

Universitetet i Sørøst-Norge
Fakultet for teknologi og maritime fag

Bacheloroppgave
2022

Cecilie Sunde, Una Kaspara Lier, Vetle Bråthe Lund, Mina Hildebrand
Fredriksen, Alison Wartenberg, Angelika Warzocha

SoilRover

SoilRover

Bachelor oppgave Våren 2022

SFHO3201-1



Sammendrag.

Soil Steam International produserer maskinerier for å forbedre jordkvalitet uten bruk av kjemikalier. De utvikler nå et autonomt kjøretøy, SoilRover, som damper jorden på åkre. Dette kjøretøyet har uvanlige funksjonaliteter, som en heisemekanisme og fullt individuelt roterende hjul økt bevegelighet. I denne rapporten blir det beskrevet hvordan vi løste oppgaven om å lage en nedskalert modell av SoilRover fysisk og simulert. De ønsker dette slik at de kan teste bevegelser uten store konsekvenser og kan visualisere sitt produkt til potensielle kunder. Det er lagt opp for at vårt produkt skal bli styrt med PS4 kontroller eller tilhørende applikasjon.

Vi har laget ferdig design av modellen og noen ferdig produserte deler, samt utviklet prototyper hvor vi har testet enkelte av egenskapene. Vi har fått kjøretøyet til å kjøre samt rotere individuelt. Dessuten en snart ferdig heisemekanisme. I tillegg har vi en app som har mulighet til å gi og få informasjon og en simulasjon som klarer å simulere de forskjellige egenskapene. Simulasjonen vår ble utviklet i Unity og applikasjonen i Android Studio. Produktet oppbygd av kryssfine plater, 3D print og kompositt. For styring av SoilRover har vi brukt Ackermann styring og PID reguleringer implementert i mikrokontrolleren Arduino. Ved innlevering av denne rapporten har vi ikke en fullstendig SoilRover men et godt grunnlag. Nødvendige endringer som er nevnt i rapporten vil føre til et ferdig produkt.

Medlemmer SoilSprouts

Cecilie Sunde

Maskiningeniør

Prosjekt leder

Deksel og kropp



Una K. Lier

Maskiningeniør

Armmodul



Vetle Bråthe Lund

Elektroingeniør

Rotasjon og batteri

Økonomi



Mina Hildebrand

Fredriksen

Elektroingeniør

Driv og heis

PowerPoint



Alison Wartenberg

Dataingeniør

App og styring

Scrumban-Master



Angelika Warzocha

Dataingeniør

Simulasjon og nettside



Innhold

Sammendrag	II
Medlemmer SoilSprouts	II
1 Innledning	2
2 Prosjekt organisering	4
2.1 Prosjekt modell	4
2.1.1 Daglige møter	5
2.1.2 Iterasjonsmøter	5
2.1.3 Bøtte planlegging	5
2.1.4 Tavlen	6
2.1.5 Prioritering	8
2.1.6 “On-demand” planlegging	8
2.1.7 Team	8
2.1.8 Funksjons frys	9

2.1.9 Møter	9
2.1.10 Interne veiledningsmøter	9
2.1.11 Møter med bedriften	10
2.1.12 Internt i gruppa	10
2.1.13 Kontakt	10
2.1.14 Bedriftens kontakt	10
2.1.15 Intern veileder	11
2.1.16 Internt i gruppa	11
2.1.17 Diskusjoner i gruppa	11
2.1.18 Endringer og kommunikasjon	12
2.2 Prosjekt hjelpemidler	14
2.3 Prosjekt plan	16
2.4 Økonomisk	17
2.5 Risiko analyse	22
2.6 Spesifikasjoner	27

3	Produksjonsmetoder	29
3.1	3D-Print	29
3.1.1	Design for 3D printing	30
3.2	Støping	32
3.3	Laserkutting	33
4	Kropp	34
4.1	Boks til rammeverk	34
4.2	Innside	43
4.2.1	Oversikt oppsett av komponenter i kropp	43
4.2.2	Mikrokontroller	44
4.2.3	Flere Mikrokontroller?	46
4.2.4	Batteri	46
4.2.5	Elektrisk oppsett	47
4.2.6	Spenningsregulering	48
4.2.7	Heisemekanisme	52

4.3 Dekse	54
5 Armmodul	57
5.1 Hjul modul	57
5.1.1 Valg av hjul	57
5.1.2 Valg av driv motor	58
5.1.3 Mekanisk design av hjulmodul	63
5.1.4 Implementasjon	67
5.2 Arm	68
5.2.1 Rotasjonsmotorer	68
5.2.2 Enkoder for rotasjon	71
5.2.3 Mekanisk design av Arm	73
5.2.4 Ackermann	76
6 Grensesnitt	80
6.1 Grensesnitt i datasystemer	80

6.1.1	Fysisk sammenkobling	80
6.1.2	Trådløs kommunikasjon	84
6.2	Brukergrensesnitt	88
6.2.1	LED	88
6.2.2	Display	89
6.2.3	APP	91
6.2.4	Kontroller	92
7	Prototyper	93
7.1	Prototype Driv	93
7.1.1	Design og oppsett prototype driv	94
7.1.2	Feilsøking og forbedringspotensiale	95
7.2	Prototype sensor boks	95
7.2.1	Design av prototype sensor boks	95
7.2.2	Kode og test av det vi vil den skal gjøre	96
7.2.3	Feilsøking og forbedringspotensiale	97

8 Applikasjon	101
8.1 Formål	101
8.2 Funksjonalitet	101
8.3 Design	102
8.4 Verktøy	107
8.5 Språk	107
8.6 Tekniske løsninger	107
8.6.1 Model-view-viewmode	107
8.6.2 Singleton	108
8.6.3 Bluetooth	108
8.6.4 Joystick	108
8.6.5 PDF-Reader	109
9 SoilRover Simulator	110
9.0.1 Verktøy	110
9.1 Modell	111

9.1.1	Prototype	111
9.1.2	Nåværende modell	112
9.2	User view	112
9.2.1	Styring	113
9.2.2	Simulasjonsmodell	116
9.2.3	User interface	117
9.2.4	Kollisjon deteksjon	123
10	Oppsummering og framtidige ideer	126
10.1	Oppsummering	126
10.2	Forbedringspotensiale	127
10.2.1	Resterende planer	128
11	Konklusjon	130
A	Risiko og spesifikasjoner	142
B	Kropp	162

C Armmodul dokumentasjon	174
D Elektrisk kapasitet	183
E Styring	191
F Teknisk løsning for simulasjon.	197
F.1 Modell	197
F.1.1 Prototype	197
F.2 Nåværende 3D modell	198
F.3 Bevegelser	204
F.4 Colliders	205
F.4.1 Wheel colliders	206
F.5 Kollisjon deteksjon	208
F.6 Kjøremoduser	210
F.6.1 Default kjøremodus	210
F.6.2 F1 modus	211

F.6.3	F2 - Valgt radius	211
F.6.4	F3 - framhjul	211
F.6.5	F4 - bakhjul	212
F.6.6	F5 - hvert hjul	212
F.7	Verden	213
F.8	User Interface	215
F.9	Oppsummering	218
G	GANTT-diagrammer	221
G.1	GANTT-diagram v.1	221
G.2	GANTT-diagram v.2	223
G.3	GANTT-diagram v.3	225

Figurer

1	Scrumban	4
2	Figur viser oppsettet av tavlen vår	7

3	Figur som viser funksjonstreet vårt	13
4	Utklipp av oversikt over økonomi	21
5	Styrken av 3D printet del med tanke på lag og retning av kraft.[7]	30
6	Innfyll prosent i 3D print.[8]	31
7	Innfyll mønster i 3D print.[9]	31
8	SoilRover med kropp uthevet i gull	34
9	Boks første versjon, simplifisert illustrasjon for å vise designet til 2.presentation	35
10	Første versjon av rammeverk som ble designet for å 3D printes.	36
11	Rammeverk versjon 2 som ble designet for å kunne laserkutte deler i kryssfi- nerplater	36
12	SoilSteam International sitt desin av rammeverk.	37
13	SoilRover fullmodell	38
14	Skulder er uthevet i gull	39
15	Feste til heiseveggen er festet til heiseskruen, den hvite delen som er på heise- skruen.	39
16	FEM analyse av feste for heiseveggen, med deformasjons skala på 1 og 50	41

17	Mesh av heisevegg feste versjon 3 for analyse	42
18	Tenkt oppsett for topp plate av kropp	43
19	Tenkt oppsett for under plate av kropp	44
20	Av/på-bryter skaffet fra Biltema	47
21	Diagram oppkobling knapp	48
22	Kjøleribbe(eng:Heatsink)	49
23	DC-DCkonverter som er handlet fra electronikit	51
25	Designet som ble valgt sammen med SSI, et karbonfiber deksel med spraylakk av SSI sin ønsket farge	55
26	Dekselformen ferdig limt, trenger å pusses ned før kan spray lakkes	56
27	Armmodul	57
28	Armmodul med arm uthevet i gult	57
29	Armmodul med hjulmodul uthevet i gult	57
30	Kjøpte hjul	58
32	Hjulmodul	63
33	3D-modell av armmodul	63

34	Seksjonsbilde av hjulmodul og rotasjonsledd	63
35	Seksjonsbilde av hjulfestet	65
36	Hjulmodul	66
37	3D-modell av armmodul	66
38	Seksjonsbilde av hjulmodul og rotasjonsledd	66
39	Hjulmodul	67
40	3D-modell av armmodul	67
41	Seksjonsbilde av hjulmodul og rotasjonsledd	67
43	Venstre bilde viser enkoder signalene når motor går mot klokka, og høyre figur viser motoren går med klokka.	73
44	Rotasjonsledd	74
45	Seksjonsvisning av låsemekanisme	75
46	Seksjonsbilde av låsemekanisme og skulder	75
47	Figur som viser til konseptet av 4-hjuls Ackermann	77
48	Siste design med lengder for Ackermann	78
49	Figur som viser sammenkobling mellom 2 Arduinoer	81

50	Bilde som viser hvordan to Arduino er koblet sammen	82
51	Bilde som viser testing av Ultralydsensor med to Arduino som er koblet sammen	83
52	Figur som viser hvordan USB-Host shield er koblet til Arduino	84
53	Figur som viser fysisk hvordan USB-Host shield er koblet til Arduino	85
54	Figur som viser hvordan HC-05 er koblet til Arduino	86
55	Figur som viser fysisk hvordan HC-05 er koblet til Arduino	87
56	Figur som viser fysisk hvordan LED-stripa er koblet til Arduino	89
57	Figur som viser fysisk hvordan LCD-skjermen er koblet til Arduino	91
58	Figur viser oppsettet av knappene på PS4-kontrolleren	92
59	Ferdig lagd driv prototype	93
60	Figur som viser konseptet til prototype 1	94
61	Lite overflate mellom hjul og gulv	94
62	LED-stripen med nye kabler	97
63	Signalene skaffet fra pwm pins med en gitt verdi på 125. Øverst er pwm pin koblet til motor 1, neste til motor 2, og så videre	100
64	Alle pwm signaler er nå like, på gitt verdi 150	100

65	Bilder viser startside til appen	102
66	Bilde viser fjernstyringside til appen	103
67	Bilde viser hjulstyringside til appen	104
68	Bilde viser siden som viser manualen	105
69	Bilde viser siden som viser QR-kode og knappene for å komme til nettsidene	106
70	Prototype av SoilRover med hjul heist opp og ned	111
71	3D modell i hovedposisjon	112
72	Kjøremodus	115
73	Damp funksjon	116
74	Start menu	118
75	Controls	118
76	Pause menu	119
77	Heads-up display	120
78	Hjulene - HUD	121
79	Moduser	121

80	Enter radius vindu	122
81	Modus F5 - 0 grader posisjon	122
82	Modus F5 - 0 grader posisjon	123
83	Rammeverk V1	162
84	Rammeverk V2	163
85	Bakre skulder festet til heisevegg	164
86	Glassfiber plate test 1 (venstre) og 2 (høyre)	165
87	Test 4 tatt av glassplaten	166
88	Test 4 tatt av vedsiden av glassplaten	167
89	Dekselform og deksel	168
90	Design for deksel	169
91	Pusset og ikke pusset 3D printede endedeler for dekselet. Venstre del på bildet er pusset og høyre del er ikke pusset	169
92	Endeform får flensen limt på og holdt sammen med klemmer	170
93	Første versjon av heisevegg festet	171
94	Heisevegg feste versjon 2	172

95	Heisevegg festet versjon 3 og det endelige designet.	173
96	Konsept fra før presentasjon 2	174
97	Armmodul	174
98	Konsept til hjulmodul	174
99	Hjulmodul ver.1	175
100	Hjulmodul ver.2	175
101	Hjulmodul ver.3	175
102	Hjulmodul ver. 4	175
103	Armmodul ver.1	176
104	Armmodul ver.2	176
105	Aksel ver. 1	177
106	Aksel ver. 3	177
107	Siste versjon av aksel	178
108	Armbunn ligger til tørk	179
109	Prototype av boks med alle armmoduler	180

110 Første hele prototype av armmodul	180
111 Siste versjon av armmodulene	181
112 Prototype av boks med alle armmoduler	181
113 Armbunn med kulelager	181
114 Feste til kulelager	181
115 3D-printet replika av felg	182
116 Gammel billader som ble vist til oppladning av batteri ved EES	188
117 Lade opp det flate batteriet fra EES med monitoring av spenningen	190
118 Motoren anskaffet fra Elfa Distrelec	192
119 Reguleringskonsept for rotasjonsmotor	193
120 Visning av lengder i Roveren	195
121 SoilRover prototype	198
122 Deler	199
123 Hjul sammenlignet med verden	200
124 Under bygging	200

125 Endret senterpunkt, fra posisjon 1. til 2.	201
126 Hierarki av SoilRover deler	202
127 RigidBody innstillinger	203
128 2 arm posisjon	204
129 Heisemekanisme	205
130 Hjulmodul beveges sammen med hjul	205
131 Kollidere	206
132 WheelCollider	207
133 Wheel Collider detaljer	208
134 Liten disk	208
135 Stor disk	209
136 Verden sett ovenfra	213
137 Terreng	213
138 Trær	214
139 Typer av busker og gress	214

140 Hjul som ligger under terrenget	216
141 Input Manager i Unity	217
142 QR-kode som sender bruker til våres nettside	219
143 GANTT-diagram v.1	221
144 GANTT-diagram v.2	223
145 GANTT-diagram v.3	225

Tabeller

1 Forskjellige programmene vi har brukt og deres bruksområde	14
2 Budsjett godkjent av SSI	18
3 Budsjett godkjent av SSI sammenlignet med utgifter	19
4 Risikomatriks konsekvens/sannsynlighet	23
5 Utkast fra tabell Prosjekt risiko	24
6 Utkast fra tabell Teknisk risiko	25
7 Utkast fra tabell HMS risiko	26

8	utklipp av fysiske kravlisten	28
9	Mikrokontroller	45
10	Variabler til (2) og (3)	53
11	Stepper motor info	53
12	Drivmotor info	59
13	Driv motor 834-7663 info	61
14	Driv Enkoder HRPG-ASCA#16C info	62
15	Beskrivelse av deler i fig.35	64
16	TS-25GA370H	72
17	LED-stripe forklaring	88
18	Display-beskjeder	90
19	Styring	113
20	Beskrivelse av styring	114
21	Prosjekt risikoanalyse	142
22	Teknisk risiko	145

23	HMS risiko	150
24	Dokument krav	153
25	Komponenter krav	154
26	Datasystemer krav	154
27	Fysiske krav	157
28	Siste strømregninger	187

Glosary

Forkostelser	
SSI	Soil Steam International
SR	SoilRover
RPM	Rotations per minute
PPR	Pulses per rotation
APP	Applikasjon
BMS	Battery Management System
EES	Electronics & Energy Solutions
FEM/FEA	Finite element method & Finite element analysis

1 Innledning

Dagens landbruksindustri står ovenfor mange utfordringer. I mars 2022 var de globale matprisene det høyeste de har vært noen gang [1] og når allerede 38 % av verdens landmasser brukes til landbruk [2] blir det tydelig hvor viktig det er å øke matproduktiviteten, for å spare natur og øke matsikkerhet. I dag brukes mange kjemikalier i landbruk som kan føre til blant annet kreft ved større eksponering [3]. Vi har også et globalt problem med introduserte arter [4] som påvirker både natur og landbruk negativt. Soil Steam International (SSI) adresserer flere av disse utfordringene. Ved å dampe jorden øker de produktiviteten av åkre med usunn jord, ved å uskadeliggjør 96-100% av frø og ugress og dreper sopp og nematoder samtidig som de minker behovet for pesticider [5].

SSI har flere produkter. De er delt opp i mobile og stasjonære produkter som dekker litt forskjellige behov. De utvikler nå et nytt mobilt produkt som er en AGV (autonomous guided vehicle). Produktet har hjul som kan rotere 360 grader om sin vertikale akse, dette gir økt mobilitet som minker kjørevei og dermed minker tid og energi nødvendig.

SSI har gitt oss som oppgave å lage en 1:10 modell av deres AGV kalt SoilRover. AGV-en er deres første produkt med 360 graders vertikal roterende hjul og de ønsker å utforske mulighetene dette gir dem til styring av kjøretøyet, og det er her vi kommer inn. De ønsker å ha en liten modell de kan eksperimentere siden det er billigere, tar mindre plass og har mindre konsekvenser om noe skulle gå galt. Som et pluss kan de bruke det som demonstrasjon for potensielle kunder. Hovedfokuset blir derfor å speile kjøretøyets bevegelser. Siden vårt produkt blir en slags showmodel trenger vi ikke å ha med damping. Følgene bevegelser er det vi skal implementere:

Hjulene skal rotere 360 grader om sin vertikale akse.

Hjulene skal kunne heises over undersiden av kjøretøyet.

Armen til hjulene skal kunne settes i to forskjellige posisjoner 75 grader fra hverandre.

Et par av hjulene skal ha et vippeledd.

Kjøretøyet skal naturligvis også kunne kjøre og rygge. I tillegg til å oppfylle disse bevegelsene er det ønskelig at modellen ser estetisk fin ut, slik at den kan brukes som showmodell.

For bedre brukervennlighet har vi implementert at kjøretøyet vårt kjører på batteri slik at den kan fritt bevege seg og den vil styres med PS4 kontroller og en app som gir mer info.

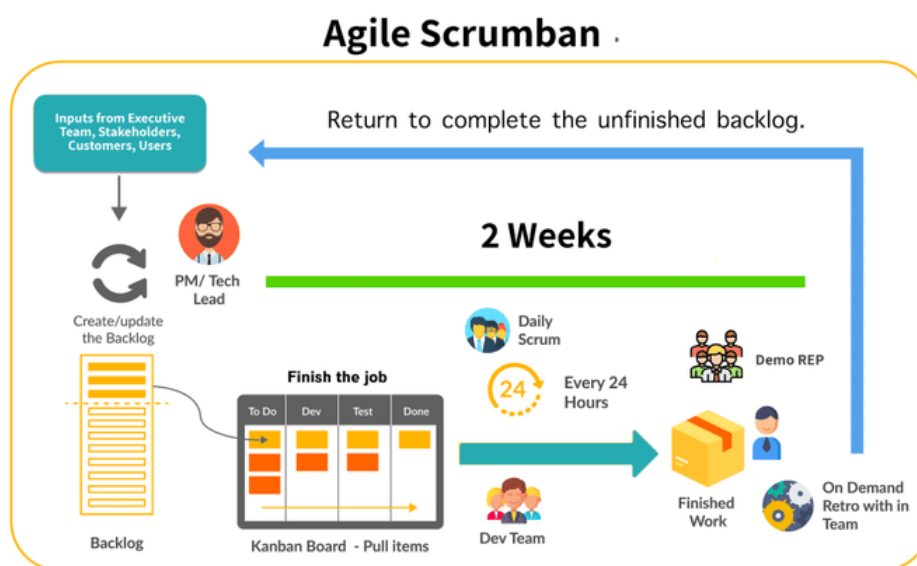
I tillegg til det fysiske kjøretøyet har vi utviklet en simulasjon der tester kan bli gjort før de utføres fysisk. Simulasjonen styres med tastatur eller PS4 kontroller.

2 Prosjekt organisering

I dette kapitlet blir vår administrative plan for bacheloroppgaven presentert. Det blir gått gjennom valgt prosjekt modell og lagt plan for mål og milepæler inn i et GANTT-diagram ble oppdatert iløpet av prosjekt perioden. Andre administrative planer lagt før utførsel av arbeid på produktene (kjøretøy, app og simulasjon) blir også diskutert, slik som økonomi, risiko analyse og krav for prosjektet.

2.1 Prosjekt modell

Vi har brukt Scrumban som prosjektmodell. Den tar i bruk korte iterasjoner og jevnlig møter kombinert med planleggingsmøter ved behov, som gjør at vi kan fokusere på oppgavene våre og vet til enhver tid hvordan de andre ligger an.



Figur 1: Scrumban

2.1.1 Daglige møter

Hver dag, med unntak av tirsdager (til og med mars), møtes vi fysisk og eventuelt via Teams og oppdaterer hverandre om hva vi har gjort siden sist møte og hva vi skal gjøre til neste. På denne måten vet man hva alle jobber med og det er en mulighet til å oppdage om noen henger etter, slik at vi kan ta tak i det tidlig. Disse møtene holder vi på morgenen kl.10:15, og blir referert til som morgenmøter.

2.1.2 Iterasjonsmøter

Annenhver mandag har vi et iterasjonsmøte der vi planlegger de to neste ukene. Da legger vi til oppgaver i tavlen og bestemmer et tema for denne iterasjonen basert på oppgavene og sluttresultatet.

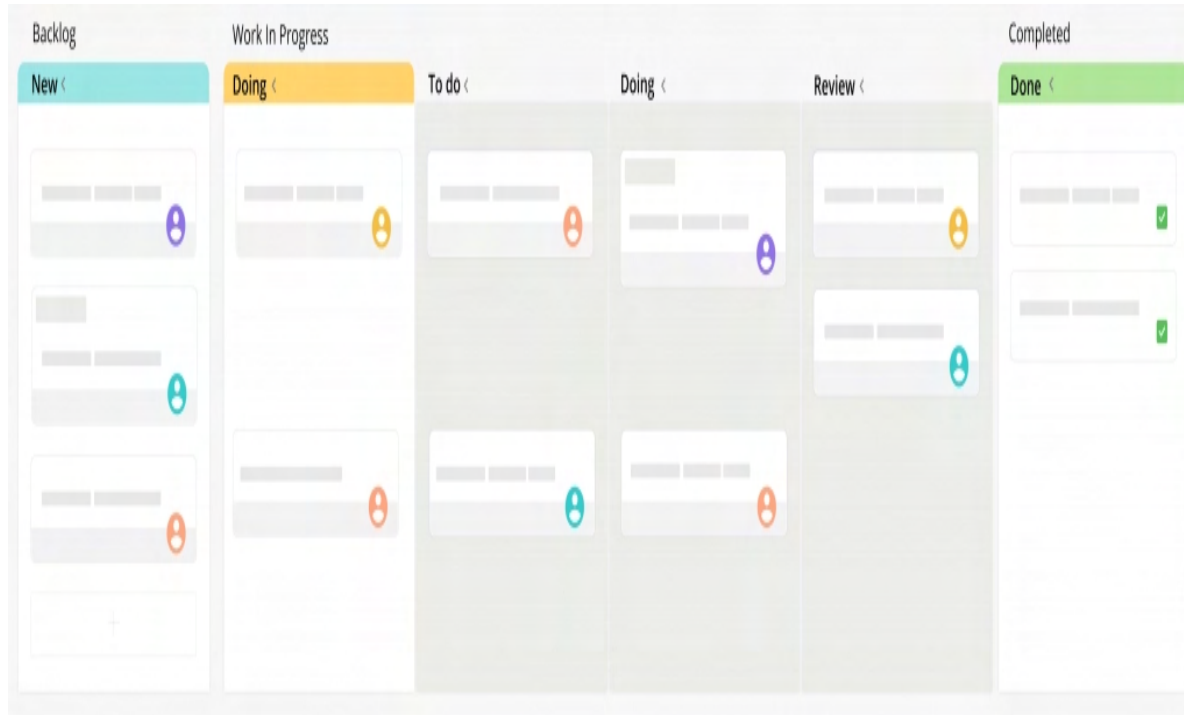
Når det har gått litt under 2 uker, har vi et nytt møte som avslutter iterasjonen. Under visse omstendigheter er iterasjonene litt lengre på grunn av ferie og lignende. I felleskapet flytter vi oppgaver fra “Vurdering” til “Ferdig” på tavlen. I tillegg snakker vi om hvordan iterasjonen har gått, hva som gikk bra og hva som kan forbedres til neste iterasjon. Ved å ha korte iterasjoner, er vi mer fleksible til å tilpasse oss endringer og uventede situasjoner noe som ikke er usannsynlig i vår situasjon (Covid 19).

2.1.3 Bøtte planlegging

Scrumban bøttene er for å dele opp prosjektet i 3 ulike stadier. Vi har delt opp våre 3 bøtter med utgangspunkt i den første, andre og siste presentasjon. Så vi ender opp med 1,2 og 4 måneder lange stadier. Slik kan vi fokusere på det som er relevant akkurat nå, uten å bekymre oss for dårlig tid mot slutten.

2.1.4 Tavlen

Vi har brukt Trello for å visualisere tavlen vår. Den består av ”Å gjøre“, ”Gjøres“, ”Vurdering“ og ”Ferdig“



Figur 2: Figur viser oppsettet av tavlen vår

Under iterasjonsmøter og planleggingsmøter legges det til oppgaver i “Å Gjøres”. Når noen begynner på en oppgave, flyttes de fra “Å Gjøre” til “Gjøres”. Når oppgaven er gjort, flyttes den til “Vurdering”. Alle oppgaver er relativt små slik at de kan fullføres i løpet av 1-2 dager med litt margin.

Siden gruppen består av tre forskjellige disipliner kan ikke alle gjøre alt. Derfor har vi både administrative oppgaver og rundt 6 oppgaver per disiplin. Hver enkelt kan ikke ha mer enn 2 oppgaver i “Gjøres”. Slik vet man at folk blir ferdig med oppgaven og ikke begynner på for mange samtidig, slik at ingen av de blir ferdig.

2.1.5 Prioritering

Hver oppgavene som legges til i Tavlen får en prioriteringsverdi. På den måten vil vi kunne velge oppgaver som gruppen synes er viktig. Det garanterer at vi har riktig fokus.

2.1.6 “On-demand” planlegging

Skulle det hende at flere blir ferdig med oppgaver som er relevant for dem, så har vi et planleggingsmøte. Her kommer gruppa med flere oppgaver som vil være relevante og blir overført til neste iterasjon hvis de ikke skulle bli fullført i den pågående.

2.1.7 Team

Vi har fordelt noen oppgaver, men det er hovedsakelig vedlikeholdsoppgaver som budsjett og Gantt-diagram.

Under iterasjonen velger man selv oppgaver fra Å Gjøres. Det fører til oppgaver som er mindre eller større ikke påvirker den enkelte, men fører heller til at man samarbeider om å få gjort alle oppgaver.

2.1.8 Funksjons frys

Ved å sette en bestemt dato der man ikke kan lengre legge til funksjoner hindrer man såkalt feature-creep. Vi satt datoen til onsdag før påske. 21. April.

Triage Mellom funksjonsfrysingen og leveringsfrist satt vi en ny dato, den 29. April. Den dagen bestemmer prosjektleder med tilbakemelding fra gruppen, hvilke funksjoner vi skal fokusere på og hvilke som forkastes. Dette vil være basert på kravene til prosjektet, men også hva som er mulig innenfor tidsfristen.

2.1.9 Møter

Vi kommer til å holde regelmessige møter med veileder og etter behov med bedriften. Internt hadde vi møtene fysisk så langt det lot seg gjøre. Som nevnt under Scrumban så har vi regelmessige møter. Vi lager møtereferat fra alle møter utenom morgen møtene. Alle møtereferatene blir skrevet på OneNote hvor vi har delt det inn i referat for veileder, bedrift og internt.

2.1.10 Interne veiledningsmøter

Veiledningsmøtene holdes på onsdager, dette er et internt møte med veileder der vi gir en kort oppsummering av hva som er gjort, og kan få svar på spørsmål, tips og lignende. Møtene gjøres digitalt over Zoom.

2.1.11 Møter med bedriften

Møtene med de eksterne veilederne tas etter behov, da spesielt siden de gjør fortløpende endringer i produktet deres som endrer våre krav. Møtene vi har med dem deles ofte inn i maskin- og elektroingeniørene sammen eller data- og elektroingeniørene sammen. Møter med bedrift eller eksterne veiledere gjøres over Teams.

2.1.12 Internt i gruppa

Internt i gruppa så bruker vi Teams til morgenmøter og ukentlige møter om noen ikke er fysisk på skolen. Vi har ukentlige møter på mandager for å planlegge uka vi går inn i og har møter annen hver fredag hvor vi oppsummerer iterasjonen vi er ferdig med og planlegge neste iterasjon som skjer mandagen som følger.

2.1.13 Kontakt

Kontakt med bedriften, intern veileder og internt i gruppa har gått over forskjellige plattformer, men alt av informasjonen har blitt lagt på OneNote for lett tilgjengelighet senere om vi skulle trenge den informasjonen senere.

2.1.14 Bedriftens kontakt

Møtene med SSI og eksterne veilederne har gått over Teams og vi fører møtereferat for hvert møte. Vi har kontakt med bedriften over e-post og over en Teams kanal som de deler med oss. På Teams kanalen deler vi progresjonen vår. Møter med dem går som tidligere nevnt mer fortløpende når vi føler vi trenger hjelp med noe eller om de føler de trenger en større

oppdatering enn ukens rapportene vi gir dem hver fredag. Vi sender dem en ukentlig rapport til intern veileder hver fredag så de også kan følge godt med på progresjonen og hva vi gjør. Vi sender også disse ukentlige rapportene til bedriften så de kan følge progresjonen også.

2.1.15 Intern veileder

Som nevnt tidligere så har vi ukentlige møter med intern veileder, men utenom de møtene så kontakter vi han via canvas eller via e-post når vi har spørsmål.

2.1.16 Internt i gruppa

Internt i gruppa så bruker vi MS Teams og Facebook Messenger for å kontakte hverandre. Vi bruker Facebook Messenger for å si ifra om vi ikke kommer på møter eller om vi har spørsmål. Teams er brukt for å oppdatere om noe er lastet opp eller endret av dokumenter på teams eller OneNote.

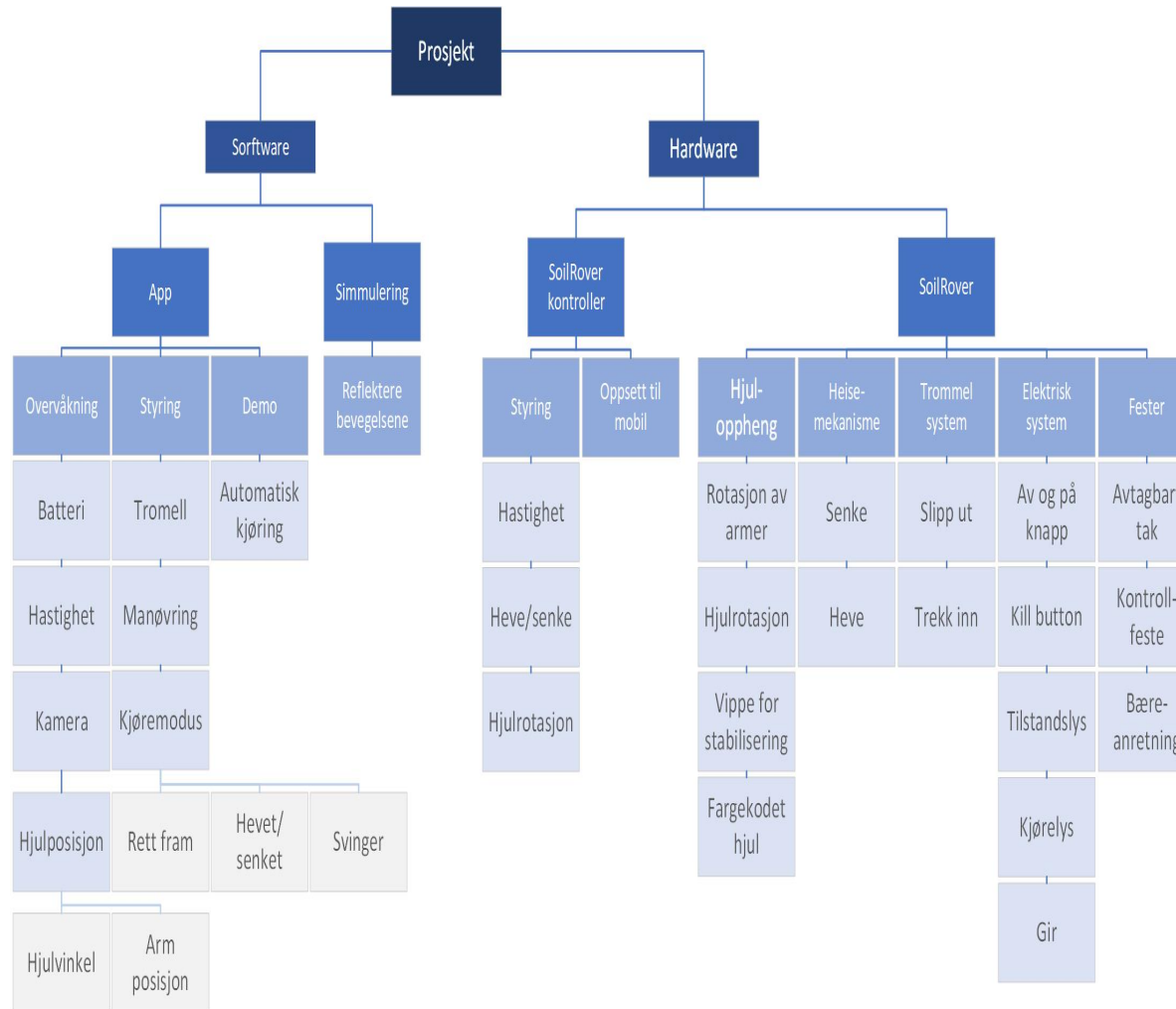
2.1.17 Diskusjoner i gruppa

Samtaler som utvikler seg til litt større diskusjoner. Disse diskusjonene prøver vi å dokumentere, men noen diskusjoner er bare små samtaler for å finne ut av større løsninger. Vi har hatt en stor diskusjon med gruppa hvor vi satt og så på prosjektet som et funksjons tre, som kan bli sett på fig. 3. Dette tok noen timer og var svært opplysende for hele gruppen, etter som dette ga alle et godt overblikk for hvordan alle tenkte seg produktet og dens funksjoner. Og selv om ikke alle er fysisk til stede i grupperommet så er alle tilgjengelig teams og vi setter det opp slik at alle kan delta og komme med innspill.

Vi informerer medlemmer som ikke kunne delta om hva som har blitt gjort, hvis det er noe viktig som kom opp i diskusjoner. Vi tar opp det som er blitt diskutert neste gang alle er samlet eller på de daglige møtene så alle er klar over dem og kan komme med sine ideer.

2.1.18 Endringer og kommunikasjon






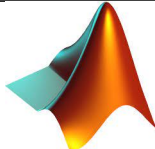



Etter som at dette prosjektet går parallelt med SSI sin utvikling av deres produkt så vil det komme endringer. At de endringene kan endre kravene til vårt produkt er forstått og forventet. Disse endringene gjør det viktig med god kommunikasjon innad i gruppen og med bedriften.










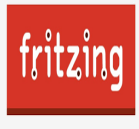
Figur 3: Figur som viser funksjonstreet vårt

2.2 Prosjekt hjelpemidler

Tabell 1: Forskjellige programmene vi har brukt og deres bruksområde

SolidWorks	Brukt til 3D-modellering og analyse	
Trello	Brukt som en virtuell Kanban tavle.	
OneNote	Brukt til å samle informasjon og ta referater.	
Microsoft Teams	Brukt som møteplattform og for deling av filer.	
Visio	Brukt til å lage diagrammer og figurer.	
Matlab	Brukt til å utføre kalkulasjoner.	
Arduino IDE	Brukt til å skrive kode til mikrokontroller.	
GitHub	Ble Brukt til dokumentasjon av kode	
Procreate	Tegneprogram for iPad som ble brukt for design.	

Krita	Tegneprogram for data-maskin som ble brukt til illustrasjoner.	
PowerPoint	Brukt til å lage plakat og presentasjoner.	
Word	Brukt til de to første dokumentasjonene, oppfølgingsdokumenter og møteinnkallinger.	
Excel	Brukt til timeliste, bestillingsliste og økonomiske utgifter.	
Facebook Messenger	Brukt som en uformell kommunikasjonskanal der det var enkelt og få tak i folk,	
Outlook	Brukt for å holde styr på kalenderen og til å sende ut møteinnkallinger.	
Unity	Brukt som Spillmotor for å lage simulasjon.	
Android Studio	Brukt til å lage applikasjonen.	
JottaCloud	Brukt for å lagre og dele filer med hverandre på samme plass.	
JabRef	Brukt til å assistere med å lage referanser i L ^A T _E X	

FreeCad	Brukt til å konvertere 3D modeller.	
Wordpress	Brukt til å lage prosjektets nettside.	
Fritzing	Brukt til å visualisere koblinger i Arduino	

2.3 Prosjekt plan

Målet med vårt prosjekt fra SSI er å lage de tre produktene etterspurt ifra gitt oppgave, hvor disse blir en fysisk prototype av deres kjøretøy, en applikasjon med tilsvarende informasjon og en simulasjon for videre bevegelses testing. Den simulerte og fysiske produktet skal kunne utføre en rekke bevegelige egenskaper i dens kropp og armmodul som etterligner til størst grad de fysiske egenskapene til SSIs kjøretøy.

Med SSI som stakeholder tregner vi å tenke på fysiske egenskaper og estetisk utseende når vi utfører prosjektet. Kjøretøyet må ligne og oppføre seg mest mulig likt som deres eget, siden det skal bli tatt med på diverse messer for å vise eventuelle kunder hvordan deres maskin vil kjøre på en åker. Den og den simulerte modellen må også fungere likens mulig med tanke på at SSI kan bruke disse videre til testing av bevegelser.

Vi har da delt opp prosjektet i noen hovedoppgaver etter egenskaper. Vi har armmodulen til SoilRover som består av driv og rotasjon av hjulene, med en låsemekanisme for å kunne sette kjøretøyet fra lastebilmodus til kjøremodus. Kroppen til SoilRover foretar seg heisemekanismen, deksel og strukturen til styring som kan kontrolleres via PS4 og app. Applikasjonen som skal inneholde relevant informasjon og til slutt simuleringen som skal kunne gjøre det samme

som det fysiske produktet og kan styres med PS4 kontroller eller tastatur.

For å få et helhetsoversiktlig bilde over prosjektets omløp bestemte vi for å visualisere det med et GANTT diagram som vi oppdaterte til hver presentasjon. GANTT diagrammet er delt opp i våre først satte iterasjoner, med de underliggende oppgavene og milepælene vi ville oppnå til spesifikke tider. GANTT diagrammet hjelper å visualisere disse tidsrammene og fristene vi har satt og ble brukt som en referanse over hvor godt vi ligger etter planen. I appendix G fig. [143](#) ser vi hvordan vi fordelte prosjektet vårt i oppgaver over perioden gitt.

2.4 Økonomisk

Etter å ha sett på priser på utstyr og budsjettet til tidligere prosjekter presenterte vi et budsjett til SSI som vi fikk godkjent.

Tabell 2: Budsjett godkjent av SSI

Inventory list	Estimert Kostnad
Motorer og gir	13000
MotorKontroller	4500
MikroKontroller	600
Deler	3000
Materiale	15000
Bluetooth Enhet	500
Ekstern Kontroller	350
Utstyr Elektrisk	2000
Skjold	1500
Batteri	8000
Diverse	8000
Kontorutstyr	1500
Total	57950

Budsjettet vi ser i tabell 2 ble godkjent av SSI. De rundet det opp til 60 000 kr for ordens skyll.

Vi brukte for lang tid på å bestille deler for testing. Med tanke på vårt budsjett skulle vi tatt mer risikoer og bestilt opp komponenter fort for å teste og heller forkaste de hvis de ikke passet til prosjektet.

Etter andre presentasjon fikk vi tilbakemelding fra SSI at med tanke på leveringstider var vi nødt til å få bestilt opp deler til prosjektet for å kunne rekke å bygge noe fysisk. Leveringstid var å medregne basert på de tidligere årene da det har vært korona, og i år så er det visse leverandører som hadde lenger leveringstid på grunn av mangel på materialer eller nåværende situasjon i Ukraina(vår 2022).

Det var med flere anledninger der det var bestilt deler der det sto de var på lager på nettstedet men etter et par dager kom det e-post tilbake med ny ordrebekreftelse med info om at det som var bestilt ikke var tilgjengelig eller at leveringsdato var i Juli eller senere. Dette førte til at vi ble nødt til å lete etter nye alternativer eller andre leverandører for lignende deler/komponenter.

Tabell 3: Budsjett godkjent av SSI sammenlignet med utgifter

Inventory list	Estimert Kostnad	Faktisk kostnad
Motorer og gir	13000	6975,5
MotorKontroller	4500	850,96
MikroKontroller	600	233,74
Deler	3000	11082
Materiale	15000	9804,3
Bluetooth Enhet	500	126,57
Ekstern Kontroller	350	899
Utstyr Elektrisk	2000	2493,2
Skjold	1500	260,89
Batteri	8000	1875
Diverse	8000	4795,4
Kontorutstyr	1500	814,9
Total	57950	39977

For å ha kontroll på alle utgifter så var det fra store leverandører best å få tilsendt faktura direkte til oppdragsgiver slik at de kunne få skrevet av moms og medlemmer slapp å være mellomledd for betalinger.

Kjøp som ble gjort i fysiske butikker eller hos leverandører som SSI ikke var kunde hos ble det lagt ut for av gruppen, kviteringene derfra ble ført opp og videresendt til SSI. Det som ble sendt til SSI var et Excelskjema med dato for utlegg, hvem det var som la ut, hva det var som ble kjøpt og kvitteringsnummer. Dette skjemaet ble sendt sammen med kviteringene som

hadde blitt lagt ut for siden sist innsending, det ble kun gjort tre innsendinger med utlegg for å unngå å sende kvittering individuelt slik at det ble masse. Utklipp av systemet er vist i fig. 4.

Pris er totalt av alt, multipliserer ikke antall med pris.

Dato	Vare	Salgssted	Godkjent av	Godkjent dato	Betalt av	Antall	Pris	Kvittering nr/stack nr	TilstandKvittering	BestillingTilstand
2022-01-28	Dekk	Hobbmann AS			Vetle	2	850		1 Tilbakebetalt	Fått
2022-01-29	Rørisolasjon	Biltema			Vetle	1	34,9		2 Tilbakebetalt	Fått
2022-01-28	Logitech Kamera	Power			Vetle	1	632		3 Tilbakebetalt	Fått
2022-02-07	Plastlommer	Brage Bokhandel			Vetle	1	50		4 Tilbakebetalt	Fått
2022-03-07	Diverse skruer	Jernia			Vetle	1	15		7 Tilbakebetalt	Fått
2022-03-14	Treskrue, Jernvare	Jernia			Vetle	1	80		6 Tilbakebetalt	Fått
2022-03-14	Møbelhjul	Clas Ohlson			Vetle	1	49,9		5 Tilbakebetalt	Fått
2022-03-25	Lead Screw	RS			SoilSteam	3	874,47	862-5284	Soilsteam la ut	Fått
2022-03-25	Flange Round Nut	RS			SoilSteam	4	2376,48	862-5326	Soilsteam la ut	Fått
2022-03-25	HybridStepper	RS			SoilSteam	1	624,44	535-0401	Soilsteam la ut	Fått
2022-03-16	RGB Lys	Kjell Company			Vetle	1	699		8 Tilbakebetalt	Fått
2022-03-27	USB Host Shield	Amazon			Vetle	2	646,46		15 Tilbakebetalt	Fått
2022-01-20	BrevOrdner	Braga			Vetle	1	39		14 Tilbakebetalt	Fått
2022-03-28	Svamper(testing)	Normal			Mina	2	33		12 Tilbakebetalt	Fått
2022-03-28	Påskegg/trelim(testing)	Europriis			Mina	1	54,83		9 Tilbakebetalt	Fått
2022-03-28	Isolasjonsteip/Nanogel	Jernia			Mina	1	139		10 Tilbakebetalt	Fått

Figur 4: Utklipp av oversikt over økonomi

Vi var i tett dialog med økonomiavdeling fra SSI og konsulterte med de om kvitteringer som vi var usikker på var godkjente utlegg. Dette var for at det ikke skulle bli diskusjon mellom bedrift og studenter om disse utleggene mot slutten av prosjektet.

2.5 Risiko analyse

For å redusere sannsynligheten for potensielle risikoer som kunne sette prosjektet i fare, var det nødvendig med en grundig forståelse av hendelser, deres natur og bakgrunn. For dette formålet har bachelorgruppen gjennomført en detaljert årsak-konsekvens risikoanalyse. Potensielle hendelser ble analysert nøye og evaluert, i forhold til alle disipliner, både i begynnelsen av prosjektet og i løpet av det da basert på tilbakemeldinger fra veiledere etter første presentasjon.

For å kartlegge og vise risikonivåer brukte bachelorgruppe en risikomatrix som er en grafisk metode av vurdering og visualisering av risikonivå på en tydelig måte via trafikklys farger - noe som alle forstår. Risikonivået er resultatet av to parametere:

- Sannsynlighet – indikerer hvor sannsynlig det er at hendelsen vil oppstå. Parameteren har tre mulige nivåer: 1-3. 1 betyr lav sannsynlighet, 2 betyr medium sannsynlighet, 3 betyr høy sannsynlighet. Sannsynlighetsparameteren går vertikalt på matrisen.
- Konsekvensklasse – indikerer hvor alvorlige konsekvensene kan være for prosjektet, Parameteren har tre nivåer I-III. I betyr lav konsekvensklasse, II betyr medium konsekvensklasse, III betyr høy konsekvensklasse. Den går horisontalt på matrisen.

De 3 fargene betyr:

- Grønnfarge betyr lavt nivå - den signaliserer at en hendelse har lav betydning for prosjektet. Den kan bli løst innen gruppa, uten ekstern hjelp.

- Gullfarge betyr medium nivå – det kan være nødvendig å snakke med en veileder eller noen andre, mer erfarne folk.
- Rødfarge betyr høyt nivå – hendelse er kritisk; det er nødvendig å handle umiddelbart.

Tabell 4: Risikomatrix konsekvens/sannsynlighet

3	Lav	Medium	Høy
2	Lav	Medium	Medium
1	Lav	Lav	Lav
	I	II	III

Nedenfor er utkast fra risikotabeller - prosjekt risiko [21](#), teknisk risiko [4](#) og HMS risiko [23](#).

Tabell 5: Utkast fra tabell Prosjekt risiko

Nr.	Hendelse	Konsekvens	Sannsynlighet	Konsekvensklasse	Forebyggende tiltak	Håndtering
R.P.01	Ny nedstengning av Kongsberg.	Vi kan bli nødt til å ha hjemmekontor, noe som kan bremse fremgang i prosjektet.	1	II	Vi følger nasjonale tiltak så langt det lar seg gjøre.	Kommunisere gjennom digitale plattformer.
R.P.05	Oppgaver blir gjort dobbelt opp.	Ikke effektivt bruk av tid, og kan skape forsinkelser.	1	II	Gruppemedlemmer burde kommunisere tydelig det de jobber med. Følge med på scrumban møter og iterasjonsmøter. Spørre, hvis en ikke vet hva man skal jobbe med.	Bedre kommunikasjon i fremtiden.

Tabell 6: Utkast fra tabell Teknisk risiko

Nr.	Hendelse	Konsekvens	Sannsynlighet	Konsekvensklasse	Forebyggende tiltak	Håndtering
R.T.01	Feil/forsinkelse ved bestilling.	Kjøretøyet blir ikke satt sammen på grunn av manglede komponenter. Taper tid og penger. Treghet i prosjektet.	2	II	Ha klart andre alternativer. Sjekke om det er fysiske butikker i nærområdet. Sjekke om skolen har noe vi kan få brukt.	Gå til andre alternativer. Redesigne det som er nødvendig.
R.T.14	U-kompatible deler	Uferdig produkt.	1	III	Sjekke datablad og koblinger. Spørre veiledere eller faglærere om hjelp.	Finne andre alternativer.

Tabell 7: Utkast fra tabell HMS risiko

Nr.	Hendelse	Konsekvens	Sannsynlighet	Konsekvensklasse	Forebyggende tiltak	Håndtering
R.H.01	Sykdom, større skader eller personligfravær av gruppemedlem/mer.	Kommunikasjon mellom gruppe-medlemmer blir vanskelig. Det kan oppstå risiko for feil eller det at prosjektet ikke blir fullstendig. Fremgang kan bli saktere.	3	II	Ikke møt opp fysisk om du er syk. Følg smitte-råd fra FHI.	Jevnlig oppdateringer mellom gruppemedlemmer. Tilrettelegge for at den som er syk kan fort bli frisk. Gjør det mulig å holde samarbeid online.
R.H.02	Personskade fra arbeid med SoilRover.	Lettere skader fra komponenter og sammensetting av SoilRover	2	I	Vær oppmerksom og forsiktig. Følg sikkerhetsprotokoller.	Følg skadeprotokoll.

2.6 Spesifikasjoner

Kravanalyse var noe av det første som ble gjort. Det samler og definerer hva produktet vårt skal oppnå, i tillegg har vi en testplan for hvert krav. Kravene blir brukt som en retningslinje for alle designvalg, og som en mållinje for hele prosjektet.

Kravene beskrives med kun en setning, og skal være så spesifikke at de ikke kan mistolkes. I kolonnen kravbeskrivelse gis en kort begrunnelse av kravene.

Kravene er delt opp på to måter, etter prioritering og etter type. De forskjellige typene er følgende; dokument krav, komponent krav, data system krav og fysiske krav. Vi bruker en tre nivå skala for å prioritere krav, med følgende prioriteringsnivå:

A – Obligatoriske krav. Disse kravene blir implementert først. Uten oppfylte A-prioritet krav er ikke prosjektets hovedmål oppnådd. Disse er markert rødt.

B – Viktige krav. Disse skal implementeres når A-prioritet krav er oppfylt. Disse er markert gult.

C – Mindre viktige krav. Disse implementeres når A- og B-prioritets krav er oppfylt og det er tid og kapasitet for videreutvikling av produktet. Disse er markert grønt.

I tabell [8](#) kan man se et utklipp av kravlisten til fysiske krav. De forskjellige typene har hver sin liste. Hvert krav har et navn der bokstavene i navnet henviser til hvilken type de er.

Tabell 8: utklipp av fysiske kravlisten

Krav ID	Krav	Prioriteringsgrad	Krav beskrivelse	Hvordan teste
F.01	Skal kunne fjernstyres ved hjelp av en kontroller.	A	En brukervennlig måte å styre kjøretøyet.	Test kjøre.
F.02	Vekten av SoilRover skal ikke overskride 20 kg.	A	Arbeidsmiljøloven (Jan 2022) tilsier at ansatte ikke skal bære mer enn 20 kg.	Kjøretøyet veies.

3 Produksjonsmetoder

Under prosjektet har gruppen benyttet seg av flere produksjonsmetoder. Vi har tatt i bruk 3D printing, støping og laserkutting.

3.1 3D-Print

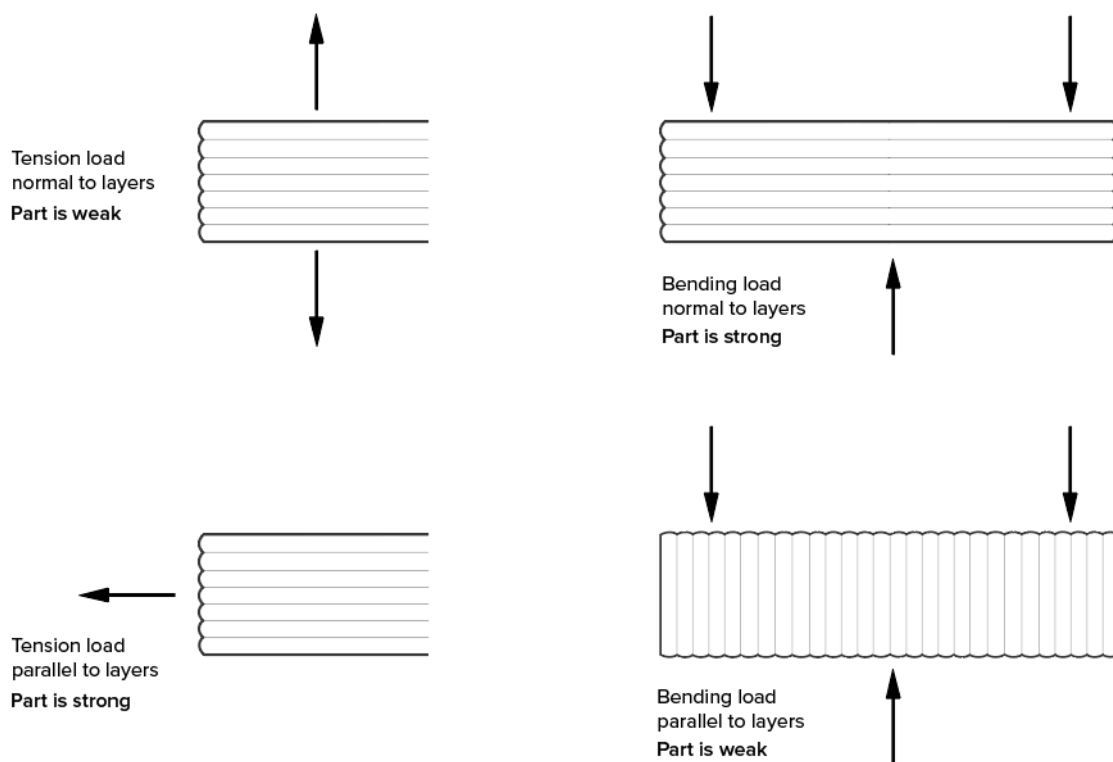
3D-print har mange fordeler, men også ulemper. 3D-printing har blitt mer og mer vanlig ikke bare i industrien men også for privatpersoner [6]. Det er blitt rimelige 3D-printere som gir høy kvalitet på produktene den printer. 3D-printing i både industrien og for den vanlige personen kan nå produsere komplekse deler nesten uanstrengt. Noen viktige ting å ha i tankene når man skal vurdere 3D-printing som produksjons metode;

- Budsjett: man kan velge billige og enkle materialer å få tak i som PLA plast filament, eller dyrere materialer.
- Kosmetisk utseende: en 3D-printer kan få god overflate men det vil da gå på bekostning av tid og vil koste mere for bedre resultat etter som jo bedre resultat man ønsker så kan det være nødvendig med dyrere printere.
- Material valg: er en mengde med materialer å kunne velge fra for FDM-printere.

Det finnes en mengde forskjellige typer 3D-printere som printer på forskjellige måter. De vi hadde tilgang på og brukte er kjent som FDM-printere. FDM står for Fused Deposition Modeling, disse printerne har mulighet til å lage høy detaljerte deler med stramme toleranser. Dette gjør at det er en bra måte å lage prototyper på eller kunne lage komplekse deler med stramme toleranser.

3.1.1 Designe for 3D printing

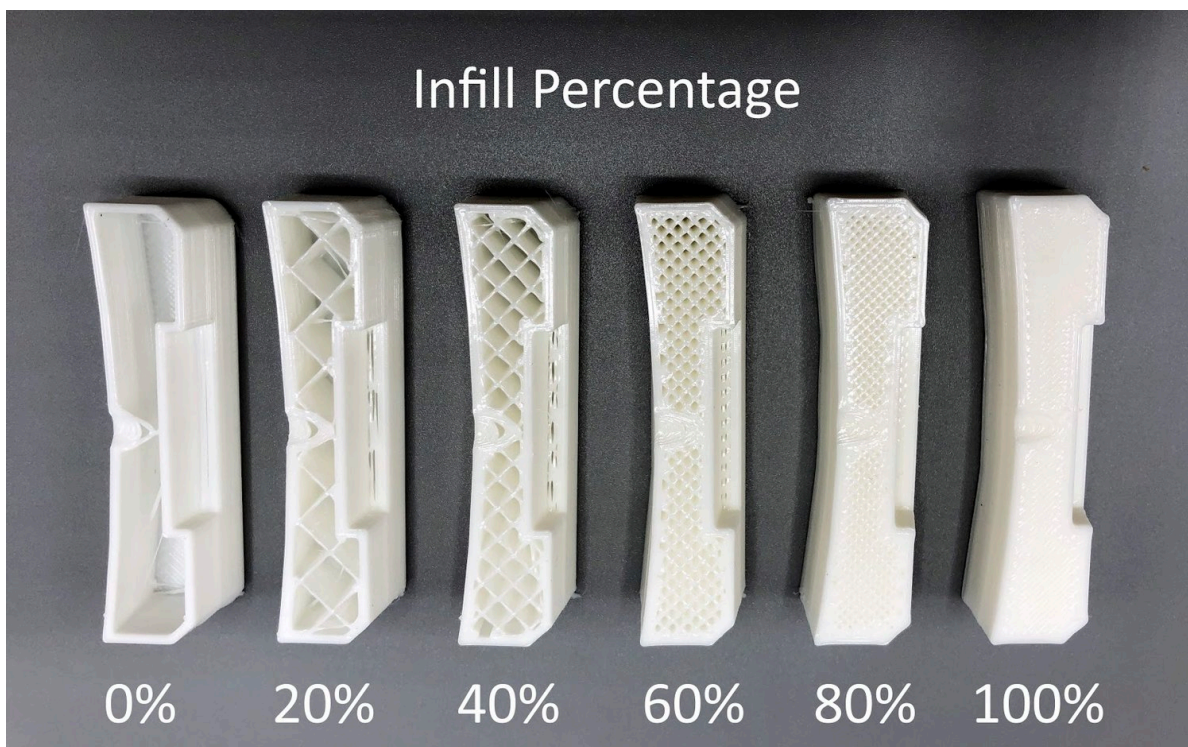
3D-printing gjør det mulig å lage komplekse deler, men det kommer med baksider. Måten 3D-printere jobber på gjør at delen vil bli sterkere i en retning enn den andre. Orienteringen av printlagene har mye å si på styrken til en 3d-printet del. Dette er fordi delen er bygget opp av lag som gjør at styrken varierer med lagtykkelse og hvor kreftene påføres i forhold til lagene. I fig. [5] ser vi at styrken er best når kreftene enten ligger parallelt med lagene eller om det er bøyemoment som er normalt på lagene.



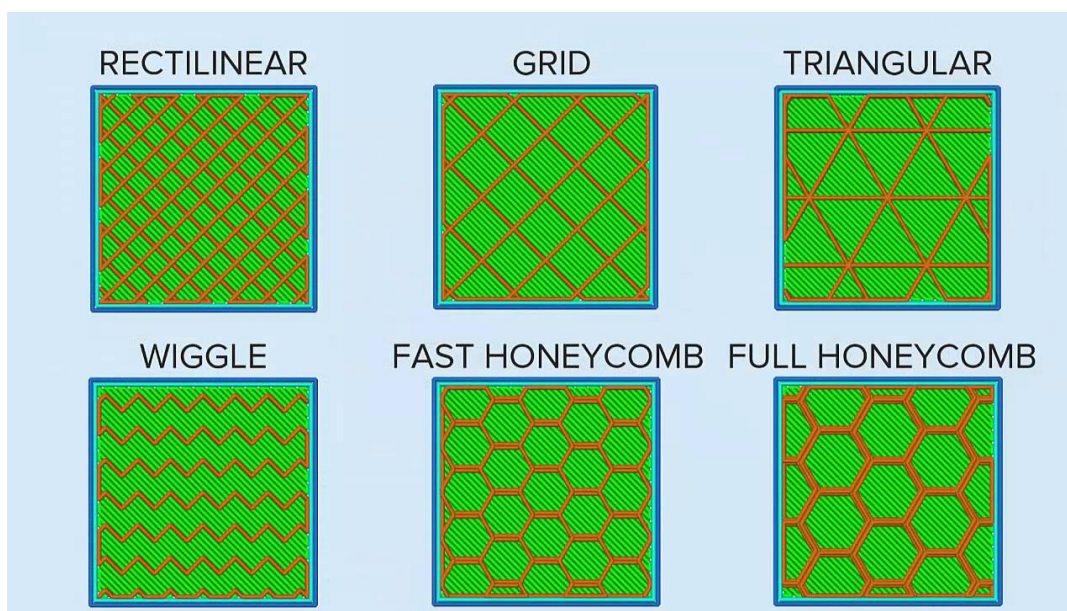
Figur 5: Styrken av 3D printet del med tanke på lag og retning av kraft. [7]

Når vi designet delere for 3D print så må vi hatt FDM-printer i tankene. Det er fordi når man lager deler hvor seksjoner er i friluft vil det være nødvendig å måtte printe med støtte. Dette gir dårlig overflate samt det bruker ekstra tid og materialer som blir forkastet. Styrken til en del kommer mye fra tykkelsen av lagene og hvor godt limt sammen de er, men også fra inn-fyll

prosenten. Inn-fyll prosent og mønster på inn-fyll velges etter ønsket vekt og styrke. På fig. [6] og fig. [7] kan man se endringen i inn-fyll prosent og forskjellige innfyll mønster.



Figur 6: Innfyl prosent i 3D print. [8]



Figur 7: Innfyll mønster i 3D print. [9]

Som alle andre produksjonsmetoder så vil resultatet på 3D-print endres med tanke på materialvalget. Dette gjelder både for hvordan delen vil se ut etter printing og for hvordan printerens vil håndtere materialet. Gruppen har valgt å printe i PLA etter som dette er det enkleste valget [10], det er lett å få tak i og det er lett å printe med. Vi har også valgt å printe noen deler i PETG-filament. PETG er sterkere enn PLA og PETG krystalliseres ikke som PLA, som gjør at PETG er mindre sprøtt [10].

Andre alternativer innen for 3D printing

3D-printing er en produksjonsmetode som nå er vanlig for selv privatpersoner. Det finnes både rimelige 3D-printere for hjemmebruk og det er de litt dyrere men mere detaljerte og har mere kapasitet til å printe. Det er forskjellige måter å printe på, som nevnt tidligere har vi brukt FDM-printere, det er andre typer som SLA-printere også, men disse er dyrere og printer på en annen måte enn FDM. SLA-printere bruker i hovedsak resin, mens FDM-printere som vi har brukt, har mulighet til å bruke en rekke forskjellige materialer i form av filament. FDM-printere er de mest brukte 3D printerene i følge [11]. Det finnes en rekke med andre typer printere [12], men da kommer man nærmere industrielt bruk og ikke for den vanlige enkelt personen.

3.2 Støping

Under prosjektet har vi støpt komposittdeler. Kompositt er et materiale som består av to eller flere sammensatte materialer [13]. De to typene gruppen har brukt er glassfiberkompositt også kjent som et glassfiberarmert plast og karbonfiber. Gruppen ville bruke glassfiberplater som innleggsplater i rammeverket og for sideplater. Dette ble forkastet etter fire mislykkede forsøk og vi fant ut at vi hadde for lite tid, dette kan leses mer om i dokumentasjonen for kroppen i avsnittet om Komposittdeler. En del som skal lages i kompositt er dekselet, den skal da lages i karbonfiber. Det da blitt laget en form for å lage en støp i, denne formen er da som nevnt

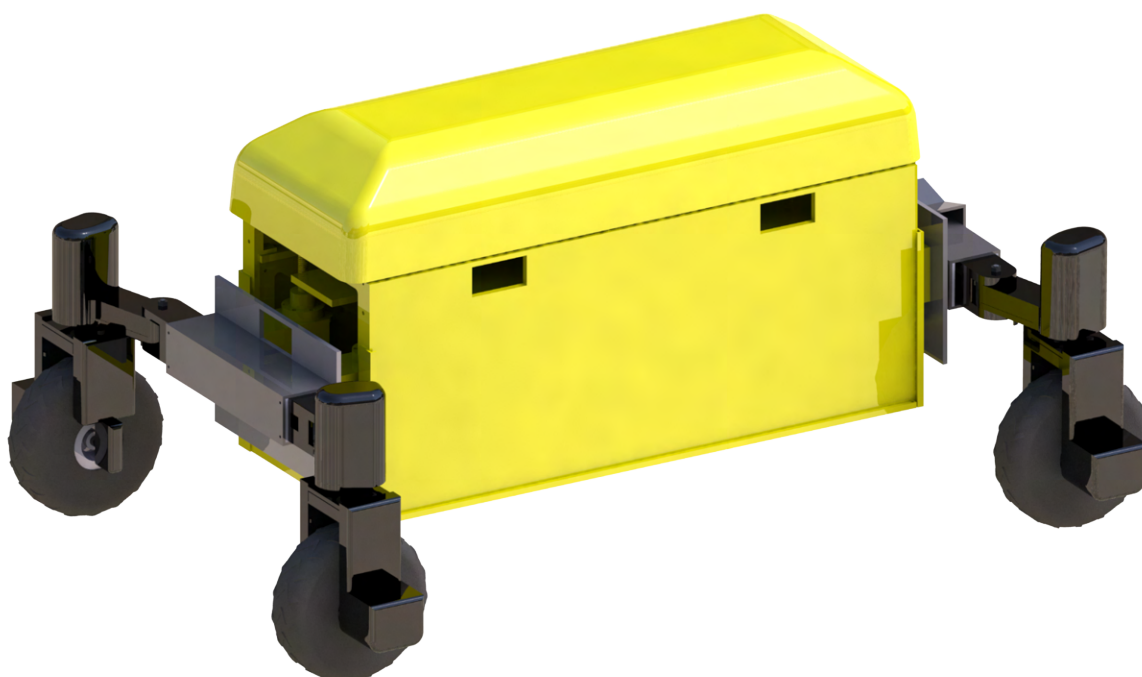
tidligere laget i 3D-print. Dette kan leses om mer utfyllt i dokumentasjonen for kroppen, i avsnittet **Komposittdeler**.

3.3 Laserkutting

Gruppen har laserkuttet med en epilog laser engraver som USN har tilgjengelig. Vi har da laserkuttet rammeverk, sidevegger og våre testtrigger i 3mm bjørke kryssfiner. Laserkutting fungerer ved at den brenner seg gjennom materialet med en sterk CO_2 laser.

4 Kropp

Kroppen, se fig. 8, er betegnelsen vi bruker for rammeverket og heisemekanismen. I rammeverket har vi to etasjer, nedre etasje er for heisemekanismen mens øvre etasje er for alt det elektriske. Boksen er brukt som en samlebetegnelse for rammeverket og alt på innsiden, de elektriske komponentene og alt det mekaniske. Rammeverket er laget av kryssfiner i bjørk og det er laget vinkler i 3D-print av PLA plast for å holde alt vinkelrett. Dekselet blir laget av en karbonfiberkompositt ved hjelp av en 3D-printet form som blir forsterket med kryssfinerplater.

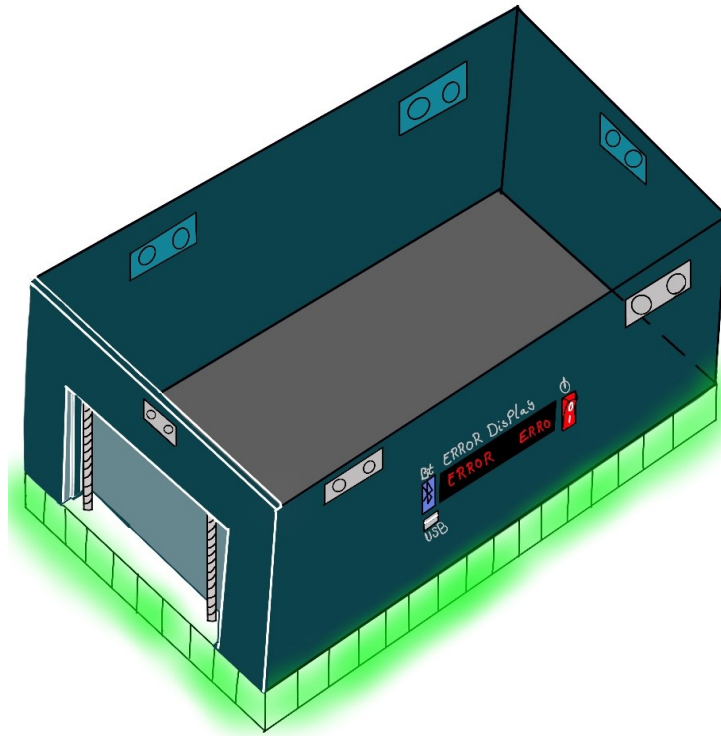


Figur 8: SoilRover med kropp uthevet i gult

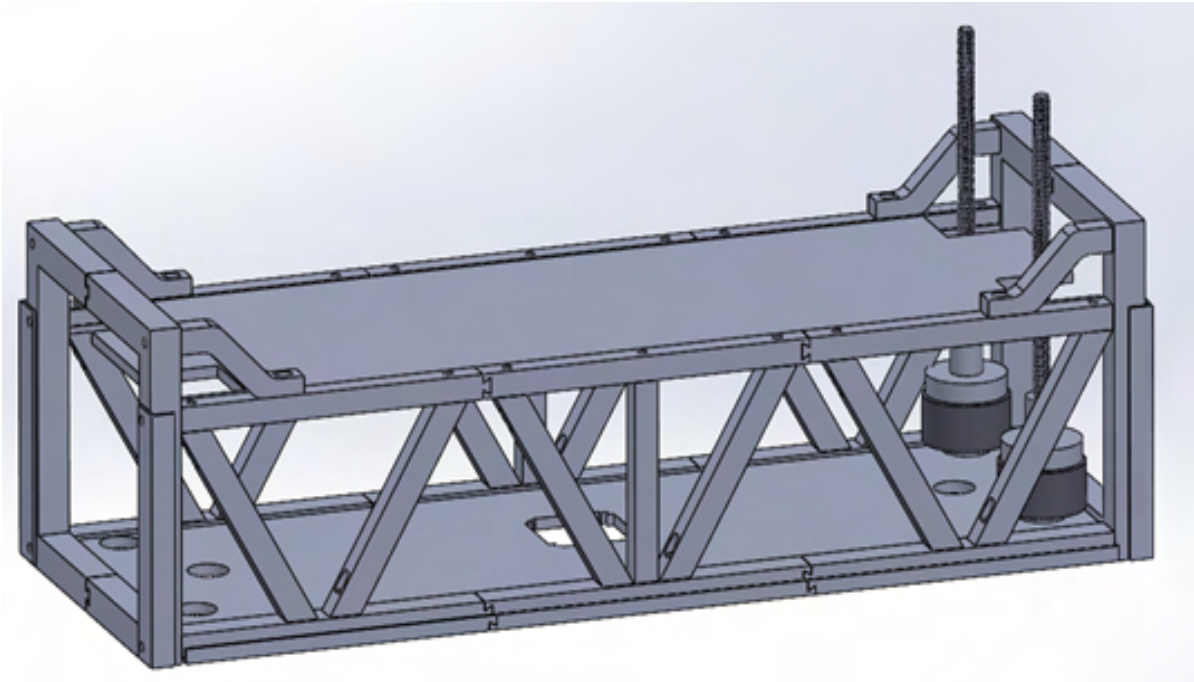
4.1 Boks til rammeverk

Når vi refererer til “Boksen” så refererer vi til rammeverket og alt på innsiden av den, det elektriske og mekaniske. Dette ekskluderer dekslet. Boksen har gått gjennom 3 hovedendringer

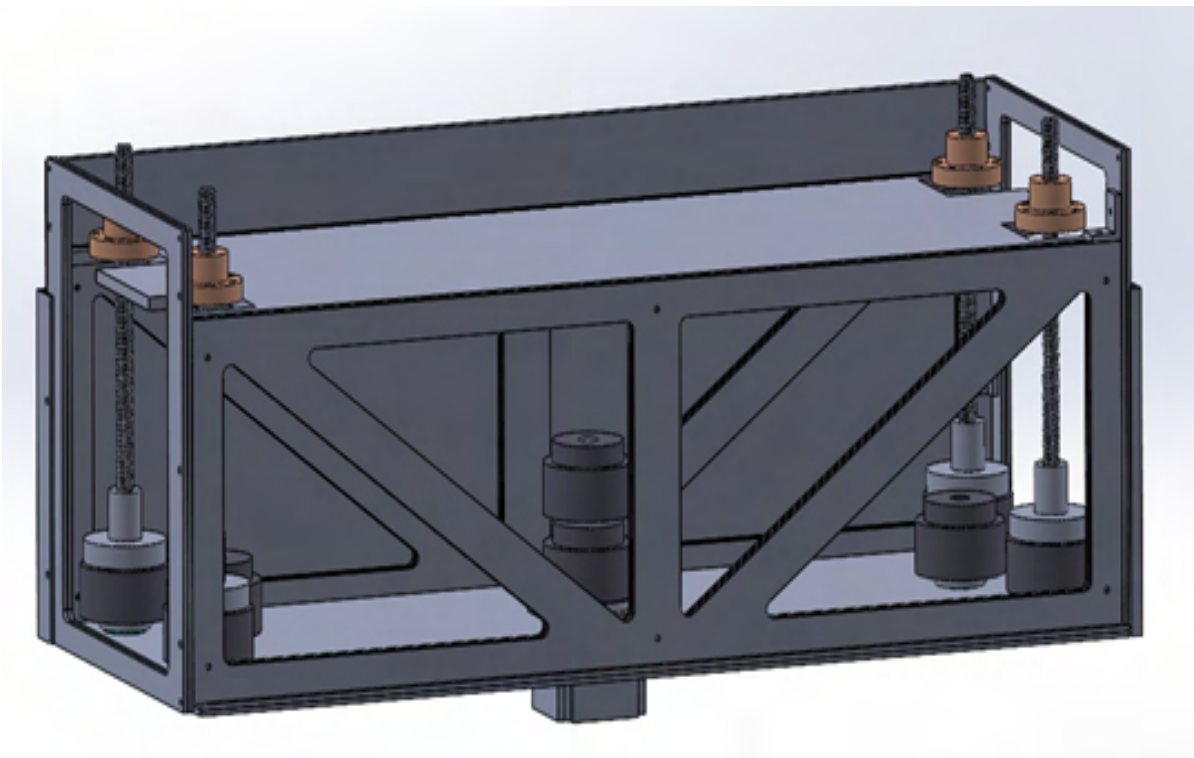
gjennom prosjektet. Etter andre presentasjon så fikk gruppen tilbakemelding om at oppgave-giver ønsket et design som representerte designet deres mere. Designet har gått fra å være en boks laget av glassfiber, se fig. 9, til et rammeverk av 3D-print, se fig. 10, til å være et ramme-verk laget av kryssfiner, se fig. 11.



Figur 9: Boks første versjon, simplifisert illustrasjon for å vise designet til 2.presentasjon.

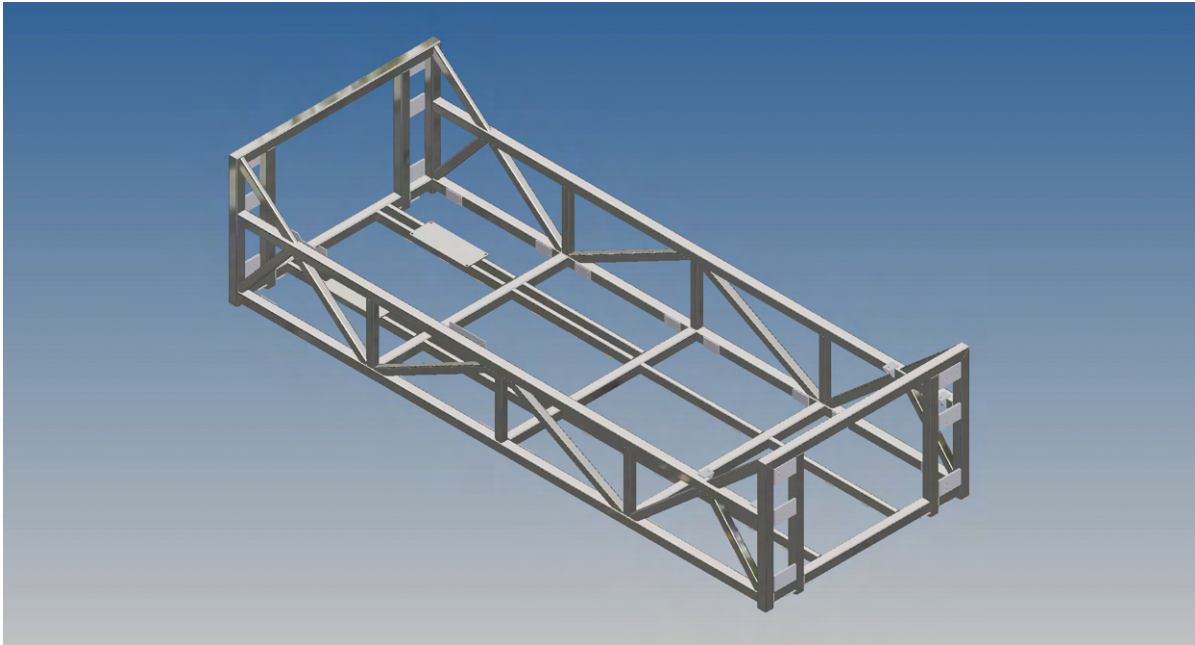


Figur 10: Første versjon av rammeverk som ble designet for å 3D printes.



Figur 11: Rammeverk versjon 2 som ble designet for å kunne laserkutte deler i kryssfinerplater.

Etter å ha fått tilbakemelding fra bedriften om at de ønsket et rammeverk isteden for en boks så ble bruken av navnet ”boksen” også endret. Som kan ses på fig. 10 så ble det første designet et rammeverk som skulle se så likt ut som SSI sitt design som mulig, se fig. 12.

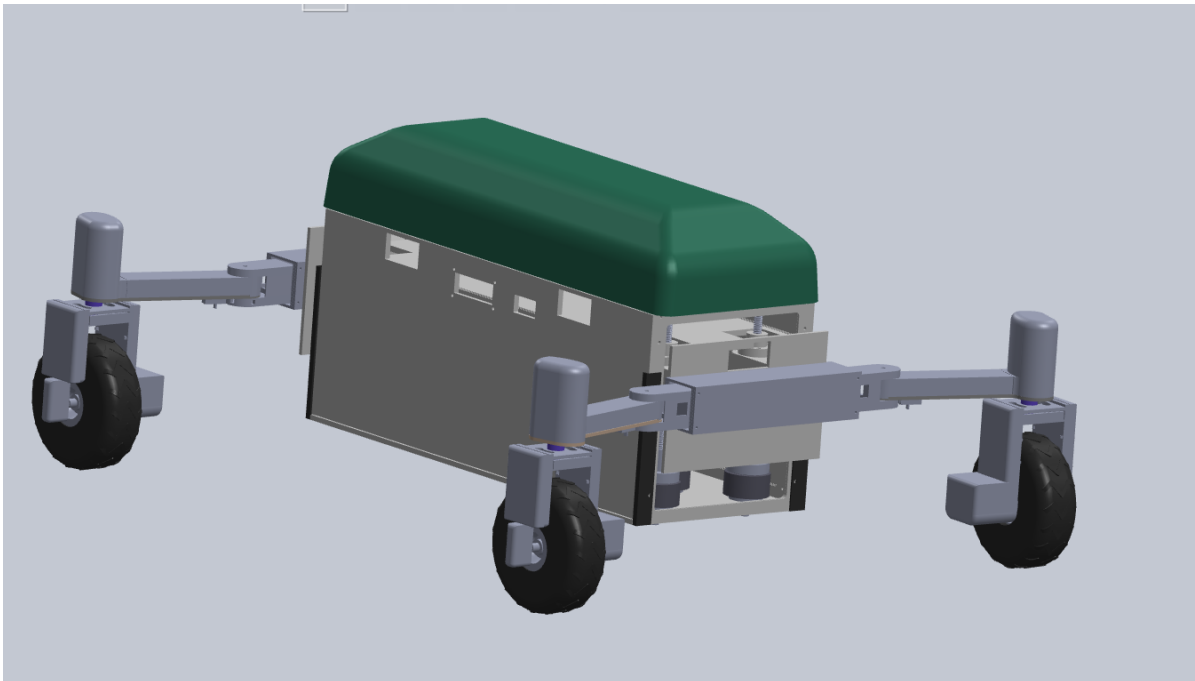


Figur 12: SoilSteam International sitt desin av rammeverk.

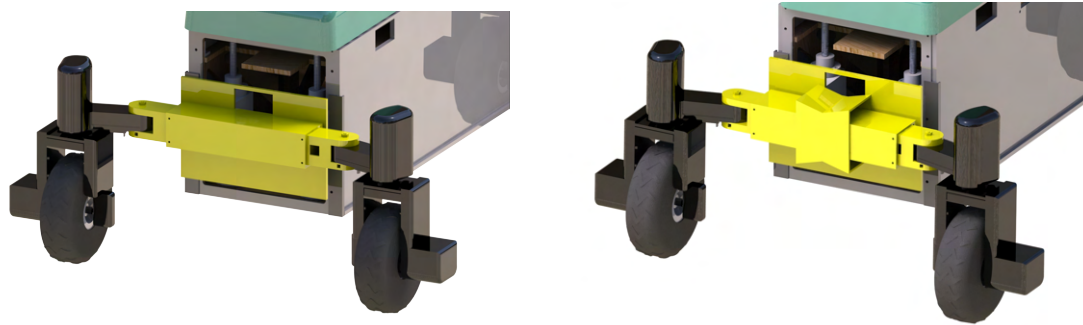
Når vi fikk tilbakemelding om hvor lang tid det ville ta å 3D printe hele rammeverket fikk vi et forslag fra faglærer (Richard Thue) om å re-designet for å kunne laserkutte kryssfinerplater, som kan ses på fig. 11. Her ble det designet for å kunne enkelt ta av og på langsideplatene, og for at platene på innsiden kan kunne skyves inn så de er enkle å byttes ut det trengs i senere tid. Vi fulgte tipset siden kryssfiner er et lett materiale, både i vekt og i å jobbe med. Det ble valgt å lime sammen flere plater, dette gjør at platene er stivere og sterkere enn om det bare er en plate. Denne metoden gjør det også mye lettere å kunne gjøre endringer når det er nødvendig, etter som at laserkutting gikk raskt og vi hadde tilgang på laserkutteren når det skulle være, og trengte ikke venter flere timer på at det skulle bli ferdig som med 3D-printing.

Når rammeverket ble ferdig designet så ble det bestemt å ha sidevegger også, disse veggene skulle egentlig bli laget i glassfiberkompositt også som innleggsplatene **Komposittdeler**. Men

dette ble endret til å ha laserkuttede kryssfiner plater for enkelhets skyld. Her laserkuttet vi også ut hull for avstandssensorer og LCD-skjerm se fig. 13. Her kan man også se skulderen og heiseveggen som kobler kroppen til armmodulene se fig. 14. Skulderen er bygd opp av både 3D-printede deler og laserkuttede kryssfinerdeler. Boksen som er på fronten er for vippeleddet og er 3D-printet. Heiseveggen er festet til heisemekanismen med et feste ledd som ble 3D-printet i PETG, dette er fordi vi ønsket noe som er mindre sprøtt men sterkere enn PLA plast. Se fig. 15.



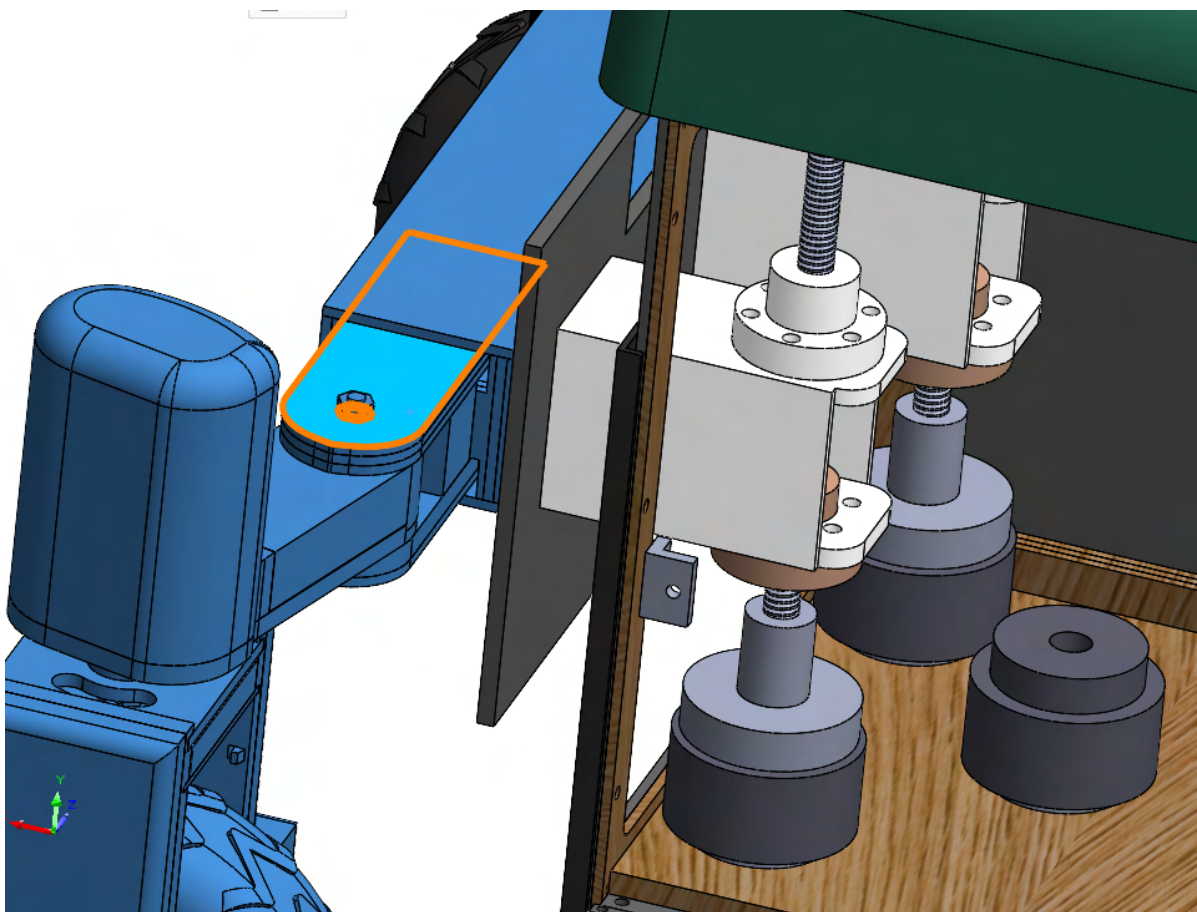
Figur 13: SoilRover fullmodell.



(a) Skulder uten vippeledd

(b) Skulder med vippeledd

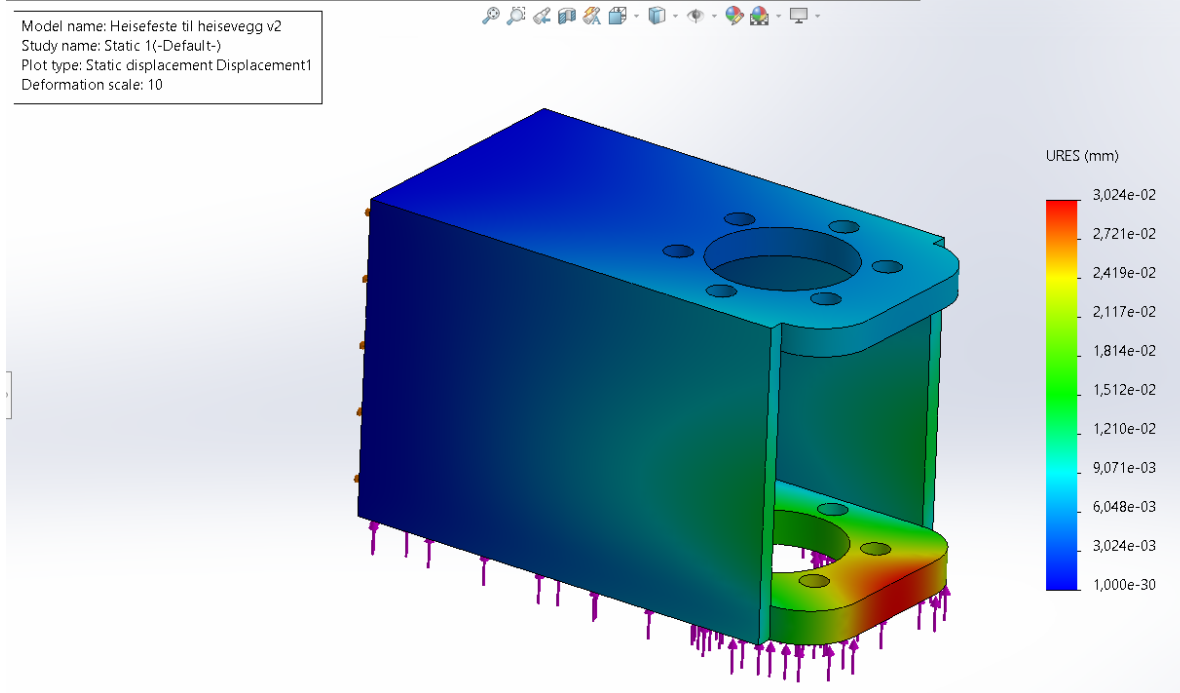
Figur 14: Skulder er uthevet i gult



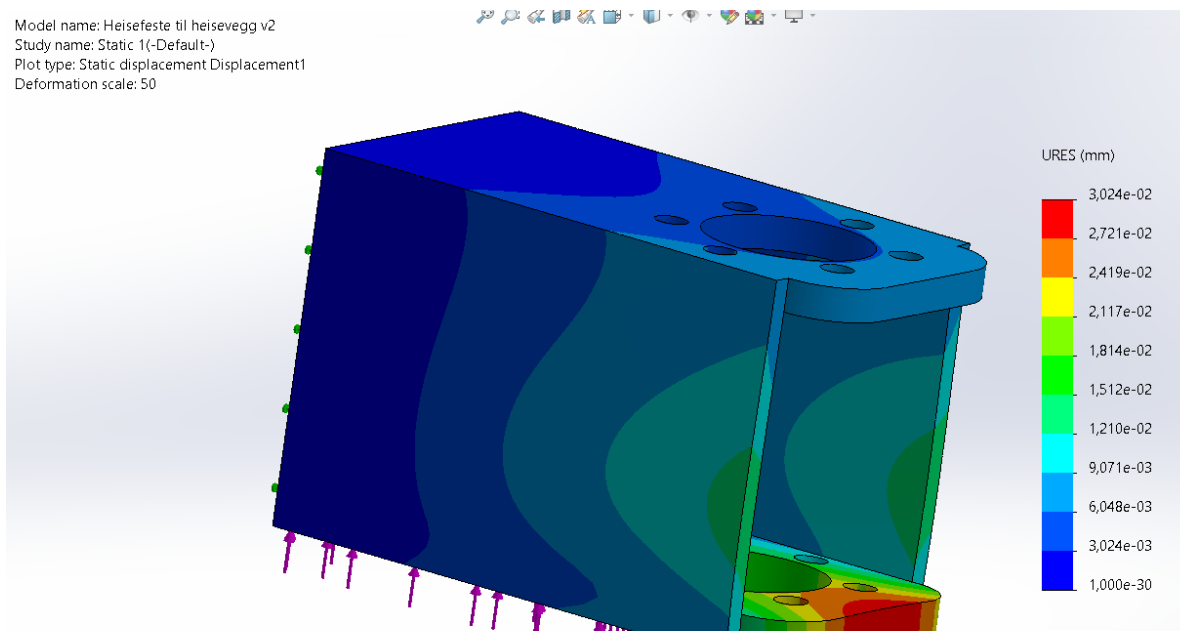
Figur 15: Feste til heiseveggen er festet til heiseskruen, den hvite delen som er på heiseskruen.

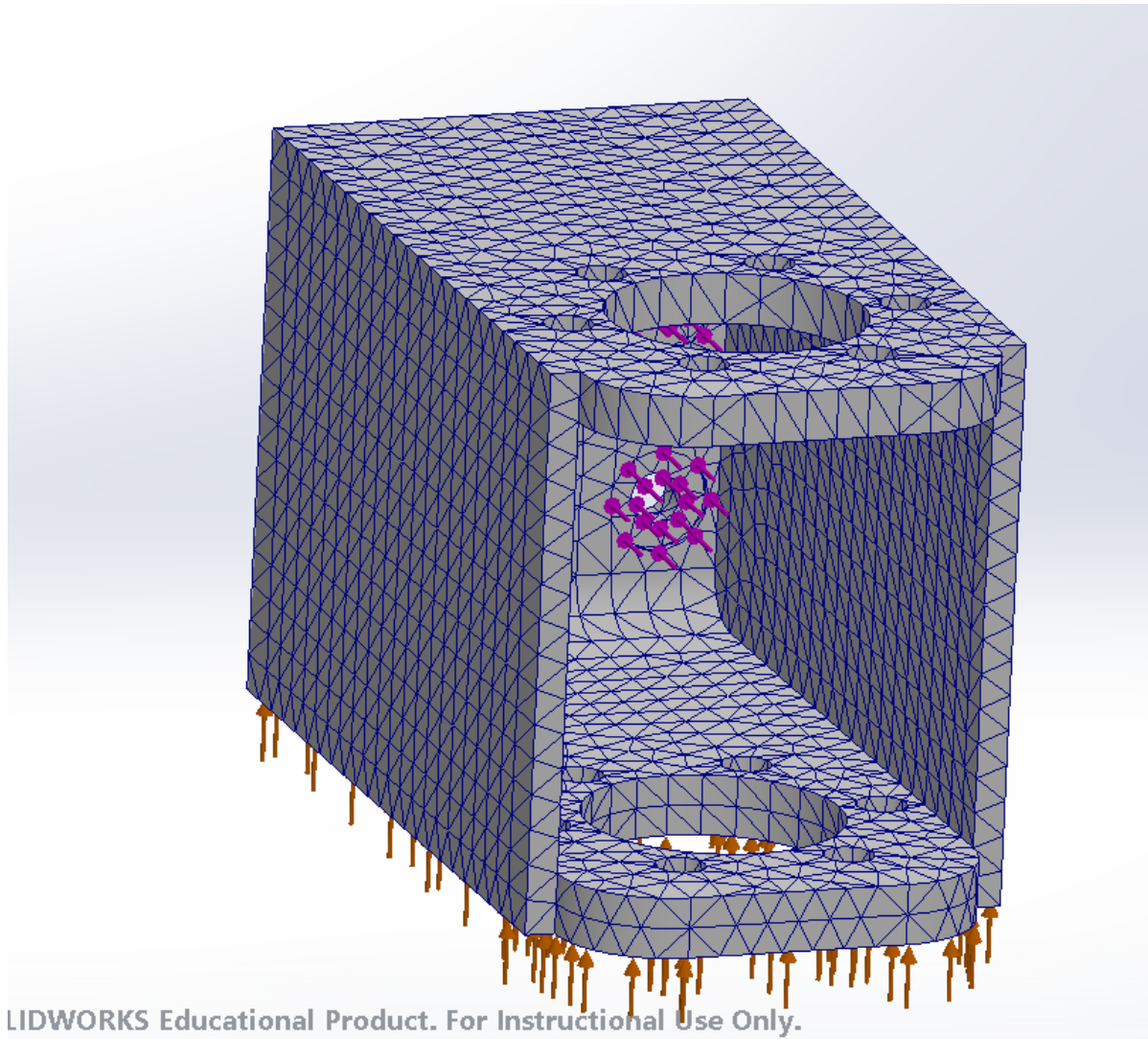
Feste til heiseveggen blir utsatt for spesielt mye vekt og bevegelse. Ettersom at dette er en

utsatt del ble den også sjekket med FEA (finite element analysis) for å se hvordan den ville holde vekten og bruken. Etter som at Solidworks ser på alle materialer som om de er massive så vi har satt en høy sikkerhetsfaktor for at det skal bli realistisk hvordan delen vil reagere på kreftene påført, se fig. 16. Her kan vi se at selv med en deformasjonsskala på 50 som vil si at de påførte kreftene blir gjort 50% større enn den største deformasjonen [14], kan vi se at deformasjonen er minimal, se fig. 16b. Analysen ble gjennomført ved at det ble satt på krefter og kontakter for å simulere hvordan den vil stå og vekten den vil holde når det er montert. Det settes på et mesh som kan justeres om man ønsker å ha det finere eller grovere. Dette er for om man ønsker å få et mer detaljert blikk på enkelte steder på en del så kan du sette på et finere mesh noen steder og grovere andre steder som ikke er like viktige å få nøyaktig. Dette krever en del fra PC-en som gjør analysen, desto finere mesh man har vil kreve mere PC kraft og øker tiden analysen vil ta. Vi gikk for automatisk mesh, se fig. 17. Når analysen går gjennom og er ferdig får vi stress plot, deformasjon og strain plot. Vi er mest interessert i deformasjonen på delen.



(a) Analyse: deformasjon med skala på 1



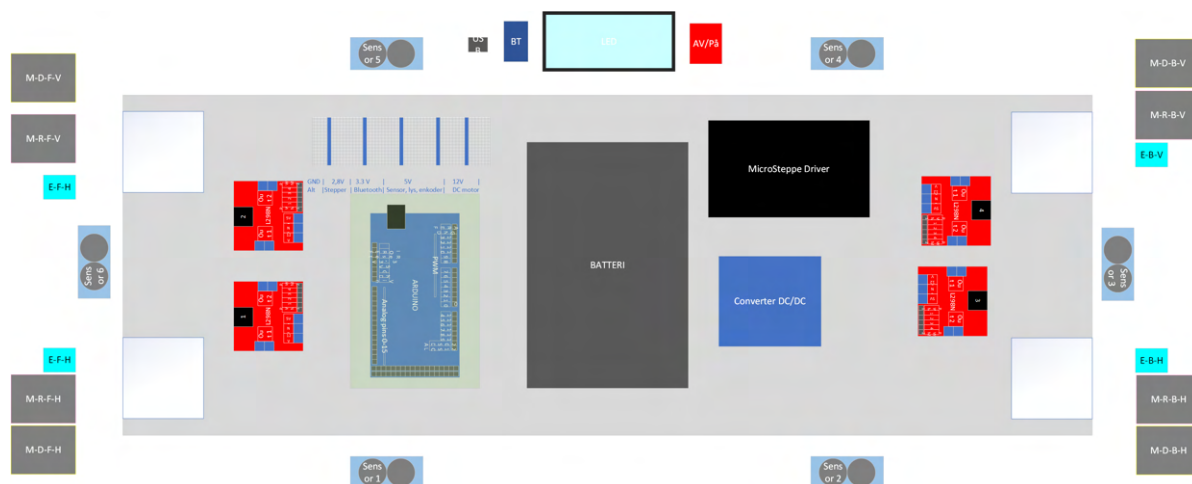


Figur 17: Mesh av heisevegg feste versjon 3 for analyse.

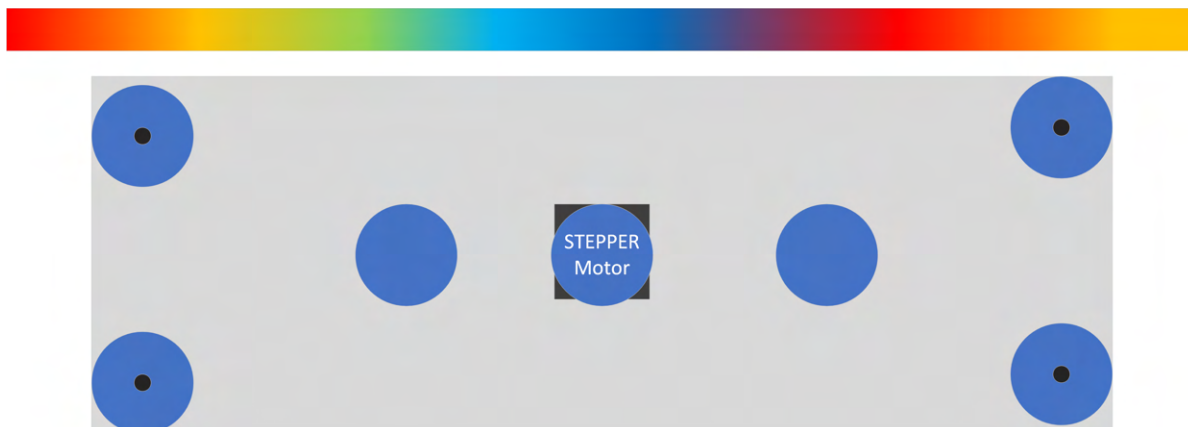
4.2 Innside

4.2.1 Oversikt oppsett av komponenter i kropp

I fig. 18 og fig. 19 ser vi planlagt oppsett av de to platene til kroppen av SoilRover. Topplaten består for det meste av elektronikk, utenom motorer og enkodere til armmodulen som man kan se visualisert på siden og utstyret i underplaten på neste figur. Vi har visualisert plassering for valgt motorkontrollere til stepper og DC motorene, Arduino, valgt batteri og DC/DC regulator. Disse er plassert ut ifra plassen med heisemekansimen som tar plass i hvert hjørne og med tanke på fordeling av vekt. På veggene til kroppen skal sensorer og LCD skjerm samt bluetooth kobling plasseres.



Figur 18: Tenkt oppsett for topp plate av kropp



Figur 19: Tenkt oppsett for under plate av kropp

I underetasjen er oppsettet til heismekanismen, hvor det er en stepper motor med 2 trinser plassert oppå. Det er tenkt å legges tre trinser til på hver side som skal styre hvert belte. Under denne skal LED slangen festes med plassert gitter på undersiden som skal representere steaming nålene til SSIs kjøretøy.

4.2.2 Mikrokontroller

Valget mellom mikrokontrollere for styring av det fysiske produktet sto mellom Arduino og Raspberry Pi. Dette er de vanligste å bruke i lignende prosjekter og har mange brukere. Arduino har også alle av oss brukt før, i motsetning til Raspberry Pi.

I tabell [9](#) sammenligner vi begge med tanke på krav til prosjektet.

Ut ifra tabellen så ser vi at begge oppfyller kravene på hver sin måte. Raspberry Pi er litt dyrere, men har innebygd Bluetooth. Men på grunn av kjennskap til Arduino og usikkerheten til Raspberry Pi både med tanke på kobling av moduler og uhell som kan føre til strømbruk og dermed også til ødelagt maskinvare.

Tabell 9: Mikrokontroller

	Arduino	Raspberry Pi
Hva	Mikrokontroller	Single Board Computer
Operativsystem	—————	Linux based
Antall oppgaver	Looper gjennom en stor oppgave	Flere task på grunn av sterk prosessor
Pris	299 kr (uno) 499 (mega)	799 kr
Muligheter for å koble til moduler Logic level	Det meste på markedet er laget for arduino og derfor også kompatibelt uten mye ekstra 5 V, samme med de fleste moduler	Veldig lite i forhold til arduino 3,3 V, så må være veldig nøye med å koble opp moduler
Strømbrudd	Må bare starte på nytt	Kan føre til ødelagt hardware
Språk	Python, Scratch, Ruby, C, C++	C, C++
Internet/bluetooth	Moduler som må kjøpes	Innebygd

4.2.3 Flere Mikrokontroller?

I utgangspunktet så prøvde vi å bare bruke en Arduino. Vi valgte Mega 2560 på grunn av antall tilgjengelig pins. Dessverre så viste det seg at det ikke var nok likevel og vi prøvde å ha et samarbeid mellom to Arduino Mega 2560. Etter litt testing så kom vi fram til at dette ville fungere og delayet ville være minimalt.

4.2.4 Batteri

Når det kom til batteri så er den er mye av elektronikken mulig å kjøre på 12 V. Det som ikke går på 12 V er det mulig å regulere spenningen for. Derfor har vi tatt valg av et 12 V, 12 Ah LiFePo4 batteri som da har kapasitet på

$$E = 12V \cdot 12Ah = 144Wh.$$

Batteri er anskaffet fra EES(Electronics & Energy Solutions) som vi hadde dialog med, der vi kom med de forskjellige nødvendighetene til vekt, spenning og kapasitet.

Batteriet som er brukt i prosjektet er satt sammen av 4 batterier på 3,2 V i serie som utgjør at batteriet har

$$4 \cdot 3,2V = 12,8V$$

som nominell spenning totalt. Samme type batteri med 6Ah er koblet i parallell med 2 som da gir oss

$$6Ah \cdot 2 = 12Ah.$$

Dette batteriet fremstår som en bra kandidat for vårt prosjekt da ingenting har behov for mer enn 12 V for å oppnå den ønskede effekten samtidig som størrelse og vekt ikke går hardt utover prosjektet. Dette er et Lithium-jern batteri som har mange fordeler i forhold til bly-batterier, eksempelvis at vekten er lavere, kapasiteten er høyere, det har flere ladesykluser, det kan lades raskere samtidig som det er mer miljøvennlig. [15] Når vi satt opp kravene våre at

den skal kunne operere i 15 minutt fra K.04 fra tabell 25, og etter første detaljerte utregning ville systemet trekke 11,212 A antok vi. Det ville tilsvare at batteriet i fulladet tilstand ville holdt i

$$\frac{12Ah}{11,212A} = 1,070h$$

som tilsvarer 64 minutter teoretisk sett.

Det ble anskaffet 2 batterier slik at SSI kan bytte ut å bruke et annet, imens de lader det som ikke er i bruk. Batteriene som er skaffet har innebygde BMS som sørger for at batteriene ikke blir for varme eller er skadet. Med dette innebærer det at hvis batteriet står for lenge på lager uten korrekt oppsett kan det hende at spenningen på batteriet blir så lav at elektronikken inni batteriet slår seg av som fører til at vi er nødt til å lade opp batteriet ved hjelp av enten power supply eller en billader som ikke har behov for å føle noe spenning for å starte å lade. Se [Lade opp det flate batteriet fra EES med monitoring av spenningen](#) og [Gammel billader som ble vist til oppladning av batteri ved EES](#).

4.2.5 Elektrisk oppsett

I henhold til å slå av og på SoilRover, er det anskaffet en simpel av/på bryter fra Biltema som du kan se i fig. 20a. Dette er for å ha en enkel metode å slå av og på SoilRover og spare elektrisk kapasitet i stedet for å måtte koble fra batteriet.



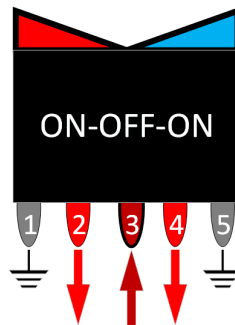
(a) Av/på-bryteren til SoilRover vist fra siden



(b) Av/på-bryteren til SoilRover vist fra toppen

Figur 20: Av/på-bryter skaffet fra Biltema

På denne måten slipper vi å koble fra batteriet hele tiden når vi skal ha den i avslått tilstand. Av/på-bryteren er lagt opp for to tilstander ved tre stadier(ON-OFF-ON) som en kan se i fig. 20a der midten er avslått og kan klikke begge veier for at den skal være på som da vil indikeres med enten blått eller rødt lys, se fig. 20b. Denne skal plasseres på siden av kjøretøyet slik at det er lett tilgjengelig for å bli avslått.



Figur 21: Diagram oppkobling knapp

4.2.6 Spenningsregulering

Vi har 2 spenningsnivåer vi er nødt til å operere på, 12 V og 5 V, derfor er vi nødt til å regulere ned spenningen for elektronikken som går på 5 V ellers kommer dette til å bli ødelagt.

Det vi har som kan gå på 12 V er alle motorene, og Arduinoen kan operere på 12 V, ellers går resten på 5 V. Stepper motoren som er brukt for heisemekanismen er oppført til å gå på 2,5 V, men motorkontrolleren til denne justerer dette ned selv. Derfor skal alle motorkontrollene kobles i parallell, mellom 12V og jord, sammen med Arduinoen. For å regulere fra 12 V til 5 V bruker vi en spenningsregulator. For alt som krever mindre strøm mengde bruker vi en lineær regulator for simplistet, men da vi skal ha med en LED-stripe som krever en del strøm er vi nødt til å ta i bruk en switching regulator også.

Den lineære regulatoren vi bruker er fra 78XX familien som en enkelt kan få tak i fra de fleste elektroleverandører. Den vi bruker er 7805ACV fra ElfaDistrelec der de siste tallene viser til

hva den regulerer spenningen til. Denne bruker vi til den mindre elektronikken på 5 V som ikke trekker så mye strøm. Effekten finner vi fra

$$P = UI, \quad (1)$$

og siden vi har et spenningsfall på 7 V over regulatoren, er vi derfor nødt til å passe på at det ikke går for mye strøm gjennom da dette vil øke effekten over regulatoren som vil føre til effekttap i form av varme. Selvom regulatoren er målt til 1,5 A, så er det hvor mye den klarer, men da er det behov for korrekt kjøling.



Figur 22: Kjøleribbe(eng:Heatsink)

I fig. 22 ser vi kjøleribben vi fikk fra skolen(USN) som er evaluert til 6k/W og skal plasseres på regulatoren for å unngå overoppheting. Dette vil si at per økning av effekt vil den øke temperatur fra romtemperatur med 6K. Eksempelvis hvis vi hadde kjørt 1 A gjennom regulator

ville vi fra (1) fått

$$P = 7V \cdot 1A = 7W,$$

som videre gir oss økningen i temperatur

$$\frac{6K \cdot 7W}{W} = 42K.$$

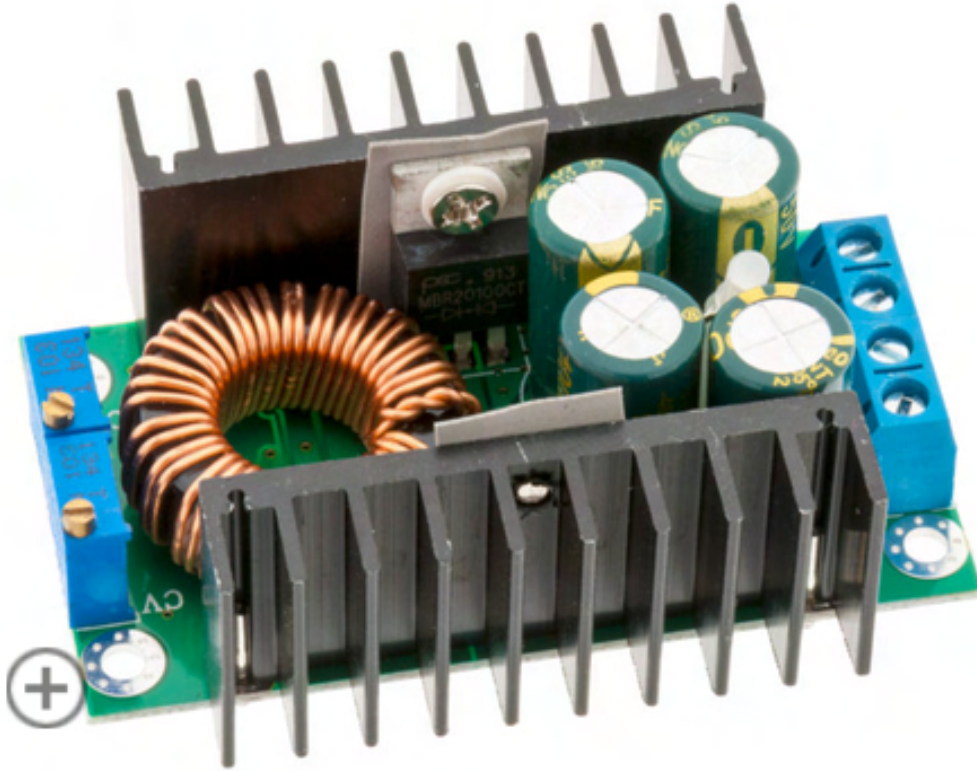
Dette er da ikke en systemfarlig varme, men er ønskelig å ikke ha komponenter så varme da dette er noe som kan forårsake brannskade på personell.

Når det kommer til LED-stripen så trekker den for mye strøm til at vi kan bruke 7805, fordi hver LED-pære trekker 60mA og det er 60 pærer så blir det for 1 meter med LED-stripe

$$60mA \cdot 60 = 3,6A.$$

Hvis vi her skulle brukt en lineær regulator så ville størrelsen/kostnad bli høy. For at vi skal kunne ha med LED-stripen for å se tilstander har vi valgt å bruke en switch regulator også. Til vanlig ville det vært stilt spørsmål til forstyrrelser av andre signaler ved bruk av en switching regulator på grunn av "ringing" ved plutselig endring av strøm og spenning[16]. Men på grunn av at regulatoren ikke blir plassert på et kretskort, og vi har mulighet til å isolere den vekk fra andre deler så vi ikke dette som et problem.

Da dette var fastsatt ble det bestilt en DC/DC konverter fra Electrokit fra Sverige med en switch regulator som kunne ha input fra 8V-37V og regulere til mellom 1,2V-35V og med max 8A ut, som da gir oss muligheten opptil vertfall 2 m med LED-stripe da dette vil trekke rundt 7,2 A. Switch regulatoren som ble kjøpt var en kraftig trapp-ned konverter med justerbar strømutgang som passer bra for oss, men levering var ikke før dokumentasjon skal inn og blir derfor ikke testet eller satt i bruk før dokumentasjonen blir levert inn.



Figur 23: DC-DCKonverter som er handlet fra electronickit

Ut ifra informasjonutveksling med produsent sa de at effektraten var på rundt 90% for dette produktet og siden lysene går på 5V og har behov for 7,2 A for 2 m med lys så regnet vi ut hvor mye effekt som ble trekt fra denne. Effekten fra LED-stripen blir på 90% av effekten som regulatoren får gitt. Da finner vi ved hjelp av (1) effekten til LED-stripen.

$$P = 5V \cdot 7,2A = 36W$$

som vi da bruker til å finne hvor mye som går inn i regulatoren.

$$0,9x = 36W$$

$$x = 40W.$$

Dette er da effekten som er potensialet utfra regulatoren, men det er noe effekttap over den.

Dette gir oss da videre

$$A = \frac{40W}{12V} = 3,333A.$$

Dette gjør da slik at vi minker kjøretiden i forhold til det som først var antatt, se **Første detaljerte antagelser av elektrisk kapasitet**. Den nye kjøretiden kan bli beregnet utfra tabell **(28)**.

4.2.7 Heisemekanisme

Heisemekanismen består av fire trapesskruer "lead screw" og muttere for hver skruer. Disse skruene vil bli drevet av belter, en stepper motor og trinser. Slik den er ønsket å funke, så vil beltene bli drevet av trinser på stepper motoren som overfører kraften til skruene, her vil skruene rotere mens mutterene vil i gjengjeld bli heiset opp eller senket nedover skruen. På mutterene så er det et feste som skrur fast til heiseveggen hvor hjulopphenget vil bli festet. Så når mutterene beveger seg vil hjulopphenget også bevege seg.

En stepper motor ble valgt for å styre løftmekanismen. Stepper motorer er ideell for nøyaktig bevegelser og har god kraft og nøyaktighet ved lav fart. Men igjengjeld blir de svært ineffektive ved høyere hastigheter. Til heisemekanismen går vi heller for nøyaktighet over fart.

Stepper motoren ble valgt ut i fra utregnet moment som er nødvendig for å kunne drive skruene, dette kan avvike med uforutsette variabler. Vi brukte formlene **2** og **3**, fra **[17]**. Her fant vi T_R som er nødvendig dreiemoment som skal til for å over komme gjenge friksjonen og for å heise vekten og T_L som er nødvendig dreiemoment for å senke vekten. Se tabell **10**.

$$T_R = \frac{F \cdot d_m}{2} \cdot \frac{\pi \cdot f \cdot d_m + l}{\pi \cdot d_m - f \cdot l} \quad (2)$$

$$T_L = \frac{F \cdot d_m}{2} \cdot \frac{\pi \cdot f \cdot d_m - l}{\pi \cdot d_m + f \cdot l} \quad (3)$$

Tabell 10: Variabler til (2) og (3)

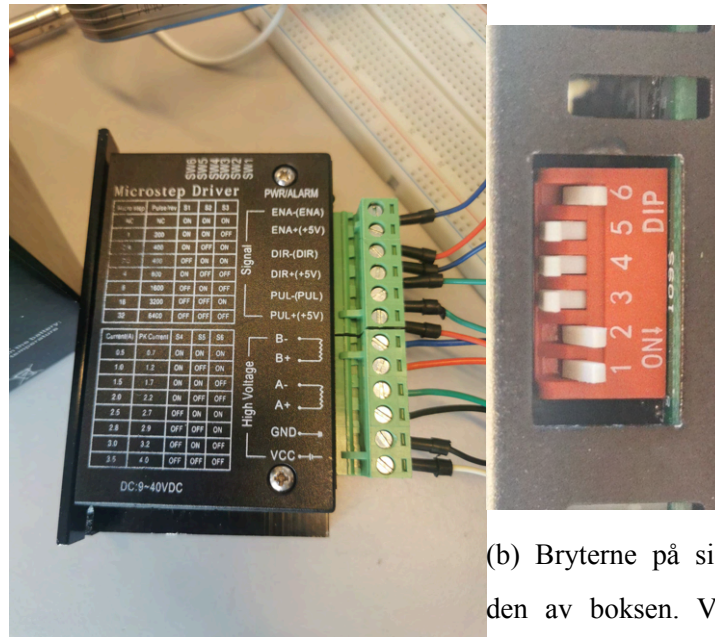
F	vekt i newton
d_m	diameter på skruen
f	friksjons koeffisient [18]
l	lengden på gjengene

Basert på beregningene gjort valgte vi å gå for motor 535-0439 som er en hybrid stepper motor på 1,26 Nm i holding torque [19].

Tabell 11: Stepper motor info

Holding torque	1,26 Nm
Faser	2
Skaft	Rundt
Spenning	2,5 V
Strøm	2,8 A
Pris	542 kr

Vi fant at vår originale motorkontroller L298N ikke kunne brukes til å styre vår stepper motor. L298N har en maks utgangsstrøm på 2 A, hvor vår stepper motor trekker 2,8 A gjennomsnittlig. Så for å kunne styre motoren trenger man en sterkere motorkontroller. Vi fikk teste en TB6600 Microstepper driver som kan gi opp til 3,5 A basert på hvordan man flipper bryterne. Microstepper driveren har en kort forklaring av hva man må stille inn bryterne på siden til for å få ønsket bruk. De tre første bryterne gir kontroll over hvor mange pulser per rotasjon motoren har og de tre siste bryterne bestemmer hvor mye ampere motoren går på.



(a) Microstepper driver TB6600. Teksten på toppen beskriver hvilke brytere man skal flippe for å stille inn riktig.

(b) Bryterne på siden av boksen. Vi flippet bryterne satt til å si 200 pulser per rotasjon, med 2,8A

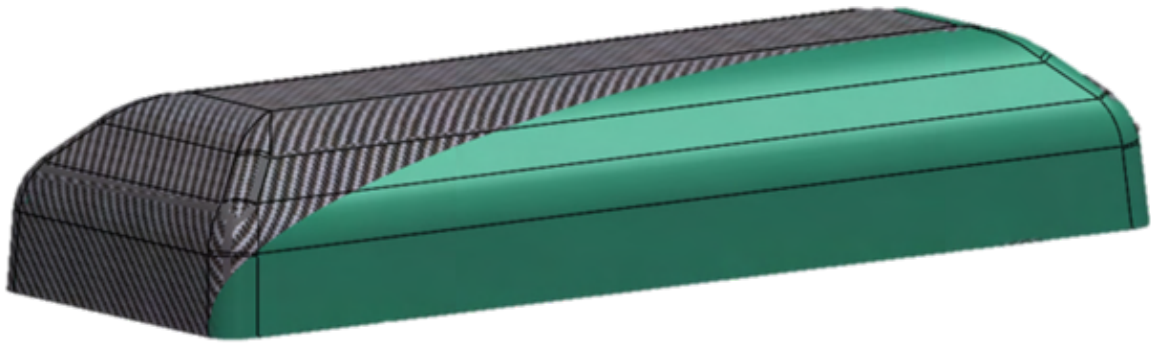
Styring

Med en enkel referanse kode [20], fikk vi satt opp motoren klar til testing. Så langt har vi plassert opp trinsene og beltene med 3D-print nesten som det skal være og sjekket at alt går rundt. Vi ser med disse testene at motoren blir mindre nøyaktig med last. Dette skal vi ta med i vurdering av videre utvikling av kode for videre oppsett.

4.3 Deksel

Dekselet er i hovedsak bare for designet, at det skal ligne designet til SSI. Men SSI ønsket også spesifikt at dekselet skulle bli laget ut av karbonfiber etter som at denne nedskalerte modellen

er ønsket å ta med på messer så de ønsket noe som kunne gripe oppmerksomheten. Noe som vi fant ut at dekslet vil nok gjøre med designet som ble bestemt se fig. 25.



Figur 25: Designet som ble valgt sammen med SSI, et karbonfiber deksel med spraylakk av SSI sin ønsket farge.

Dette blir da laget av karbonfiberkompositt, så det vil bli brukt en form for å lage dekslet. Denne formen ble laget i 3 hoveddeler og med flenser. Alle delene har blitt limt på og holdt under press for at det skulle bli limt godt sammen se fig. 26a og fig. 26b.



(a) Full deksel form limt og flenser på limt sett fra utsiden.



(b) Full deksel form limt, sett fra innsiden.

Figur 26: Dekselformen ferdig limt, trenger å pusses ned før kan spray lakkes.

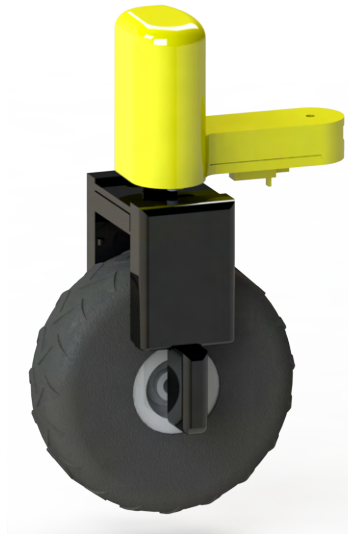
Når dekselformen er blitt pusset og rengjort så kan formen bli spraylakkert for å få en jevn overflate før vi starter med karbonfibermattene. Før mattene legges så må overflaten renses for både støv og fett, når dette har blitt gjort så vil vi legge flere lag med slippmiddel for at delen skal kunne enkelt tas ut. Så fort alt dette er gjennomført så kan vi starte med epoksy og mattene. Her vil vi bruke epoksyen svapox 110 med herder TL-1. Epoksyen og herderen er blandet til 24% herder og 76% epoksy. Denne blandingen har et «potlife» på 80 til 90 minutter. Dette vil si at etter 80 til 90 minutter så vil blandingen begynne herdeprosessen, dette er ut fra datablad for disse materialene [21]. Når alle lagene har blitt mettet med epoksy, så vil vi prøve å få ut så mye som mulig av luften mellom lagene med en rulle, her er det viktig å ikke dra for hardt etter som at vi ønsker ikke å ødelegge flettingen av matten.

5 Armmodul

I dette kapitlet vil vi ta for oss armmodulen se fig. 27. Armmodulen består av armen (se fig. 28) og hjulmodulen (se fig. 29). Under hjulmodul kan man lese om motor og enkoder til driv og mekanisk design av hjulmodul. Under arm kan du lese om rotasjonsmotor, styring av rotasjon av hjulmodul, Ackermann-styring og mekanisk design.



Figur 27: Armmodul



Figur 28: Armmodul med arm uthevet i gult



Figur 29: Armmodul med hjulmodul uthevet i gult

5.1 Hjul modul

5.1.1 Valg av hjul

Før vi kjøpte hjul var det flere faktorer å ta hensyn til; tåler de 5 kg per hjul? Hvor mye kraft trengs for rotasjon? Er det lett å feste noe til felgen? Hvor raskt kan vi få hjulene? Hvor mye ligner de på SSIs produkt? Et ideelt hjul hadde vært et hardt og relativt glatt hjul, siden vårt produkt skal kjøre innendørs eller på hardt underlag som asfalt. De tåler generelt mer vekt

og det er mindre friksjon når du roterer hjulet. Men vi følte det var viktigere å kunne visuelt signalisere at vår modell var basert på et landbrukskjøretøy, og den letteste måten er å bruke hjul som ligner på traktorhjulene SSI bruker. Vi kjøpte hjul ment for terreng RC-biler med dimensjoner nærmest lik 1:10 av SSI's hjul som vi kunne få ta i. De vi fant var ikke laget for å tåle 5 kg. Dekkene var limt til felgen og fylt med en myk svamp. Vi løsnet dekket og byttet ut svampen med rørisolasjon. Hjulet ble da hardt nok til å tåle vektkravet vårt, i tillegg blir det mindre friksjon når hjulmodulen roterer. Å få ned den friksjonen er viktig siden dekkene er svært klebrige. Hjulene er fortsatt myke og fungerer som dempere.



(a) front side



(b) bakside med heksagon utkutt

Figur 30: Kjøpte hjul

5.1.2 Valg av driv motor

En av de større design faktorene til hjulmodulen var å få implementert en sterk nok driv motor. Det er et variert valg av motorer ute i markedet, DC eller AC motorer, børstet eller børsteløs og stepper motorer. Kilde ([22]) ga oss en oversiktlig start oversikt over disse forskjellige motortypene som hjalp oss velge en standard type DC-motor som vår driv motor. Dette er på grunnlag av at AC-motorer (alternative current) går på en annen strømforsyning enn det resten av SoilRover går på, batteri som gir DC strøm (direct current). Dette krever da mer planlegging og komponenter for å få til å konvertere over fra DC til AC, slik at motoren kan fungere. Vi valgte også bort de to andre vanlige motorene, servo og stepper, hvor disse er mer ment for bruk av nøyaktige bevegelser. En DC motor er en av de vanligste motorene i bruk for driv av

roboter.

For vårt valg av riktig motor handler neste steg om å velge en med nok dreiemoment. En DC motor klarer å få høyt dreiemoment, men man ofrer RPM (rotasjon per minutt) som gir farten. Jo høyere kraft, jo mindre fart og motsatt. Men siden vårt kjøretøy ikke er ment for å kjøre fort, så er ikke RPM like viktig som høyt dreiemoment som tåler å flytte på 20 kg. Vi baserte oss også på å oppnå en fart på 5 km/t når vi gjorde beregningene for motoren.

Beregningene baserte vi oss på kilde ([23]) som gir dyp info om oppbygging av de forskjellige ligningene og hvorfor de trengs. Våre parametere for SoilRover kommer fra våre forventninger og krav til produktet. Disse parameterne er som følger:

Tabell 12: Drivmotor info

Masse	20kg
Maks fart	1,4 m/s
Maks helling	12 °
Hjul radius	0,0755m

Ut ifra valgt hjul ble hjul radiusen satt til 0,0755m. Kravet for maks vekt av kjøretøyet ble satt til å være 20 kg da arbeidstilsynet har satt en anbefaling på maks enkelt bærevekt [24]. Vi valgte å følge denne anbefalingen fordi vår SoilRover vil bli transportert i bagasjerom og må derfor bæres inn og ut av bagasjerommet. Tanken bak maks fart på 1,4 m/s var å sette hastigheten til normal gange og saktere, som er 5 km/t (1,4 m/s).

Med tanke på at alle hjul skal kunne rotere individuelt og at SoilRover skal kunne kjøre i mange diverse retninger, så har vi satt at vi trenger en driv motor per hjul for å få lik driv i alle de diverse retningene. Dette gir oss 4 motorer å fordele kraften over. Vi har også satt at spenningen i SoilRover skal være maks 12V med tanke på batteristørrelse. For å kunne finne riktig motor trenger vi å finne:

• Dreiemoment

• Effekt

• RPM

• Akselerasjon

For å finne nødvendig dreiemoment trenger man å bruke $kraft \cdot radius$. Akselerasjon satte vi som $0,7 \frac{m}{s^2}$. Med denne infoen regnet vi ut de resterende parameterne. Dreiemomentet fant vi ved:

$$T = \frac{Masse \cdot radius \cdot (akselerasjon + g \cdot \sin(Helning))}{Antallmotorer} \cdot \frac{100}{\mu} \quad (4)$$

$$T = \frac{20kg \cdot 0,0755m \cdot (0,7 \frac{m}{s^2} + g \cdot \sin(12))}{4} \cdot \frac{100}{50}$$

$$T = 2,06$$

Hvor μ utgjør tenkt effektivitet av systemet. Med dreiemomentet ble effekten som kreves:

$$P = T \cdot \frac{maks\,fart}{radius} \quad (5)$$

$$P = 2,06 \cdot \frac{5 \frac{km}{t}}{0,0755m}$$

$$P = 38,2W$$

Mange motorer spesifiserer dens maks RPM fart, så vår krav for RPM fant vi via å finne RPS:

$$RPS(\text{Rotasjonpersekund}) = \frac{\text{maksfart}}{2 \cdot \pi \cdot \text{radius}} = \frac{1,4 \frac{m}{s}}{2 \cdot \pi \cdot 0,0755m} = 2,95rps \quad (6)$$

$$RPM(\text{Rotasjonperminutt}) = RPS \cdot 60s = 2,95rps \cdot 60s = 177rpm \quad (7)$$

Etter å ha forhørt oss med forskjellige fagfolk om parameterne vi fant, valgte vi motor fra RS Components med stock nr. 834-7669 [25]. I tabell 13 ser vi oversikt over dens parametere. Den funker på opptil 15V, hvor den da får en maks fart på 194 rpm. Som er mer enn vi trenger for å oppnå en fart på $5 \frac{km}{t}$. Den var en av få komponenter som var på lager og klar til å sendes med en gang, hvor vi så på en annen motor med stock nr. 834-7663 som hadde høyere RPM som ikke lå inne på lager. Vi hadde uansett tatt første motoren siden i vårt tilfelle er kraft viktigere enn fart.

Tabell 13: Driv motor 834-7663 info

Effekt	41,3 W
Dreiemoment	78 Ncm
RPM	194 rpm
Gir rate	51:1
Spenning	6V - 15V
Pris	407 kr

Selv om fart ikke er prioriteten i prosjektet, så er det at alle hjul kjører like fort viktig for å kunne kjøre rett. Vi bestilte da også 4 inkrementelle optiske enkodere for å måle farten til hvert hjul. Det var viktig å velge en enkoder som var liten, da vi helst ville feste den til og lese av hjulet direkte. Jo større enkoderen kunne ha vært, jo vanskeligere å få plass til den og feste den stabilt til hjulet. Vi ville også ha en enkoder med et skaft som ikke var rundt, for bedre grep. De vanligste skaftene var da formet som en D. Vi hadde ingen krav til hvor mange punkter per

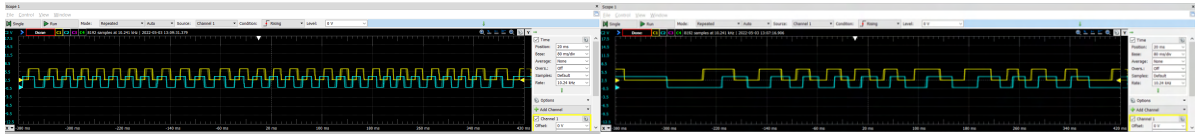
rotasjon (ppr) enkoderen skulle ha, da dette skal brukes til å måle driv, og vi bare trenger å se hvor mange ganger den tar en rotasjon, og dermed egentlig bare trenger 1 ppr.

Valget ble enkoder HRPG-ASCA#16C fra Avago Technologies/Broadcom [26], som er en 2 kanals enkoder, 120 ppr med D-cut skaft. Når enkoderen har 2 kanaler gjør det at vi kan se hvilken vei enkoderen spinner og skille mellom frem og tilbake. Denne enkoderen var også ekstra spesiell med at den roterte jevnt. Det er vanlig for inkrementelle enkodere å kjenne alle ppr som hakk.

Tabell 14: Driv Enkoder HRPG-ASCA#16C info

Anbefalt fart	300 RPM
	smooth 120ppr
Skaft	D-cut
Kanaler	2
Spenning	5V
Pris	332 kr

Det finnes flere typer enkodere hvor magnetisk og optisk er de vanligste. Vi valgte å gå for en optisk roterende enkoder. Optiske enkodere har veldig enkle design, hvor man bruker en LED og en skive med hull for å sende signaler. Når enkoderen roterer, vil skiven rotere med og lyset fra LEDet vil passere hullene som de roterer forbi. Når lyset skinner igjennom treffer det lyssensorer på andre siden, som gir ut analoge pulser. Hvert hull er et punkt. Med en rad av hull får man vite fart og rotasjon, men ikke retning. Ved hjelp av 2 kanaler, 2 lys sensorer, får vi to analoge utganger og basert på hvem som pulserer først, kan man vite hvilken retning enkoderen spinner.



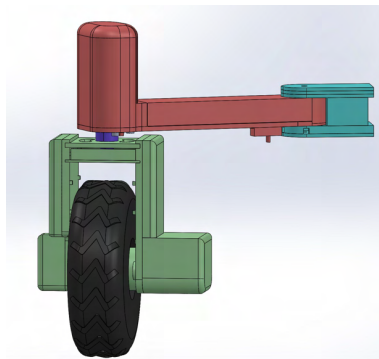
(a) Enkoder mål når vridd i motsatt klokkeretning, (b) Enkoder mål når vridd i klokkeretning med kanal A leder hvor kanal A leder
kanal B som leder.

5.1.3 Mekanisk design av hjulmodul

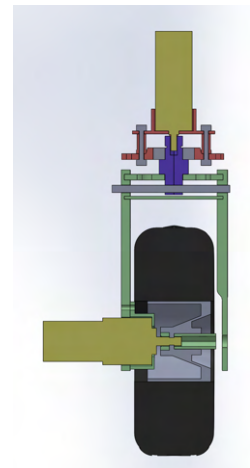
Det grønne på fig 33 er hjulmodulen som er det vi tar for oss i dette delkapitlet. På grunn av størrelsesforskjell vil det være ulike designvalg for vår SoilRover og SSI sin. Når vi designet drivet valgte vi de designene som virket mest realistiske at vi klarte å gjennomføre i tide. Siden vi har separat motor og enkoder fant vi at den enkleste løsningen var å ha feste på begge sidene av hjulet, i ulikhet med SSI.



Figur 32: Hjulmodul



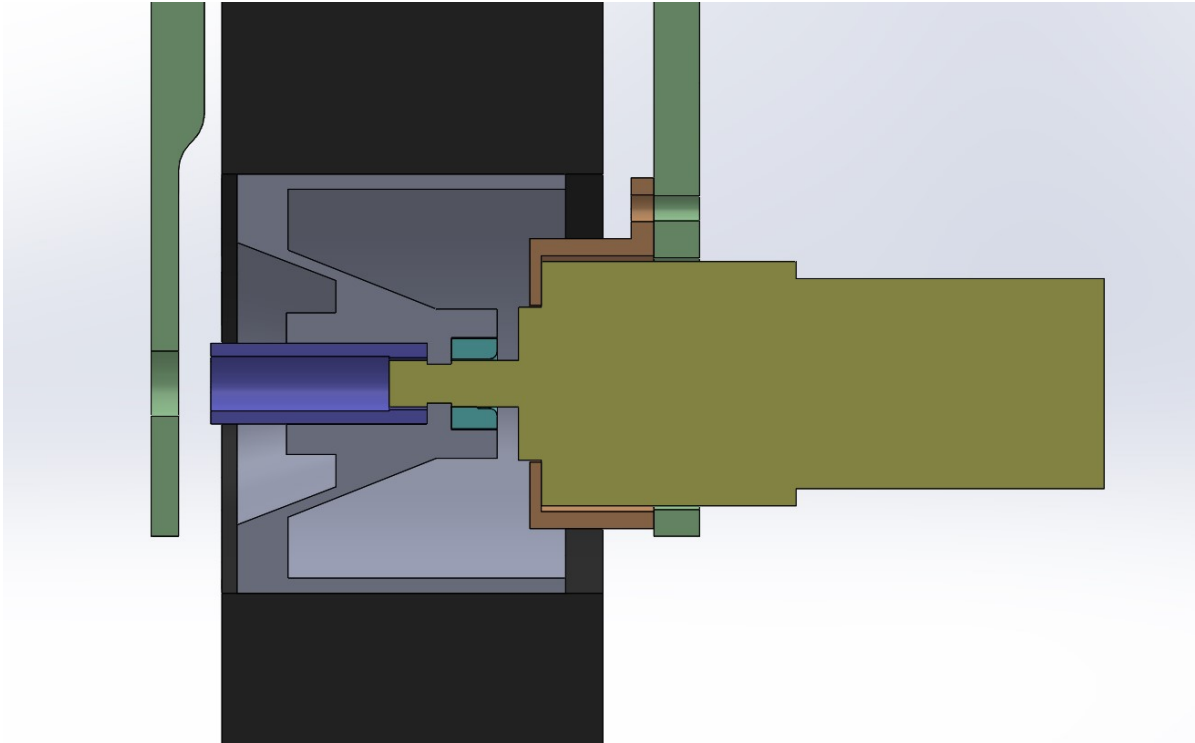
Figur 33: 3D-modell av arm-modul



Figur 34: Seksjonsbilde av hjulmodul og rotasjonsledd

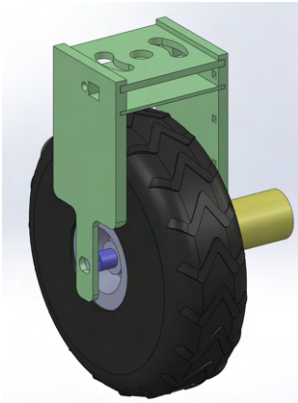
Tabell 15: Beskrivelse av deler i fig. 35

3D-modell	Farge	Beskrivelse	Materiale
	Gul	Drivmotor: DC motor med D-shaft	-
	Grå	Felg: Grov modulert opp model av den innkjøpte felgen	Ukjent plast
	Svart	Dekk: Grov modulert opp. Mønster er illustrativt og reflekterer ikke virkelighetetn.	Ukjent gummi
	Lyseblå	HeksagonFeste: del med D-hull til drivaksel. Overfører kraften fra motor til	3D-print PLA
	Mørke-blå	EnkoderFeste: Aksel med D-hull til enkoder og drivmotor	3D-print PLA
	oransje	DrivmotorHolder: Hindrer motoren fra å rotere og fester den til rammeverket.	3D-print PLA
	Grønt	Rammeverk: Holder konstruksjonen sammen.	3D-print PLA

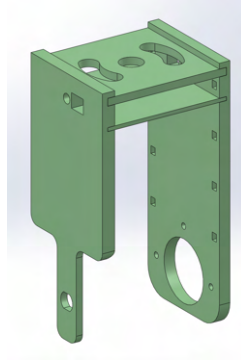


Figur 35: Seksjonsbilde av hjulfestet

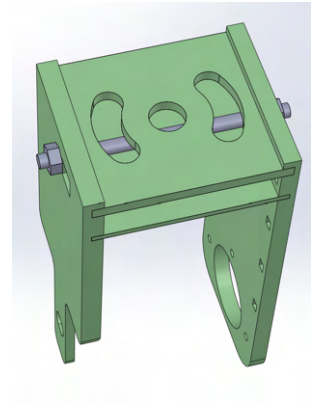
Tabell 15 gir en beskrivelse av deler i fig. 35. På fig. 35 kan man se at drivmotorakselen går gjennom felgen. Akselen er en millimeter større enn felgets hull så vi drilllet det opp. Drivmotoren og enkoderen festes direkte til felgen og vil dermed bære vekten av SoilRover. For hver rotasjon vil de oppleve en stresssyklus, men siden produktet ikke er for daglig bruk har vi vurdert at det går bra.



Figur 36: Hjulmodul



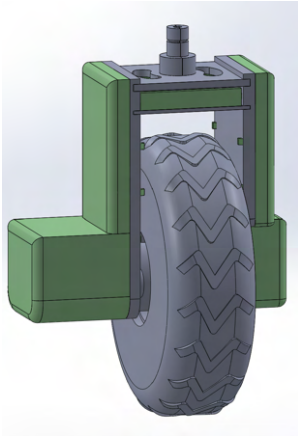
Figur 37: 3D-modell av armmodul



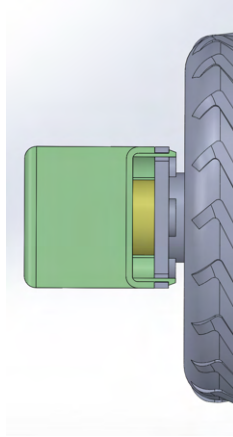
Figur 38: Seksjonsbilde av hjulmodul og rotasjonsledd

Rammeverket til hjulmodulen holdes sammen av en gjenget stang. Toppflaten av rammeverket har hull til ledningene fra enkoderen og drivmotoren. Platen under toppflaten er for at ledningene ikke skal treffe hjulet.

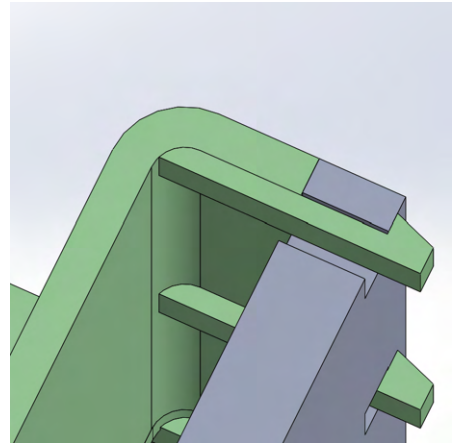
Hele hjulmodulen skal til slutt bli dekt med et deksel. For å gjøre dekslet fint er det tenkt å legge over et eller flere lag med epoxy. Dette glatter ut imperfeksjoner som kommer med 3D-printing. Etterpå skal vi spraymale dekslet svart. Dekselet festes ved hjelp av cantilever snapfit joints[27].



Figur 39: Hjulmodul



Figur 40: 3D-modell av arm-modul



Figur 41: Seksjonsbilde av hjulmodul og rotasjonsledd

Grunnen til at vi valgte å bruke 3D-print på de delene som ikke er kjøpt er materialets egenskap til å være sterk men lett. Det er også enkelt, lite tidskrevende og relativt raskt å printe. Det gjør det mulig å lage mange iterasjoner, og gir oss frihet til å ta sjanser. Når det er sagt så kan etterbehandling av 3D-print ta lang tid avhengig av delen.

Forbedringspotensiale: HeksagonFeste (se tabell[15]) har lett for å ødelegge D-sporet sitt, noe som fører til at motoren ikke lenger overfører kraft til hjulet. Hovedårsaken til at sporet blir ødelagt er at den blir dyttet over på den runde delen av motorakselen når EnkoderFeste monteres. Dette kan bli løst med å ha en liten plate der som stopper HeksagonFeste fra å gå over på den runde delen.

5.1.4 Implementasjon

Ved hjelp av diverse referanse koder for kontroll av standard DC motor og avlesing av enkoder fikk vi utført diverse tester. Vi fikk videre implementert avlesing av enkoderen til å vise farten

i km/t og testet først maks farten til hjulet uten last på 9 V batteri. Vi så at potensialet til drivmotoren var da ikke like mye som ønsket, rundt 3 km/t. Men når vi fikk byttet over til 12 V batteriet fikk vi se at hjulet roterte over 5 km/t uten last. Motoren kommer til å ikke oppnå farts målet med lasten. Motoren fungerer på en spenning mellom 6 V og 15 V, så hvis vi hadde brukt en spenning på 15 V kunne vi kanskje oppnå målet på 5 km/t med last. Videre er det tenkt å teste på selve SoilRover, slik at vi kan prøve ut fartskontroll av motorene.

kmt-3.10	rpm-109 -220	kmt-5.52	rpm-194 -390
kmt-3.10	rpm-109 -218	kmt-5.55	rpm-195 -391
kmt-3.10	rpm-109 -218	kmt-5.55	rpm-195 -391
kmt-3.07	rpm-108 -217	kmt-5.55	rpm-195 -390
kmt-3.10	rpm-109 -218	kmt-5.52	rpm-194 -389
kmt-3.10	rpm-109 -218	kmt-5.55	rpm-195 -390
kmt-3.07	rpm-108 -217	kmt-5.55	rpm-195 -391
kmt-3.07	rpm-108 -217	kmt-5.52	rpm-194 -389
kmt-3.04	rpm-107 -215	kmt-5.52	rpm-194 -389
kmt-3.04	rpm-107 -214	kmt-5.55	rpm-195 -390
kmt-3.07	rpm-108 -216	kmt-5.55	rpm-195 -390
kmt-3.07	rpm-108 -216	kmt-5.55	rpm-195 -390
kmt-3.07	rpm-108 -217	kmt-5.55	rpm-195 -390
kmt-3.07	rpm-108 -216	kmt-5.55	rpm-195 -390
kmt-3.07	rpm-108 -216	kmt-5.58	rpm-196 -392
kmt-3.07	rpm-108 -216	kmt-5.55	rpm-195 -390

(a) Avlesing av fart på 9 V batteri

(b) Avlesing av fart på 12 V batteri

5.2 Arm

5.2.1 Rotasjonsmotorer

Siden hjulene skal kunne operere separat fra hverandre er det tatt valget å bruke motor på alle hjulene. Dette krever da at vi har signifikant med moment for at hjulene skal klare å svinge. Da det ikke var lagt opp for ekstern giring så ble det nødvendig å lete etter en motor med innebygd giring som hadde nok moment til å vri hjulet, men samtidig ikke ble for stor for designet. Det nødvendige momentet ble funnet ved hjelp av [28] der formlene er lagt opp for å finne kraften

nødvendig med å ta integralet over kontaktflaten til dekket.

$$T_{gkfsz} = \int_{x_0}^{x_1} \int_{y_0}^{y_1} |r_{kp}| \mu p(x, y) dx dy \quad (8)$$

Ligning (8) er bygget opp av flere funksjoner som da er

$$|r_{kp}| = \sqrt{(e_x - x)^2 + (e_y - y)^2} \quad (9)$$

som er lengden på armen av kraften. Denne kan forenkles i vårt tilfelle da e_x og e_y er variabler tilhørende hvis vi hadde hatt kingpin akse som er vanlig på biler, men da vår styringsmekanisme er direkte over hjulet kan vi se på disse som 0, og får da

$$|r_{kp}| = \sqrt{x^2 + y^2}.$$

Samtidig er den bygget opp av $p(x, y)$ som fremvises som

$$p(x, y) = \frac{6F_z}{a^3b} \left(\frac{a}{2} - x\right) \left(\frac{a}{2} + x\right), \quad (10)$$

der $p(x, y)$ er kraften på kontaktflaten mellom dekk og bakken. F_z er kraften og a og b er brukt for å finne arealet til kontaktflaten. F_z er kraften som er på dekket

$$F_z = 5kg * 9,81m/s^2 \approx 50N.$$

For simplisitet sin skyld så sier vi at kontaktflaten er kvadratisk, derfor antar vi at a og b har lengdene 0,05m eller 5 cm etter dekkene blir komprimert når de er utsatt for tiltenkt vekt, for fullstendig utregning av a og b , se [28].

Da det er vanskelig å bestemme nøyaktig friksjonskoeffisient siden vi ikke har talla på materiale den skal bli brukt på, setter vi den til å være 0,7 da dette er en forholdsvis vanlig/høy koeffisient.

Når vi regner ut dette med alle innsatte variabler og setter grensene for integralet får vi:

$$T_{gkfs} = \int_{-a/2}^{a/2} \int_{-b/2}^{b/2} \mu \frac{6F_z}{a^3b} \left(\frac{a}{2} - x\right) \left(\frac{a}{2} + x\right) \cdot \sqrt{x^2 + y^2} dx dy = 0,5927 Nm$$

En annen utregning funnet fra [29], viste til en annen framgangsmåte av å finne nødvendig omdreining. Det viste til

$$T_s = W f \sqrt{\frac{I_0}{A} + e^2} \quad (11)$$

der W er vekten av SoilRover i Newton, f er friksjons koeffisient, I_0 er treghetsmoment, A er arealet av kontaktflata og e er en samling av verdiene e_x og e_y som er i [8], og vi kan se bort ifra denne da vi ikke har noen kingpin akse som tidligere nevnt.

Forenkling av [11] ga oss

$$T_s = W f \sqrt{\frac{R^2}{2}}, \quad (12)$$

der R er halvparten av bredden til kontaktflaten som for oss blir 2,5cm eller 0,025 m.

Når vi setter disse verdiene inn i [12] får vi

$$T_s = (5kg \cdot 9,81 \frac{m}{s^2}) \cdot 0,7 \sqrt{\frac{0,025^2}{2}} = 0,6187 Nm,$$

som er forholdsvis nærme til resultatet til svaret vi får fra [8].

Dette er momentet som teoretisk er nødvendig for at vi skal kunne rotere dekkene når vi står parkert. Når vi sammenligner verdiene vi får fra (8) og (12) ligger nødvendig moment på rundt 0,6 Nm.

Når vi bestilte motor så var dette før denne utregning var utført, og bestilling ble gjort med tanke på størrelse, behov for innebygd enkoder og nøyaktighet. Moment ble antatt at ikke var såpass høy ved rotasjon. Første utregning kan bli funnet i vedlegg [Gamle torque utregning for rotasjon](#), feilen ble gjort ved innsatt desimalfeil. Når utføring av dobbeltsjekk og vi fikk de nye resultatene, var det for sent til å gjøre endring på design eller bestille ny motor.

Men selv om det ikke går å rotere når den står stille med vekt ned på ekskluderer ikke det muligheten for å styre når vi beveger oss eller har dekkene hevet og for mer nøyaktig kontrollering ønsker vi å bruke PID-kontroller. Dette er noe som kan implementeres i Arduinokoden for hver type motor, og enten er for å få den til å reagere raskere, saktere eller stabilisere sluttsignalet. Det var et problem da den oppgitte informasjonen til motorene ikke inneholdt de verdiene vi trenger for å finne transferfunksjonen til motoren og gjorde det vanskelig for å prøve å lage en simulasjon. Dette resulterer i at vi heller blir nødt til å teste oss fram.

5.2.2 Enkoder for rotasjon

Da vi skal måle hvilken posisjon hvert dekk er i så er det behov for en sensor som vi kan bruke for å måle vinkelen til dekkene. Ett av kravene våre K.05 i tabell [25](#), var at rotasjonen skulle kunne treffe på ± 1 grad°. For at vi skal få til dette må vi ha vertfall 360 PPR på enkoderen. For å finne sensitivitet enkoderen har, bruker vi

$$\frac{360^\circ}{1^\circ} = 360PPR \quad (13)$$

Siden design gjør at vi har enkoderen på samme akse som motoraksen, ble valget av enkoder til å være ferdig montert i motor. Første motoren vi så på kan du lese om i Vedlegg A under

Styring, men det var et problem med denne da den bare hadde 1 kanal som da kun lar oss lese av fart nøyaktig.

Motoren vi endte med å bestille var TS-25GA370H fra TsinyMotors gjennom Amazon, verdiene oppført for motor er i tabell **(16)**.

Tabell 16: TS-25GA370H

TS-25GA370H	
Spenning	12V
Strøm	0,15A
Moment	0,11Nm
RPM N.L	130
Gir rate	1/45
Encoder pulser	12PPR

Enkoderen er som vist i tabell **(16)** bare 12 PPR, men den er koblet på selve motoren. Motoren har igjen en 1:45 gir rate som da gir oss enkoder sensitiviteten for staget til å bli

$$12PPR \cdot 45 = 540PPR.$$

Hvis vi vrir om på **(13)** så blir det da en oppløsning på

$$\frac{360^\circ}{540PPR} = 0,666^\circ.$$

Enkoderen oppfylder da kravene vi hadde for sensitivitet samtidig som den har 2 kanaler som da også lar oss lese av posisjon.



(a) Enkoder måler av motoren i motsatt klokkeretning
(b) Enkoder måler av motoren i klokkeretning.

Figur 43: Venstre bilde viser enkoder signalene når motor går mot klokka, og høyre figur viser motoren går med klokka.

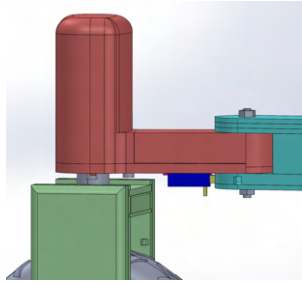
Årsaken til at vi trengte en 2-kanals enkoder kan bli sett i fig. 43a og fig. 43b. Der kan vi observere at pulsene ligger 90° for hverandre. Dette betyr at vi kan fastslå både retning basert på hvilken kanal som er Rising” først.

Dette fremsto som en motor som egnet seg for vårt bruk basert på tidligere antagelser. Det var noe problemer med at selskapet vi fikk motor fra ikke kunne gi oss mer detaljerte datablad enn informasjon som var oppført på kjøpside og kontrollering av denne ble da nødt til å bli mer testet fram enn regnet ut. Siden tanken var å bruke PID-kontroller for å styre motoren slik vi ønsket ble vi da nødt til å teste fra P,I og D verdien. Det er tatt i bruk ferdig bibliotek for PID-kontroller i Arduino for testing.

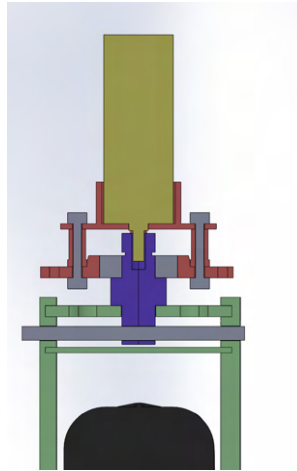
Vi har mer behov for nøyaktigheten enn hvor raskt det blir utført og har da det i tankene når vi velger verdier.

5.2.3 Mekanisk design av Arm

Rotorasjonsmotorakselen er festet direkte til akselen. Dette er mulig siden motoren har innebygd gir. Motoren er festet til armen slik at den står stille i forhold til den.



(a) 3D-modell av armmodul

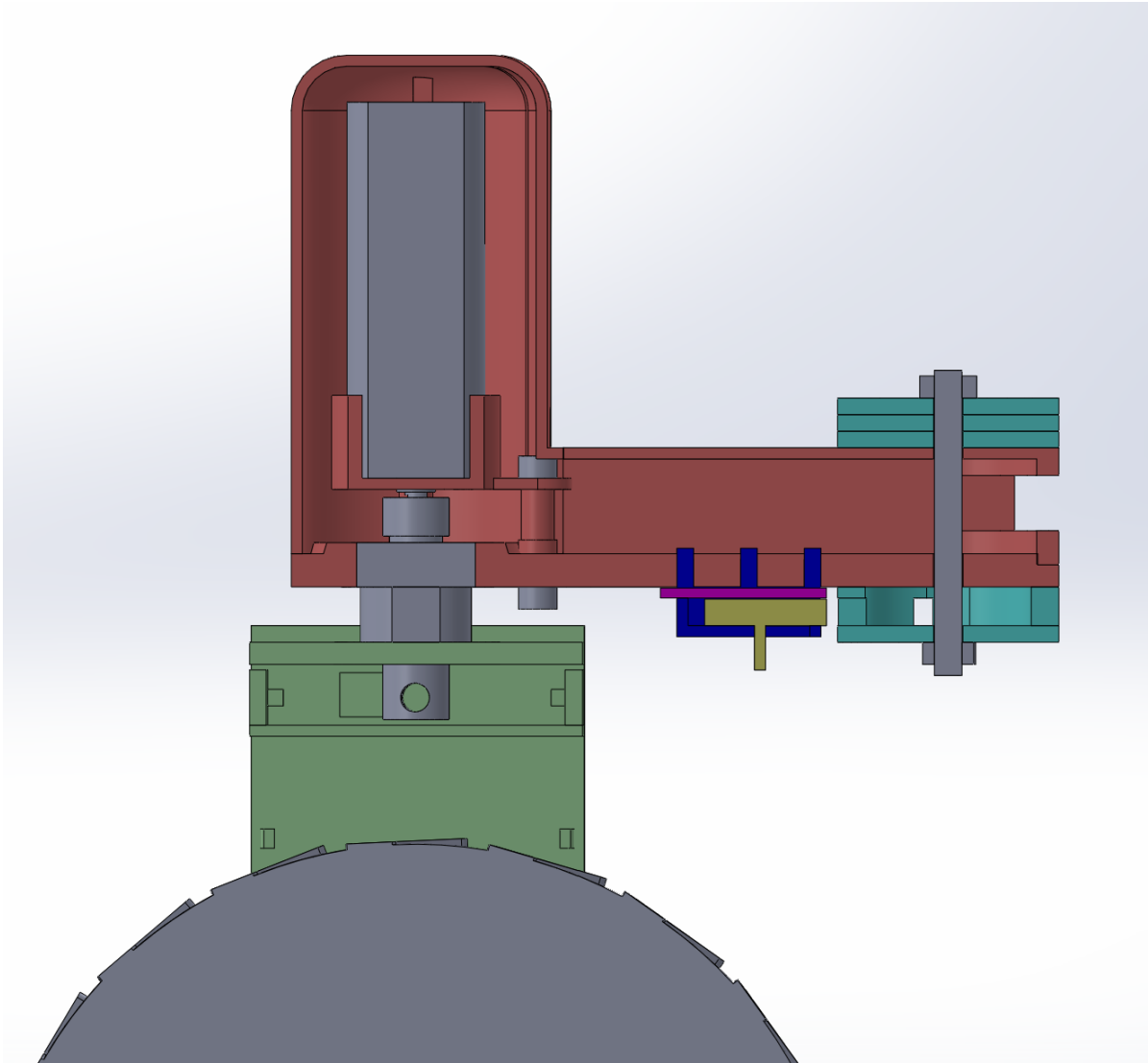


(b) Seksjonsvisning av rotasjonledd

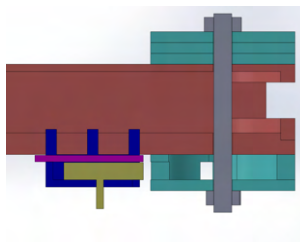
Figur 44: Rotasjonsledd

Kulelageret er festet til armen ved hjelp av tightfit. Akselen går gjennom kulelageret og festes ved hjelp av en låsering. Kulelageret gjør at friksjonen mellom akselen og armen er så liten at den kan ses bortfra under kalkulasjoner. Det er lagt opp spor til låsering på akselen for å stanse hjulmodulen fra å falle av armen når SoilRover bæres. Når SoilRover står på bakken er det den gjengede stangen som bærer vekten. Akselen er den viktigste delen for å få et stødig ledd. Spesielt viktig er det å ha god tightfit mellom kulelageret og toppflaten på rammeverket i hjulmodulen. Disse delene ble limt for å øke stabiliteten.

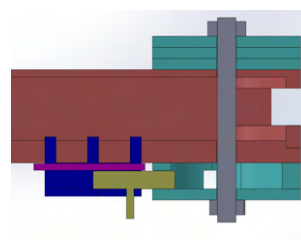
3D-print er sterkest når kreftene står normalt på printlagene, derfor har vi valgt å printe akselen i to deler og så lime den sammen. Vi valgte bort å maskinere den i stål, for det gir ikke mulighet til D formet hull.



Figur 45: Seksjonsvisning av låsemekanisme



(a) Åpen



(b) Lukket

Figur 46: Seksjonsbilde av låsemekanisme og skulder

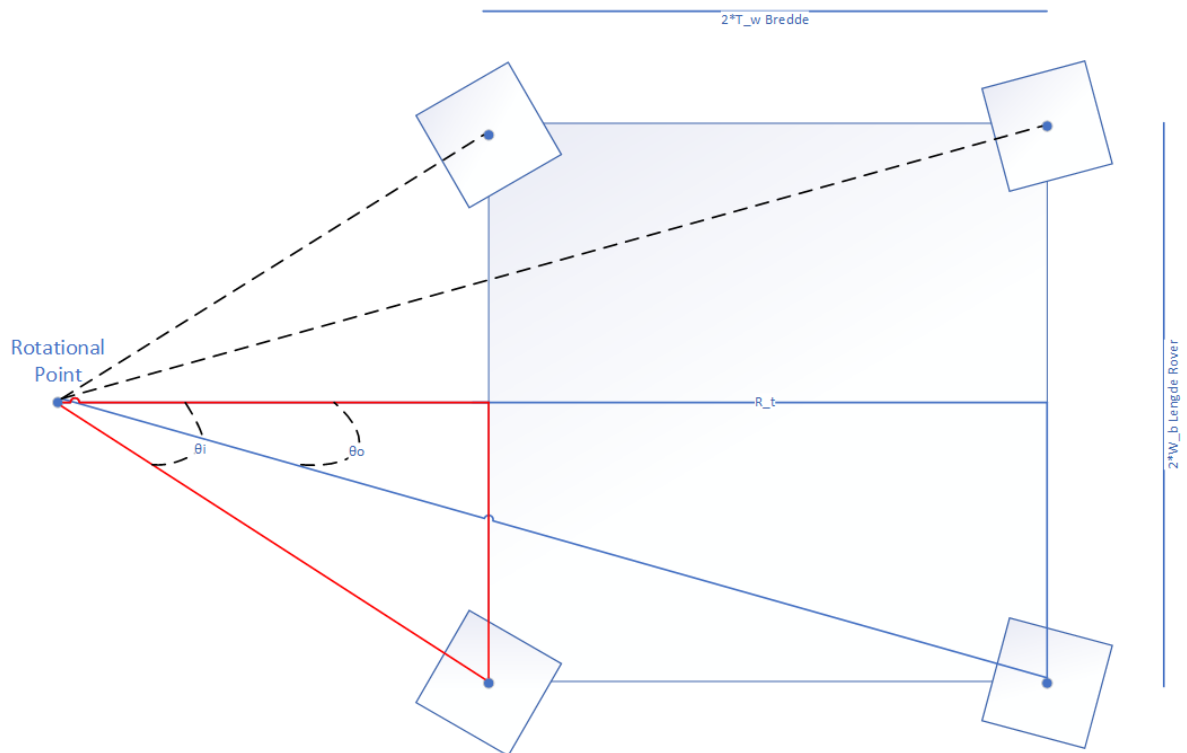
Armen festes til skulderen ved hjelp av en gjenget stang og to muttere. Armmodulen kan låses i to forskjellige posisjoner som ligger 75 grader fra hverandre. Stangen (se gult på fig. 45) holdes i stillingen som du kan se på fig. 46b ved hjelp av en fjær. For å bytte stilling er det bare å dra i spaken og flytte armen.

Låsemekanismen blir 3D-printet. Skulderen blir laget av både kryssfiner og 3D-print, og en gjenget stang med to muttere.

5.2.4 Ackermann

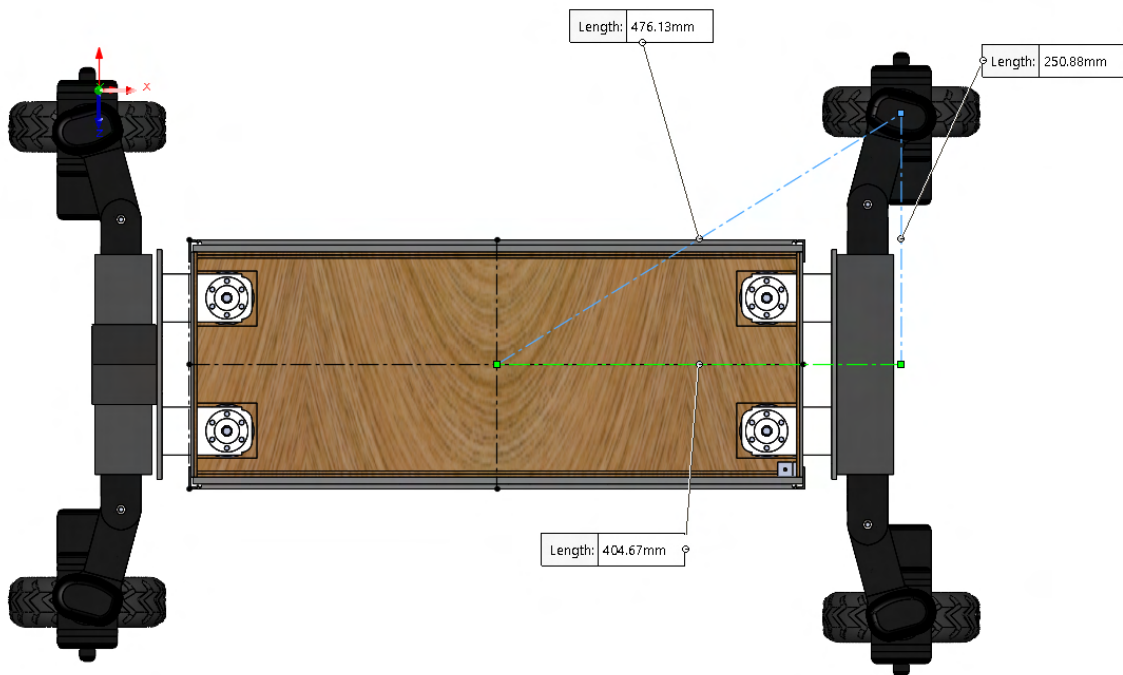
SSI bruker Ackermannstyring så det er naturlig at vi gjøre det samme, da oppgaven vår er å etterligne SSIs produkt.

Ackermann er et styringskonsept der en setter vinklene til hjulene slik at fartsretning er ortogonale på et punkt. Dette er et konsept som ofte er brukt i biler med 2-hjuls styring, men nå er det nødt til å kordinere seg etter 4 hjul. Her blir utfordringen å få fire motorer til å kordinere med hverandre for å rette seg mot et punkt samtidig som hjulene er nødt til å ha forskjellige hastigheter basert på vinkelen for å unngå skjeining. [30] En annen fordel med 4-hjuls Ackermann er at ved styring/svinging så er det fremt at styrer du rundt et hjørne der framparten kommer forbi, vil da også bakenden komme forbi på grunn av de bakre hjulene følger samme spor som framhjulene hvis du ikke justerer svinging. [31]



Figur 47: Figur som viser til konseptet av 4-hjuls Ackermann.

Fig. 47 viser kun til konseptet av 4-hjuls Ackermann, vi skal også ha plassert armer utifra kroppen så fig. 47 gjenspeiler ikke detaljene i vårt prosjekt, men heller variablene som er nødvendig for utregning. Hvis referansepunktene vi setter er langs roveren sin sentrum så vil vinklene til de indre hjulene snu seg den gitte vinkelen motsatt av hverandre. Det er det samme med de ytre hjula.



Figur 48: Siste design med lengder for Ackermann

For at vi skal kunne rotere på plass er alle hjulene nødt til å stå mot midten med fartsretning ortogonal på dette. Fra tegninger anskaffet fra Solidworks finner vi vinkelen som er nødvendig med standard trigonometri

$$\theta = \arccos\left(\frac{\text{hosliggende}}{\text{hypotenusus}}\right). \quad (14)$$

Ved å bruke siste design så får vi

$$\theta = \arccos\left(\frac{250,88\text{mm}}{476,13\text{mm}}\right) = 58.20275571^\circ.$$

Dette blir vinkelen som hjulene er nødt til å bli plassert i for at siste design skal rotere om seg

selv.

Ligninger til Ackermann

Ligningene som er nevnt er i forhold til en venstre sving. Vinkelen til dekket er funnet med å bruke trigonometri.

Vinkelen til de indre dekket er funnet ved å regne ut den røde trekanten i fig. 47 ved å ta \cos^{-1} av hosliggende katet/ hypotenus.

$$\theta_i = \arccos\left(\frac{R_t - T_w}{\sqrt{W_b^2 + (R_t - T_w)^2}}\right) \quad (15)$$

Vinkelen til de ytre dekket er funnet ved å regne ut den blå trekanten i fig. 47 ved å ta \cos^{-1} av hosliggende katet/ hypotenus.

$$\theta_o = \arccos\left(\frac{R_t + T_w}{\sqrt{W_b^2 + (R_t + T_w)^2}}\right) \quad (16)$$

Vinkelen til det de ytre hjulene finner vi fra (16) og da blir det bakre og fremre hjulene justert til samme vinkel, men motsatt retning av hverandre.

Både (15) og (16) er funnet fra [31].

Slik vi har tenkt oppsettet nå så er startposisjon stående i 180° å rotere mellom 0° og 360° . Så da er det at ene hjulet blir θ_o og det andre hjulet som følger samme hjulspor blir da vinkel $-\theta_o$ så lenge SoilRover svinger om et punkt langs x-aksen fra sentrum. Blir punktet justert vekk fra x-aksen til SoilRoveren vil ikke dette være tilfellet og utregningene blir mer komplisert samtidig som vi er nødt til å ha noe til SR som klarer å forholde seg til det samme punktet. Slike utregninger er å finne i [29].

6 Grensesnitt

For å kunne ha kommunikasjonen i vårt system trenger vi flere grensesnitt. Vi har flere digitale og noen fysiske. Dette fører til bedre brukervennlighet og en enklere forståelse for hva som foregår. Da Arduinoen ikke har et operativsystem og simulasjonen kan kjøres uavhengig av operativsystemet, så blir det ikke noe fokus på Programmeringsgrensesnitt. Brukeren kan kommunisere med produktene våre via en PS4 kontroller, en applikasjon eller via mus og tastatur. Dette kalles for brukergrensesnitt. Dessuten så er det en fysisk sammenkobling mellom begge Arduinoene og en trådløs kommunikasjon mellom Arduino og applikasjon/PS4 kontroller. Begge disse går under kategorien grensesnitt i datasystemer.

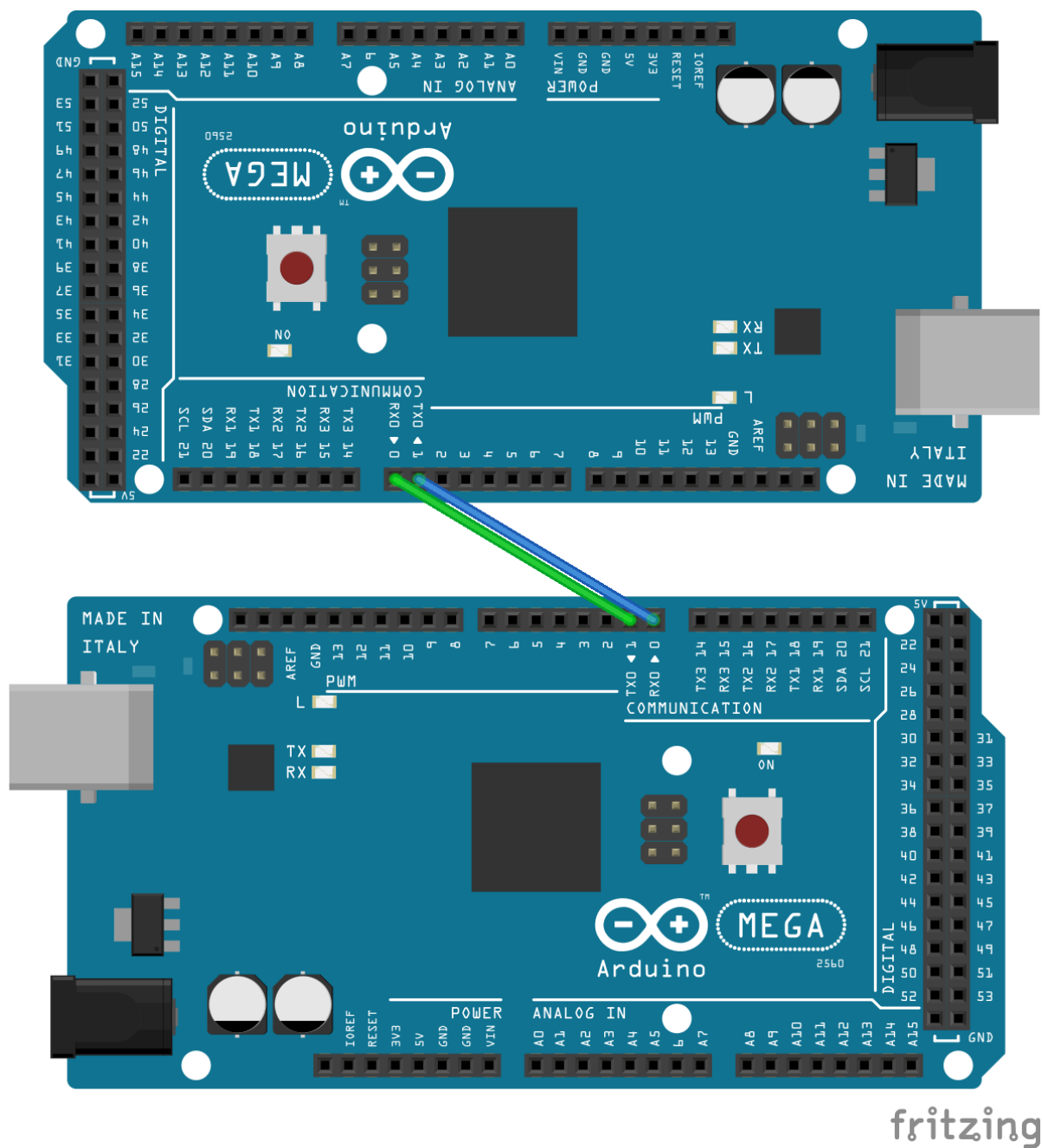
6.1 Grensesnitt i datasystemer

For at datasystemer skal kunne snakke med hverandre så må vi konstruere et grensesnitt mellom de. Dette kan være trådløs, som ofte er enklere å håndtere fordi man slipper ledninger men er tregere, eller fysisk med ledninger men raskere.

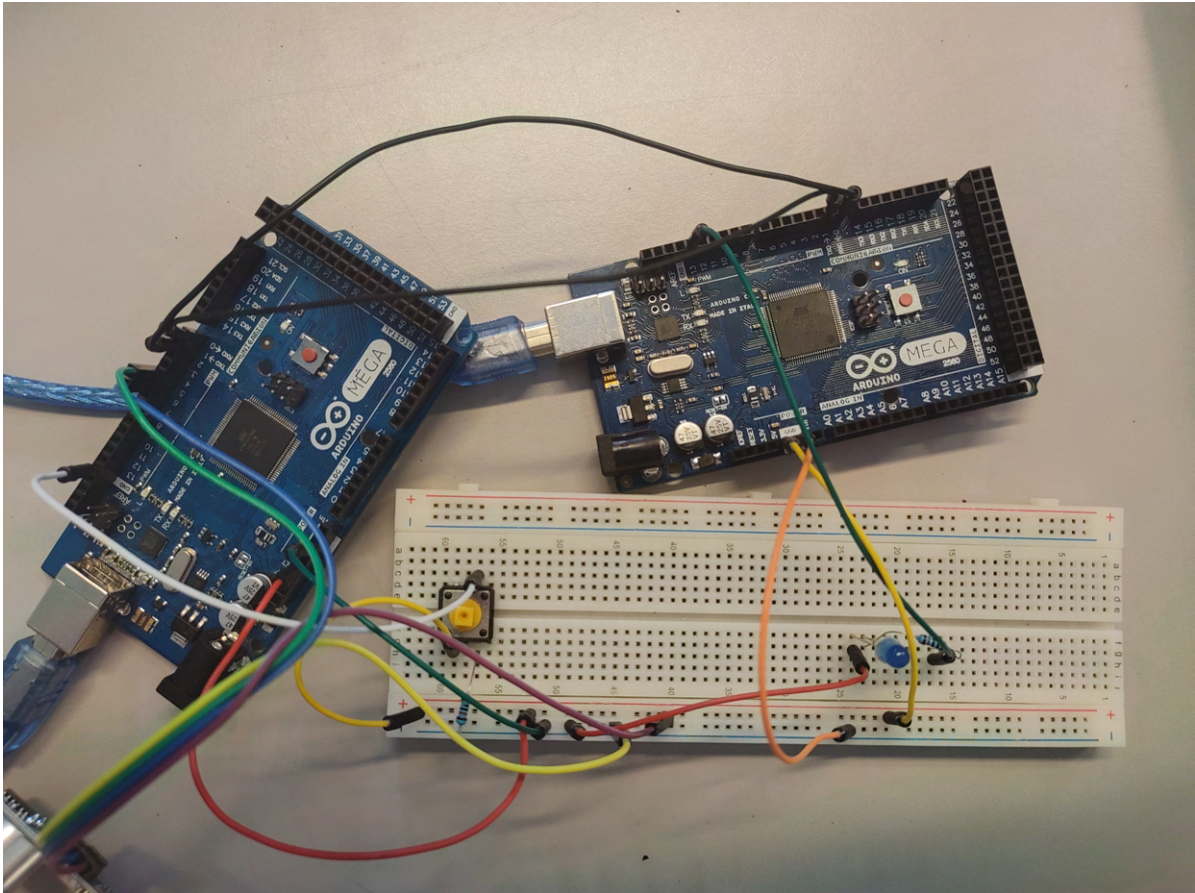
6.1.1 Fysisk sammenkobling

I utgangspunktet så prøvde vi å bare bruke en Arduino. Vi valgte Mega 2560 på grunn av antall tilgjengelig pins. Dessverre så viste det seg at det ikke var nok allikevel og vi prøvde å ha et samarbeid mellom to Arduino Mega 2560. Etter litt testing så kom vi fram til at dette ville fungere og delayet ville være minimalt. Hovedarduinoen, som også kalles for Master, vil ta seg av ting som må skje i sanntid, dette inkluderer trådløs kommunikasjon og selve styring av kjøretøyet. Funksjoner sånn som heisemekanisme blir ikke påvirket av et eventuelt delay noe som fører til at dette kan være på den andre Arduinoen, som kalles for Slave. Arduinoene er

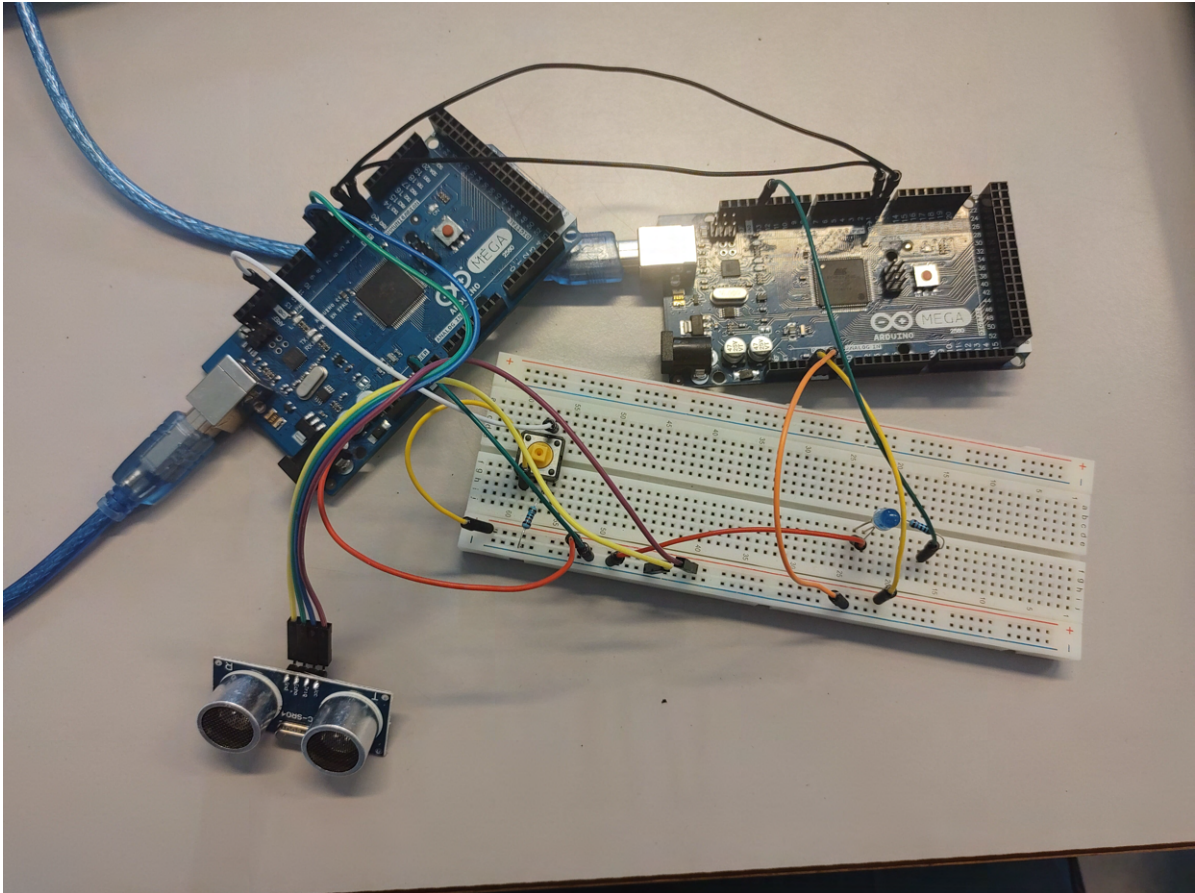
koblet sammen via 2 ledninger, der transmit (å sende) er koblet til receive (å motta) og samme tilbake. Begge mikrokontrollerne får overført sitt eget skript som da inneholder forskjellige instruksjoner basert på oppgavene som skal utføres.



Figur 49: Figur som viser sammenkobling mellom 2 Arduinoer



Figur 50: Bilde som viser hvordan to Arduino er koblet sammen

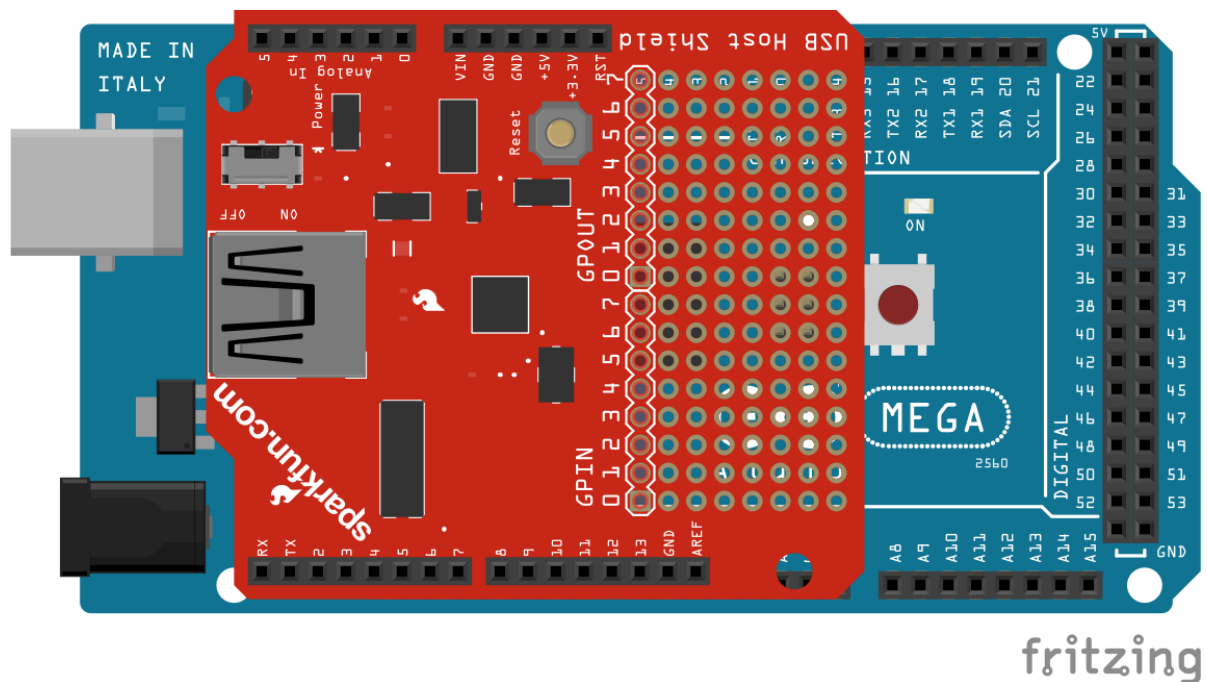


Figur 51: Bilde som viser testing av Ultralydsensor med to Arduino som er koblet sammen

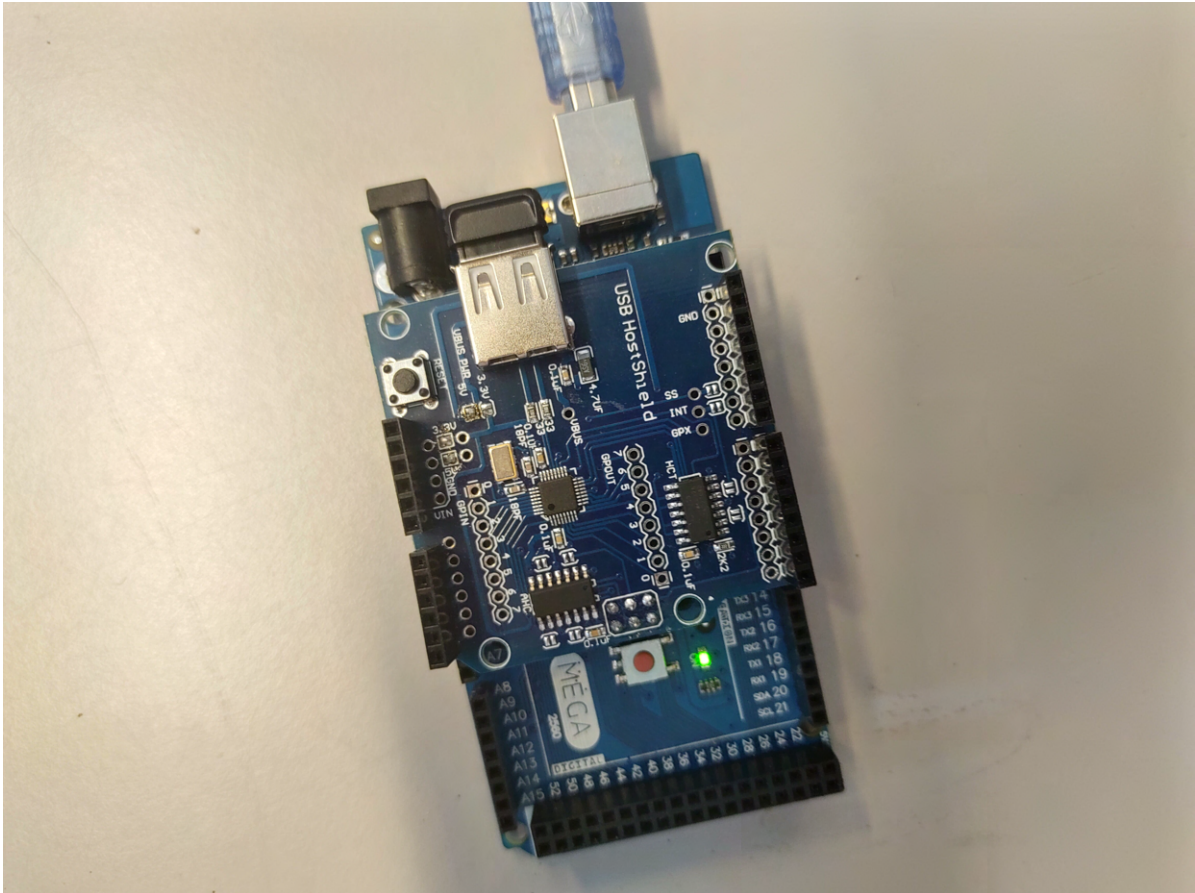
6.1.2 Trådløs kommunikasjon

Produktet vårt tar også i bruk trådløs kommunikasjon. En USB-Host shield med en USB-hunn, som er noe man setter opp på Arduinoen for å kunne sette inn en bluetooth dongle, som er koblet til en PS4 kontroll. Først så brukes det kodelinje 1 når man skal parre kontrolleren og donglen for første gang og så brukes kodelinje 2 istedenfor for å koble sammen to kjente enheter.

1. PS4BT PS4(&Btd , PAIR);
2. PS4BT PS4(&Btd);

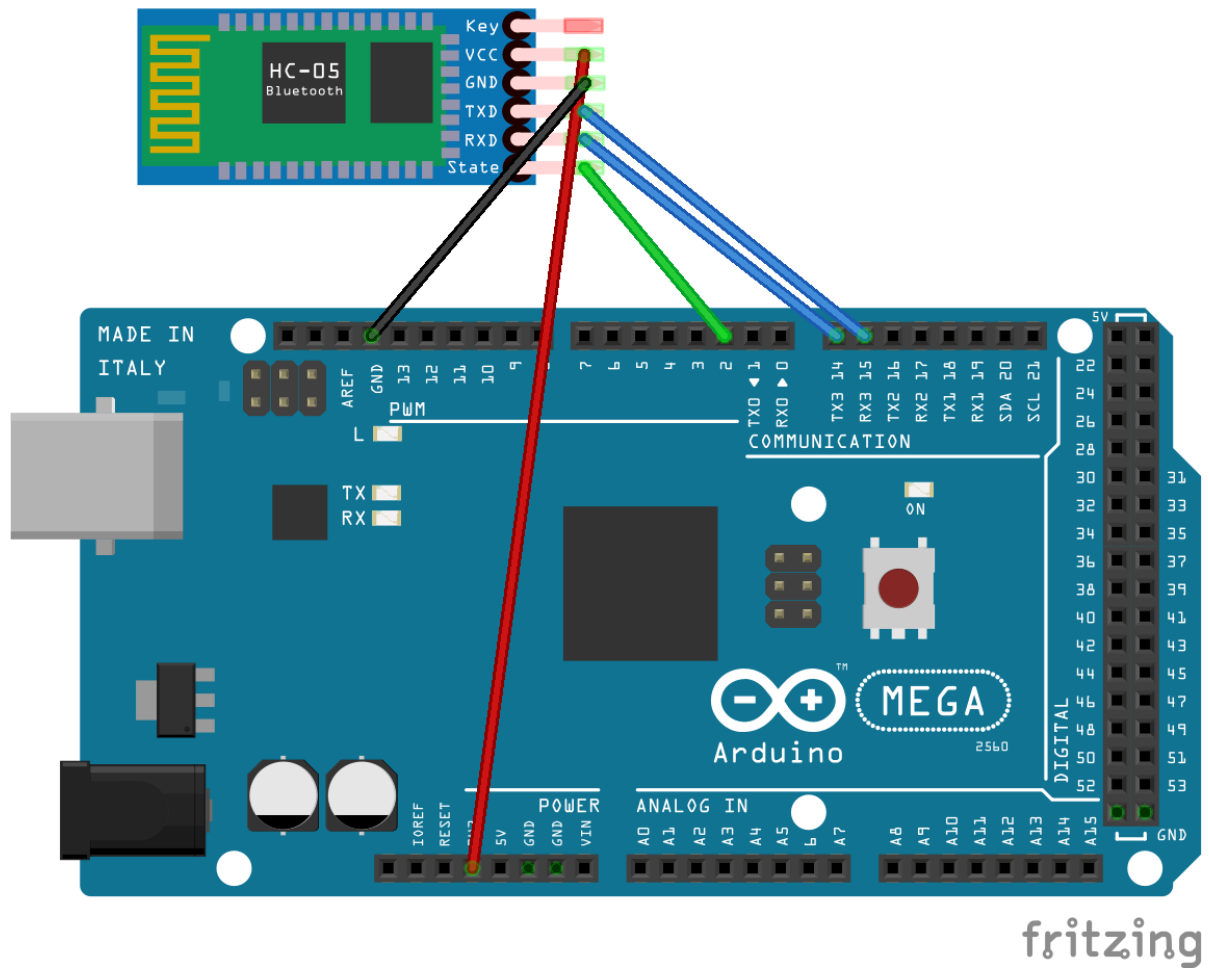


Figur 52: Figur som viser hvordan USB-Host shield er koblet til Arduino

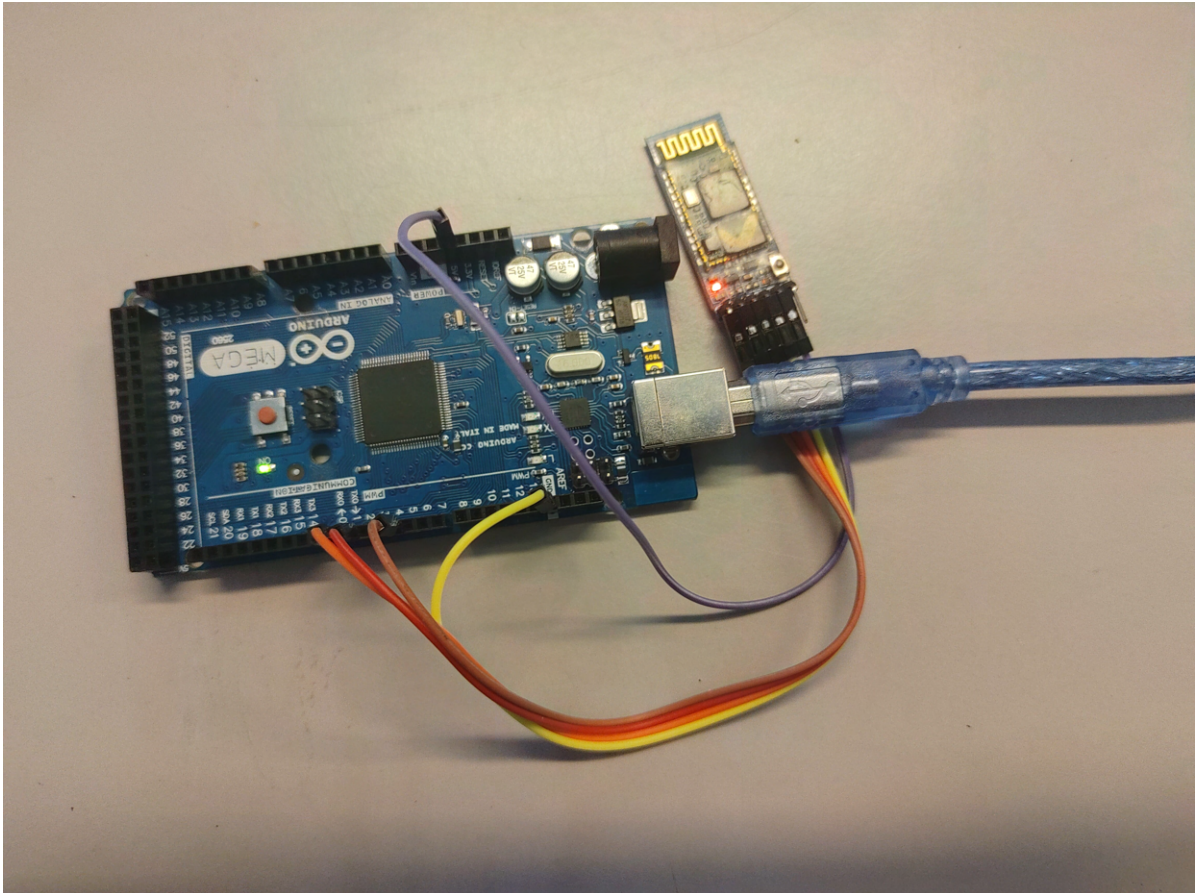


Figur 53: Figur som viser fysisk hvordan USB-Host shield er koblet til Arduino

Videre så finnes det en trådløs kommunikasjon mellom HC-05, som er en bluetooth-enhet og en Android-mobil telefon via appen vår. Appen leter etter enheten og så sender en forespørsel til enheten om sammenkobling, tillatelsen blir gitt hvis det brukes riktig passord. Vi har også koblet til STATE-pinnen slik at vi kan etterspørre om det er koblet til en telefon.



Figur 54: Figur som viser hvordan HC-05 er koblet til Arduino



Figur 55: Figur som viser fysisk hvordan HC-05 er koblet til Arduino

6.2 Brukergrensesnitt

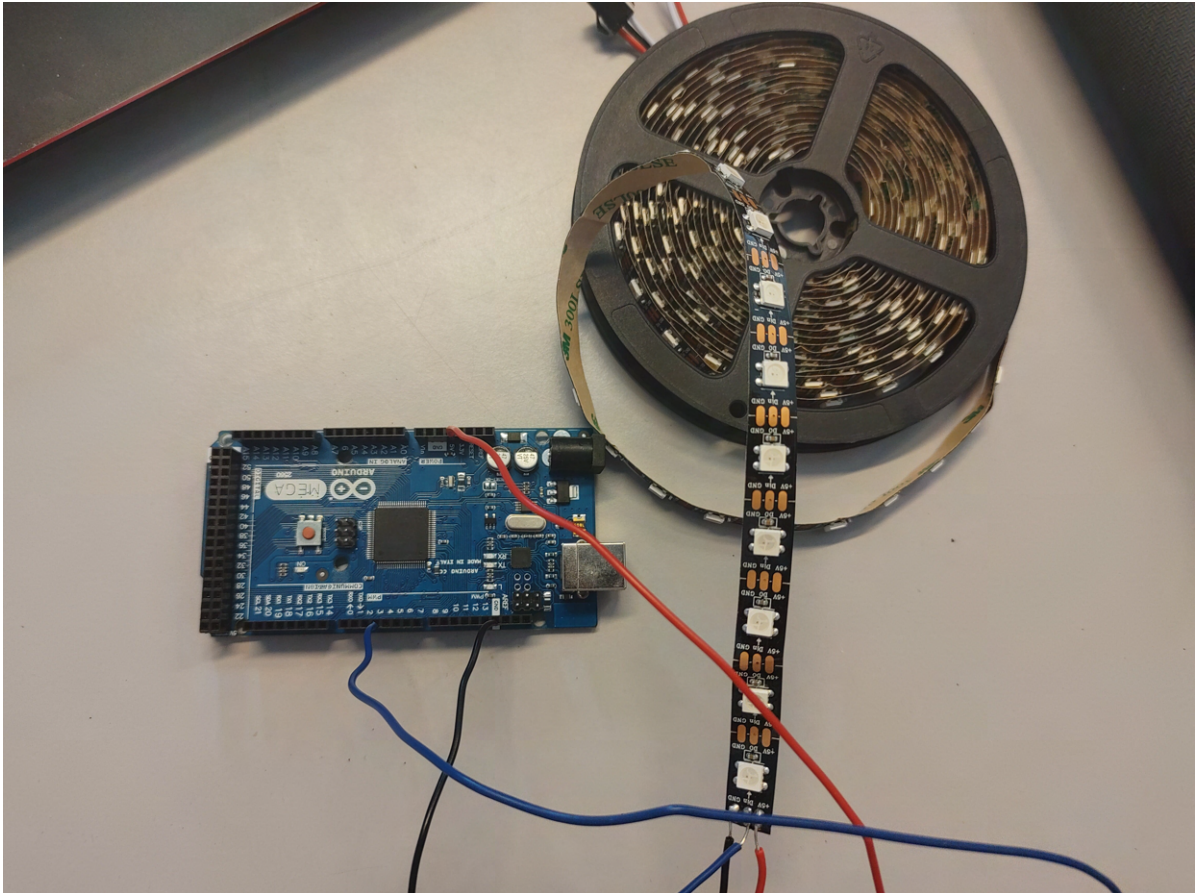
For at noen skal kunne bruke produktet vårt så trengs det et eller flere brukergrensesnitt. SoilRover kan styres både via applikasjon og kontroller. Dessuten så har vi flere grensesnitt for informasjonsformidling og enklere bruk.

6.2.1 LED

Vi ønsket å kunne gi raske tilbakemeldinger til brukeren som også er synlig ved litt avstand. Først vurderte vi en RGB-LED pære, men fant fort ut av at en LED-stripe under hele kjøretøyet ville være mer synlig.

Tabell 17: LED-stripe forklaring

Betydning	lys	varighet
SoilRover starter	fader inn Grønt	5 sek
Bluetooth venter på tilkobling	Blinker Blått	
Bluetooth er koblet til	Blått	5 sek
Feilmelding	Rødt	
For å indikere hvilken arm som har blitt valgt	RGB	
Alt er i orden	Grønt	



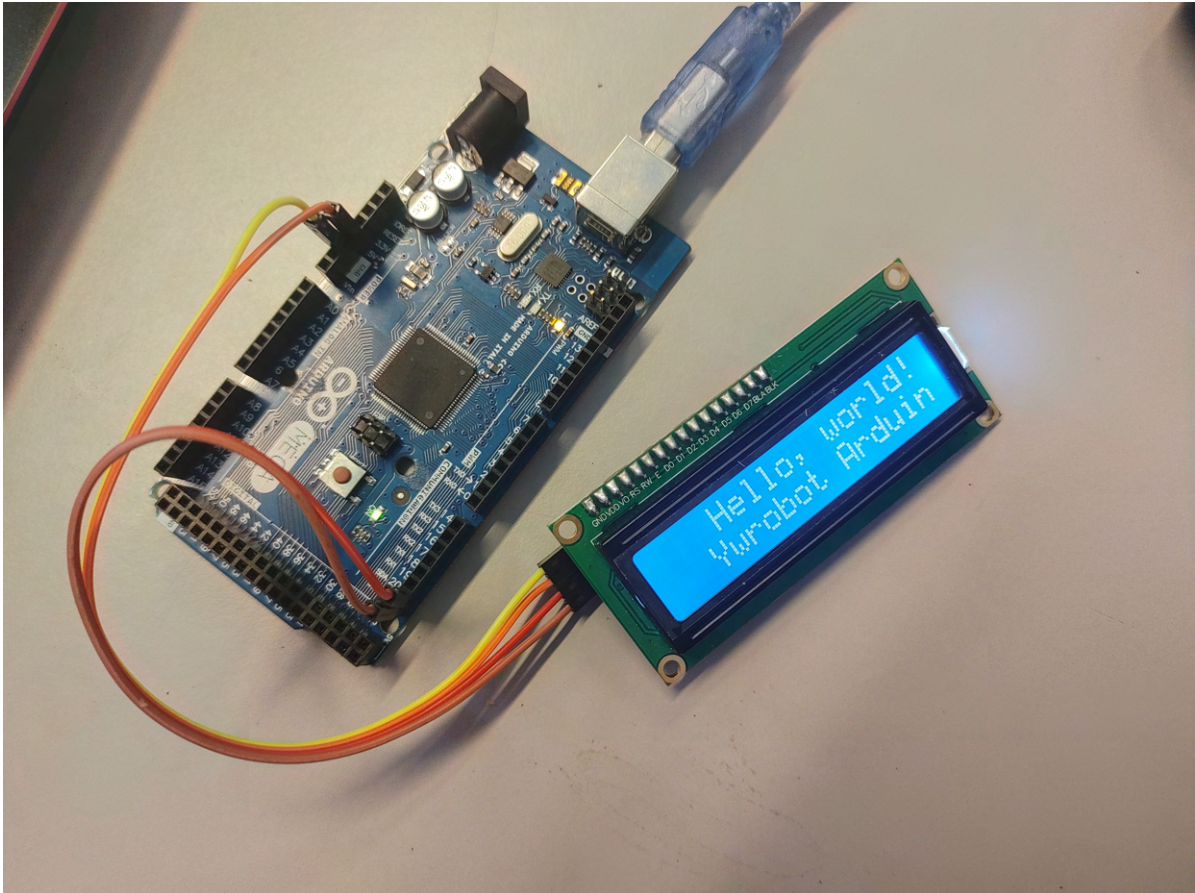
Figur 56: Figur som viser fysisk hvordan LED-stripa er koblet til Arduino

6.2.2 Display

Lys i seg selv vil ikke være nok som tilbakemelding for feilsøking og håndtering av kjøretøyet vårt. Derfor har vi også inkludert et display som kan vise forskjellige tilstander.

Tabell 18: Display-beskjeder

Tilstandsmelding	Betydning	Varighet
Styres av app	Appen og Arduino er koblet sammen	fram til annen beskjed
Styres av PS4	PS4 kontroller og Arduino er koblet sammen	fram til annen beskjed
Kan ikke heise mer opp	Brukeren prøvde å heise opp selv om den allerede er heiset opp	5 sek
Kan ikke heise mer ned	Brukeren prøvde å heise ned selv om den allerede er heiset ned	5 sek
Venter på tilkobling	Arduinoen er ikke koblet til noe Bluetooth enhet	fram til annen beskjed
STOOOPPPP	Ultralydsensorene har registrert en objekt som er i veien	til objektet er fjernet eller man kjører en annen retning



Figur 57: Figur som viser fysisk hvordan LCD-skjermen er koblet til Arduino

6.2.3 APP

Vi ønsket en app i dette prosjektet for mer presis styring og mer nøyaktig justering. Dessuten så er det veldig nyttig å kunne lese av informasjon i sanntid på avstand. Manualen er også lettere tilgjengelig. Basert på våre krav så kom vi fram til 3 funksjoner. At appen skal kunne styre SoilRover, motta informasjon fra SoilRover og vise en manual.

6.2.4 Kontroller

Når vi skulle velge kontroller for styring så fokuserte vi på XBOX og PS4 kontroller da disse er de mest vanlige. Vi vurderte om vi skulle lage vår egen kontroller, men fant fort ut av at det vil være enklere å bruke en kjent. Hvis vi hadde hatt mer tid, så er det noe man kunne ha sett på.

Vi gjorde research og utførte tester der vi sammenlignet begge kontrollerne men konkluderte med at det er det samme uansett hvilken vi bruker. Konklusjonen ble at vi bruker PS4 kontroller da de fleste er mer vant til den.

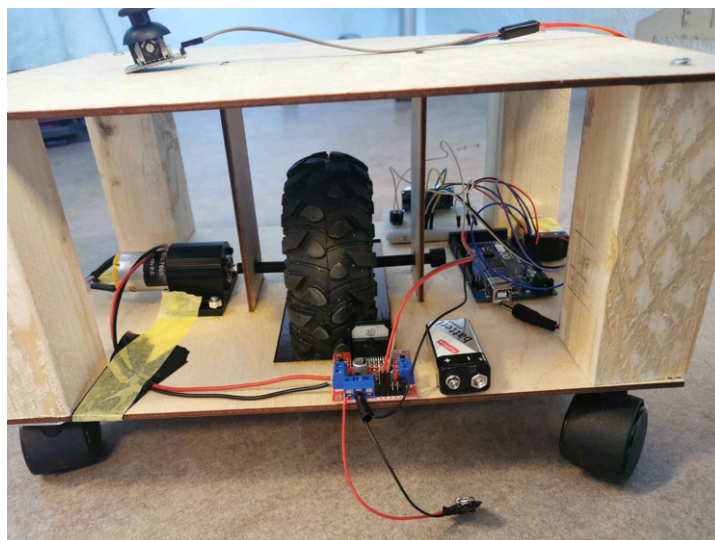


Figur 58: Figur viser oppsettet av knappene på PS4-kontrolleren

7 Prototyper

7.1 Prototype Driv

Første prototypen vi lagde var for test av driv motoren. Denne ble designet og bygd på en uke før andre presentasjon, slik at vi hadde et proof of concept vi kunne vise frem. Denne prototypen ga oss test muligheter for flere applikasjoner som skal implementeres i den endelige versjonen av SoilRover. Vi fikk koblet opp og testet driv motor, så sjekket om avstands sensorene HC-SR04 funket slik vi ville og hvor effektive de var, forså å koble styringen opp til en trådløs PS4 kontroller. Med denne lille prototypen fikk vi startet arbeidet med å feilsøke og teste kode i små steg, som gjør det enklere å feilsøke.

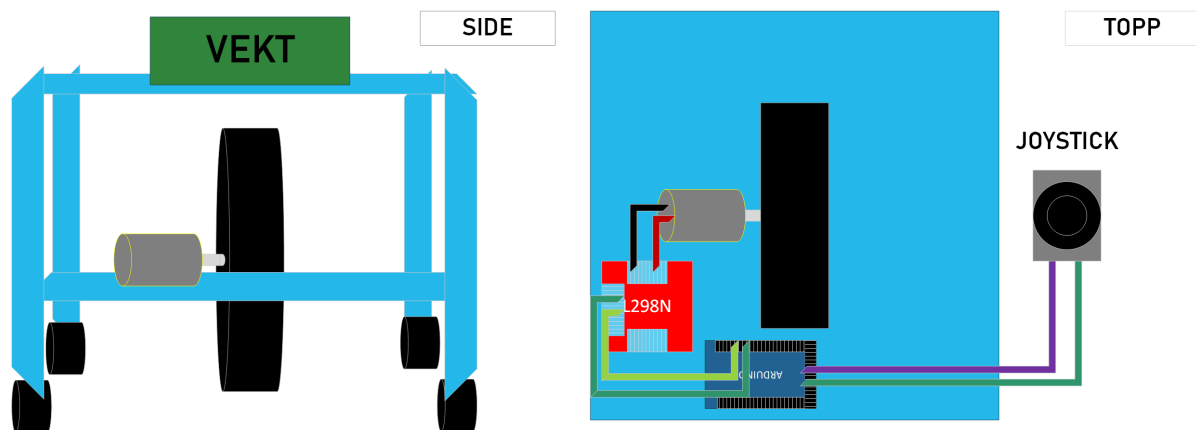


Figur 59: Ferdig lagd driv prototype

Prototypen ble laget av kryss fineplater og kontor hjul som støttehjul med 3D printet aksler og støtter. Elektronikken innebærte bruk av en Arduino Mega 2560 R3 som kontrollerte en L298N motorkontroller og en HC-SR04 avstandssensor koblet til en buzzer. Vi fikk lagt til et USB-host shield på arduinoen slik at vi kunne koble til PS4 kontrolleren via bluetooth.

7.1.1 Design og oppsett prototype driv

Vi startet med å lage konsept av designet vi ønsket. Tanken var å ha et hjul styrt av en motor, som skulle få en vekt lagt på seg slik at vi kunne teste styrken til motoren. Det ble beregnet at hver motor skulle tåle iallefall 5 kg med last. Design visualisert i visio blir vist i Fig. (60).



Figur 60: Figur som viser konseptet til prototype 1

Basert på ideen ble designet overført til kryssfine plater og skrudd sammen. Det var først etter den ble satt sammen og test kjørt at vi så problemene med designet og bruk av harde kontorhjul på samme nivå som hovedhjulet. På fig. 61



Figur 61: Lite overflate mellom hjul og gulv.

7.1.2 Feilsøking og forbedringspotensiale

Hvis vi skulle forbedret designet til neste gang, måtte vi tenkt mer på å overføre friksjonen og lasten over på hovedhjulet. Med det nåværende designet fikk vi ikke testet kraft potensialet til motor og hjulet, og på grunn av at kontor hjulene tok mesteparten av vekten, ble det også de som styrte hvor kjøretøyet skulle kjøre.

7.2 Prototype sensor boks

Vi ønsket å lage en prototype boks slik at vi fikk testet oppkobling og koden mens vi ventet på det endelige produktet. Boksen inneholder Arduino Mega 2560 R3, USB-shield, 4 driv motorer, 2 HC-SR04 ultralydsensorer, lcd skjerm og en lysslynge. Poenget var å få disse delene til å samarbeidene og eliminere eventuelle problemer i god tid. Da vi ikke hadde noen hjulmoduler, så kjører boksen bare fram og tilbake.

7.2.1 Design av prototype sensor boks

Startet med å vite hvilke krav som trengtes for bruken av test riggen. Kravene som trengtes var at størrelsen skulle være så lik som slutt resultatet, plass for motorer for at boksen kan drives fremover, og plass for sensorene. Med dette så ble det 3D modulert ut ifra disse kriteriene. Laget en enkel boks som skulle bare kunne limes sammen og åpninger for de forskjellige delene. Så ble laget hull for sensorene og festene for sensorene, hull for motorfeste og for hvor motor akslingen går. Dette gjorde det enkelt å kunne teste at sensorene funker i drift. Festene for sensorene blir 3D printet og skrudd fast i rammen. Motorfestene skrues bare ned i rammen de også. De blir også 3D printet. Disse design-ene er laget så det er enkelt å endre på dem om størrelsene trenger å endres på eller om de må modifiseres for andre festemuligheter.

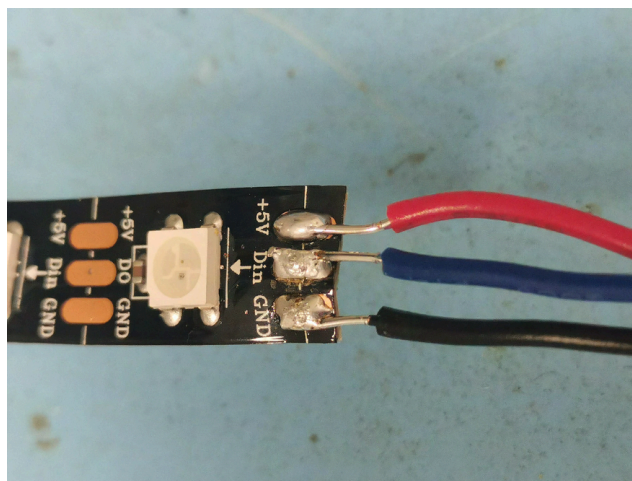
Når designet virket greit så laserkuttet vi kryssfiner plater og limte boksen sammen. Sensor festene endret seg litt på design ut fra hva vi trengte og hvordan de stod. Forsiden endret seg ikke ut fra første versjon. Men baksiden endret seg til å få det vi kaller vinger som skal være for å skru feste til rammen.

De fire driv motorene ble skrudd fast til motorfestene og ned i boksen slik at akselen stikker ut av hullet på sideveggen. To og to av motorene ble koblet til L298N motorkontrollere som videre kobles til arduinoen. Etterhvert som ting ble testet med koble, la vi på flere komponenter som avstandssensorene og lysslangen.

Vi startet å kjøre alt dette med hjelp av to 9V batterier. En koblet til arduino og en til koblet til begge motorkontrollere i parallell, slik at spenningen holdt seg lik. Senere fikk vi tak i et 12V batteri som vi fortsatte å bruke. Vi fikk og koblet opp og testet av/på knappen.

7.2.2 Kode og test av det vi vil den skal gjøre

Vi begynte med å skrive koden til en og en del. Når prototyp-boksen var ferdig bygd, så begynte vi med å få kontroll på alle 4 motorer. Deretter la vi til USB-shield slik at de kunne styres ved hjelp av PS4-kontrolleren. Der hadde vi noen problemer som blir beskrevet litt senere i rapporten. Når det var på plass, fikk vi den korresponderende ultralydsensoren til å stoppe motorene da noe var nærmere enn 40 cm. Samtidig koblet vi opp en lcd-skjerm som viser avstanden. Siste delen som ble koblet opp var led-stripa. Her klarte en av oss å koble den rett på 12V, noe som førte til at en bit ble ødelagt. Vi endte opp med å klippe av den ødelagte delen og loddet på nye kabler.



Figur 62: LED-stripen med nye kabler

Det ble skrevet kode til hver enkel del. På den måten fikk vi bli kjent med hver del og kunne være sikkert på at delene og tilhørende kode fungerer.

Ved å sette sammen koden bit for bit, unngikk vi en god del problemer som kunne ha oppstått ved å skrive koden til hele boksen med en gang.

7.2.3 Feilsøking og forbedringspotensiale

Det kom opp en del problemer med testing og styring av prototypeboksen. Et av de mer mekaniske problemene var at hjulene var så ustabil når vi fikk festet de på den 3D printet akselen til motoren. Vi løste dette ved å endre på design av akselen, samt skru fast motoren ordentlig til motorfestet. I tidlig fase var et av de større elektriske problemene med bakerste høyre motor, eller utgang 3 og 4 fra motorkontroller 2. Den var alltid svakere enn de tre andre. I feilsøking fikk vi målt spenning over de andre, som kjørte på ca 9 - 11 V. Mens den siste lå på 7 V en veg og 1.8 V andre vegen. Problemet var likt uansett hva vi byttet på. Motorer, kontrollere, Arduino, ledninger, etc. Dette problemet var der med bruk av 9V batteri. Tanken var at det gikk for lite spenning eller strøm, men problemet fortsatte etter vi byttet til et 12V batteri.

Første tanke var om vi ikke hadde klart å koble parallell. Da hadde den siste motorkontrolleren og siste motoren fått spenning sist og hadde gitt mening at den fikk mindre. Problemet var at motorkontrolleren er parallell så begge motorene til motorkontroller 2 burde være lave, og det burde være samme spenning hver veg også. Vi testet allikevel å koble opp batteriene og motorkontrollerne anderledes, samle spenning i ett punkt, for så å koble videre ut til to nye punkter for motor kontrollerne.

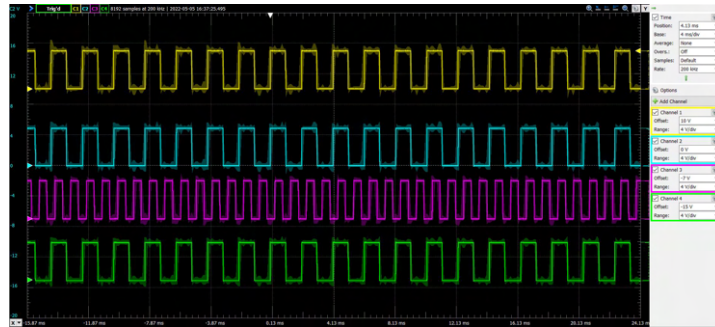
Vi fikk senere noen problemer med at motor 2 ikke kjørte i det heletatt ene vegen, men det fikset seg når vi koblet spenningen anderledes. Vi fikk en samtale med faglærer (Dag Samuelsen). Han foreslo å koble om på arduino, siden det var det eneste vi ikke hadde gjort, og vi hadde koblet alle motor pins på PWM pins som ikke er nødvendig. Bare en fra hver motor trenger å være pwm pin. Når vi flyttet ikke-pwm pins til vanlig analog pins på arduino for første gang ble problemene enda værre enn før. Flere motorer som rotet og stoppet. Vi koblet alle til analog pins 22-29, de første 8 pins på dobbelt rekka. Pwm pins var de første som ble flyttet på etter det. Det var lite forskjell i oppførsel for hver gang de ble flyttet. Så vi flyttet alle 8 analog pins til nederste av dobbelt rad pinsene på arduino 52 og oppover. Da fikk vi nye problemer igjen. Flere motorer oppførte seg anderledes. Vi flyttet så bare 2 av de 8 ikke-pwm pins, som tilhørte motor 4. Da oppførte systemet seg nesten perfekt. Flyttet da videre alle ikke-pwm pins til å ligge annenhver og ikke i en klynge. Vi brukte pin 22, 23, 24, 25, 26, 27, 28, 29 før, men nå 22, 24, 26, 28, 30, 32, 34, 36. Da ble alle motorene nesten like raske.

Etter noen flere tester ble utført på prototypen kom det nye problemer. En motor begynte å ikke funke som den skal. Vi så at den hadde dårlig ledningstilkobling, så vi byttet bare ledningens. Men når vi testet igjen funket ingenting. Kontrolleren og motorene frøs og ville ikke reagere. Ingen knapper hjalp og motorene peip som om de skulle kjøre men hadde fått for lite spenning til å flytte seg. Vi hadde vært borti det prøblemet tidligere, da det ble løst med å koble til og fra PS4 kontrolleren flere ganger. Denne gangen hjalp ikke det. Vi gikk til en eldre kode som bare handlet om å styre motorene igjen, i tilfelle det var noe med de andre sensorene som ble lagt til som kunne ha rotet til noe. Ved feilsøking fant vi at motorene fikk spenning, melleom 0.1 V - 0.001 V for hver gang vi lastet opp koden på nytt og prøvde å kjøre med PS4

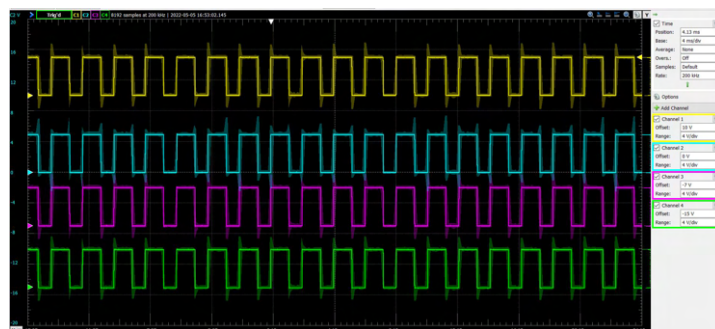
kontrolleren. Systemet funket fint helt til vi skulle begynne å styre farten til motorene med PS4. Med en test kode som ble laget for å sjekke hver og en motor for seg, for å se om det var noe galt med oppkoblingen, viste at alle motorene kunne kjøre optimalt. Så vi la til på den gamle motor kjøre koden, at pcen skulle kjenne når vi trykket på R2 og L2 på PS4 kontrolleren. Den la merke til R2 og L2 helt fram til joysticken ble flyttet. Da frøys verdien rett over overgangen fra motorene til å stå i ro til kjør. Stå i ro verdiene ligger mellom 150 - 110. Når vi dytter oppover på joysticken, skal farten øke mens verdien senkes mot 0. Men etter vi dytter joystick oppover, ble verdien låst på 109 og lignende, og motorene fikk en svak spenning mellom 0.1 V - 0.001 V. Vi kunne ikke flytte den vekk fra 109, og den fikk ikke lest om L2 og R2 ble trykket lengere. Vi tenkte den hadde fryst fast i posisjon, men den fortsatte å printe 109, som sier at den fortsatte å gå i loopen, men den trodde joysticken var i posisjon 109 og ingenting annet skjedde.

Etter mer testing fant vi i koden at verdien mSpeed (mappet fart fra kontroller til motorene) til pwm pins rotet til styringen. Ved å gi en konstant fart på 255 til motorene istedet, så funket alt normalt. Det funket også når pwm1 til motor 1 hadde mSpeed, men problemene oppsto med en gang pwm2 ble styrt av mSpeed.

Vi fikk til et møte med Gustav fra SSI. Vi testet med å gi hver pwm pin farten til mSpeed. Først pwm1, så med pwm4, så med pwm 3. Alt gikk fint med mSpeed, helt til vi ga pwm2 farten til mSpeed. Fortsatte å teste med konstant tall, loope konstant tall også videre. Fikk forskjellige feil. motor 1 og 3 hørte etter, motor 2 kjørte lengere tid enn det den skulle, motor 4 kjørte ikke i det hele tatt den ene vegen. Gustav tipset da om å sjekke signalene til pwm pinsene på arduinoen. Da tok vi fram Waveforms kittet, fra texas instruments. Koblet opp og sjekket. Fant at pwm pin 4, som motor 3 var koblet til, gikk mye raskere enn de andre signalene, se fig. ???. Ved å flytte motor 3 sin pwm pin til pin 8, som gikk likens som de andre, ble koblings problemet og styring løst. I fig. ser man alle pins har like signaler etter vi byttet motor 3 pwm pin.



Figur 63: Signalene skaffet fra pwm pins med en gitt verdi på 125. Øverst er pwm pin koblet til motor 1, neste til motor 2, og så videre



Figur 64: Alle pwm signaler er nå like, på gitt verdi 150

Vi prøvde å få led-stripa til å blinke blått når den ikke er koblet til noe PS4-kontroller. Dette førte til at den aldri kobla til og derfor måtte vi fjerne funksjonen. Dette er nok noe vi skal se på mellom EXPO og siste presentasjonen.

Da vi begynte å se på ultralydsensorene, så endte vi opp med en funksjon som ga oss en delay på ca 5 sek. Delayet påvirket ikke avlesningen og boksen stoppet ganske så nøyaktig. Det som ble påvirket var styringen. Når joysticken ble dyttet framover, så tok det ca 5 sek til den faktisk begynte å kjøre. Dette fikk vi løst ved å inkludere det offisielle arduino biblioteket til denne sensoren.

Sensorene og lyset fungerte hver for seg. Når de to delene ble skrevet i samme kode, så fikk vi plutselig helt rare verdier fra ultralydsensoren. Dette ble løst ved å bruke biblioteket.

8 Applikasjon

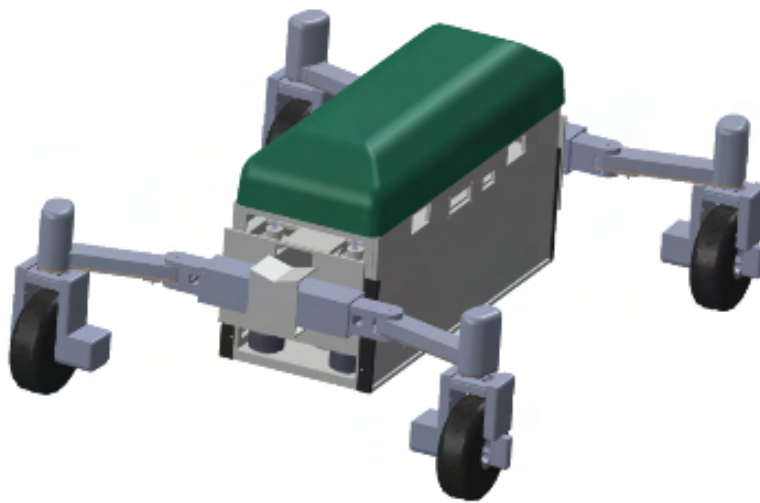
8.1 Formål

Under designprosessen kom vi tidlig fram til at det ville være hensiktsmessig med en tilkoblet app for å kunne kommunisere med kjøretøyet. Dette vil kunne gi oss mer presis kontroll, gi brukeren bedre tilbakemeldinger og det er et mer fleksibelt brukergrensesnitt enn fysiske knapper direkte på kjøretøyet. Dette vil også gjøre det enklere for oss å legge til eller endre funksjonalitet underveis i prosjektet.

8.2 Funksjonalitet

Vi har kommet frem til at vi ønsket mulighet for toveiskommunikasjon, parallelt med en eventuelt tilkoblet spillkontroll. Dette lar oss både sende og motta informasjon i sanntid. Dermed kan appen vise feilmeldinger, tilstand og hjulposisjon, samt styre kjøretøyet. Her har vi gitt en veldig detaljert input, ned til individuelle grader på hvert enkelt hjul. Manualen blir også lagt inn i appen i sin helhet, slik at den alltid er lett tilgjengelig under bruk.

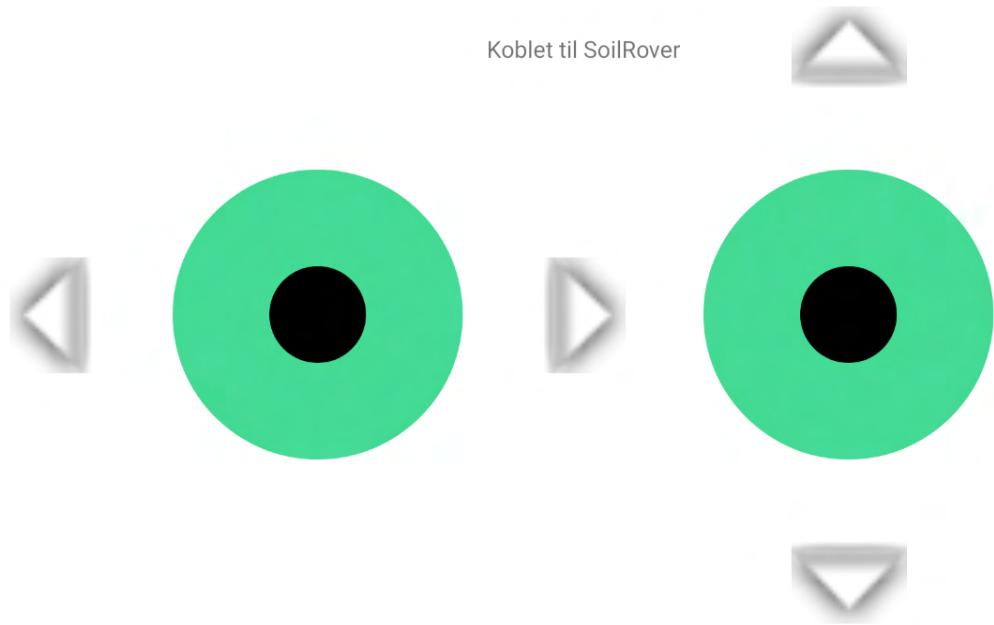
8.3 Design



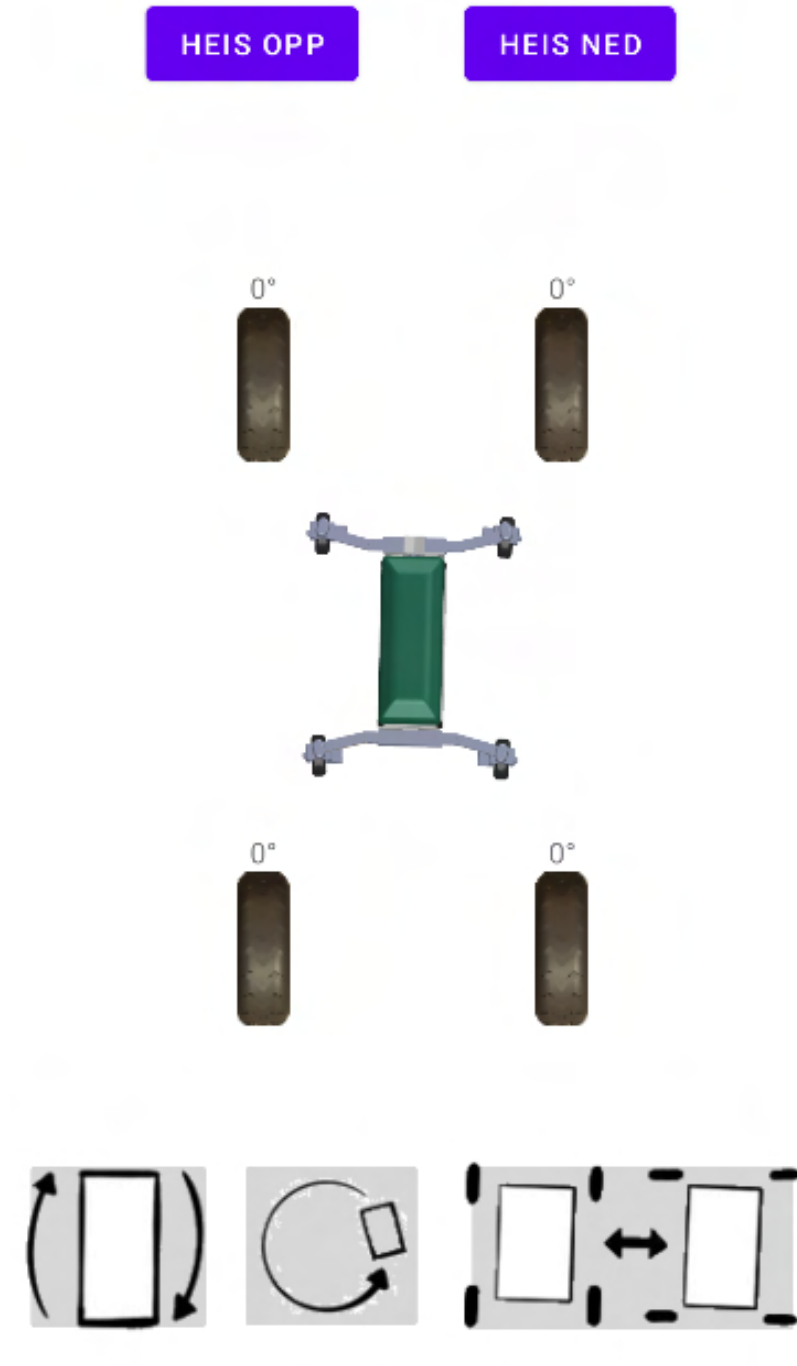
Koblet til PS4-kontroller og app

[KOBLE TIL ARDUINO](#)

Figur 65: Bilder viser startside til appen



Figur 66: Bilde viser fjernstyringside til appen



Figur 67: Bilde viser hjulstyringside til appen



Figur 68: Bilde viser siden som viser manualen



SOIL STEAM INTERNATIONAL
NETTSIDE

SOILSPROUTS NETTSIDE

Figur 69: Bilde viser siden som viser QR-kode og knappene for å komme til nettsidene

8.4 Verktøy

Vi endte på å utvikle til Android da det er et mer åpent system, som gjør det enklere å distribuere og installere en app uten å gå via en app-butikk med kompliserte godkjenningsrutiner. Her har Google har utviklet et dedikert utviklingsverktøy som heter Android Studio. Dette gir mange fordeler, slik som kodeeksempler, grafisk designverktøy, samt mulighet for å kjøre og teste appen på en virtuell mobiltelefon.

8.5 Språk

Vi har kommet fram til det beste vil være å skrive appen i kotlin og ikke java. Det er flere grunner til det. Kotlin er mer fleksibelt og moderne. Java-kode kan inkluderes i kotlin, men ikke andre veien. Dette fører til at vi kan benytte oss av begge deler ved behov. Kotlin kode er mer kortfattet som fører at den er enklere å lese og at det er vanskeligere å gjøre feil. Ofte så skyldes kræsje i java programmer NullPointerException, det vil si at programmet prøver å bruke en variabel ikke har noe verdi men som er tom. Når man skriver kotlin kode så er det ikke noe som kan skje da kotlin variabler er nødt til å ha en verdi. [32]

8.6 Tekniske løsninger

8.6.1 Model-view-viewmodel

Denne arkitekturen separerer dataverdier fra deres fremvisning. Det gir mer fleksibilitet i både design og i forretningslogikk, da disse er mer frikoblet. Det gjør det også mulig å gjenbruke data uten å måtte kopiere den. Det gjør det også enklere å få appen mer responsiv, da en annen tråd kan jobbe med å modifisere dataene i bakgrunnen enn den som også jobber med fremvisningen.

8.6.2 Singleton

En Singleton er et dataobjekt som det kun finnes en instans av i prosjektet. Denne instansen er dermed gjort globalt tilgjengelig, og det er mulig å få tak i uten å måtte sende rundt en referanse. I vårt tilfelle er dette brukt for å holde på tilstanden til Bluetooth-tilkoblingen til kjøretøyet, og gjøre tilgjengelig metoder for å sende og motta data.

8.6.3 Bluetooth

For å fasilitere toveiskommunikasjon via Bluetooth er det en rekke ting som må være på plass. Først og fremst kreves det en hel rekke tillatelser fra brukeren. Dersom disse ikke gis, må vi vise en feilmelding og be brukeren prøve på nytt. Deretter må en finne korrekt tilkoblet enhet, da mobilen kan være koblet til flere andre enheter samtidig (klokke, headset etc). Dette har vi løst ved å gi enheten vår et unikt navn. Når vi så er koblet til spinner vi opp en dedikert tråd for å håndtere kommunikasjonen. Denne tråden har som oppgave å overvåke eventuelle innkommende kommunikasjon, og så eventuelt oppdatere riktig datasett når dette kommer. Denne tråden kan også motta data fra andre tråder, som den så skal sende til kjøretøyet. Dette gjør appen mer responsiv, da denne kommunikasjonen kan ta litt tid.

8.6.4 Joystick

Vi ønsket også styring via appen og den skulle være så lik som mulig PS4-kontroller styringen. Derfor valgte vi å inkludere 2 virtuelle joysticks i appen. Et bibliotek ble inkludert for dette. Valget falt på et bibliotek fra github.[\[33\]](#) Grunnen til det var at biblioteket hadde all funksjonalitet vi trengte.

8.6.5 PDF-Reader

Vi ønsket også å kunne vise en manual via appen. Et bibliotek ble inkludert for dette. Valget falt på et bibliotek fra github. [\[34\]](#) Biblioteket hadde innebygget scrolling og zooming, som gjør manualen mer brukervennlig.

9 SoilRover Simulator

En del av oppgave gitt av SSI var å lage simulasjon av SoilRover, basert på en dynamisk modell. SoilRover Simulator er en blanding av et enkelt spill og en simulasjon. Et spill involverer interaksjon og handlinger med user interface eller input enhet, som for eksempel en kontroller eller tastatur, for å generere visuell tilbakemelding på en skjerm. [35] Simulering er etterligning/driftsimulasjon av en virkelig prosess, avgrensede deler av et system, eller hele systemet ved hjelp av modeller som representerer nøkkelfunksjoner eller oppførselen til det valgte systemet. [36] SoilRover Simulator kan identifiseres med begge - SoilRover har alle nøkkelfunksjoner til den reelle maskinen, og samtidig kan den styres med både tastatur og PS4 kontroller for å få en visuell tilbakemelding på skjermen.

9.0.1 Verktøy

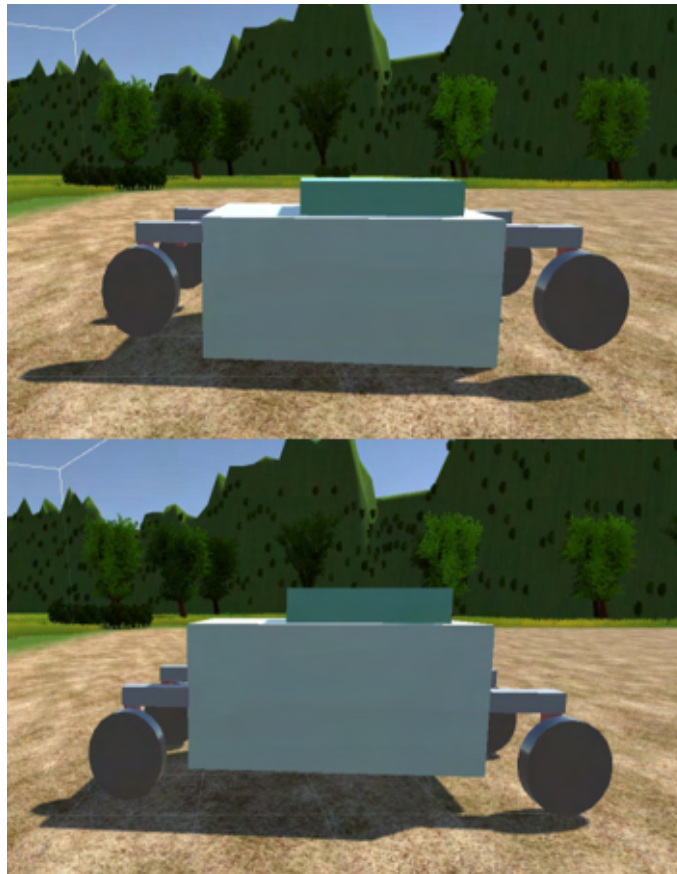
Vi har valgt å simulere kjøretøyet i Unity, som er en multiplattform spillutviklings motor laget av Unity Technologies og utgitt 2005. Programmeringsspråket som ble brukt for skriptene er C#.

Unity tilbyr mange av de viktigste funksjonene som utviklere trenger for å lage spill eller en simulasjon – det er blant annet forskjellige fysikkmotorer som kan brukes i henhold til prosjektbehov: 3D, 2D, objektorientert eller dataorientert. De gjør det lettere å implementere riktig fysikk - for eksempel akselerasjon, kollisjon eller gravitasjon.

9.1 Modell

9.1.1 Prototype

I begynnelsen laget bachelorgruppen en forenklet simulert versjon av kjøretøyet. Formålet var å vise på 2. presentasjon hvordan simulasjonen kommer til å se ut, fordi 3D modell ikke var ferdig på det tidspunktet. Via en demo-modell var det lettere å presentere nøkkelfunksjoner, men også gi en generell forståelse av denne oppgavedelen. Demo-modellen var også brukt for å implementere enkle bevegelser, teste kjøring og «Wheel Colliders». Mer om prototypen kan leses i vedlegg [F.1.1](#).



Figur 70: Prototype av SoilRover med hjul heist opp og ned

9.1.2 Nåværende modell

Modellen som brukes i sluttversjon av simulasjon har bachelorgruppen laget i SolidWorks. Modellen ble konvertert til OBJ format, som er en generisk 3D modell format, kompatibel med Unity. For at delene av SoilRover skal kunne beveges, var det nødvendig å implementere bevegelige deler som separate objekter, og deretter nøyaktig koble de sammen.



Figur 71: 3D modell i hovedposisjon

9.2 User view

Generell beskrivelse av simulasjonssystem.

9.2.1 Styring

Tabell 19: Styring

Knapp	Hva den gjør	PS4 DS kontroller
W, S, A, D	Styring	LT
E	Åpne/lukke armer	PS4 knapp
1-9	Legge til hindringer	-
PageDown	Heise ned	L1
PageUp	Heise opp	R1
F1	Rotere om seg selv	Trekant
F2	Kjøre i en sirkel	Sirkel
F3	Framhjulsving	X
F4	Bakhjulsving	Firekant
F5	Styre hvert hjul separat	Options
Space	Brems	-
Shift	Aktivere damp	-

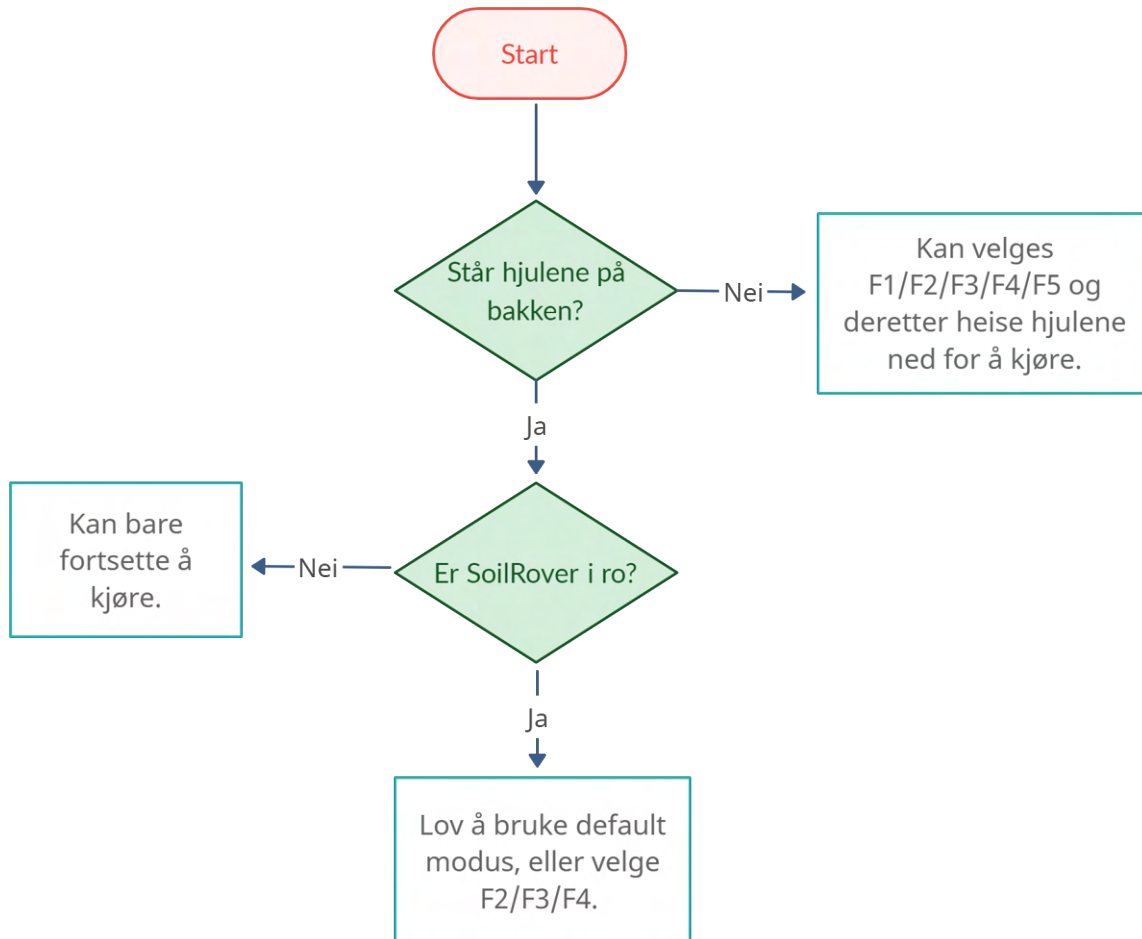
Når simulasjonen starter står SoilRoveren i en default kjøremodus i hoved posisjon, som vist på fig. [126](#). Default kjøremodus betyr at alle hjulene svinger samtidig ved styring, og maksimal sving for alle hjul er 15 grader. Hoved posisjon er følgende: armene står i 0 grader posisjon, hjulene står på bakken. Ut fra denne posisjonen, er det følgende muligheter:

- kjøre frem eller rygge - brukt hovedsakelig for å kjøre inn og ut av transportbil
- aktivere F2, F3 eller F4 modus
- heise opp hjulene, og deretter har man følgende muligheter:
 - åpne armene, heise hjulene ned og kjøre videre
 - åpne armene, velge mellom F1-F5 modus, heise hjulene ned og kjøre

Tabell 20: Beskrivelse av styring

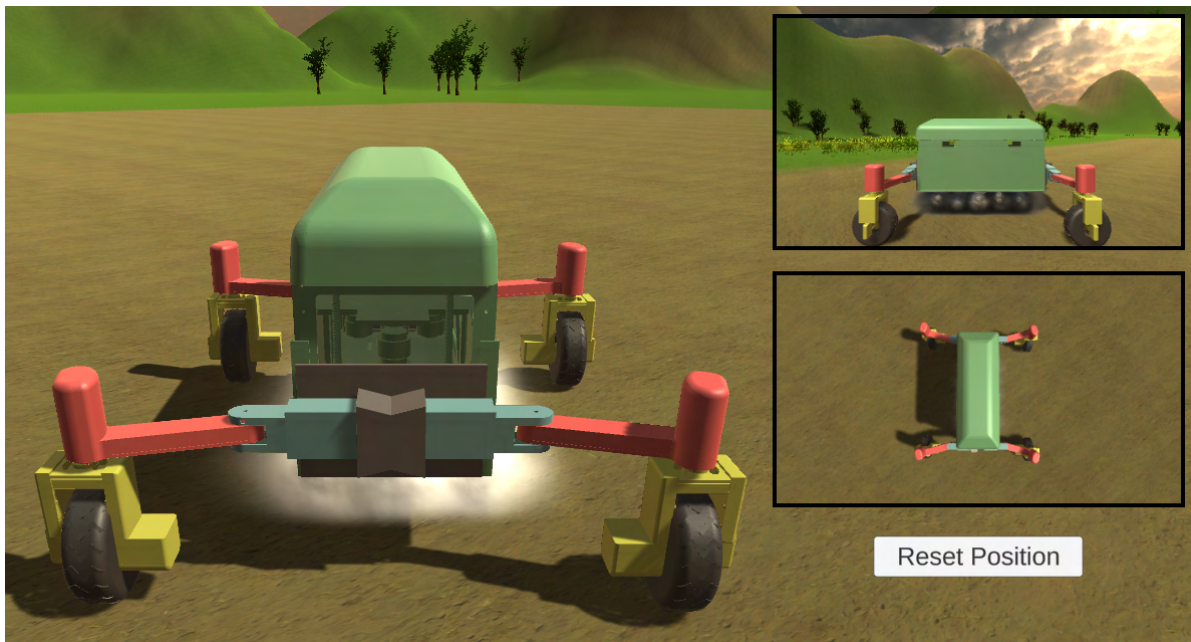
Kjøremodus	Beskrivelse	Detaljert styring
F1 - Rotere om egen akse	SR skal snu på stedet; hjulene skal automatisk settes til $50,73^\circ$, noe som ble beregnet ved hjelp av standard trigonometri i Ackermann .	Styres ved å presse W, ↑ eller LT på kontroller.
F2 – Rotere om ett punkt	SR skal kjøre i en sirkel med radius valgt av bruker. Det er kun mulig å skrive inn integer verdier	Skrive inn radius i input text field som dukker opp på skjermen og bekrefte med å trykke Enter”. Trykke W, ↑ eller LT på kontroller.
F3 – Framhjulsving	SR skal kunne kjøre og det er bare to fremme hjula som skal svinge. Tillat sving er 15 grader.	WSAD knapper, piler eller LT på kontroller for å styre.
F4 - Bakhjulsving	SR skal kunne kjøre og det er bare to bakerste hjula som skal svinge. Tillat sving er 15 grader.	WSAD knapper, piler eller LT på kontroller for å styre.
F5 – Styre hvert hjul separat	Bruker skal kunne velge vinkel for hvert hjul separat. Hvis ingen vinkel blir skrevet, skal hjulene ha 0 grader posisjon. Det er kun mulig å skrive inn integer verdier.	Skrive inn ønskede vinkler i 4 input fields som dukker opp for hvert hjul og bekrefte med å trykke Setknapp

Når man er i en aktiv modus, er det kun mulig å kjøre etter bestemte regler som definerer enten et kjøremønster eller styringsoppsett, hvor kun bestemte knapper skal svare på brukerens input. Etter at en modus blir valgt, må hjulene heises ned for å kunne kjøre.



Figur 72: Kjøremodus

SoilRover har en tilleggsfunksjon som imiterer damp ved å trykke «Left shift».



Figur 73: Damp funksjon

9.2.2 Simulasjonsmodell

En modell representerer alle viktige funksjoner og egenskaper til et system som man er interessert i å simulere. Presentert modell er enklere enn den reelle SoilRover, men har alle nøkkelfunksjoner til maskinen:

- hjulene roterer 360 grader
- maskinen har samme kjøremoduser
- det er mulig å heise hjulene opp og ned
- armer kan settes i to posisjoner: 0 eller 75 grader
- det brukes Ackermann styring

For rotasjon av elementer brukes det Quaternion Unity class og dets funksjoner. Quaternions i matematikk forlenger komplekse tall [37]. I Unity representerer Quaternions vektorer med 4 komponenter - w, x, y, z i 3D [38]. Likevel er rotasjon i systemet fremstilt i grader, siden Unity oversetter Quaternions til Euler for å vise dem i Unity Inspector (Euler rotasjon er representert i grader).

9.2.3 User interface

User Interface sørger for HCI (Human-Computer Interaction), og er en viktig del av programmet. Den lar brukeren velge mellom mulige moduser og justere innstillinger for maskinen gjennom direkte input fra tastatur, eller gjennom handlinger på HUD (Heads-Up Display).

Start Menu

Start Menu er det første scene som brukeren ser etter at programmet blir executed/utført?. Den gir 3 mulige valg:

- Start – simulasjon begynner.
- Controls – viser liste med knapper for både tastatur og PS4 kontroller.
- Exit – programmet avsluttes.

Det er ikke mulig å gå tilbake til Start Menu fra simulasjonen. Den lastes opp kun når simulasjon applikasjon blir utført.



Figur 74: Start menu

CONTROLS			X
ACTION	KEYBOARD	PS4 DS CONTROLLER	
STEERING	ARROWS / WSAD	LT	
BREAK	SPACE	-	
LIFT UP / LIFT DOWN	PAGE UP / PAGE DOWN	R1 / L1	
OPENING / CLOSING ARMS	E	PS4	
SPECIAL MODES	F1 / F2 / F3 / F4 / F5	TRIANGLE / CIRCLE / X / SQUARE / OPTIONS	
STEAM ON / OFF	LEFT SHIFT	-	
ADD OBSTACLES	1-9	-	

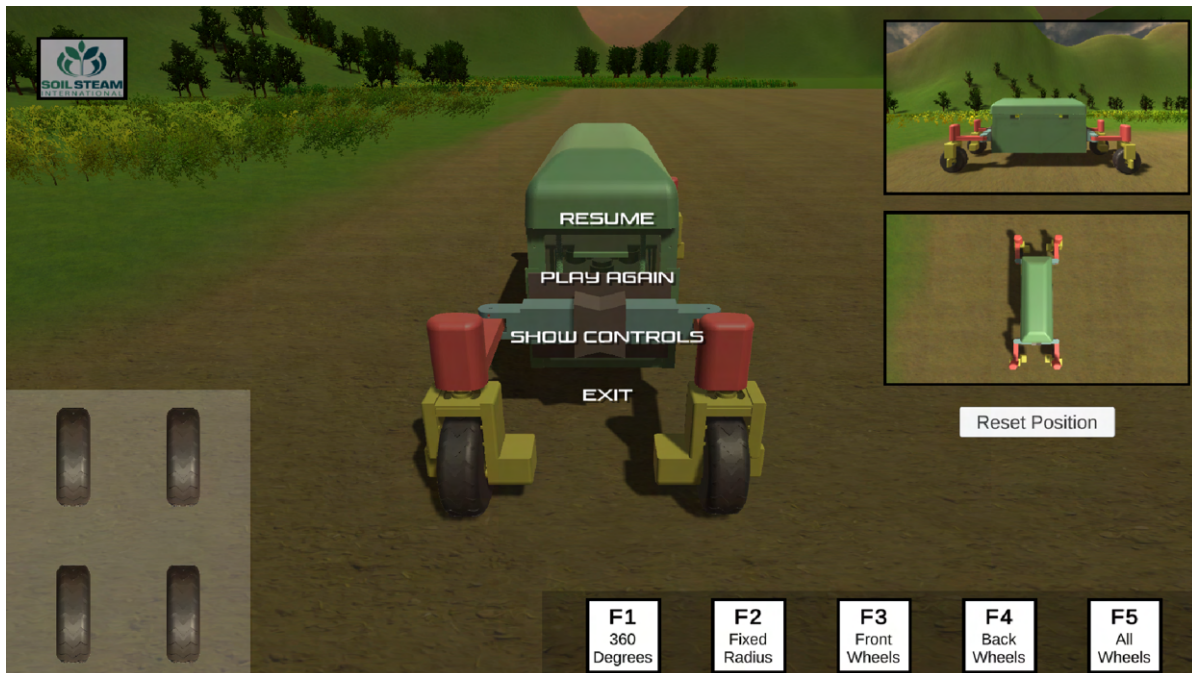
Figur 75: Controls

Pause Menu

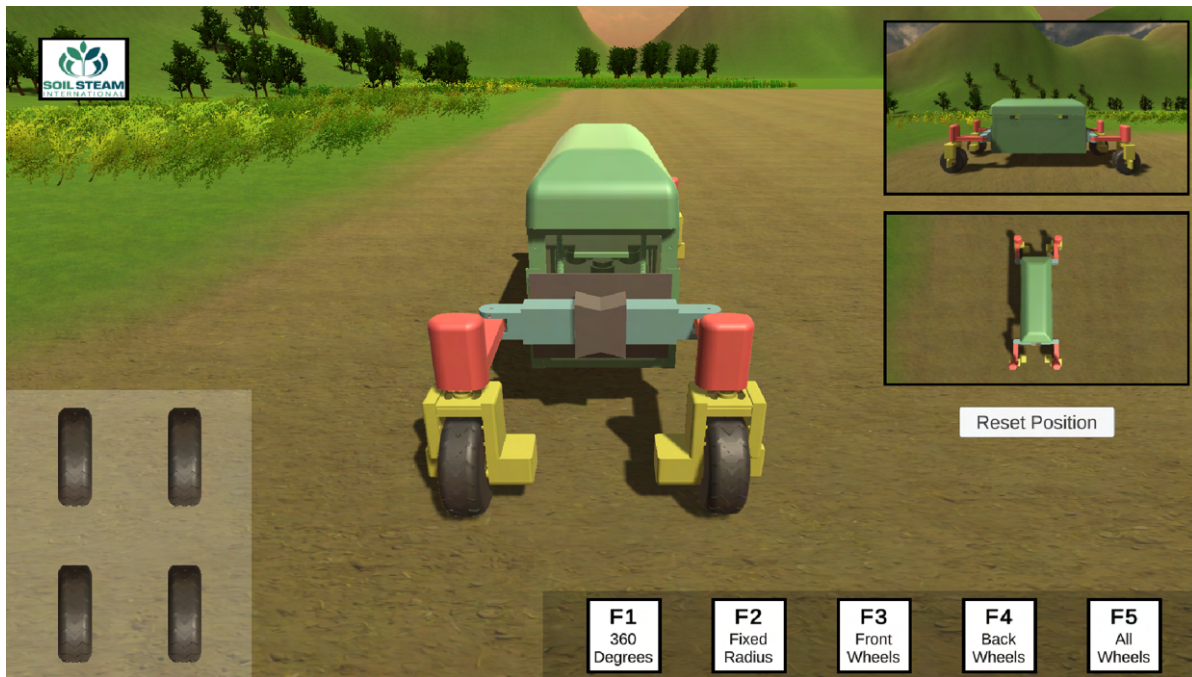
Pause Menu kan aktiveres fra simulasjon ved å trykke på «Esc» knapp. Den gir 4 mulige valg:

- Resume – tilbake til simulasjonen med samme tilstand.
- Play again - scenen blir lastet på nytt.
- Show controls – viser liste med knapper for både tastatur og PS4 kontroller.
- Exit – programmet avsluttes.

Det er også mulig å gå ut av Pause menu tilbake til simulasjonen ved å trykke «Esc» igjen.



Figur 76: Pause menu



Figur 77: Heads-up display

Heads-Up Display er en viktig del av programmet som gjør at informasjon om SoilRover videresendes visuelt til spilleren og vises på kanter av skjermen (riktig norsk?):

1. Hjulposisjon – hjulene reflekterer bevegelser til hjulene av SR. Ved styring av SR, HUD-hjulene roterer med samme hastighet og står i samme vinkler som hjul til SR.



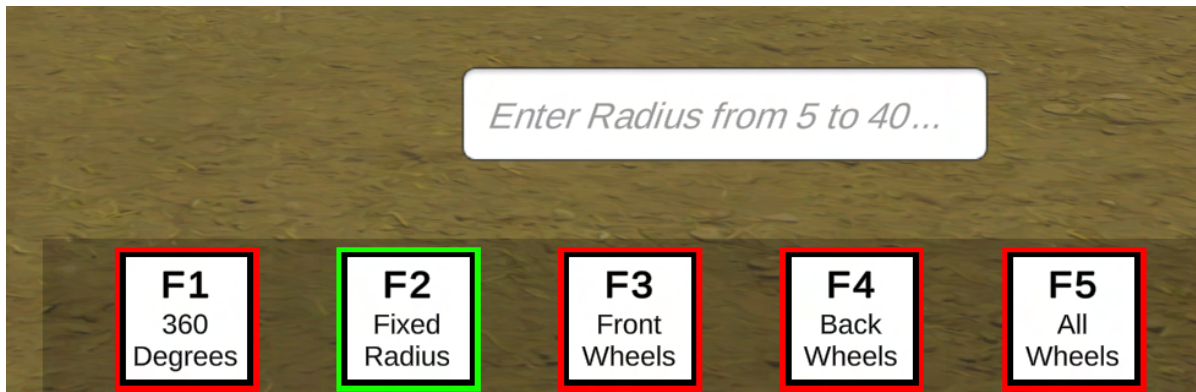
Figur 78: Hjulene - HUD

2. Kjøremodus display – viser 5 tilgjengelige kjøremoduser, samt knapper som er knyttet til hver av dem. Når en modus er valgt, vises den med en grønn ramme, og resten vises med røde rammer.



Figur 79: Moduser

3. Kun for F2 modus: det dukker opp et input vindu, hvor brukeren kan velge en radius som SR skal rotere om. Verdien skal bekreftes med «Enter». Verdier skal bekreftes med «Set» knapp/ «Enter».



Figur 80: Enter radius vindu

4. Kun for F5 modus: det dukker opp 4 input vinduer, hvert vindu er tilkoblet ett hjul. Brukeren kan skrive inn en numerisk verdi for å sette vinkel. Verdier skal bekreftes med «Set» knapp/«Enter». Skriver mann 0, skal hjulet stå i 0 posisjon.



Figur 81: Modus F5 - 0 grader posisjon

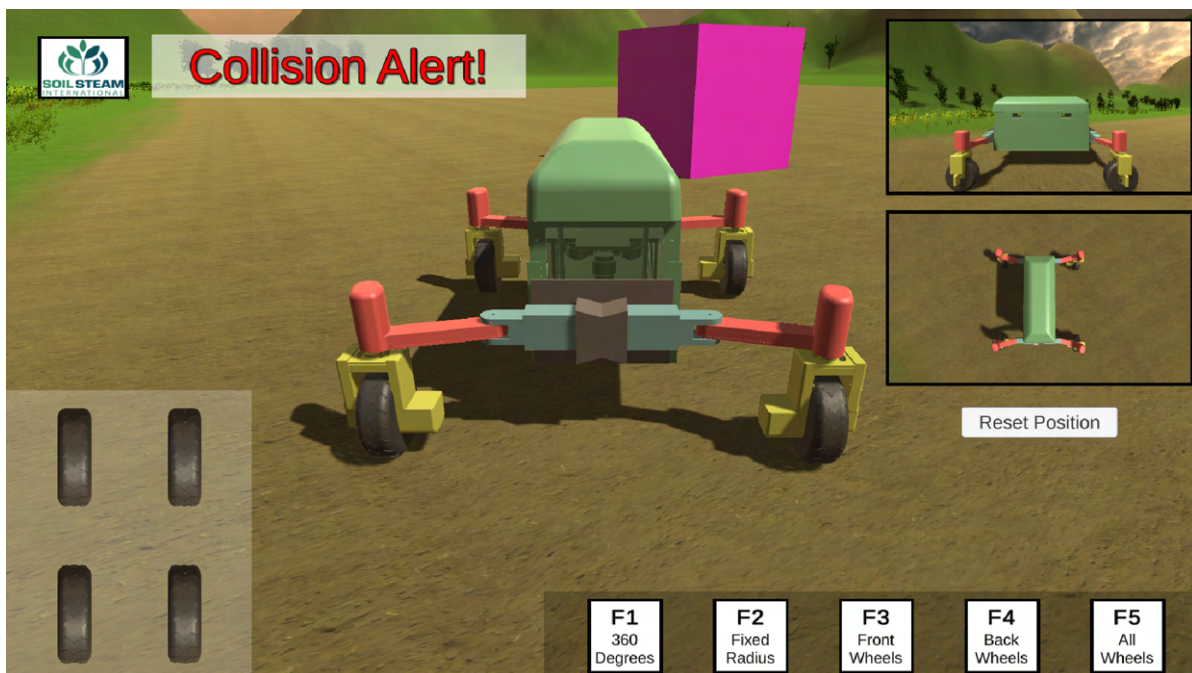
5. Reset Position knapp restarter scene. SoilRober vil bli flyttet tilbake til startposisjon.

6. Det er to ekstra kameraer i høyre hjørnet av skjermen. De er aktive hele tiden, for alle kjøremodusene.

9.2.4 Kollisjon deteksjon

SR er i stand til å detekttere objekter som befinner seg i alle 4 retninger rundt kjøretøyet. Når SR detekterer et objekt i nærheten, dukker det opp en rød tekst «Collision alert» i øvre delen av skjermen. Dersom bruker bestemmer å kjøre videre mot en hindring, vil SR bli stoppet og det blir kun mulig å rygge.

Hindringene kan aktiveres med numeriske knapper: 1-9. Knappene gjør at skjulte objekter faller ned på åkeren. Objekt-knappene kan brukes når som helst, i alle kjøremodusene.



Figur 82: Modus F5 - 0 grader posisjon

Mer om tekniske løsninger kan leses i kapittel [F](#).

Mekanisk

Ting som kan forbedres i etter tid er en forsterket skulder, med den tiden vi hadde så ble den forhastet og ikke testet like grundig som ønsket. Ytterveggene kan bli byttet ut om ønsket å ha dem i et annet materiale.

Elektrisk

Motorkontrollene er nok best å bytte ut med noe som har mindre spenningsfall og forhåpentligvis kan koble til flere motorer om gangen. Per nå bruker vi 5 stykk, 2 framme og 2 bak til driv og rotasjon og 1 til heisemekanismen, men dette kan greit justeres til 1 framme og 1 bak. Det er mulig å finne alternativer som allerede er ute eller så er det mulig å designe en selv.

For mer sikkerhet av rotering ved lavere hastigheter anbefales det enten ekstern giring hvis dette kan implementeres med lettere designendringer for å øke momentet til nåværende motor for rotasjon ellers er det et behov for en sterkere motor enn det som er nå. Nødvendig moment er å finne fra utregninger i (12) som da er rundt 0,6 Nm med nåværende dekk og vekt.

Data

Simulasjon

Gjennomføre en generell og nøyaktig feilsøking, det vil si: å finne ut hva som skjer når maskinen glider etter en stund av kjøring, finne årsak til «Mellomrom» trykk som gjør at scenen lastes inn på nytt. Dessuten kan det legges til ekstra funksjonaliteter: mulighet å justere flere SR variabler fra simulasjonsvinå, uten å måtte bruke Unity, implementere algoritme som skal legge til objekter i tilfeldige steder (nå dukekr de opp på bestemte steder). Det kan også legges

til bevegelige objekter, for eksempel et menneske - lage en timeline og animasjon for det. For mer brukervennlighet kan det også vises instruksjoner for kjøring i selve applikasjonen, avhengig av aktiv kjøremodus.

10 Oppsummering og framtidige ideer

10.1 Oppsummering

Ved innlevering av denne rapporten har vi utviklet design av alle de forskjellige egenskapene som SoilRover ønsket og implementert dette i en simulering. I tillegg har vi bygd to forskjellige bokser som inneholder tester og oppsett av disse egenskapene. Sensor prototypeboksen er brukt til oppsettet av kommunikasjon og armmodulene. Den har noen ultralyd sensorer koblet til, samt en Arduino Mega med et host shield for bedre oppkobling og et USB shield for styring via PS4 kontroller. Armmodulene er skrudd på undersiden av boksen, og det utvikles en skulder for disse videre, slik at armmodulene står stødigere. Armmodulene er fullt utstyrt med rotasjons motor, driv motor og driv enkoder. Målet videre med denne er å lodde og koble sammen alle komponentene og så styrt rotasjon og driv med samme kode. Den andre boksen er starten på siste versjonen. Den består av laserkuttet rammer og trefinne plater som vegger, som kan sklis ut og inn. På ene lang vegg er det utkuttet plass til LCD skjerm, USB for bluetooth tilkobling for PS4 kontroller og utkutt til bluetooth tilkobling til applikasjonstilkobling samt en av/på knapp. På alle veggene, lang veggene og heise veggene, er det tilsammen plass til 6 ultralyd sensorer. Innsiden av boksen består av to plater, hvor øverste plate skal inneholde alt av elektriske komponenter som ikke er på armmodulene og underplaten. Tenkt plassering vist i fig. 18. Underplaten er det satt opp for heisemekanismen. Det er kuttet hull for stødigere plassering av trinsene og skruene i bunnen, samt et hull for å få steppermotoren lavest mulig. Planen videre er å få satt opp og koblet opp heisemekanismen slik at den kan kjøre, og få inn av/på knappen i kretsen. Dekselet laget av kompositt er også under produksjon. Hvis vi hadde hatt tid til å implementere alt, skulle alt av komponenter overføres fra sensor prototype boksen til den siste versjonen, og LED slangen skulle blitt festet til underiden av nederste plate, med et make-shift gitter laget av.. under der igjen. Vi ville prøvd å lage sideveggene av glassfiber igjen. Vi har fått appen til å kommunisere med bluetooth enheten for å få og gi informasjon tidligere, og grunnfundamentet for å gjøre det igjen er fortsatt der. Den kan også bli brukt

til å vise forhåndsbestemte PDFer. Selv om simulasjonen ikke kan anses som et fullstendig ferdigstilt produkt grunnet feil nevnt i [10.2.1](#), klarte vi å oppnå de viktigste funksjonalitetene til SoilRover. Dette gjør det mulig å virtuelt prøvekjøre SoilRover med samme styringsoppsett som den fysiske SoilRover har.

Når vi sammenligner det vi har oppnådd mot original planen vår, visualisert i GANTT-diagrammet v.1 fig. [143](#), ser vi at vi ligger bak planen på det fysiske produktet. Mye av tiden ble brukt til valg av elektriske komponenter og DAK Design og analyse av produktet. Etter vi fikk laget prototypene ble også mye tid brukt på de for å feilsøke koblinger og kode, som vist i det nyeste GANTT-diagrammet v.3 fig. . Oversikt over app og simulasjon. App ligger etter plan, men blitt nedprioritert etter prototypene ble laget.

10.2 Forbedringspotensiale

Vi har en del ting som vi ønsker å få gjort mellom innlevering av denne rapporten og presentasjon 3 som er 1.juni.

- Fungerende heis separat
- Deksel på plass
- Skulder til prototype
- Spray paint
- Videre styring og mer test og prøve implementere driv og rotasjon i sammensatt prototype boks
- Montere av/på
- Simulasjon: fiks kræsjer , finne årsak gliding, lage funk som vier instruks på skjerm for alle modus.

- App skal kunne sende og motta data via bluetooth.
- Rydde opp i app kode.
- Ha en fungerende app med minimale problemer
- Skrive manual for bruk av kjøretøyet og simulasjonen
- Legge til QR kode i app, til nettsiden
- legge til en knapp som åpner nettsiden vår

Og hvis man hadde hatt mer tid, eller omgjort teknisk

10.2.1 Resterende planer

Som nevnt i dokumentasjonen for kroppen, **Komposittdeler**, gikk det galt med glassfiberplattene. Om vi hadde hatt mere tid, er dette noe vi ville gjort på nytt. Vi ønsker også å minke total vekten, det er da i hovedsak av heisemekanismen I ettertid så vi at trinsene kunne hvert 3D printet. På grunn av vekten til heisemekanismen så har den blitt separert fra resten av SR. Som nevnt tidligere er det vist at det er nødvendig med sterkere rotasjons motorer eller ekstern giring. Dette vil føre til endringer i designet. Vi ser også at det er behov for en forsterket skulder, men med den tiden vi hadde så ble det forhastet og ikke testet like grundig som ønsket. Ytterveggene kan bli byttet ut om det er ønsket å ha dem i et annet materiale.

Motorkontrollene er nok best å bytte ut med noe som har mindre spenningsfall og forhåpentligvis kan koble til flere motorer om gangen. Per nå bruker vi 5 stykk, 2 framme og 2 bak til driv og rotasjon og 1 til heisemekanismen, men dette kan greit justeres til 1 framme og 1 bak. Det er mulig å finne alternativer som allerede er ute eller så er det mulig å designe en selv.

For mer sikkerhet av rotering ved lavere hastigheter anbefales det enten ekstern giring hvis

dette kan implementeres med lettere designendringer for å øke momentet til nåværende motor for rotasjon ellers er det et behov for en sterkere motor enn det som er nå. Nødvendig moment er å finne fra utregninger i (12) som da er rundt 0,6 Nm med nåværende dekk og vekt.

Lettere ladning av batteri blir med å anskaffe en stikkontakt som brukes til å lade forskjellige typer kjøretøysbatterier for eksempel moped. Dette blir da for å gjøre det enklere å lade SR når den ikke er i bruk. Dette blir da gjort med en stikkontakt med ledning istedet for at må koble til polene.

Simulasjon kunne ha vært mer nøyaktig, vi burde ta mer hensyn til maskinens dynamikk. Kode kunne ha vært mer ryddig og strukturert. Ting som burde fikses: fikse mellomrom som crasher, gliding, vise instruks på skjermen for alle modus for å informere brukeren om hva som er lov eller ikke, fikse en feil med hjulmodul i modus F3 og F4. Dessuten kunne simulasjonen ikke sammenlignes med fysiske SoilRover for å få mest realistisk resultat.

Simulasjon

Gjennomføre en generell og nøyaktig feilsøking, det vil si: å finne ut hva som skjer når maskinen glider etter en stund av kjøring, finne årsak til «Mellomrom» trykk som gjør at scenen lastes inn på nytt. Dessuten kan det legges til ekstra funksjonaliteter: mulighet å justere flere SR variabler fra simulasjonsvinnå, uten å måtte bruke Unity, implementere algoritme som skal legge til objekter i tilfeldige steder (nå dukker de opp på bestemte steder). Det kan også legges til bevegelige objekter, for eksempel et menneske - lage en timeline og animasjon for det. For mer brukervennlighet kan det også vises instruks for kjøring i selve applikasjon, avhengig av aktiv kjøremodus. Appen og koden til mikrokontrollerne kan man alltid forbedre. Appen kan ha mer funksjonalitet. Siden vi ikke er designer, så er det nok noe som kunne forbedres. Det er heller ikke tatt hensyn til universell utforming.

Dersom prosjektet skulle føres videre, et det en del ting som burde organiseres annerledes:

- Ikke bruke altfor mye tid på valg og kalkulasjon av risiko. Beslutninger burde tas tidligere.
- Bedre kommunikasjon og informasjons henting fra tredjeparter - henvende oss til lærere, fagfolk, veiledere med spørsmål og usikkerhet.
- Ta i bruk ressurser.
- Tettere samarbeid mellom gruppemedlemmer, for å få bedre kommunikasjon og informasjonsflyt.
- Holde oss til Scrumban modell bedre, og følge strategisk satte ting tettere, for å ha bedre oversikt over framgang.
- Bedre dokumentering, gjort med jevne mellomrom.

11 Konklusjon

De separate delene i prosjektet er forholdsvis ferdig hver for seg, men de er enda ikke blitt testet i sin helhet, som er noe vi ønsker å få gjort fram til presentasjon. Appen er i fungerende bruk, men mangler noe den kan testes med. Simulasjon er utført og lar seg styres med både tastatur og PS4-kontroller, men har noen resterende feil, og er ikke så nøyaktig som den burde. Det mekaniske designet for SR er ferdigstilt for kroppen og armmodulene så det er kun å settes sammen. Plassering og implementering for motorerkontrollering er tiltenkt og fungerer, en bør utbedres for bedre resultat av fullt sammensatt modell. Utbedringene er utledet i teksten om forbedringspotensial. Ting skulle ha vært bestilt tidligere slik at gruppen hadde mulighet til å teste diverse ting før, men konseptene til de forskjellige bevegelsene er kommet fram hver for seg. Samarbeidet i gruppen tok noe lengre tid enn ønsket.

Bidrag

Vetle

Administrativt

Vetle har hatt ansvaret for den økonomiske delen av prosjekt der medlemmer har lagt inn i bestillingsliste med hvor det er å anskaffe, pris, hvor mange av det vi trenger og behovet for det. Fra leverandører som oppdragsgiver var kunde hos, så bestilte vi på vegne av oppdragsgiver slik at faktura gikk gjennom de. Leverandører eller ting som ble handlet i fysisk butikk la medlemmer ut for og registrerte kvitteringer i en utgiftsliste med hvem som la ut for diverse ting, som Vetle deretter sendte til regnskapsavdeling hos oppdragsgiver for å fikse tilbakebetalinger. Den økonomiske arbeidet er blitt gjort kontinuerlig gjennom prosjektet, men innsendte kvitteringer til SSI er bare blitt gjort tre ganger løp av prosjektet.

L^AT_EX ansvarlig med å flytte gamle ting fra Word inn i L^AT_EX og finne de nødvendige pakkene for å få ønsket utseende på dokumentet samtidig som å rette opp i pakkekonflikter. Dette er gjennomført fra April og til per dags dato.

Teknisk

Har hatt ansvar for å finne moment nødvendig for rotasjonsmotorer som er å finne i **Rotasjonsmotorer**, sensitivitet for enkoder ved styring for av hjula som er å finne i **Enkoder for rotasjon** (siste motor for rotasjon bestilt i start av april), samtidig som matematisk løsning for **Ackermann** (vært i arbeidsplan hele semesteret) og utregning på behov for batterikapasitet basert på levetid og nominell spenning **Batteri**. Anskaffet passende spenningsregulatorer for det som kjører på lavere spenning. Dette kan leses om i **Spenningsregulering** (Arbeida med fra

20.april og utover).

Angelika

Administrativt

Angelika har hatt ansvar for å lage og drifte [nettside](#). Tidligere har hun også laget Excel templates for felles bruk - timeliste og Utgifter.

Teknisk

Angelika har hatt ansvar for å lage interaktiv simulasjon av SoilRover i Unity, basert på det fysiske kjøretøyet, ved bruk av 3D modellen som ble laget av maskiningeniører. Arbeidet i Unity ble påbegynt i mars, før 2. presentasjon og avsluttet i Mai før innlevering. Dokumentasjon som forklarer hennes del finnes i [9](#) og [F](#).

Cecilie

Administrativt

Cecilie har hatt ansvar for å holde kontakten med bedriften, universitetet og gruppen. Satt opp møter med bedriften, holdt kontakten med intern veileder. Holde oversikt over tidsfrister og sette opp presentasjons dato-er og sende ut innkallinger for presentasjoner til veileder og til bedrift. Laget malen for oppfølgingsdokument og sendt dem inn til intern veileder og sendt til bedriften.

For arbeidet Cecilie har gjort så kan man lese dokumentasjon for møter i dokumentasjon 1 for hvordan organisering av møter har hvert og den generelle kontakten med bedriften og veileder og gruppen. Hun har skrevet og har gjort det som er beskrevet i kropp dokumentasjon. Hun hadde ansvar for kapittel 3.1 og 3.2 i dokumentasjon 2, om det mekaniske for dekslet og boksen. Man kan lese om den første test riggen som ble laget før 2.presentasjon i kapittel 4.4.2 i dokumentasjon 2 som hun samarbeidet med Mina på. Har skrevet om produksjons metode i rapporten.

Teknisk

Cecilie har hatt ansvar for mekanisk design av kroppen, her inngår det rammeverk, deksel og heisemekanisme, dette har pågått fra Mars og frem til midten av Mai. Heisemekanismen ble påbegynt i slutten av mars og ble ferdig designet i slutten av April. Dekselet har holdt det samme designet gjennom hele prosjektet, men deksel formen har endret seg litt, men ble påbegynt i mars og designet ble ferdig i starten av april og ble 3D-printet etter påske. Så har det blitt gjort klart i April og starten av Mai for å skulle få et ferdig første forsøk til USN-Expo 25 mai. Har også hatt ansvar for å designe vippeleddet, dette ble startet i April og ble ferdig i Mai. I samarbeid med Una ble skulderen hvor armmodulen blir festet på designet i April og ferdig designet i midten av Mai. Cecilie 3D modulerte sensorboken og limt den sammen i April, og test for driver boksen, som ble brukt under 2.presentasjon, denne ble laget fra 28 mars til og med 1 april i samarbeid med Mina og Alison. Cecilie laserkuttet ut og satt sammen begge boksene. Hun har testet med glassfiber kompositt som kan leses om i kropp dokumentasjon. Dette ble gjennomført i April. Hun skrev kapitlet om produksjonsmetoder for rapporten.

Mina

Administrativt

Mina har hatt ansvar med å sette opp og oppdatere GANTT diagrammet, som er prosjekt oversikten til gruppen. Hun har vært skribent i møter og jobbet mye med presentasjonenes utseende og animasjoner.

Teknisk

Mina har hatt hovedansvar for driv motorer **Valg av driv motor** som har blitt jobbet med siden 12.01, kjøpt og testet siden. Også hatt ansvaret for stepper motoren for heisemekanismen **Heisemekanisme**, hvor hun begynte å finne motor rundt etter presentasjon 2. Hun har jobbet med diverse prototyper med hjelp av bygging, oppkoblinger og feilsøking ?? Dette er for driv som ble bygget i uken før presentasjon 2, sensor boksen som ble startet med rett etter presentasjon 2 og til dags dato og heise prototypen som var et lite oppsett med Cecilie testet 5. mai.

Una

Administrativt

Una har tatt ekstra ansvar når det kommer til rettskrivning og redigering av andres og felles dokumentasjon. Hun har laget SoilSprouts logoen (siste uke av mars) og plakaten til expo (18.mai). Hun har 3D-printet noen av delene selv (April-Mai). 3D-modulerte og lagde prototype av alle armmodulene (april), se Armmodul dokumentasjon. Ble ferdig med alle siste ver-

sjon av armmodulene i uke 20, uten deksel og ikke malt. Hun startet på et nytt skulder design uke 20 som skal bli ferdig til expo.

Teknisk

Una har hatt gjort det mekaniske designet til hele armmodul. Hun og Cecilie har samarbeidet om designet av skulderen. Før presentasjon kan du finne hva jeg har gjort i dokumentasjon 1 kapittel 8.1 og 8.2. Mellom presentasjon 1 og 2 kan du finne hva jeg har gjort i dokumentasjon 2 kapitel 2.3. Etter presentasjon 2 kan du se hva jeg har gjort i denne rapporten under: Valg av hjul [5.1.1](#), Mekanisk design av hjulmodul [5.1.3](#), mekanisk design av arm [5.2.3](#) Armmodul prosess [C](#).

Alison

Administrativt

Alison har vært Scrumban-Master (hele semester) for prosjektet [2.1](#) dok1 kap 2.1, rap 2.1. Hun har hatt ansvar at gruppa følger scrumban blant annet ved å ha daglige møter, kanban-tavle og tidsfristene som ble bestemt. I tillegg så har hun hatt hovedansvar for å 3d-printe deler for gruppen, dette var til og med april. Dessuten så var hun ansvarlig skytjenesten som ble brukt, både med tanke på oppsett og support for de andre gruppemedlemene.(hele semester)

Teknisk

Alison har ansvaret for applikasjonen [8](#) (hele semester) rap kap 8 og programmering av begge mikro-kontrollerene(hele semester) dok1 kap 9.6 og dok2 kap 5, rap 4.2.2; 4.2.3; 6; vært med

på 7. Ansvar for testing av alle moduler, bortsett fra motor og enkoder. Hun hadde også ansvaret for å kombinere testkodene til en fungerende sammenhengende kode. Alison har vært kodeansvarlig for drivboksen (28 mars til og med 1 april) og sensorboksen (april).

Referanser

- [1] Wood Johnny. *2 charts that show the sharp rise in food prices*. URL: <https://www.weforum.org/agenda/2022/04/food-prices-fao-index-cereals-commodities-exports/>. (accessed: 17.05.2022).
- [2] Food og Agriculture Organization of the United Nations. "Land use in agriculture by the numbers". I: (). URL: <https://www.fao.org/sustainability/news/detail/en/c/1274219/>. (accessed: 16.05.2022).
- [3] Arbeidstilsynet. "Plantevernmidler". I: (). URL: <https://www.arbeidstilsynet.no/tema/kjemikalier/plantevernmidler/>. (accessed: 22.05.2022).
- [4] D.R. Paini mfl. "Global threat to agriculture from invasive species". I: *PROCEEDINGS OF THE NATIONAL ACADEMY OF SCIENCES* 113.27 (jul. 2016). DOI: <https://doi.org/10.1073/pnas.1602205113>.
- [5] Soil Steam International. *Our Science*. URL: <https://soilsteam.com/our-science/>. (accessed: 17.05.2022).
- [6] Smithers. *Three reasons why 3D printing is reaching the mainstream*. URL: <https://www.smithers.com/resources/2017/jul/reasons-why-3d-printing-is-reaching-the-mainstream>. (accessed: 22.05.2022).
- [7] Xometry. *How to get stronger 3D printed parts*. URL: https://xometry.eu/en/how-to-get-stronger-3d-printed-parts/?fbclid=IwAR0wMlhYZmOWdYMNpvnBJ03AauBiU58mt_19GezTC76FcFdcaY9_4Ch0c_w. (accessed: 15.05.2022).
- [8] 3DPROS LLC. *Choosing Infill Percentage for 3D Printed Parts*. URL: <https://www.3d-pros.com/choosing-infill-for-3d-printed-parts>. (accessed: 15.05.2022).
- [9] ERASMUS3D+. *Methodology for defining 3D printing exercises suitable for transversaleducation*. URL: <https://ec.europa.eu/programmes/erasmus-plus/project-result-content/280bdb28-2846-480f-9190-fabd75e6de0b/I01>

- A1_Transfer%20of%20knowledge%20about%20basics%20of%203D%20printing%20concepts%20to%20the%20VET%20teachers_LongVersion_English.pdf.
- [10] Anatol Locker Shawn Frey. *PETG Filament: Everything you Need to Know*. URL: <https://all3dp.com/1/petg-3d-printer-filament-all-you-need-to-know/>. (accessed: 16.05.2022).
- [11] Formlabs.com. *FDM vs. SLA: Compare Filament and Resin 3D Printers*. URL: <https://formlabs.com/blog/fdm-vs-sla-compare-types-of-3d-printers/>. (accessed: 19.05.2022).
- [12] Matt Ahart. *Types of 3D Printing Technology*. URL: <https://www.protolabs.com/resources/blog/types-of-3d-printing/>. (accessed: 19.05.2022).
- [13] Bjørn Pedersen. *komposittmaterialer*. URL: <https://snl.no/komposittmaterialer>. (accessed: 19.05.2022).
- [14] Dassault Systèmes. *Viewing the Deformed Shape of the Model*. URL: https://help.solidworks.com/2020/English/SolidWorks/cosmosxpresshelp/t_Viewing_the_Deformed_Shape_of_the_Model.htm. (accessed: 19.05.2022).
- [15] EES. *Fordeler med Lithium-jern batterier*. EES, s. 11.
- [16] Ablic. *What is a Switching Regulator?* URL: <https://www.ablic.com/en/semicon/products/power-management-ic/switching-regulator/intro-2/>. (accessed: 22.05.2022).
- [17] Richard G Budynas og J.Keith Nisbett. “Shigley’s mechanical engineering design”. eng. I: Eleventh edition in SI units. USA: McGraw Hill, 2021. Kap. 8, s. 428. ISBN: 9789813158986.
- [18] Richard G Budynas og J.Keith Nisbett. “Shigley’s mechanical engineering design”. eng. I: Eleventh edition in SI units. USA: McGraw Hill, 2021. Kap. 8, s. 433. ISBN: 9789813158986.
- [19] RS Pro. *57mm 1.8’ HIGH TORQUE STEPPER*. Stepper motor datablad. URL: <https://docs.rs-online.com/1996/0900766b8157a730.pdf>. (accessed: 22.05.2022).

- [20] mertarduinotech. *microStep*. Code from Arduino Editor. URL: <https://create.arduino.cc/editor/mertarduinotech/97cbbc77-ebcd-43ef-8543-a8c15308f2ff/preview>. (accessed: 22.05.2022).
- [21] SVASKJEMI AS. *SvaPox110TL*. SVASKJEMI AS. Råskveien 19, 1900 FETSUND. (accessed: 15.05.2022).
- [22] Hai Prasaath K. *Choosing motors for robots*. URL: <https://www.engineersgarage.com/choosing-motor-for-robots/>. (accessed: 16.05.2022).
- [23] cbenson. *Drive Motor Sizing Tutorial*. URL: <robotshop.com/community/blog/show/drive-motor-sizing-tutorial>. (accessed: 21.05.2022).
- [24] Arbeidstilsynet. *Tungt arbeid*. URL: <https://www.arbeidstilsynet.no/tema/ergonomi/manuelt-arbeid/tungt-arbeid/>. (accessed: 22.05.2022).
- [25] RS Pro. *RS PRO, 12 V dc, 2399 gcm, Brushed DC Geared Motor, Output Speed 228 rpm*. Driv motor Datablad. URL: <https://docs.rs-online.com/7068/A700000007082493.pdf>. (accessed: 22.05.2022).
- [26] Avago Technologies. *HRPG Series - Miniature Panel Mount Optical Encoders*. Datablad optisk enkoder. URL: <https://docs.broadcom.com/doc/AV02-3612EN>. (accessed: 22.05.2022).
- [27] Bayer MaterialScience. *Snap-Fit joint for Plastic*.
- [28] Biao Ma mfl. "Analysis of vehicle static steering torque based on tire-road contact patch sliding model and variable transmission ratio". eng. I: *Advances in mechanical engineering* 8.9 (2016), s. 1–11. ISSN: 1687-8132.
- [29] Gerald Chikosi. "Autonomous guided vehicle for agricultural application". Masteroppg. Nelson Mandela Metropolitan University, 2014.
- [30] William Norris. *Modern steam road wagons, 1906 / by William Norris*. English. David og Charles Newton Abbot, 1972, xiv, 173 p. : ISBN: 0715359096.
- [31] Jørgen Torgersen. "Mobile Agricultural Robot : Independent Four Wheel Ackerman Steering". I: *Norwegian University of Life Sciences, Ås* (2014).

- [32] Bansal Anshul. *Java vs. Kotlin*. URL: <https://www.baeldung.com/kotlin/java-vs-kotlin>. (accessed: 17.05.2022).
- [33] Damien Brun. *VirtuellJoystick*. URL: <https://github.com/controlwear/virtual-joystick-android>. (accessed: 22.04.2022).
- [34] barteksc. *PDF-reader*. URL: <https://github.com/barteksc/AndroidPdfViewerV1>. (accessed: 19.04.2022).
- [35] A Eilertsen og A Holm. *Dataspill*. URL: <https://snl.no/dataspill>. (accessed: 14.05.2022).
- [36] Eirik Rossen. *Simulering*. URL: https://snl.no/simulering_-_IT. (accessed: 19.05.2022).
- [37] UiO Institutt for biovitenskap. *Hyperkomplekse tall og kvaternioner*. URL: <https://www.mn.uio.no/ibv/tjenester/kunnskap/plantefys/leksikon/h/hyperkomplekse-tall-og-kvaternioner.html>. (accessed: 9.05.2022).
- [38] Unity Technologies. *Rotation and orientation in Unity*. URL: <https://docs.unity3d.com/Manual/QuaternionAndEulerRotationsInUnity.html>. (accessed: 09.05.2022).
- [39] Unity Technologies. *The Hierarchy window*. URL: <https://docs.unity3d.com/Manual/Hierarchy.html>. (accessed: 16.05.2022).
- [40] Unity Technologies. *Colliders*. URL: <https://docs.unity3d.com/560/Documentation/Manual/CollidersOverview.html>. (accessed: 16.05.2022).
- [41] Shapes.hu. *Nature Starter Kit 2*. URL: <https://assetstore.unity.com/packages/3d/environments/nature-starter-kit-2-52977#description>. (accessed: 10.03.2022).
- [42] Avionx. *Skybox Series Free*. URL: <https://assetstore.unity.com/packages/2d/textures-materials/sky/skybox-series-free-103633#description>. (accessed: 3.05.2022).

- [43] Magicpot Inc. *Effect textures and prefabs*. URL: <https://assetstore.unity.com/packages/vfx/particles/effect-textures-and-prefabs-109031#description>. (accessed: 11.05.2022).

A Risiko og spesifikasjoner

Tabell 21: Prosjekt risikoanalyse

Nr.	Hendelse	Konsekvens	Sannsynlighet	Konsekvensklasse	Forebyggende tiltak	Håndtering
R.P.01	Ny nedstengning av Kongsberg.	Vi kan bli nødt til å ha hjemmekontor, noe som kan bremse fremgang i prosjektet.	1	II	Vi følger nasjonale tiltak så langt det lar seg gjøre.	Kommunisere gjennom digitale plattformer.
R.P.02	Ikke oppnå A-prioritet prosjektkrav.	Får et ufullstendig produkt som ikke kan brukes til ønsket formål.	1	III	Ha en tydelig og godt formidlet målsetning. Følge bestemt plan. Ha fokus på hovedkrav under design. Bruk av triage og feature freeze.	
R.P.03	Deler av prosjektet er plagiert.	Prosjektet blir ikke godkjent.	1	III	Ha kildeliste med dato. Referere til kilder fortløpende. Dobbeltsjekk kilder og tekst.	

R.P.04	Manglende dokumentasjon.	Dårligere karakter på grunn av dårlig dokumentasjon.	1	III	Passe på at alle gruppe-medlemmer fører dokumentasjon, så mye som mulig. Bruke skytjenester for backup.	Når vi ser at noe mangler før hver innlevering, si ifra slik at den manglende dokumentasjonen blir lagt med til slutt rapporten.
R.P.05	Oppgaver blir gjort dobbelt opp.	Ikke effektivt bruk av tid, og kan skape forsinkelser.	1	II	Gruppemedlemmer burde kommunisere tydelig det de jobber med. Følge med på scrumban møter og iterasjonsmøter. Spørre, hvis en ikke vet hva man skal jobbe med.	Bedre kommunikasjon i fremtiden.

R.P.06	Større misforståelser mellom Gruppemedlem/-mer og/eller veileder/-e.	Kan føre til feil forståelse av prioriteringer og oppgaver som gjør at vi kan gå i feil retning.	1	III	Samtaler, utveksle meninger ved uenigheter, prøve å forstå begge parter. Samtale mellom leder og gruppe medlem, prøve å forstå begge parter, spørre en veileder om å bli mediator under en slik samtale.	Bedre kommunikasjon i fremtiden.
--------	--	--	---	-----	--	----------------------------------

Tabell 22: Teknisk risiko

Nr.	Hendelse	Konsekvens	Sannsynlighet	Konsekvensklasse	Forebyggende tiltak	Håndtering
R.T.01	Feil/forsinkelse ved bestilling.	Kjøretøyet blir ikke satt sammen på grunn av manglete komponenter. Taper tid og penger. Treghet i prosjektet.	2	II	Ha klart andre alternativer. Sjekke om det er fysiske butikker i nærområdet. Sjekke om skolen har noe vi kan få brukt.	Gå til andre alternativer. Redesigne det som er nødvendig.
R.T.02	Skade på deler	Forsinkelse, eller uferdig produkt.	2	II	Ha ekstra av enkelte utsatte deler, dobbelt sjekke styrken av deler. Utføre tester.	Bytte ut og/eller bruke ekstra deler.
R.T.03	Tap av arbeid.	Mer tid brukt på å gjenopprette mistede ressurser.	1	II	Bruke Git og Github for version control av kode. Bruke skytjenester for sikkerhetskopiering av dokumentasjon.	Gjøre arbeid på nytt og lage sikkerhetskopier.

R.T.04	Feil dimensjonering.	Ikke plass til hele systemet. Høyere kostnad og forsinkelse av sammensetning. Uferdig produkt	1	II	Sørge for at gruppe-medlemmer har god oversikt gjennom hele prosjektet. Kommunisere alle endringer med gruppa. Modellere innkjøpte deler for å sjekke at alt vil passe.	Re-design eller innkjøp av nye deler.
R.T.05	Ustabil signal.	Elektrisk system blir ikke som ønsket. Kostnad blir høyere og risiko for ødelagte deler. Fører til forsinkelse i prosjekt.	1	II	Skaffe ledninger av god kvalitet. Feil-søke og bytte der det er feil. Teste deler.	Feil-søke og reparere feil.

R.T.06	Treg responstid.	Ubehagelig kjøring som ikke er synkron med styringen. Dårlig brukeropplevelse.	1	II	Teste gjennom Matlab om overføringen/PID-kontrolleren er god nok.	Endre PID-kontroller. Mere testing.
R.T.07	Komplikasjon med rotasjonsstyring .	Ugunstig rotering som går ujevnt. Oppnår ikke ønsket styringsegenskaper. Dårlig brukeropplevelse.	2	II	Nøye forarbeid. Testing.	Omberegninger og justeringer.
R.T.08	Mye slark.	Økt slitasje og lydnivå. Nedsatt kontroll.	2	II	Fokusere på å minke slark under design. Stramme skruer godt.	Re-design og/eller etter stramme.

R.T.09	Større skade på verktøy og materiell.	Tidlig brudd og/eller utmatelse. SoilRover blir ikke komplett, svakere konstruksjon, tap av penger.	1	II	Plasser utstyr og materiell sikre steder. Informer alle om plassering om noe ligger ut-satt til. Når det gjelder beregninger og design - ha den andre fra samme disiplin se over analysen/utregningen.	Ty til alternative løsninger, eller re-design.
R.T.10	Kommunikasjon mellom Bluetooth og app fungerer ikke.	Fjernstyring fungerer ikke. Kjøretøyet blir ikke brukbart via app.	2	III	Fortløpende problemløsning på egen hånd, f.eks. på Stackoverflow. Hvis ikke finnes på nett - forhøre oss med de med mer erfaring rundt oss, f.eks. en faglærer.	

R.T.11	Git konflikter.	Mye tid brukt på å løse konflikter, kan skape forsinkelser i prosjektet. Frustrasjon.	1	I	Ikke «push» til master branch før alle har oppdatert sin versjon.	Løse konfliktene, og alle utviklere jobber på sine egne «branches»
R.T.12	Manglende kunnskap og erfaring.	En del av prosjektet blir ikke fullført.	3	III	Rådføre med en tredjepart.	Rådføre med en tredjepart.
R.T.13	Har ikke differensial ligning som er nødvendig for simulasjon.	Simulasjon blir ikke ferdig.	2	III	Prøve å finne en differensial ligning ved bruk av andre kilder.	Spørre faglærer om hjelp til å lage en simplifisert differensialligning
R.T.14	U-kompatible deler	Uferdig produkt.	1	III	Sjekke datablad og koblinger. Spørre veiledere eller faglærere om hjelp.	Finne andre alternativer.

Tabell 23: HMS risiko

Nr.	Hendelse	Konsekvens	Sannsynlighet	Konsekvensklasse	Forebyggende tiltak	Håndtering
R.H.01	Sykdom, større skader eller personligfravær av gruppemedlem/mer.	Kommunikasjon mellom gruppe-medlemmer blir vanskelig. Det kan oppstå risiko for feil eller det at prosjektet ikke blir fullstendig. Fremgang kan bli saktere.	3	II	Ikke møt opp fysisk om du er syk. Følg smitte-råd fra FHI.	Jevnlig oppdateringer mellom gruppemedlemmer. Tilrettelegge for at den som er syk kan fort bli frisk. Gjør det mulig å holde samarbeid online.
R.H.02	Personskade fra arbeid med SoilRover.	Lettere skader fra komponenter og sammensetting av SoilRover	2	I	Vær oppmerksom og forsiktig. Følg sikkerhetsprotokoller.	Følg skadeprotokoll.
R.H.03	Større personskader av verktøy.	Farlige maskinerier på labber som gir større skader.	1	III	Oppmerksom, forsiktig, sikkerhet. Bruk av verneutstyr.	Følg skadeprotokoll.

R.H.04	SoilRover kjører på noen.	Lettere skader. Dårlig fremvisning av produkt.	1	II	Bruke distanse sensor for å forebygge kollisjoner. Passe på klar bane. Avrunde skarpe kanter.	Reparere eventuelle skader. Passe på klar bane under kjøring.
R.H.05	Konflikt i gruppe.	Kan føre til dårlig kommunikasjon, arbeidsmiljø og moral. Gruppedlemmer bruker tid for å komme frem til enighet, noe som kan skape forsinkelser i prosjektet.	2	II	Skape et åpent og trygt miljø, der det er åpent for å diskutere og si sine meninger og følelser.	Samtaler, utveksle meninger ved uenigheter, prøve å forstå begge parter. Bruke en tredjepart som megler.

R.H.06	Brudd på grupperegler/gruppekontrakt.	Frustrasjon i gruppen, dårlig tidsbruk.	1	II	Samtaler mellom leder og gruppemedlemmer, prøve å forstå begge parter når konflikter oppstår. Trekke konklusjoner. Gruppedlem kan bli utvist fra gruppa.	Ta opp problemet med personen det gjelder og finn løsninger sammen.
--------	---------------------------------------	---	---	----	--	---

Spesifikasjoner

Tabell 24: Dokument krav

Krav ID	Krav	Prioriteringsgrad	Krav beskrivelse	Hvordan teste
D.01	Dokumentasjonen skal inneholde all relevant informasjon av prosjektet.	A	For at SoilSteam får fullstendig dokumentasjon og kan videreutvikle produktet. Og at gruppemedlemmer kan få riktig karakter for arbeidet gjort.	Lese igjennom dokumentasjonen og via tilbakemeldinger.
D.02	Det skal lages en bruksanvisning til hvordan man bruker kjøretøyet.	B	Grunnet flere funksjoner og brukervennlighet er det behov for bruksanvisning.	Ha en tredjepart SoilRover ved hjelp av bruksanvisningen.

Tabell 25: Komponenter krav

Krav ID	Krav	Prioriteringsgrad	Krav beskrivelse	Hvordan teste
K.01	Batteritid skal vare en messedag med tiltenkt bytte av batterier.	A	SoilRover skal brukes på messer som varer over flere timer.	En rekke forskjellige bruks tester som gir svar om hvor lenge ett batteri varer. EXPO.
K.02	Batteriet skal være oppladbart.	A	Miljøvennlig og billig i lengden.	Koble det til lader og se om spenningen øker.
K.03	Hjulposisjon skal kunne måles internt i SoilRover.	B	Hjulposisjon trengs for nøyaktig styring.	Avlese fra app og kryssreferer ved ekstern måling.
K.04	Kjøretøyet skal klare å kjøre 15 min med en fart på 5 km/t i en rett linje.	B	For å ha en tilstrekkelig batteritid for forventet bruk	Kjøre SoilRover i en rett linje så lenge den varer.
K.05	Rotasjon skal ha sensitivitet på 1 grad.	B	For mer nøyaktig og større område for styring.	Visuell inspeksjon og eventuelt måleapparat.

Tabell 26: Datasystemer krav

Krav ID	Krav	Prioriteringsgrad	Krav beskrivelse	Hvordan teste
DA.01	SoilRover skal styres via mobilapplikasjon.	A	Informativt styringsalternativ for SoilRover.	Test kjøre.

DA.02	Simulasjonen skal være stabil.	A	Simulasjon skal kunne operere uten forstyrrelser, dvs. oppførselen til systemet skal være som forutsett. Systemvariabler skal ikke endres, med mindre brukeren ønsker å gjøre det.	Kjøre simulasjon på to forskjellige datamaskiner (forskjellige med tanke på ytelse) for å verifisere stabilitet.
DA.03	Masser og treghetsmomenter sammenlignbare med ekte maskin.	A	Masser og treghetsmomenter av den reelle SR skal overføres til den simulerte SR.	Visuell inspeksjon
DA.04	Hjul og armer roterbare og kontrollerbare fra software. Ta hensyn til deres dynamikk.	A	Den simulerte SoilRoverens hjul og armer skal oppføre seg mest mulig lignende som i den reelle SR.	Visuell inspeksjon.

DA.05	Verden skal kunne populeres, dvs. andre ting kan legges inn til bakkeplan.	A	Det skal være mulig å velge forskjellige objekter, som blir bestemt av bachelorgruppe i forkant, og legge de til bakkeplan på et valgt sted i simulasjonsverden.	Legge til et objekt, visuell inspeksjon.
DA.06	Det skal være mulig å teste forskjellige styringsoppsett	A	Det skal være mulig å aktivere styringsoppsett som ble definert i avsnitt 6.1.3.	Visuell inspeksjon av spesifikke styringsoppsett
DA.07	Det skal være mulig å teste forskjellige kjøremønstre for maskin.	A	SR skal kunne styres i ønskede mønstre.	Prøvekjøre SR i enønskede mønster. Visuell inspeksjon.
DA.08	Visualisere andel av terrenget som ble dekket av SR.	B	Terrenget som SR kjørte på skal markeres med mørkere farge.	Visuell inspeksjon etter å ha kjørt en del.

DA.09	Mulighet å koble en ekstern kontroller.	B	Det skal lages en API for å kunne koble en ekstern kontroller (for eksempel PS4 kontroller), for å kunne styre simulasjonen på samme måte som med tastatur.	Koble til en kontroller og sjekke om den fungerer.
DA.10	Sette nøyaktig posisjon av dekkene via mobilapplikasjon.	B	Mulighet for større presisjon via kontroll fra telefon applikasjonen.	Visuelt sjekke om hjulene står slik de er oppsatt i appen.
DA.11	Det skal visualiseres damping funksjon som kan aktiveres ved et tastetrykk, når SR kjører på en åker.	C	Damping funksjon skal kunne slåss på for å imitere dampproduksjon når SR kjører på en åker.	Visuell inspeksjon.

Tabell 27: Fysiske krav

Krav ID	Krav	Prioriteringsgrad	Krav beskrivelse	Hvordan teste
F.01	Skal kunne fjernstyres ved hjelp av en kontroller.	A	En brukervennlig måte å styre kjøretøyet.	Test kjøre.

F.02	Vekten av SoilRover skal ikke overskride 20 kg.	A	Arbeidsmiljøloven (Jan 2022) tilsier at ansatte ikke skal bære mer enn 20 kg.	Kjøretøyet veies.
F.03	Armene til hjulopphenget skal kunne svinges 75 grader.	A	Bevegelse av armene gir stabilitet under kjøring og mulighet til lastebilmodus.	Via målinger
F.04	Et vippeledd mellom framhjulene som gjør at kroppen ligger plant med bakken mens hjulene har en høydeforskjell på 10mm.	A	Bedre stabilitet og grep i ujevnt terreng.	Plassere noe under et av hjulet og ser om begge hjulene fortsatt er plantet i bakken og om kroppen er i vater.
F.05	Elektronikk skal tildekkes.	A	For å beskytte elektronikk fra skade og fukt.	Visuell inspeksjon
F.06	Det skal ikke være løse deler i SoilRover.	A	Unngå skader og frakoblinger som fører til defekter.	Utføre fysiske tester som sjekker om noe er løst.
F.07	Tilstandsinformasjon fra SoilRover vises på mobilapplikasjonen.	A	Ha kontroll over tilstandene til SoilRover.	Avlese fra app og kryssreferer ved ekstern måling.

F.08	SoilRover skal styres via mobilapplikasjon.	A	Informativt styringsalternativ for SoilRover.	Test kjøre.
F.09	Hjulene skal kunne drives i koordinasjon med hverandre.	A	Kjøretøyet kjøres behagelig og kjører jevnt i den retning som er satt.	Visuell inspeksjon og kontrollsjekk.
F.10	Hjulene skal kunne rotere 360* om sin egen vertikale akse individuelt.	A	Gir oss mer mulighet for styring i den retningen vi ønsker.	Visuell kontrollsjekk, avlese i app. Et punkt blir markert over rotasjonsstedet på hjulmodulen og et rett under. Så roteres hjulet og hvis det underste punktet roteres helt tilbake til sin originale plass er det godkjent.
F.11	Dimensjonene skal være nesten nøyaktig nedskalert 1:10 (original 9m x 3m x 2m (LxBxH))	A	Passe størrelsesmessig til å brukes på messer samt transporteres dit i bagasjerom.	Utføre målinger

F.12	Hjulene skal kunne heises til de står i friluft.	A	For å minimere slitasje på hjulmodul når den utfører større rotasjoner.	Visuell inspeksjon.
F.13	Jevn vektfordeling for hjulene.	B	For enklere utregninger, mindre slitasje og bedre styring.	Vekt inspisering per hjul.
F.14	Hjulene skal senkes slik at kroppen har minimum 40 mm klarering til bakken	B	Med tanke på bruk på messer må den kunne kjøre over ledninger og lignende uten skade.	Vi måler med linjal mellom bunnen av ramma og bakken/gulvet.
F.15	Styringssystemet skal ha en responstid på under 1 s.	B	Unngå forsinkelse fra styring til faktisk utførsel.	Ved simulering, målinger og tidtaking
F.16	Oppnå 5 km/t toppfart.	B	Kan få en behagelig rask styrings fart, men også ikke være så rask at man mister kontroll.	Test kjøre og beregne.
F.17	Klem og strømfare områder skal skjermes.	B	Personskader skal unngås og deler skal beskyttes.	Visuell og fysisk inspeksjon.
F.18	Komponenter kan enkelt skiftes ved behov.	C	Enkelt bytte ved slitasje eller feil. Spare tid og penger.	Ta fra komponenter og sette sammen igjen.

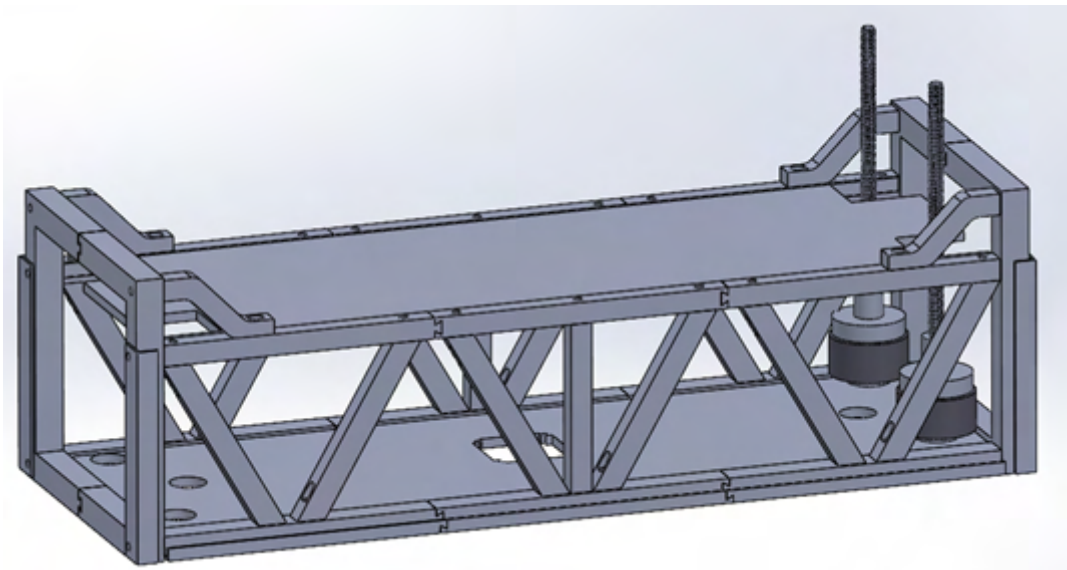
F.19	Kan kjøre over IP44 ledning.	C	Siden kjøretøyet skal kunne brukes på messer og ofte er det plassert ut ledningsbeskyttere der. Derfor er det ønskelig at den skal kunne kjøre over mindre hindringer.	Testkjøring over IP44 ledning.
------	------------------------------	---	--	--------------------------------

B Kropp

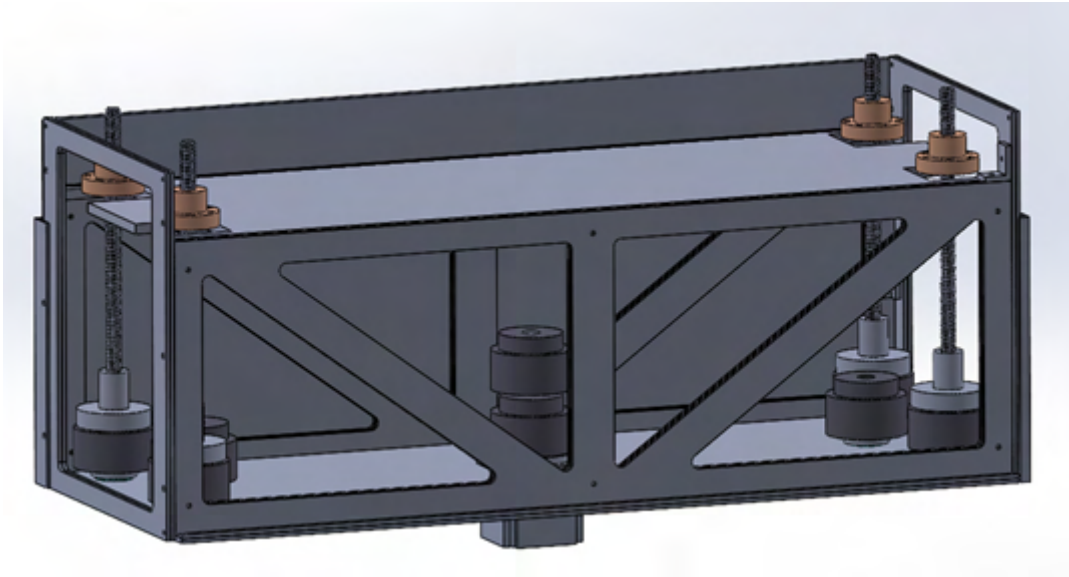
Kroppen er samlebetegnelsen gruppen har valgt å gå for når vi snakker om dekkelet, rammeverket og heisemekanismen. I rammeverket har vi to etasjer, nedre del er for heisemekanismen mens øvre etasje er for alt det elektriske.

Design av rammeverket

Etter andre presentasjon så fikk vi tilbakemelding om vi kunne utfordre oss med at istedenfor å ha en boks, så laget vi et rammeverk som kunne være litt mere representativt for SSI sitt design. Etter å ha sett litt på deres design sånn det var på det tidspunktet (fredag 8.april). Personen ansvarlig for boksen designet ett rammeverk som var tiltenkt å lage i 3D-printer. Etter en samtale med faglærer (Richard Thue) og ansvarlig for 3D-printere på USN, så fikk gruppen tilbakemelding om at om dette designet skulle printes, ville det ikke bare ta tid, men designet ville ikke være like sterkt som om designet ble laget med limte kryssfiner plater, se fig. 83.



Figur 83: Rammeverk V1.

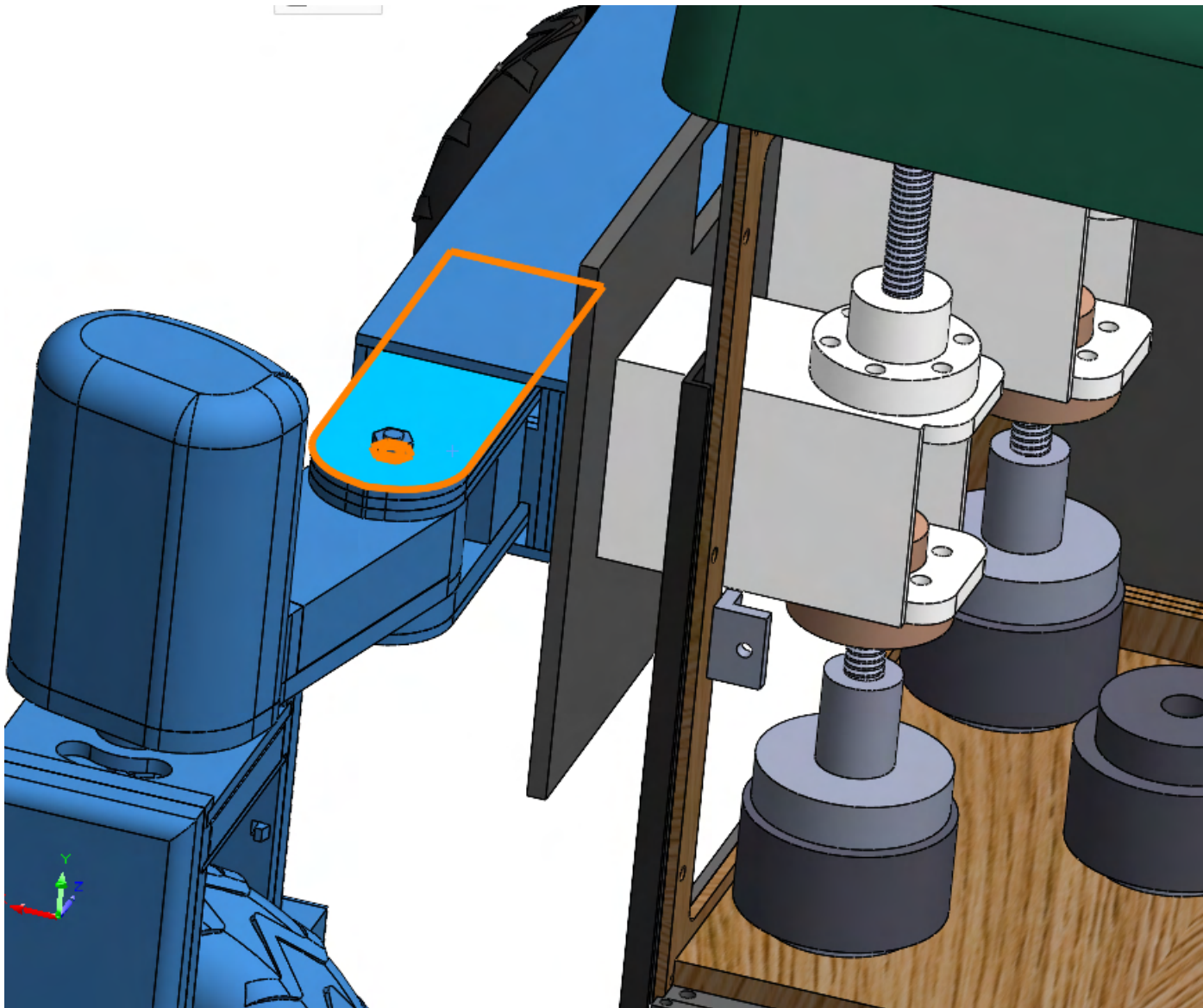


Figur 84: Rammeverk V2.

Etter å ha endret på designet slik at det vil passe for å kunne laserkuttet, se fig. [84](#), så begynte vi med å teste ut designet ved å laserkutte ut designet via en epilog laserkutter på USN. Etter at alt hadde blitt limt, skrudd sammen og skrudd på 3D printede vinkler, for å støtte veggene, og for å hindre at veggene heller utover eller innover. Alle skrue hullene ble forboret med en drill for at treverket ikke skulle sprekke opp når man skrur i det.

Når alt var limt og skrudd sammen så testet vi med hvordan delene for heisemekanismen passet i det nederste laget. Det måtte pusses litt til for at hjullagrene skulle passe nok at de var flush med platen, men at du ikke ville falle gjennom. Når de ble satt inn så fikk vi sett hvordan det ville bli med beltene og trinsene. Vi så at hvor det midterste hjullageret ble plassert så ville belte vi hadde bestilt på 1040mm i lengde bli for langt. Så vi boret opp et nytt hull etter å ha målt for å lage nye hull for å kunne stramme til belte. Vi måtte være sikre på at belte ikke var for slakt, er det ikke stramt nok vil belte ikke få godt nok tak på trinsen og vil ikke få heisemekanismen til å virke. Når rammeverket ble ferdig så startet vi på heiseplaten som skulle ha skulderen som fester seg til hjulopphenget. Her ble det modullert og 3D printet et feste som skulle festes til heiseveggen og gå på mutteren som beveger seg opp og ned på heiseskruen se fig. [85](#). Det

markerte er skulderen som er festet til heiseveggen og til armmodul.



Figur 85: Bakre skulder festet til heisevegg.

Komposittdeler

Sideveggene som skal kunne enkelt fjernes var originalt tenkt å bli laget i glassfiber kompositt. Oppsettet for disse platene ble det brukt 4-6 lag med CSM glassfiber matter, i midten av disse

lagene ble det laget en 5mm tykk PVC plate som kjerne. Forsøk 3 og 4 hadde PVC platen fått boret hull for bedre gjennomstrømming av epoksy. Epoksyen som ble brukt var svapox 110 med herder TL-1. Epoksyen og herderen er blandet til 24% herder og 76% epoksy. Denne blandingen har et «potlife» på 80 til 90 minutter. Dette vil si at etter 80 til 90 minutter så vil blandingen begynne herdeprosessen, dette er ut fra datablad for disse materialene [21]. Så med denne tiden så får man god tid til å jobbe inn epoksyen inn i glassfiber mattene. Etter fire forsøk på forskjellige tider, kom testene ikke ut som ønsket. Ut fra resultatene for disse testene så ble det konkludert med at slippmiddelet bruk kan ha blitt forurenset, noe som har ført til at glassfiber platen har satt seg fast til glass platen. Designet for rammeverket tillater å kunne bytte ut både bunnplater og sideplatene så det er enkelt å sette inn plater så lenge de er i riktig dimensjoner, noe som er gjør det lettere for vidreføring av prosjektet om det er ønskelig. Dette kan ses på fig. [86], fig. [87], fig. [88].



Figur 86: Glassfiber plate test 1(venstre) og 2 (høyre).

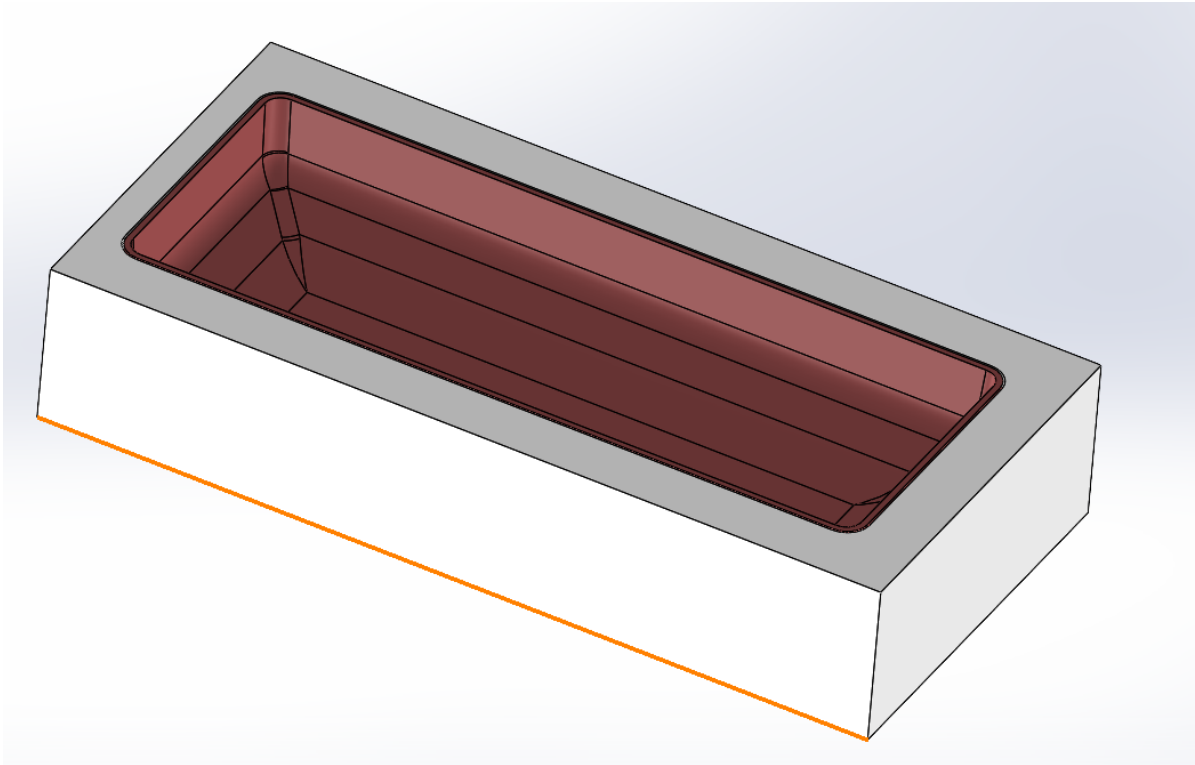


Figur 87: Test 4 tatt av glassplaten.



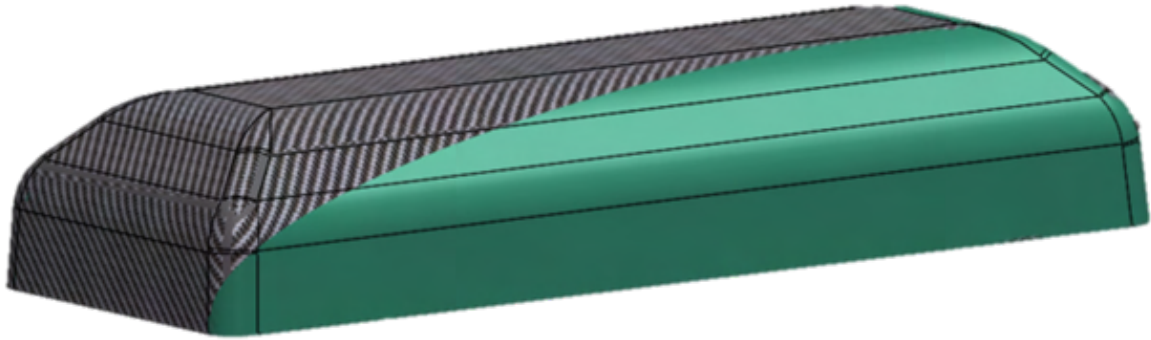
Figur 88: Test 4 tatt av vedsiden av glassplaten.

En av utfordringene for dette prosjektet er også å lage dekselet i karbonfiber for så å kunne lakkere den med forespørt farge fra SSI. Det ble laget en 3D modell av dekselet for å kunne lage så en form som kunne 3D printes. Dette kan ses på fig .[\[89\]](#).



Figur 89: Dekselform og deksel.

Når modellen var klar så ble den delt opp for å kunne 3D printes. Gruppen lånte printere fra USN som hadde mulighet til å printe deler i maksimal størrelse 150mm x 150mm x 150mm. Da dekslet delt opp i 3 deler med flenser som kunne limes på. Denne formen ble laget for å kunne lage dekslet ut av karbonfiber kompositt. Valget for karbonfiber er etter som at dette produktet er tenkt å kunne brukes til visning på messer, noe som er kjent for å gripe oppmerksomheten til folk på messer er karbonfiber. Designet som gruppen ble enig med SSI om kan ses på fig.[90].

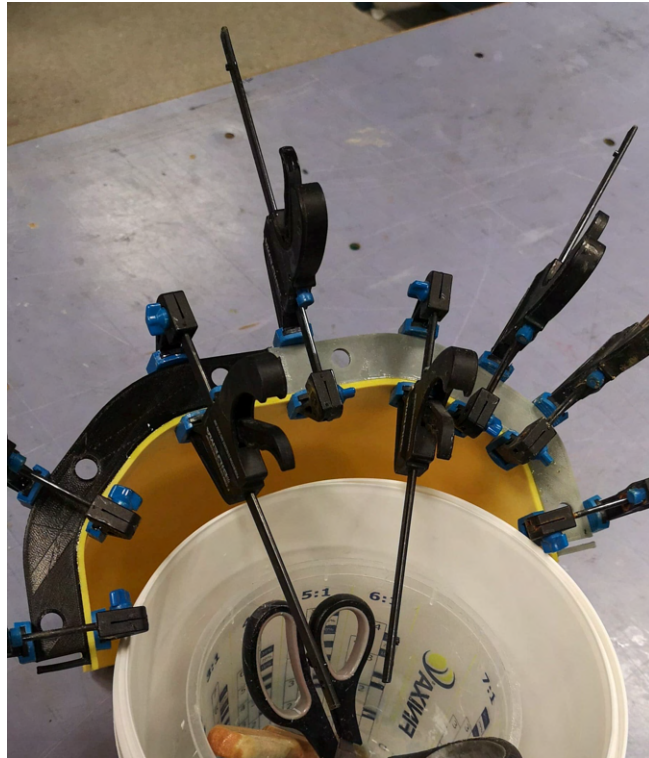


Figur 90: Design for deksel.

Vi startet med å pusse med sandpapir for å få en jevn overflate på de 3D printede delene. Dette er for at når innsiden av formen skal lakkkes så blir det en så jevn og fin overflate som mulig. Dette er fordi den flaten som lakkkes er den siden som vil bli utsiden av dekslet, så for et godt resultat er det viktig at overflaten er bra. De pussede delene kan ses på fig. [91]. Flensene ble limt på formene med superlim og holdt sammen med klemmer se fig. [92].



Figur 91: Pusset og ikke pusset 3D printede endedeler for dekslet. Venstre del på bildet er pusset og høyre del er ikke pusset.

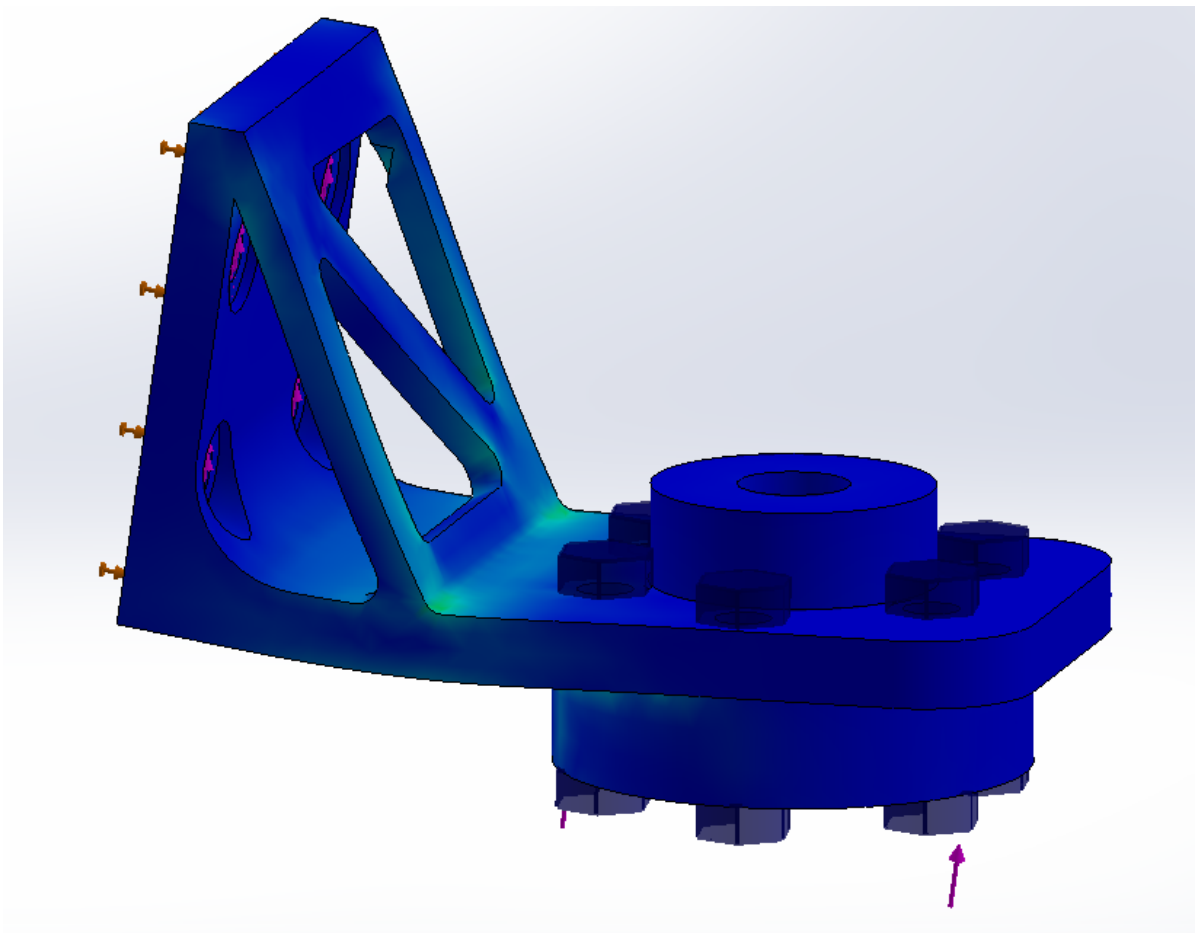


Figur 92: Endeform får flensen limt på og holdt sammen med klemmer.

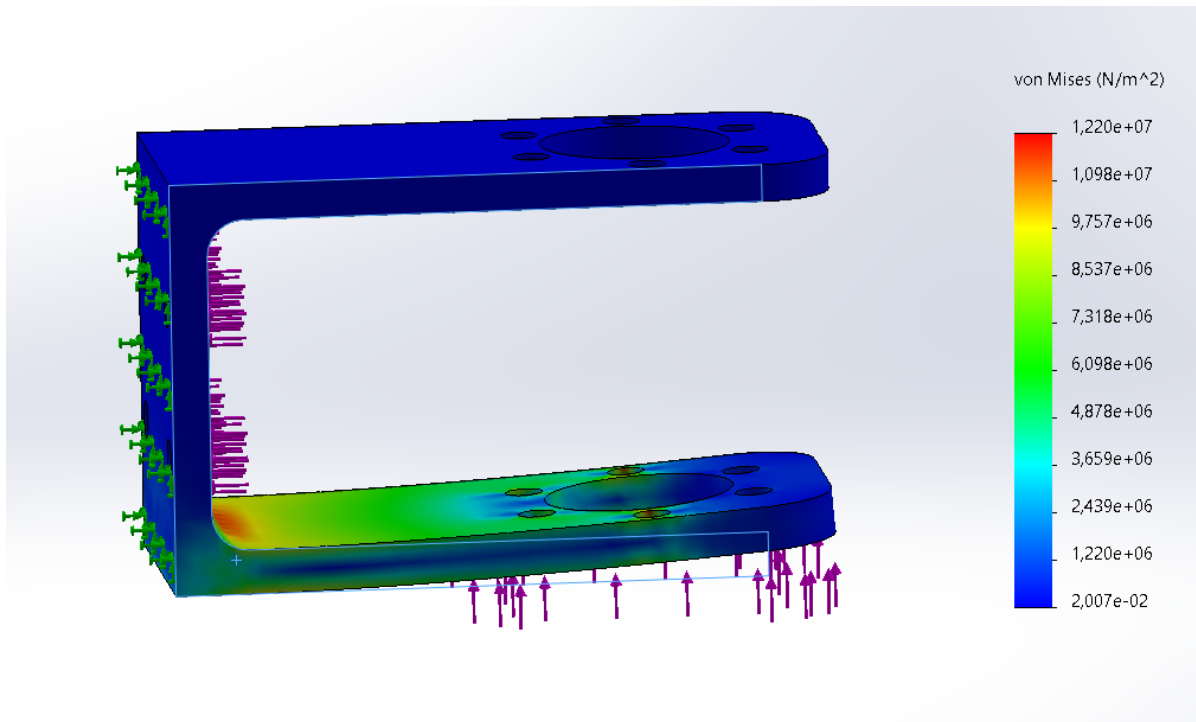
Når alle flenser blir limt fast vil formen settes sammen med en kombinasjon av skruer og super lim for å holde delene fast og unngå at det vil bli noen sprekker mellom delene. Når alt blir satt sammen, så vil innsiden bli dekket med en sprøyte sparkel. Når dette er blitt gjort og den ønskede overflaten er oppnådd så kan vi starte med å legge et slippmiddel så vi kan få støpen ut fra formen når den er ferdig. Etter at flere lag med slippmiddel er lagt på og alle flater er dekket med slippmiddel, så kan vi starte med støpen. Her vil vi legge et tynt lag med epoxy først for at karbon fiberet har noe å feste seg til og legger seg slik vi ønsker det. Når første lag er lagt på og er smørt godt inn med epoxy så kan de neste lagene legges på. Med jevnlig bruk av rulle med en metall rulle for å få vekk fangede luftbobler i mellom lagene. når alle lagene er lagt så vil støpen settes i vakuum for å bli kvitt siste rest med luft og for at alle algene skal kunne få nok trykk til å knytte seg sammen mens epoxyen herder. Når epoxyen har herdet ferdig kan delen separeres fra formen. når alt av kanter er pusset ned så vil vi starte med en overflate behandling. Dette blir bare en polering for å få en fin og blank overflate.

FEM analyser

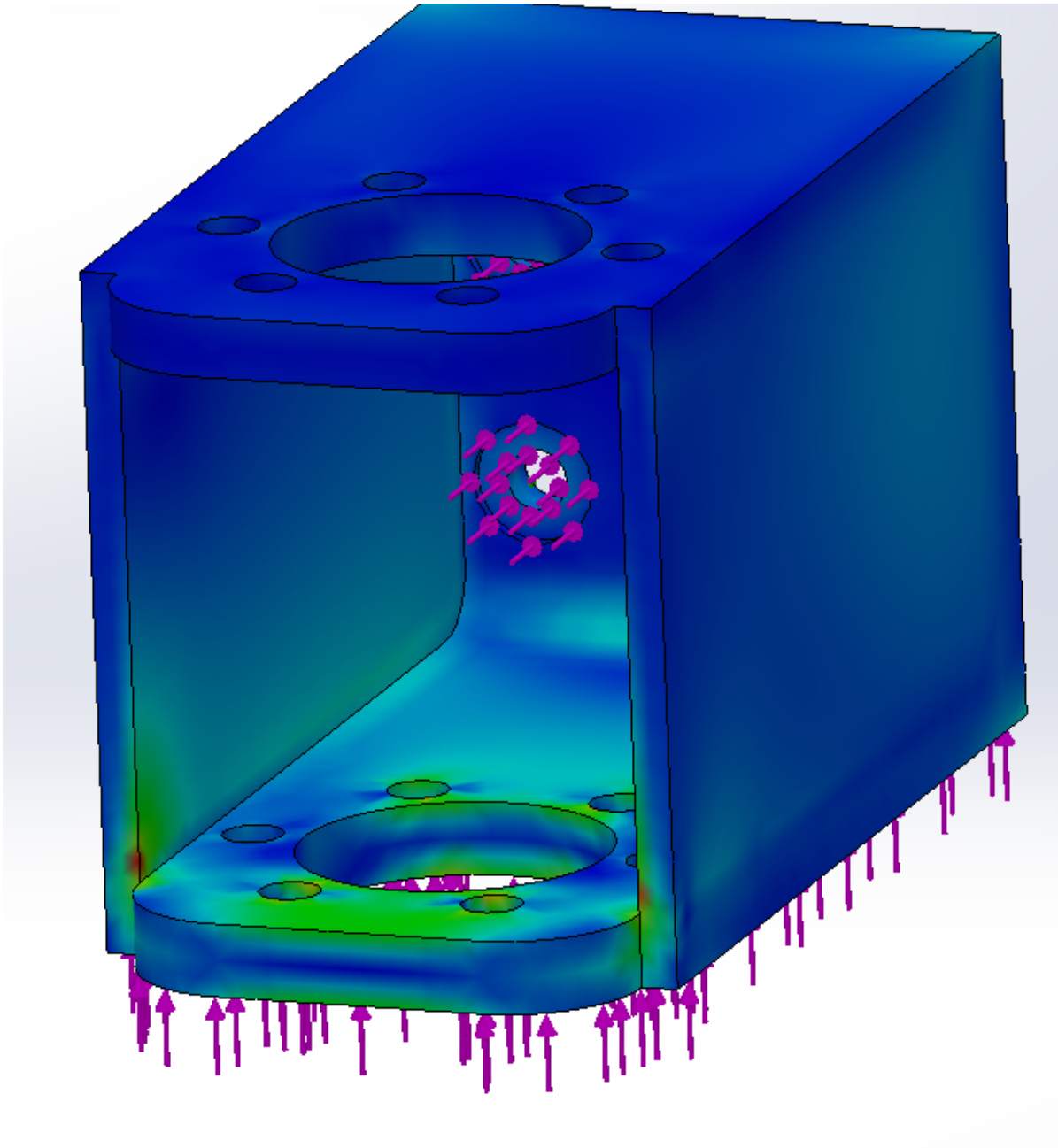
Ved enkelte deler så var det viktig å ta analyser for å kunne garantere at styrken på delen var god og at delene ikke deformerte seg i utsatte punkter. En av delene som var mest utsatt for utmatting og mye krefter er festet til heiseveggen. Den ble tatt under en analyse for å se hvordan deformasjonen er og om den vil greie å holde ut under de påførte kreftene. Delen gikk gjennom noen optimaliseringer med litt veiledning fra veileder fra SSI (Jo Myroldhaug). Vi så ut i fra analysene at versjon 1 og 2 ville være for svake, se fig. 93 og fig. 94. Etter å ha forhørt med veileder kom vi frem til en endring som ble da versjon 3, se fig. 95, dette designet ut i fra analysen og en høy sikkerhets faktor viser at det skal holde.



Figur 93: Første versjon av heisevegg festet.



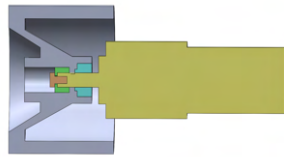
Figur 94: Heisevegg feste versjon 2.



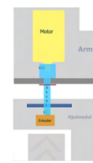
Figur 95: Heisevegg festet versjon 3 og det endelige designet.

For alle analysene ble det satt på loads som skal simulere de kreftene delen ville hvert under i virkeligheten og kontakter som skal kunne simulere friksjon. Dette gjør simulasjonen så nøyaktig som mulig og viser et så nøyaktig bilde av hvordan delen vil deformere seg under disse kreftene.

C Armmodul dokumentasjon



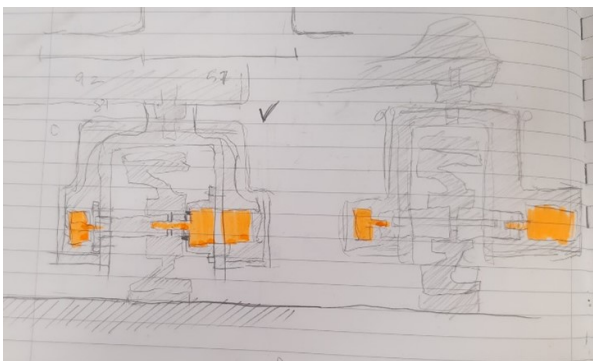
(a) Konsept til drivmotor feste til felg



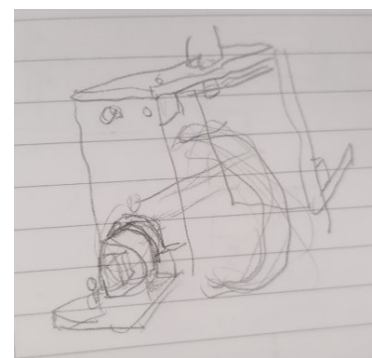
(b) Konsept til rotasjon

Figur 96: Konsept fra før presentasjon 2

Figur 96a er konseptene vi satt med etter 3. presentasjon. Før presentasjon 3 var det ikke klart at vi trengte enkoder til alle hjulmodulene, da det ble klart så vi at med konseptet over var det ikke mulig å legge til en enkoder. Det er spørsmål om vi hadde gått for det konseptet uansett siden det er vanskelig å bore i en aksel som kan rotere. Vi kom med to nye konsepter som du kan se på fig. 97. Det som er farget oransje er motorene og enkoderene. I konseptet til venstre i fig. 97 festes motoren direkte til felgen og bærer vekt mens enkoderen festes slik at den ikke bærer noe vekt. Tegningen er basert på en mindre enkoder enn den vi endte opp med. I den andre tegningen bærer verken enkoderen eller drivmotoren vekten, men en metallstang. Dette er bedre for levetiden til akselen til drivmotoren og enkoderen, men det tar mer plass. Siden den kan bli brukt som en showmodell og testmodell så er det estetiske en større prioritet enn levetid. Derfor valgte vi å gå for det første konseptet.

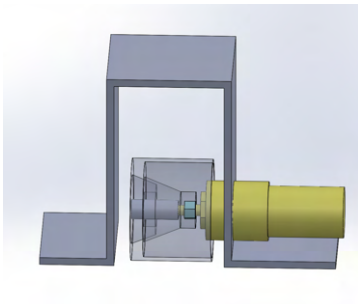


Figur 97: Armmodul

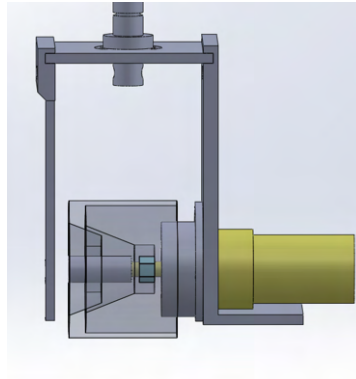


Figur 98: Konsept til hjulmodul

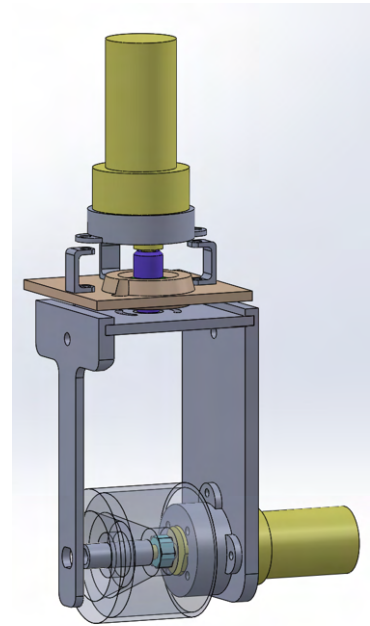
Fra fig. 99 til fig. 104 kan du se utviklingen av 3D-modellen av driv og så armmodul. Med den nye større enkoderen valgte vi å feste den delvis direkte til felgen, og la den bære vekt.



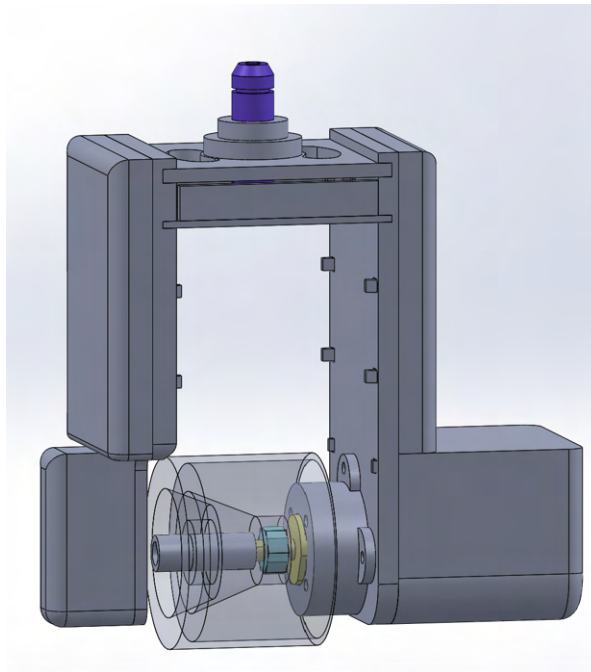
Figur 99: Hjulmodul ver.1



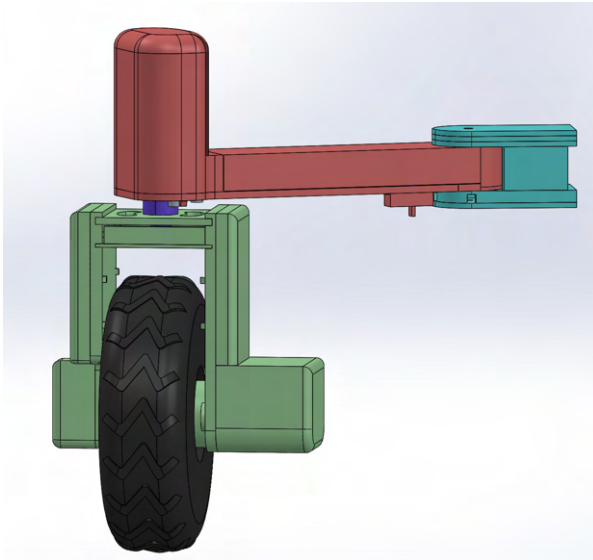
Figur 100: Hjulmodul ver.2



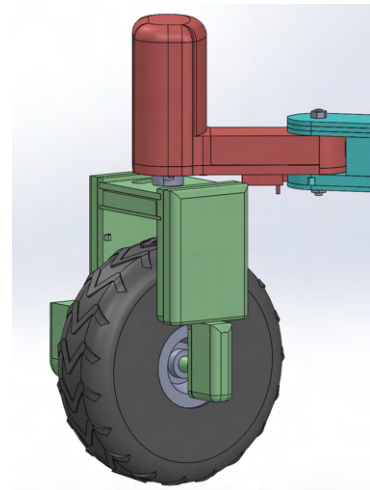
Figur 101: Hjulmodul ver.3



Figur 102: Hjulmodul ver. 4



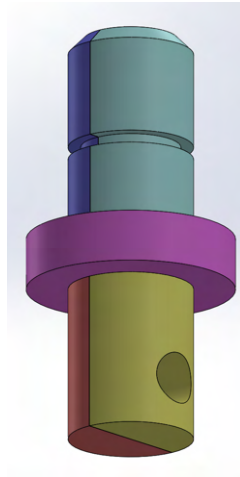
Figur 103: Armmodul ver.1



Figur 104: Armmodul ver.2

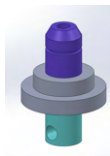
Mange av forandringene er ikke synlige, som å gjøre rammeverket til drivet høyere og andre justeringer av dimensjoner og justeringer av hullstørrelser ettersom vi printet de fysisk. Armen har blitt kortere som du kan se fra fig. [99](#) og fig. [104](#).

Akselen har gått gjennom mange iterasjoner. Det er en liten del som skal kunne bære $\frac{1}{4}$ av vekten til hele kjøretøyet. Første versjon kan du se på fig. [105](#). Når vi printet den ut fant vi at å lime sammen så mange deler og få det nøyaktig var for vanskelig så vi utviklet et nytt konsept. Alle akslene trenges å pusses for å passe gjennom alle hull, og borres opp så at metall stangen og motorakselen kan trenge gjennom. Dette var en tidskrevende prosess til tider.

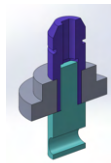


Figur 105: Aksel ver. 1

Andre versjon (fig. 106a) var den versjonen som hadde best toleranser. Dessverre hadde den store svakheter. Krefte den ble utsatt for er i en retning som er minst gunstig i forhold til printerretningen. Flere av akslene knakk under sammenkobling av armmodulen, og når armmodulen ble festet sammen til en boks.



(a) 3D-modell



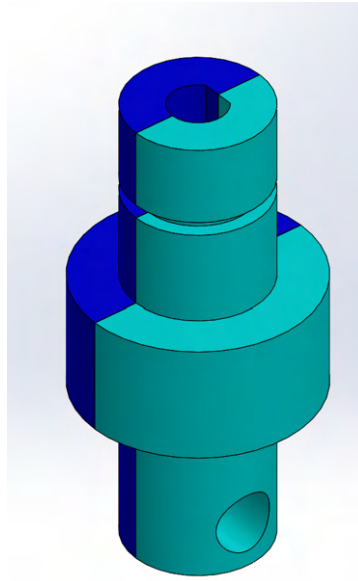
(b) Seksjonsbilde



(c) knekt 3D-print

Figur 106: Aksel ver. 3

På fig. 107 er det avbildet versjonen vi valgte å gå for. Den blir printet i to deler og så limt sammen. Det førte til litt skjevheter i noen av delene, men de var fortsatt godt fungerende



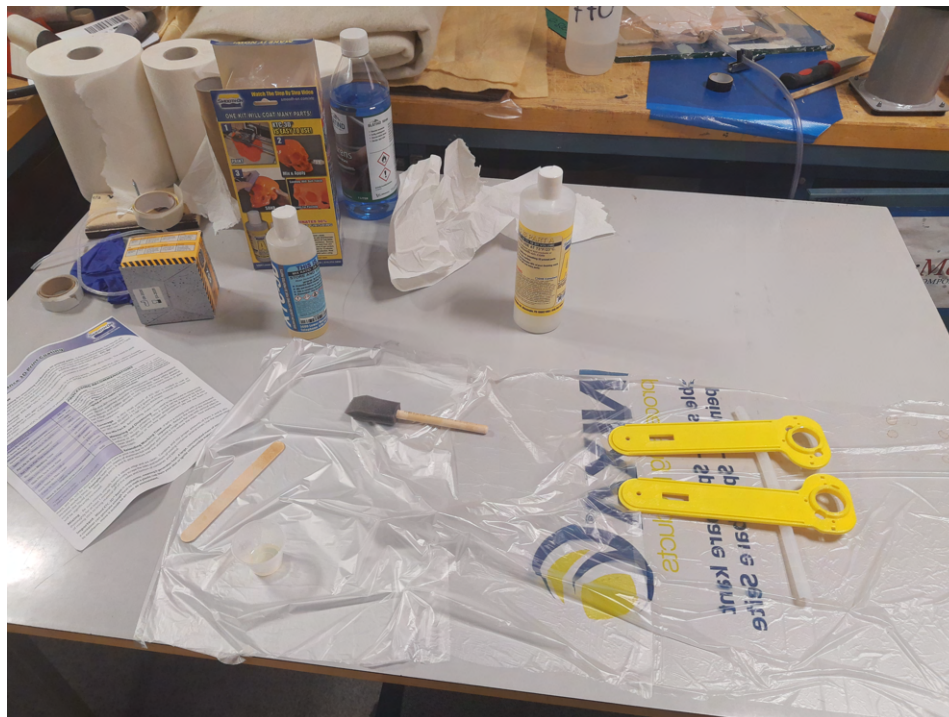
Figur 107: Siste versjon av aksel

Fysiske prototyper

tilpassing av 3D-print

En del tid har blitt brukt til å tilpasse 3D-print deler. 3D-print deler er gjerne (i vårt tilfelle alltid) mer massiv enn 3D-modellen. Vegger blir tykkere, hull blir mindre og lignende. Denne størrelsesforskjellen varierer med printer, innstillinger, filament og romtemperatur. Siden vi bruker forskjellige printere har det ikke vært mulig for oss å kompensere helt for de forstørrelsene ved å forandre på dimensjoner i 3D-modellen, men vi har tatt en forhåndsregel å gjøre alle indre diameter minst 0.3 mm større. Hvor stort man trenger å øke indre diameter varierer med modellene. Det er bedre å ha hullene for små siden det er lett og raskt å bore opp, andre geometriske former derimot må pusses (om det er mulighet for det), dette er betydelig mer tidskrevende. D-formet hullet i akselen måtte bli boret opp, men siden det var essensielt å beholde formen var dette også en tidskrevende prosess. Hvis deler er blitt printet og du har et hull som

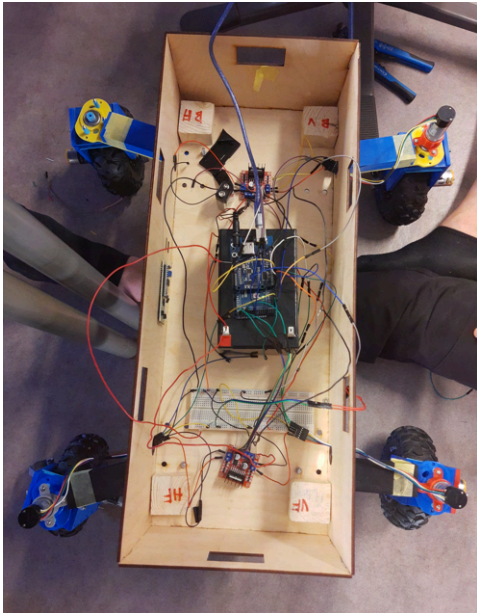
er for stort betyr det ikke nødvendigvis at du må printe å nytt. Dette skjedde oss med hullet til kulelageret på armbunnen. Vi la over et tykt lag med epoxy. Epoxyen var egentlig ment som en finish til dekslet. Vi fulgte HMS instruksene nøye og hadde på oss hansker, vernebriller, gassmaske og dekte til armene. Forsøket var en suksess.



Figur 108: Armbunn ligger til tørk

Armmodul

På fig. [109](#) kan du se fire prototyper av armmodulen. De er festet til boksen ved hjelp av en gjenget skrue. Boksen er en prototype og ble ikke bygget for armmodulene, derfor er armmodulene festet slik som de er. Festet er ustødig og armene kan rotere, men det gir oss fortsatt en ide om hvor stødig hjulmodulene og leddet mellom hjulmodulene og armen er. De var ikke like stødig som ønsket, så vi limte akselen og toppflaten til rammeverket. Prototypen under har lengre armer enn det endelige produktet vil ha. Dekslet er teipet til armen for bedre støtte.

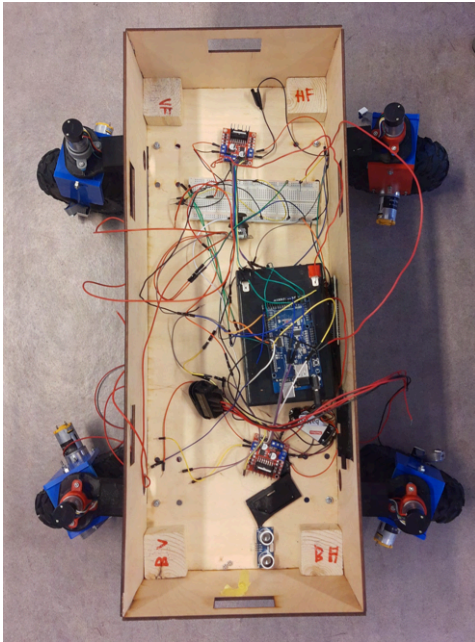


Figur 109: Prototype av boks med alle armmoduler

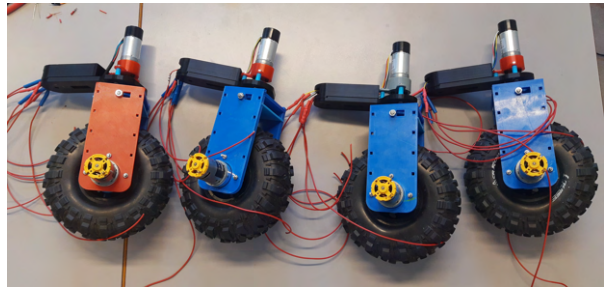


Figur 110: Første hele prototype av armmodul

På fig. [113](#) kan du se boksen med nye armer og med støtteklossene i boksen flyttet for bedre stabilitet av boksen. Siste versjon av armmodulen kan du se på fig. [112](#).



Figur 111: Siste versjon av armmodulene



Figur 112: Prototype av boks med alle armmodulene

For å feste kulelageret til armbunnen var det først tenkt til å klemme den mellom to deler (se fig. [114](#)), men vi fant under sammensetning at kulelageret satt godt nok med kun tight-fit.



Figur 113: Armbunn med kulelager



Figur 114: Feste til kulelager

Diverse

Vi printet ut en grov replika av felgen så flere kunne jobbe med hjul samtidig.



Figur 115: 3D-printet replika av felg

D Elektrisk kapasitet

Grovt start estimat av batteri nødvendighet

Ved valg av batteri er vi nødt til å velge et batteri som har nok spenning for kjøretøyet og det er nødt til å ha nok batterikapasitet til å kjøre kjøretøyet vårt i 15 minutter på full kraft. Altså $\frac{1}{4}$ av en time. Vi hadde tiltenkt 12 motorer, 8 enkodere, en mikrokontroller og andre mindre enheter. Antar at hver motor trekker mellom 1 A og 1,5 A, enkodere trekker 60-100 mA hver, og vi tar en antagelse på 2 A til for verste tilfelle scenario på andre elektriske ting. Da får vi mengde strøm systemet trekker på 1 time

Dette var før vi satt sammen flere deler av heisemekanismen til å kun klare seg med en motor for å heve begge sider.

$$12Motorer + 8Enkodere + 2A, \quad (17)$$

der vi setter inn tiltenkt worst case, så det blir

$$12 \cdot 1.5A + 8 \cdot 0.1A + 2A = 15A + 0.8A + 2A = 20.8A.$$

For at vi da skal kunne kjøre det i 15 minutter blir nødvendig batterikapasitet

$$Batterikapasitet = \frac{1}{4} \cdot 20.8A \approx 5.25Ah$$

der vi har runda opp for sikkerhet.

Dette var et grovt estimat som var fra starten av oppgaven da vi hovedsak visste kun om motorene og enkoderene og vi tenkte at det skulle være en motor for hvert hjørne til heis også. Valg av resten av det elektriske var ikke bestemt og derfor antatt til å være forholdsvis høy. Disse antagelsene ble gjort siden det er bedre å ha noe med for høy kapasitet istedenfor for lav. Det ble ikke handlet noe batteri på så tidlig antagelser.

Batteri

Vi har sett fra krav i tabell(25) at vi ønsker at Roveren vår skal kunne være operativ i vertfall 15 minutter. Ut fra dette ser vi til hvor mye strøm systemet vårt trekker og valgte batteri med gitt batterikapasitet ut ifra dette.

Første detaljerte antagelser av elektrisk kapasitet

Dette er fra tidlig i fasen av prosjektet når vi hadde gått til å bruke DC-motor istedet for stepper for rotasjonen , et annet oppsett med mindre antall sensorer, men at vi skulle plassere sensorene på små servomotorer som bevegede seg kontinuerlig.

Arduino Mega 2560 R3

$$0,2A$$

Rotasjonsmotorer

$$4 \cdot 580mA = 2,32A$$

DrivMotorer

$$4 \cdot 990mA = 3,960A \approx 4A$$

Forholdsvis stor Heisemotor

$$2.8A$$

Driv enkodere, 4 stykker rs sine

$$4 \cdot \text{MaxSupplyCurrent} = 4 \cdot 28mA = 122mA$$

Eller driv enkodere fra Mouser, fortsatt 4 stykk

$$4 \cdot \text{MaxSupplyCurrent} = 4 \cdot 5mA = 20mA$$

Ledstripe per meter

$$1,8A$$

Denne verdien viste seg seinere og være feil, da det var strømforbruket fra $\frac{1}{2}$ m.

Sensorer for avstand 4 eller 6

$$15mA \cdot 4 = 60mA \text{ eller } 15mA \cdot 6 = 90mA$$

Skulle hatt motorer som sørget for større område dekket av sensorer, men grunnet høyt strømforbruk så er det bedre med 6 sensorer for å dekke.

Micro servo 9G 4 stykker der vi medregner worst case

$$650mA \pm 80 \cdot 4 \rightarrow 730mA \cdot 4 = 2,920A$$

Både pin-mangel og strømforbruk gjorde at dette ble forkastet, og vi heller gikk for 6 sensorer.

Den tidligste utregningen førte da til at vi fikk

$$0,2A + 2,32A + 4A + 2,8A + 0,002A + 1,8A + 0,090A = 11,212A.$$

Dette konsulterte vi om, og det fremsto som en rimelig verdi for et slikt system, før vi gikk til å bestemme batteri.

Siste antagelser av elektrisk behov

Da vi imot slutten har fastsatt det som skal bli tatt i bruk i prosjektet har vi nå et mer konkret svar på hvor mye strøm som kommer til å være nødvendig og hvor lenge systemet vil klare å holde seg i drift.

Tabell 28: Siste strømregninger

Komponent	Antall	Strøm/per	Strøm totalt
Arduino Mega 2560 R3	1	200mA	0,2A
Digital I/o pins	53	20mA	1,060A
Drivmotorer	4	990mA	3,96A
Rotasjonsmotorer	4	150mA	0,6A
Drivenkodere	4	40mA	0,16A
Led-stripe	2	3,6A	3,33A se Spenningsregulering
HC-SR04	6	15mA	0,09A
LCD-skjerm	1	160mA	0,16A
Heisemotor	1	2,8A	2,8A
Motorkontroller	4	36mA	0,144A
HC-05	1	40mA	0,040A
USB-host shield	1	500 mA	0,5A
Total			13,044A

Oppladning av batteri

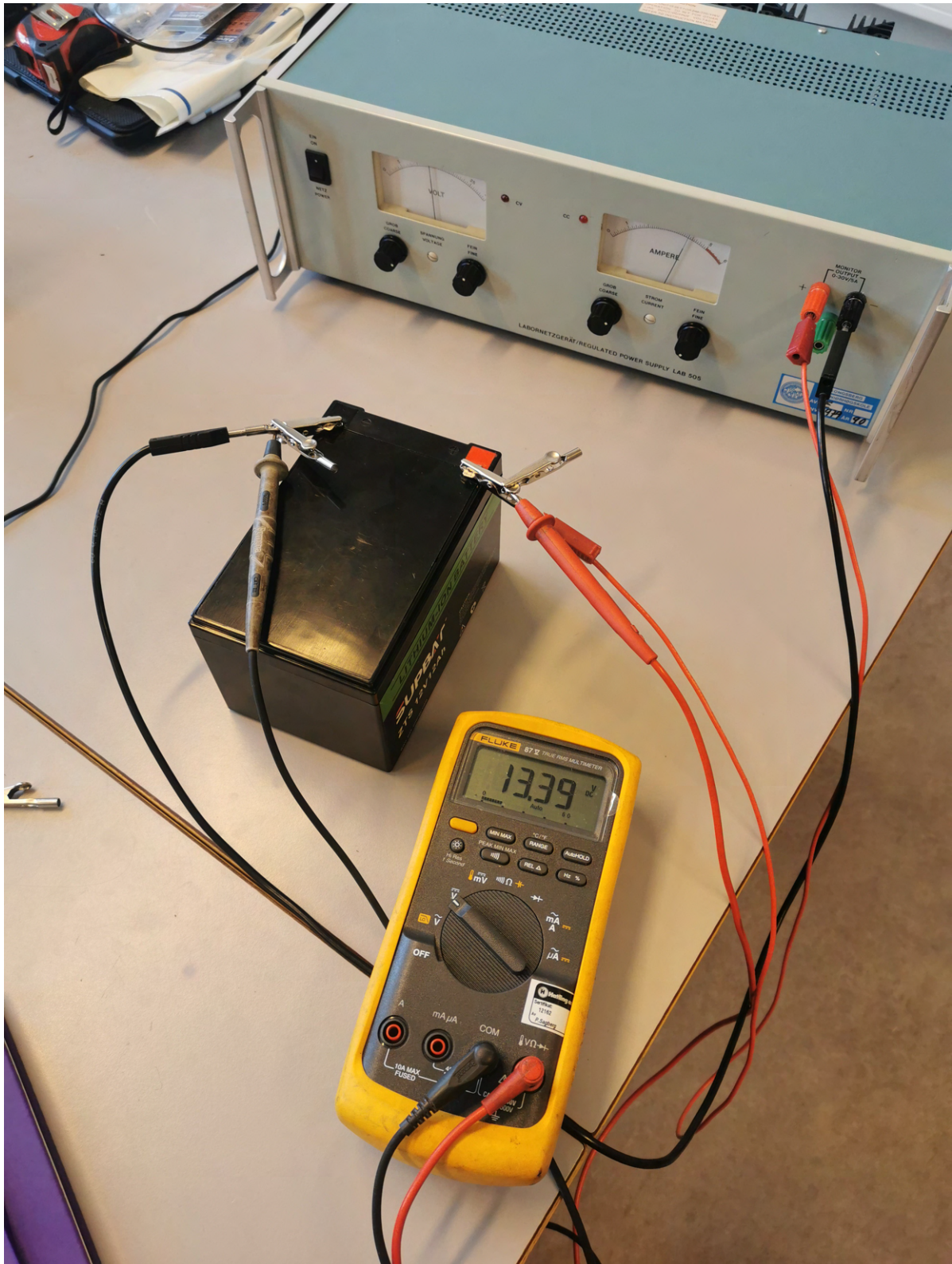
Det andre batteriet vi ansaffet fra EES hadde stått på lagring en stund og var derfor tomt for spenning til et nivå der det var for lavt for en ny smartlader og detektore liv i batteriet og ville da ikke lade.

Didrik fra EES viste til oppladning ved hjelp av en gammel billader fra Biltema som vist i fig. **116**.



Figur 116: Gammel billader som ble vist til oppladning av batteri ved EES

Årsaken til at billader fra fig. 116 fungerer, er at dette er en type som ikke bryr seg om den registrerer noe spenning på det den skal lade, den skal bare gjøre jobben sin. Nye billadere kan du ikke bruke til å lade flate batterier da de må føle en spenning for å faktisk lade. For at vi skulle kunne lade det igjen med smartlader ble vi nødt til å bruke en power supply fra skolen til å lade det opp til et nivå der elektronikken i BMS i batteriet slo seg på igjen. Se fig. 117.



Figur 117: Lade opp det flate batteriet fra EES med monitoring av spenningen.

E Styring

Vi startet med å se til stepper motor for rotasjon fordi med disse kunne vi styre kontrollen med steps veldig nøyaktig med åpen sløyfe, men det viste seg at disse tatt mye strøm samtidig som at det krevde mange pins for å kjøre disse fremfor DC-motorer og servoer. Vi bestilte en stepper for å teste ut nøyaktighet og styring. Motoren som ble bestilt var 535-0401 fra RS-components. Med tanke på at vi ønsket å kunne lese av vinklene som hjulene sto i var det ønskelig med steppermotor med innebygd enkoder, men det viste seg å være vanskelig å oppdrive, og det ble sett etter andre motorer med innebygd enkoder.

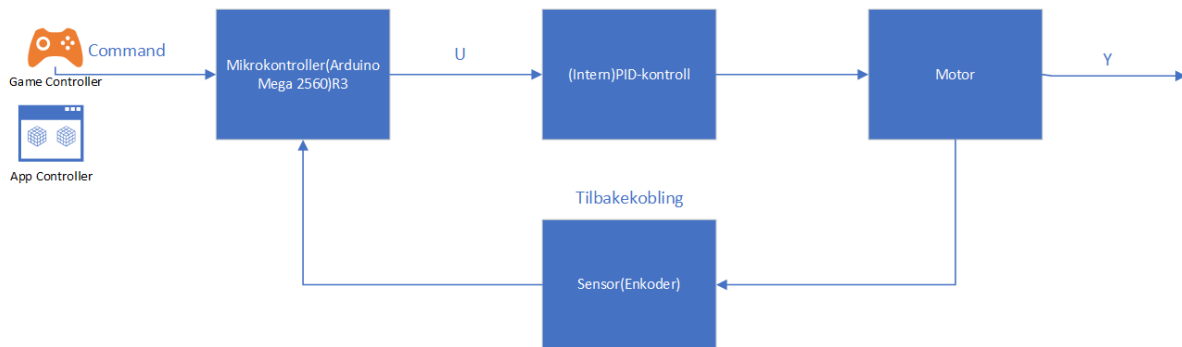
Etter vi gikk bort ifra stepper motorer, så vi til DC-motorer med innebygd enkoder. Etter råd fra faglærer(Dag Samuelsen) til en rekke med motorer fra Elfa Distrelec med innebygd koder. Vi fant frem til RHE 158.12.200 som var oppført som en DC-motor med høyt moment og den hadde ikke så høyt forbruk heller, og det hadde blitt montert på en enkoder. Databladet var kun i henhold til motoren, og det var ikke oppført type enkoder, men det var såpass sent i prosjekt der det var behov å anskaffe motorer til å teste med samtidig som at vi hadde fått tilbakemelding på at noen risikoer må av og til bli tatt. Da vi fikk motor så vi et problem, da det var kun en-kanals enkoder og derfor ikke mulig å kalkulere retning forskyvelsen var i, men kun fart. Da ble denne motoren forkastet fra bruk til rotasjon da vi ikke har interesse av å måle farten til rotasjonen, det er plasseringen som er innteressant. Motoren kan du se i fig. [118](#).



Figur 118: Motoren anskaffet fra Elfa Distrelec

PID-kontroller

Etter det ble bestemt at vi skulle ha DC-motorer istedet for stepper motorer ble vi nødt til å se til å bruke lukket sløyfe tilkobling for presis styring. Det er her enkoderen kommer til bruk.



Figur 119: Reguleringskonsept for rotasjonsmotor

For styring av motorene tenkte vi å bruke PID-kontroll som lar seg greit implementeres ved hjelp av bibliotek fra Arduino. Når vi brukte tilbakekobling og interrupts for å lese av enkoderene er det viktig at enkoderene er riktig satt i henhold til hvilken som skal være interrupt. Ellers ender motoren med å fortsette å gå til den ønska posisjonen uten noen gang å finne den fordi den går i feil retning.

I tilbakekoblingen blir da PID-reguleringen fra Arduino fungerende som kontroller, som da går til motorkontrollere/motorer som deretter gir utgangsposisjon, enkoderen blir fungerende som sensor og gir beskjed tilbake til Arduinoen om posisjonen og om det stemmer med det vi ønsker å plassere den i. Visuell figur ser du fig. [119](#).

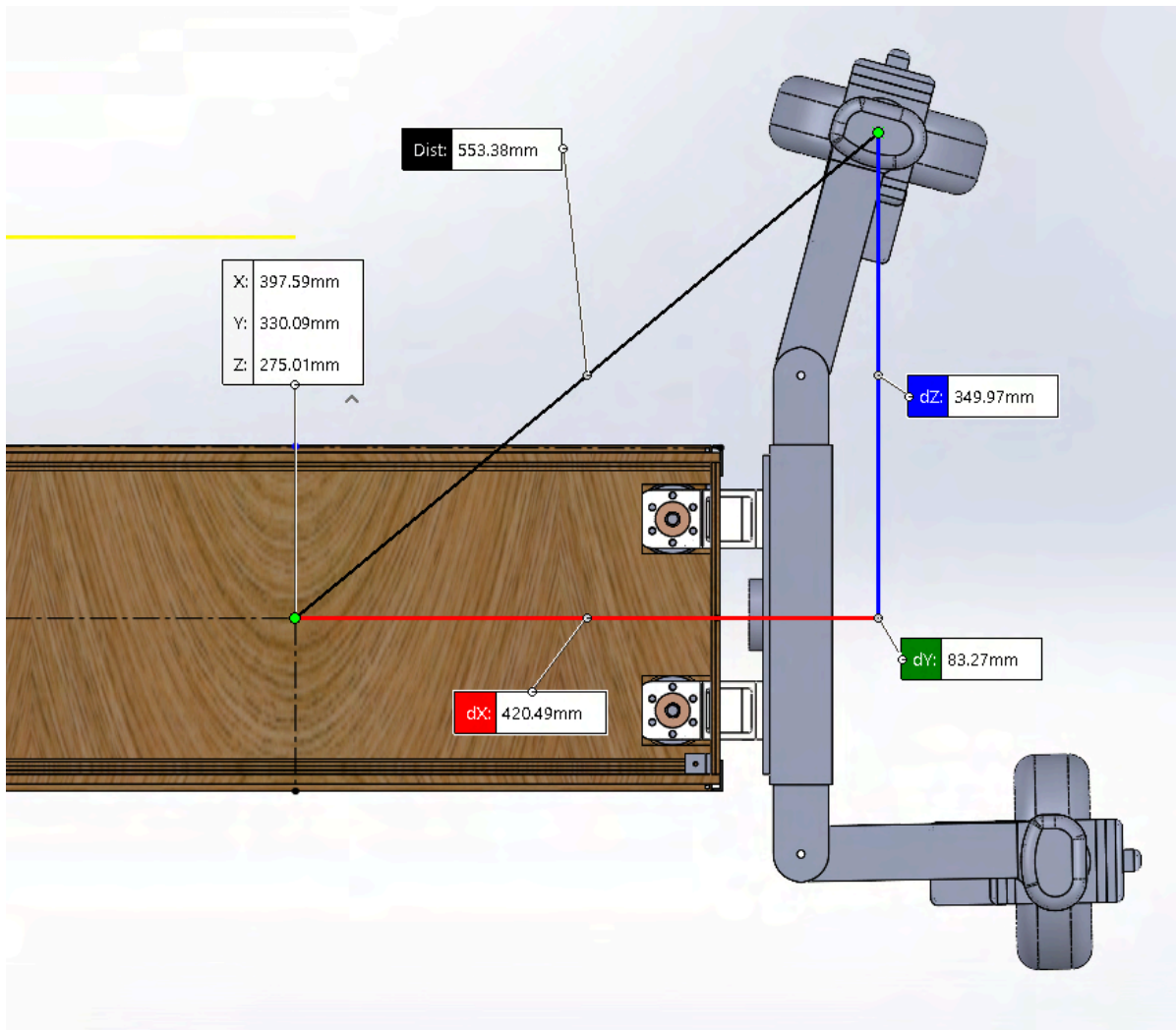
Gamle torque utregning for rotasjon

Det ble gjort utregning i Matlab der verdiene var regnet ut istedet for å bruke variabler og sette inn funksjon, da ble det regnet ut med at $p(x,y)$ utregna ble

$$p(x, y) = 30000 \cdot 48000000x^2$$

og grensene ble feiltastet inn som -0,0025 til 0,0025 istedet for -0,025 til 0,025, som da førte til sluttresultatet ble bare 0,0010 Nm. . Grensene var satt utifra arealet på kontaktflaten, og oppdagelsen av feil innsatte verdier ble gjort for sent til å klare å finne ny motor som klarte utføre jobben og passet det nåværende designet.

Gammelt design Ackermann



Figur 120: Visning av lengder i Roveren

Når vi setter inn verdiene vi ser fra Solidworks inn i (14) får vi

$$\theta = \arccos\left(\frac{3,5\text{cm}}{5,53\text{cm}}\right) = 50,73^\circ,$$

dette blir da vinkelen hjula er nødt til å stå i for at Roveren skal snu på stedet. Slik vi tenker at styringen skal fungere så starter hjula i 180° og roterer mellom 0° og 360° . Hvis vi medregner med klokka går fra 180 til 0 og mot klokka går mellom 180 og 360 så får at hjula som er på

diagonal av hverandre plasseres i

$$180^\circ - 50,73^\circ \approx 129^\circ,$$

og det andre paret med diagonale hjul plasseres i

$$180^\circ + 50,73^\circ \approx 231^\circ.$$

Ved en slik justering er parene med motorer på venstre og høyre side nødt til å gå hver deres retning. For rotering mot høyre, vil motorene på venstre side gå framover, og motorene på høyre side gå bakover.

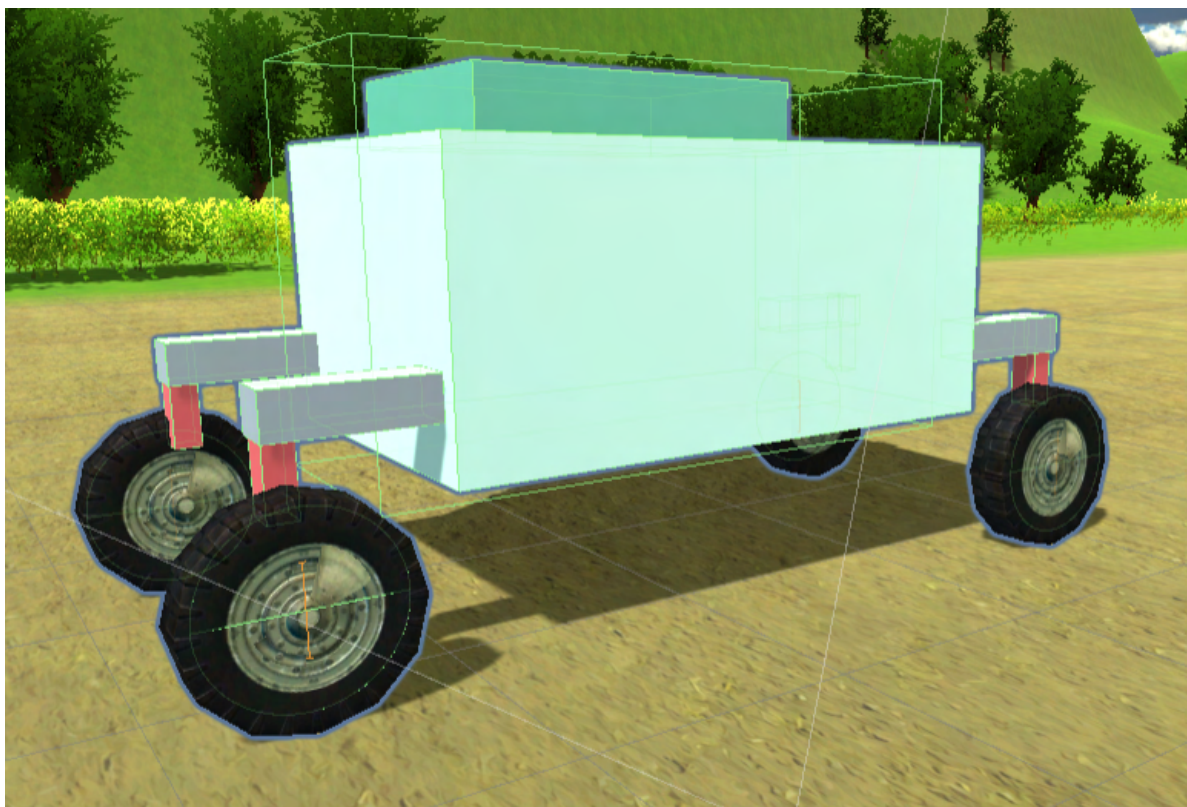
F Teknisk løsning for simulasjon.

Dette kapitlet inneholder mer informasjon om de tekniske løsninger og brukte innstillinger som står bak systemet.

F.1 Modell

F.1.1 Prototype

SoilSteam International har gitt oss deres 3D modell, for at vi skal kunne se maskinen og dra inspirasjon til vårt kjøretøy. Tanken var også at bachelorgruppe skal bruke modellen til testing av simulasjon. Etter flere forsøk viste det seg å være umulig. Årsaken til det var at SSI sin modell var allerede satt sammen og lagret som en STP fil. Ved konvertering av STP til OBJ format mister man de fleste informasjon om modellen, blant annet informasjon om separate deler som den består av, og i konsekvens mister modellen bevegelse av deler. OBJ modeller beholder kun geometriske former av objekter. Så produsert modell ble en statisk mesh, det betyr at deler av maskinen ikke kunne beveges. Derfor har bachelorgruppe bestemt å lage en forenklet versjon med geometriske former: terninger og sylindre. Denne prototypen ble brukt hovedsakelig for testing av WheelColliders og implementasjon av enkle bevegelser: kjøre frem og tilbake, heise hjul opp og ned.



Figur 121: SoilRover prototype

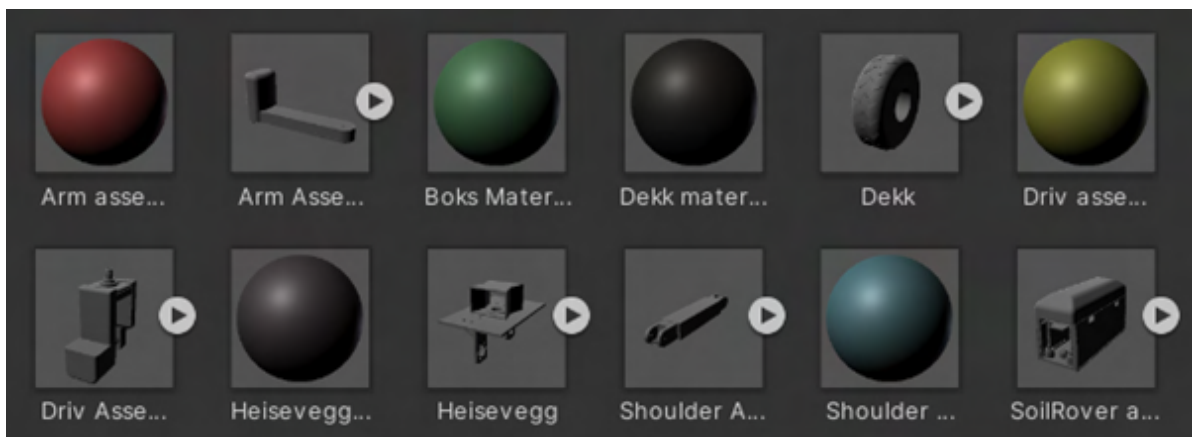
F.2 Nåværende 3D modell

3D modellen som er brukt i simulasjon ble laget i SolidWorks. Separate deler av SoilRover ble laget i STL format, og deler av maskinen som ble satt sammen ble lagret i STEP format. For å kunne importere modeller til Unity, måtte bachelorgruppe konvertere modellen til OBJ format, som er et generisk format for 3D modeller.

Bachelorgruppe har analysert modellen og har forstått at for å bli i stand til å bevege deler av en modell (for eksempel heise opp hjulene), må disse importeres til prosjektet som separate filer/objekter. For å unngå å måtte importere nesten 500 separate deler som SoilRover består av, bestemte gruppen at hele SoilRover kan deles i kun et par subdeler som blir bevegelige,

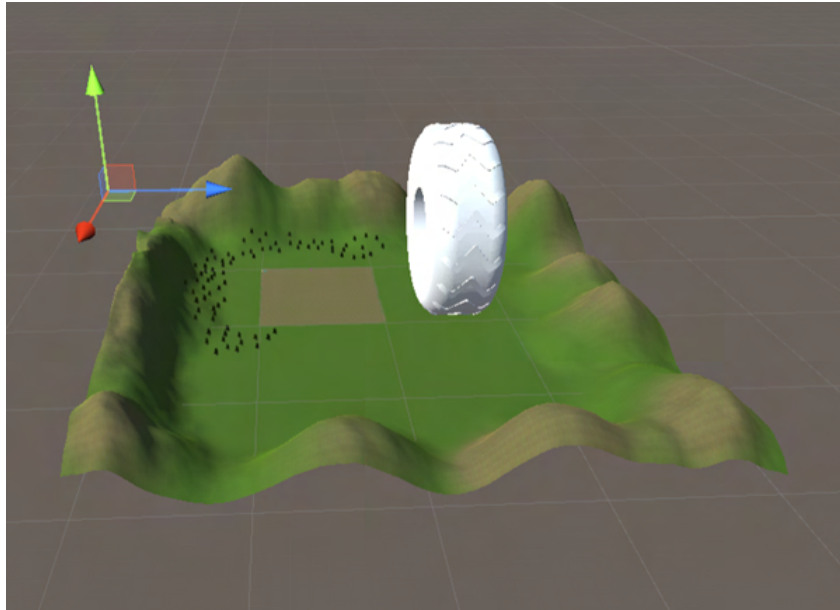
det vil si:

- kropp
- dekk
- hjulmodul
- heisevegg
- skulder
- arm

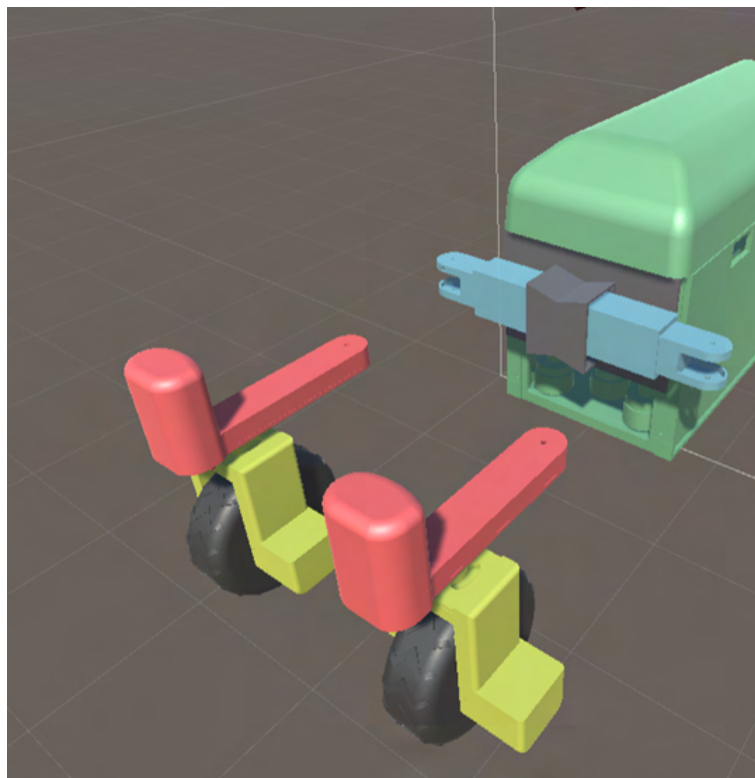


Figur 122: Deler

I originalformen hadde deler ingen farger og var ekstremt store. For at delene kunne tilpasses til størrelse av verden, ble hver del skalert ned fra skala 1 til 0.005. Dette ble bestemt etter en visuell inspeksjon.

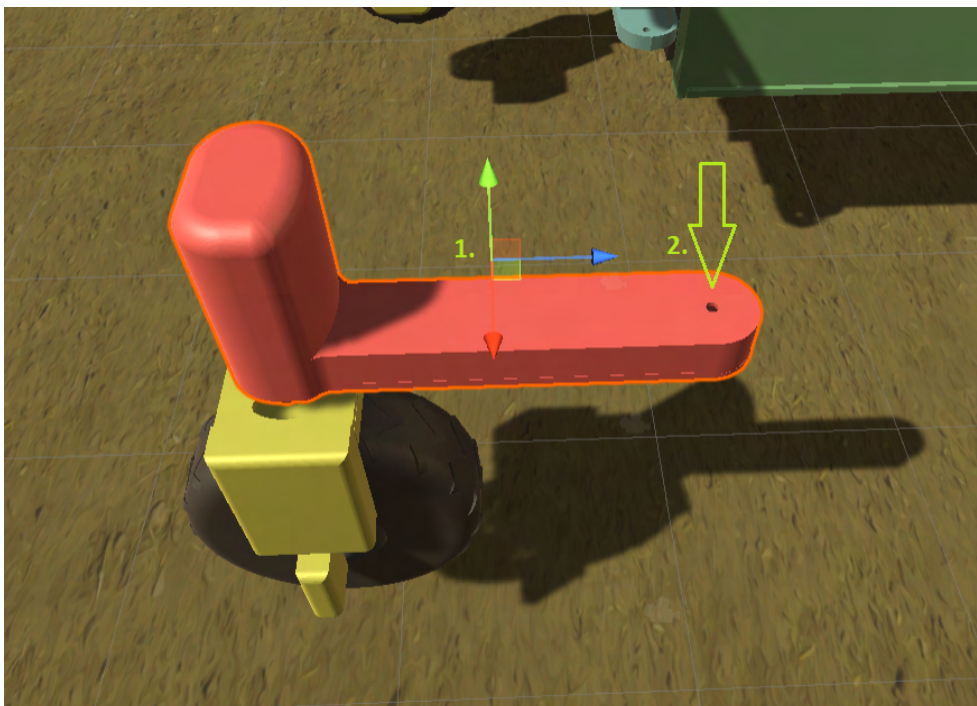


Figur 123: Hjul sammenlignet med verden



Figur 124: Under bygging

SoilRover ble bygget og koblet sammen manuelt. For hver del som skulle beveges, måtte bachelorgruppe endre senterpunkt, og samtidig rotasjonsakser. For å oppnå dette har gruppen laget tomme GameObjects, plassert disse i punkter for rotasjonsakser av objekter, og deretter ble hver del koblet til som et barn av GameObject. På denne måten ble det lett å få rotasjon rundt den nye aksen.

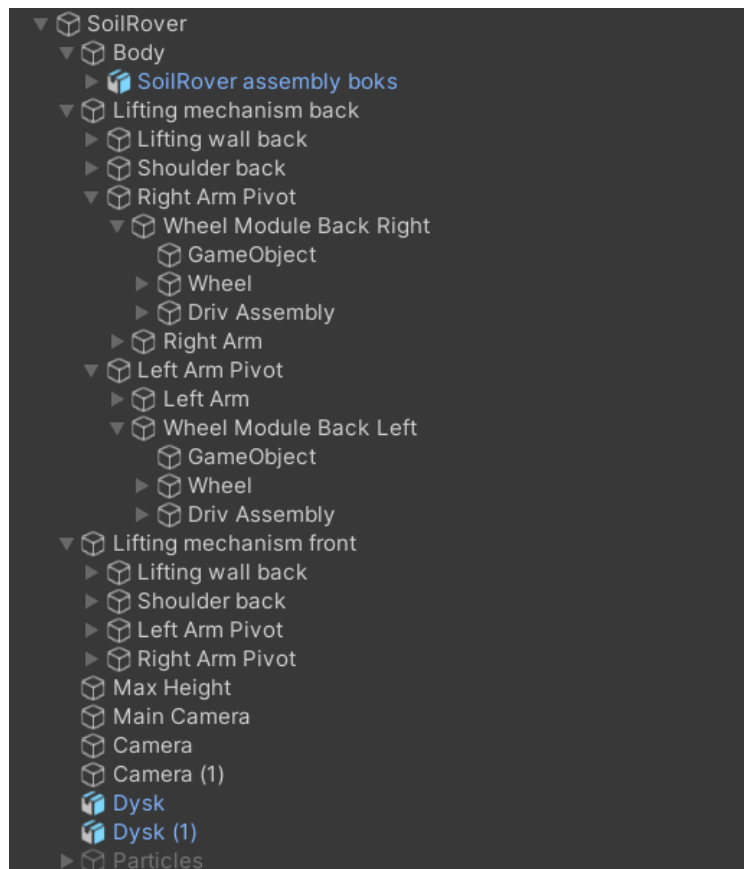


Figur 125: Endret senterpunkt, fra posisjon 1. til 2.

I Unity brukes det foreldre-barn hierarki for gruppering av objekter. Det betyr at et objekt – i Unity kalt «GameObject», kan inneholde andre objekter som arver dets egenskaper – posisjon, rotasjon og skala. Når objekter sitter sammen i en hierarki, hver bevegelse og skalering av en foreldre-objekt påvirker også barn-objekt. Barn-objekter kan ikke lenger flyttes relativt til globale koordinater (verdens koordinater), dets bevegelse skjer bare relativt til forelderens posisjon. Objekter er også laget i nestede foreldre-barn hierarkier. Et eksempel av nestede hierarki er «Lifting Mechanism Back» hierarki.

Alle nestede objekter behandles som etterkommere av objektet som står på toppen av hierarkiet – i dette tilfelle «SoilRover» er forelderens (root) til alle andre objekter. Dessuten har root ob-

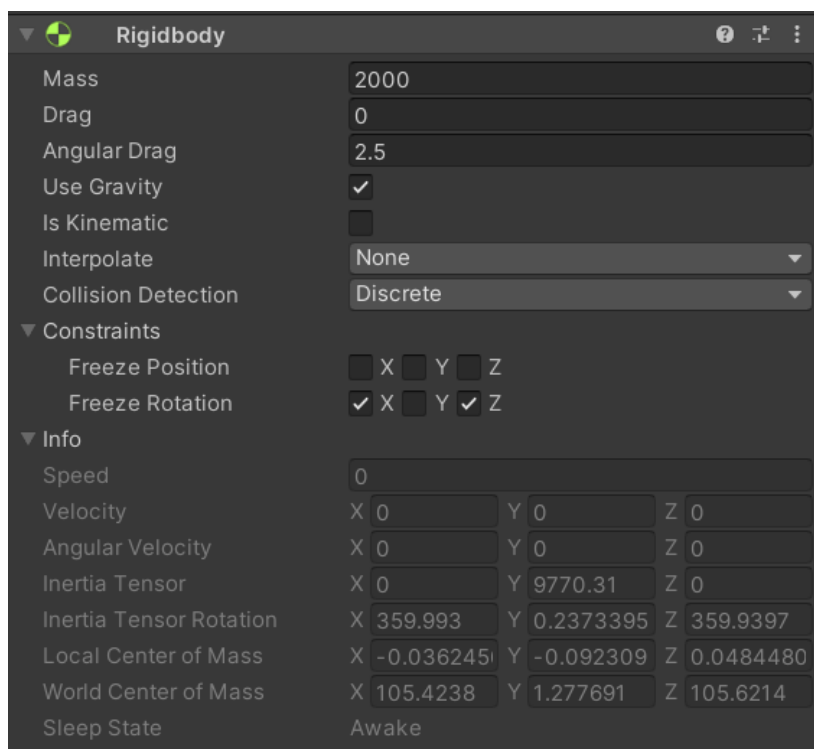
jektet tilgang til egenskaper av alle sine barn-objekter, og derfor var det mest korrekt å knytte alle skripter for bevegelse til SoilRover objekt. [39]



Figur 126: Hierarki av SoilRover deler

En av utfordringene var å finne en riktig hierarki for deler. Det er objekter som ikke kan være knyttet sammen som barn-forelder, for eksempel dekk og hjulmodul. Grunnet manglete erfaring med lignende prosjekter, har bachelorgruppen brukt mye tid for testing av bevegelser i forskjellige hierarki-oppsett.

For at SoilRover skal kunne oppføre seg som et fysisk objekt, var det nødvendig å legge til en komponent som heter «RigidBody»:



Figur 127: RigidBody innstillinger

«RigidBody» er en komponent som gjør at spillobjekter kan være påvirket av fysikken. Det vil si at objekter kan falle på grunn av tyngdekraften, eller ha andre fysikk egenskaper som for eksempel masse, drag kraft og hastighet. For å oppnå riktig oppførsel, måtte bachelorgruppe tilpasse en del innstillinger for «RigidBody»:

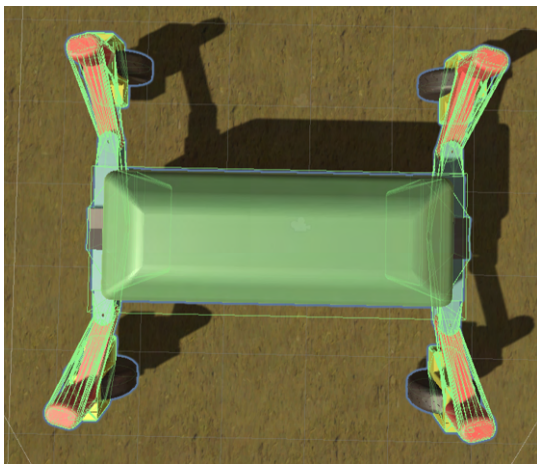
- Massen til SoilRover ble satt til 2000 enheter. Den gir ganske realistisk følelse, og gjør at SoilRover ikke hopper eller rister, noe som var problemet når massen var for liten.
- Angular Drag kraft ble satt til 2.5. Bachelorgruppe har utført mange tester for å komme til at 2.5 er en optimal verdi for at svingning blir forutsigbar og realistisk. Hvis angular drag var altfor lav, kunne det observeres overstyring, det vil si maskinen svinget mer enn den skulle. Det motsatte skjedde når angular drag var altfor høy - svingning var altfor treg.
- «Freeze Rotation» har blitt aktivert for å unngå at SR roterer om X og Z akse. Rotasjon er

tillatt kun om Y akse. Dette ble aktivert som tilleggssikkerhet. Det skjedde et par ganger under testing at SoilRover falt til side ved svingning.

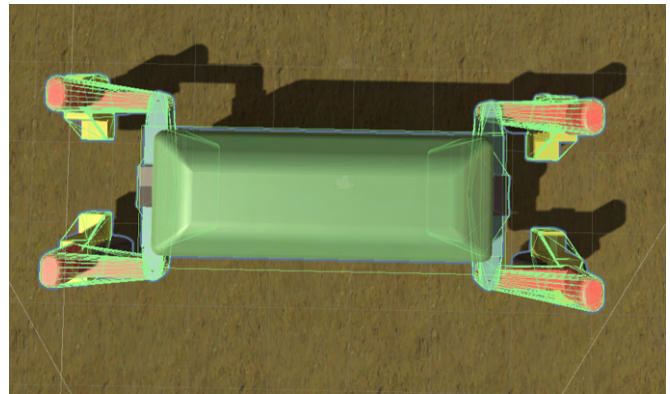
Resten av innstillinger er standard innstillinger som ikke ble endret.

F.3 Bevegelser

Med interne bevegelser av SoilRover menes det endring av arm posisjon, heisemekanisme og hjul rotasjon, sammen med hjulmodul.

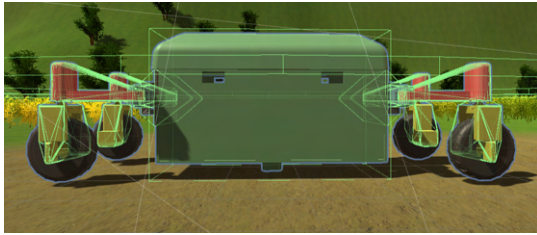


(a) Armene i 75 grader

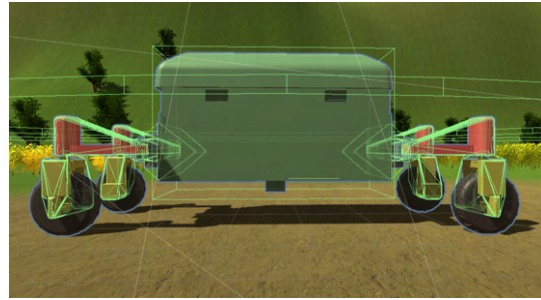


(b) Armene i 0 grader

Figur 128: 2 arm posisjon

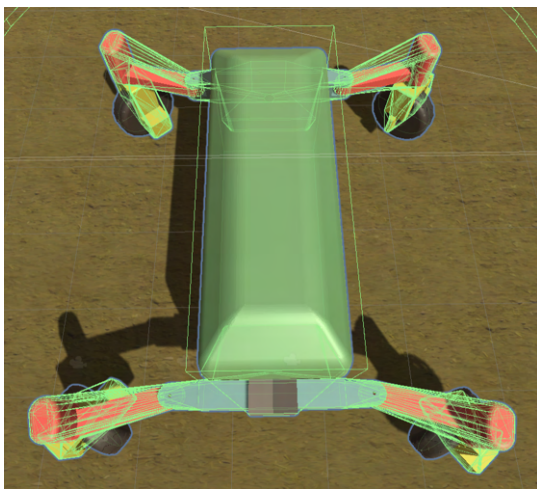


(a) Hjul heist opp.

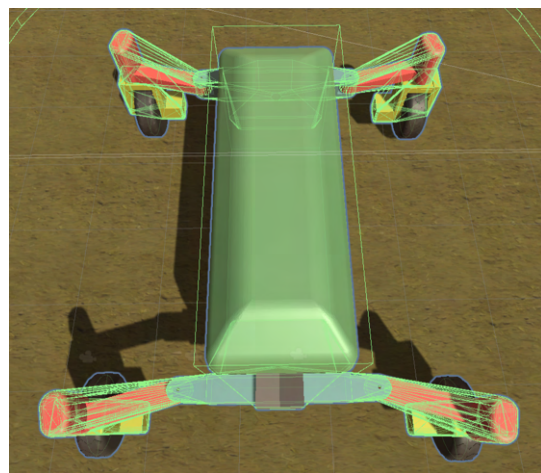


(b) Hjul heist ned.

Figur 129: Heisemekanisme



(a) Hjulmodul og hjul i posisjon 1.



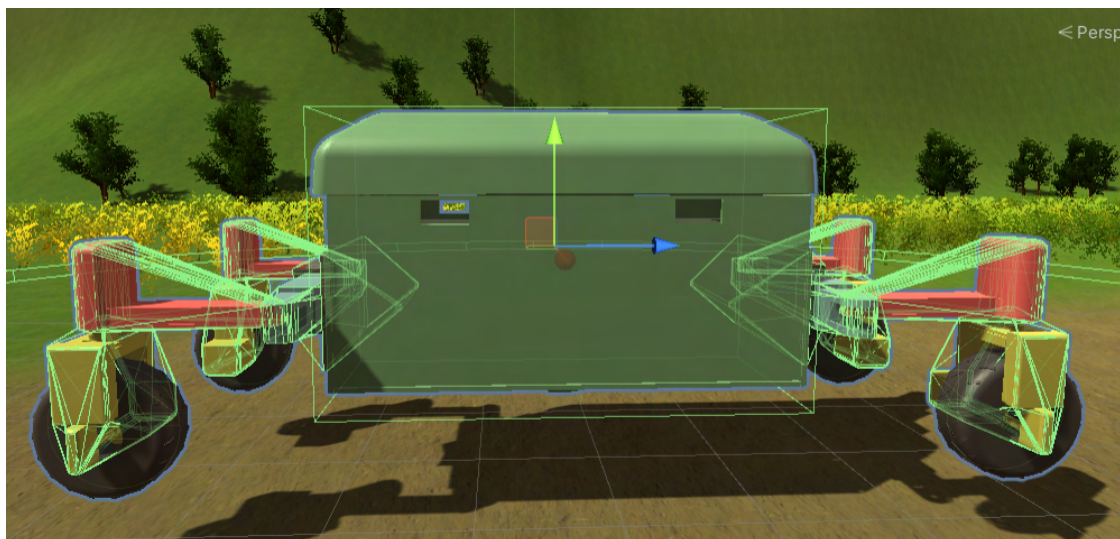
(b) Hjulmodul og hjul i posisjon 2.

Figur 130: Hjulmodul beveges sammen med hjul

F.4 Colliders

Før SoilRover kunne beveges, var det viktig med riktig implementasjon av kollidere for hele modellen. Kollidere i Unity er komponenter som definerer en form til et objekt. Når kollidere samhandler, simulerer overflater egenskapene til materialet de skal representere. Kollidere hjelper å gjenkjenne fysiske kollisjoner. [40]

Kollidere som var brukt for SoilRover er «MeshCollider» og en «BoxCollider». «MeshCollider» kobler sammen alle toppunktene - vertices, og lager en separat plan kollider på overflaten, som i utgangspunktet er en stor tilnærming. «BoxCollider» ble brukt for å lage kollider for boksen, fordi det var viktig å ha en jevn overflate under boksen, for at maskinen ikke skal vippe når hjul heises opp.



Figur 131: Kollidere

F.4.1 Wheel colliders

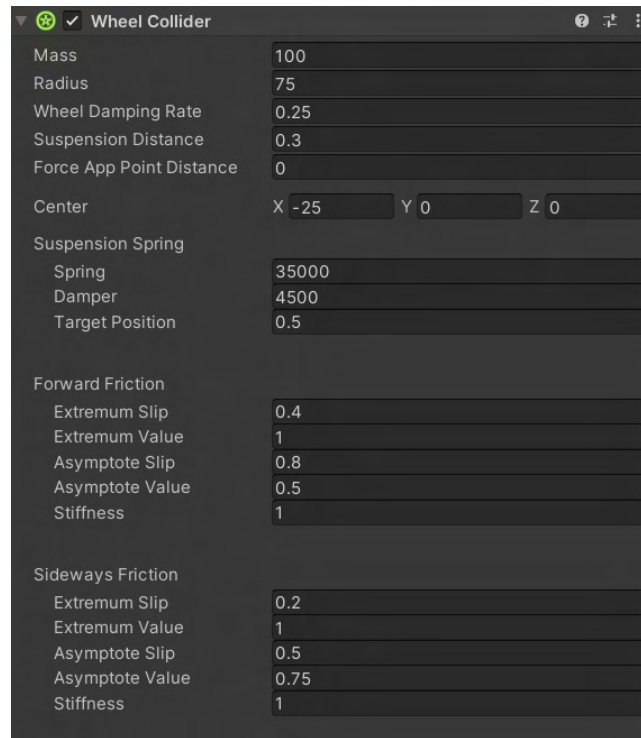
«Wheel Colliders» er spesiell type kollidere som kan brukes for kjøretøy. De har innebygget kollisjon deteksjon, hjulfysikk og en slip-basert dekkfriksjonsmodell. [40] Som vist på figur 133 er det er flere parametere som kan tilpasses for å oppnå ønsket hjul oppførsel. For denne simulasjonen ble kun masse og radius tilpasset vårt kjøretøy. Resten er standard verdier bestemt av Unity.



Figur 132: WheelCollider

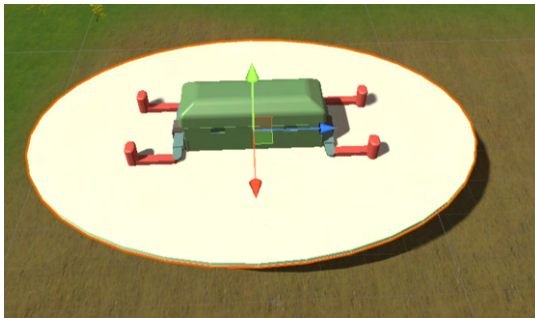
Den grønne sirkelen som sitter midt i dekkene er «Wheel Colliders», figur [132](#). Det å finne en riktig posisjon for kollideren var en stor utfordring, fordi det måtte gjøres manuelt med ekstremt stor presisjon/nøyaktighet. Kollideren må sitte nøyaktig i midten, og kan ikke være større enn selve dekkene.

Ved å bruke «Wheel Colliders» er det risiko for gliding av kjøretøy, og det var en stor utfordring som bachelorgruppe har møtt. Etter en del undersøkelse ble det antatt at problemet kan skyldes bruk av «brakeTorque» eller «motorTorque» som hindrer den «RigidBody» mot å stoppe. («brakeTorque» og «motorTorque» er funksjoner av «WheelCollider» public class). Dette problemet har gruppen ikke klart å finne riktig kilde til og fikse før innlevering.

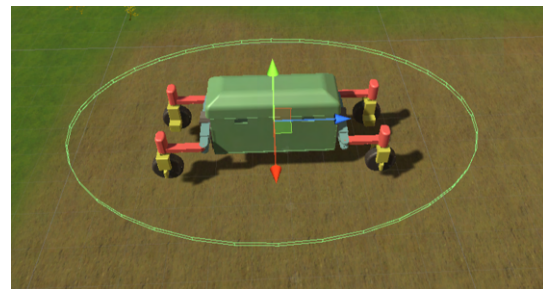


Figur 133: Wheel Collider detaljer

F.5 Kollisjon deteksjon



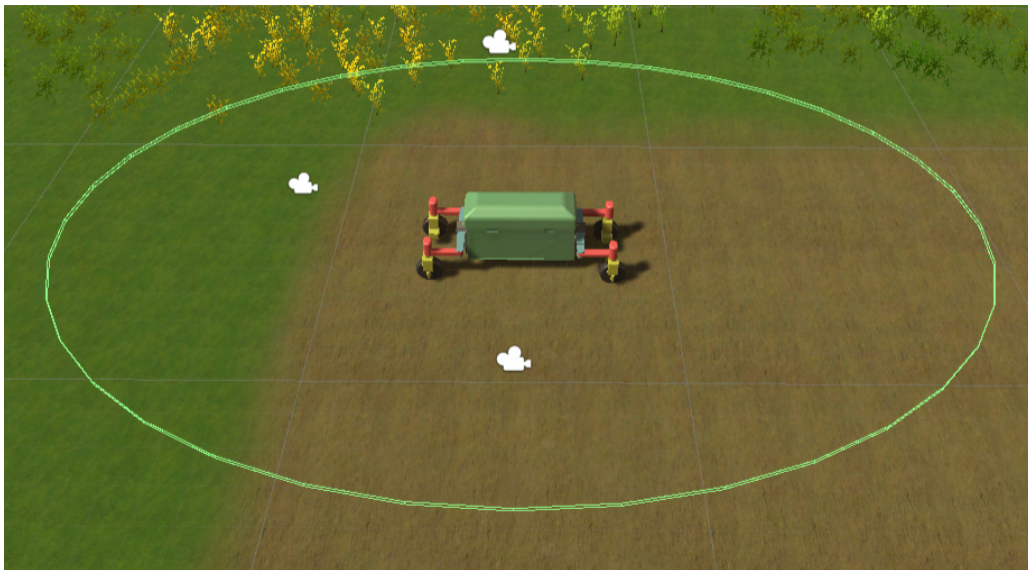
(a) Disk med MeshRenderers



(b) Disk uten MeshRenderers

Figur 134: Liten disk

For å implementere kollisjon deteksjon har bachelorgruppe brukt et annet konsept enn for den reelle maskinen- det er brukt to diskere som er en del av SoilRover. Som vist på figur 134, står kjøretøyet i sentrum av en av diskene. Disken ble laget som et vanlig 3D objekt 134a, men «MeshRenderere» ble deaktivert for den 134b. Det gjorde at disken ble usynlig i spill modus, men fortsatt eksisterer som et spill objekt. Komponenten som har ansvar for å detektere objekter heter «MeshCollider». Den fungerer som usynlig overflate av et objekt som den er lagt til, som tidligere nevnt i F.4. Alle spill objekter som har RigidBody og kommer i kontakt med disken, skal håndteres som hindringer, og SR skal ikke kunne kjøre videre mot dem, det vil si at ingen objekter kan krysse grensen som disken setter. Dessuten, når SR ankommer et objekt som har tag «Obstacle» i avstand mindre enn radius til den store disken 135, på skjermen vises det en varslingsstekst som sier «Collision alert!», for å advare om et objekt i nærheten. Den store disken blokkerer ikke objekter, den har parameter «Is Trigger» som gjør at disken sjekker når objekter passerer dens grenser. Sentre av begge diskere ligger i samme punktet som senter til SoilRover.



Figur 135: Stor disk

Hindringer

Objekter som er tilknyttet knappene 1-9. Det å trykke på en av knappene gjør at «MeshCollider» til det valgte objektet aktiveres, det vil si at objektene begynner å bli synlige i spillmodus, får en fysisk form og blir påvirket av fysiske krefter- blant annet gravitasjon, og derfor faller de ned på åkeren. Objektene er alltid plassert på faste lokasjoner over åkeren.

Objekter som regnes som hindringer:

- objekter som er tilknyttet knappene 1-9, dessuten har de «Obstacle» tag, som avtrekker/trigger varselteksten.
- trær, har ikke «Obstacle» tag.

Gress regnes ikke som en hindring.

F.6 Kjøremoduser

F.6.1 Default kjøremodus

Når simulasjonen starter, er maskinen i en default kjøremodus. Det er mulig å kjøre i alle retninger, alle hjula svinger samtidig. Maksimal tillat sving er på 15 grader. Vinkelen kan justeres fra Unity prosjekt view. Det er en generell hastighetsgrense 3.5 enheter. Dette ble bestemt etter visuell inspeksjon, men det er mulig å endre etterpå, dersom det er behov for det. For beregning av vinkler for hvert hjul brukes det Ackermann styring. Maskinen kan styres ved å trykke WSAD knapper, piler eller LT på PS4 kontrollere.

Maskinen kommer tilbake til denne modusen når moduser F1-F5 deaktiveres.

F.6.2 F1 modus

I F1 modus skal SoilRover rotere om sin egen vertikale akse. For å kunne aktivere F1, må hjulene være oppe. Når de er det, begynner maskinen å automatisk sette vinkler for rotasjon: 50.7, 129.3, -50.7, -129.3. Deretter må hjulene heises ned og bruke W eller pil på tastatur, eller LT på kontroller. Ackermann styring er deaktivert her.

F.6.3 F2 - Valgt radius

Modusen kan aktiveres direkte fra default modus. Den gjør det mulig for maskinen å kjøre i en sirkel, hvor brukeren kan selv velge radius til sirkelen. Radiusen kan skrives inn i et input vindu som dukker opp, og bekrefte med Enter. Det er en begrensning at radiusen må være mellom 5-40. Den kan ikke mindre med tanke på maskinens størrelse. Hvis radius blir større - maskinen vil kjøre ut av åkeren. Input vinduet tillater kun integer verdier. Det brukes Ackermann styring i denne modusen. Maskinen kan styres ved å trykke WSAD knapper eller piler på tastatur, eller LT på kontroller.

F.6.4 F3 - framhjul

Modusen kan aktiveres direkte fra default modus. Når maskinen er i F3 modus, kan bare framhjul styres. Bakhjul er blokkert mot svinging. Vinkel er begrenset til 15 grader, det kan justeres fra Unity prosjekt view etter behov. Maskinen kan styres ved å trykke WSAD knapper, piler eller LT på PS4 kontrolleren.

F.6.5 F4 - bakhjul

Modusen kan aktiveres direkte fra default modus. Når maskinen er i F4 modus, kan bare bakhjul styres. Framhjul er blokkert mot svinging. Vinkel er begrenset til 15 grader, men den kan endres fra Unity prosjekt view etter behov. Maskinen kan styres ved å trykke WSAD knapper, piler eller LT på PS4 kontrolleren.

F.6.6 F5 - hvert hjul

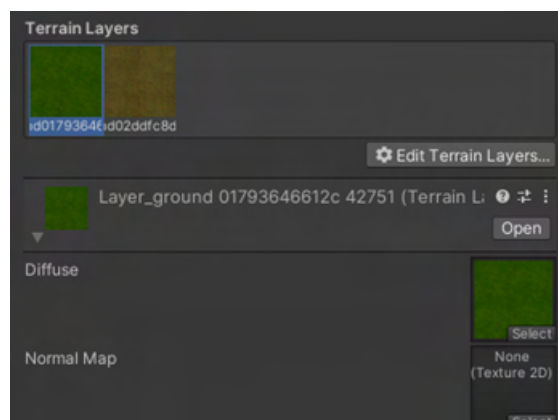
Denne modusen tillater valg av ønsket vinkel for hvert hjul. For å kunne aktivere F5, må hjulene til SoilRover være heiset opp. Vinkler skal skrives inn i fire input vinduer - hvert vindu for hvert hjul. Det er mulig å skrive verdier mellom 0-360, og det er tillatt kun integer verdier. For denne modusen er Ackermann styring deaktivert, siden det ikke er behov for å beregne vinkler.

E.7 Verden



Figur 136: Verden sett ovenfra

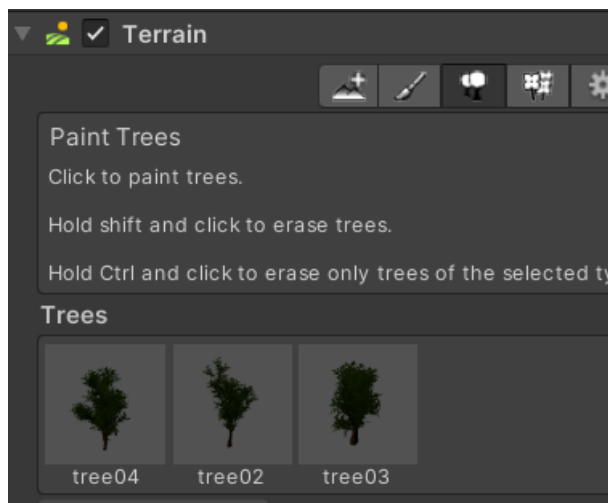
Verden er laget med en ekstern assets pakke «Nature Starter Kit 2» som ble lastet ned fra den offisielle Asset store nettsiden. [41]



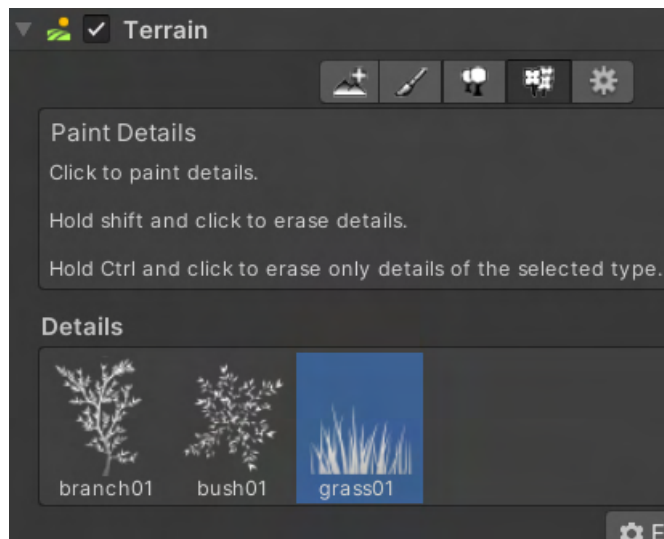
Figur 137: Terreng

Terrenget består av to lag. Terrenget ble malt ved bruk av «Paint Texture» innebygd verktøy. Terrenget er omgitt av fjell. Fjell ble laget ved bruk av to verktøy: «Raise or Lower terrain» og «Smooth Height».

Det er trær, busker og gress:



Figur 138: Trær



Figur 139: Typer av busker og gress

Himmelen som ble implementert kommer som en ferdig teksture. Den ble lastet ned Asset

Store Objekten som var brukt heter «CloudedSunGlow». [42]

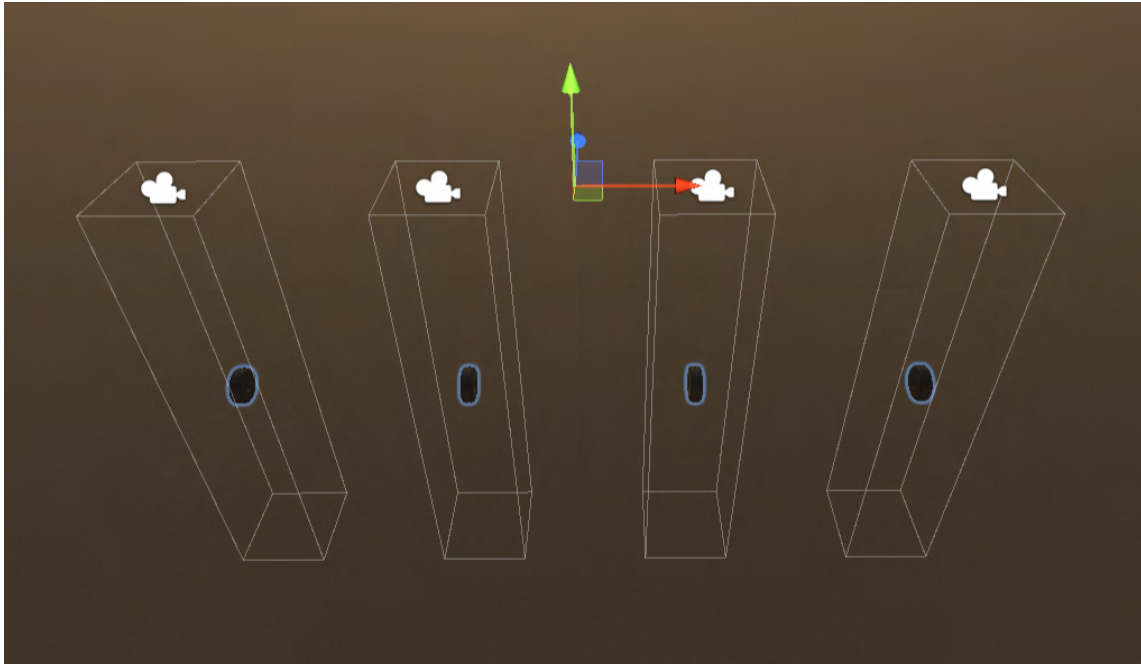
Damp funksjon som SoilRover har knyttet til «Left shift» knapp er lastet ned fra Asset Store [43]. Opprinnelig var det røyk animasjon, som ble redigert for å ligne på damp.

Enheter i Unity verden:

- distanse - meter
- masse - kilogram
- vinkel - grader
- hastighet - m/s
- kraft - Newton

F.8 User Interface

Som tidligere nevnt i kapittel 9.2.3, hovedmålet for User Interface er å gjøre det lettere å navigere i simulasjonssystemet. En del av User Interface er også Heads-up Display, som viser informasjon om maskinens tilstand på skjermen til enhver tid. En viktig del av Heads-up Display er hjul

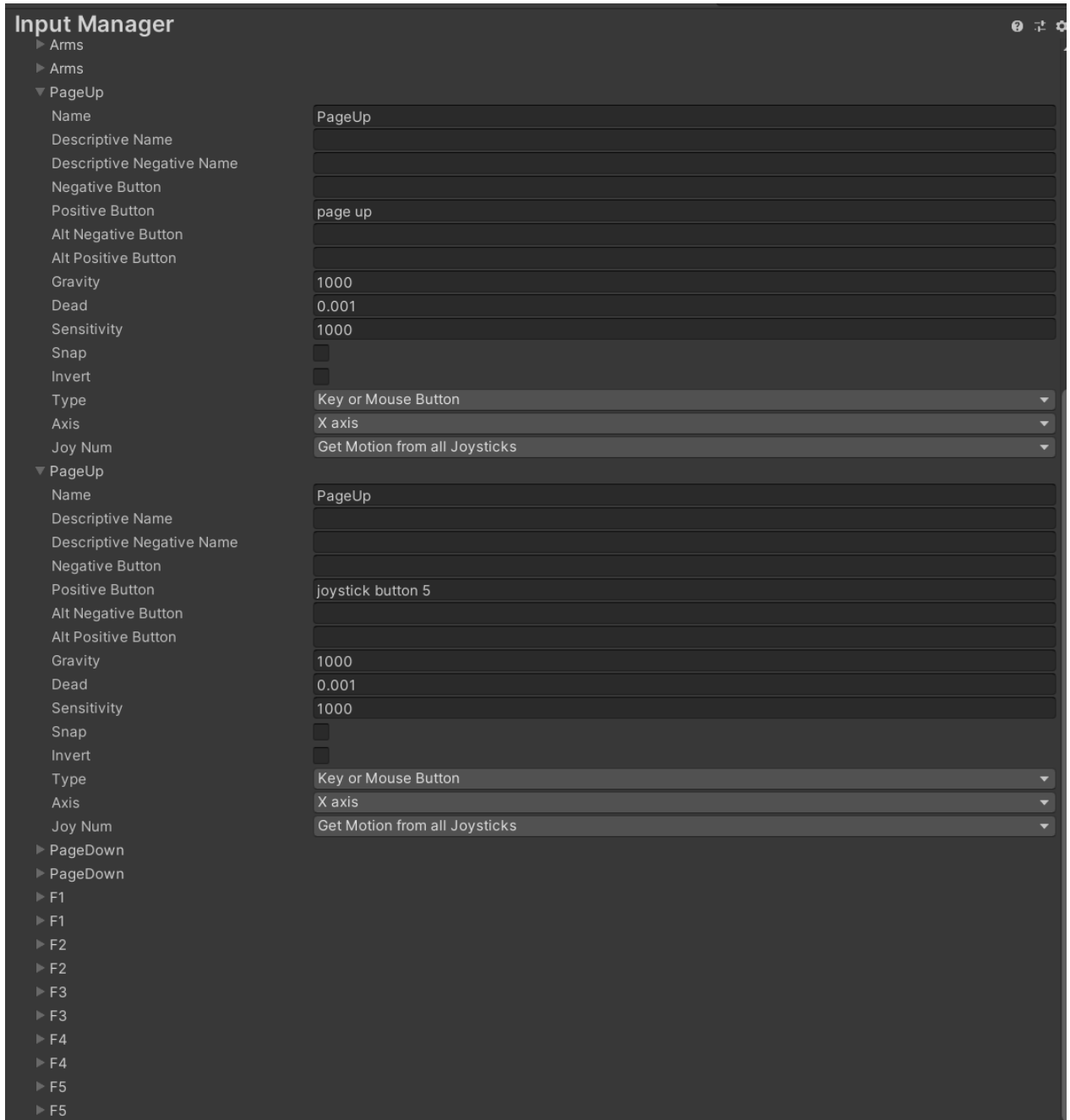


Figur 140: Hjul som ligger under terrenget

Under terrenget er det plassert 4 hjul objekter som er kopier av hjul implementert til kjøretøyet. Hvert hjul har sitt kamera som er rettet mot det og som observerer dets bevegelse ovenfra. Hvert hjul er tilkoblet en skript som arver rotasjon av SoilRover sine hjul, og sender variabler til hjul-kamera objekter. Skripten gjør at hjulene vist på fig [140](#) roterer på stedet, samtidig med SoilRover sine hjul, men uten bevegelse i X, Y og Z retninger. Deretter sender kameraer bilder direkte til UI, som viser hjul rotasjon på skjermen. Disse hjul-kameraer er ikke synlige når man befinner seg i en spillmodus. Det kan bare ses fra prosjekt modus i Unity.

Styring

Simulasjon kan styres med både tastatur og PS4 kontroller. Simulasjonen har en del funksjoner som ikke kunne overføres til kontroller grunnet for lite tilgjengelige knapper, så det å styre den med tastatur er mest brukervennlig. Funksjoner som ikke er tilgjengelige med kontrollere: damp funksjon, legge til objekter for å teste kollisjon deteksjon og brems. Knapper ble definert i Input Manager i Unity.



Figur 141: Input Manager i Unity

F.9 Oppsummering

Selv om det å bruke simulasjonen ikke er det samme som å styre den reelle maskinen, gir simulasjonen en behagelig følelse av kjøring som tilfredsstillende et ønske om å teste maskinen uten at man trenger å tenke på eventuelle konsekvenser ved feilbruk. Dessuten kan det noen ganger være enklere å presentere en simulasjon, fremfor å bruke en fysisk modell. For eksempel kan simulasjonen enkelt vises til en potensiell kunde under en video samtale.

Simulasjonen er også et forsøk på å svare på et spørsmål: «Hva skjer, hvis...?». Maskinen er ikke bygget ennå, så den kan gi et referansepunkt for ingeniører som i fremtid kommer til å utvikle maskinen i original størrelse, og denne simulasjonen kan hjelpe til å identifisere mulige driftsproblemer. Et viktig poeng er at videreutvikling av simulasjon blir mulig, dersom det blir behov for det.

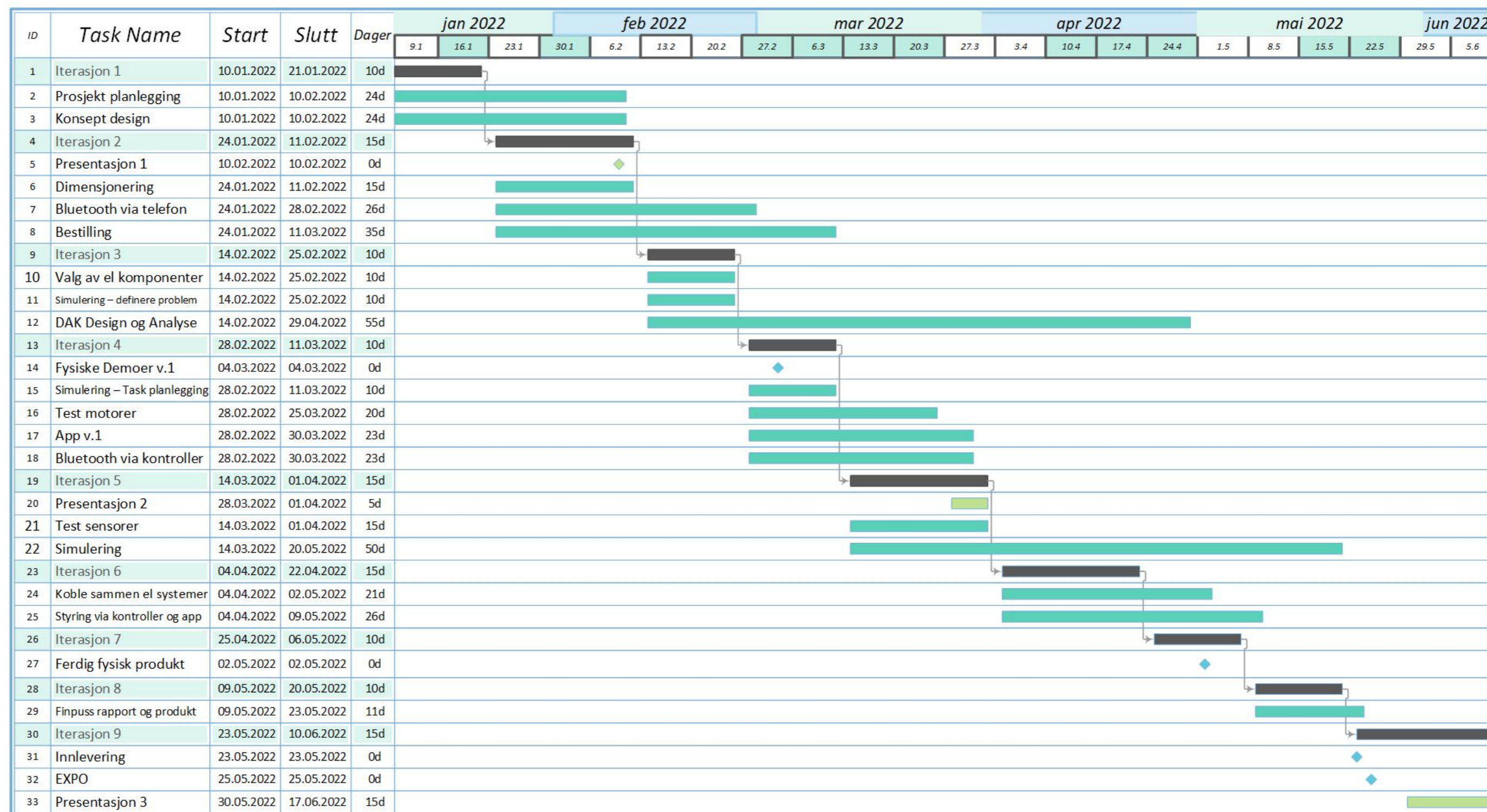


Figur 142: QR-kode som sender bruker til våres nettside

G GANTT-diagrammer

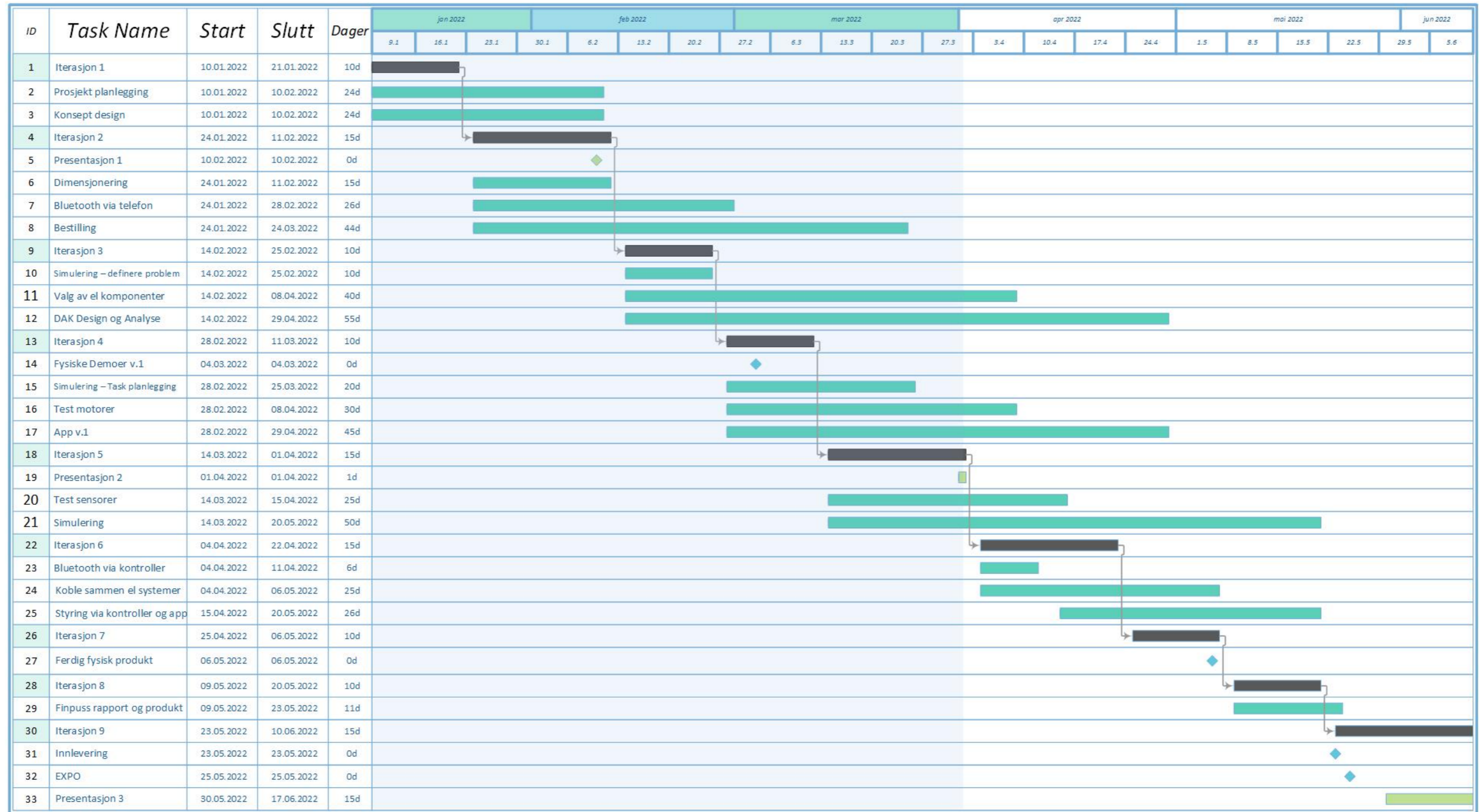
G.1 GANTT-diagram v.1

Gantt Bachelor



G.2 GANTT-diagram v.2

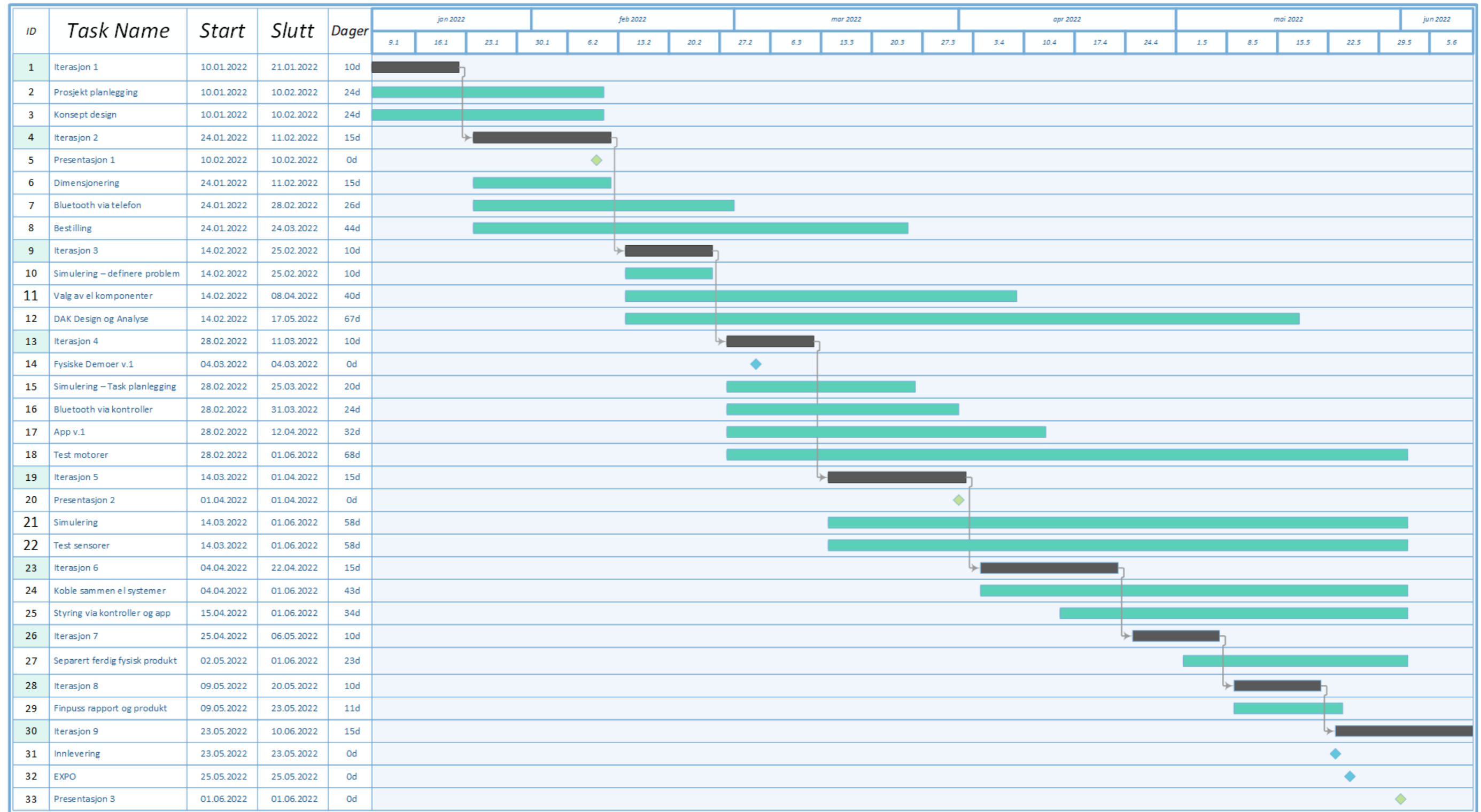
Gantt Bachelor



Figur 144: GANTT-diagram v.2

G.3 GANTT-diagram v.3

Gantt Bachelor



Figur 145: GANTT-diagram v.3