ska understiga fyra sekunder. På samma vis sätts krav för orienteringsföljningen, det vill säga i vilket håll prototypen är riktad åt. För detta väljs att överslängen ska vara mindre än 40 % och att insvängningstiden inte får överstiga fyra sekunder. För banföljning sätts ett krav att vinkeln mellan prototypens korriktning och banan vid en raksträcka maximalt får vara 25°. Det sista prestandakravet som ställs på prototypen är att den autonomt ska klara banföljning med kurvor med räta eller trubbiga innervinklar. Kravet följs av ett önskemål med viktfaktor fem att prototypen ska klara kurvor med alla olika innervinklar. Vidare sattes även ett önskemål med viktfaktor tre att prototypen ska klara av att köra 0.8 m/s.

Prototypen önskas till viss del efterlikna en vanlig nerskalad bil därmed ska proportionerna mellan chassits längd och en konventionell bil vara ungefär 1:10, eftersom hjulen, Sphero Bolt, förhåller sig till den skalan [5]. Prototypens bredd bestäms av att alla komponenter ska få plats, därmed blir förhållandet mellan bredd och längd ungefär 0.6. Detta gav ett chassi med längd 400 mm och bredd 240 mm. Hjulhusen där hjulen är monterade ska vara i ungefär samma storleksordning som själva hjulen för att på så vis uppta minimalt utrymme. Tjockleken på chassit ska vara så tunn som möjligt för att minimera vikten men ändå tillräckligt tjock för att ge styvhet och stadga till konstruktionen. Detta gav en önskad tjocklek på omkring 1.5 mm. Dimensionerna för bilen viktades med en faktor fyra i urvalsmatrisen.

För att belastningen på hjulen inte ska bli för stora sätts krav på chassits vikt. Dessa är att chassit ska väga mindre än 0.5 kg men helst mindre än 0.3 kg och att infästningsanordningarna mellan hjul och chassit ska väga mindre än 0.1 kg vardera men helst mindre än 0.085 kg. Vikterna sätts med tanke på att materialen som används troligtvis kommer vara plast och aluminium, storleken på prototypen samt för att kunna uppfylla prestandakraven och helst även prestandaönskningarna. De önskade vikterna för prototypens delar viktades med en faktor fyra.

För att verifiera huruvida slutprodukten kan leverera enligt kravspecifikationen testas de olika funktionerna, prestandakraven och vikterna på olika vis som kort nämns i kravspecifikationen under verifieringskolonnen. Om inte produkten hinner färdigutvecklas kan kraven i stället undersökas genom simuleringar.

Chalmers	Dokumenttyp Projekt	Kravspecifikation SPHERiCAR			
Utfärdare: EENX15-19-18		Skapad: 2019-02-16 Modifierad: 2019-04-09			
Kriterier		Målvärde	K/Ö	Vikt	Verifieringsmetod
Funktioner					
	Färdas i godtycklig riktning		K		Prototyp/simulering
	Reglera hastighet och riktning med hjälp av fjärrstyrning		K		Prototyp
	Autonom banföljning		K		Prototyp/simulering
	Rotera kring egen axel		K		Test med hjälp av prototyp
	Rotera under färd		K		Test med hjälp av prototyp
	Hålla hjulen i infästningen		K		Empirisk testning
	Hjulen skall kunna monteras på och av		K		Test med hjälp av prototyp
1. Prestanda					
1.1	Maximal hastighet	0.6 m/s	K		Uppmätning av tid & sträcka
1.2	Maximal hastighet	0.8 m/s	Ö	3	Uppmätning av tid & sträcka
1.3	Behålla färdriktning i minst 10 m	< 5 %	K		Uppmätning av avvikande sträcka
1.4	Behålla orientering vid godtycklig hastighet	< 5°	K		Uppmätning av avvikande vinkel
1.5	Behålla position vid rotation (kring egen axel)	< 20 mm	K		Uppmätning av avvikande sträcka efter rotation
1.6	Banföljning med svängar	90° - 180°	K		Prototyp/simulering
1.7	Avvikelse mellan körriktning och bana	< 25°	K		Prototyp/simulering
1.8	Banföljning med svängar	0° < θ	Ö	5	Prototyp/simulering
1.9	Hastighetsföljning översläng	< 10 %	K		Simulering
1.10	Orienteringsföljning översläng	< 40 %	K		Simulering
1.11	Insvängningstid hastighetsföljning	< 4 s	K		Simulering
1.12	Insvängningstid orienteringsföljning	< 4 s	K		Simulering
2. Storlek					
2.1	Chassi	400 x 240 x 1.5 mm	Ö	4	CAD-modell
2.2	Infästning	Ungefär lika stor som hjulen	Ö	4	CAD-modell
3. Vikt					
3.1	Chassi	< 0.5 kg	K		CAD-beräkningar
3.2	Chassi	< 0.3 kg	Ö	4	CAD-beräkningar
3.3	Infästning	< 0.1 kg	K		CAD-beräkningar
3.4	Infästning	< 0.085 kg	Ö	4	CAD-beräkningar

Figur 1: Kravspecifikation för prototypens krav och önskemål samt deras verifieringsmetoder

3 Metodik

I följande kapitel presenteras vilka metoder och processer som låg till grund för projektets resultat. Kapitlet är indelat i två huvuddelar som innefattar den fysiska konstruktionen av prototypen samt simulering av prototypen. Konstruktionen förklarar hur infästningen mellan hjul och chassi är konstruerad samt hur chassit är utformat. I simuleringsavsnittet beskrivs hur simuleringen av prototypen fungerar och även hur reglersystem och banföljningsalgoritm framtogs med hjälp av simuleringen.

3.1 Konstruktion och Elektronik

Arbetet för att utveckla en infästning initierades med att undersöka den befintliga marknaden för fordon och produkter med sfäriska hjul. Därefter genomfördes en idégenerering för dellösningar till infästningen, en organiserad session riktad till att generera nya idéer och koncept. De olika områdena som sessionen behandlade var: infästningens kontakt med hjulen, montering och demontering av hjul samt korrelation mellan infästning och chassi. De olika dellösningarna kombinerades sedan för att generera fullständiga lösningar, helhetslösningar.

Med idégenereringen och marknadsundersökningen som grund sammanställdes åtta koncept för att utvärderas noggrannare baserat på funktionalitet och realiserbarhet. Med hjälp av en elimineringsmatris, se appendix A, betygsattes de olika koncepten i syfte att identifiera det bäst lämpade konceptet. För att objektivt rangordna koncepten viktades de gentemot varandra utefter ett antal kategorier essentiella för prototypens funktionalitet. Ett exempel på en kategori och värdeskala som användes visas i tabell 1, där varje helhetslösning tilldelas ett värde baserat på hur väl kategorin uppfylls. Exemplet involverar tiden det tar att montera hjulen i infästningen. De kategorierna som ansågs ha störst inverkan på prototypens funktionalitet var: friktion i infästningen, vikt samt montering och demontering av hjulen.

Tabell 1: Värdeskala

Montering	
Tid [s]	Värde
< 30	5
30 – 60	4
60 – 100	3
100 – 600	2
> 600	1

Det konceptet med högst inbördes poäng utvecklades vidare på detaljnivå med CAD-modellering. I CAD modellerades även ett chassi utefter infästningens utformning med stöd för montering av tillhörande elektronik. Modellen nyttjades sedan till simulering av köregenskaperna och för att 3D-printa komponenter till konstruktionen av prototypen.

Elektroniken som ska monteras på prototypen ska upprätthålla kommunikationen mellan en användare och hjulen. Detta ska ske genom en lokal minidator som valdes till att vara en Raspberry Pi 3 model B+ som också skall hantera regleringen och banföljningen på prototypen. Denna valdes på grund av dess kapacitet att upprätthålla BLE-kommunikation (Bluetooth Low Energy) utan extra adaptrar, vilket är nödvändigt för kommunikation med hjulen [6]. Vidare stärktes valet av det breda utbudet av programspråkskompatibilitet. Till sensorer så valdes reflektansmätande sensorer på grund av avgränsningen i 1.3. Sensorerna är av sorten “SEN0017” och valdes på grund av dess låga pris, att de var digitala och att de var kompatibla med Raspberryn [7]. Som strömförsörjning till dessa komponenter kommer en powerbank att användas som valdes för dess låga pris samt goda egenskaper [8].

3.2 Simulering

För att göra en simulering valdes Simscape Multibody för Simulink i MATLAB. Detta på grund av tidigare kännedom av MATLAB och att det genom Chalmers var det lätt-tillgängligt. I detta program fanns det också bra möjligheter att visualisera systemet i 3D.

När simuleringen utvecklades var ett antal parametrar gällande konstruktionen fortfarande okända, därav skiljer sig prototypen i simuleringen lite mot den faktiska konstruktionen. Den stora skillnaden är framförallt att stötdämparna som är monterade på den reella konstruktionen, se figur 16, inte är med i simuleringsmodellen. I simuleringen bortses det även från den elektriska utrustningen som ska vara monterad på den fysiska konstruktionen, förutom en visualisering av var sensorerna sitter. Detta på grund av att det var osäkert exakt vilken elektronik som skulle användas och att detta ändå kunde ses som en störning. För vidare diskussion se kapital 5.2.1.

De värden som användes i simuleringen på olika parametrar visas i tabell 2. Parametrarna förklaras närmare när de används och värdena är tagna från uppskattningar, tabellvärden, beräkningar samt från konstruktionen.

Tabell 2: Värden i simuleringen

Hjulsfärens yttre radie	r	36.5 mm [5]
Hjulsfärens tjocklek	t	2.4* mm
Innertyngdens*** radie	r_i	31.5 mm
Densitet för chassit (aluminium)	ρ_{al}	2.70 g/cm ³ [9]
Densitet för infästningshusen (plast)	ρ_{pl}	1.25 g/cm ³ [10], [11]
Hjulsfärens massa	m_s	45* g
Innertyngdens*** massa	m_i	150* g
Konstruktionens**** massa	M	686** g
Konstruktionens**** tröghetsmomentet (z-axeln)	I	0.01** kg·m ²
Längd från chassicentrum till hjulcentrum	l	120 mm
Bredd från chassicentrum till hjulcentrum	b	100 mm
Radie från chassicentrum till hjulcentrum	R	156.2 mm

* Estimerade mått.

** Från numeriska beräkningar i Matlab.

*** Beskrivs närmare i kapitlet 3.2.1.

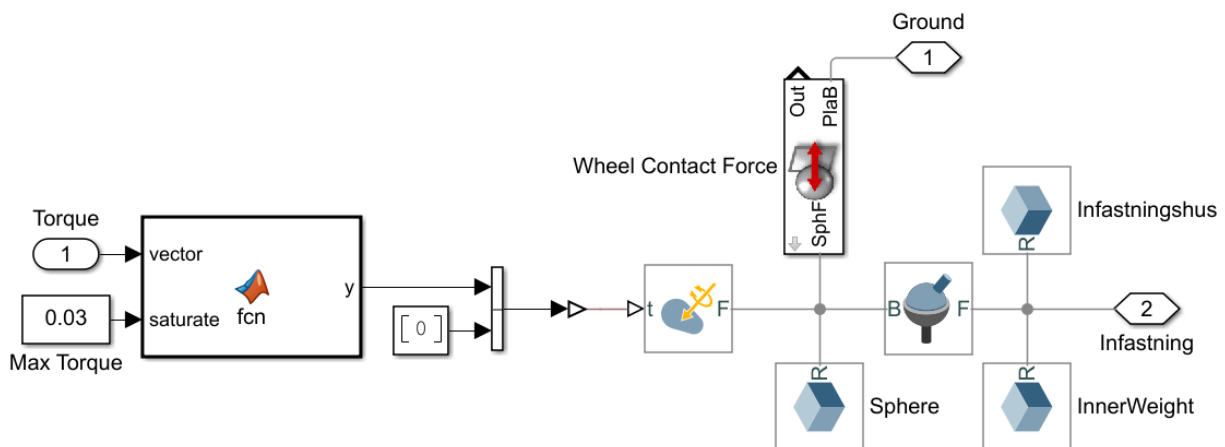
**** Konstruktionen innefattar både chassi och infästningar men inte hjul.

3.2.1 Modellering av Systemet

För att skapa en simulering behövdes först prototypen samt en omvärld byggas upp i Simscape. Detta gjordes genom att konstruera olika delsystem som sedan kopplades ihop för att skapa ett system som efterliknade verkligheten.

Det lägsta delsystemet var för ett hjul, Sphero Bolt, och tillhörande infästning, se figur 2. Delsystemet bestod av CAD-modellen för infästningshuset, se figur 17, och en ihålig sfär med ett halvklot inuti som hjul. Halvklotet modellerar hjulets inre massa, den som i de verkliga hjulen rör sig för att på så vis få hjulet att rulla. I modellen gjordes en förenkling som innebär att den inre massan i simulationen inte tilläts röra sig i förhållande till infästningshuset. Denna förenkling gjordes eftersom sfärens inre massa i verkligheten nästan är stationär i förhållande till den roterande sfären, det vill säga hjulet. I verkligheten regleras den inre massans position och påverkan på moment av Sphero Bolts egna reglersystem.

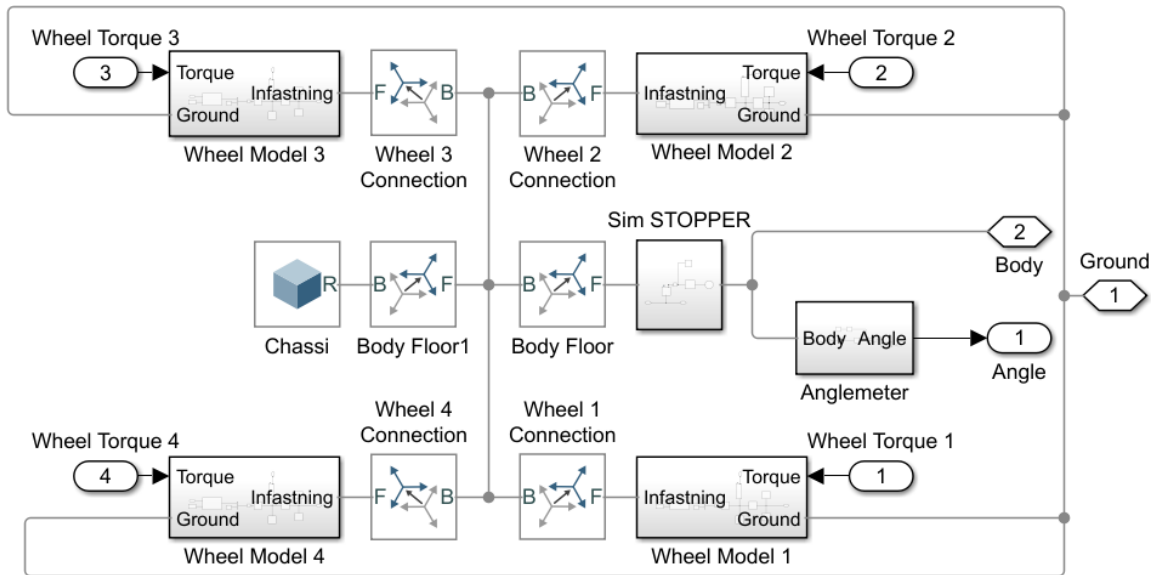
Translation i planet modellerades därefter med hjälp av en kontaktkraft mellan sfären och markytan. Detta görs med hjälp av ett fristående bibliotek kallat "Simscape Multibody Contact Forces".



Figur 2: Modelluppbyggnad av hjul och infästning i Simscape

Eftersom det efterfrågade momentet från reglersystemet på sfärerna inte får vara för stort, utan är begränsat till hur mycket moment den inre viktmassan kan producera, saturerades momentet vid 0.03 Nm. Momentet är en estimering utifrån uppskattade värden från det fysiska hjulet. $\tau = mgl \approx 150 \text{ g} \cdot 10 \text{ m/s}^2 \cdot 20 \text{ mm} \approx 0.03 \text{ Nm}$.

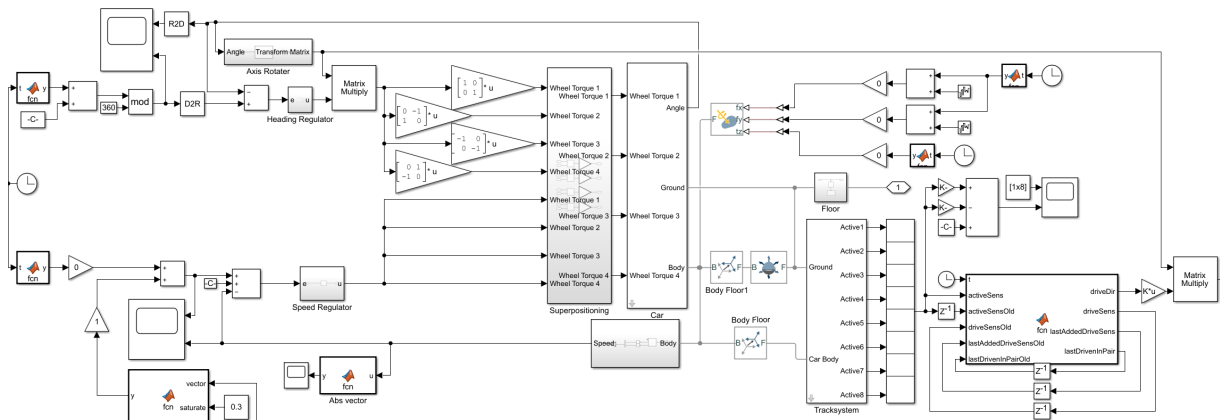
Prototypen är sedermera uppbyggd av en solid kropp (chassit) importerad från CAD, en tidigare version av figur 16. Chassit är sammankopplat med fyra stycken av det ovan beskrivna delsystemet för hjul och infästningshus, se figur 3.



Figur 3: Modelluppbyggnad av prototypen i Simscape

Prototypen är därefter kopplad till markytan med en 6-graders-frihet-fog (6-DOF-joint). Ifrån fogen avläses prototypens hastighet och orientering som sedan återkopplas in i ett regelsystem. Se kapitel 3.2.2 för mera detaljer kring regelsystemet. På prototypen implementeras även en motsvarighet till sensorernas funktion och dess signaler skickas vidare till ett funktionsblock som hanterar logiken för banföljning. Se kapitel 3.2.4 och 3.2.5 för mera detaljer om hur banföljningen är uppbyggd. Se figur 4 för en övergripande bild på simulationsmodellen i Simscape.

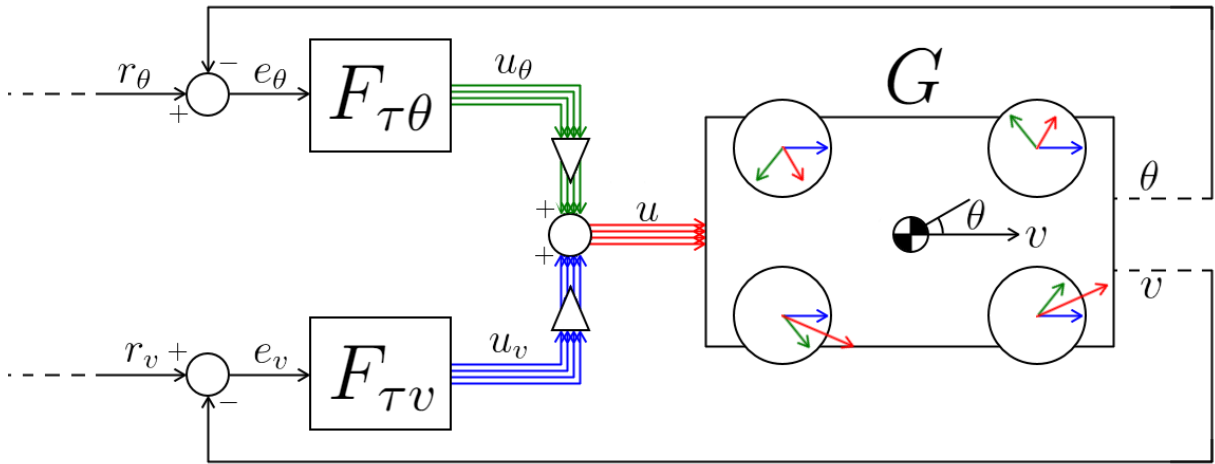
För alla delsystem och ytterligare figurer av simuleringsmodellen se appendix C.



Figur 4: Modelluppbyggnad av prototypen i Simscape

3.2.2 Dimensionering av Reglersystem

De mest vidspridda regulatorerna inom industrin är fortfarande de proportionerliga, integrerande och deriverande (PID) regulatorerna, tack vare deras enkelhet och goda karakteristik [12]. På grund av detta implementerades en PID-regulator vardera för reglering av prototypens orientering och hastighet. Deras styrsignaler superpositionerades med möjlighet för olika viktfaktorer beroende på vad som prioriteras. I simuleringen användes viktfaktorn 1 för båda styrsignalerna. Med orientering menas vinkeln, θ , som prototypens front har gentemot en godtyckligt dragen linje i det horisontella planet som inte förändras under körning. Det vill säga prototypens vinkel kring z-axeln. På grund av de extra friheterna som prototypen besitter i förhållande till en vanlig bil behöver inte orienteringen vara samma som körriktningen. Reglerprincipen visualiseras i figur 5.



Figur 5: En skiss över prototypens reglersystem. Notera att de blå vektorerna är styrsignaler från hastighetsregulatorn att köra högerut med en viss hastighet. Dessutom får prototypen en styrsignal från orienteringsregulatorn att rotera θ rad moturs vilket genererar de gröna styrsignalerna till vardera hjul. De röda vektorerna är en superpositionering av styrsignalerna från de båda regulatorerna.

För konstruktion av reglersystemen framtoqs kontrollorienterade modeller över prototypens translation och rotation med följande basantaganden:

- Tillräcklig friktion mellan hjulen och marken finns så att ingen slirning sker
- Prototypens masscentrum ligger i chassits geometriska centrum, mittemellan alla hjul
- Rullmotstånd och luftmotstånd kan försummas
- Innermassan antages vara stilla i förhållande till infästningshuset. Det vill säga att den inte roterar med hjulet

Detta ger då att Newtons andra lag för ett hjul blir enligt ekvation 1.

$$F = a \left(m + \frac{I}{r^2} \right) \quad (1)$$

F för kraft, a för acceleration, m för massa, I för tröghetsmoment och r för radie.

I modellen för prototypens translationshastighet i planet antogs även att alla hjulen kör i samma riktning och producerar samma vridmoment. Prototypen har fyra hjul och därför blir den totala kraft som driver prototypen framåt fyra gånger så stor som kraften från ett hjul, det vill säga $F = 4\tau/r$ där τ är vridmomentet från ett hjul. Med hjälp av formelsamling för tröghetsmoment skrevs ekvation 1 om till ekvation 2 med beteckningar från tabell 2 [13].

$$4\frac{\tau}{r} = \dot{v} \left(M + 4 \left((m_s + m_i) + \frac{I_s}{r^2} \right) \right) \quad (2)$$

$$I_s = \frac{2}{5}m_s \left(\frac{r^5 - (r-t)^5}{r^3 - (r-t)^3} \right) \quad (3)$$

τ för vridmoment från vardera hjul, v för hastighet och övriga konstanter ur tabell 2.

För orienteringsmodellen antogs utöver basantagandena att alla hjulen kör i perfekta cirkelbanor kring masscentrum med samma vridmoment. Rotationsrörelser ges av ekvation 4.

$$\tau = I\dot{\omega} \quad (4)$$

τ är moment kring rotationsaxeln, I är tröghetsmomentet och ω är vinkelhastighet

Ekvation 4 skrevs om med hjälp av ekvation 3, tröghetsmomentet för innermassan kring z-axeln samt Steiners sats, vilket gav ekvation 5 och 6 [13].

$$\frac{4R\tau}{r} = \left(I + 4 \left(I_s \left(1 + \frac{R}{r} \right) + I_i + (m_s + m_i)R^2 \right) \right) \ddot{\theta} \quad (5)$$

$$I_i = \frac{1}{5}m_i r_i^2 \quad (6)$$

τ för vridmoment kring xy-axlarna från vardera hjul,
 θ för vinkeln och övriga konstanter ur tabell 2.

Genom Laplacetransformering av ekvation 2 erhöles överföringsfunktionen från vridmoment till hastighet, då alla hjul producerar lika mycket vridmoment i samma riktning. Resultatet visas i ekvation 7. Likaledes Laplacetransformerades ekvation 5 vilket gav överföringsfunktionen från vridmoment till roterad vinkel. Även i detta fall produceras lika mycket vridmoment i alla hjul fast i sådan riktning att prototypen endast roterar. Denna överföringsfunktion visas i ekvation 8.

$$G_{\tau v}(s) = \frac{4r}{(Mr^2 + 4((m_s + m_i)r^2 + I_s))s} \quad (7)$$

$$G_{\tau \theta}(s) = \frac{4R}{\left(I + 4 \left(I_s \left(1 + \frac{R}{r} \right) + I_i + (m_s + m_i)R^2 \right) \right) rs^2} \quad (8)$$

Givet regulatorn i ekvation 9 och rimliga värden på ω_c , ζ , β och φ_m såsom i tabell 3 bestämdes τ och K_i för båda regulatorerna [14]. Därefter, när reglersystemet var dimensionerat beräknades K_p , K_i , K_d och N från nedanstående ekvationer. Dessa parametrar behövdes inför realiseringen i Simulink eftersom Simulinks implementering av PID regulatorer följer ekvationen 10. Se tabell 3 för slutliga värden.

$$F_{PID}(s) = K_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{1 + T_f s} \right) = K_i \frac{1 + 2\zeta s\tau + (s\tau)^2}{s(1 + s\tau/\beta)} \quad (9)$$

$$F_{PID}(s) = K_p + \frac{K_p}{s} + K_d \frac{N}{1 + N\frac{1}{s}} \quad (10)$$

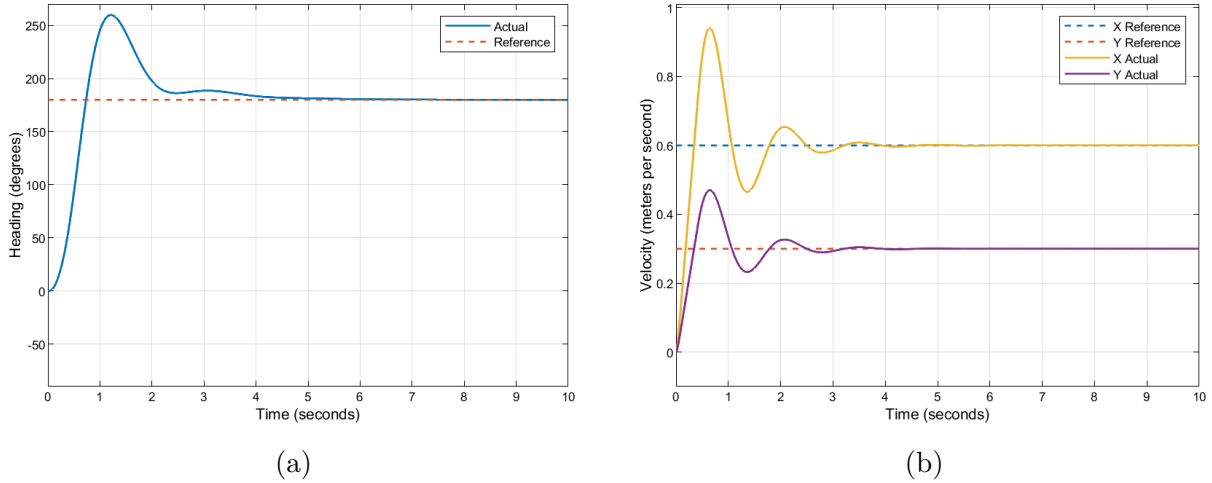
$$\begin{aligned} T_f &= \frac{\tau}{\beta} & T_i &= 2\zeta\tau - T_f & T_d &= \frac{\tau^2}{T_i} - T_f \\ K_p &= K_i T_i & K_d &= K_p T_d & N &= \frac{1}{T_f} \end{aligned}$$

Tabell 3: Regulatorparameterar vid beräkning

	Orientering	Hastighet
	$F_{\tau\theta}(s)$	$F_{\tau v}(s)$
ω_c	5	6
ζ	1	1
β	10	10
φ_m	40	45
τ	0.7800	0.0750
K_i	0.0132	0.3166
K_p	0.0196	0.0451
K_i	0.0132	0.3166
K_d	0.0065	0.0014
N	12.82	133.33

3.2.3 Justering av Reglersystemet

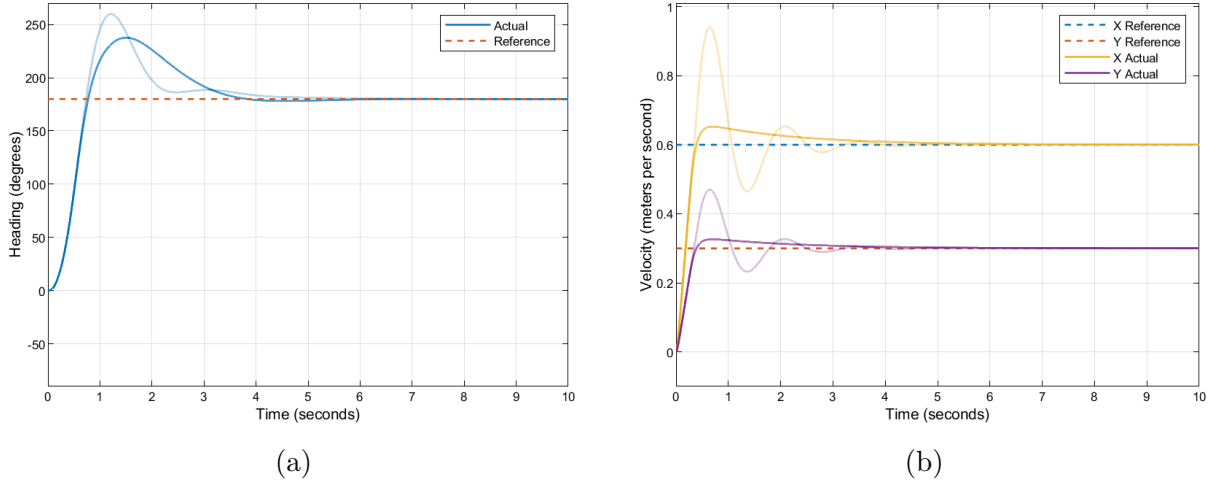
Värdena för regulatorerna i tabell 3 implementerades i Simscapemodellen. Detta gav stegsvaren för orientering- och hastighetsreglering som visas i figur 6, där figurerna 6a och 6b är oberoende av varandra. Att graferna är oberoende innebär att de är tagna vid två olika tidpunkter. I figur 6a roterar prototypen stillastående medan i figur 6b ändras endast hastigheten, prototypen har alltså ingen rotation samtidigt som hastigheterna ändras. Graferna visar fungerande reglersystem med rum för förbättring, särskilt när det kommer till överslängarna.



Figur 6: Stegsvär för orienterings- (6a) och hastighetsregulatorn (6b) med parametrarna tagna från tabell 3. Notera att 6a och 6b sker vid olika tidpunkter.

För att förbättra reglersystemet och få ett lugnare och mer robust beteende utan så extremt höga överslängar justerades värdena i regulatorerna i små steg. Detta visas i figur 7, där de blekta linjerna är samma som i figur 6 och de mest framträdande linjerna är de slutliga reglersystemen. Justeringen av parametrarna för hastighetsföljningen slutade då kraven 1.9 och 1.11 från kravspecifikationen, se figur 1, var uppfyllda. Det vill säga när överslängen var mindre än 10 % och insvängningstiden var mindre än fyra sekunder.

På motsvarande sätt slutades justeringen av parametrarna i orienteringsregulatorn då överslängen var mindre än 40 %, och insvängningstiden även här understeg fyra sekunder från kraven 1.10 och 1.12 i kravspecifikationen, se figur 1. Detta resulterade i att derivationsfaktorn för hastighetsregulatorn blev så liten att den kunde försummas och $F_{\tau v}$ blev alltså endast en PI regulator. Figurerna 7a och 7b sker oberoende av varandra.

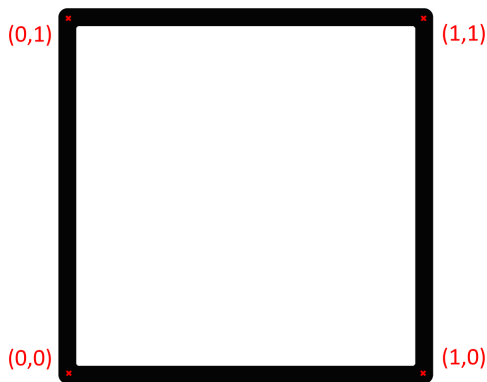


Figur 7: Stegsvär för orienterings- (7a) och hastighetsregulatorn (7b) med justerade parametrarna tagna från tabell 6 i kapitel 4.2.1. De blekta linjerna är reglersvaren med de tidigare parametervärdena som visas i figur 6a. Notera att 7a och 7b sker vid olika tidpunkter.

Egentligen vore mindre överslängar, särskilt för orienteringsregleringen, se figur 7a, önskvärt men detta skulle medföra att tiden för att kompensera för en störning blir längre. Alltså fick en kompromiss mellan översläng och möjligheten att kompensera för störningar inom rimlig tid göras. Se tabell 6 för slutgiltiga värden och kapitel 4.2.1 för verifiering av regelsystemet vid störningar.

3.2.4 Banans Uppbyggnad i Simscape

För att simulera banföljning krävdes någon typ av logik för att veta vart banan existerade och hur simulatorns sensorer kände om de var inom banan eller ej. Denna logik modellerades upp i Simscape genom att låta banan representeras av en lista med koordinatpunkter som sammanbands med raksträckor. Exempelvis skulle banan i figur 8 representeras av matrisen i ekvation 11.



$$\mathbf{B} = \begin{bmatrix} x_0 & y_0 \\ x_1 & y_1 \\ \vdots & \vdots \\ x_{n-1} & y_{n-1} \\ x_n & y_n \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (11)$$

Figur 8: Exempelbana

För att bestämma om en sensor med godtyckliga koordinater $\mathbf{p}_s = (x_s, y_s)$ ligger på banan itereras denna lista av banans punkter från 0 till $n - 1$. För varje iteration plockas två efterföljande punkter ut, det vill säga $\mathbf{p}_i = (x_i, y_i)$ och $\mathbf{p}_{i+1} = (x_{i+1}, y_{i+1})$. Med hjälp av transformationsformeln i ekvationen 12 kan alla punkterna roteras medurs så att punkterna $\mathbf{p}_i$ och $\mathbf{p}_{i+1}$ utgör en horisontell linje.

$$\hat{\mathbf{p}} = \mathbf{A}\mathbf{p} \quad (12)$$

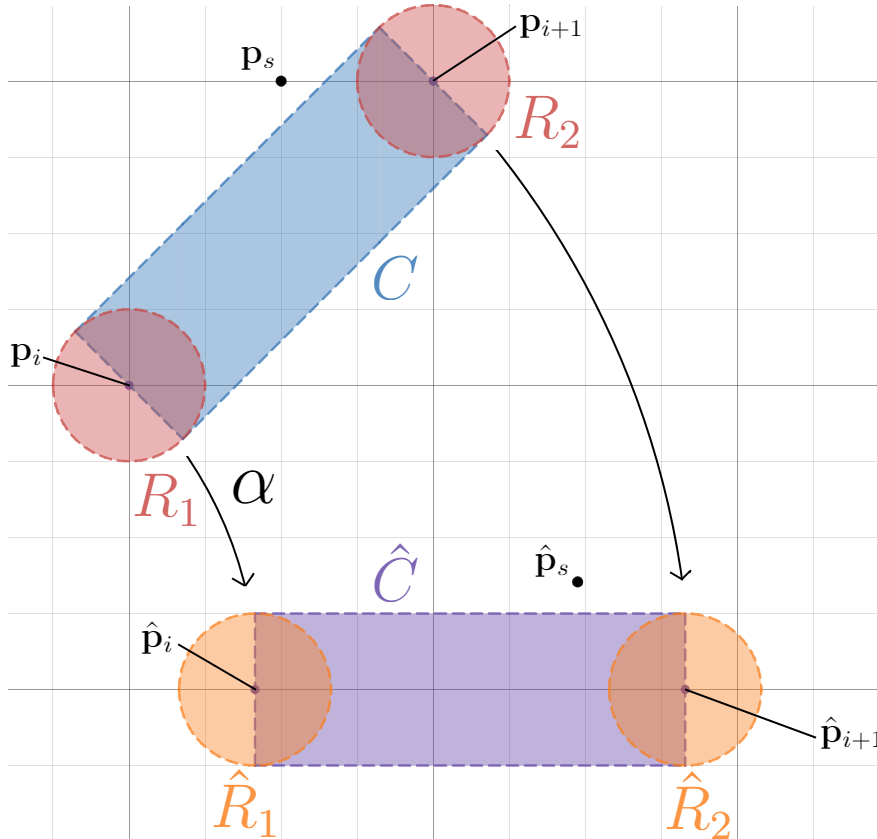
$$\mathbf{p} = \begin{bmatrix} x \\ y \end{bmatrix}, \quad \mathbf{A} = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}, \quad \alpha = \arctan2(y_{i+1} - y_i, x_{i+1} - x_i)$$

Detta innebär alltså att $y_i = y_{i+1}$. Då kan det bestämmas om punkten $\mathbf{p}_s$ ligger på banan mellan $\mathbf{p}_i$ och $\mathbf{p}_{i+1}$ genom att evaluera uttrycken i ekvation 13 och 14. Är en eller båda av dessa uttrycken sanna ligger punkten på banan. Se figur 9 för visualisering.

$$\underbrace{\left(\left((\hat{x}_i < \hat{x}_s < \hat{x}_{i+1}) \vee (\hat{x}_i > \hat{x}_s > \hat{x}_{i+1}) \right) \wedge (|\hat{y}_i - \hat{y}_s| < T) \right)}_{\hat{C}} \quad (13)$$

$$\underbrace{\left(\min(|\hat{\mathbf{p}}_i - \hat{\mathbf{p}}_s|, |\hat{\mathbf{p}}_{i+1} - \hat{\mathbf{p}}_s|) < T \right)}_{\hat{R}_1 + \hat{R}_2} \quad (14)$$

T är hälften av vägbredden.



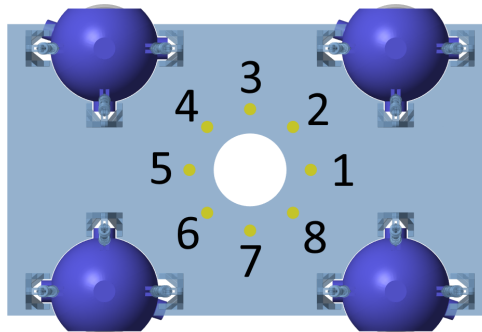
Figur 9: Visualisering av transformation och evaluering enligt ekvationerna 12, 13 och 14.

3.2.5 Utveckling av Banföljningslogiken

En avgränsning som gjorts är att inget framåtseende används till banföljningen. Detta begränsar prototypens banföljningskapacitet då den endast är varse om en mycket liten del av banan åt gången. Banan bestämdes till att vara en svart rand på en vit bakgrund och ska detekteras med hjälp av reflektansmätande sensorer [7].

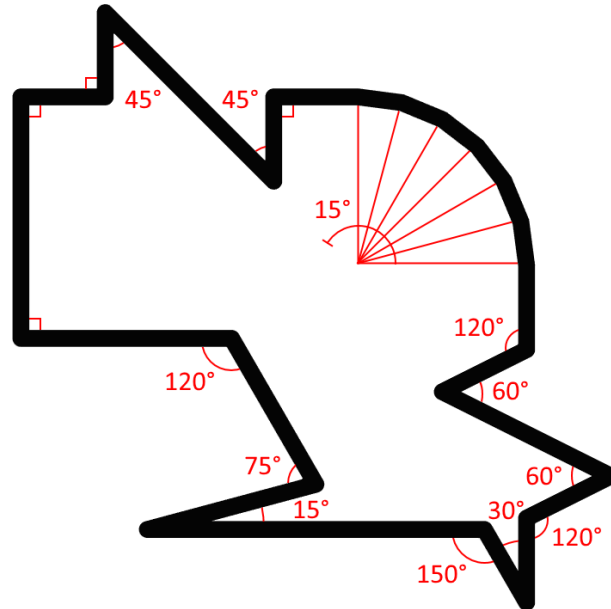
Skillnaden gentemot vanlig banföljning för en bil med cylindriska hjul är att dessa nyttjar en banföljningsalgoritm som svänger bilen så att samma sensorer alltid ska vara på banan [15]. Detta fungerar eftersom körriktningen är bunden till att vara i frontens riktning som i sin tur följer banan. Med sfäriska hjul öppnas möjligheten att följa en bana samtidigt som prototypen roterar godtyckligt kring sin z-axel, vilket försvårar banföljningsalgoritmen då det innebär att samma sensorer inte alltid ska vara på vägen. Vidare har det också gjorts en förenkling i form av att prototypen inte själv ska bestämma vilken referenshastighet den ska ha under banföljningen, utan den sätts av användaren före start och är konstant under en körning. Det enda som banföljningsalgoritmen styr är således vilken riktningen prototypen kör i.

Eftersom prototypen ska ha samma banföljningsegenskaper i alla riktningar oavsett prototypens orientering valdes sensorerna att vara placerade med jämna mellanrum i en cirkel. Anledningen till att åtta sensorer valdes är en sammanvägning av olika faktorer, som kostnad, fysisk plats och vinkelupplösning. Ju fler sensorer som används desto högre vinkelupplösning erhålls, vilket i sin tur möjliggör en mjukare banföljning då mer information om banan kan uppmätas. Många sensorer ger också möjlighet till smalare väg utan att riskera att vägen passerar emellan sensorerna vilket innebär att alla sensorer missar banan trots att prototypen är på banan. Nackdelarna med många sensorer är den fysiska plats de upptar samt kostnaden för sensorerna. Totalt sett gav detta att åtta sensorer i cirkeln med radien 50 mm som visas i figur 10 var rimlig då banans bredd också var 50 mm. Åtta sensorer visade sig också uppfylla kravet 1.7 i kravspecifikationen, se figur 1. Kravet var att den maximala vinkeln mellan banan och masscentrums körriktning vid banföljning av rak bana ska vara mindre än 25° . Vid implementation av den nedan utvecklade banalgoritmen blir den maximala avvikelsen 22.5° .



Figur 10: Sensorerna är de gula cirklarna och de är placerade symmetriskt kring prototypens geometriska centrum med radie 50 mm.

För att utveckla en banföljningsalgoritm simulerades även banföljning i Simscape. En bana utformades, se figur 11, för att testa och utveckla banföljningsalgoritmens förmåga att klara av flera olika typer av svängar, enligt kapitel 3.2.4. Anledningen till att banföljningen utvecklas är att detta är ett funktionskrav från kravspecifikationen, se figur 1, satt för att kunna simulera och utveckla autonoma köregenskaperna med sfäriska hjul.



Figur 11: Banan som prototypen ska köra autonomt. Notera att de sex vinklarna i kvartscirkeln är 15° styck. Banans bredd är 50 mm

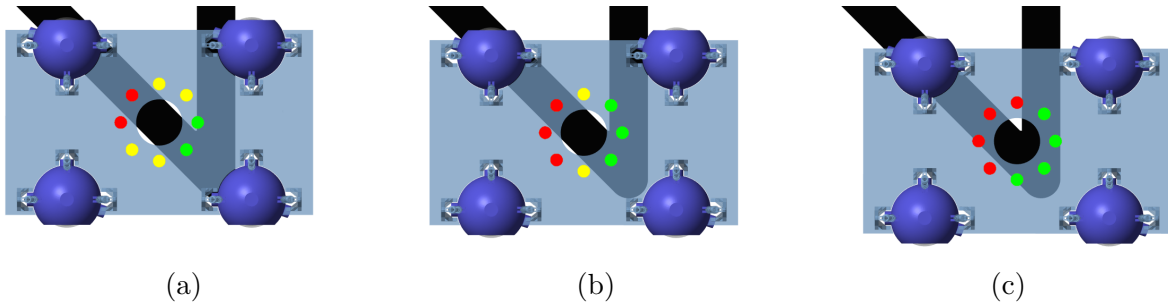
För att förstå utvecklandet av banföljningsalgoritmen förklaras först lite terminologi som presenteras i tabell 4. De färgade cirkelarna i tabellen används i figurerna nedan vid visualisering av sensorernas tillstånd under simulering. Om en sensor både är drivande och aktiv blir färgen i simuleringen grön och likaså blir sensorn röd om den både är aktiv och släckt.

Tabell 4: Terminologi

Aktiv sensor [●]	En sensor som är på banan
Inaktiv sensor [●]	En sensor som inte är på banan
Flanka positivt	En sensor som går från inaktiv till aktiv
Drivande sensor [●]	Prototypen får order att köra i sensors riktning, eller i en superpositionering av drivande sensorer
Släckt sensor [●]	En sensor som inte kan bli drivande endast genom att flanka positivt
Komplementsensor	Sensorn på motsatt sida av sensorcirkeln

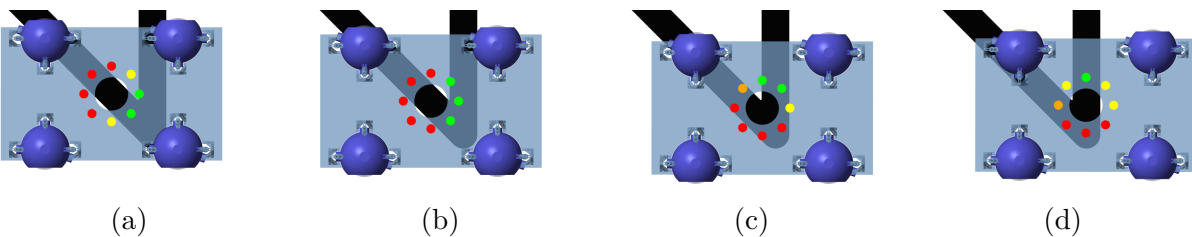
Grundidén till banföljningsalgoritmen bygger på att en eller flera sensorer kan vara drivande. Om flera sensorer är drivande samtidigt adderas de olika drivriktningarna i en vektoraddition som renderar en resultant. Prototypen ges sedermera en referenshastighet i resultantens riktning, alltid med lika stor norm. Svårigheten i att utveckla en banföljningsalgoritm låg således i att utifrån sensordata och simuleringstid bestämma huruvida en given sensor skulle vara drivande eller ej. Vid utvecklingen av algoritmen sattes rotationen till att vara låst eftersom detta var mer överskådligt och en tillräckligt allmän algoritm även ska fungera vid godtycklig rotation.

En enkel algoritm som prövades var att betrakta varje sensor och dess komplementsensor som ett par, där endast en kunde vara aktiv åt gången. Vid simuleringstart sattes en sensor som drivande och därefter blev varje sensor som flankade positivt drivande förutsatt att dess komplement var icke-drivande. Sensorerna för alla figurer numreras enligt samma konvention som i figur 10, det vill säga moturs med sensor 1 längst till höger. Vid kraftigare svängar än 90° stöter denna algoritm på problem som illustreras i figur 12. Problemet är att sensor 7 blir drivande före sensor 3 och därmed blir sensor 3 släckt och prototypen kör av banan snart efter 12c.



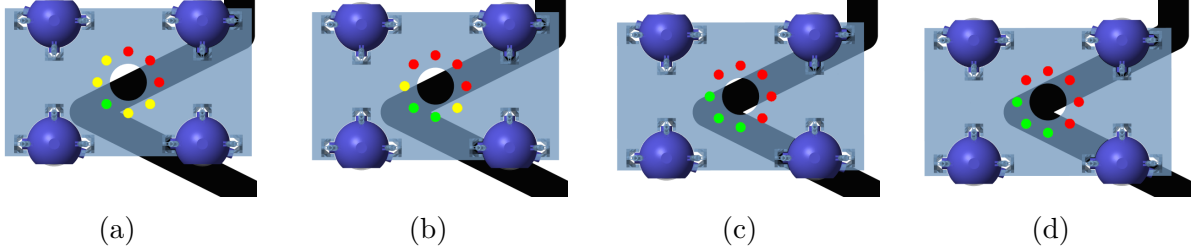
Figur 12: Illustration över hur banföljningsalgoritmen fungerar i en 45° -kurva. Prototypen kör i de gröna sensorernas superpositionerade riktning. Prototypen kör av banan i nästa sekvens.

Algoritmen modifierades så att varje sensor utöver sin komplementsensor även släcker ut de två grannsensornas komplementsensorer. Dessutom modifierades algoritmen så att en släckt sensor numera även kan bli drivande om den flankar positivt då dess grannsensor är den senaste sensorn som blivit drivande. Ytterligare krav för detta är att komplementsensorn är icke-drivande och grannsensorn fortfarande är drivande. Dessa ändringar medför att prototypen klarar av 45° -svängar. Kurvtagningen med denna algoritm visas i figur 13. Den orange cirkeln är en sensor som är aktiv men som flankades på släckt, den måste därmed flanka positivt på nytt innan den kan bli drivande.



Figur 13: Illustration över hur banföljningsalgoritmen fungerar i en 45° -kurva. Algoritmen klarar nu av kurvan.

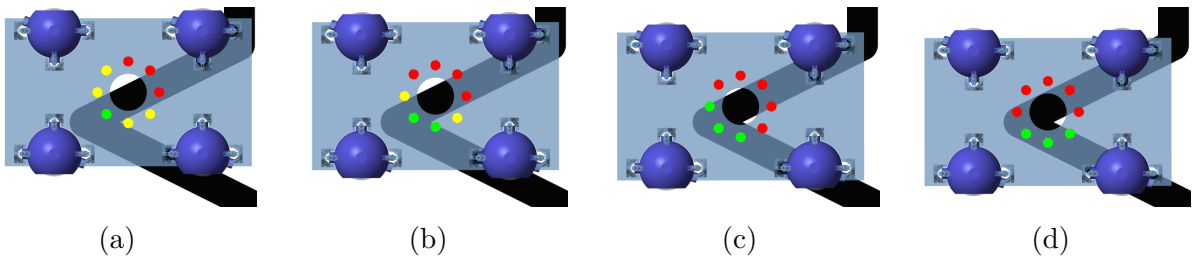
Vissa omständigheter resulterar i att en sensor som egentligen ska bli drivande för att prototypen ska följa banan är släckt och ingen av dess grannsensorer är den sista sensorn som blivit drivande. Ett sådant fall visas i figur 14 där sensor 5 blir drivande efter sensor 7 och därmed omöjliggör att sensor 8 blir drivande.



Figur 14: Illustration över hur banföljningsalgoritmen fungerar i en 60°-kurva. Prototypen kör av banan vid nästa sekvens.

För att undvika att prototypen kör i fel riktning behöver sensor 8 bli drivande när denna flankar positivt mellan figur 14c och 14d och samtidigt släcka ut sensor 5. Detta åstadkoms genom att modifiera banföljningsalgoritmen så att en släckt sensor även kan bli drivande när den flankar positivt och någon av dess grannsensorer är den näst senaste sensorn som blivit drivande. Ytterligare krav för att detta ska ske är att den givna grannsensorn fortfarande är drivande samt att den påflankade sensorns komplementsensor är icke drivande. Detta ger kurvtagningen i figur 15.

I vissa fall är det dock oönskat att en sensor ska bli drivande endast på grund av att den är granne med den näst senaste sensorn som blivit drivande och därför implementerades ett tidskriterie. Kriteriet säger att tiden mellan flankningen av den näst senaste och den senaste drivande sensorn måste understiga $T_c = 0.1$ s. Detta minskar risken att en felaktig sensor ska bli drivande trots att dess grannsensor är den näst senaste tillskottet till drivsensorerna. Anledningen till att 0.1 s användes är för att det gav ett bra resultat i de flesta simuleringarna, men T_c borde egentligen variera med vilken hastighet prototypen kör i, banans utformning och sensorradien.



Figur 15: Illustration över hur banföljningsalgoritmen fungerar i en 60°-kurva. Algoritmen klarar nu kurvan eftersom sensor 7 är den näst senaste sensorn som blivit drivande vilket medför att sensor 8 kan bli drivande.

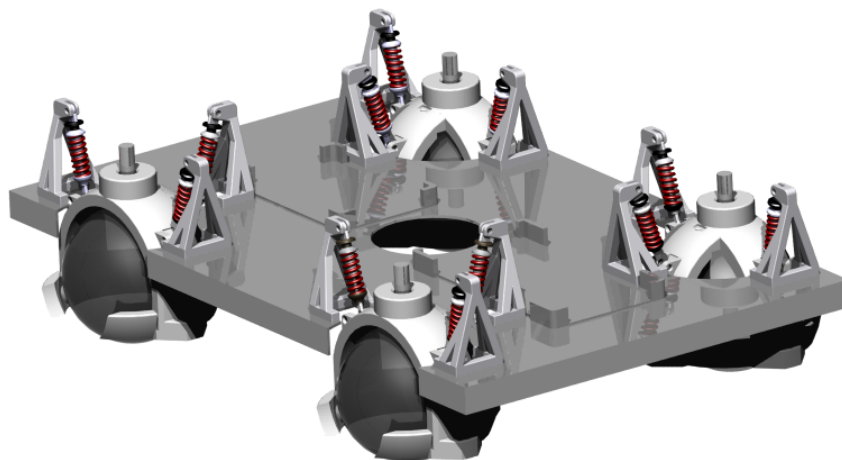
4 Resultat

I resultatdelen beskrivs de resultat som projektet kommit fram till och är uppdelad mellan konstruktionen och de resultat som erhöles från simuleringen. I konstruktionsavsnittet presenteras resultat från modelleringen och den fysiska prototypen. Vidare behandlar simuleringsavsnittet resultatet från reglersystem samt banföljningsalgoritm.

4.1 Fysisk Konstruktion och Modellering

Som underlag till både konstruktionen och simuleringen, samt för att verifiera krav på storlek och vikt modellerades prototypen med CAD-programvaran Catia V5. En montering av flera komponenter (assembly) av CAD-modellen visas i figur 16. Till konstruktionsarbetet användes CAD-modelleringar av komponenter med komplexa former som underlag till 3D-utskrift för delarna till infästningen samt stöd till stötdämparna och utskärning i metall till chassit. Assemblyn av CAD-modellen sparades även som en stepfil för att kunna läsas in i simulatorverktöget.

De ingående komponenterna består av ett chassi, stötdämpare med ställningar och infästningar [16]. Det rektangulära chassit består av 1.5 mm aluminiumplåt och är konstruerat för att stödja montering av övriga komponenter. Vidare är kanterna runt chassit bockade för att ge en stabilare konstruktion. Aluminium valdes som material till chassit då det är ett lättbearbetat material med låg vikt. Aluminiumchassit med dimensionerna ur tabell 5 uppfyller därigenom krav 3.1 ur kravspecifikationen från figur 1. Längs mitten återfinns två formgivningar för fasthållning, den ena med mått för att passa en Raspberry Pi 3 B+ och den andra för att passa den strömförsörjande powerbanken. I mitten återfinns en utskärning med syfte att stödja monteringen av sensorer och förbättra räckvidden vid kopplingen av sladdar. Vid infästningarna finns skruvhål för att möjliggöra montering av ställningarna för stötdämparna.



Figur 16: Helhetsbild på prototypen

Infästningarna består av en modulariserad kåpa i plast som omsluter hjulet till mer än hälften för att på så vis förhindra att hjulen kan trilla av. Modulariseringen av infästningarna innebär att kåpan 3D-printas i tre separata delar där plasttråden läggs längs med den riktning som kraften verkar och på så sätt fås en robustare konstruktion [17]. Delarna skruvas sedan ihop med 2.5 M skruv. Plast används som material då det har låg vikt och även tillåter additiv tillverkning av komplexa former [18]. På insidan av kåpan monteras en kulrulle (ball transfer unit) som har kontakt med hjulet i syfte att minimera friktionsmotståndet [19]. I figur 17 presenteras två helhetsbilder och en sprängskiss av infästningen för att visualisera modulariseringen och positionen av kullagret. Det svarta klotet i figurerna illustrerar hjulet.



Figur 17: Helhetsbilder och sprängskiss av infästningen

Den undre delen, del ett, i sprängskissen utgör botten av infästningen och har fem spetsiga utbuktningar på insidan som agerar kontaktpunkter mellan hjul och infästning. Den spetsiga utformningen minimerar kontaktytan, vilket därmed reducerar friktionen. Den andra delen har en ihålig cylinder i toppen där kulrullen monteras från insidan. Både del två och tre har dessutom fästen på utsidan för att möjliggöra montering av fjädringen. Konstruktionen möjliggör montering och demontering med hjälp av modulariseringen. Den undre delen, del ett, av infästningen går att skruva loss med tre skruvar från undersidan. På så vis kan varje hjul monteras och demonteras oberoende av andra komponenter.

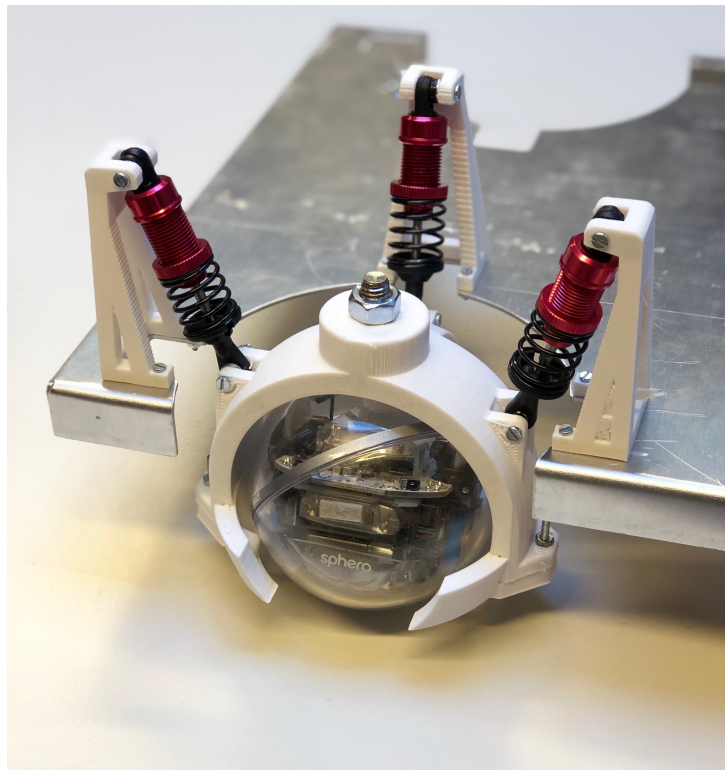
Med hjälp av modelleringen av komponenterna och med materialvalen från kravspecifikationen erhöles vikter och mått för chassit och infästningen, se tabell 5. Se appendix D för detaljritningar på komponenterna.

Tabell 5: Vikt och mått för chassit och infästning

Komponent	Vikt [kg]	Mått [mm]
Chassi	0.33	400 x 240 x 1.5
Infästning	0.083	Ø83

Till följd av att det inte släppts ett officiellt SDK till Sphero Bolt (se appendix B) kunde inte styrningen implementeras fullt ut i den fysiska prototypen. Detta medför att den tilltänka elektroniken inte längre fyller någon funktion för den fysiska prototypen. Därav beslutades att simplificera prototypen genom att rekonstruera chassit till att inte innefatta stöd för montering av elektronik och helt bortse från att montera elektroniken.

Detta resulterade i att prestandakraven 1.1 – 1.5 från kravspecifikationen i figur 1 inte kunde verifieras av den fysiska prototypen. Istället användes den simulerade prototypen till dessa tester, resultatet visar att kraven är uppfyllda. Den förenklade fysiska prototypen användes däremot till att genomföra tester med syfte att verifiera kraven på infästningarna. Figur 18 illustrerar en infästning på den fysiska prototypen.



Figur 18: Infästningen på den fysiska prototypen

Dessa tester riktade sig till att undersöka om hjulen hålls inneslutna av infästningarna och om hjulen går att montera och demontera från konstruktionen. Resultaten visar dels att infästningen omsluter hjulet så att det inte trillar ut och dels att det går att montera eller demontera hjulet genom att installera eller avlägsna den understa delen av infästningen med tre skruvar. Det noteras även att hjulet kan rotera i konstruktionen och har tre till fyra kontaktpunkter av sex möjliga gentemot infästningen vid drift, vilket reducerar friktionsmotståndet.

4.2 Utfall av Simulering

I detta kapitel presenteras resultaten från simuleringarna som gjordes i Simscape. Resultaten är uppdelade enligt hur reglersystemen fungerar samt hur banföljningen presterade.

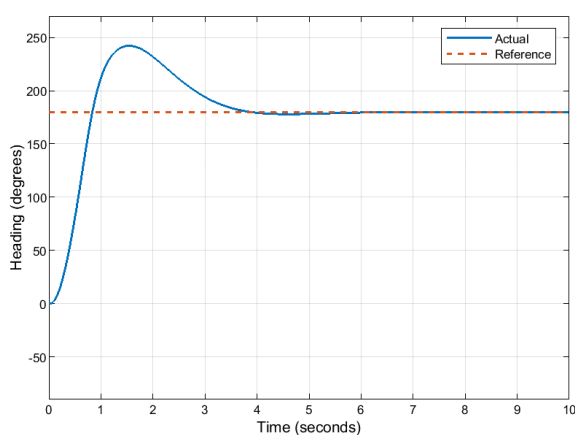
4.2.1 Verifiering av Reglersystem

Från reglersystemets dimensionering och justering i kapitlena 3.2.2 och 3.2.3 erhöles värdena i tabell 6.

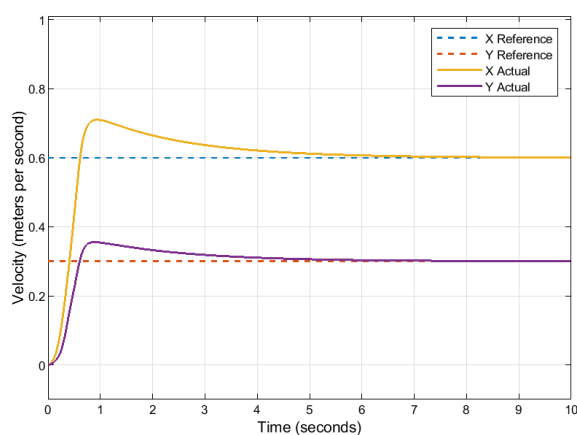
Tabell 6: Justerade regleringsparameterar

	Orientering $F_{\tau\theta}(s)$	Hastighet $F_{\tau v}(s)$
K_p	0.027	0.12
K_i	0.026	0.063
K_d	0.01326	
N	12.82	

Ur justeringen av parametrarna i kapitel 3.2.3 verifieras att översläng och insvängnings-tiden uppfyller kraven från kravspecifikationen för vardera system. Ett stegsvar för båda systemen samtidigt, se figur 19, visar något större översläng och längre insvängningstid. Detta är förväntat då båda reglersystem nu gör anspråk på hjulens begränsade momentkapacitet och därmed ger varandra störningar. Att reglersystemen i figur 19 inte uppnår kraven i kravspecifikationen är i detta fallet acceptabelt. Vilket motiveras genom att kra-ven är satta för vardera reglersystem enskilt och inte då de ska prestera samtidigt.



(a)

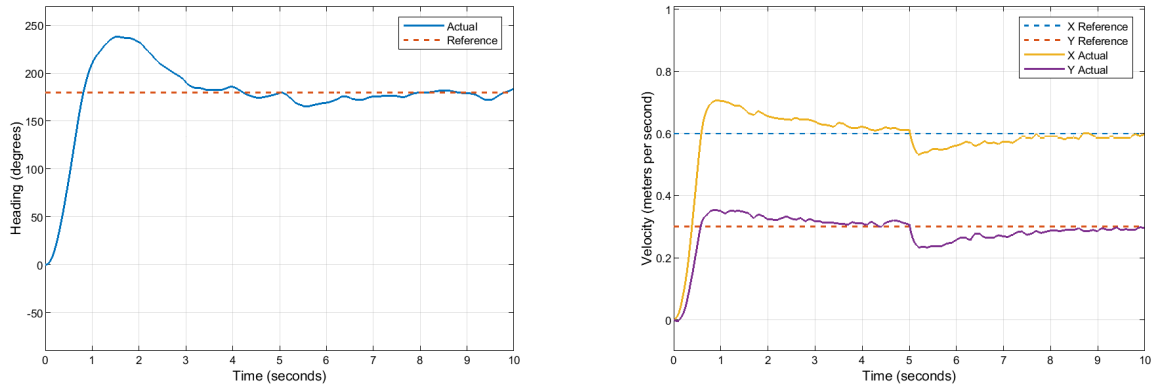


(b)

Figur 19: Reglersvar för orientering (19a) och hastighet i både x- och y-riktningen (19b), då 19a och 19b sker samtidigt. Notera att överslängarna och insvängningstiderna är aningen sämre än i 7, där 7a och 7b sker oberoende av varandra.

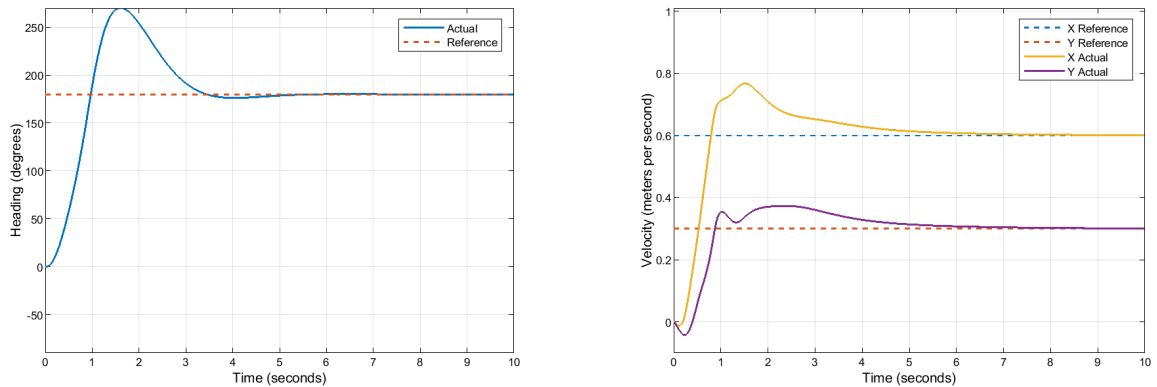
Reglersystemet ska även klara av att hantera störningar, vilket verifieras genom simuleringar. Störningar för orienteringen simuleras med vitt brus med en brusstyrka (noise power) på 0.0005 och samplingstid 0.1 s. Detta ger störningar på ungefär 0.1 ± 0.05 Nm i magnitud kring prototypens egna z-axel. Efter 5 s läggs även en konstant störning på 0.15 Nm medurs in. Detta är ungefär 10 – 30 % och 30 % respektive av det maximala momentet som prototypen kan generera kring sin egna z-axel.

Störningarna för hastighetsregleringen simuleras också med vitt brus med en brusstyrka (noise power) på 0.001 och samplingstid 0.1 s. Störningarna blir omkring 0.15 ± 0.1 N i magnitud längs med både x- och y-axeln. Även för hastighet läggs en konstant störning på 1 N in efter 5 s i båda riktningar. Dessa störningar är ungefär 2 – 11 % och 43 % respektive av den maximala kraften som prototypen kan producera. Resultatet då störningarna som nämnts ovan läggs på samtidigt visas i figur 20. Detta resultat visar att reglersystemet kan hantera störningar väl.



Figur 20: Reglersvar för orientering- och hastighetsreglering respektive vid störningar. Vitt brus är tillagt för båda system. Efter 5 s läggs dessutom konstanta störningar på, vilket förklarar varför avvikelser mellan bör- och ärvärde ökar vid denna tidpunkt.

Ett annat verkligt scenario som kan uppstå är att ett hjul slirar eller slutar fungera. Detta simuleras genom att inte låta ett av hjulen producera något vridmoment. Resultatet av att endast tre hjul fungerar visas i figur 21. Figuren illustrerar att överslängarna samt insvängningstiderna ökar, vilket är förväntat vid en så stor störning. Reglersystemet klarar ändå av att rätta till sig och följa referenssignalerna vilket fortfarande tyder på att regleringen fungerar som önskat.

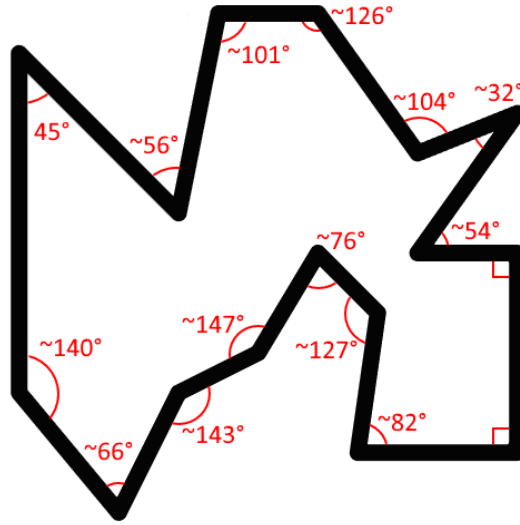


Figur 21: Stegsvär då endast tre av fyra hjul fungerar

4.2.2 Prestation av Banföljningsalgoritm

Prototypen kunde i simuleringsmiljön följa banan autonomt i figur 11 då referenshastigheten för banföljningen låg mellan ~ 0.1 m/s och ~ 0.35 m/s samt då rotationshastigheten kring z-axeln var mellan $0 < \omega_Z < \sim \pi/2$ rad/s. Vid hög rotationshastighet kunde inte banföljningen gå lika snabbt eftersom hjulen endast kan producera ett begränsat moment. En banföljning av testbanan, se figur 11, med hastighet 0.30 m/s och rotationshastigheten $\omega_Z = 0$ visas i videon ur referens [20]. Även en banföljning med hastighet 0.3 m/s och rotationshastighet $\omega_Z = \pi/2$ visas i videon ur referens [21]. Eftersom algoritmen utvecklats på denna bana testas även algoritmen på en mer godtycklig bana, se figur 22. Resultatet visas i videon ur referens [22]. I denna banföljning är hastigheten nu 0.30 m/s och $\omega_Z = 0$. Till sist visas även banföljningen på samma bana med hastighet 0.20 m/s och $\omega_Z = \pi/2$ i videon ur referens [23].

I videoklippen visas utöver banföljningen även graferna för hastighetsföljning i både x- och y-led, den totala hastigheten ($\sqrt{v_x^2 + v_y^2}$) som är ungefär samma som den satta referenshastigheten, orienteringsregleringen samt vilka sensorer som är på banan. Från dessa videor syns att den totala hastigheten minskar lite i varje kurva på grund av ändringen i körriktning för att därefter återgå till referenshastigheten.

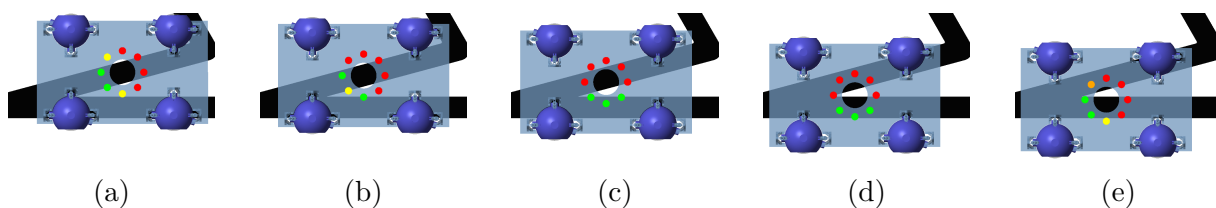


Figur 22: En ny bana för verifiering av banföljningsalgoritm.

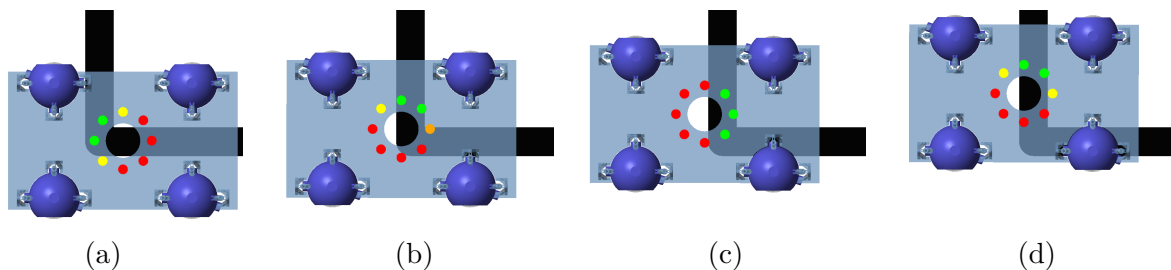
Då rotationen inte medförde att nya fel eller problem tillkom så behandlar resterande del av resultatet specialfallet då $\omega_Z = 0$ på grund av ökad överskådlighet.

Vid högre referenshastighet än ~ 0.35 m/s blev bankörningen mycket instabil och i vissa kurvor hann inte sensorerna ändra prototypens körriktning tillräckligt snabbt. Problemet vid för hög hastighet ligger således inte i banföljningsalgoritmen utan i bilens tröghet i förhållande till banans bredd. Vid lägre hastighet än 0.1 m/s fick däremot banföljningsalgoritmen problem i 15°-kurvan i banan, se figur 11, vid kurvtagning moturs, vilket visas i figur 23. Anledningen till detta är att vid låg hastighet anpassas prototypens körriktning efter en kortare körsträcka från dess att sensordrivningen förändrats än vad den gör vid hög hastighet.

Detta medför att vid kurvtagningen av den 15°-kurvan moturs med låg hastighet reagerar prototypen så snabbt till följd av påflankning av sensor 7, se figur 10, att sensor 5 flankar av, figur 23d. Efter det att sensor 5 flankat av kör prototypen ortogonalt mot banan tills dess att någon av sensorerna 1 eller 5 återigen flankar på. Vilken av sensorerna 1 eller 5 som flankar på först är slumpmässigt eftersom prototypen kör ortogonalt mot banan, men ordningen av påflankning avgör vilken riktning prototypen ska köra i. I figur 23e flankar sensor 5 på ett par klockningar före sensor 1 vilket gör att prototypen kör åt fel håll. Ett sensormässigt liknande fall uppstår då prototypen kör vid så hög hastighet att den nästan kör av banan vid en 90°-sväng, se figur 24 där referenshastigheten är 0.4 m/s. Även i detta läge kommer prototypen endast att ha tre sensorer på banan som ger prototypen en referenshastighet ortogonalt mot banan. Fallet visas i figur 24c som är mycket lik sensorsituationen i figur 23c. Anledningen att den klarar kurvan är att sensor 3 flankar på före sensor 7 i figur 24c, vilket är en tillfällighet.



Figur 23: Illustration över hur banföljningsalgoritmen fungerar i en 15°-kurva när referenshastigheten för banföljningen är 0.1 m/s. Notera att i 23d är sensor 5 utanför banan.



Figur 24: Illustration över hur banföljningsalgoritmen fungerar i en 90°-kurva då referenshastigheten för banföljningen är 0.4 m/s. Anledningen att den klarar kurvan är att sensor 3 flankar på före sensor 7 i 24d, vilket är slumpmässigt.

Vid körning medurs i den 15° -svängen påvisas ett annat problem vid låg hastighet. Först flankar sensor 3 på, sedan flankar sensor 4 på. För att sensor 2 ska flankar på och bli drivande så att prototypen drivs åt rätt håll måste tiden mellan påflankning av sensor 3 och 4 ($t_{3 \rightarrow 4}$) vara mindre än tidskonstanten T_c som introducerades i kapitel 3.2.5. Därefter flankar sensor 5 på och denna sensor blir drivande om tiden mellan påflankning av sensor 4 och sensor 2 ($t_{4 \rightarrow 2}$) är mindre än T_c . Alltså måste $t_{3 \rightarrow 4} < T_c < t_{4 \rightarrow 2}$. Således bör det finnas ett optimalt värde på tidskonstanten T_c som beror på prototypens hastighet.

Detta innebär att det istället för en konstant tid som avgör om en sensor som flankar positivt kan bli drivande då den är granne med en sensor som är den näst senast som blivit drivande, borde det vara en funktion. Funktionen bör vara proportionerlig mot inversen av referenshastigheten, $f(v) \sim \alpha \cdot v_{ref}^{-1}$. Anledningen till att detta problem sker vid låg hastighet är att sensor 5 inte flankar av vid hög hastighet, vilket innebär att så länge $t_{3 \rightarrow 4} < T_c$ klarar prototypen kurvan.

Sammanfattningsvis finns det två stycken kända problem med banföljningsalgoritmen, dels när prototypen får en referenshastighet ortogonalt mot banan och dels att T_c ska variera som en funktion av hastighet, sensorradie och bana. Det första problemet kan troligtvis lösas genom att lagra mer information från sensorerna medan det andra problemet är mer svårlöst. Att bestämma en funktion för T_c som gäller för alla sorters referenshastigheter, sensorradier och banor skulle kräva extensiva simuleringar och tester. Tester med vinklar under 15° har inte heller undersökts, och där finns möjligtvis fler problem. Den nuvarande algoritmen klarar dock de flesta kurvor vid rimliga hastigheter då $T_c \approx 0.1$ med $\omega_Z < \pi/2$. I synnerhet har algoritmen inga problem vid vinklar trubbigare än 90° , vilket innebär att kravet från kravspecifikationen är uppfyllt med marginal.

5 Diskussion

I detta kapitel diskuteras resultaten från föregående kapitel uppdelat i konstruktion och simulering, där reglersystem och banföljning ingår. Dessutom finns det en underrubrik om samhällliga och etiska aspekter som diskuterar vilka effekter en bil med sfäriska hjul kan få på samhället.

5.1 Utvärdering av Konstruktion

En stor del av konstruktionsarbetet utfördes i CAD. Det innebar att komponenterna enkelt kunde korrigeras och anpassas då nya idéer och behov uppdagades. Möjligheten att anpassa modellerna innebar även en stor fördel då det var problematiskt att hitta stötdämpare och rullager som passade den initiala designen. På så sätt kunde utvecklingsarbetet fortgå och anpassas efter de inköpta komponenterna i ett senare skede.

Genom att tilldela de olika komponenterna det tilltänkta materialvalen kunde modellen användas till att verifiera om kraven ställda på vikt och mått var uppfyllda. Det visade sig att storlekskravet, krav 2.1 i figur 1, på chassit är uppfyllt samt att vikten för komponenten är 0.33 kg. Från kravspecifikationen ställdes ett krav, se krav 3.1, på chassits vikt som medför en vikt under 0.5 kg och ett önskemål på under 0.3 kg, se önskan 3.2. Slutsatsen dras om att kravet är uppfyllt med marginal. Det ställs även ett krav, se krav 3.3, på att infästningen inte får väga mer än 0.1 kg och ett önskemål, se önskan 3.4, på under 0.085 kg, samt att infästningen ska förhålla sig storleksmässigt ungefär lika stora som hjulen, se önskan 2.2. Med en innerradie som är 5 mm större än hjulets diameter anses önskemålet avseende storlek vara uppfyllt. Infästningens vikt är 0.081 kg, därmed dras slutsatsen att önskan på att väga under 0.085 kg är uppfylld, med knapp marginal.

CAD-modellen användes även som grund för det slutgiltiga konstruktionsarbetet, att tillverka en prototyp. Genom att först modellera bilen erhöles en grundläggande förståelse för korrelationen mellan de olika komponenterna. Denna kunskap var till stor nytta vid konstruktionen av prototypen. CAD-modellen kunde delvis användas till att 3D-printa komponenter med komplexa former men även till att skära ut urtagen i chassit. Med hjälp av 3D-utskriftarna kunde ett iterativt arbete för att testa delfunktioner genomföras. Till exempel kunde en förenklad version av infästningen skrivas ut i syfte att erhålla kunskap om monterings- och demonteringsprocessen. Med den iterativa processen som grund kunde till slut ett koncept anpassat för sfäriska hjul fastställas, vilket innebär att infästningen möjliggör fri rotation.

Det finns en del problem med konstruktionen om den skulle skalas upp. Ett problem är att den nuvarande konstruktionen bygger på att hjulen har en inre drivning, vilket skulle vara problematiskt storskaligt då väldigt mycket energi skulle behöva lagras i hjulen. Att istället ha en centraliserad energiutvinning eller energilagring som sedan skall överföra denna energi till hjulen skulle ställa nya krav på konstruktionen. Även infästningen skulle få problem i större skala eftersom hela prototypens vikt vilar på hjulen genom en kulrulle. Med tyngre konstruktion skulle detta ge ett högt tryck på en liten del av hjulet vilket skulle vara mycket slitsamt. Med den nuvarande konstruktionen kan inte heller hjulen ha

ett för djupt mönsterdjup eftersom detta skulle leda till att kulrullen skulle fastna i hjulet. I större skala skulle det vara önskvärt om chassit och hjulen inte hade kontakt, exempelvis genom magnetisk levitation som Goodyear föreslog i sitt koncept [4]. Magnetism var dock aldrig relevant i detta projekt eftersom detta störde sensorerna hos Sphero Bolt. Om det framtagna konceptet ändå ska återanvändas i en storskalig produkt bör chassit vila på betydligt fler kulrullar, för att på så vis fördela vikten. Ett annat förslag är att göra en enda stor kulrulle, där kulan är själva hjulet.

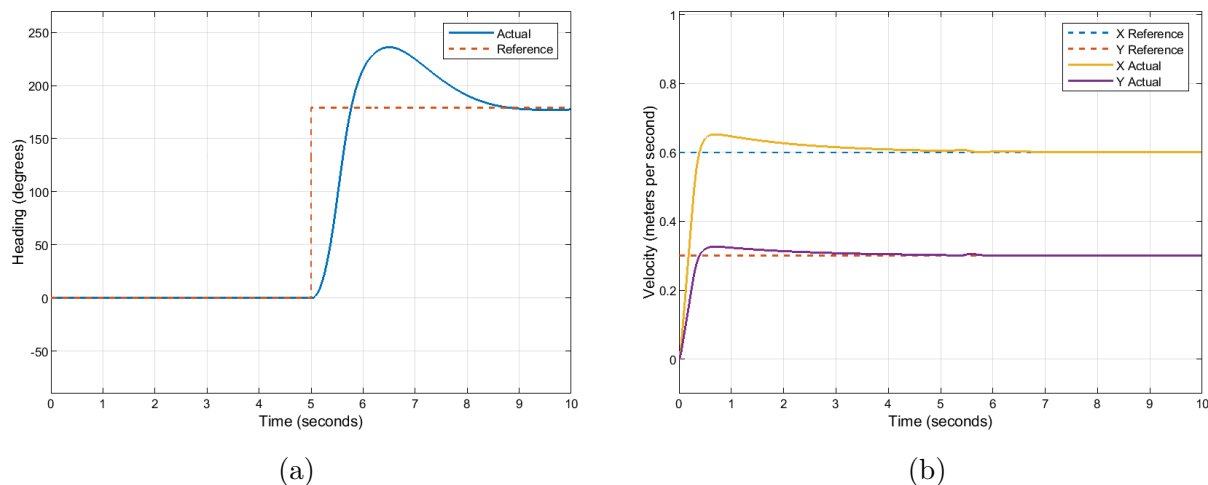
5.2 Evaluering av Simulering

I detta kapitel diskuteras resultaten från simuleringarna som gjordes i Simscape. Diskussionen är uppdelade mellan reglersystemen och banföljningen.

5.2.1 Eventuella Problem med Reglersystemet

En nackdel med det nuvarande reglersystemet (två stycken PID-regulatorer) är att de skapar störningar för varandra, vilket beror på att regulatorerna är designade oberoende av varandra. När ett av reglersystemen får en ändring i sin referenssignal kan detta därmed påverka och ge störningar i det andra systemets dynamik. Se figur 25 för ett exempel på hur orienteringens referenssignal ändras, åtgärdas och ger störningar i hastighetsregleringen.

Med mer sofistikerade reglertekniker kan dessa effekter i bästa fall helt elimineras. Från figur 25 syns dock att dessa effekter nästan är helt försumbara och därmed tas ingen vidare hänsyn till detta.

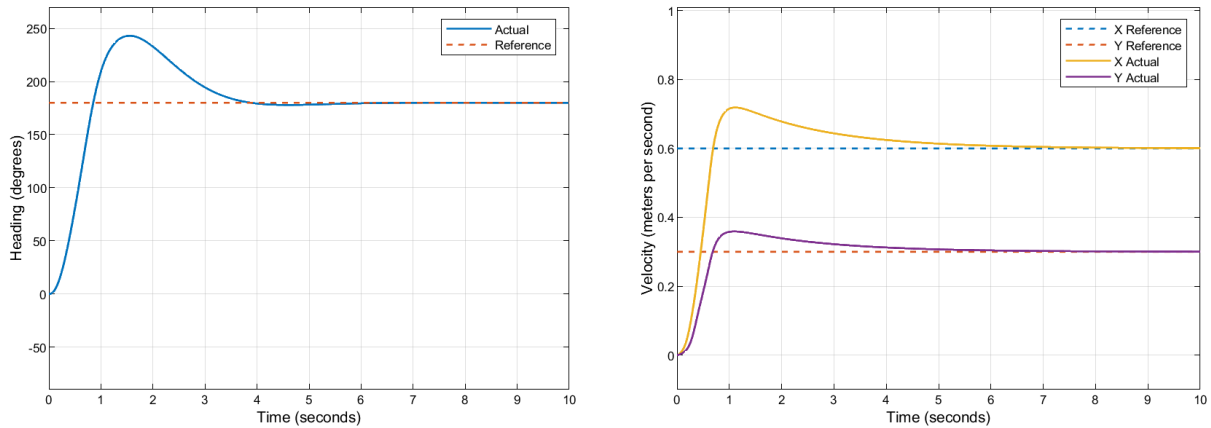


Figur 25: Reglersvar där orienterings- och hastighetsregleringen sker samtidigt. Notera hur en ändring i orienteringens börvärde efter 5 s påverkar hastigheten, om än mycket lite.

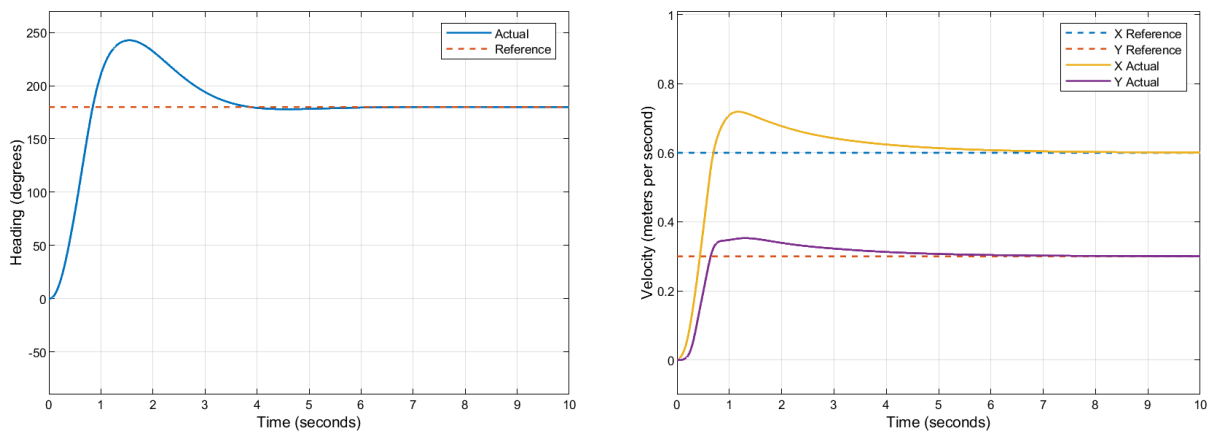
Alla simuleringsberäkningar är utförda med ett samplingsintervall på 0.0001 s. Om klockningsintervallet i det realiserade reglersystemet överstiger samplingsintervallet i simuleringen finns det risk att systemet får sämre karakteristik. Ytterligare är sensordatan från

hastighet och orientering exakt i simuleringen till skillnad från vad de skulle vara i verkligheten. Detta är också något som skulle kunna påverka reglersystemets beteende. Dock ansågs reglersystemet prestera tillräckligt bra med brusstörningar i form av kraft och vridmoment. Dessa kraft- och vridmomentsstörningar kan ses som störningar i sensordatan i stället för yttre störningar och därmed anses sensorstörningar inte ge några större problem.

Vid simulering så togs ingen hänsyn till elektronikens massa på prototypen. Undersökning visade att en punktmassa på 0.5 kg i mitten på prototypen hade en mycket liten inverkan på reglersystemets stegsvar, se figur 26. Om punktmassan istället placerades 1.2 cm åt höger (i breddled) samt 2 cm framåt (i längdled) från prototypens geometriska mittpunkt erhöles resultatet i figur 27. Även detta har försumbar inverkan på systemet. Anledningen till att dessa avstånd valdes är att de är 10 % av halva sidans längd och den verkliga avvikelsen borde vara mindre än 10 % i både x- och y-riktningarna. Alltså fungerar reglersystemet även då elektroniken är monterad på prototypen, dock ökar insvängningstiderna och överslängningarna blir aningen kraftigare på grund av den adderade massan.



Figur 26: Reglersystemet då en punktmassa på 0.5 kg placerats mitt på prototypen. Figurerna sker samtidigt. Notera att karakteristiken är snarlik den i figur 19.



Figur 27: Reglersystemet då en punktmassa på 0.5 kg placerats 2 cm framför och 1.2 cm till höger om masscentrum. Figurerna sker samtidigt. Notera att karakteristiken är snarlik den i figur 19, fast med en liten störning för y-hastigheten.

Till sist kan det nämnas att i simulationen antogs vridmomentet från hjulen kunna byta riktning direkt. Detta gäller inte i verkligheten för Sphero Bolt då dess inre massa måste förflyttas kontinuerligt. Detta skulle göra att vid realisering av regelsystemet skulle troligtvis små förändringar av parametrarna behöva göras.

5.2.2 Problem och Användningsområden för Banföljningsalgoritmen

Som beskrivits i resultatet 4.2.2 fungerar algoritmen för de flesta kurvor med rotationshastighet $\omega_Z < \pi/2$ rad/s då hastigheten är mellan ungefär 0.1 m/s till 0.35 m/s. Speciellt så klarar algoritmen trubbiga och räta vinklar vilket innebär att kravet 1.6 i kravspecifikationen är uppfyllt. Däremot är inte önskan 1.8 att algoritmen ska klara alla typer av kurvor uppfyllt, även om vissa framsteg i den riktningen gjorts. Banföljningsalgoritmen har dock endast testats i en Simuleringsmiljö där sensorerna fungerar felfritt. I verkligheten däremot finns det risk att en reflektansmätande sensor kan ge fel värde under ett par stycken klockningar, vilket skulle leda till stora problem med den utvecklade algoritmen. Ett sätt att åtgärda detta skulle möjligtvis vara att använda ett genomsnittligt sensorvärde av ett antal klockningar eller att helt enkelt använda tillräckligt pålitliga sensorer.

Problemen med den nuvarande algoritmen beskrevs i slutet av kapitlet 4.2.2 och dessa behövs åtgärdas innan algoritmen kan bli helt pålitlig. Dessutom är det möjligt att det existerar fler problem med den nuvarande algoritmen som inte upptäckts eftersom de endast uppstår i vissa sällsynta specialfall. Därmed behövs många fler undersökningar och simuleringar genomföras innan en helt färdig algoritm kan framtas.

Ett område där en förbättrad variant av algoritmen kan vara användningsbar är i lagerutrymmen för AGV:er (Automated Guided Vehicle). En AGV med sfäriska hjul skulle möjliggöra banföljning utan att AGV:n behöver rotera kring sin egen axel och kan därmed manövrera trängre lagermiljöer. Visserligen finns det andra tekniker än reflektansmätande sensorer som kan få en AGV att uppnå samma resultat, exempel på detta är att förprogrammera en bana eller användning av en grafisk sensor. Av dessa alternativ har dock det undersökta konceptet med reflektansmätande sensorer och tillhörande algoritm fördelen att vara mer allmän än en förprogrammerad bana samt billigare och enklare än kameror. Den utvecklade banföljningsprincipen skulle också kunna användas i andra studentprojekt där banföljning med godtycklig rotation är önskvärt. Detta på grund av dess simplicitet.

5.3 Samhälleliga och Etiska Aspekter

En viktig faktor som alltid måste beaktas vid nya innovationer och framförallt till nya innovationer inom bilindustrin är säkerheten. Dels säkerheten för passagerarna men även för alla medtrafikanter, både skyddade och oskyddade.

För bilar med sfäriska hjul är inte den konventionella ratten tillräcklig för att styra bilen, vilket skulle innebära att nya körsätt behöver utvecklas. Dessa kommer dock vara mer komplicerade på grund av de utökade frihetsgraderna som riskerar att göra trafiken mer osäker. På grund av den utökade friheten behöver föraren dessutom hålla uppsikt i alla riktningar vilket är svårt. Dessa problem medför att sfäriska hjul i praktiken är beroende

av att fler hjälpmedel implementeras för att avlasta föraren. Ytterligare ett alternativ är att helt självkörande bilar utvecklas för att säkerhet ska kunna upprätthållas. Det är också först med autonoma fordon som sfäriska hjuls fulla potential kan utnyttjas. Med

de utökade frihetsgraderna som en bil med sfäriska hjul besitter borde det vara möjligt att utveckla bättre köregenskaper än för en vanlig bil. Vid sladd eller vattenplaning kan en autonomt styrd bil med sfäriska hjul påverka situationen genom att individuellt styra varje hjul i fördelaktiga riktningar. Detta är något som en förare i en konventionell bil kan göra. Möjligheterna till säkrare körning genom att försöka anpassa styrningen till rådande vägunderlag är något som undersökts i detta arbete. Dock kan det konstateras att det finns potential för att sfäriska hjul kan förbättra säkerheten vid plötsliga olycksrisker.

Beroende på hur en bil med sfäriska hjul framförs kan riskerna för krockar sidledes öka på grund av möjligheterna att köra i alla riktningar. Detta ställer i så fall nya krav på bilens utformning. Traditionellt sett är bilarna mycket sårbara för krockar i sidan eftersom det endast finns en tunn plåt mellan passagerarna och bilens utsida, till skillnad från fronten där hela motorhuven skyddar passagerarna vid krock. Bilar med sfäriska hjul skulle möjligtvis behöva ha fler deformationszoner, krockkuddar eller någon ny innovation för att på så vis skydda passagerarna. Om bilarna blir för breda skulle detta kunna medföra andra problem, som till exempel begränsad framkomlighet då vägar eller parkeringar har en begränsad bredd. En storskalig implementation av bilar med sfäriska hjul, givet att konstruktionen måste vara bred för att involvera till exempel deformationszoner, skulle leda till ett omfattande behov av ändringar i infrastrukturen.

En annan risk med bilar som utöver att vara autonoma också har total frihet i planet skulle vara dess oberäknelighet. Jämfört med en vanlig personbil skulle den autonoma bilen med sfäriska hjul eventuellt kunna uppvisa ett helt annat körmönster vilket skulle kunna förvirra medtrafikanter. Det skulle åtminstone i början uppfattas som väldigt onaturligt när bilen istället för att köra med fronten fram börjar köra sidledes, vilket kan ställa till det för andra i trafiken. Därför är det viktigt att vid utformningen av bilens autonoma system till viss del försöka efterlikna en vanlig bil men detta skulle begränsa fördelarna med de sfäriska hjulen.

Ytterligare en anledning till att bilen med fördel framförs liknande en traditionell bil är en förhöjd olycksrisk under inkörningstiden av det nya styrsystemet. Dessutom kan antalet fall av åksjuka öka om bilen till fullo utnyttjar de ökade frihetsgraderna. Det går möjligtvis att till viss del motverka åksjuka genom sinnrik design av bilens interiör men detta är något som arbetet inte undersökt.

6 Slutsatser

Det framtagna konceptet för prototypen möjliggör körning i alla riktningar med total frihet i planet. Infästningen som tagits fram för det sfäriska hjulet är stabil, håller fast hjulet så att den inte kan trilla ur samt är utformade för att minimera friktion. Dessutom är infästningen utvecklade så att det snabbt går att montera på och av hjulet.

Inget officiellt SDK för Sphero Bolt var utgivet och på grund av projektets tidsbegränsning kunde inget välfungerande kommunikationssystem utvecklas. Därav var kommunikationen med Sphero Bolt strikt begränsad, vilket i sin tur ledde till att tillhörande elektronik inte kunde implementeras.

En banföljningsalgoritm har utvecklats som möjliggör för en simulerad prototyp med sfäriska hjul att följa en bana samtidigt som prototypen roterar godtyckligt kring sin egen z-axel. Detta åstadkoms med åtta reflektansmätande sensorer jämnt placerade i en cirkel kring prototypens centrum. Algoritmen klarar samtliga testade banor och brister endast vid få specialfall.

En simuleringsmodell över systemet har utvecklats som förenklade framtagandet av regler-systemet och banföljningsalgoritmen. Reglersystemet kan skicka individuella signaler till var och ett av hjulen och uppfyller de krav som ställts i kravspecifikationen.

Sammanfattningsvis har projektets essentiella utmaningar lösts med lovande resultat. Dock återstår mycket arbete för att kunna säga något konkret om huruvida storskaliga fordon med sfäriska hjul kan implementeras i samhället. Det skulle därför behövas fortsatta studier både innanför och utanför detta projekts ramar.

6.1 Rekommendationer till Fortsatt Arbete

Till följd av att projektet haft en del avgränsningar och en tidsbegränsning har inte alla delar av projektet undersökts. Denna sektion tar upp några av de delar som ej undersökts men som det rekommenderas att utföra vidare studier inom.

Något som ligger nära projektet och som tiden inte räckte till att göra var färdigställandet av prototypens fulla kapacitet. Vilket innebär montering av sensorer, Raspberry Pi och powerbank på prototypen följt av implementering av styralgoritmer. Detta skapar möjligheter att utforska prototypens funktioner och prestanda i verkliga förhållanden istället för enbart i en simulation. Vidare öppnar det för fortsatta studier inom friktion mellan hjul och mark eller hjul och infästning. Implementation av andra sensorer kan undersökas men i detta projekt skulle reflektansmätande sensorer användas. I vidare studier kan mer avancerade sensorer användas där ett exempel kan vara optiska.

En sak som kan vidareutvecklas från detta projekt är prototypens banföljning som inte blivit realiserad. Med hjälp av sensorerna och den data de skickar kan en styralgoritm utvecklas som kan ta sig igenom vilken bana som helst. För att vidareutveckla banföljningen rekommenderas att undersöka möjligheterna med framåtseende sensorer.

Vidare finns det mycket mer att undersöka gällande infästningarna. Det som tagits fram i detta projekt är bara ett koncept på hur sfäriska hjul kan fästas till ett chassi. På grund av projektets begränsningar kring tid och budget, samt avgränsning att använda Sphero Bolt som hjul har andra möjligheter som magnetisk levitation inte undersökts. Att undersöka möjligheterna med magnetisk levitation är ett område som det rekommenderas att genomföra vidare studier inom. Till detta kan även ett drivsystem för magnetiska hjul utvecklas där energiutvinningen sker inne i bilen istället för i hjulen.

Referenser

- [1] J. Walker, "The Self-Driving Car Timeline – Predictions from the Top 11 Global Automakers", febr. 2019. [Online]. Tillgänglig: <https://emerj.com/ai-adoption-timelines/self-driving-car-timeline-themselves-top-11-automakers/> (hämtad 13 maj 2019).
- [2] Best Movie Cars, "All Cars in "I, Robot" (2004)", mars 2017. [Online]. Tillgänglig: <https://bestmoviecars.com/all-cars-in-i-robot-2004/> (hämtad 2 maj 2019).
- [3] A. Liszewski, "Spherical Wheel Motorcycle Makes It Easier To Dangerously Weave Through Traffic", nov. 2012. [Online]. Tillgänglig: <https://gizmodo.com/spherical-wheel-motorcycle-makes-it-easier-to-dangerous-5942186> (hämtad 2 maj 2019).
- [4] Goodyear, "Goodyear Unveils the Eagle 360 Urban, a Concept Tire Powered by Artificial Intelligence", mars 2017. [Online]. Tillgänglig: <http://news.goodyear.eu/latest-news/goodyear-unveils-the-eagle-360-urban--a-concept-tire-powered-by-artificial-intelligence/s/9da0b0a2-1bf6-4e98-91a4-26430db82749> (hämtad 26 jan. 2019).
- [5] Sphero. (2019). Bot Comparison, [Online]. Tillgänglig: <https://www.sphero.com/education/bot-comparison/> (hämtad 10 maj 2019).
- [6] The Raspberry Pi Foundation. (2018). Raspberry Pi 3 Model B+, [Online]. Tillgänglig: <https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/> (hämtad 10 april 2019).
- [7] DFROBOT. (). Line Tracking Sensor for Arduino, [Online]. Tillgänglig: https://wiki.dfrobot.com/Line_Tracking_Sensor_for_Arduino_V4_SKU_SEN0017 (hämtad 9 mars 2019).
- [8] Elfa Distrelecs. (). SP5K0MAPBKS50P0W - Powerbank 5 Ah,500 cycles,3 h,Vit, Silicon Power, [Online]. Tillgänglig: <https://www.elfa.se/sv/powerbank-ah-500-cycles-vit-silicon-power-sp5k0mapbks50p0w/p/30099731> (hämtad 8 maj 2019).
- [9] "Aluminium", *Wikipedia, the free encyclopedia*, [Online]. Tillgänglig: <https://en.wikipedia.org/wiki/Aluminium> (hämtad 15 mars 2019).
- [10] J. Flynt, "What is the Density of PLA?", *3DINSIDER*, mars 2019. [Online]. Tillgänglig: <https://3dinsider.com/pla-density/> (hämtad 20 mars 2019).
- [11] "Polylactic acid", *Wikipedia, the free encyclopedia*, [Online]. Tillgänglig: https://en.wikipedia.org/wiki/Polylactic_acid (hämtad 15 mars 2019).
- [12] K. Astrom and R. Murray, *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2011.
- [13] "List of moments of inertia", *Wikipedia, the free encyclopedia*, [Online]. Tillgänglig: https://en.wikipedia.org/wiki/List_of_moments_of_inertia (hämtad 11 mars 2019).
- [14] B. Lennartson, *Reglerteknikens grunder*, 4. utg. Studentlitteratur, 2002.

- [15] N. H. Chowdhury, D. Khushi och M. Rashid, "Algorithm for Line Follower Robots to Follow Critical Paths with Minimum Number of Sensors", *International Journal of Computer*, årg. 24, nr 1, s. 13–22, febr. 2017. [Online]. Tillgänglig: <http://ijcjournal.org/index.php/InternationalJournalOfComputer/article/download/819/412> (hämtad 12 maj 2019).
- [16] CDON. (). Reely 1:10 Aluminium-oljetryckstötdämpare Röd (metallic) med fjäder Svart 65 mm 4 st, [Online]. Tillgänglig: <https://cdon.se/hushallsapparater/reely-1-10-aluminium-oljetryckstotdampare-rod-metallic-med-fjader-svart-65-mm-4-st-p34746727> (hämtad 2 maj 2019).
- [17] wematter, "3D-utskrifter i plast – 5 tips för att lyckas!", juli 2015. [Online]. Tillgänglig: <https://wematter.se/3d-utskrifter-i-plast/> (hämtad 28 april 2019).
- [18] 3DVerkstan, "VAD ÄR 3D-PRINTING?", [Online]. Tillgänglig: <https://3dverkstan.se/om-3d-prinm%C3%B6jligghetning/> (hämtad 18 mars 2019).
- [19] hjul-online. (). Ball Transfer Unit, 11.90 mm, with M8 threaded end, for heavy load, [Online]. Tillgänglig: <https://hjul-online.se/ball-transfer-unit-1190-with-threaded-end-for-heavy-load/80041213-p-41213.html> (hämtad 6 maj 2019).
- [20] Osvald Lindholm, *SPHERiCAR Autonomous Line Following Track 1 (Steady Heading)*, 2019, [Online]. Tillgänglig: https://youtu.be/G_8PHQ4WR0k.
- [21] Osvald Lindholm, *SPHERiCAR Autonomous Line Following Track 1 (Spinning)*, 2019, [Online]. Tillgänglig: <https://youtu.be/bsUQAoYpDMk>.
- [22] Osvald Lindholm, *SPHERiCAR Autonomous Line Following Track 2 (Steady Heading)*, 2019, [Online]. Tillgänglig: <https://youtu.be/K9hpRimRZqA>.
- [23] Osvald Lindholm, *SPHERiCAR Autonomous Line Following Track 2 (Spinning)*, 2019, [Online]. Tillgänglig: <https://youtu.be/QVP8DWI2Lzk>.
- [24] Orbotix. (maj 2016). Sphero.js, [Online]. Tillgänglig: <https://github.com/orbotix/sphero.js> (hämtad 18 febr. 2019).
- [25] The Hybrid Group. (okt. 2016). Sphero BLE, [Online]. Tillgänglig: <https://cylonjs.com/documentation/platforms/sphero-ble/> (hämtad 20 febr. 2019).
- [26] K. Szuster. (maj 2016). Kulka, [Online]. Tillgänglig: <https://github.com/karol-szuster/kulka> (hämtad 19 febr. 2019).
- [27] darin-costello. (sept. 2017). Unofficial Python SDK for Sphero, [Online]. Tillgänglig: <https://github.com/darin-costello/spheropy> (hämtad 19 febr. 2019).
- [28] jchadwhite. (febr. 2016). SpheroBB8-python, [Online]. Tillgänglig: <https://github.com/jchadwhite/SpheroBB8-python> (hämtad 19 febr. 2019).
- [29] The Hybrid Group. (2019). SPRK+, [Online]. Tillgänglig: <https://gobot.io/documentation/drivers/sprkplus/> (hämtad 22 febr. 2019).
- [30] The Hybrid Group. (2019). BB-8, [Online]. Tillgänglig: <https://gobot.io/documentation/drivers/bb8/> (hämtad 22 febr. 2019).

Appendix

A Elimineringsmatris

Chalmers		Kesselinngmatris:															
Utfärdare: EENX15-19-18				Skapad:				Modifierad: 2019-02-28									
Kriterier		Alternativ															
		Referenslösning		Koncept 1		Koncept 2		Koncept 3		Koncept 4		Koncept 5		Koncept 6		Koncept 7	
Namn	w	v	t	v	t	v	t	v	t	v	t	v	t	v	t	v	t
Montering	0.13	4	0.52	3	0.39	5	0.65	4	0.52	4	0.52	4	0.52	1	0.13	4	0.52
Demontering	0.13	4	0.52	3	0.39	5	0.65	4	0.52	3	0.39	3	0.39	1	0.13	4	0.52
Belastning	0.08	2	0.16	4	0.32	3	0.24	5	0.4	2	0.16	5	0.4	5	0.4	5	0.4
Vikt	0.17	3	0.51	2	0.34	3	0.51	3	0.51	1	0.17	1	0.17	2	0.34	3	0.51
Stötdämpning	0.06	1	0.06	1	0.06	3	0.18	2	0.12	4	0.24	5	0.3	1	0.06	4	0.24
Pris	0.07	4	0.28	4	0.28	3	0.21	4	0.28	1	0.07	1	0.07	3	0.21	4	0.28
Fiktion	0.19	4	0.76	3	0.57	3	0.57	4	0.76	5	0.95	5	0.95	4	0.76	4	0.76
Tillverkning	0.03	3	0.09	4	0.12	3	0.09	5	0.15	2	0.06	2	0.06	2	0.06	4	0.12
Komplexitet	0.02	4	0.08	4	0.08	4	0.08	4	0.08	3	0.06	1	0.02	5	0.1	4	0.08
Anpassningsbar	0.11	2	0.22	2	0.22	5	0.55	5	0.55	4	0.44	4	0.44	1	0.11	5	0.55
Storlek	0.02	3	0.06	4	0.08	3	0.06	4	0.08	2	0.04	4	0.08	3	0.06	4	0.08
T (Totalt viktat värde)		34	3.26	34	2.85	40	3.79	44	3.97	31	3.1	35	3.4	28	2.36	45	4.06
T / Tideal		1.00	1.00	1.00	0.87	1.18	1.16	1.29	1.22	0.91	0.95	1.03	1.04	0.82	0.83	1.13	1.07
Medel	3.09	0.30	3.09	0.26	3.64	0.34	4.00	0.36	2.82	0.28	3.18	0.31	2.55	0.21	4.09	0.37	0.37
Std-avvikelse	0.83	0.20	0.83	0.13	0.81	0.22	0.55	0.20	1.11	0.21	1.44	0.21	1.32	0.16	0.33	0.33	0.19
Median	3.00	0.22	3.00	0.28	3.00	0.24	4.00	0.40	3.00	0.17	4.00	0.30	2.00	0.13	4.00	0.40	0.40
Antal svaga punkter	3		3		0		1		5		4		6		0		
Rangordning	5		7		3		2		6		4		8		1		
Beslut	Går vidare med koncept 7 då den erhöill högst betyg och inte har några svaga punkter																

B Undersökning av Kommunikation med Sphero Bolt

Det första steget i undersökningen av hjulens möjligheter att programmeras var att utforska den utvecklingsmiljö som Sphero har designat. Det framgick att miljön inte gav möjlighet till kommunikation med mer än ett hjul i taget. Vilket medförde att andra sätt att kommunicera undersöktes. De alternativ som började undersökas var redan existerande API:er som har fungerat för att kommunicera med föregående Sphero-produkter. Huvudsakligen är det tre programmeringsspråk som har undersökts. Dessa är JavaScript med hjälp av NodeJS, Python och Golang.

För JavaScript fanns två API:er, sphero.js och Cylon.js, där det förstnämnda är det officiella SDK:et från Sphero och det sistnämnda är en utbyggnad på den officiella API:n [24], [25]. Båda gav i grunden samma felmeddelanden, vilket var att API:n kunde varken hitta eller identifiera enhetens karakteristiker (characteristics).

```
1 sudo node ./src/sphero.js
2 /home/pi/spherojs/node_modules/sphero/lib/adaptors/ble.js:405
3   return this._connectedService(serviceId).characteristics;
4   ~
5 TypeError: Cannot read property 'characteristics' of null
```

Listing 1: Felmeddelande från sphero.js SDK,
för fullständigt meddelande se appendix B.1.

För Python hittades och testades tre olika inofficiella API:er vilket var Kulka, spheropy och SpheroBB8-python [26]–[28]. De två förstnämnda kompilerade och körde utan felmeddelanden men skripten visade sig inte fungera. Utan felmeddelanden gick det inte att identifiera felet därför undersöktes även den sistnämnda av API:erna. Vilket var den senast utvecklade och var ämnat för Sphero BB-8, en robot som även den använder sig av BLE-kommunikation. Vid test av detta API gav det felmeddelande där problemet ligger i identifikationen av enhetens karaktäristik.

```
1 python BB8test.py
2 Traceback (most recent call last):
3   [...]
4   self.antidos = self.getSpheroCharacteristic('2bbd')
5   File "/home/pi/SpheroBB8-python/BB8_driver.py", line 160, in getSpheroCharacteristic
6     return self.peripheral.getCharacteristics(uuid='22bb746f' + fragment + '75542d6f726568705327')[0]
7   File "/usr/local/lib/python2.7/dist-packages/bluepy/btle.py", line 508, in getCharacteristics
8     rsp = self._getResp('find')
9     [...]
```

Listing 2: Felmeddelande från SpheroBB8-python API,
för fullständigt meddelande se appendix B.2.

Gobot är ett Golang-baserat API för många tekniska enheter. Det finns stöd för BB-8 och Sphero SPRK+ [29], [30]. Dessa använder sig av BLE och därför testades möjligheten att använda Gobots API ämnat för Sphero SPRK+. Resultatet av detta blev likt tidigare test av Python-skripten, det vill säga utan felmeddelande men ingen kontakt med hjulen etablerades.

På grund av uteblivna resultat från tidigare tester söktes hjälp från Sphero samt andra fristående utvecklare på Github. Responsen från Sphero bidrog med information om att en ny SDK för deras nya robot, RVR, var under utveckling. De kunde inte garantera stöd för Sphero Bolt samt att den planerades att släppas under hösten 2019. Istället gavs rekommendationer att kontakta utvecklaren bakom det inofficiella Javascriptet spherov2.js. Vilket visade sig mycket framgångsrikt då utvecklaren gick med på att hjälpa till utifrån bästa förmåga. Han var redan medveten om att hans API inte fungerade med Sphero Bolt, men kunde själv inte undersöka och åtgärda det då han saknade tillgång till en Sphero Bolt.

B.1 Felmeddelande från spherov2.js SDK

```
1 sudo node ./src/sphero.js
2 /home/pi/spherojs/node_modules/sphero/lib/adaptors/ble.js:405
3   return this._connectedService(serviceId).characteristics;
4   ^
5 TypeError: Cannot read property 'characteristics' of null
6     at Adaptor._connectedCharacteristics (/home/pi/spherojs/node_modules/sphero/lib/adaptors/ble.js:405:43)
7     at Adaptor._connectCharacteristic (/home/pi/spherojs/node_modules/sphero/lib/adaptors/ble.js:358:12)
8     at /home/pi/spherojs/node_modules/sphero/lib/adaptors/ble.js:303:12
9     at /home/pi/spherojs/node_modules/sphero/lib/adaptors/ble.js:346:9
10    at Peripheral.<anonymous> (/home/pi/spherojs/node_modules/noble/lib/peripheral.js:73:7)
11    at Peripheral.g (events.js:260:16)
12    at emitOne (events.js:77:13)
13    at Peripheral.emit (events.js:169:7)
14    at Noble.onServicesDiscover (/home/pi/spherojs/node_modules/noble/lib/noble.js:251:16)
15    at emitTwo (events.js:87:13)
16    at emit (events.js:172:7)
17    at NobleBindings.onServicesDiscovered (/home/pi/spherojs/node_modules/noble/lib/hci-socket/bindings.js:305:8)
18    at emitTwo (events.js:87:13)
19    at emit (events.js:172:7)
20    at null.<anonymous> (/home/pi/spherojs/node_modules/noble/lib/hci-socket/gatt.js:361:12)
21    at Gatt.onAclStreamData (/home/pi/spherojs/node_modules/noble/lib/hci-socket/gatt.js:132:26)
```

B.2 Felmeddelande från SpheroBB8-python API

```
1 python BB8test.py
2 Traceback (most recent call last):
3   File "BB8test.py", line 8, in <module>
4     bb8.connect()
5   File "/home/pi/SpheroBB8-python/BB8_driver.py", line 258, in connect
6     self.bt = BTInterface(self.deviceAddress)
7   File "/home/pi/SpheroBB8-python/BB8_driver.py", line 144, in __init__
8     self.antidos = self.getSpheroCharacteristic('2bbd')
9   File "/home/pi/SpheroBB8-python/BB8_driver.py", line 160, in getSpheroCharacteristic
10    return self.peripheral.getCharacteristics(uuid='22bb746f' + fragment + '75542d6f726568705327')[0]
11   File "/usr/local/lib/python2.7/dist-packages/bluepy/btle.py", line 508, in getCharacteristics
12     rsp = self._getResp('find')
13   File "/usr/local/lib/python2.7/dist-packages/bluepy/btle.py", line 407, in _getResp
14     rsp = self._waitResp(wantType + ['ntfy', 'ind'], timeout)
15   File "/usr/local/lib/python2.7/dist-packages/bluepy/btle.py", line 368, in _waitResp
16     raise BTLEGattError("Bluetooth command failed", resp)
17 bluepy.btle.BTLEGattError: Bluetooth command failed (code: 10, error: No attribute found within the given range)
```

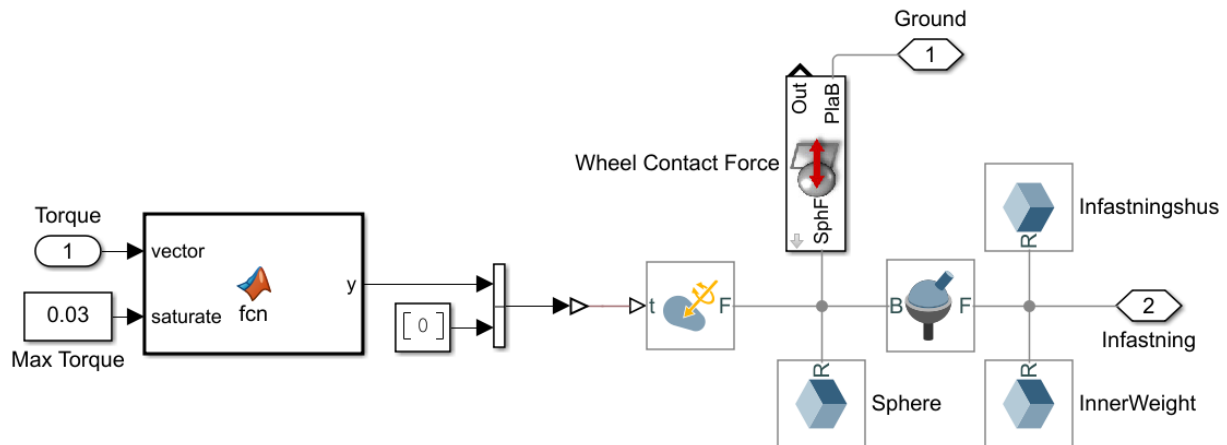
B.3 Konsolmeddelande från Gobot API

Gobot ger inget felmeddelande men fastnar utan framsteg. Efter att man stoppar processen BLE kommandon längre. Detta kan fixas genom att starta om datorn.

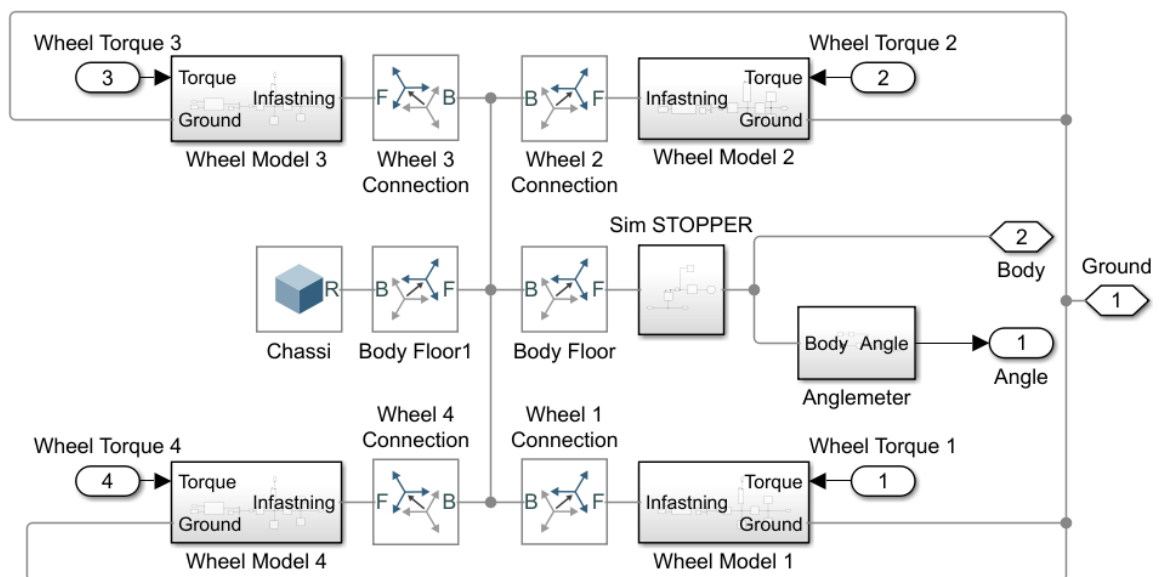
```
1 sudo ./sprkplus SB-EE1B
2 2019/02/22 10:24:42 Initializing connections...
3 2019/02/22 10:24:42 Initializing connection BLEClient-63AA2859 ...
4 2019/02/22 10:24:42 Initializing devices...
5 2019/02/22 10:24:42 Initializing device SPRKPlus-52627B34 ...
6 2019/02/22 10:24:42 Robot sprkBot initialized.
7 2019/02/22 10:24:42 Starting Robot sprkBot ...
8 2019/02/22 10:24:42 Starting connections...
9 2019/02/22 10:24:42 Starting connection BLEClient-63AA2859...
```

C Modeluppbyggnad i Simscape

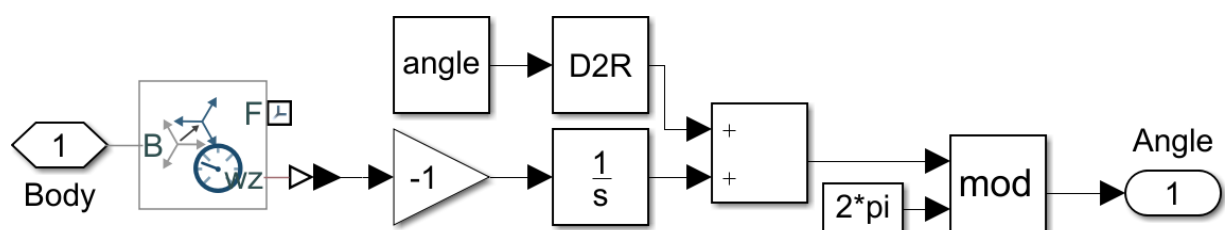
C.1 Hjul och Infästningsmodell



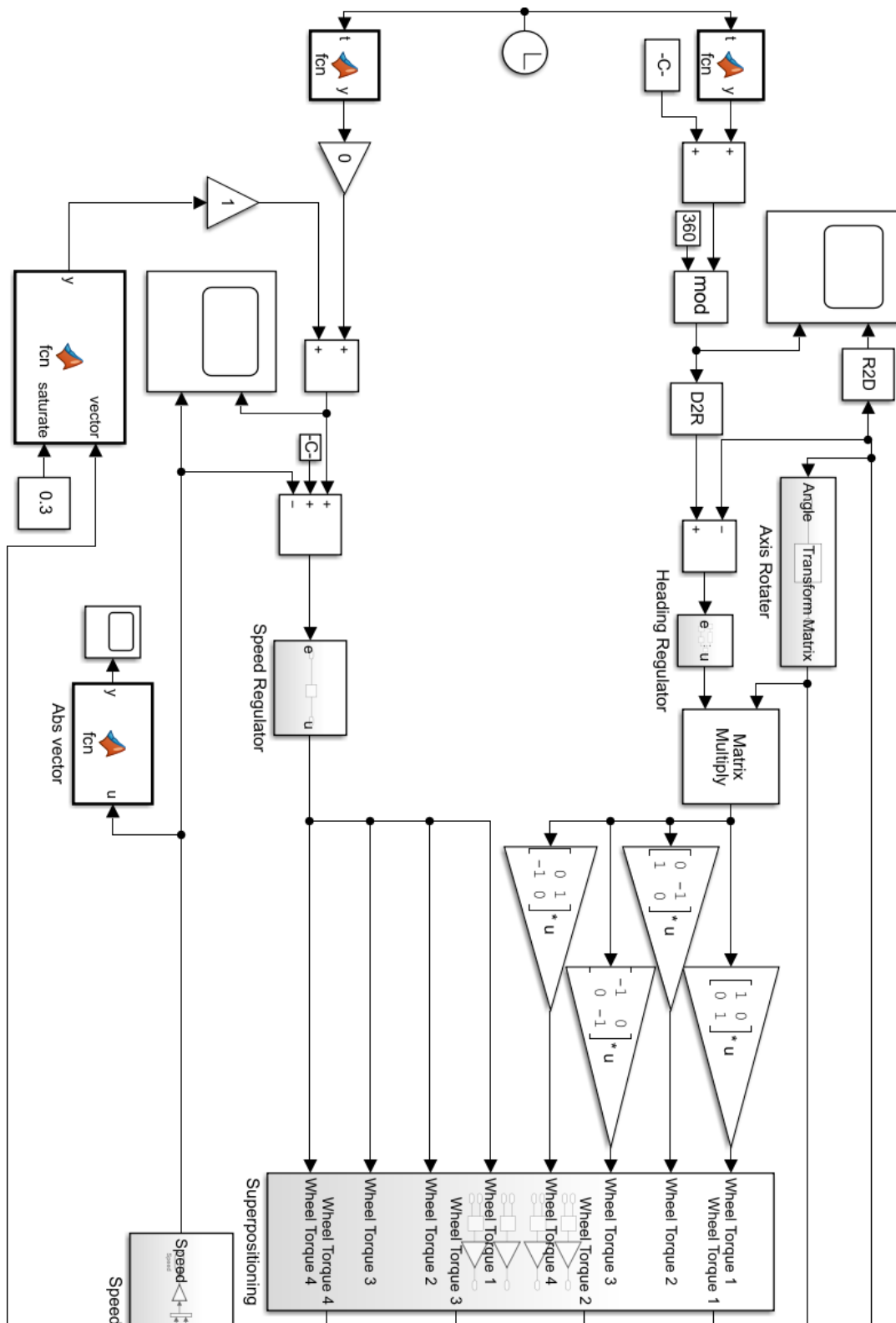
C.2 Prototypsmodell



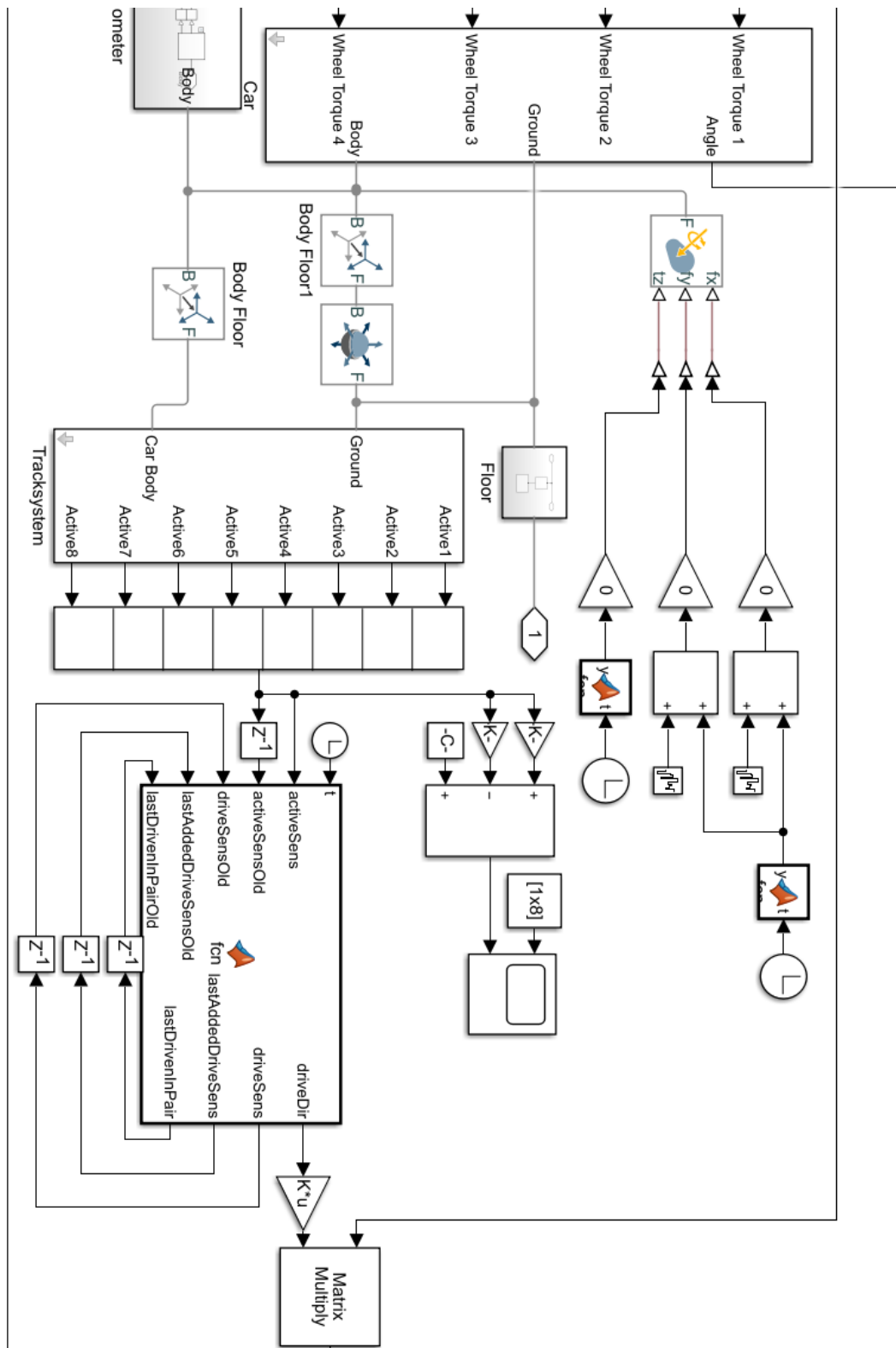
C.3 Anglemeter



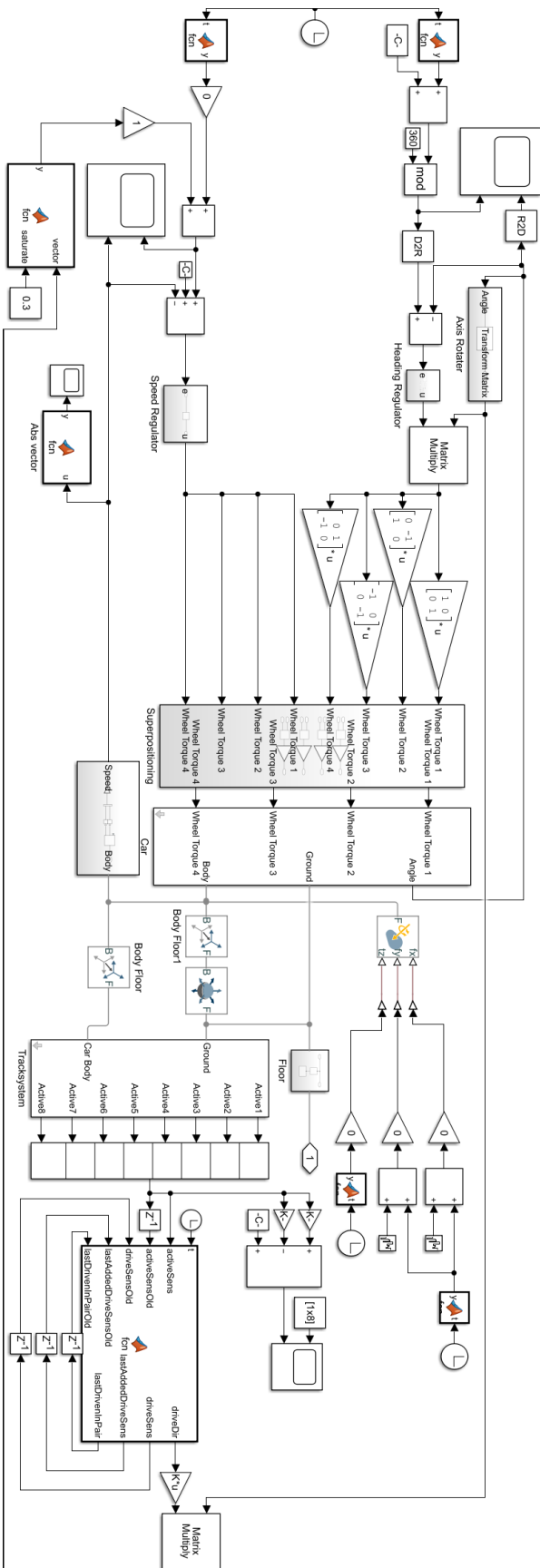
C.4 Helhetsmodel (vänster)



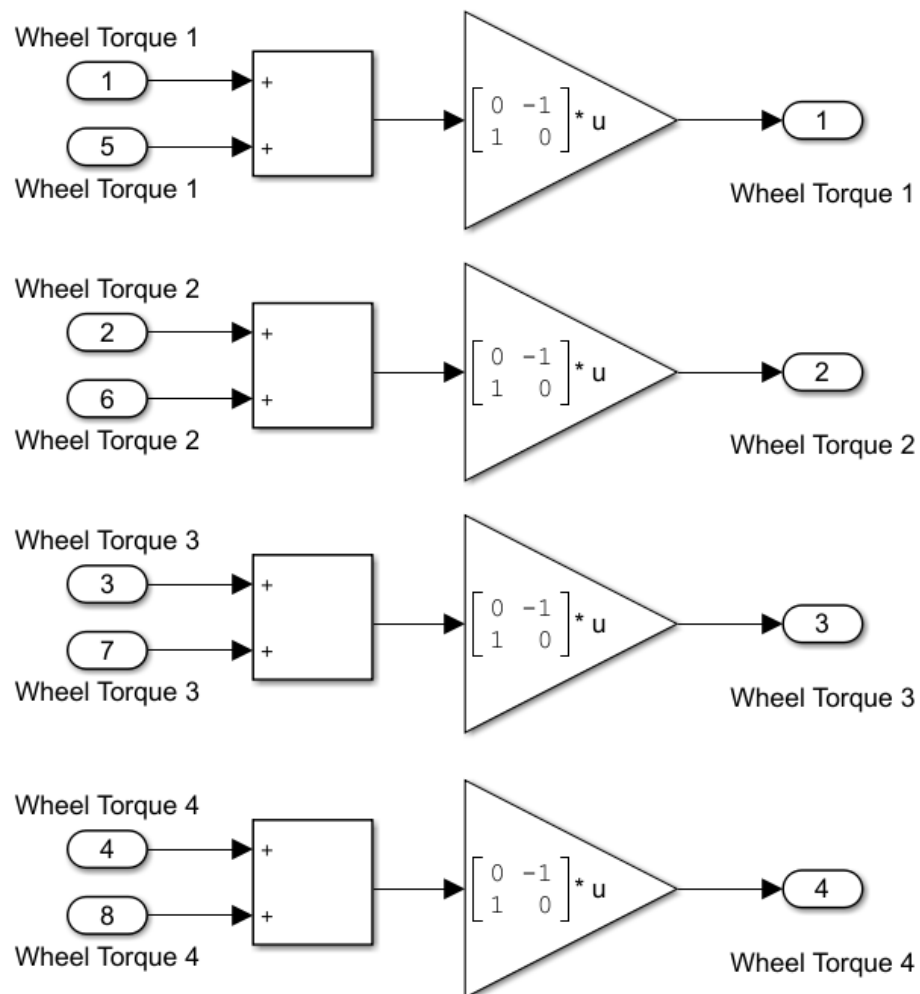
C.5 Helhetsmodel (höger)



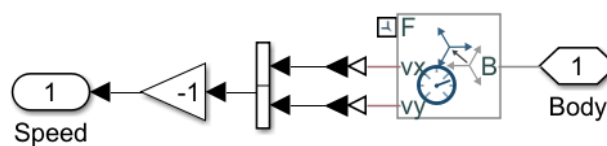
C.6 Helhetsmodel (hel)



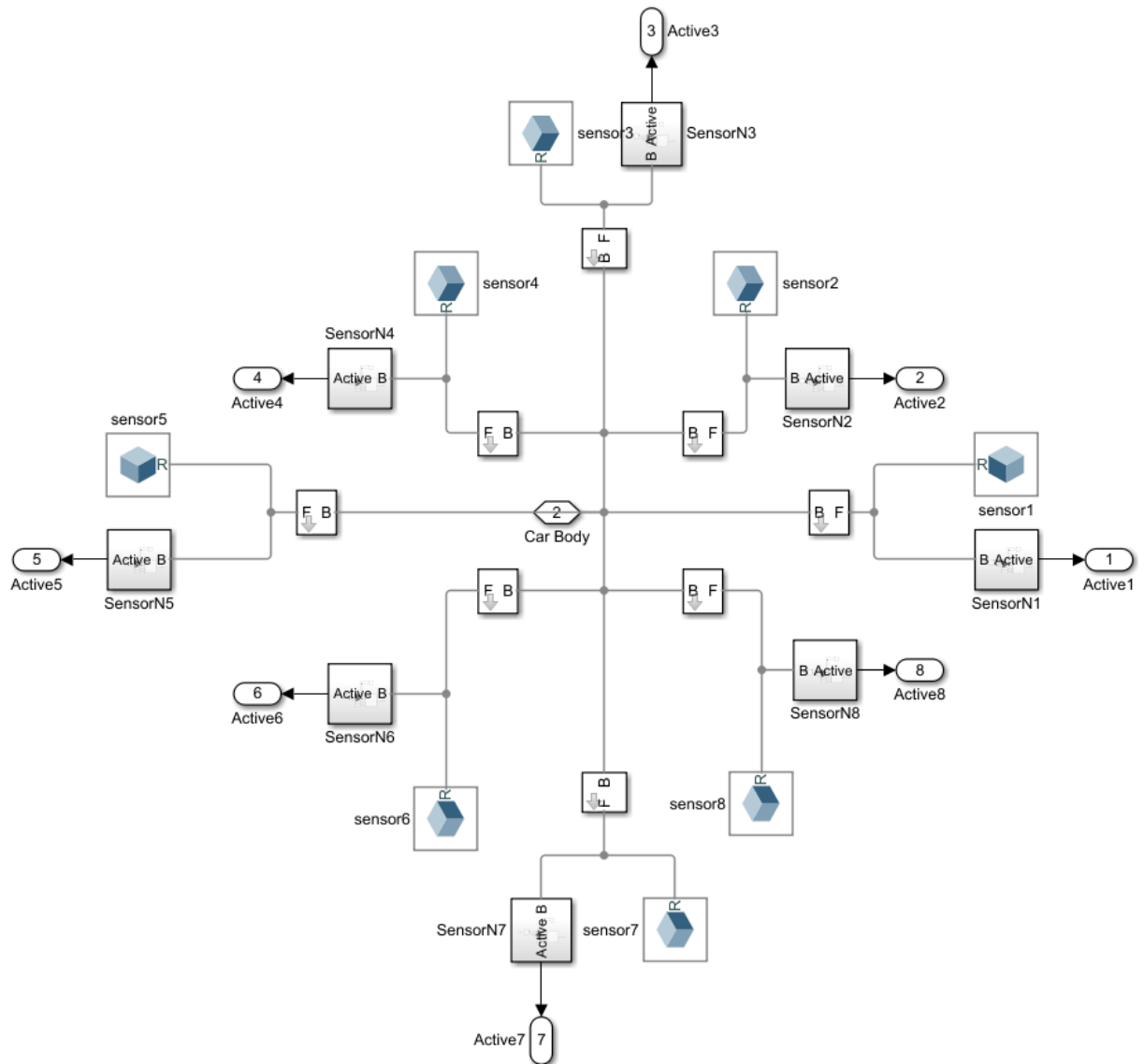
C.7 Superpositonering



C.8 Speedometer

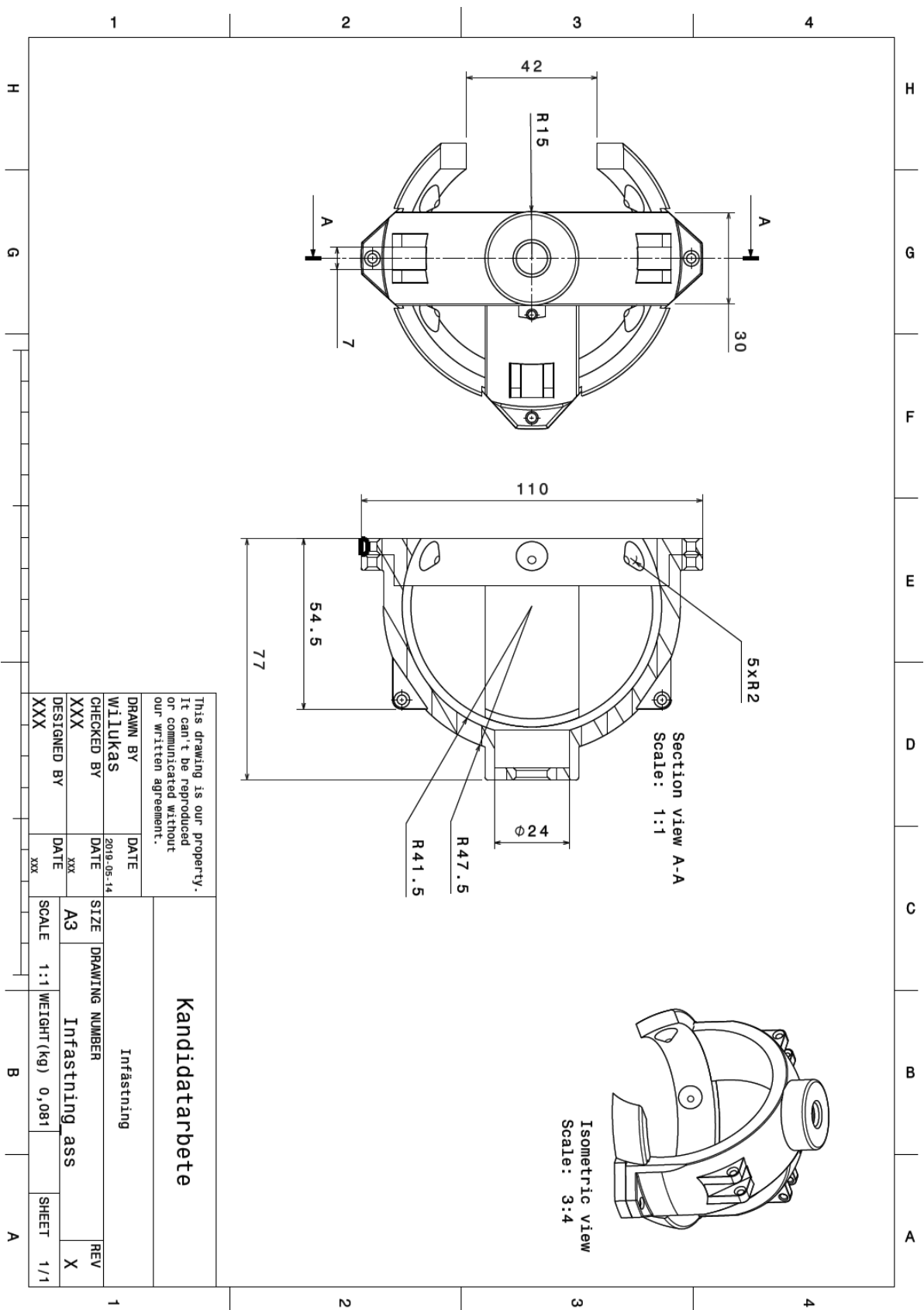


C.9 Sensormodel

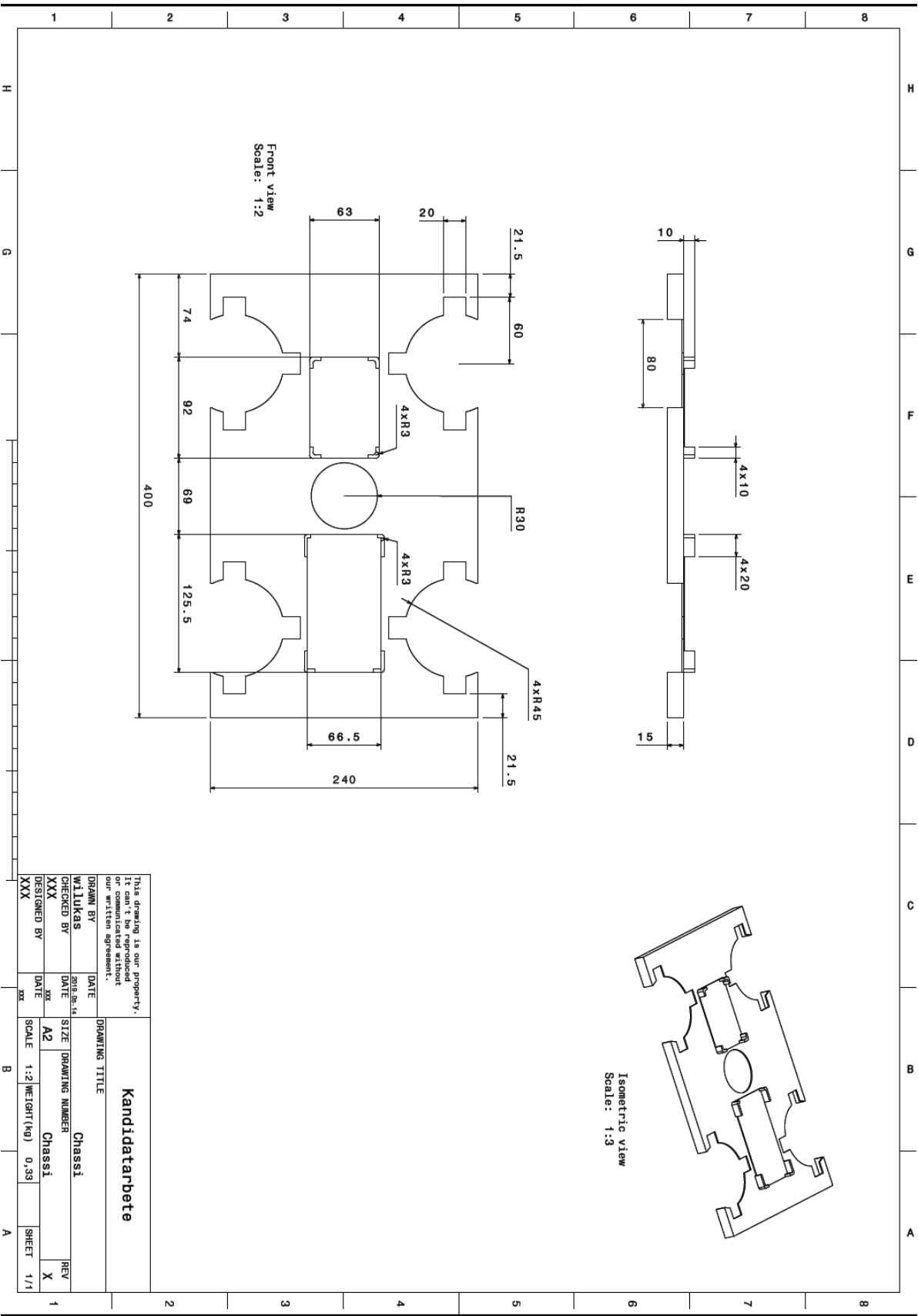


D Detaljritningar

D.1 Infästning



D.2 Chassi



E Matematiska Beräkningar

I detta avsnitt presenteras några av de beräkningar som gjorts för en bil med sfäriska hjul. Syftet är att dels visa vissa intressanta resultat samt vara till hjälp för framtida studier inom området.

E.1 Matematisk Modell för Translation och Rotation

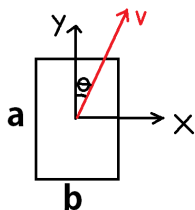
Den matematiska modellen syftar till att relatera hastighetsvektorn från bilens mittpunkt till hjulens rotation, både till riktning och storlek. Dessutom skall hjulens rörelse relateras till bilens rotation kring dess mittpunkt. Bilens totala rörelse fås sedan som en superponering av rotationen och translationen.

För translation med hastighet v och vinkeln θ mot y-axeln, se figur 28, ska hastigheten längs med x-axeln vara $v_x = v \sin \theta$ och längs y-axeln $v_y = v \cos \theta$. Hjulens rotation kring x-axeln blir därmed

$$\omega_x = \frac{v_y}{2r\pi} 2\pi = \frac{v \cos \theta}{r} \text{rad/s},$$

där r är hjulets diameter. På samma sätt blir

$$\omega_y = \frac{v \sin \theta}{r} \text{rad/s}.$$



Figur 28: Koordinatsystemet för bilen.

För rotation kring mittpunkten (z-axeln) ska rotationsaxeln för hjulet längs upp till vänster (se figur 28) vara

$$P = -\frac{b}{2}\hat{x} + \frac{a}{2}\hat{y},$$

där a och b är bilens sidlängder definierade enligt figur 28. Om bilens rotation ska vara θ rad medurs ska hjulet röra sig i en cirkelbåge med vinkel θ och radie

$$t = \sqrt{\left(\frac{b}{2}\right)^2 + \left(\frac{a}{2}\right)^2}.$$

Längden på cirkelbågen är $t\theta$, vilket motsvarar

$$\frac{t\theta}{2r\pi}$$

varv för hjulet. Alltså ska hjulet längst upp till vänster rotera

$$\frac{t\theta}{r} \text{ rad}$$

kring rotationsaxeln P när prototypen ska rotera θ rad medurs (kring negativa z-axeln). Analog räkning ger rotationerna för de tre övriga hjulen.

E.2 Maximal Acceleration Beroende på Friktion

Antag att prototypens vikt är m kg vilket medför att normalkraften från hjulen totalt måste motsvara $N_{total} = mg$. Antag vidare att vikten är fördelad lika på alla hjul vilket ger normalkraft per hjul

$$N = \frac{mg}{4}.$$

Friktionen mellan hjul och underlaget är maximalt $F_f = \mu N$ och måste motsvara

$$F_f = \frac{ma_{max}}{4},$$

där a_{max} är bilens maximala translationsacceleration. Alltså gäller

$$\mu \frac{mg}{4} = \frac{ma_{max}}{4}$$

vilket ger $a_{max} = \mu g$. Om försök görs att accelerera snabbare än a_{max} kommer något eller några av hjulen börja slira vilket kommer leda till att bilen börjar rotera och förlora kontroll. Därför bör marginal till a_{max} hållas.

Eftersom bilen ska kunna köras säkert i alla riktningar är det viktigt att masscentrum är så nära mitten av bilen som möjligt. Om detta inte skulle vara fallet finns det en risk att bilen börjar rotera vid kraftig inbromsning då krafterna som krävs för homogen inbromsning varierar mellan hjulen. Detta är förvisso något som ska regleras men det är ändå en stor fördel om masscentrum korrelerar relativt väl med dess geometriska centrum. Vid beräkning fanns att ojämn massfördelningen inte påverkade maxaccelerationen genom behov av ökad friktionskraft eftersom ökad normalkraft för ett hjul ger högre möjlig friktion men samtidigt mer massa att accelerera. Dessa effekter tar ut varandra. Dock krävs mer kraft från hjulen om normalkraften på detta hjul ökar, och häri föreligger en begränsning. Alltså kommer en bil med masscentrum i dess geometriska centrum att förutom vara säkrast också kunna accelerera snabbast.

En annan orsak till att masscentrum bör placeras så nära mitten som möjligt är på grund av tröghetsmomentet. Lagen för rotation säger

$$M = I\dot{\omega}$$

där $\dot{\omega}$ är vinkelaccelerationen, M är momentet kring rotationspunkten och I är tröghetsmomentet. Eftersom

$$I = \int mr^2 dr$$

minskar tröghetsmomentet om massan är placerad så centrerat som möjligt och då krävs mindre moment för rotation.

F Övrigt

F.1 Projekttes SPHERiCAR

SPHERiCAR - Take the ball and go with it!

Background

Unlike regular wheels that only move forward or backward, **spherical wheels** allow the car to move fluidly in all directions and contributes to comfort, safety and maneuverability to match the demands of autonomous mobility and mobility as a service. The shape also copes with space limitations in smart cities (smaller parking spaces, platoon driving, maneuvering, easy to replace...).



A car can do much more on these tires than it can on today's version, but doing so also demands an insane amount of **control**. When four wheels go in basically the same direction, a person can control them with a steering wheel. When they can go in any direction, you need a **computer** and **control algorithms** to manage that. Now that our cars are about to drive themselves, it's high time to see what's possible.

Problem description

The main goal is to **design** and **build** a *remote controlled* vehicle with 4 spherical wheels (using Sphero robotic balls). Furthermore:



- from the vehicle's 3-degree of freedom (x, y, rotation), develop a mathematical model of the vehicle and describe equations of the wheels.
- Design a control algorithm for motion/speed control of the vehicle.
- Implement the control algorithm on the vehicle.

Målgrupp:	TKAUT, TKMAS, TKELT, TKDAT, TKTFY,
Gruppstorlek:	3-6
Antal grupper:	1-2
Förkunskapskrav:	Reglerteknik
Kontaktperson:	Adrian Ilka, adrian.ilka@chalmers.se

(Handledare och Examinator kan meddelas senare när projekten är tilldelade)