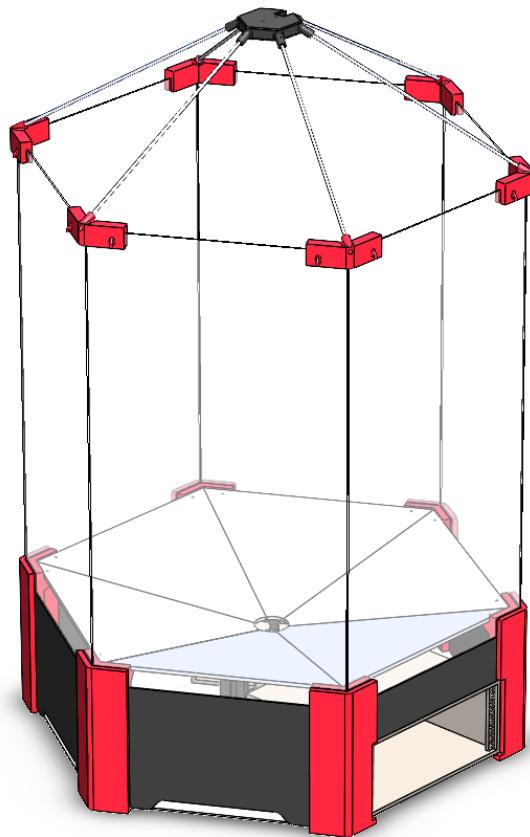


Bachelor thesis
DESIGN AND DEVELOPMENT OF A DRONE REINFORCEMENT
LEARNING ARENA



Jørgen Borge
Martin Sauar Wad
Martin Økter

Veileder: Kristian Muri Knausgård

Universitetet i Agder, 2022
Fakultet for teknologi og realfag
Institutt for ingeniørvitenskap

Innhold

1	Introduksjon	10
1.1	Droner som en mulig hjelper i samfunnet	10
1.2	Formål og bakgrunn for behov ved UIA	10
1.3	Toppnivåkrav	11
1.4	Avgrensing	12
1.5	Arbeidsmetoder	12
1.5.1	Planer	12
1.5.2	Kommunikasjon	13
1.6	Budsjettramme	13
2	Teori	15
2.1	Reinforcement learning i fugleperspektiv	15
2.2	3D printing som produksjon	15
2.2.1	Fused Filament Fabrication	16
2.2.2	Stereolithografy	17
2.2.3	Selective Laser Sintering	18
2.2.4	Med hensyn på produksjon	18
2.2.5	Med hensyn på droneproduksjon	19
2.3	Plast typer	19
2.3.1	Innvirkning på material egenskaper ved 3d print	19
2.3.2	PLA	19
2.3.3	PLA varianter	19
2.3.4	PP	20
2.3.5	PETG	20
2.3.6	TPU	20
2.4	ArUco-markør	20
2.5	Kretskort linjebreddeberegning	21
2.6	Kamera krav	22
2.7	Intel RealSense Depth Camera D435	22
2.8	QI-Lading	22
2.9	Aerodynamikk	23
2.9.1	Effekten av kontinuitetslikningen	23
2.9.2	Bakkeeffekt	24
2.9.3	Vortex ring state	25
2.10	Kakute F4V2 med betaflight	26
3	Konseptutvikling	27
3.1	Glassbur konseptet	27
3.2	Midt-aktuator konseptet	28
3.3	Kube-arena konseptet	28
3.4	Plattform på hjul	29
3.5	Konseptvalg	30
4	Design	33
4.1	Systemdesign	33
4.1.1	Arena	34
4.1.2	Drone	35
4.2	Mekanisk design	36

4.2.1	Hjørnestøttene	36
4.2.2	Midtseksjon	37
4.2.3	Glideplatene	37
4.2.4	Midt-aktuatoren	38
4.2.5	Bunnplaten	42
4.2.6	Topp-seksjonen	42
4.3	Drone-design	44
4.3.1	Drone skrog	44
4.3.2	Skrogbunn	45
4.3.3	Toppskrog	46
4.3.4	Støttestruktur	46
4.3.5	Prop guards	47
4.4	Komponenter og infrastruktur for maskinlæring	49
4.4.1	Beregningskraft	49
4.4.2	Generelt om Hardware	49
4.4.3	GPU	49
4.4.4	Hovedkort	50
4.4.5	CPU	50
4.4.6	Skjerm	51
4.5	Kamera	51
4.6	Detektering med ArUco	52
4.7	Komponenter for styring av arena	53
4.7.1	Raspberry Pi 4	53
4.7.2	Stepperdriver DVR8825	54
4.7.3	Motordriver TB66FNG	54
4.7.4	Stepper motor og DC motor	55
4.8	PCB design	56
4.9	Aktuator-skuff	59
4.10	Software design	61
4.10.1	Kommunikasjonsplatform	61
4.10.2	Dybde måling	61
4.10.3	ArUco	62
4.10.4	Aktuator styring	64
5	Eksperimenter	69
5.1	Glide-test	69
5.2	Falltest	69
5.3	Falltest Solidworks	70
5.4	Flygetest	70
5.5	Kamera test	71
5.6	Åpen dag sammenstilling	71
5.7	QI-test	73
6	Resultater	74
6.1	Glidetest	74
6.2	Falltest	75
6.3	Falltest Solidworks	76
6.4	Fly test	78
6.5	Kamera test	79
6.6	QI-test	80

7 Diskusjon	81
7.1 Konsept	81
7.2 Glideplater og friksjon	81
7.3 Aerodynamikk	82
7.4 Versatilitet	83
7.4.1 Mekanisk mutiplisering	83
7.4.2 Felles hub	84
7.4.3 Trening for nybegynnere	84
7.5 Porterbarhet	84
7.6 Flygbarhet av dronen	85
7.7 Dronens holdbarhet	86
7.8 Kameradeteksjon	87
7.9 QI-ladingen	88
8 Konklusjon	89
9 Videre arbeid	90
9.1 Arenaen	90
9.1.1 Glideplater	90
9.1.2 Bedrer motor valg	90
9.1.3 LED	90
9.1.4 Nett	90
9.2 Drona	90
9.2.1 Innmat	90
9.3 Kommunikasjon	91
9.3.1 Kommunikasjon mellom RPI og data	91
9.3.2 Kommunikasjon mellom drone og data	91
9.4 Innebygget Skjerm	91
Bibliografi	92
A Raspberry Pi kode	96
B Python grunnkode styringsmaskin	101
C Dybde kode styringsmaskin	103
D Bill of Materials	106
E Tekniske tegninger	108

Figurer

1.1	Oppgavefordeling med tidslinje fra midtfasen av prosjektet	13
2.1	Statediagram for Reinforcement Learning.	15
2.2	Kartesisk printer med dual nozzle setup	15
2.3	Tre forskjellige printer typer illustrert med Kartesisik, Core XY og Delta.	16
2.4	Anycubic Photon Zero SLA printer	17
2.5	Del til arenaen under produksjon på en Artillery SW-X2	18
2.6	ArUCo-markør med ID 42.	20
2.7	RealSense D435 uten fremdeksel.	22
2.8	QI system	22
2.9	Illustrasjon av en drone som hoverer med bakkeeffekt	24
2.10	Illustrasjon av en drone som opplever vortex ring state	25
3.1	Skissetegning av glassbur konseptet	27
3.2	Skisse av Midt-aktuator konseptet.	28
3.3	Skisse av Kube-arena konseptet.	28
3.4	Plattform med PC og virtuelt rom	29
3.5	Stolpediagram av poengene gitt til de forskjellige konseptene.	31
3.6	Total poengsum gitt til konseptene.	32
4.1	Systemdesign	33
4.2	Tilstander dronen kan være i og hvordan disse følger hverandre	33
4.3	Systemdesign av Arena	34
4.4	Systemdesign av Drona	35
4.5	Drone Arena hjørne, dimensjoner i mm sett i fullprofil ovenfra, teknisk tegning i figur E.3	36
4.6	Drone Arena hjørne, sett fra siden, teknisk tegning i appendiks E.3	36
4.7	Midtseksjonen sett ovenfra og fra siden, full teknisk tegning i appendiks 4.2.2	37
4.8	Aktuator-sammenstilingen sett fra midten og ut, teknisk tegning i appendiks E.2	38
4.9	Grafisk illustrasjon av rotasjonsplaten, teknisk tegning i appendiks E.16	39
4.10	Aksialfart vektorer fra rotasjonskiven og rotasjonshjulet	40
4.11	Dimensjoner fra toppen av aktuatorplatformen til platekanten	41
4.12	Teknisk tegning av axial-lager holder, full teknisk tegning i appendiks E.2	41
4.13	ATopplokk med integrert QI-sender, teknisk tegning uten QI 4.14	42
4.14	Rendring av ToppHjørneV2	42
4.15	Utdrag av teknisk tegning av topphjørne V3, full teknisk tegning i appendiks E.9	43
4.16	Topplaten som grafisk illustrasjon, teknisk tegning E.24	43
4.17	Bunn skrog section view	45
4.18	Bunn skrog fra siden	45
4.19	Toppskrog ovenfra	46
4.20	Toppskrog nedfra	46
4.21	Oversiden av støttestruktur	46
4.22	Underside av støttestruktur	46
4.23	støttestruktur montert i bunn skrog	47
4.24	Propguard utkast	47
4.25	Propguard med ekstra støtte	47
4.26	Propguard endelig model	48
4.27	Intel RealSense Depth Camera D435.	52

4.28	Brukte motorer	55
4.29	Kretsskjema for Drone Arena PCB	56
4.30	Strømbeskytelseskrets for kretskortdesignet	57
4.31	PCB tegning sett fra oven med top silkelag	58
4.32	Sammenstilling av Aktuator-skuff	59
4.33	Oversiktstegning av kolinger i aktuatorskuff	59
4.34	Monteringstøtter for aktueringskuff	60
4.35	Runder per encoder per tid	67
5.1	Glide-tester ble utført på flere materialer, to av de fremvist her.	69
5.2	Falltest med maulet flyging i arenaen	70
5.3	Flygetest i forskjellige miljøer	71
5.4	Oppsett for kameratest	71
5.5	Åpen dag sammenstilling, Dekan Michael Rygaard Hansen flyr dronen	72
5.6	QI-lading test oppsett	73
6.1	Resulatvinkler framstilt som et linjediagram	75
6.2	Falltest inni arena	76
6.3	Falltest ute på grass	76
6.4	Von Mises ved fall fra 1600 mm	77
6.5	Områder hvor von Mises er over 10MPa	77
6.6	Bevegelse i skroget [mm]	78
6.7	Utklipp fra flygning i arena føre kolisjon	78
6.8	Utklipp fra flygning ute	79
6.9	Dybdemåling med RealSense D435	79
6.10	Dybdemåling med drone stasjonert for lading eller letting	80
7.1	Sammenlikning mellom mangekanter	83
E.1	Teknisk tegning av ArenaV4	108
E.2	Teknisk tegning av Actuator platformen assembly	109
E.3	Teknisk tegning av HjorneV3R	110
E.4	Teknisk tegning av HjorneV2L	111
E.5	Teknisk tegning av Top-ring	112
E.6	Teknisk tegning av Bunnplate-del1	113
E.7	Teknisk tegning av Bunnplate-del2	114
E.8	Teknisk tegning av Bunnplate-del3	115
E.9	Teknisk tegning av ToppHjorneV3	116
E.10	Teknisk tegning av MovingStickV2	117
E.11	Teknisk tegning av top-stick	118
E.12	Teknisk tegning av Beskyttelsesvegger	119
E.13	Teknisk tegning av Bunring	120
E.14	Teknisk tegning av convstand	121
E.15	Teknisk tegning av MidSuport	122
E.16	Teknisk tegning av platform	123
E.17	Teknisk tegning av PowerBack	124
E.18	Teknisk tegning av PowerStand	125
E.19	Teknisk tegning av RotTannhjul	126
E.20	Teknisk tegning av RPIstand	127
E.21	Teknisk tegning av Sideplate	128
E.22	Teknisk tegning av SideplateLiten	129
E.23	Teknisk tegning av SlideV1	130
E.24	Teknisk tegning av ToppPlate	131
E.25	Teknisk tegning av batteriholder	132
E.26	Teknisk tegning av speilet batteriholder	133

E.27 Teknisk tegning av Midtplate	134
E.28 Teknisk tegning av skrogbunn	135
E.29 Teknisk tegning av skroglokk	136
E.30 Teknisk tegning av propgard	137

Tabeller

3.1	Tabell over score for de ulike konseptene.	30
6.1	Skrogets egenvekt og etter opplasting for test	74
6.2	Resulatvinkler: gjennomsnittsvinkel (måleserie)	74
6.3	Maks og min akselerasjoner(g) i alle akser under falltest	75
6.4	Avleste verdier under QI-test	80
D.1	Bill of Materials for full assembly	106
D.2	Bill of Materials for full assembly 2	107

Sammendrag

Det foregår i dag en svært rask utvikling innen automatiserte maskiner og roboter. Likevel er reinforcement learning i samspill med droner et lite utforsket område. Flightcontrollere trent opp av reinforcement learning er i noen grad blitt forsøkt utviklet, men er foreløpig kun testet i simulerte miljøer. Testing i et fysisk miljø er helt nødvendig for at en flightcontroller med kode basert på reinforcement learning skal være utviklet tilstrekkelig og være trygt nok for å kunne bli tatt i bruk i andre produkter og innen andre sektorer.

Internt på Universiteter i Agder jobbes det parallelt med flere prosjekter hvor droner og flight-controllere i samspill med reinforcement learning utforskes. Disse prosjektene vil kunne gjøre banebrytende arbeid med tilgang på et fysisk miljø hvor en drone kan trenere på en situasjon som er svært nærliggende uassistert flyging.

Viktigheten av å skape et sikkert miljø både for dronen som trener, og for mulige tilskuere, er et av hovedfokusene med tanke på design. Gjennom valg av materialer og utforming er det forsøkt å skape et miljø som muliggjør både trening over lengre tid mulig samt begrense ødeleggende aerodynamiske effekter. Drone framen er designet spesifikt med tanke på produksjon i polypropylen plastikk, da dette materialet gjør deler svært motstandsdyktige mot slag og dermed forlenger brukstiden til delen. Dette aspektet er balansert opp mot behovet for å å skape et mest mulig aerodynamisk design for flyegenskaper og flytid.

3D-printing, og mulighetene dette gir for produksjon av deler med avanserte og komplekse former, er blitt tatt i bruk for produksjonen av sentrale deler. Drone arenaer er ment til å kunne brukes som utstillingsobjekt på offentlige steder og et visuelt pent utsende er derfor en sentral del av designet, som passer 3D-printede deler godt.

Drone Arenaen består hovedsakelig av tre deler, den mekaniske konstruksjonen av arenaen, en komplett drone med styringssystem og flere aktuatorer styrt av en styringsdatamaskin via kamera-syn. Kode for implementasjon av maskinlæring og kommunikasjonsplattform mellom enhetene, er ikke implementert på dette stadiet, men tester på enkeltdeler av systemet har gitt lovende resultater. Før Drone Arenaen kan tas i bruk må det gis tid til utvikling for kommunikasjonsplattformen. Konklusjonen er derfor at plattformen ser ut til å gi et godt grunnlag for et system for trening av en reinforcement learned flightcontroller.

1 Introduksjon

1.1 Droner som en mulig hjelper i samfunnet

Autonome kjøretøy, enten det er snakk om fly, bil eller andre plattformer, er i dag på veg til å åpne opp for helt nye måter å løse samfunns oppgaver og om mulig danne en ny samfunnsstruktur [1]. Mulighetene for bruken av disse er mange, autonome robotarmer har i mange år blitt mer og mer vanlige innenfor industrien for utførelse av tungt arbeid. Som et resultat av økt presisjon innen sensor teknologi samt muligheten for stadig mer avansert styrings programvare, har prosjekter med mål om å automatisere større og mer komplekse prosesser oppstått. Eksempler på dette kan ses i selskaper som Ocean Infinity [2] som benytter seg av en flåte med store autonome skip for rekognosering til sjøs. Eller den autonome båten som skal transportere containere mellom Herøya og Brevik for frakt av kunstgjødsel [3]. Ettersom teknologien blir stadig mer avansert åpner det også for flere områder hvor autonome kjøretøy kan implementeres.

Bruken av droner og multicoptere er et felt hvor store framskritt er gjort på teknologien i løpet av få år. Bruken har lenge vært aller mest benyttet innen militærteknologi, hvor store flygende droner blir benyttet for overvåkning eller som bærere for våpen [4]. Disse dronene har stort sett en pilot stasjonert på en bakkestasjon og styrer farkosten manuelt. Mindre risiko for menneskelige tap har likevel vært med å senke barrierer og forandret strukturer i moderne krigføring.

For sivile formål har utviklingen kommet noe kortere, men store selskaper som Amazon har gjennom flere år jobbet med et leveringssystem ved hjelp av autonome droner. Også innenfor andre sektorer benyttes i dag droner for eksempel innenfor inspeksjon av kraftledninger og for søker og redning. Dette er områder hvor bruken av droner kan senke risikoen betraktelig. Systemene begrenses likevel betraktelig av kravene til helt eller delvis manuell styring.

Utfordringen ligger i dag mye i begrensningen ved menneskelige produserte koder. Slike koder har en tendens til å være massive, hvor en person ikke vil klare å ha oversikt over hele koden. Gjennom smart koding vil man til en viss grad kunne skape en kode som har muligheten til å reagere på et stort antall situasjoner, men koden vil fort få problemer i den virkelige verden hvor situasjoner har en tendens til å være varierte og sjeldent helt like [5].

Kunstig intelligens eller "Artificial intelligence" er et annet felt som i dag forskes mye på og som ser ut til å kunne revolusjonere flere felt. Reinforcement learning, som er et type kunstig intelligens, hvor systemet utsettes for gjentatte situasjoner over tid lager sin egenproduserte kode og handlingsmønster som best tilpasser seg de omgivelsene den blir utsatt for basert på belønninger. Slike systemer benyttes i dag av store selskaper som blant annet Tesla som samler inn store mengder data for å så benytte disse til å utvikle sine egne algoritmer for bedre og bedre autopiloter. Denne metoden hvor koden selv utvikler sin egen kode har muligheten til å skape en kode som i mye større grad enn mennesker kan handle riktig i et betraktelig større antall situasjoner.

Reinforcement learning innenfor drone feltet er i dag lite utforsket, og dagens droner styres vanligvis av en autopilot programvare som baserer seg på menneskeskapte koder og PID kontrollere. Forsøk med reinforced flight kontrollerer er i noe grad forsket på og testet men da kun i et simulert miljø [6]. Hvilke utfordringer og fordeler implementeringen av et slikt system ville hatt i den virkelige verden er lite utforsket.

1.2 Formål og bakgrunn for behov ved UIA

To design and create an container which provide sensordata and security for both observers and an autonomus drone training an AI flightcontroller.

Det har gjennom flere bachelor prosjekter vært jobbet med et droneprosjekt som har et overordnet mål om å kunne lage et effektivt system for oppsett, bruk og lagring av en sverm med droner.

Dette prosjektet har sett på flere muligheter rundt design med tanke på produksjon, samt innmat og styringsplattformer. Også mekanisk design som omhandler utskytermekanismer og kommunikasjonsplatformer er noe som har vært gjort arbeid på tidligere [7][8].

Det eksisterer i dag prosjekter på universitetet som utforsker muligheten for reinforcement learning med tanke på droner. Disse er fortsatt hovedsakelig testet i simulerte miljøer, men ved mulighet for testing i et fysisk miljø kan dette gi verdifull informasjon. Som videre vil være et steg på veien til implementasjon i enheter som kan tas i bruk for å løse prosesser.

Et slikt system vil ha mulighet til å kunne benytte et stort antall droner som snakker sammen for å løse oppgaver på en effektiv måte eller dekke store arealer, uten behov for store menneskelige ressurser. I dag benyttes drone svermer til blant annet ”simulert” fyrverkeri hvor disse flyr med lys i formasjoner. Mulighetene for et slikt system kan likevel tenkes å kunne være svært nyttig i for eksempel søk og redning av mennesker.

En drone under opplæring vil miste kontrollen og oppføre seg uforutsigbart og uhensiktsmessig. Selv med liten vekt vil det likevel kunne være en fare og spesielt med tanke på spinnende propeller som har mulighet til å lage dype sår. Et miljø som derfor tar vare på potensielle tilskuere samtidig som omgivelsene holdes tilnærmet så likt som ved virkelige forhold for flyving er derfor nødvendig.

Et system som krever en høy grad av vedlikehold vil ikke være realistisk å bruke da mange forsøk er nødvendig for å utvikle et godt nettverk for en reinforcement learning kode. Et system som over mange sykluser kan stå å gå alene er derfor nødvendig. For å realisere dette viste det seg nødvendig å tilpasse drone skroget, både i form av design og materialer.

1.3 Toppnivåkrav

Før designprosessen ble det satt 10 MÅ krav, henholdsvis 7 for design av arenaen og 3 for designet av dronen. I tillegg ble det satt 3 ikke må krav. Disse ble laget i samarbeid med veileder og med tanke på potensielle sluttbrukere av produktet.

Arena MÅ krav

1. Beskytte observatør mot flygende drone
2. Trådløs og automatisk lading av drone (QI)
3. Dronen har samme utgangsposisjon før hvert forsøk
4. Dronen inne-blir i arenaen så lenge programmet kjøres
5. Arenaen skal kunne kommunisere med dronen
6. Skal være porterbar og demonterbar
7. Hardware specs må kunne håndtere deep-learning og opptrenings av AI

Drone MÅ krav

1. Vekt mindre enn 250 gram
2. Trådløs lade-oppkobling
3. Skal kunne operere uavbrutt i minst en måned

Ikke MÅ krav

1. Mest mulig kompatibelt med tidligere iterasjoner av dronesverm-prosjektene
2. Rent og stilig visuell utforming som skaper blikkfang

3. Dronen bør være vantett

Noen av kravene ligger tett opp mot selve ideen bak Drone Arenaen og definerer dens funksjon. Dette gjelder spesielt krav en og fire for arenaen. Med tanke på reinforcement learning er krav tre viktig for å oppnå best mulig læring og for dette er også krav fem og syv for arenaen en sentral del. Krav to for arenaen og to for dronen henger sammen med krav tre for dronen og gir et konkret mål for hvilken støt og vedlikehold det skal designes for. Krav seks for arenaen er viktig med tanke på designet og hvilke løsninger som prioriteres.

Krav en til dronen henger sammen med Ikke MÅ krav nummer en da dette er en viktig grense satt av luftfartstilsynet og har mye å si for hvordan dronen kan brukes og hvem som kan bruke den [9]. Ikke MÅ krav nummer 2 legger visse føringer rundt designet for at det skal fungere som et utstillingsobjekt og ligger også tett opp mot Arena krav 6 for at den skal kunne flyttes til ulike lokasjoner.

I tillegg har flere av de tidligere prosjektene hatt et bør krav om at droneskroget skal være vanntett [8][10]. Dette er i utgangspunktet ikke en del av Drone Arena kravene, men grunnet nødvendig redesign, har dette blitt fokusert på for å legge til rette for at designet kan benyttes i senere forgreninger hvor dette igjen blir løftet fram som et krav.

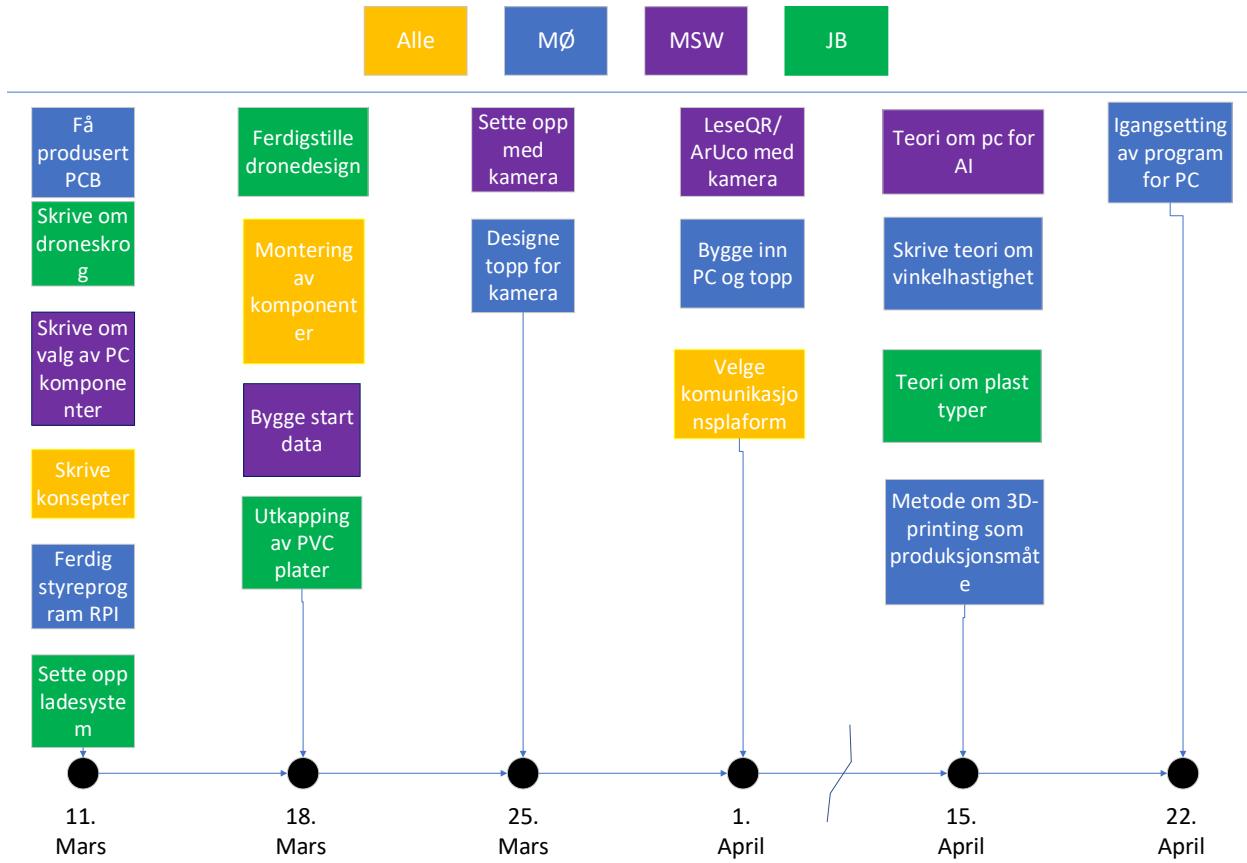
1.4 Avgrensing

Da ferdigstilt produkt er en lengre prosess som også krever noe mer kunnskap om hvilke plattformer som ønskes for reinforcement learningen er det i denne oppgaven satt som mål å utvikle en arena hvor en UiA drone kan flys manuelt for testing av krasj og enkelt systemer kan kjøres for funksjons tester. Det er ikke et mål å kunne kjøre tester basert på reinforcement learning eller fullstendig autonom arenaaktuator, men legge alle delsystemene til rette for dette.

1.5 Arbeidsmetoder

1.5.1 Planer

Under prosjektet har det vært tatt i bruk flere måter på planleggingen av arbeidet, for å best mulig tilpasse arbeidet etter behovet i prosessen. Det ble tidlig laget en overordnet tidslinje for hvilke viktige frister og milepæler som skulle nås under prosjektet. En del komponenter og produksjoner ble forventet å ha noe lang leveringstid så disse ble satt opp på tidlige datoer for å sikre at disse kom i tide. Denne oversikten ble laget i Miro som fungerer som en online tavle.



Figur 1.1: Opgavefordeling med tidslinje fra midtfasen av prosjektet

I midtfasen av prosjektet ble enkeltoppgavene mer tydelige og en oversikt hvor disse ble fordelt på personer og satt opp med frister ble laget, se figur 1.1. Det ble også her benyttet plattformen Miro. I denne fasen ga det god oversikt for alle i gruppen hva som måtte prioriteres og det var lett å se hva slags prosgresjon som krevdes til alle tidspunkt.

I siste fase av prosjektet ble skrivingen prioritert. Det ble da laget et digitalt post-it lapp system i Miro hvor alle delkapitler ble skrevet opp på hver sin lapp. Disse ble så fordelt og satt under navnet på den personen som hadde ansvaret for å skrive denne delen. Når seksjonen var skrevet ble lappen flyttet vekk og i en ferdigkategori. Dette ga god oversikt og flyt i arbeidet. Arbeid som dukket opp underveis ble ført på nye lapper og fordelt.

1.5.2 Kommunikasjon

Det har blitt avholdt oppdateringsmøter på mandager under arbeidet med primær veileder og gruppemedlemmene, og noen med sekundær veileder tilstede. Disse har blitt holdt hver uke i perioder med høy grad av utvikling. I perioder hvor arbeider har båret preg av mer detalj design har de blitt holdt etter behov.

Digitale møter har primært blitt holdt på Microsoft Teams og meldingskommunikasjon mellom gruppen og veileder har foregått på Discord. Denne kommunikasjonen har vært viktig da veileder også representerer den potensielle brukergruppen. Flere fysiske møter har blitt avholdt med mindre grad av planlegging. Progresjonen har underveis også blitt vist fram til potensielle brukere for tilbakemeldinger for å forbedre designet i utviklingsfasen.

1.6 Budsjetttramme

Drone Arenaen er et relativt stort og nytt konsept og kostnadene ved et slik prosjekt er betydelige. Midler er tilgjengelige for realisering av fysisk produkt, med noe begrensete muligheter. Flere avgjørelser med tanke på komponent valg, testing og design valg er derfor preget av at beste løsning

ikke nødvendigvis er tilgjengelig. Kompromisser som å utvikle software på en midlertidig underdimensjonert maleware har påvirket det endelige resultatet. Dette vil realisere seg i komponent valg og design gjennom oppgaven som et viktig aspekt ved alle avgjørelser.

For å få ferdigstilt et fysisk produkt i størst mulig grad, har det blitt benyttet komponenter som var lett tilgjengelige gjennom delelageret ved Mektronikk labben. Det ble også tatt i bruk gjenbrukte komponenter, med liten grad av produksjonsnummere og datablad. Med tanke på et endelig produkt er dette uheldig, men helt essensielt for å få til prototyping. Løsninger som gjør at deler enkelt kan byttes ut til andre deler ble derfor lagt vekt på for å kompensere for dette og legge til rette for utbytte til bedre komponenter i senere iterasjoner.

Det har blitt lagt ned mye arbeid i å finne gode løsninger for å få til prosjektet innenfor rammene, som har materialisert seg i et svært bra sluttprodukt. Tilslutt rettes det en takk til alle som har lånt komponenter, deler, smøremiddel og likende til prosjektet. Uten dette hadde prosjektet på ingen måte gått rundt med tanke på budsjett, design og prototyping.

2 Teori

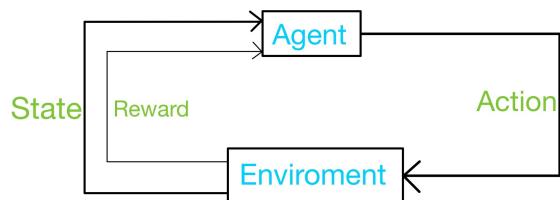
2.1 Reinforcement learning i fugleperspektiv

Reinforcement Learning, forkortet RL, er en underkategori av Maskin Læring, som videre underligger hovedkategorien kunstig intelligens. Maskin læring kan bli definert som beregningsmessig metoder som bruker erfaring til å forbedre prestasjon eller gjennomføre presise forutsigelser [5]. Erfaring i dette tilfellet referer til opparbeidet informasjon som er analysert. Reinforcement learning på den andre siden baserer seg på de samme prinsippene, men i praksis fungere det litt anderledes. Innenfor RL vil programvaren eller det som lærer, kalles gjerne agent, jobbe basert på Rewards, også kalt belønninger.

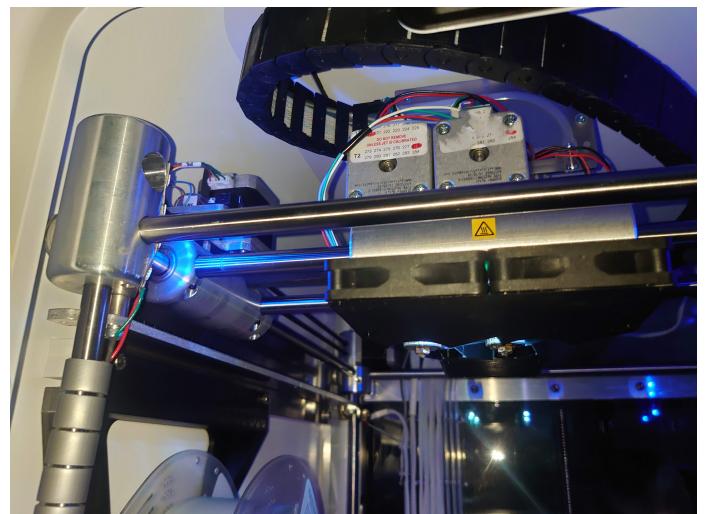
Disse blir gitt fra Enviromentet (omgivelsene) ved å gjennomføre Actions (handlinger) og motiverer Agenten for å komme nærmere målet [11]. Det vil si utifra feedbacken Enviromentet gir til Agenten vil justeringer bli gjort i henhold til forutsigelsene kalkulert av Agenten. Videre vil ny input bli formidlet og videre justering foretatt, i praksis vil dette bety prøving og feiling gjentatte ganger over et tidsrom. Agenten jobber utifra å maksimere rewarden og vil dermed fastslå den beste handlingen, eller retningslinje, for å oppnå målet [5]. I forbindelse med RL er det viktig å være klar over at informasjonen som Agenten får fra Enviromentet er umiddelbart basert på rewarden den akkurat fikk, og avhengig av Actionen den foretok. Enviromentet gir ikke noen fremtidig- eller langvarig reward. Dette medfører at et viktig aspekt ved reinforcement learning er å ta hensyn til forsinkede rewards eller i noen tilfeller straff. Det vil si at agenten står ovenfor et dilemma mellom å utforske ukjente områder eller gjennomføre actions for innhenting av mer informasjon om Enviromentet og belønningene, dermed utnytte kjent informasjon for å optimalisere rewarden. Innenfor reinforcement learning er dette kjent som "The exploration versus exploitation trade-off" [5].

2.2 3D printing som produksjon

3D-printing er en relativt ny produksjonsmetode som første gang ble patentert av 3D-systems i 1984 [12]. Dette var en teknikk hvor en laser stråle ble benyttet til å stivne en flytende plast masse. Ettersom lagen var stivnet flyttet flaten seg nedover slik at et nytt lag kunne lages. Som samlebetegnelse blir i dag teknikkene kalt additiv tilvirkning, og beskriver alle produksjonsmetoder hvor deler skapes ved at lag på lag bygges ved hjelp av materialer stivner. Det finnes en lang rekke måter å produsere disse lagene på, men majoriteten av printere kan deles inn i gruppene FFF (fused filament fabrication), SLA (Stereolithography) og SLS (selective laser sintering) [13]. Disse gir ulike muligheter med



Figur 2.1: Statediagram for Reinforcement Learning.

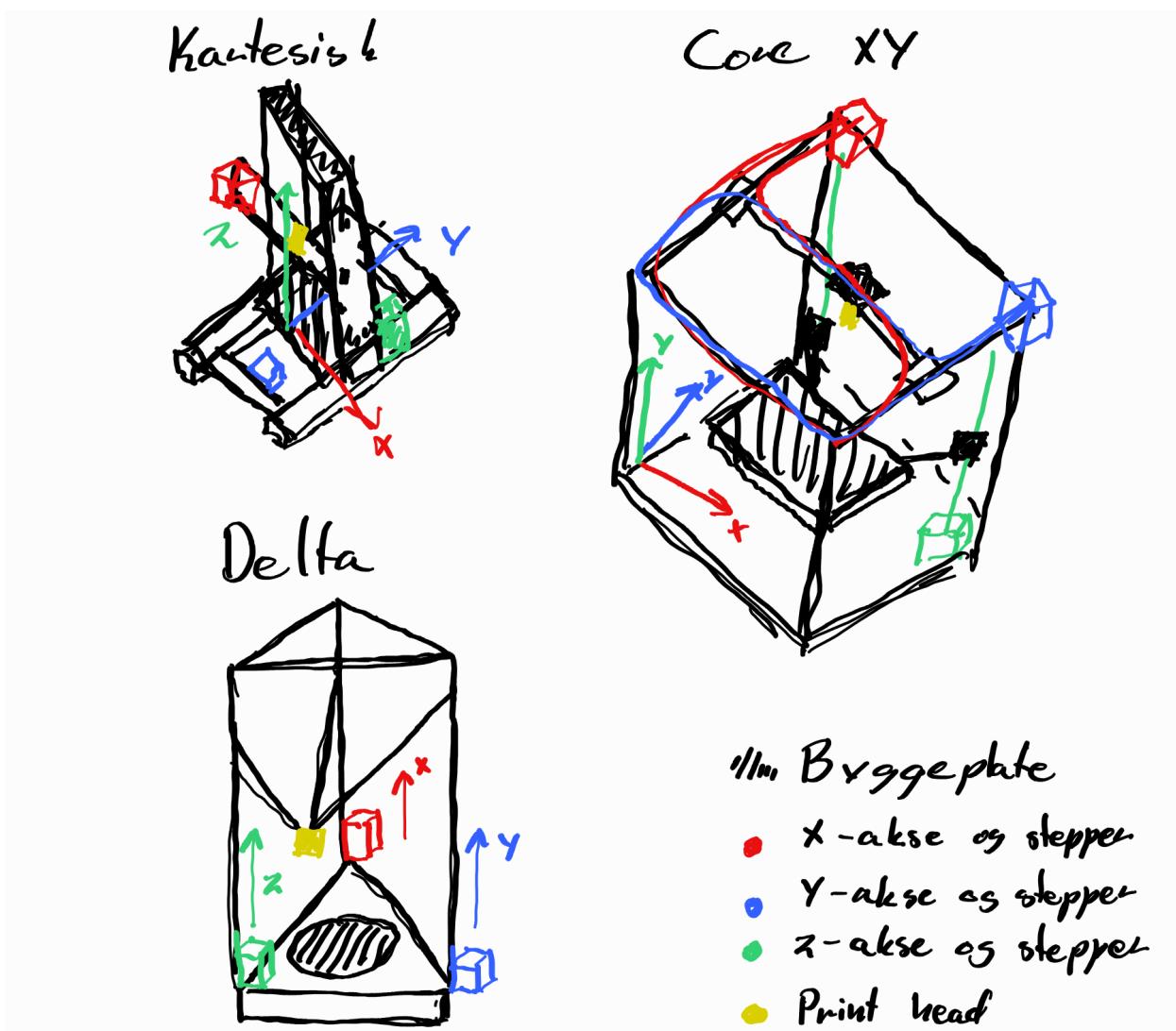


Figur 2.2: Kartesisk printer med dual nozzle setup

tanke på design, men også sluttproduktets styrke. En rekke mer inngående testing på dette har blitt utført og kan gi en pekepinn på hvilke produksjonsmetode som passer best for produktet [14].

2.2.1 Fused Filament Fabrication

FFF eller FDM printing er i dag den mest utbredte teknologien med størst variasjon og enklest å komme i gang med [13]. FFF og FDM (filament deposition modeling) er i all enkelhet det samme, men begge navnene benyttes forsatt parallelt da Stratasys patenterte FDM i 1989 og FFF ble brukt for å beskrive samme teknologi fra andre produsenter [15]. Teknikken går ut på at termoplast føres inn i en oppvarmet dyse hvor den smelter til en semifast form, dysen kan ses på figur 2.2. Trykket i dysen bygges opp av filamentet som dyrtes inn og presses så ut gjennom en åpning. Størrelsen på åpningen avgjør hvor mye plast som kan dyrtes ut av gangen og er derfor en av hovedfaktorene for hastigheten printen kan utføres i. Lagene legges på en print plate som under printen kan senkes slik at neste lag vil bli liggende i riktig tykkelse oppi det første. Større printere tiltenkt industriell produksjon har ofte et oppvarmet kammer som det printes i. Dette er for noen mer avanserte plast typer helt essensielt for å få gode resultater, imidlertid brukes en oppvarmet plate istedenfor hovedsakelig på grunn av betydelig redusert pris [15]. Noen printere er spesielt bygget for å kunne kjøpes billig og har derfor ikke oppvarmet plate.



Figur 2.3: Tre forskjellige printer typer illustrert med Kartesisik, Core XY og Delta.

Den oppvarmede dysen beveger seg rundt på byggeplaten som oftest aktuert av stepper motorer. Det finnes flere forskjellige måter å bevege dysen og platen i forhold til hverandre. Den mest

tradisjonelle er å la dysen kunne bevege seg i x og y retning, mens byggeplaten senkes nedover ved hjelp av en skrue, såkalt kartesisk struktur. Dette vil man for eksempel se på Ultimaker sine printerer, som er en anerkjent produsent [16]. På grunn av kostnadene ved strukturen rundt det å holde byggeplaten stødig samtidig som den beveges nedover er en billigere variant utviklet hvor byggeplaten beveges i x, dysen i y og også oppover i z ved å flytte y akse føringen oppover. Dette blir i dag benyttet på en rekke ”standard” v-slot profil ramme printere som man får for en ikke alt for høy pris. Eksempler på dette kan være Creality CR-10S Pro og Artillery Sidewinder X2 [17] [18].

For mer presisjon og hurtigere printer benyttes mer moderne og komplekse bevegelses strukturer, som for eksempel Core-XY og Delta [19]. Disse typene benytter andre strukturer og motorplasseringer for å oppnå andre egenskaper. Core-XY baserer seg på en tradisjonell kartesisk oppbygging, men plasserer x og y motorene på selve rammen. Via et komplekst reimsystem styres x og y aksen ved hjelp av koordinering av disse to reimene. Ved å plassere motorene vekk fra printer hodet, begrenses vekten betraktelig og hastighet og presisjonen økes. Delta er en annen funksjon og baserer seg på en konfigurasjon inspirert av delta roboter. Hodet henger i tre armer som styres ved å justere høyden i festepunktet. Dette gir høy presisjon uten å øke antall motorer, baksiden er en begrenset rekkevidde i x- og y-aksen og en rund byggeplaten. Konstruksjonen tar også mye plass i høyden.

Filamentet kommer stort sett i ruller med tråder, med en standard diameter på 1.75 mm og 2.85 mm. Dette gjør at det finnes et stort utvalg av produsenter for filament alle med litt forskjellige egenskaper og printer karakteristikkene. Disse karakteristikkene kan ofte også påvirkes av både oppbygningen og omgivelsene rundt printeren og en kombinasjon av kjent informasjon sammen med empirisk testing i det aktuelle miljøet er ofte nødvendig for optimalt resultat. Også oppbevaringen og forbehandling av filament kan ha mye å si for kvaliteten. Spesielt ved bruk av mer avanserte materialer, dette blir mer inngående diskutert i kapittel 2.3. Utvalget er imidlertid svært bredt og man kan finne filmenter som inneholder karbon for økt styrke eller på den andre siden TPU som er et mykt materiale og kan benyttes som for eksempel støtdemping [20][21].

2.2.2 Stereolithografi

SLA, eller ”resin printing” som det ofte blir kalt, er en av de nyere formene for 3D-printing. Teknologien baserer seg på den flytende substansen kaldt resin som helles ned i en skål som ligger oppå en skjerm. Resin substansen er inspirert av harpiks sine egenskaper til å stivne når det blir utsatt for ytre påvirkning [22]. Resin som benyttes til 3D-printing er UV lys sensitivt og vil herde fort ved eksponering for dette. Ved printing blir derfor en plate senket ned til en kjent avstand fra bunnen av skålen før UV-lyskilden i bunnen slås på og skjermen åpnes [13]. Skjermen kan dermed vise sorte piksler der UV-lyset ikke skal treffe og ”åpnes” for å slippe gjennom de stedene hvor resinen skal stivne.

Etter en kjent eksponeringstid løftes byggeplaten opp et stykke før den igjen føres ned mot bunnen for nytt lag. Fordelen med systemet er blant annet at det kun kreves en akse med aktuering, nemlig den for å heve og senke byggeplaten. Den mekaniske enkelheten bidrar til å kunne senke prisen på slike printere. Skjermens oppløsning er hovedfaktorer for printens kvalitet og er derfor en sentral del av SLA printere. Hvor små detaljer som kan produseres avgjøres av størrelsen på pikslene. I slike maskiner benyttes opp til 8K oppløsnings skjermer på en relativt liten overflate, noe som kan gi et svært stort detaljnivå og representerer en av de store fordelene med SLA printere. UV-kilden vil være avgjørende for hvor fort og effektivt printeren er i stand til å herde lagene.

Et produkt laget i en SLA printer krever betraktelig mer



Figur 2.4: Anycubic Photon Zero SLA printer

etterbehandling enn en FFF produsert del. Resin er i sin ubehandlete form forsatt til en viss grad helseskadelig og uten maks styrke. Modellene må derfor først vaskes i isopropanol eller liknende alkohol for å løse opp all medfølgende ubehandlet resin. Videre etterbehandles modellen med UV-lys for å få en sterk og helt ufarlig overflate. Denne prosessen kan være vanskelig og bør gjøres i sikre omgivelser som i et avtrekkskap.

Den endelige delen har en betydelig større hardhet sammenliknet med en modell laget med en FFF printer. Den vil kunne ha betydelige finere detaljer noe som gjør produksjonsmetoden svært egent til for eksempel produksjon av tannhjul. Delene blir en god del sprøere sammenliknet med FFF produserte deler, og vil derfor under trykk eller slag ha en tendens til å sprekke opp eller splintre. Deler som skal oppleve stort trykk vil derfor være lite egnet for SLA printing.

2.2.3 Selective Laser Sintering

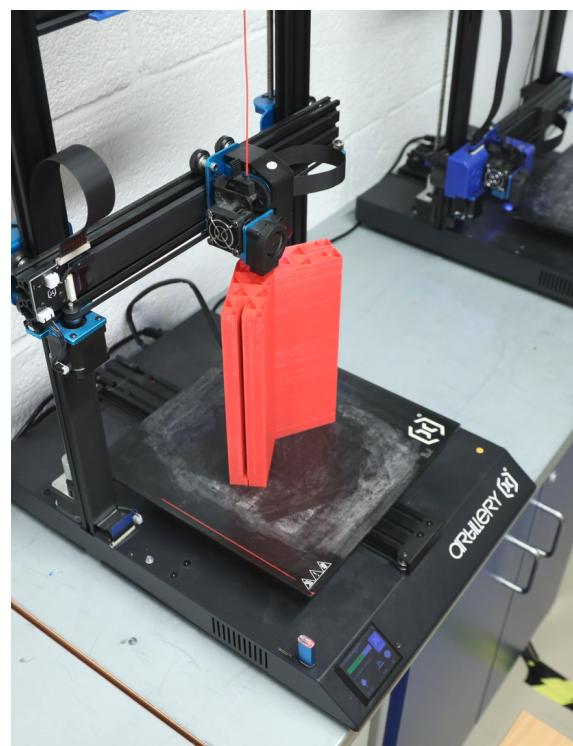
SLS printing benytter en kraftig laserstråle til å smelte partiklene sammen fra et ”støvlag” av råfilament. Teknologien ligner i seg selv svært på SLA printing hvor en byggeplate senkes ned mot en overflate hvor den eksponeres for å stivne. Måten det gjøres på med en laser gjør imidlertid at partiklene får mulighet til å synke godt inn i hverandre og kan fjerne nesten all lagdeling [13]. Delene som produseres er derfor tilnærmet like sterke i alle retninger og ikke bare på tvers av print-retningen. Man har også muligheten til å benytte et bredt spekter av materialer på grunn av enkelheten ved å male opp stoff for så å smelte dem sammen igjen. Prosessen krever relativ farlig utstyr for å skape energien som trengs, samtidig som det frigjorte støvet kan være helsefarlig å puste inn.

Grunnet pris, kompleksitet og faremomenter benyttes denne teknikken i dag stort sett i industrielle prosesser og av spesialiserte firmaer. Teknikken har imidlertid potensiale for fremtidig produksjon i materialer som i dag bare lar seg gjøre ved støping, som rent glassfiber eller karbon fiber.

2.2.4 Med hensyn på produksjon

Ettersom teknologien de siste årene har utviklet seg svært fort både med tanke på presisjon, hastighet og materialer har bruksområdene gått fra å være forskningsbasert og hobby aktivitet til å kunne benyttes som en faktisk produksjonsmetode. Det finnes i dag en rekke stor produsenter av såkalte industri printere som skal kunne brukes til å produsere deler og liknende ved et akseptert nivå av vedlikehold. 3D Systems og Stratasys er noen av de mest kjente produsentene av slike printere i dag, og leverer et bredt utvalgt etter forskjellige behov [23][24]. Disse printene er i imidlertid svært dyre for enkeltpersoner eller mindre bedrifter.

Som en del av fokuset på lokal produksjon har også mer hobbybaserte 3D printere blitt løftet fram som en måte å kunne produsere spesialdesignede deler raskt og enkelt, uten lang og tidkrevende frakt. Figur 2.5 er et eksempel på hvordan dette kan se ut. Slik printing kan bidra til mer effektivt og miljøvennlig produksjon [25][26]. Skal en del printes på en hobbybasert printer vil en viss tanke på hvordan den skal produseres være nødvendig for å få et tilfredsstillende resultat. Dette vil alltid være nødvendig ved design av deler for produksjon, men på grunn av at 3D printing nettopp skiller seg så drastisk fra konvensjonell produksjon, legger dette helt andre krav til designet.



Figur 2.5: Del til arenaen under produksjon på en Artillery SW-X2

Dette gir også videre utfordringer for fremvisning som konvensjonelle tekniske tegninger, da strukturer som er små eller mange men ujevne er vanskelige og uoversiktlig. Mer moderne filformater som .stl, .obj og .3mf tar i noen grad over for tekniske tegninger for slike komponenter.

For masseproduksjon er fremdeles støping langt billigere og lar seg gjøre i større skala. 3D printing har i dag i noe grad blitt populært som fenomen og man vil finne produkter som gir seg ut for å være 3D printet selv om dette faktisk sjeldent er tilfellet [27]. Ved mindre antall og komplekse deler vil derimot 3D printing kunne dekke et behov for rask og enkel delproduksjon.

2.2.5 Med hensyn på droneproduksjon

3D printing muliggjør komplekse strukturer under designet av dronen. Kravet om at dronen skal veie mindre enn 250 gram 1.3 er mulig å oppnå ved printing på de aller fleste 3D printere. Prototyping av dronen ved bruk av 3D printere gjør det lett å tilpasse designet til komponentene og teste forskjellige profiler. Dette er både raskt, enkelt og man slipper å vente på produksjonskøer og frakt. Blant annet er aerodynamikken en vanskelig variabel som gjør det utfordrende å treffe på skrogdesignet ved få iterasjoner. Det finnes også mange komponenter som kan brukes i dronen, samt at det antagelig vil være nødvendig å legg til eller bytte ut komponenter på et senere stadium. Dette kan da raskt gjøres ved å lage ett nytt design for 3D-printing, og man kan ha delen klar iløpet av en dag eller to.

2.3 Plast typer

2.3.1 Innvirkning på material egenskaper ved 3d print

3D printere gjør det mulig å bruke forskjellige materialer, men det er flere faktorer som spiller inn på egenskapene til materialene gjennom 3D printing. Resultatet på overflaten av en 3D print blir ujevn og svekker de mekaniske egenskapene til plasten ved at det danner seg flere stresskoncentrasjoner [14]. Siden en 3D printer produserer en del ved å legge lag på lag, blir det også svake punkter i bindingene mellom lagene [28]. Faktorer som temperatur, hastighet, kjøling og mengden support vil påvirke hvor godt lagene smelter sammen. Og dermed være avgjørende for styrken til delen [14]. Disse parameterene blir bestemt i slicern, som er et program hvor man kan laste inn .3mf og .stl filer, deretter beregner denne hele banen som printeren skal gjennomføre. Banen lagres som en .gcode som printeren kan lese.

2.3.2 PLA

PLA er et av de mest brukte filamentene når det kommer til 3D printing og er laget av termoplasten Polylactic Acid [29]. Dette filamentet er blant de lettere materialene å printe med, grunnet at det fester seg godt til bygge platen og har svært lite krymping. For å printe PLA er det ikke nødvendig å ha en printer med oppvarmet byggeplate, men det vil gjøre det lettere for modellen å feste seg. Byggeplaten bør da settes til en temperatur på rundt 40 til 60 grader, da vil modellen feste seg godt og fortsatt ikke være så varm at modellen begynner å få deformasjoner. PLA har også en lav temperatur på dysen fra 185 til 220 grader celsius. Filamentet har en egenvekt på 1.25 g/cm^3 og en del vil bli relativt massiv om den har 100% infill. Ved bruk av PLA vil printen oppnå en strekkfasthet på 63 MPa , men dette vil variere mellom forskjellige printere og forskjellige innstillinger i sliceren.

2.3.3 PLA varianter

Det finnes flere forsterkede varianter av PLA, blant annet har eSun PLA+ og Prima har PLA pro. Siden disse er en variant av PLA, deler de derfor de fleste materialegenskaper. PLA pro filamentet har en egenvekt på 1.24 g/cm^3 , men krever høyere temperaturer under printing enn vanlig PLA [30]. For å oppnå en god print bør byggeplaten ligge på en temperatur mellom 45 og 65 grader, og det kreves en temperatur mellom 210 til 250 grader på dysen avhengig av print hastigheten. Når det kommer til de mekaniske egenskapene har PLA PRO flere fordeler. Den har en strekkfastheten

på 110 MPa , en forlengelse på 160% ved brudd og en strekk modulus på 3310 MPa . PLA PRO er derfor mindre sprøtt og kan blant annet absorbere støt godt, uten å sprekke, dette er egenskaper som gir sterkere deler.

2.3.4 PP

PP filament er laget av polypropylen, og er en mye brukte plastikk til innpakning av mat, matbokser og plastikkbeholdere med hengslefunksjoner [31]. Under print er polypropylen blant materialene som trenger middels høy temperatur. Byggeplaten bør ligge på rundt 90 grader mens dysen trenger 230 til 250 grader. Polypropylen har både positive og negative egenskaper. Det er mykt ved tynne strukturer og absorberer derfor støt veldig godt, men det er samtidig vanskelig å oppnå en stiv struktur. Plasten har en egenvekt på 0.9 g/cm^3 og flyter derfor i vann og har også fordelen at det er vanntett. Filamentet er noe utfordrende å printe grunnet mangel på festeevne til bygge platen, og stor sammentrekning. PP filament fester seg best til seg selv og det kan derfor brukes en PP film på byggeplaten eller en tape laget av PP. Warpingen, betyr oppløfting fra byggeplaten grunnet sammentrekning og oppstår ved nedkjøling. Hvis modellen kjøles ned uproporsjonalt i tverrsnitte ne vil den kunne sprekke opp. Hvis det kommer kald luft inn på modellen vil den kjøles ned mer på innsiden enn utsiden. Printen vil da brettes oppover og etterhvert sprekke opp om det er tynne lag. For å unngå dette må printeren dekkes til slik at miljøet som det printes i beholder en stabil temperatur.

2.3.5 PETG

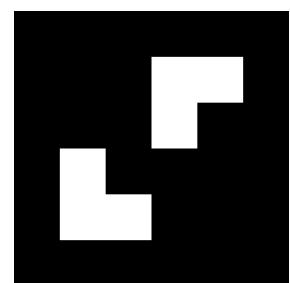
PETG er et filament laget av polyethylene og printes med noe høyere temperatur enn PLA. For å printe PETG bør dyse temperaturen ligge på 195 til 235 grader [32]. Byggeplaten bør optimalt sett ligger på 45 til 60 grader, men det er ikke nødvendig med varme i utgangspunktet. Egenvekten til PETG ligger på 1.27 g/cm^3 og har en strekkfasthet på 50 MPa .

2.3.6 TPU

TPU er ett filament som er middels vanskelig å printe, og krever flere finjusteringer i sliceren for å få gode resultater. Egenvekten til TPU er 1.21 g/cm^3 , har en strekkfasthet på 55 MPa og er laget av polyurethane [21]. For å printe med dette filamentet bør det unngås å bruke temperatur på byggeplaten, mens dysen bør settes fra 180 til 230 grader. Siden filamentet er fleksibelt vil det være både strekking og kompresjon i filamentet når det ekstruderes. Det er derfor en stor fordel å ha en direkte ekstruder eller dualekstruder på printeren. Med ett sånt oppsett vil det være kort veg fra ekstruderen og frem til dysen, som vil gjøre at strekningen i filamentet blir minimalt. Når TPU filamentet ligger åpent i luft vil det trekke til seg fuktighet, det vil resultere i en dårligere print. For å unngå dette bør filamentet derfor ligge i en filamenttørker eller legges inn i ovnen på 50 grader før bruk.

2.4 ArUco-markør

ArUco-markør er en del av samlebetechnelsen Fiducial-markers, og det kan sammenlignes med QR-kode og April-tags. En ArUco-markør inneholder noe mindre informasjon som fører til at bitmengden er lav. Denne typen markør er basert på et opensource bibliotek som benyttes for estimering av kameraposisjon og identifikatorer, ved bruk av kvadratiske markører satt sammen med en unik kombinasjon. Markørene er bygd opp med en ytre svart kant og et indre område med et binært mønster i hvitt [33]. Dette hvite mønsteret i midten er unikt for hver ArUco-markør, ofte fungerer dette som en ID og kan bli satt opp mot kommandoer som for eksempel å skrive ut posisjon. Et eksempel på en markør er fremstilt i Figur 2.6, dette er ArUco-markøren med ID 42. En markør kan inneholde informasjon av forskjellig grad, med mindre informasjon liggende i markøren vil det også



Figur 2.6: ArUco-markør med ID 42.

sette lavere krav til hvor god opplosning det må være på bildet. Om man har mer informasjon liggende i mønsteret vil sannsynligheten bli mindre for feillesning, men det krever altså betydelig høyere opplosning og det vil ta lengre tid å prosessere informasjonen.

Utseende til ArUco-markøren med svart kant og hvitt mønster gjør det enkelt for Maskinsyn å detektere den unike informasjonen i flere scenarier. Varierende rotasjon, annerledes lysforhold samt skalering vil være ubetydelig for resultatet. Dette fører til at disse type markørene har en rekke bruksområder, som blant annet kalibrering av kamera, estimere størrelse på et objekt, måle avstand, posisjonering i 3D og generell robotikk [34]. En annen styrke med ArUco er hvor godt etablert denne formen er, den finnes integrerte biblioteker i det fleste programmeringsspråk og er spesielt egnet for bruk gjennom OpenCV som er selve benchmarken for computer vision.

2.5 Kretskort linjebreddeberegning

Friksjon og varme vil oppstå ved all strømgjennomgang i ledere, sett bort fra superleder feltet. Denne friksjonen stammer fra hvordan elektrisiteten beveger seg gjennom det strømførende materialet og hvordan dette materialets oppbygging forholdsvis øker eller senker elektrisitetens muligheter for å bevege seg [35]. I metaller flyter delvis elektronene fritt mellom atomene som delvis deler disse. Dette kalles ledningsbåndet og er grunnen til at metaller har en relativt høy ledningsevne [36].

Det er likevel en viss grad av friksjon og denne vil skape varme da strømmen vil dytte på elektronene. Denne varmen kan bli et stort problem for elektrisk kretser i første omgang da dette bruker strøm på å utvikle unyttig varme, og i ytterste konsekvens kan føre til at ledere eller komponenter smelter eller tar fyr. Størrelsen på strømmen er den avgjørende fysiske faktoren i denne sammenhengen gjennom den faktum at denne nettopp er et resultat av likning 2.1 hvor dQ er ladningen som passerer et tverrsnitt på tiden dt [37].

$$I = \frac{dQ}{dt} \quad (2.1)$$

$$dR = \frac{dx}{\lambda A} \quad (2.2)$$

Likning 2.2 er for å regne ut den termiske resistansen i et tynt sjikt. Dette er avhengig av bredden i x , arealet og materialets ledningsevne λ . Ved å se på hvor mye friksjon som oppstår med hvor mye som forsvinner finner man den varmen som vil bygge seg opp i lederen. Dette er svært avhengig av hva som ligger rundt lederen samt hva lederen er laget av.

For kretskort design benyttes en standard for hvilke verdier som vil gi et godt estimat for varmeoppbygningen. Denne er utviklet av *ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES* og ligger som standard for utregningene [38]. Gjennom forkortede likninger vist i likning 2.3 og 2.4 kan den nødvendige bredden kalkuleres etter en typekonvertering [39]. Standard for brede på kobberlengder er som oftest beregnet i oz/ft^2 , og krever derfor en typekonvertering til mm basert på kunnskapen om tykkelse.

$$A[mm^2] = \frac{I[A]}{k \cdot Temp_{Rise}[deg.C]^b}^{\frac{1}{c}} \quad (2.3)$$

$$Width[mils] = \frac{Area[mils^2]}{(Thickness[oz] * 1.378[mils/oz])} \quad (2.4)$$

Grunnlaget for disse beregningene blir da hvor mye temperaturøkning man vil tillate i koblingene. Dette er også sterkt bundet opp til syklusen på kjøringen. Ved lange sykluser vil trådene få lang tid til å avgive varme og en mindre bredde kreves. Ved korte sykluser blir disse beregningene svært viktig da varme vil fortsette å bygge seg opp i systemet.

For linking 2.3 blir konstanter fra IPC-2221 [39][38] standarden benyttet og hvor henholdsvis $k = 0.024$, $b = 0.44$ og $c = 0.725$ for ytre lag med tilgang på luft og $k = 0.048$, $b = 0.44$ og $c = 0.725$ for indre lag i kretskortet. Disse verdiene har et gyldighetsområde med trådtykkelse mellom 0.5 og 3 oz/ft^2 [39].

2.6 Kamera krav

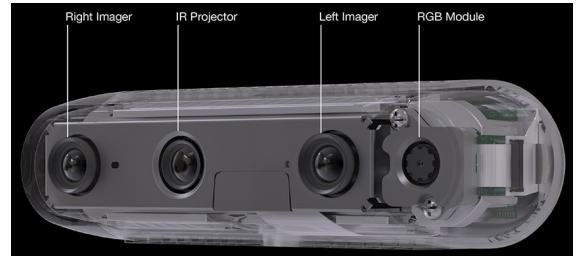
Innenfor kamerateknologien er det to egenskaper som er relevante for denne applikasjonen, henholdsvis oppløsning og Field of View (FoV), på norsk er sistnevnte det samme som synsfelt. Oppløsning bunner i hvor høy grad av detaljer et bildet eller en video inneholder [40]. Ved høyere oppløsning vil bilde ha flere detaljer og dermed være klarere og tydeligere. Oppløsning kan måles på flere måter, den mest brukte er antagelig antall piksler representert innenfor et bestemt område. Field of View er en spesifikasjon som oppgir hvor bredt kameraet kan fange opp tydelig bilde til enhver tid, altså omfanget av det som blir observert.

2.7 Intel RealSense Depth Camera D435

Intel RealSense Depth Camera D435 er et flerfunksjons kamera, som innehar høykvalitets dybdemåler for en rekke bruksområder [41]. Dette kameraet har det bredeste FoV blant alle Intel sine kameraer, i kombinasjon med en "global shutter" på dybde sensoren er dette et ideelt kamera for applikasjoner med høy hastighet.

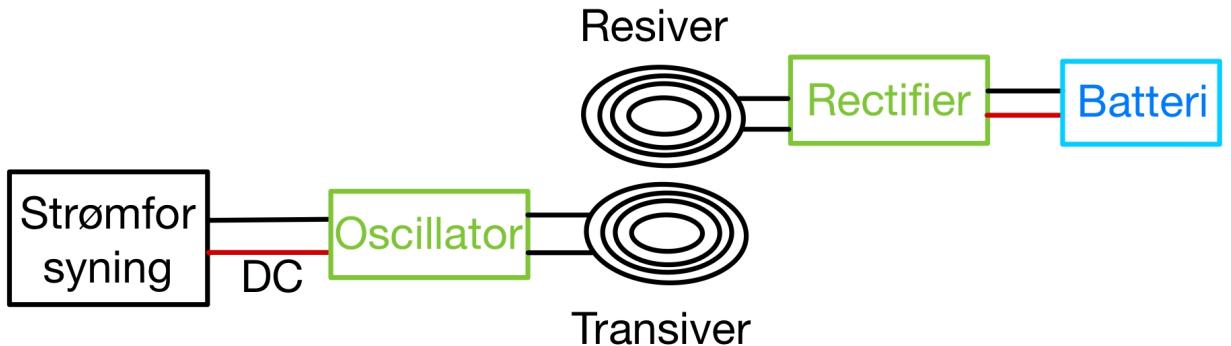
Utfra bildet av kameraet uten fremdeksel, Figur 2.7, kan man se at RGB-modulen er plassert helt til høyre på kameraet, dette er farge-modulen som fanger opp det vanlige bildet. Videre ser man at IR-projectoren er plassert litt til venstre for senteret, dette er modulen som detekterer dybde, og er derfor en sentral del av kameraet.

Field of View for D435 er på $87^\circ \times 58^\circ$ [41]. Samtidig er det oppgitt at "Minimum depth distance (Min-Z) at max resolution" er tilnærmet 280 mm [42].



Figur 2.7: RealSense D435 uten fremdeksel.

2.8 QI-Lading



Figur 2.8: Qi system

Qi er en lade standard for trådløs lading, over korte avstander [43]. Dette systemet benytter seg av ett magnetfelt generert av en varierende strøm i en coil. Over ligger det en mottaker coil som vil ta opp magnetfeltet, og en strøm vil bli induksert. Magnetfeltet må være varierende for at det skal bli induksert strøm i mottaker coilen [44]. For at magnetfeltet skal være varierende kjøres DC strømmen inn i en oscillator og videre inn i coilen som gjør at magnetfeltet endrer seg etter frekvensen på oscillatoren. Denne oscillatoren ligger i en PCB i transiveren, for å få strømmen tilbake til DC strøm er det også en likeretter i PCB-en på receiveren [43].

Avstanden mellom de to coilene er avgjørende for effekten av strøm overføringen. Jo større avstanden er, desto lavere blir magnetfeltet som fanges opp i receiver coilen, og den indukserte strømmen

vil bli mindre. En skisse av systemet kan ses i figur 2.8.

2.9 Aerodynamikk

Objekter i bevegelse vil konstant måtte forholde seg til den luften den passerer gjennom. Gasser er fritt flytende partikler og vil derfor til en viss grad forme seg etter den fasongen et objekt har. Kjennskapen til hvordan luften krummer seg om et objekt, basert på fasong og hastighet, er det som primært omfattes av aerodynamikken [45]. Bernoulli likningene sammen med Newtons bevegelseslikninger danner grunnlaget for store deler av feltet [46]. Bernoulli´s likning for friksjonsfri strømning er vist i likning 2.5 og viser hvordan statisk trykk, dynamisk trykk og hydrostatisk trykk henger nøyne sammen [37]. Allerede her kan man se noe av kompleksiteten ved aerodynamikk, med mange parametere man må kjenne til for spesifikke situasjoner.

$$p + \frac{1}{2} \rho v^2 + \rho g h = konstant \quad (2.5)$$

Aerodynamikk er et svært komplekst fagfelt da særlig fysikken rundt turbulent flyt er svært vanskelig og kompleks å beskrive. Veske eller gass kan bevege seg enten laminert eller turbulent basert på forutsettingene, ved høy hastighet får man turbulent flyt og konsekvensene kan være uforutsigbare. Det er derfor en blanding av både eksperimentell og teoretisk forsking som benyttes på dette området [45]. Bruk av vindtunneler for empirisk testing er en av metodene som benyttes for testing av objekter ved flere forskjellige metoder for både over- og underlydshastighet [47].

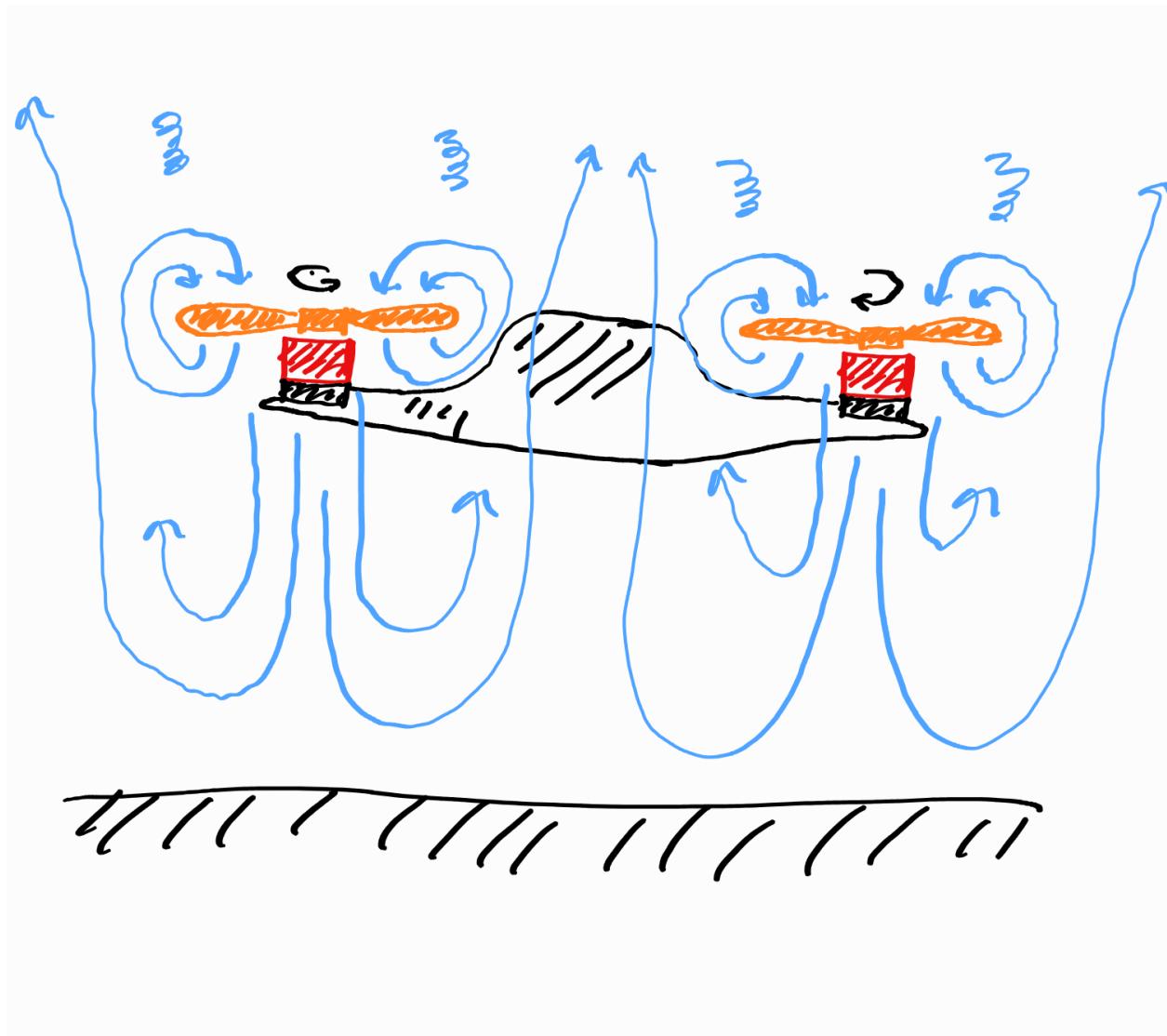
Basert på de store datamengdene som blir samlet inn i vindtunnel forsøkene er det etterhvert laget relativt kraftig data simulerings verktøy for simulasjon av luftstrømmer rundt et objekt [45]. Denne teknikken kalles CFD (Computational Fluid Dynamics) og er blitt et viktig verktøy når utvikling av overflater eller fasonger i luft er essensielt.

2.9.1 Effekten av kontinuitetslikningen

$$\rho_1 A_1 v_1 = \rho_2 A_2 v_2 \quad (2.6)$$

I likning 2.6 er kontinuitetslikningen vist [37]. Denne er en av de sentrale likningene innenfor fluiddynamikk og forklarer sammenhengen mellom hastighet, areal og trykk. Med tanke på aerodynamikk og løft er denne spesielt interessant hvis det er en forskjell i arealet som luften trekkes fra. Skulle det aktuelle området med luft en propell kan benytte seg av minke vil dette gi et mindre trykk og tap av løft. Denne effekten kan oppleves som et ”sug” inn mot vegger eller liknede som kan begrense luftareal og strømning.

2.9.2 Bakkeeffekt

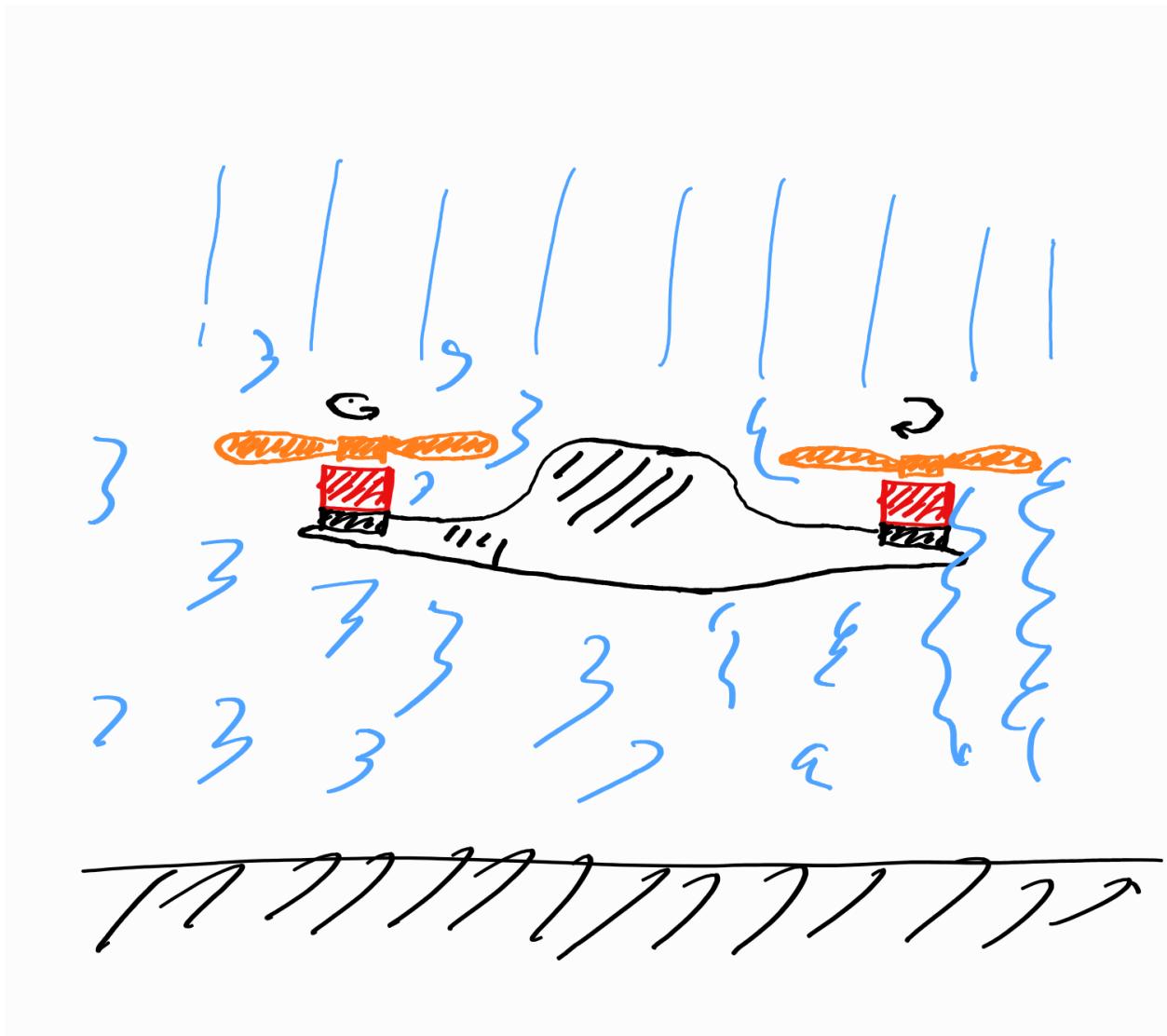


Figur 2.9: Illustrasjon av en drone som hoverer med bakkeeffekt

Bakkeeffekten er et velkjent aerodynamisk konsept hvor fly eller helikoptre har større løftekraft når farkosten er relativt nærmere bakken. Denne effekten er lett observerbar som flyger da motorhastigheten som kreves for en viss trust reduseres betydelig [48]. Denne effekten er skapt av at den turbulente luften som blir presset nedover av propellene treffer bakken, blir reflektert og denne turbulente luften er så den luften propellene skjærer gjennom, se figur 2.9. Dette betyr mindre motstand for propellene og høyere hastighet ved samme effekt [49].

Ved eksperimentell testing rundt flyvning og propeller testing må dataene som sammensettes være korrigert i form av matematisk korrektsjon av dataene eller test enheten flyttet høyt nok til at effekten ikke har en betydelig innvirkning [49]. Ved analyse av fly vil effekten være maksimal ved en høyde rundt et halvt vingespenn over bakken [48], effekten ved helikoptre er mer sammensatt men kan hovedsakelig utelukkes hvis rotoren er mer enn en rotordiameter over bakken [49].

2.9.3 Vortex ring state



Figur 2.10: Illustrasjon av en drone som opplever vortex ring state

Vortex ring state er et komplisert aerodynamisk effekt som kan oppstå når en propell har en for høy luft gjennomgang i motsatt retning av den retningen hvor løftekraften settes opp [50] [49]. Hvis ikke effekten tas hensyn til kan den imidlertid skape farlige situasjoner på grunn av betydelig mangel av løft og vibrasjoner. I drone miljøer blir denne effekten ofte feilaktig referert til som "prop wash", og er en kjent problemstilling.

Effekten har sin opprinnelse i luft som blir fanget rundt rotorene. I figur 2.10 er dette vist for hvordan det kan oppstå ved lave høyder. Luften som strømmer opp rundt propellene vil fange den luften som blir presset ned av propellene og dratt ned og rundt. Dette er igjen den luften som suges ned i propellene igjen [50]. Et turbulent luft område rundt propellene vil derfor dannes og resultatet er propellen tråkler den samme luften uten særlig løft [49]. Denne ringen av gjenbrukt luft er det som refereres til som Vortex ring state. Området vil bevege seg fra tuppen av propellene og nærmere og nærmere midten av rotoren ved økende hastighet av luften rundt og senke det løft givende området.

Ved Vortex ring state vil betydelige vibrasjoner oppstå og graden av kontroll vil bli betydelig redusert, ofte sammen med ukontrollert tap av høyde. For å bryte effekten vil man ofte ha en viss horizontal fart for å bryte opp luften under propellene [50]. Det vil sjeldent fungere å øke kraften da dette kan forsterke effekten og føre til økt tap av kontroll. Effekten krever god og hurtig respons fra piloten eller programvaren for å kontre på en god måte. Turbulent Wake stateer en mer moderat

form for denne effekten [49].

2.10 Kakute F4V2 med betaflight

Kakute F4V2 er en flight controller til droner. Denne styrer hele dronen og er bindepunktet til alle komponentene på dronen. I tillegg har den en innebygd "Power Distribution Board" (PDB) som fordeler strøm til de fire escene (motorkontrollerne) som styrer motorene. Kakute kortet har også en IMU og ett barometer for å kunne styre dronen [51]. IMUen er av typen IMC20689 som har akselerometer og gyroskop. Gyroskopet har ett måleområde på $\pm 2000 \text{ dps}$, mens akselerometret har ett måleområde på $\pm 16 \text{ g}$ [52].

Betaflight er et open source program hvor det er mulig å sette opp flight controllere [53]. Programmet er hovedsakelig satt opp for hobby flygere, og har ett grunnleggende oppsett for droner og RC fly. Det er derfor raskt og enkelt å komme i gang for testing av en platform. Betaflight har støtte for flere fc kort, deriblant kakute F4v2. Med programmet er det blant annet mulig å sette opp fjernkontrolleren sammen med receiveren, sette opp forskjellige fly moduser, kalibrere escene, teste motorer og sensorer. Betaflight har også en blackbox funksjon som gjør det mulig å samle all data fra sensorer under flyvning om fcen har minne til dette.

3 Konseptutvikling

3.1 Glassbur konseptet

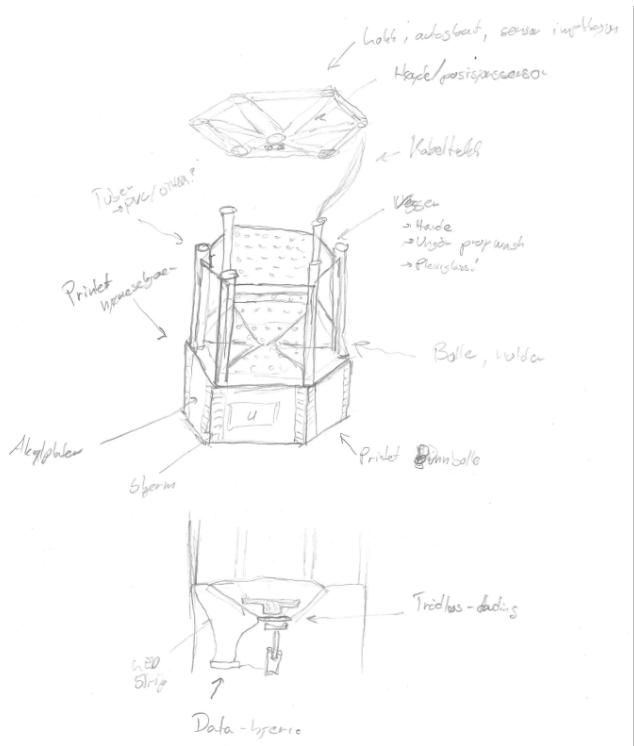
Glassbur-konseptet er en løsning som byr på en til dels svært mekanisk løsning på utfordringen ved transport av dronen tilbake til utsgangsposisjon, samt en fysisk barriere ved hjelp av plexi-glass veggger. I figur 3.1 kan en skissetegning av konseptet ses. Selve basen består av en heksagonal plate hvor hjørneseksjonene skrus fast. Den heksagonale formen gir et relativt stort areal inne i buret, samtidig som det gjør at arenaene kan settes ved siden av hverandre på en arealeffektiv måte, hvis kvantitet er ønskelig for mer effektiv AI-trening. Mellom disse hjørneseksjonene settes svarte Polyethylen plater ned i hakk, slik at basen ikke er gjennomsiktig. Dette gjør produksjonen av sideveggene enklere samtidig som basen også i stor grad er modifiserbar og demonterbar. På en av sidene finner man en skjerm bygget inn i sideplaten hvor viktig og interessant informasjon vises til observatører. I toppen av sidehjørnene vil det ligge en list hvor trekantede plater hviler mellom hjørnene på langsiden og ned mot midten slik at en heksagonal-pyramidetrakt formas. Dette for at når dronen faller ned, vil den treffe platene, og hellingen vil føre til at dronen glir ned mot midten. I midten er det et hull hvor en aktuator styrer en pinne. Denne pinnen vil kunne bli brukt til å flippe dronen om den ligger opp ned i buret og også rotere dronen slik at den vil ha samme utgangspunkt både i orientering og rotasjon før hver generasjon. Plexi-glass benyttes som veggger og settes ned i spor i sidehjørnene, den vil så bli festet på samme måte i toppen slik at det danner en stabil struktur. I toppen lages det et lokk bestående av synkende pilarer fra hver av hjørnestykke og inn i en midtseksjon. I denne midtseksjonen kan en sensor integreres, som har mulighet til å plotte dronens x, y og z posisjon samt dronens rotasjon. Når den plasseres i toppen, vil den få maks måling på dronen og ha oversikt over hele buret.

Siden veggene er solide vil disse, i tillegg til platene dronen sklir på, være perforert i nok grad til at det ikke blir skapt downwash fra propellene. En slik turbulens vil være svært ødeleggende for å kunne simulere et faktisk miljø og spolere best mulig trening.

For å kunne ha en aktuator i midten og samtidig ha skrå plater ned mot midten, vil hjørnene måtte ha en viss høyde. Dette vil skape et relativt stort tomrom i basen under buret. Dette gir mulighet til å montere og «gjemme» bort elektronikk, styringssystemer og datamaskiner her.

For å øke den visuelle utsynet til buret og gjøre det attraktivt å se på, kan også LED-belysning integreres i basen slik at det lyser opp buret. Dette LED-lyset kan også gjøres slik at det skifter og blinker med dronens bevegelser.

Burets modulære stil gjør også at det enkelt kan skrus fra hverandre og transporteres til ønsket lokasjon.

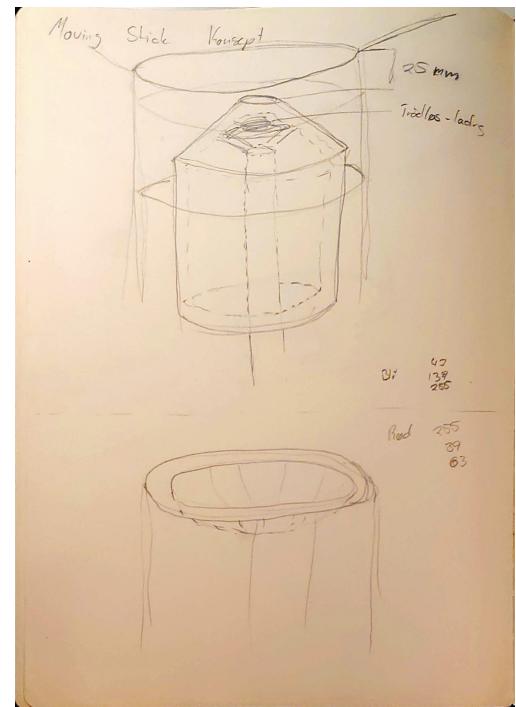


Figur 3.1: Skissetegning av glassbur konseptet

3.2 Midt-aktuator konseptet

I midten av bur-konseptet trengs det en måte å rottere og flippe dronen på. Midt-aktuator konseptet løser dette ved å ha en søyle i midten, nederst i trakten som kan heves samt roteres. I dette konseptet blir hevingen gjort av en lineær-aktuator som er plassert i midten av søylen og festet i bunnen inne i basen på buret. Denne søylen kan være utstyrt med tannhjul inne i søylen samt en motor med et tannhjul som sitter festet i et tårn fra festet til lineær-aktuatoren, slik at når søylen heves eller senkes vil dette tannhjulet gli opp og ned på de indre sporene. Siden de fleste lineær-aktuatorer ikke har en låst rotasjonsaks om lengdeaksen vil denne motoren kunne rotere sylinderen om toppfestet til aktuatoren. Midtsøylen kan dermed designes på en slik måte at dronen vil sentreres hvis den ligger riktig flippet, eventuelt flipper over hvis den ligger opp ned. Midtseksjonen vil dermed kunne benyttes til å flippe dronen samt rotere dronen slik at utgangsposisjonen er den samme uansett hva slags inngangsvinkel eller posisjon dronen havner opp med. Midtsøylen vil trenge støtte fra kantene for å ha en stabil vandrings.

En annen funksjon vil være at midtseksjonen også kan heves, når dronen har rett orientering og rotasjon, til et stykke over trakten. Dette vil være med på å minimere problemet med «downwash» da avstanden mellom propellene og platene vil bli større, og skaper mer plass for luften og dermed minimerer faren for turbulens.

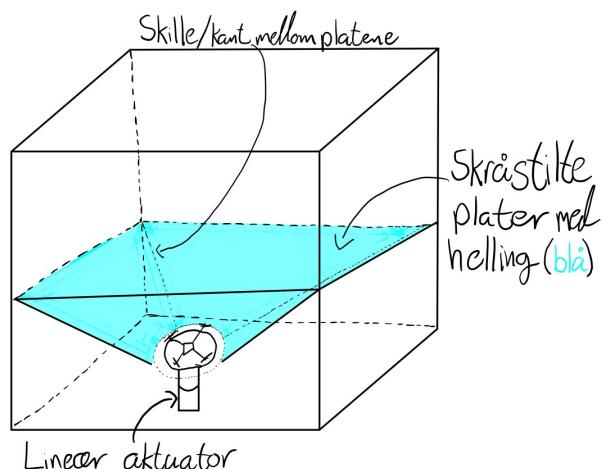


Figur 3.2: Skisse av Midt-aktuator konseptet.

3.3 Kube-arena konseptet

Kube-arena konseptet tar utgangspunkt i en kube som utforming, hvor bunnen vil være en konet form ved bruk av skråstilte plater, samt glatt overflate som visualisert med blå farge i Figur 3.3. Ved bruk av denne type underlag og utforming vil dronen kunne skli ned til senter, som er det laveste punktet, hvor den så vil bli snudd og rotert til samme utgangspunkt for orientering etter hver iterasjon. Et viktig aspekt ved denne type design er det faktumet at mesteparten av beskyttelsen av dronen vil ligge i designet av selve dronen. Dermed blir arenaen en platform for å gi dronen samme standardiserte utgangspunkt og sikre den fra å fly avgårde, men den vil være såpass solid at konseptet er avhengig av en svært godt beskyttet drone.

Dronechassiet er tiltenkt en god beskyttelse i form av propell-guards og en overgående bue på toppen av dronekroppen, en essensiell detalj her er flat bunn. Propell-guards er et omfattende hjelphemiddel for å beskytte droner ved flyvning, spesielt for uerfarne droneoperatører på hobbybasis. Dette er en del som kan 3D-printes etter ønsket design, dermed kan det ved hjelp av 3D-CAD bli tegnet for å passe dronen slik at designet blir robust og optimalisert for den spesifikke dronen.



Figur 3.3: Skisse av Kube-arena konseptet.

Som følge av disse beskyttelsene vil dronen kunne krasje hvor som helst i arenaen, uten å ta skade. På grunn av utformingen av beskyttelsen vil dronen flippes rundt om den lander på hodet, siden undersiden av dronen er flat vil den stabilt kunne skli ned til senter av arenaen uavhengig hvor den lander. I senter av bunnen må det være en form for løftmekanisme, tiltenkt en lineær-aktuator eller lignende. Denne plattformen skal også kunne roteres for å skape riktig orientering før hver flyvning. En annen positiv ting dette reduserer er groundeffekten, dette er på grunn av at dronen vil løftes noe opp fra bunnen og dermed skape en trygg måte for å ta av på.

Kamera skal implementeres i lokket, og dette vil kunne være avtakbart. Veggene i dette konseptet kan være laget av akrylplater som festes sammen med utgangspunkt gjennom fôringer ned i bunnseksjonen. På denne måten kan lokket ha samme kvadratiske utforming som resten av arenaen, ved hjelp av slisser kan lokket enkelt tres på toppen, dette vil hjelpe med å stive av hele arenaen samt skape en fysisk blokade slik at dronen ikke stikker av gårde eller flyr direkte på noen av observatørene.

Kuben vil stå på en type hub i bunn, dette vil bli grunnmurent til arenaen. Innvendig av bunnen vil alle komponentene bli plassert, dette innebærer alt fra PC til supply og andre fysiske komponenter dette systemet vil kreve. Som følge av høye sidevegger i grunnmuren er plassen betydelig nok til å kunne romme det som vil være nødvendig for at arenaen skal stå selvstendig å jobbe over et lengre tidsrom.

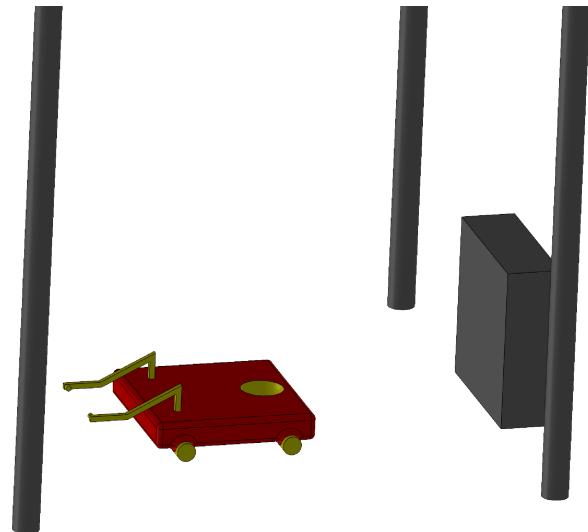
3.4 Plattform på hjul

En av de vurderte løsningene på problemstillingen er en plattform på hjul som kan ses i figur 3.4. Denne skal kunne settes i et tomt rom sammen med fire søyler som vil avgrense ett virtuelt rom som dronen kan fly i. I dette rommet vil dronen kunne bruke RL til å lære seg å fly, ved hjelp av plattformen som vil resette dronen hver gang den krasjer. Plattformen skal ved hjelp av fire mecanum hjul bevege seg til alle lokasjoner i rommet hvor dronen kan lande. Ved bruk av disse hjulene vil det ikke være nødvendig med en svingmekanisme i plattformen, noe som vil gjøre den mer pålitelig.

Dronen vil være designet slik at den kun vil ligge på to mulige posisjoner etter en krasj. Dermed vil det bli enklere for plattformen å løfte dronen tilbake til startposisjonen. Dette utføres ved bruk av to stive armer som roterer om innfestningspunktet til plattformen, disse er gule i figur 3.4. For å unngå en gripe funksjon vil det bli brukt to magnetar som vil ta tak i dronen i bestemte punkter. Dette vil gjøre at det ikke vil kreves like høy presisjon når dronen skal plukkes opp, i tillegg vil tyngdepunktet flippe dronen om den har lagt seg på hodet. Orienteringen av dronen vil bli nullstilt ved at selve arenaen roterer.

Søylene som brukes til å avgrense det virtuelle rommet gjør at denne løsningen vil bli veldig lett å flytte rundt på, grunnet at det kun er et virtuelt rom. Så fort dronen forlater det virtuelle rommet vil søylene sende informasjon til datamaskinen, dermed vil datamaskinen terminere drone umiddelbart når den forlater det virtuelle rommet. Ved hjelp av teleskop funksjon på søylene vil det være enkelt å feste dem i ett rom. Da kan man stramme dem opp mot taket uten å bruke noen annen innfestning. Det vil også være kroker på søylene får å kunne feste et nett rundt, for ekstra sikkerhet om det skal være personer i rommet.

Både plattformen og sensorene på søylene skal kommunisere sammen med en datamaskin, denne kan ses som den grå boksen i fig 3.4. På denne maskinen vil AI'en kjøre og all bildebehandlingen vil foregå. For hver generasjon vil datamaskinen sende nye verdier til dronen, det vil ikke skje underveis



Figur 3.4: Plattform med PC og virtuelt rom

i generasjonen. Ved siden av PCen vil det være en trådløs lader til dronen. Det er da tenkt at det skal bli brukt to droner, den ene kjører mens den andre lader. Siden alt styres fra PCen vil dronene utvikle seg samkjørt, altså så fort drone en skal lade så tar den ny ladede dronen over verdiene fra dronen som skal lades. Dette vil gjøre at treningen kan foregå kontinuerlig uten lade-stopp.

3.5 Konseptvalg

Gjennom prosessen med konseptutvikling ble flere varianter og forslag lagt på bordet. For å best mulig kunne ta en avgjørelse for designutgangspunkt ble det opprettet kriterier hvor hvert enkelt konsept blir poeng-lagt med score fra 1 til 10 utifra hvor godt de egner seg i de forskjellige kategoriene. En fordel med dette er å kunne rangere de forskjellige konseptene opp mot hverandre og dermed kunne ta avgjørelsen om hvilket konsept som er ønskelig, eventuelt om flere aspekt kan kombineres fra de forskjellige forslagene. For å kunne vektlegge forskjellene og filtrere ut det som er gode løsninger ble kategorier i forskjellige disipliner laget, dette er kriterier som var forutbestemt. Derav ble muligheten for å ha kriteriene i bakhodet under designet av konseptene. Gjennom poeng-utdelingen og videre presentering av konseptene vil Konsept 1 være glassbur konseptet, Konsept 2 vil være Kube-arena konseptet og Konsept 3 vil være Plattform på hjul.

Tabell 3.1: Tabell over score for de ulike konseptene.

	Konsept 1	Konsept 2	Konsept 3
Utseende	9	7	7
Modularitet	9	7	8
Kompleksitet	9	7	9
Produksjon	8	8	9
Porterbarhet	8	6	10
Robusthet	9	8	6
Nødvendige komponenter	10	9	10
Flyområde	8	6	10
Kontinuerlighet	10	8	9
Kostbarhet	8	9	6
Presenterbar	10	6	6
Vekt	7	7	9
Fremvisning	10	8	6
Kamera/sensor implementering	10	9	10
Versatilitet	10	8	8
Omsorg for drona	9	8	8
Total-score	144	121	131

De forskjellige kategoriene tar utgangspunkt i egenskaper den endelige arenaen ideelt burde inneha, derfor er de fleste bruksområde og avgjørende funksjonene dekket av disse kategoriene. En oversikt over kategoriene og poengsummene gitt er presentert i Tabell 3.1 og i Figur 3.5.

Kategorien Utseende tar for seg det estetiske utseende av arenaen, herunder hvor stort potensial konseptet har for å bli et blikkfang, samt hvor godt det egner seg for utstilling foran et eventuelt publikum. Potensialet er størst ved Konsept 1, med bakgrunn i den stilige fasongen og muligheten til å variere med farger på de ulike komponentene.

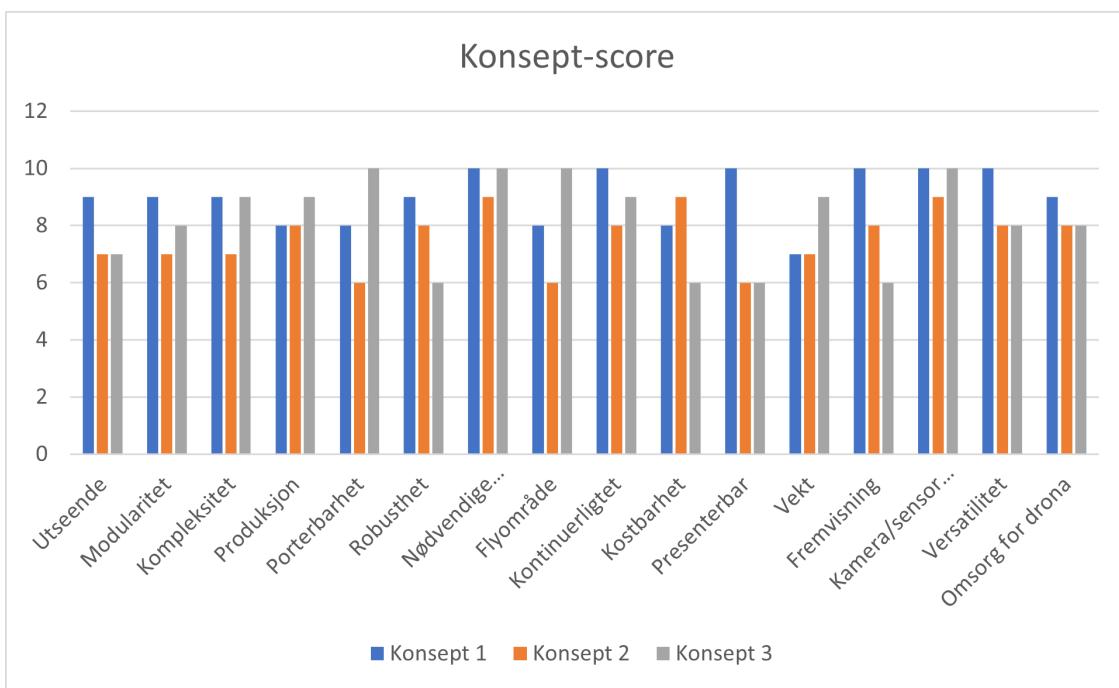
Neste kategori er Modularitet, denne kategorien presenterer hvor godt konseptet kan utvides og bygges på, altså evnen til å kunne skalere opp størrelsen på selve arenaen og dermed få et større flyområde enn utgangspunktet for Konseptet. Konsept 3 vil initialet ha størst flyområde, men konsept 1 muliggjør det å sette flere arenaer ved siden av hverandre og dermed skape et større miljø hvor dronen kan fly. Dette gjør at Konsept 1 får høyest poengsum i denne kategorien.

Kompleksitet er neste, denne kategorien tar for seg hvor sammensatt og komplekst konseptene er. Her scorer konsept 1 og 3 likt med en høy poengsum, mens konsept 2 har ikke like godt potensiale under denne kategorien.

Videre er kategorien produksjon, konsept 1 og konsept 2 baserer seg på mye av de samme komponentene og får dermed lik poengsum. Produksjonen av disse vil være på et moderat nivå, men baserer seg stort sett på 3D-printing og deler/komponenter som er tilgjengelige for en alminnelig person, uten nødvendigvis å være tilknyttet et fakultet eller en bedrift. Produksjonen rundt Konsept 3 vil være lettere, da det ikke er nødvendig med like mange komponenter, samt ingen behov for et komplekst rammeverk. Derimot vil utfordringen være robot-bilen, men disse delene kan kjøpes hos en elektroforhandler eller bestilles.

Dette fører til neste kategori som er porterbarhet, denne kategorien innebærer konseptets evne til å kunne demonteres og flyttes på enkel og rask måte. Her er det konsept 3 som byr på den letteste løsningen og dermed får høyest poengsum, konsept 1 er nest enklest og konsept 2 kommer sist. Utfordringen til konsept 1 er den massive bunnplaten, samt en del enkeltstående komponenter, derimot skal det være ganske gjennomførbart.

Dette overlapper i neste kategori som handler om hvor robust konseptet er, her vektlegges evnen til å kunne stå for seg selv over et lengre tidsrom i kombinasjon med motstandsdyktighet mot eksterne påvirkninger. Her scorer både konsept 1 og konsept 2 til en høy poengsum, da dette er tilnærmet lukkede systemer som vil ha minimal påvirkning fra ytre faktorer. Konsept 3 stiller litt svakere under denne kategorien, som gjør den den får en temmelig lav poengsum.



Figur 3.5: Stolpediagram av poengene gitt til de forskjellige konseptene.

En svært viktig kategori er hvorvidt konseptet har inkludert alle de nødvendige komponentene, eller i det minste lagt opp til at alt kan implementeres på naturlig vis. Her får konseptene ganske lik uttelling, men konsept 2 har ikke en fullstendig løsning på senteraktuatoren - derfor får den ikke full uttelling. Videre er neste kategori flyområde, her er det uten tvil konsept 3 som er har både potensialet og det initiert største flyområdet, uten noen form for påbygging, men kun ved hjelp av enkel justering på utstyret. På grunn av den geometriske utformingen til konsept 1 får den betydelig bedre enn det kubeformede konsept 2.

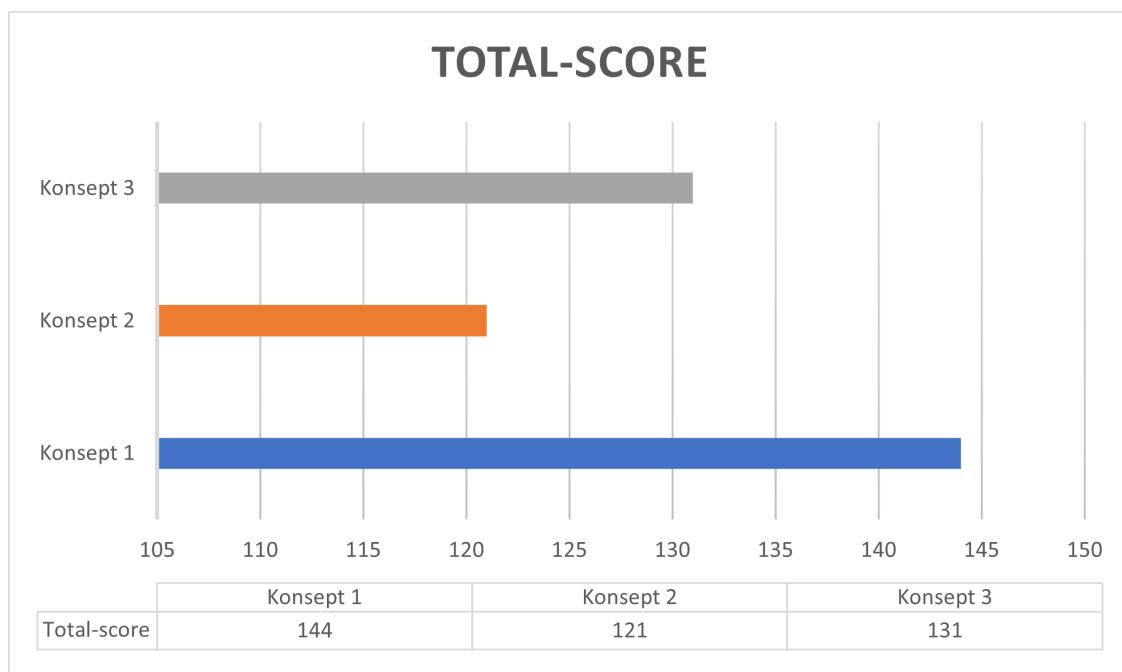
Så kommer kategorien Kontinuerlighet, denne kategorien tar for seg evnen konseptene har til å stå å trenne over ett lengre tidsrom, helt alene. Her scorer konsept 1 best, da denne har en god kombinasjon med pålitelig løsning for orienteringen av dronen og robusthet mot eventuelt andre påvirkende faktorer. Neste kategori tar for seg kostbarheten til konseptene. Her er det enkleste konseptet det som får høyest poengsum, konsept 3 får lavest da dette vil være betydelig mer kostbart

å gjennomføre enn de to andre.

Presenterbarhet tar for seg hvor godt konseptene egner seg til og stilles ut og eventuelt kunne benyttes som en form for fremvisning eller demonstrasjon av utviklingen til dronen, altså i praksis se hvordan prosessen bak reinforcement learning fungerer. Det konseptet som scorer best her er nummer 1, det er på bakgrunn av utseende og størrelsen, samt evnen til å visualisere. Vektaspektet til konseptene er noenlunde likt mellom 1 og 2, da disse baserer seg på temmelig likt utgangspunkt og overordnet fasong. Derimot har konsept 3 den beste løsningen på dette, da den er svært mye enklere i utforming og blir satt opp på nytt når den skal flyttes. Nummer 1 og 2 har en solid utforming, som gjør at de scorer dårligere.

Fremvisning går på hvor lett det er å vise konseptet frem, enten kun for demonstrasjon eller for faglige hensikter. Konsept 3 vil trenge betydelig mer plass enn de andre konseptene, derfor scorer den dårligere. Konsept 1 vil være mest egnet da alt er ferdig implementert, og den har et visuelt stilistisk design. Neste kategori tar for seg hvor godt kamera og sensorer er implementert i konseptet. Her har både 1 og 3 alt ferdig implementert, derimot har 2 en plan på det, men den inneholder ikke en helt ferdig løsning.

Versatilitet går ut på hvordan konseptet stiller seg i forhold til å kunne sette opp flere simultant og hvor tilbøyelig konseptet er. Her er det konsept 1 som er mest gjennomtenkt og har de beste løsningene på dette området. Siste kategori går ut på hvor beskyttende konseptet er ovenfor drona. Herunder vektlegges hvor hardt krasjene vil påvirke drona, samtidig som potensialt konseptet har for og enklest mulig rette eller fikse på noe om det blir behov for det. Her scorer konsept 1 best som følge av sin løsning med et litt rundere flyområde, samt høyden og vinkelen som dronen vil lande på.

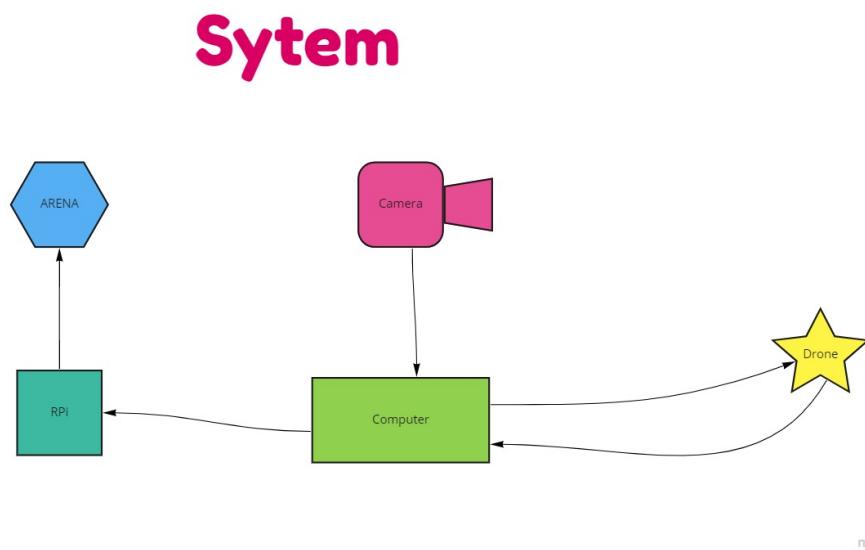


Figur 3.6: Total poengsum gitt til konseptene.

Som presentert i Figur 3.6 er det konsept 1 som får samlet den beste poengsummen på totalt 144 poeng, dette er et forslag som har svært godt potensial og inneholder gode løsninger innenfor de fleste kategoriene. Samlet sett vil dette være utgangspunktet for videre design av Dronearenaen, men det vil også bli inkludert alternativer fra de andre konseptene. Total poengsum på konsept 2 og 3 er også presentert, det endte til slutt med konsept 2 på lavest poengsum og konsept 3 i midten. Det er slik at tankene rundt propellguards vil være essensielt uavhengig av konsept, og er dermed en viktig del og forutsetning for videre design og implementering.

4 Design

4.1 Systemdesign

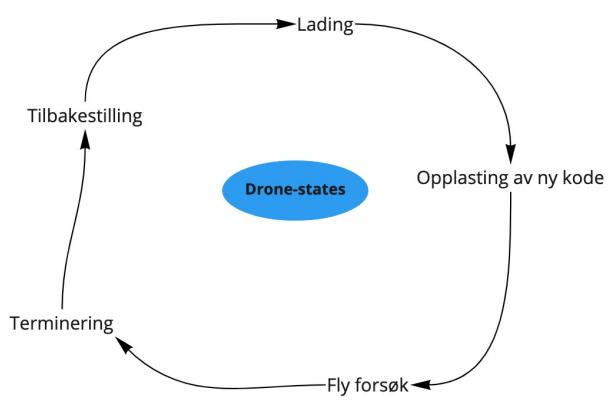


Figur 4.1: Systemdesign

Drone arenaen består av en rekke mindre systemer som hver for seg og i samarbeid dekker kravene for arenaen. Disse subsystemene må fungere sammen på en slik måte at de nødvendige systemene snakker sammen på en raskest og best mulig måte. Design valg av enkeldeler kan ha stor påvirkning på hvordan andre subsystemer må tilpasses. Et godt system design som kan legge til rette for gode plattform valg basert på krav fra alle de påvirkede subsystemene var derfor essensielt i planleggingen. Et godt system design bidrar også til at design valg av små deler og komponent valg er lettere å vurdere og tilpasse da de fleste påvirkende faktorer allerede er skrevet opp.

I figur 4.1 er en overordnet representasjon av hvordan det totale systemet er tenkt delt opp og hvilke enheter som må kommunisere med hverandre. I sentrum av systemet finner man "Computeren". Dette er hovedmaskinen og er den som hovedsakelig kjører bildegenkjenningen fra kamerat i toppen, for plassering av dronen i x, y og z. Denne datamaskinen vil være hovedsakelig ansvarlig for reinforcement learning i en selvstendig hub og har derfor direkte kommunikasjon med dronen. Denne kommunikasjonen er toveis da også data om kjøringen kan sendes tilbake til datamaskinen om nødvendig eller ønskelig.

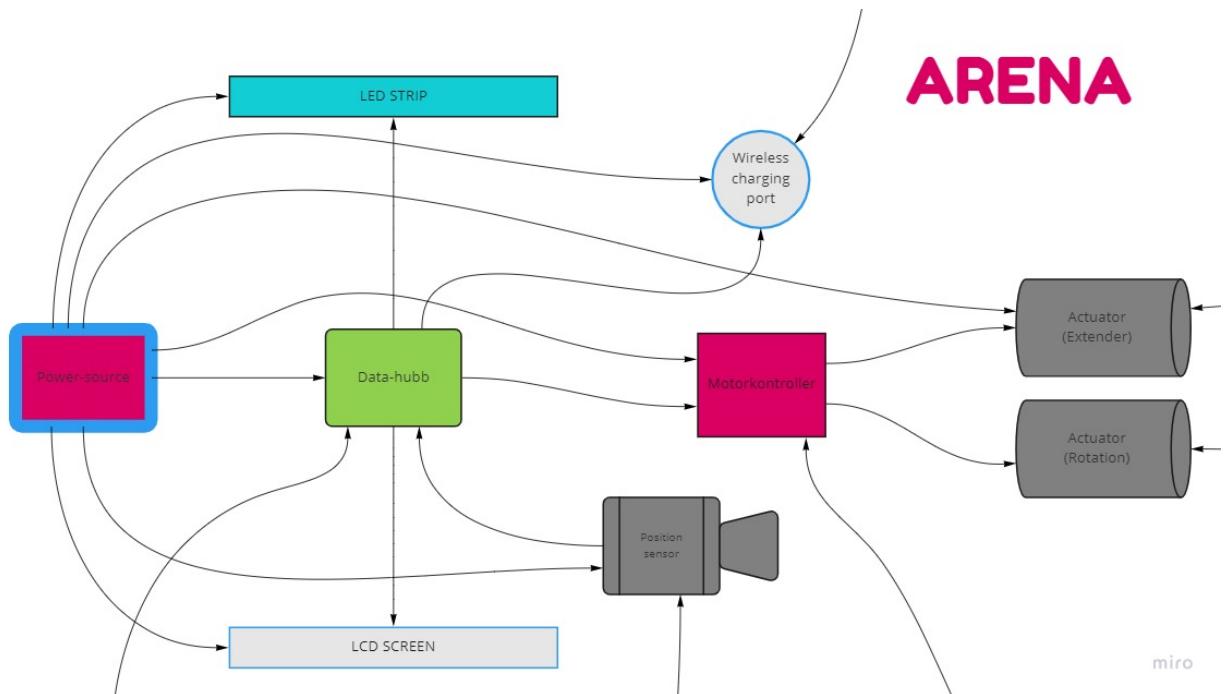
Basert på inputten fra kamera og den informasjonen datamaskinen har angående tilstanden til dronen, befinner seg i løkken for kjøring (altså "lading", "flyging" osv se



Figur 4.2: Tilstander dronen kan være i og hvordan disse følger hverandre

figur 4.2) sendes kommandoer til en Raspberry Pi 4. Denne er markert som "RPi" i figur 4.1 og er ansvarlig for styring av midtaktuatoren, samt start og stopp av QI ladingen. I figuren markert som Arenaen.

4.1.1 Arena



Figur 4.3: Systemdesign av Arena

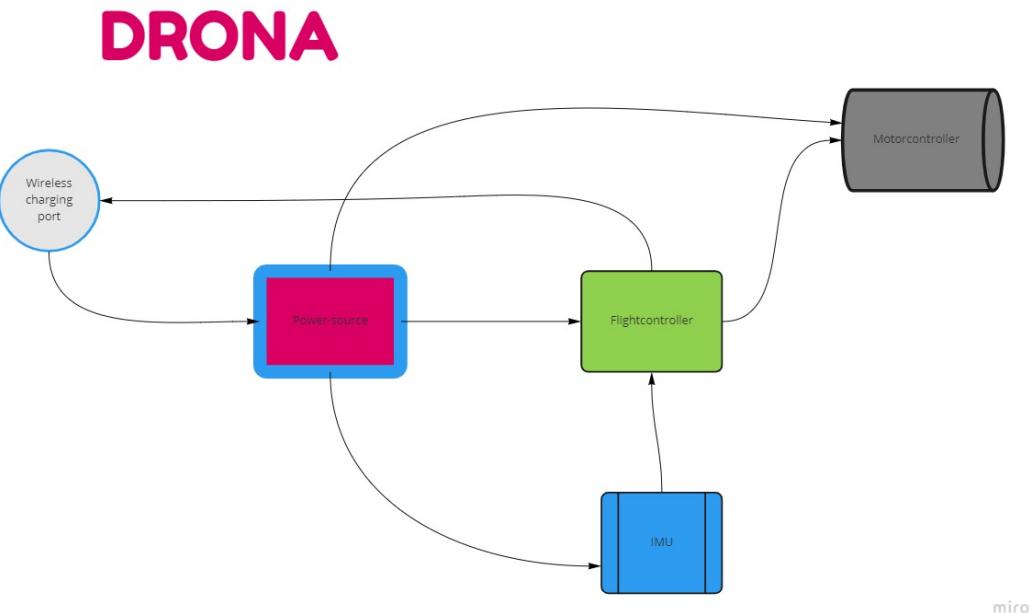
Arenaen er her omtalt som burdelen av systemet og inkluderer da den mekaniske sammenstillingen av arenaen (med for eksempel hjørner, vegger osv.) i tillegg til aktuatorer og prosesseringsenhet plassert i arenaen. I figur 4.3 er et skjema skissert over hvilke enheter som kommuniserer med hverandre. Linjer som går utifra Power sourcerer strømforsyninger. Denne endte opp med å bli delt opp i to, en for RPi systemet samt aktuatorer og en til datamaskinen. Dette har midlertidig liten påvirkning. Litt større betydning er det at den her omtalte "Data-hub" også blir ansett som en sammenstilling av Datamaskinen og Raspberry Pi enheten. Fordelingen mellom disse er beskrevet i 4.1.

Figur 4.3 er primært benyttet til å få oversikt over hvilke krav de forskjellige enhetene har til kommunikasjon med andre enheter, samt oversikt over aktuelle enheter for de forskjellige komponentene. Når det er kjent hvilke krav enhetene har til kommunikasjon er det også betydelig enklere å vurdere komponenter opp mot hverandre.

I de firkantede boksene i figur 4.3 er aktuelle komponenter til de forskjellige rollene listet opp. Dette ga muligheten for alle i gruppen til å enkelt legge inn passende komponenter som ble funnet. Og senere vurdere disse opp mot hverandre. LCD Screensom her er tegnet inn som en blokk ble senere bestemt til å skulle være implementert på datamaskin delen av systemet. Da dette vil gjøre det betydelig lettere å lage et enkelt brukergrensesnitt, samt inkludere AI-læringen i det visuelle brukergrensesnittet på en god måte. Dette er en av de mest interessante delene som kan vises fram.

Beslutningen om å dele power suppliet opp i to deler kom som en naturlig løsning når det ble bestemt å integrere to rack mount i bunnseksjonen. Denne må deles opp i to deler på grunn av plass og to strømforsyninger ble derfor naturlig.

4.1.2 Drone



Figur 4.4: Systemdesign av Drona

Selve drone enheten i systemet blir implementert som et eget subsystem som i stor grad styrer seg selv. Denne er i stor grad utviklet i tidligere oppgaver ved Universitetet i Agder [10] [8] [7], samt betydelige oppgraderinger i Flight Controller prosesseringsenhet i en parallel oppgave [54]. Det er derfor lagt vekt på å kun implementere de nødvendige enhetene som enkeltkomponenter og dokumentasjon. På denne måten kan det legges opp til et fremtidig arbeid hvor disse komponentene integreres på best mulig måte.

Systemdesignet for drone enheten er derfor relativt minimal, begrenset til kun de nødvendige enhetene som må tas i betraktning med tanke på drone-arenaen som system. For å kunne integreres i systemet er strukturen på flight controlleren viktig og det har derfor vært et tett samarbeid med gruppen som utvikler nettopp denne delen. IMU data er en sentral del av målingene som kan benyttes til overvåking av dronen og kan være en sentral del av styringen av dronen. Denne vil bli implementert i flight controlleren og er derfor ikke en egen enhet som vist i figur 4.4, men en sentral del av systemet.

Ladningen av dronen er en sentral del av systemet da dette inngår i kravene til systemet 1.3. For å kunne drive dronen over en lengre tid uten vedlikehold vil en kontinuerlig oppladning av batteriet være nødvendig. Dette gjøres ved hjelp av QI-lading som har vært forsøkt utviklet på disse dronene av tidligere grupper og veileder. Denne ladingen må kunne fungere i et større system og integreres slik at lik lading og balansering foregår over cellene. Optimalt bør derfor batteriets tilstand kunne bli rapportert tilbake til flight controlleren og til hoved huben (her datamaskinen).

Kommunikasjonsplattformen samt ladningen er i stor grad basert på tidligere arbeid med utskytermekanisme for at løsningene enkelt også kan benyttes i et videre arbeid på dette området [7]. Viktigheten i at systemene spiller godt sammen kommer derfor tydelig fram.

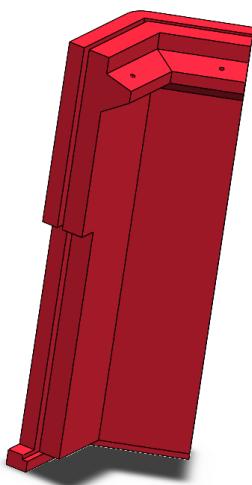
4.2 Mekanisk design

4.2.1 Hjørnestøttene

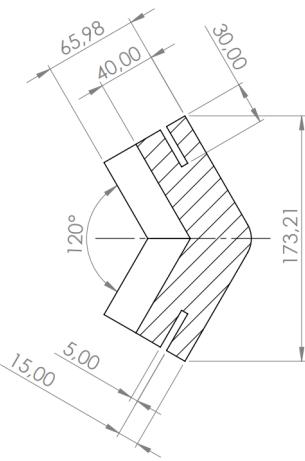
Drone-arenaen består i all hovedsak av en midtseksjon med aktuator, yttervegger med hjørner og selve glassburet med gulv og tak. Selve glassburet og taket er i stor grad veldig lite avhengige av resten av designet med tanke på høyde og utforming, men resten av dimisjonene er tett bundet opp til hverandre og begrensinger på enkeltdeler har stor påvirkning på de andre delene. Små endringer på dimisjonene på disse delene, og da spesielt midtseksjonen og ytterhjørnene vil gjøre store utslag på dimensjonen på den andre, og for hvilken vinkel gulvet i glassburet kan oppnå.

Hjørnene i arenaen er den mest begrensende faktoren for designet. For å kunne gi rom for mest mulig vinkel og høyde nok til aktuatoren i midten vil det å kunne utnytte tilnærmet maks mulig høyde på disse være optimalt. På grunn av valget om å benytte 3D-printing som produksjonsmetode, vil maksimalene på printeren som skal benyttes sette de ytre grensene for designet. De tilgjengelige printerene er av en type som har maks høydemål på 400 mm, men av erfaring vil de aller siste lagene være svært usikre og 390 mm ble derfor tatt som utgangspunkt for maks høyde på hjørnene.

Fra konseptet ble en sekskantet konfigurasjon på buret valgt. Hjørnet er derfor en 60 graders vinkel som er ment til å holde sideveggene på plass i slisser langs siden og glassveggene på plass på toppen. Dette er hovedmålet bak designet. Det vil bli en del press på glassveggene på toppen av delen og en viss tykkelse og styrke kreves derfor. Et annet perspektiv er tiden det vil ta å produsere en slik del. Selv med 3D-printing vil små endringer i tykkelse ha mye å si på både printe tiden og hvor mye filament som kreves. På grunn av styrkeaspekte kreves også en viss mengde infill i printen som igjen vil bety at små endringer i tykkelse, kan føre til mange timer lengre produksjonstid og potensielt sett flere hundre kroner i konstnadder på filament. En balanse i tykkelsen gir både god nok støtte og styrke, men samtidig har en rimelig minimal profil for å redusere produksjonstid og kostnad.



Figur 4.6: Drone Arena hjørne, sett fra siden, teknisk tegning i appendiks E.3



Figur 4.5: Drone Arena hjørne, dimensjoner i mm sett i fullprofil ovenfra, teknisk tegning i figur E.3

Videre i prosessen ble det integrert to rack holdere på bunnplaten som blir benyttet til datamaskiner og annen styringselektronikk. Disse var ønskelig at ble benyttet i en standardstørrelsen som 19 tommer. Som nevnt tidligere er vinklene i buret svært viktige, og en øking i sidelengdene vil bety at enten vinkelen inni buret blir mindre, eller hjørnene måtte blitt høyere. Siden hjørnene allerede er tilnærmet maks høyde, er det eneste som kan gjøres å øke sidene. Dette vil midlertidig også senke en allerede litt på kanten lav vinkel i buret. Det ble derfor heller bestemt å lage små forsenkninger i den ene siden på fire av hjørnene. Kabinettene til rackene ble plassert på to av "baksidene" av arenaen med en side imellom. På denne måten ble det plass til å plassere kabinettene med relativ lang lengde bakover uten å krasje, samtidig som en skjerm kan plasseres på veggen lengst unna disse to sidene og en viss symmetri beholdes.

Små innhakk for å muliggjøre at et 19 tommers kabinett kan monteres halvveis inn i sidene på hjørnet ble derfor laget. To på venstre side av hjørnet og to på høyre. For produksjonstid og kostnad har dette svært lite å si, da disse innkappene vil kreve support under printing og dermed kreve omtrent samme mengde plastikk og tid. Styrken i hjørne blir litt svekket da det ikke er helgående struktur ned, men et rett innkapp som vil skape en stress konsentrasjon. Tatt

vekten i betrakning bør ikke dette være et for stort problem, spesielt siden kappet er i den nedre halvdelen og dermed har gods over til å fordele vekten.

4.2.2 Midtseksjon

I midten av arenaen ligger det en rund seksjon som er skrudd fast i bunn. På toppen er flaten vinklet slik at platene som kommer fra hjørnene ligger flatt nedpå ringen og gir en nedre støtte for platene. For å skape mest mulig fleksibilitet i platene og dermed litt demping er festet i bunn kun tenkt til å kun være en støtte nedover uten å være skrudd fast eller festet på andre måter. Dette for å gjøre det enkelt å bytte platene og også gjøre det muliggjør for platene å potensielt dempe fallet til dronen noe.

I bunnen av midtseksjonen er det forsenkning for plassering av et nålelager. På dette lageret skal midt-aktuatoren som sørger for vipping og rotasjon av drona plasseres. Det er derfor også lagt inn et kryss i bunn inn mot midten som ender i en ring. Fra midtaktuatoren ligger det lagt inne en M8 skru som tres ned gjennom denne ringen og skrus fast med et ytre lager imellom bolten og denne ringen. Dette vil forhindre enhver bevegelse av midtaktuatoren og gjøre hele midtseksjonen mulig å flytte i arenaer uten at noe kan forflytte seg.

Rundt ringen er det laget 8 M8 hull for montering til bunnplaten. Det ene av disse hullene vil bli dekket av rotasjonsmotoren og vil derfor ikke kunne brukes til montering av midtseksjonen. Det vil i midlertid holde helt fint med 7 stykk og antallet er mer for å fordele vekten jevnt utover og hindre enhver mulig rotasjon, samt muliggjøre for å kunne montere et mindre antall skruer.

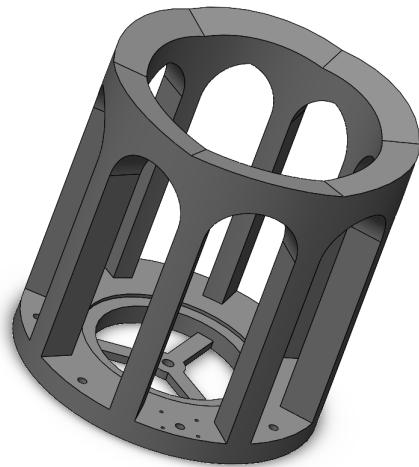
Under midtplateen vil det plasseres en printet ring omtrent med samme dimensjoner som veggene på midtseksjonen. Vekten av dronen, midtseksjonen og platene i arenaen vil i stor grad ligge helt på midten av platen. Tatt i betrakting av at diametern på bunn er 1,1 meter vil dette kunne føre til en forsenkning av platen. Ved å skru ringen på i bunn vil man kunne komme unna med en mindre dimensjon på bunnplaten, som igjen vil føre til mindre vekt og bedre porterbarhet.

Midtseksjonens toppling skal ha en betydelig høyde, for å spare produksjonstid og materiale er det derfor lagt opp til en sylinderstruktur. Disse kommer sammen i toppen før skråingen for glideplatene begynner. For å slippe støttestrukturen under printing av topplingen er det lagt på store avrundinger i hjørnene som møter hverandre. Dette gjør at printeren er kapabel til å lage topplingen uten å måtte bygge lange broer. Dette halverte den estimerte printetiden av delen. Slike broer er vanskelige å få finne, da varmt plastikk faller ned, og med mellomrom i denne størrelsen ville det antagelig ikke fungert i det hele tatt. Den endelige printen hadde minimal nedsenking av plastikk og et svært godt resultat. Denne avrundingen vil også gi en økt styrke til topplingen selv om dette antagelig ikke er nødvendig for de kreftene det er snakk om.

Ringene i bunn har en høyde på 10 mm for å fungere som et ben på midten. Det er hull som blirliggende på linje med hullene i midtseksjonen med forsenkning rundt slik at metallhodet på skruen blir beskyttet. Hele arenaen står derfor kun på de printede hjørnene og ene bunnringen slik at den vil kunne bli plassert også på finere og mer skjøre flater uten å lage riper.

4.2.3 Glideplatene

Glideplatene i arenaen utgjør en av de mest sentrale komponentene for å kunne tilfredsstille kravene for systemet. Det har også vist seg at er en av de mest krevende komponentene når det kommer til materialvalg, utstede og plassering. Som nevnt i kapittel 4.2.1 og 4.7 er høydeforskjellen



Figur 4.7: Midtseksjonen sett ovenfra og fra siden, full teknisk tegning i appendiks 4.2.2

mellan disse to komponentene utslagsgivende for den mulige vinkelen til glideplatene. Tanken bak konseptet er nettopp at vinkelen til disse platene i seg selv skal være nok til at dronen glir ned mot midten. I designfasen var ikke de eksakte materialene som skulle benyttes for droneframen kjent og den nødvendige vinkelen var derfor heller ikke mulig å finne. Basert på begrensningene til komponentene rundt ble vinkelen gjort så stor som mulig for å så kunne finne løsninger for materialer som tillater dronen å gli på denne.

For å finne den eksakte formen på platene ble i stor grad Solidworks og CAD assemblyet benyttet. En tilnærmet ferdig hjørnedel og en midtseksjon med indre ring i riktig høyde ble plassert med korrekt avstand og vinkelen ble målt. Denne vinkelen sammen med den kjente avstanden inn til sentrum vannrett ble benyttet til å finne lengden av hypotenusen i denne vinkelrette trekanten. Bredden på platen inn mot hjørnene er kjent basert på lengden mellom hjørnene. Platen ble derfor konstruert med denne bredden og lengden. Siden det er kjent at platene skullehypotetisk sett møtes i midten ble dette laget som et felles punkt og sidene på trekanten gjort like lange. Hullet til midtaktuatoren kunne så skjæres ut med et kutt i riktig vinkel slik at de endelig hullet blir loddrett rundt.

I toppen av platene ble det lagt til et 3 mm hull, et på hvert side ut mot hjørnene slik at disse treffer ned mot listen på sidehjørnene. Dette slik at hvis platene skulle ha en tendens til å skli eller flytte på seg kan de festes i hvert hjørne med en M3 skrue. Dette hullet ble senere fra assemblyet prosjektert med riktig vinkel ned på hjørnestøttene slik at dette kunne printes og da senere gjenges opp.

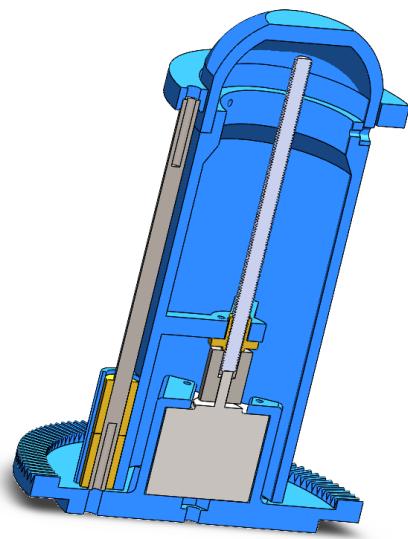
Platene er laget like for å forenkle produksjonen og holde kompleksiteten på designet så enkelt som mulig. På grunn av vinklene disse platene lener seg mot hverandre med, vil tykkelsen på platene bygge ”inn i” hverandre. Vinkelen er igjen den samme, bundet geometrisk, som platene heller nedover. Den korresponderende avstanden dette utgjorde ble derfor kalkulert og kuttet inn fra langsidene. Platene skal derfor teoretisk kunne ligge tett i tett langs sidene i bunnseksjonen og lage en tett flate.

4.2.4 Midt-aktuatoren

I midten av midtseksjonen er det nedfelt et trust nålelager hvor en aktuator plasseres. Denne aktuatoren er kernen av arenaen og er det som benyttes til å flippe og rotere dronen tilbake til utgangsposisjon. I tillegg skal et QI-lading system bygges inn slik at dronen vil ha mulighet til å lade når den er plassert nederst i arenaen. Dette er viktig da et av kravene for produktet er nettopp at dronen skal kunne operere uavbrutt uten bruk for vedlikehold over en lengre tid. Dette er derfor helt sentralt i designet for aktuatoren.

Diameteren som kreves blir derfor i stor grad diktert av coilen til senderen på QI-laderen som minstemål. På andre siden så vil dronen kunne bliliggende oppe på pinnen i midten hvis den er for stor. Så tynn som mulig vil derfor være målet for aktuatoren som helhet.

Bredden vil da også ha mye å si for hvor stor aktuator som kan drive denne pinnen. Det må derfor tas hensyn i designet for hvilket type aktuering som er aktuell. Det ble vurdert en rekke forskjellige metoder blant annet elektrisk lineäraktuator og forskjellige servomotor løsninger. Etter et bredt søk etter tilgjengelige lineäraktuatorer ble det funnet at disse bygger en god del både i bredden og ekstra i høyden i forhold til tilgjengelig utstrekning. En av de mer kompakte mulighetene var en elektrisk lineäraktuator fra RS Pro [55]. Denne bygger 106 mm mer enn tilgjengelig



Figur 4.8: Aktuator-sammenstillingen sett fra midten og ut, teknisk tegning i appendiks E.2

forlengelse trekt sammen, samt har en minste radius fra senter på aktuatoren på 58 mm. Dette ville da blitt minste indre mål på pinnen. Dette er uten montering da denne ville bygd 41 mm ekstra i høyden. Servomotor løsninger ville krevd enda mer da lengden på armen ville vært bestemmende for tilgjengelig utrekning. For å få en ønsket utstrekning på minst 100 mm ville armen minst måttet være 50 mm. Selv om det ikke er mye kraft som skal til ville kun maks vekten på dronen på 250 g 1.3, gi et moment på 0,125 Nm som allerede begynner å bli mye, selv uten å legge til friksjon og egenvekt av systemet. Dette er sett bort ifra av at plassen denne løsningen krever.

Det ble derfor bestemt at en løsning hvor en steppermotor benyttes sammen med en kuleskrue og et flexledd. Denne formen for aktuering er idag svært vanlig å benytte som aktuering på z-aksen på 3D-printere, og det finnes derfor relativ lett tilgjengelige deler, samtidig som det er et kompakt system som ikke bygger mer en høyden på stepper og kobling. Steppermotorer har også en relativ høy styrke og er relativt kompakte. Satt sammen med en skrustang mekanisme er den tilgjengelige kraften høy. Det er derfor mulig å velge en liten og kompakt motor som gir høy kraft. I figur 4.8 vises et utsnitt av hvordan steppermotoren er plassert i midten av plattformen med kobling og kuleskrue montert.

Ved å benytte løsningen med steppermotoren gir det også muligheten til å designe en aktuator som passer best mulig til det intenderte bruken og spesifikasjonene. Samtidig bringer det også inn større usikkerhet i prosjektet, ved å designe systemet selv, istedenfor å benytte en profesjonell og gjennomtestet aktuator. Denne løsningen ble derfor ikke valgt før konklusjonen om at det ikke fantes ferdige løsninger som ville møte spesifikasjonskravene.

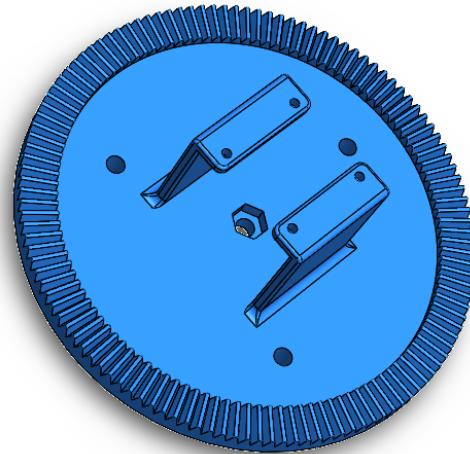
Rotasjonsplaten

I bunn av midt-aktuatoren finner man rotasjonsplaten, se figur 4.9. Dette er delen som midtaktuatoren bygges oppå, det som festets til midt-seksjonene og er også denne som muliggjør rotasjonen. Platen består av en rund seksjon med en flens nedover. Dette er laget for at et lager kan plasseres i underkant og ned i midtseksjonen. Rundt kanten av platen er 240 tenner for rotasjon. Disse er laget på en slik måte at de står med en vinkel på 1,5 grader i forhold til hverandre. Dette muliggjør bruken av et konet tannhjul for driften. Det er 4 hull gjennom bunnplaten.

Tre hull som ligger med en diameter på 95 mm og er 6.2 mm store. Dette for å kunne montere ledestengene. Det ble valgt å benytte M6 skruer for montering på grunn av størrelsen på de tilgjengelige rørene har en indre diameter på 5 mm, som passer bra for oppgjenging til M6. Hullene er overdimensjonert noe, for at ikke skruene skal gjenge i plastikken da printe-produksjonsmetoden, seksjon 2.2, naturlig vil bygge litt inn.

I midten av platen er et gjennomgående hull med en sekskantet nedsenkning. En av kravene for arenaen, se 1.3, er porterbarhet. Kun ved feste med lageret ville antagelig holdt midtaktuatoren på plass i stor grad under bruk, men ved forflyttelse vil slag eller brå bevegelser, eventuelt store vinkler kunne føre til utglidning mellom midt-seksjonen og lageret eller lageret og midtaktuatoren. Hullet i midten er derfor designet slik at en M6 sekskant maskingskrue kan festes under steppermotoren, og vil bli holdt på plass av denne, og videre ned i et hull i midtseksjonen. På undersiden av denne legges et nytt nållager med en bolt. Dette vil holde platen på plass selv ved uvoren forflytning.

På toppen i midten har platen to veggger med kant som benyttes for å skru fast steppermotoren i midten av aktuatoren og holde denne på plass. Den er rotert slik at steppermotoren kan dyttes inn

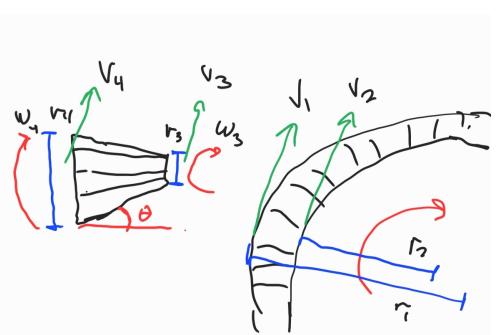


Figur 4.9: Grafisk illustrasjon av rotasjonsplaten, teknisk tegning i appendiks E.16

fra den ene sidene mellom to føringer.

Rotasjonsmotor og rotasjonstannhjul

Rotasjonsmotoren monteres på midtseksjonen ved hjelp av fire skruer og printede selvgjengende hull i midtseksjonen. På akselen til denne motoren plasseres et resin printet rotasjonshjul. Dette rotasjonshjulet vil ved hjelp av tenner rotere rotasjonsplaten og sørge for at midt aktuatoren kan roteres 360° . Som illustrert på figur 4.10 vil farten ytterst på tannraden på rotasjonsplaten ha en høyere fart enn innerst, vist matematisk likningene under. Det er derfor nødvendig med et konet rotasjonshjul for å ta høyde for denne effekten, og sørge for at rotasjonshjulet kan følge rotasjonsplaten. Dette er spesielt viktig med tanke på at det ble valgt rette tenner på rotasjonspalten uten gradskompensasjon.



Figur 4.10: Aksialfart vektorer fra rotasjonsskiven og rotasjonshjulet

$$\omega_1 = \frac{V_1}{r_1} = \frac{V_2}{r_2} \quad \omega_4 = \frac{V_3}{r_3} = \frac{V_4}{r_4} \quad (4.1)$$

$$\omega_1 = \omega_4 \Rightarrow \frac{r_1}{r_2} = \frac{r_4}{r_3} \quad (4.2)$$

$$r_3 = r_4 \quad (4.3)$$

Ledestenger

Valget av steppermotoren og kuleskrue som aktueringsmetode krever at konstruksjonen har en god stabilitet og støttes ved føringer i aktueringretningen. Her ble flere muligheter og variasjoner vurdert. Bland annet ble et design med v-slot rails vurdert. Både med en printet glider rundt dette med en sik plassert i midten og med muttere glidene inni profilen. Dette ble vurdert til å skape for mye friksjon og fra tidligere erfaringer med slike systemer vil det skape vibrasjoner som sliter på delene. Basert på designet av CubePro 3D-printere, se bilde 2.2, ble det foreslått et design ved bruk av runde føringer med axial kulelager.

For å minimere produksjonstid ble det valgt å benytte rør og lagre som var på lager. Et rør med ytre diameter på 8 mm og indre på 5 mm ble derfor valgt og gjengen opp, sammen med 8 mm axial-lager. I toppen festes disse sammen med en toppring som går rundt aktuerings-pinnen og holder ledestengene i riktig lengde fra hverandre. Denne toppringen har forsenkning for at ledestengene skal gå litt ned i hakket og skrus fast med en M6 skrus fra toppen.

Aktuerings-pinne og topplokk

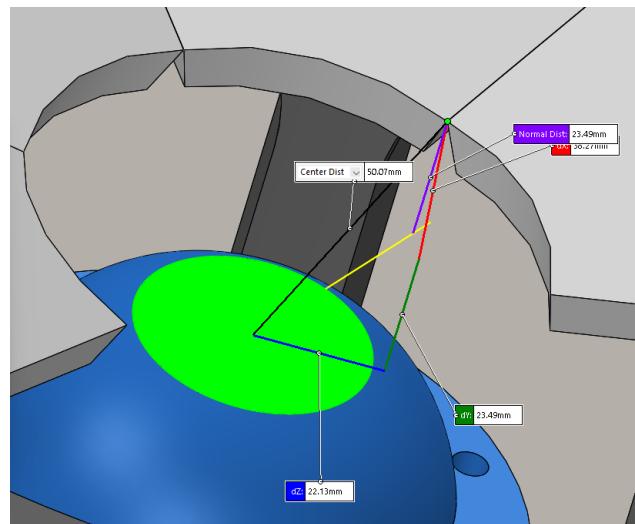
Selve pinnen for dyttingen og roteringen av dronen ble utviklet fra konseptet som vist i kapitel 3.2. Hovedsakelig er det tre eksterne faktorer som dikterer mye av formen og funksjonaliteten til pinnen. Diameteren er som nevnt tidligere diktert nedre mål av QI-ladespolen som skal passe inni denne. Det ytre målet henger igjen tett sammen med hvilken helning som er mulig i buret 4.2.2, samtidig som den må være stor nok til å kunne romme en steppermotor. Designet gjør at lengden henger nøyne sammen med lengden det er mulig å heise ”pinnen”. Det ble gjennom en rekke tester testet empirisk hvor langt opp dronen måtte være før den ville kunne flippe ved et fall ned fra en høyde. Det ble tatt litt ekstra høyde slik at det var litt å gå på. Det ble funnet at en fallhøyde på 100 cm skulle holde med 10 % margin. Marginene i høyden er svært små og romme for noe større margin er derfor vanskelig.

Det var også ønsket å kunne la pinnen gå noe nedover inn i hullet i bunn for å kunne rette opp dronen fullstendig ved å la den henge på armene på kanten. Det ble derfor designet slik at aktuatoren gikk ned under kanten på bunnplattene. Ved å gi skroget mulighet til å kunne ligge slik at dette massesenteret ligger under kanten vil skroget i stor grad forsøke å rette seg opp i hullet og oppnå den ønskede effekten. Ved å minimere de dimensjonene som ikke bidrar til mulig aktuering, ble det mulig å senke plattformen 23.5 mm ned under platekanten, se figur 4.11.

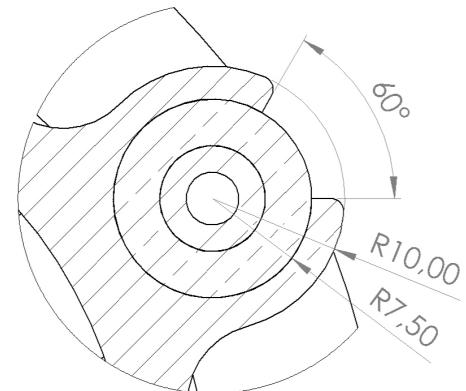
Inni pinnen er den hul i en cylinderform. Toppen er laget i en egen del for enkelt å kunne redesigne, men også installere deler her, noe som kreves ved implementeringen av QI-laderen. Toppen festes ved hjelp av tre M3 skuer som skrus inn i hull som passer rundt hodet i topplokket og inn i hull i sylinderen som ligger i en indre flens. Hullene er printet små nok til at M3 skruene gjenger i plastikken og dermed holder toppen godt på plass. Ved hjelp av denne flensen vil topplokket ligge helt kant i kant med sylinderen og dermed danne en tilnærmet uniform overflate.

I midten inni sylinderen er det laget et trearmet kors med en sylinder i midten. Denne skrus lede-mutteren inn i ved hjelp av M2 skruer. Dette vil da danne det sentrale punktet for heving og senking av sylinderen. All vekt som sylinderen utsettes for er derfor i utgangspunktet sentrert i dette punktet. Samtidig vil all kraft som ikke legges på direkte nedover skape rotasjon, som vil legge seg skjeft på lagrene og ledestengene og skape friksjon. Noe som vil skape økt friksjon og skape tre ekstra punkt for støtte mot rotasjon. Korset i midten av sylinderen trenger dermed ikke være veldig kraftig dimensjonert. I overgangen mellom korset og sylinderen er det laget fillets for å unngå stress konsentrasjoner.

Langs ytre nedre kant er det laget tre klikk punkter for feste av axial-lagrene som skal gå langs ledene. Patenten kan ses i figur 4.12, hvor den består av en semisikrel med et 60 °åpning. Den indre radiusen er 7.5 mm og med vegger som er 2.5 mm tykke. Dette designet vil gi mulighet til å dytte inn lagerene med en form for klikkpå plass form, men holder likevel nok motstand til at lagerene ikke sklir ut av formen. For økt vridnings-støtte blir det brukt to lagre per føring og dermed dobbel høyde på lagerholderen. Lagerene kan forsatt enkelt presses inn et fra toppen og et fra bunn.



Figur 4.11: Dimensjoner fra toppen av aktuatorplatformen til platekanten



Figur 4.12: Teknisk tegning av axial-lager holder, full teknig i appendiks E.2

Topplokk

Formen på topplokket blir i stor grad spesialtilpasset de krav som QI-ladesystemet stiller, samtidig som en enkel utskifting av dette med tanke på reparasjon og oppgraderinger er nødvendig. Toppen av aktueringspinnen er derfor laget for å kunne enkelt tas av ved hjelp av tre skruer som er forsenket inn langs nedre kanten. Dette vil også gjøre det enklere å installere ledninger og ledning trekk da man enkelt kan tre disse ned gjennom sylinderen og ut siden.

På siden av sylinderen er det også laget et hull for å holde ledningene fra QI laderen vekk fra steppermotoren og rotasjonsmotoren. Denne versionen kan ses i figur 4.13 Etter at QI coilen ble integrert ved å lage en slisse på topp tilpasset ledning coilen for å ligge helt rett med overflaten. Dette ble gjort kun for test av coilen og ikke implementert i total sammenstillingen da delen kom for sent til dette.



Figur 4.13: ATopplokk med integrert QI-sender, teknisk tegning uten QI 4.14

4.2.5 Bunnplaten

Alle delene monteres i bunn av arenaen på en bunnplate. Dette blir selve skelettet i arenaen, og vil stå for mye av det strukturelle avstivingen til arenaen. Platen er i utgangspunktet en sekskant hvor hvert hjørne er montert oppunder flensen på hjørnestøttene og oppunder midtseksjonen med midtringen som støtte under. Selve platen blir derfor løftet 7 mm over bakken.

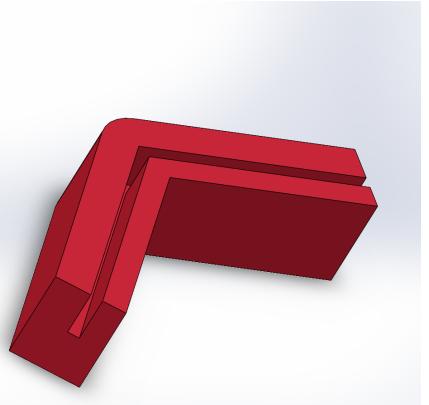
I to av sidene vil det monteres rack-skap for montering av pc til styring og annen styringselektronikk. I begge disse to vil det sørget for at det i skuffene er montert håndtak. For å få plass til disse seksjonene monteres de med 120 graders avvik. I den gjenværende siden monteres en skjerm 4.4.6 for å kunne vise nytlig og interessant informasjon for tilskuere under trening. For beskyttelse av skjermen monteres det også her et håndtak. Dette vil da gi arenaen tre steder med håndtak montert i bunnplaten for enkelt å kunne flytte eller transportere arenaen trygt. Dette vil da kreve en hvis styrke av bunnplaten da denne vil være det som bærer i.

4.2.6 Topp-seksjonen

Topp-hjørner

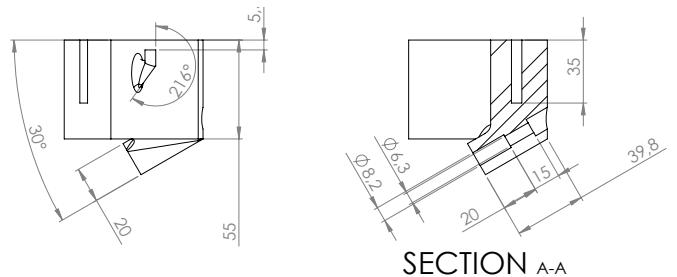
I toppen av glassplate veggene festes 3D-printede hjørner som benytter samme festemetode som bunn hjørnene. Slissen i disse hjørnene er som på bunnhjørnene 5 mm og dybden er 35 mm. Under første sammenstilling, den såkalte åpen-dag-sammenstillingen (5.6 ble det benyttet en tidligere versjon av hjørnene som kan ses i figur 4.14. Denne benytter i utgangspunktet kun vinkelen sammen med slissen til å holde veggene sammen. Topp nettet ble her stiftet på under spenn for å holdes på plass, samtidig som hjørnene ble boltet fast med to M3 boltr per hjørne. Uten boltene viste veggene seg å gli utover og lage relativt store gliper mellom seg.

I versjon 3 4.15 er en holder for stenger lagt til i toppen i en 30 graders vinkel. Fra denne rørformede holderen går et 6.3 mm hull gjennom og ut på utsiden hvor den igjen åpner opp til 10.5 mm. Gjennom dette hullet kan stengene skrues fast med en M6 bolt. På grunn av erfaringen med utgliding på versjon to ble det prioritert måter å holde sammen toppen.



Figur 4.14: Rendring av ToppHjørneV2

På utsiden av hjørnene er det to kroker, en per side, hvor nettet festes ved å stramme det ned over kanten. Nettet vil bli liggende på undersiden av topplaten og spilene. Når det da strammes ned over kanten vil også dette bidra til å holde sammen topp seksjonen. Ved å la toppseksjonen kunne festes helt både med og uten nett vil denne kunne monteres separat før den settes på og også bidra til å holde glassplatene sammen. Ved inspeksjon av bunnen må en av veggene løftes av, de sammenkoblede hjørnene vil gjøre det enklere å holde resten av platene sammen selv om to hjørner løftes av.



Figur 4.15: Utdrag av teknisk tegning av topp-hjørne V3, full teknig i appendiks E.9

Topp-spiler

Spilene er laget på samme måte som ledestengene for midt-aktuatoren. Disse er laget av et rør med yte diameter på 8 mm og indre 5 mm. Det er derfor enkelt å gjenge opp for en M6 skrue. Dette var en komponent som var tilgjengelig på materiallageret og ble funnet til å ha tilstrekkelig tverrsytre for lasten som var ønsket å bruke.

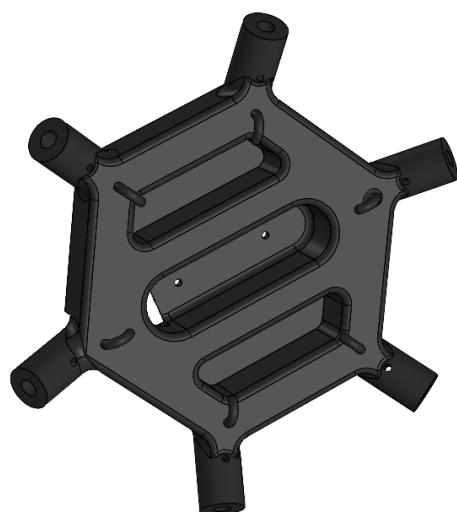
Lengden på stengene er strengt avhengig av designet på topp-hjørnene og topp-platen samt vinkel som var ønskelig. Dette var mulig å regne ut, men det ble tatt utgangspunkt i hvilken synsvinkel som var mulig å oppnå med kameraet 2.6. For å hindre at dronen i lengre perioder havner utenfor synsbredden til kamerat ble denne vinkelen plassert rett over synsrekkevidde. Dermed vil heller ikke toppen og nettet risikere å havne i synsbredden til kamera. Den endelige vinkelen ble valgt til å være 30 grader. Dette ble både top-hjørnene og top-platen designet for. Disse ble deretter linet opp med riktig avstand i Solidworks assambelyet og avstanden mellom innerste punkt på feste rørene målt. Denne ble funnet til å være 636,54 mm. Lengden på rørene ble derfor satt til å være 636 mm for å ikke risikere å lage for lange spiler. For korte vil naturlig rette seg ut under sammenstilling, til forskjell fra for lange som vil vri på vinkelen.

Topp-platen

Topp-platens funksjon er i all hovedsak å binde spilene sammen og holde nettet sammen, samtidig som det er plasseringen for kamera som brukes for monitoring, se figur 4.16. Kamera som benyttes er av den dyrere sorten 2.6, og for å beskytte dette mot dronen har det blitt bygget inn i platen i et tilpasset hull. Dette gjør også at det på oppsiden er laget et vertikalt hull inn hvor man enkelt kan koble til USB-porten på kamera. Kamera festes ved hjelp av to M3 skruer gjennom topplaten og inn i baksiden i festene på kamera.

På begge sider av kamera-forsenkningen er det kuttet ut to hull med lik dybde som kameraforsenkningen. Dette er for å spare materiale for produksjon. Det skal her nevnes at dette skaper vegger rundt dette hullet isteden for det mye mer material effektive infillet. Besparelsen av filament er derfor minimal, men designet ser mer helhetlig og mindre kompakt ut.

I hvert av hjørnene på den sekskantede platen er det plassert et mottaker rørfeste for spilene. Disse har en indre diameter på 8.2 mm for at spilene skal gli lett inni, og med en vinkel på 30 grader for å matche vinkelen på spilene.



Figur 4.16: Topplatens som grafisk illustrasjon, teknisk tegning E.24

Nettet

For åpen-dag-sammenstillingen 5.6 ble det benyttet en hønsenetting for å luke toppen. Nettet skal ha en konet form å må derfor antagelig tilpasses endel for å passe best mulig. Dette bør kunne løses lett men er ikke ferdig testet.

4.3 Drone-design

Dronen skal fly inni buret og trenes opp ved hjelp av AI, den må derfor tåle både å krasje hardt inn i veggene og falle ned fra toppen av arenaen og ned på aluminiumsplatene. Det ble derfor gjort mange vurderinger rundt dronen og materialbruk på dronen med hovedide om bruk av polypropylen plast som har gode egenskaper for slike fall 2.3.4. Det ble først testet litt med de tidligere versjonene for å finne ut hva som måtte endres. Etter hvert som justeringene skulle implementeres ble det funnet ut at det var nødvendig med et nytt dronedesign, grunnet behov av nye egenskaper og udefinerte sketsjer i den tidligere versjonen.

Behovet for å beskytte propellene er stort ettersom dronen kan krasje inn i veggene i buret. En propguard i polypropylen plast ble designet for å unngå at den knekker og for å beskytte propellene. Samtidig måtte designet på beskyttelsen være stift nok til at den ikke ble dyttet inn i propellene under støt med veggene.

For å stive opp drone skroget ble det designet en struktur i PLA plastikk. Denne strukturen ble designet får å kunne feste de nødvendige komponentene inni dronen samtidig som den økte styrken og effektiviserer sammensetningen av dronen. Både vekten og utformingen på dronen er essensielt for å få en effektiv og stabil drone. Denne innvendige strukturen gjør det derfor mulig å spare mye vekt på skrogveggene.

De tidligere skrogene ble printet i PLA plast for å kunne teste og visualisere dronen. PLA plasten viste seg å være veldig sprø når veggene kun var 1 mm tykke. Den ble så forsøkt å printes i polypropylen for at den ikke skulle sprekke. Dette endte med at printeren ikke klarte å printe skroget grunnet de litt utfordrende printe egenskapene til polypropylen 2.3.4. Under printen fremkom mye warping både på første laget ved byggeplaten og oppover i lagene, dette resulterte i at det ble flere sprekker i skroget og det fungerte ikke som tiltenkt. Skroget ble derfor endret til å være 2 mm tykt for å se om printingen gikk bedre med tykkere veggger.

Med 2 mm veggger ble skroget printet fint og det tålte mye, etter skroget ble lagt på vekta ble det også funnet ut at polypropylen har en mye mindre egenvekt enn PLA 2.3.4, 2.3.2. Skroget var fortsatt ikke stift nok og den første løsningen ble å bruke polypropylen i bunnen av skroget og PLA på topplokket for å gjøre det stift nok. Da kom det tidligere problemet med den sprø PLA plasten frem igjen. Dermed ble det bestemt å lage en innvendig struktur som kunne stive av dronen mens det er polypropylen plast på yttersiden av dronen for å absorbere støtene.

Etter at kunnskapen om printing med polypropylen plast økte og et telt for den aktuelle 3D-printeren (Artillery Sidewinder X2) ble tatt i bruk for å beholde en stabil temperatur under printingen 2.3.4 ble resultatene på skroget enda bedre. Dermed ble det igjen forsøkt å printe et skrog i 1 mm tykkelse. Selv om polypropylen er lettere enn PLA så er det viktig å få dronen så lett som mulig. Printen på 1 mm ble hel og sammen med støttestrukturen inni ble også dronen overraskende stiv.

4.3.1 Drone skrog

Konseptet for det nye drone designet tar utgangspunkt i det tidligere skroget som har motorene plassert i en firkant på $140mm \times 100mm$. Noen bestemte endringer ville implementeres i det nye skroget, det første for å gjøre dronen vanntett. Dette skulle gjennomføres ved hjelp av en list på bunnskroget som skulle klikkes inn i en slisse i toppskroget. På det tidligere skroget var sammenkoblingen mellom topp og bunn helt flat, dette ville endres slik at sammenkoblingen hadde forskjellige høyder for å øke styrken. Til slutt har konseptet ett mål om å at dronen skal bli lettere, som blir løst ved bruk av polypropylen og lavere total høyde på skroget. Redusjonen av høyden kunne lett implementeres grunnet at det ikke var nødvendig med en Pixhawk inni skroget lengre.

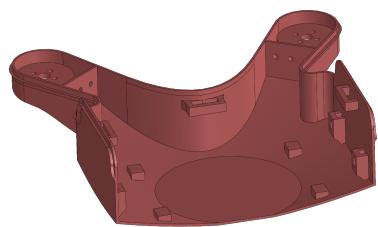
4.3.2 Skrogbunn

Bunnskroget er hovedsakelig designet utifra komponentene som skal plasseres inne i dronen. Alle komponentene skal festes og plasseres på en fornuftig måte. Siden de aller fleste komponentene er boksformet, så har kjernen av skroget fått en firkantet profil. Speiselt for å få plass til kretskortet med rpi cm4 montert i center. Innvendig er derfor profilen 97×62 mm, og vil da ha noen millimeter klaring i begge retninger til kretskortet. Den sirkulære profilen er lagt inn for å kunne plassere coilen til QI laderen helt i bunnen av dronen, da kommer den så nære transiveren som mulig. Sirkelen ble dannet samtidig som den skrå profilen til armene ble kuttet ut ved hjelp av revolved cut featuren. Kuttet fjerner da mye masse fra armene og lager samtidig en solid konet struktur som har god styrke mot oppoverbøyning i armene. I tillegg vil denne utformingaen bidra til at dronen letttere vil sentrere seg i midten av buret når den sklir ned i midten.

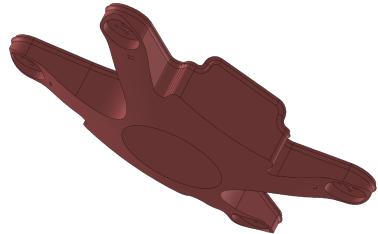
Et horisontalt kutt har blitt utført på toppen av dronene for å kunne feste motorene sentrert på totalhøyden til dronen. Ønsket om å plassere motorene i denne høydene er for å få dronen så balansert som mulig og få alle rotasjoner om massesenteret til dronen. Dette kuttet lager også en vegg som står 10 mm høyere enn armene. Dermed unngås ett flatt snitt mellom bunnskroget og toppskroget. Dette vil gjøre dronen stivere når den settes sammen og veggene var også nødvendige for å få festene til støttestrukturen som står inni. Senterne til motorene danner en firkant på 100×140 mm, dette er for å kunne bruke 3 tommers propeller akkurat uten at de kræsjer med skroget. Fire runde kutt underfra på armene lager en 2 mm tykk plate som motorene kan festes i med fire like lange skruer.

På alle fire armene har det blitt lagt til en vegg med 3 små hull i. Grunnet tidligere bør-krav om en vantett drone1.3 har disse veggene blitt lagt til, hvor man kan trekke ledningene til motorene mellom disse tre hullene. Etter ledningene er trekt gjennom hullene kan det legges på superlim for å tette mellrommet mellom ledningene og hullene de er trukket i. Dreneringshull har blitt kuttet ut i kammeret som fremkommer rundt motoren får å slippe oppsamling av vann. Langs kanten på toppen av bunnskroget har det blitt lagt på en kant ved hjelp av sweptbossbase og en 3D sketch. Kanten er der slik at topsskroget kan ha en vegg både på innsiden og utsiden av denne kanten. Dette vil gjøre at de to delene av skroget vil bli plassert helt korrekt oppå hverandre samtidig som det er enda ett tiltak mot vann. Hvis det så legges superlim i denne slissen vil dronen bli limt sammen og bli tilnærmet vanntett.

Som det kan ses i figur 4.17 er det fire små firkanter på bunnen av skroget. Disse er plassert der slik at de 3D printede batteriholderne kan støtte seg på disse. Batteriet vil da være med på å støtte opp bunnen på skroget samtidig som skroget demper støtet på batteriet når dronen krasjer. Hovedfestet til batteriholderne ligger i venstre og høyere vegg i figur 4.17. Grunnet at polypropolen er forholdsvis elastisk gjør det mulig å klikke batteriholderne inn i disse fire hullene hvor hempen vil låse den fast 2.3.4. Festene til topplata i støttestrukturen ligger på alle fire veggene. Her brukes samme prinsipp ved at en firkantet profil med en løse hempe presses gjennom de fire løkkene og sitter fast. Dermed vil det ikke være behov for verktøy under monteringen av støttestrukturen, som vil gjøre sammensetningen rask og enkel.



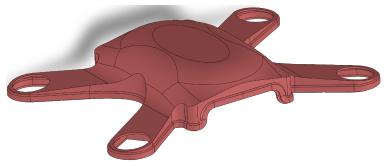
Figur 4.17: Bunn skrog section view



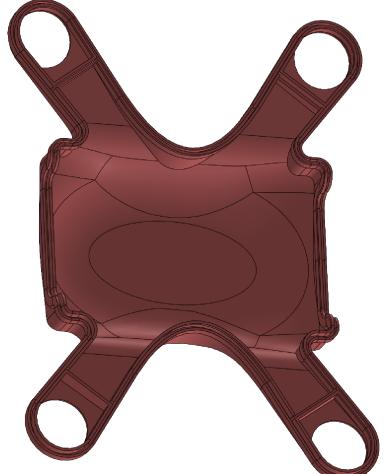
Figur 4.18: Bunn skrog fra siden

4.3.3 Toppskrog

Toppskroget er kun ett skall som settes oppå bunnskroget. Dette vil da tette dronen helt igjen og øke stivheten. Toppen er designet med samme tykkelse som bunnskroget på 1 mm tykkelse. Som det kan ses i figur 4.19 er utkappet på bunnen den inverterte til kappet på bunnskroget. På figur 4.20 kan man også se slissen som fylles med lim og klemmer seg fast rundt listen på bunnskroget. For å få så stor stivhet som mulig på armene er skråkuttet på den dratt så høyt som mulig uten at de treffer propellene. Videre har kuttet på toppen av dronen blitt dratt så langt inn som mulig i forhold til komponentene inne. Kurven på dette kuttet har blitt laget jevnt og slakt for å opprettholde stivheten, unngå skarpe kanter og firkantede profiler. Optimalt sett for stivheten burde den konete formen fortsatt helt inn til midten, men grunnet ønske om en lav drone ble det valgt en flat ellipse form på toppen. I tillegg var det viktig å ha en flat topp for å plassere ArUco-markøren. Oppå den flate toppen i figur 4.19 kan man også se en kuppel som er lagt inn for at dronen skal kunne snu seg om den lander på hodet. Når tåret da løfter seg opp vil dronen bli ustabil, velte rundt på siden og oppnå riktig orientering. Om dronen faller fra toppen av buret og ned på hodet vil kupelen også være med på å dempe støtet på dronen og komponentene inni.



Figur 4.19: Toppskrog ovenfra

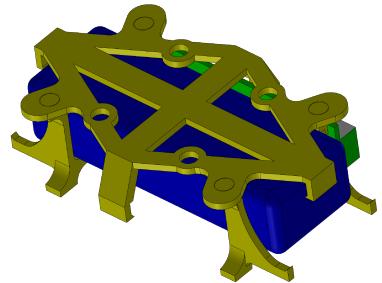


Figur 4.20: Toppskrog nedfra

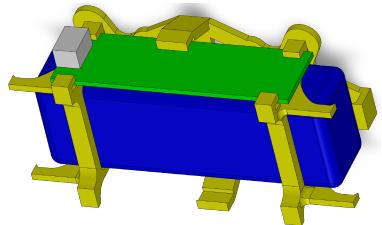
4.3.4 Støttestruktur

Støttestrukturen er helt essensiell i dronedesignet for å stive opp dronen og holde komponentene på plass. For å unngå mange redesign av skroget ble støttestrukturen med alle komponentene designet først for å ha kontroll på de minste målene som var mulig å benytte på skroget. Denne ble printet i PLA plast etter at et forsøk med polypropolen ble for mykt. Av hensyn til monteringen er strukturen delt inn i tre individuelle deler som settes sammen til en. Om det hadde vært en del ville den blitt stivere, men ville gjort monteringen mye vanskeligere. Det hadde blitt vanskelig å feste alle innfestningene samtidig til skroget, men når det er i tre deler blir det mulig å sette inn de to delene i bunnen først og deretter legge inn batteriet og låse alt sammen med topp platen til slutt.

Topplaten har blitt designet til å være 2 mm tykk for at den ikke skulle bygge for mye i høyden, men fortsatt ha en god stivhet. Platen måtte nå inn til de fire sidene i skroget, for å oppnå dette ble det laget ett parallelogram med ett kryss i midten. Dette danner en trekantet struktur i hvert av hjørnene som lager en stiv struktur med lav vekt. Fire hull har blitt lagt inn i hvert av hjørnene for å kunne feste inn de to siste dele i støttestrukturen. Disse hullene er designet med en diameter klarering på 0,3 mm til knottene som presses inn. Grunnet feilmarginene under 3D printing med Sidwinder X2 var dette en margin som gjorde at delene kilte seg sammen når man presset dem på plass 2.2. For å kunne feste flight controlleren til topplaten ble det lagt inn fire hull i et kvadrat med 30,5 mm i sentervavstand mellom hullene. Hullene ble valgt til å være 5 mm i diameter for å kunne presse inn gummi vibrasjonsdempere som kortet kunne festes inn i. Fire fester til bunnskroget var det siste som ble lagt til på topplata. To av disse ble lagt på høyde med platen, siden de skulle festes inn i samme høyde som de høyeste veggene på bunnskroget, mens de to andre måtte bygge litt ned for å nå sideveggene. De sist nevnte ble laget skrå utover for å nå ut til veggene og samtidig gjøre det mulig å lage topsskroget så smalt som mulig.

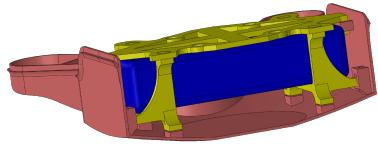


Figur 4.21: Oversiden av støttestruktur



Figur 4.22: Underside av støttestruktur

Sammen med topplaten danner de to siste delene i støttestrukturen en holder til batteriet. Disse delene er samme del bare at den ene er speilvendt, så det var bare nødvendig å designe en av dem. På figur 4.22 kan man se at det er lagt inne et feste til laderen på venstre siden. Laderen føres inn i de to delene føre de festes i skroget. De fire pinnene som stikker ut på bilde har en krok som huker seg fast i skroget. Den firkantede profilen som batteriet hviler på har fått en stor fillet i endende, denne måtte være der for at den ikke skulle knekke under kollisjonene. Rett under profilen ligger det fire små firkantede bein som skal lene seg ned på bunnskroget. Når det monteres er det ganske stort mellomrom mellom batteriet og bunnskroget. Tomrommet er lagt til for å kunne plasere en fremtidig pcb med esc'er og strømforsyninger til cm4 kortet. Som vist i figur 4.23 ser man at alt blir låst sammen når støttestrukturen blir montert til bunnskroget.



Figur 4.23: støttestruktur montert i bunn skrog

4.3.5 Prop guards

For å beskytte propellene under kollisjonene mellom dronen og arenaen har det blitt utviklet ett konsept på propguards. PP filament har blitt brukt til å printe propguardene, da dette filamentet er fleksibelt og absorberer støt godt 2.3.4. Første konsept av propguardene kan ses i figur 4.24, denne er et simpelt design for å ha lite luftmotstand og være lett. Ved hjelp av halvsirkelen rundt propellen var det tenkt at denne ville unngå at propellene traff veggene. Samtidig blir denne halvsirkelen plassert litt over propellene for å hindre at de krasjer med gulvet eller taket. Det var tenkt at det skulle printes ut fire stykk av denne propguarden for å kunne feste en på hver arm. Propgarden skulle da sitte helt tett inntil skroget og festes fast med skruene som holder motorene på plass. Første forsøk viste at PP print-en tålte mye og absorberte støt godt, men den var forholdsvis myk og kunne lett bøyes inn i propellene.

Denne ble derfor utviklet til et nytt konsept med en støtte over diameteren til halvsirkelen og en støtte inn mot skroget som kan ses i figur 4.25. Dette gjorde propguarden stivere og den beskyttet også mere over propellene, men når den dekker så mye over propellen går det også mye utover fly egenskapene til dronen.

Grunnet at propguarden ville begrense luftstrømninger ble det testet å legge alle fire propguardene inn i en tegning. Dette ble gjort for å prøve å stive opp ytterringen ved at de er festet sammen i en ring og unngår dermed behovet for avstivning over propellene.

Ettersom drone designet ble endret under utviklingen av propguardene, ble det gjennomført en endring i innfestningen av propguarden som i figur 4.26. Med det nye drone designet er det hull opp i motor festene og innfestningen på propgarden har da blitt omgjort til en sylinder som er 2 mm høy og en diameter på 19.8 mm. I denne sylinderen er det fire hull som sentreres opp med motorfeste hullene på droneskroget. Dermed vil sylinderen kunne presses opp i hullet på skroget og festes fast med motor skruene samtidig som de vil feste motoren. hullet i midten av hullene til motorfestet er der for å gi plass til rotoraksen til motoren.

Resultatene fra første testflyvning 6.7 viste at propguarden ble lett dyttet inn i propellene ved kollisjoner med veggene. Det går derfor en vegg opp på utsiden av sylinderen som legger seg på utsiden av skroget. Veggene er vertikal uten utsnitt til slissen som binner toppskroget og bunnskroget sammen. Dette gjør at propguarden vil bli presset litt utover av slisen på skroget. Nederst på utsiden av veggene er det lagt til en fillet på 4 mm for at den ikke skal huke seg fast i bunnpalatene i arenaen når dronen sklir ned mot midten. Armene som holder ringen rundt propellene ble forsterket og endret til en rektangulær profil for øke styrken. Den firkantete profilen fikk også fillets i hvert hjørne for å unngå stress konsentrasjoner. Siden de tidligere propguardene lett ble dyttet inn

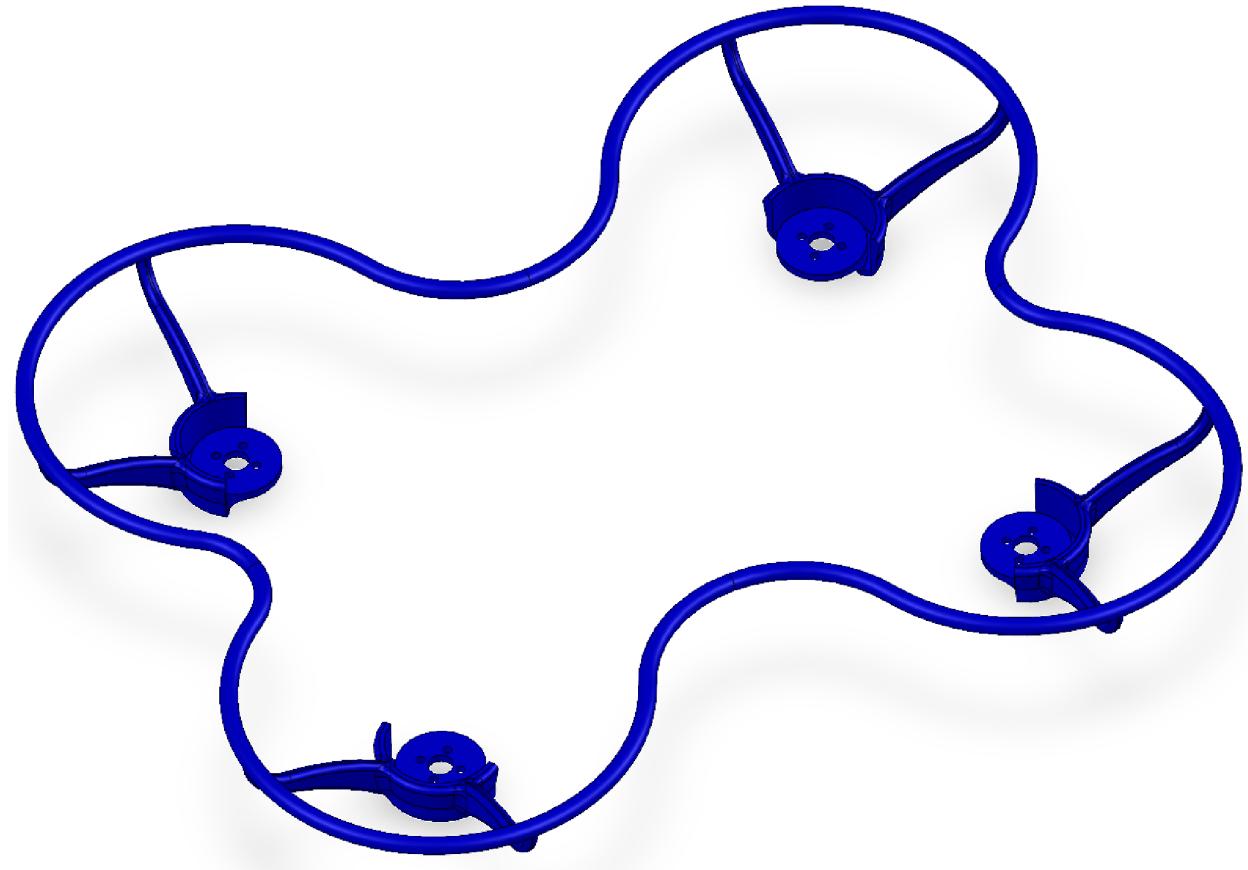


Figur 4.24: Propguard utkast



Figur 4.25: Propguard med ekstra støtte

i propellene ble radiusen på ringen rundt propellene økt fra 40 mm til 45 mm, som gir en klaring på 7 mm mellom propellen og ringen. Mellom propellene går ringen innover mot senteret av dronen. Dette er for å flytte så mye som mulig av vekta på dronen inn mot senteret, det vil gjøre det lettere for dronen å rottere om aksene.



Figur 4.26: Propguard endelig model

4.4 Komponenter og infrastruktur for maskinlæring

4.4.1 Beregningskraft

Når det kommer til muligheten for å implementere en kraftig form for beregningskraft er det sett på ulike alternativer som innebærer det å benytte plassen under selve arenaen, altså inne i "hubben". Dette gjelder både for den praktiske funksjonaliteten til arenaen, samt det faktiske systemet for reinforcement learning og dermed en Data for AI-trening. Med bakgrunn i dette vil det være dedikert en 19" 4U Rackmount til bygging av denne beregningskraften. Mye av bakgrunnen til at dette er ønskelig størrelse er hvor mye plass det kreves til bygging av en AI-kompatibel datamaskin, bestående av alle komponentene nødvendig for den omfattende jobben med reinforcement learning. En viktig komponent det er lagt opp til plassmessig er et Extended ATX-hovedkort i full skala, dette kortet bygger 305 mm × 330 mm [56]. Denne standardiserte størrelsen åpner opp for valg av hvilket som helst hovedkort, tatt i betraktning at den har den standardiserte EATX-størrelse eller er mindre. Dermed er det lagt opp til at man, uavhengig av det som omtales her, kan velge et hovedkort som er egnet til andre applikasjoner for et eventuelt annet prosjekt basert på Drone Arenaen.

4.4.2 Generelt om Hardware

Prosessen bak å lære en drone å fly av seg selv er kompleks, den består av mange individuelle deler som må kobles sammen til et helhetlig system. På grunn av oppsettet laget til Drone Arenaen er et viktig aspekt nettopp det at dronen skal lære å fly av seg selv, dette gjøres ved bruk av Reinforcement Learning. Denne prosessen krever mye regne- og prosesseringskraft noe en CPU, Central Processing Unit (prosessor), ikke vil være kapabel til å gjøre alene – mye grunnet de få kjernene den består av. Dette er kraftige, men få kjerner og egner seg derfor ikke for denne type datamengde, samt regning og behandling. På den andre siden eksisterer det en kjent løsning på dette nemlig ved bruk av en kraftig GPU, Graphics Processing Unit (grafikkort). Denne komponenten fungerer på mange måter som en CPU, men den dedikerer all prosesseringskraft til behandling av grafikkdata. Dette nevnes grunnet et system som Drone Arenaen er helt avhengig av å ha et kraftig grafikkort.

4.4.3 GPU

Med overnente informasjon som bakgrunn er et godt alternativ til valg av GPU en GeForce RTX 3090 produsert av NVIDIA. Dette er en svært sterkt GPU, og avgrenset til prisklassen "vanlig" får man antagelig ikke et bedre alternativ i skrivende stund [57], dog er prisen fortsatt høy, det er ikke til å utelukke. Samtidig om en sammenligner med kraftigere og langt mer egnet GPUer, får en mye for pengene ved valg av et RTX Grafikkort. Det finnes også mindre kraftig GPUer i RTX serien, 3080 og 3080ti er to andre alternativer som også kan fungere. Disse grafikkortene er sammenlignbare med 3090, men selvsagt ikke helt på samme nivå. Uavhengig av hvilken type som er ønskelig, er en stor anbefaling RTX-serien, da gjerne 30 versjonene, fra NVIDIA. En utfordring vedrørende et 3090 kort er etterspørselen opp mot tilgjengeligheten, dette er et grafikkort som er svært vanskelig å få tak i.

Når man kommer opp på et industrielt nivå er det mulig å få dedikerte GPUer/innretninger basert på GPU-spesifikasjonene til disse regne- og prosesserings-prosessene. Da er det snakk om enheter som ikke har et grafisk grensesnitt, dermed ingen mulighet for å bearbeide bilde som skal disponeres på en skjerm. Det er god grunn til dette oppsettet, mye basert på den enorme regnekraften GPUer har, siden de i utgangspunktet skal bearbeide et høyt antall piksler med høy frekvens og displaye dette med høy oppløsning. Det krever mange kjerner og ekstrem prosesseingskraft, når dette er byttet ut med dedikerte kjerner utelukkende til regning – vil ytelsen bli kolossal.

NVIDIA tilbyr slike systemer som er dedikert til utvinning og jobbing rundt Kunstig Intelligens. Om budsjett ikke hadde vært en faktor ville en slik løsning gitt et svært tilfredsstillende resultat, på kort tid. Derimot er disse systemene beregnet på AI trening i en større skala, derav den utrolige kapasiteten. Et eksempel på slik rå kraft er NVIDIA DGX A100, dette er et universalt system for bruk til alle AI-arbeidsoppgaver [58]. NVIDIA hevder selv at DGX A100 kan gi kraften til et

datasenter uten et faktisk datasenter [59]. Denne stasjonen er relativt liten i størrelse, og kan bringes med overalt - ergo er den svært bærbar. Dette hadde vært utrolig hendig til bruk i kombinasjon med et slikt system som Drone Arenaen bygges til å være, derimot blir det litt i overkant med kraft for å trenere opp en drone. Hvis dette skulle vært relevant ville skalaen nødvendigvis blitt hevet noen nivåer.

4U høyde gjør det mulig å plassere et Full-size grafikkort av den tidligere omtalte normale typen. Det er nok fornuftig med en slik tilnærming på GPUen, dette vil være det mest sannsynlige for denne arenaen, samt skalaen dette systemet skal arbeide i. Om andre aktører eller oppgaver skal benytte seg av arenaen og selv ønsker å bygge en egen PC for AI-treningen vil det være gunstig at andre typer GPUer også har plass, samt at de vil være brukervennlige og enkelt utskiftbare.

4.4.4 Hovedkort

Standardiserte størrelser på PC-komponentene er et fordelaktig utgangspunkt, derav vil et ATX- eller EATX-hovedkort være en god forutsetning. Basert på foregående erfaringer er hovedkort produsert av ASUS et godt alternativ, dette er pålitelige og gode hovedkort som vil gi høy ytelse. Fra ASUS hevdes det at de tilbyr «best-in-class» ytelse med intelligent overclocking og kjøling [60]. Herunder er det blitt sett på serien som heter PRIME, dette er en serie som er veldig fleksibel og innehar alle de nødvendige kvalifikasjonene og mulighetene. Videre blir anbefalingen snevret ned mot hovedkortet ASUS PRIME Z690-P WIFI, LGA1700 Motherboard, til applikasjonen som skal drives i Drone Arenaen er dette kortet mer en kompatibelt til å gjennomføre alle de nødvendige operasjonene. Fordelaktig med dette kortet er at det allerede er lagt opp til og kunne benytte 12. generasjon av Intel sine prosessorer [61], i skrivende stund er dette siste generasjon av Intel CPU som er lansert. PRIME Z690-P kortet er den standardiserte størrelsen ATX, noe som gjør at den vil passe helt fint inn i 4U-racken den potensielt skal stå i. Det er viktig at hovedkortet må være kompatibel med egnede CPUer, i tillegg til å inneha de portene som trengs. ASUS kortet huker av for majoriteten av disse punktene og vil dermed være det anbefalte alternativet.

4.4.5 CPU

CPU er forkortelsen for Central Processing Unit og er det som på norsk heter Prosessoren, dette er selve hjertet til enhver datamaskin uavhengig av hva slags applikasjon den blir brukt til. Det kan strekke seg fra en helt enkel spillautomat og opp til et sofistikert og avansert AI-system [62]. Når det kommer til valget av CPU for denne applikasjonen er det fordelaktig å gå få de nyere typene. Nyere generasjon vil naturligvis gi vesentlig bedre ytelse, dette er som følge av forbedret arkitektur og samhandling internt i prosessoren. Det er derfor ønskelig å ha flere kjerner i CPUen, det vil gjøre at datamaskinen kan belage seg på bedre multitasking - da en fusjon mellom flere kjerner kan bli benyttet til å gjøre én oppgave og dermed blir prosessorkraften fordelt utover flere kjerner.

Tråder i en prosessor sier noe om antall oppgaver en CPU kan utføre samtidig. Intel sine gode prosessorer innehar Hyperthreading, dette er kort fortalt en evnen hver kjerne har til å utføre to oppgaver samtidig. Dette medfører at det som regel blir brukt CPUer med 4 kjerner og 8 tråder eventuelt 6 kjerner og 12 tråder på nivåene som ikke krever for mye prosessorkraft. Om ytelsen må økes for mer avanserte oppgaver vil det være fordelaktig å øke til betydelig flere kjerner. Disse ekstra trådene er på langt nær like raske som hovedkjernen, dette er fordi de bruker den resterende plassen til kjernene effektivt. Med dette som grunnlag ønskes det en prosessor med høyt antall tråder i harmoni med effektive kjerner. Hvor klokkeraten er høy, altså hvor raskt de individuelle kjernene ”runner” [62].

All denne informasjonen er det essensielle når det kommer til valg av CPU for denne applikasjonen. Det er lagt opp til at en hvilken som helst CPU skal kunne benyttes, dette er i samsvar med det valgte hovedkortet. I skrivende stund er det 12. generasjon av Intel sine i9 prosessorer som er mest aktuelle, dette er CPUer med kolossalt høy ytelse og vil kunne evne alle applikasjoner som kreves av Droneareanen. Den har de høyeste klokkehastighetene og en banebrytende ny arkitektur som fører til revolusjonerende ytelse og multitasking, i følge Komplett sine sider på prosessoren [63]. Anbefaling vedrørende den ideelle prosessoren vil derfor lande på Intel Core i9-12900K CPU.

4.4.6 Skjerm

Det er gjort opp tanker rundt det å implementere en skjerm, her skal det være et visuelt kult brukergrensenitt som kan visualisere de forskjellige genrasjonene av programmet med Reinforced learning på drona. Dette har ikke blitt prioritert med bakgrunn i at det ikke eksisterer en fungerende PC hittil. Oppsettet er lagt opp til at en hvilken som helst skjerm kan benyttes, så lenge størrelsen er under $516mm \times 330mm$ ($20,3'' \times 12,99''$), dette betyr at det akkurat skal være plass til en skjerm på $23''$ inne i den tiltenkte lokasjonen da denne størrelsen bygger akkurat mindre i høyde og bredde.

4.5 Kamera

For å finne plassering på dronen inne i arenaen er man avhengig av koordinater og posisjonen til dronen. Dette er verdier som er vanskelig å hente ut uten bruk av noen form for referanser eller referanse-verdier. Den ønskelige posisjoneringen tar utgangspunkt i posisjonen til dronen, i kombinasjon med orienteringen. Derfor kan det være fordelaktig å hente ut informasjonen i form av xyz-verdier, hvor x og y er orientering og lokasjon, mens z vil være dybden - sett fra kameraet sitt perspektiv

Gjennom prosessen med valg av kamera er det en rekke kriterier som må overholdes, systemet er avhengig av å ha en plattform som er kompatibel med ROS2, OpenCV og Python eller C++. Gjennom dette vil det settes enklere forbehold når det kommer til oppbygningen av softwaren som senere vil gi posisjonen til dronen, dette er grunnet tilgangen på bibliotek og eksempelkoder. Videre er en innspillende faktor tilgjengelighet, kamera av den nødvendige kaliberen er en dyr komponent – det vil derfor være til stor hjelp om et kvalifisert kamera allerede kan være til disposisjon. På grunn av gruppens tilgang på komponenter gjennom Universitet er dette noe som er innenfor rekkevidde. Lageret som blir disponert har flere kameraer som er mer enn kvalifisert nok til å gjennomføre oppgaven vedrørende posisjonering i Arenaen. Det ble derfor undersøkt hva slags andre kriterier som vil være nødvendig for et passende kamera til applikasjonen.

For at ikke designet av toppen skal bli for høyt vil et kriterium for kameraet være et bredt Field of View, FoV er forklart i kapittel 2.6 Kamera Krav. Dette medfører at kameraet kan plasseres lavere og dermed få et bedre utgangspunkt for implementasjon i Drone Arenaen, mer om designet av Toppen er i delkapittel 4.2.6. Denne plasseringa gjør at et nytt kriterium må bli tatt i betrakting, dette omhandler hva slags avstand kameraet kan fange opp klare bilder i. For Drone Arenaen vil dette si avstanden dronen vil være fra kameraet, altså oppløsning i Z-aksen. Dette er en faktor som er viktig for valg av kamera, da dette vil bestemme hvor høyt dronen kan fly før den termineres - dette for å unngå unødvendig skade på både dronen og eventuelt arenaen.

Basert på foregående informasjon ble det undersøkt hva slags alternativer som er relevante og oppfyller kravene. Kandidatene som pekte seg ut var kameraer fra Intel, nærmere best RealSense-serien, Arzure Kinect produsert av Microsoft og et ordinaert WEB-kamera av typen Trust WEBCAM EXIS. De kameranene av relevans er kapable til og implementeres med ROS2, OpenCV og Python eller C++, derfor vil ikke dette være utslagsgivende for valget. Kameraet skal stå fastmontert, dette utelukker nødvendigheten med ekstrene sensorer slik som IMU. Tatt dette i betrakting vil et egent kamera være Intel RealSense Depth Camera D435, det innehar de nødvendige funksjonene og generelt er spesifikasjonene i tråd med det som kreves av systemet. Kameraet er presentert i Figur 4.27 og informasjon er presentert i delkapittel 2.7. En ekstra fordel D435 innehar muligheten til å måle avstanden fra kameraet til dronen på to forskjellige måter. Den ene metoden er ved bruk av den innebygde funksjonen for dybdemåling, den andre baserer seg på samme grunnlag som skal brukes for posisjoneringen altså ved bruk av ArUco-markør, utfyllende om dette i seksjon 4.6.



Figur 4.27: Intel RealSense Depth Camera D435.

Som fremlagt i delkapittel 2.7 er modulene plassert med litt avstand, dette skaper et grunnlag for utfordring ved bruk av dybde-funksjonen med RGB-modulen som hoved-ramme. I enkelte tilfeller har D435 problemer med å detektere nøyaktige avstandsmålinger om arealet av det den skal måle er for lite. Det betyr at det å hente ut dybde ved bruk av IR-modulen i den vanlige kamera-feeden kan føre til feilavlesninger på dybden. Det ble i den sammeheng utviklet et Python-script for å kunne teste dette ved bruk av en empirisk metode, forklaring av scriptet finnes i delkapittel 4.10.3. Man kan gjennom dette scriptet velge et sted i Color-framen, altså den vanlige kamera-feeden, med musepekeren og få ut dybden gjennom bruk av IR-projectoren. Gjennom dette forsøket kan det bli avdekket to ting som er vesentlig å ha med videre, det første er målinger om minste avstand kameraet kan detektere i Z-retning. Det andre er hvor stor innvirkning og hvor utstellende avstanden mellom RGB-modulen og IR-projectoren vil være for detektering av avstand i fusjon med avlesning av ArUco-markør.

Simultan med posisjonering skal kameraet ha funksjon som en "kill-switch" for dronen, denne skal optimalt implementeres i programvaren om dronen kommer for høyt - altså overstiger en satt maks-verdi. Dette vil gjøre at potensialet for skade vil minimeres på selve dronen og arenaen. I praksis vil det fungere slik at dronen blir terminert i luften når den overstiger en bestemt høyde. Dette medfører at kontrollen over fall-høyden vil være justert i henhold til hvor hardt dronen håndterer å falle, samt risikoen for at dronen krasjer hardt i taket vil nødvendigvis ikke være en faktor.

4.6 Detektering med ArUco

All annen informasjon foruten høyden av dronen vil bli avlest ved bruk av RGB-modulen på D435 kameraet. Disse målingene skal gi grunnlag for flere operasjoner, inkludert hvilken orientering dronen skal ha i det den letter, avgjøre om dronen ligger snudd på hodet eller har riktig orientering. Den skal være referanse for "Killswitchen", samt lage grunnlaget for videresending av målinger og avlesninger til kalkulasjonsenheten for Reinforcement learning. Disse avlesningene må være mest mulig effektive, i form av hurtighet og nøyaktighet. Et komplekst system som er satt opp til RL er avhengig av å ha disse parameterne så gode som mulig. Dette er et følge av at tradisjonell reinforcement learning baserer seg på at 'the feature identification' blir etablert av brukeren, dermed er kvaliteten på RLen definert av hva slags input systemet får [64].

Denne inputen kan bli lest av ved bruk av flere metoder, ofte blir det tatt utgangspunkt i OpenCV biblioteket i lignende applikasjoner eller systemer, eksempelvis er det blitt brukt til å lage et omfattende drone-deteksjonssystem ved bruk av AI [65]. Det er også viktig at metoden er kompatibel med RealSense D435, dog kan dette kameraet måle dybde uten bruk av markør jfr. delkapittel 2.7.

Det er nå redgjort for hva slags bibliotek som skal benyttes, dermed gjenstår hvilken markør som skal brukes for detektering av orienteringen. Et alternativ som peker seg ut er en ArUco-markør, dette er en markør som kan benyttes for lignende applikasjoner, eksempel er et landingssystem som er blitt utviklet basert på ArUco-markører som gjenkjenningsreferanse [66]. Som presentert i delkapittel 2.4 er dette en effektiv metode for å uthente betydelig informasjon, uten at det går på bekostning av hvor raskt systemet fungerer eller presisjonen på målingene. Det finnes flere lignende markører, April-tag og QR-kode, men for systemet Drone Arenaen er ArUco-markør det naturlige valget [34]. Relevant informasjon som kan hentes ut ved bruk av denne markøren kan være ID, orientering, størrelse og posisjon - uten at det opptar betydelig mengde bit jfr. 2.4.

Som følge av at ArUco-markører er omfattende brukt, finnes det eksempler på tidligere prosjekter som har en viss tilnærming til det markøren skal brukes til i Drone Arena-sammenhengen. Ideelt vil total detekteringen gi tre verdier - tilsvarende X, Y og Z. X og Y tilsvarer orientering og posisjon, de vil komme fra ArUco-markøren, samtidig som Z vil tilsvare dybden og kan hentes ut ved bruk av IR-projektoren på kameraet. Siden hver enkelt ArUco-ID har et unikt utseende vil orienteringen kunne bli avlest uavhengig av hvordan dronen oppfører seg, tatt i betraktning at den ikke ligger opp ned, eventuelt andre grunner til at ArUco-markøren ikke er synlig for kameraet.

4.7 Komponenter for styring av arena

Kalkulasjonsenheten er i stor grad diktert av hvilke spesifikasjoner som kreves av reinforcement leaning mekanismene. Denne enheten kan derfor ha mye igjen for å kunne bli byttet ut enkelt, samt bytte deler. Med dagens utvikling til grunn ser det ut som grafikkortene og kalkulasjonskraften vil fortsette å øke i utvikling. Det er derfor rimelig å tro at dette vil føre til at man vil kunne tjene mye på å bytte ut disse delene og at dette er ønskelig om få år. Det ble derfor valgt å plassere styringskoden for aktuatoren i drone arenaen på en enhet som kommuniserer med denne PC'n slik at minimalt med kode er avhengig av arkitekturen på kalkulasjonsenheten.

4.7.1 Raspberry Pi 4

Denne enheten ble valgt til å være en Raspberry Pi 4. Fordelen med dette systemet er at det er en plattform som fungerer med den valgte kommunikasjonsplatformen, og kan kjøre Ubuntu. Under utvikling ble det valgt å boote i Raspbian 64 bit da denne byr på et enkelt brukergrensersnitt og Thonny for lett utvikling av software for styring.

Raspberry Pi har også maskinstøttet IO-pins med et bredt utvalg av type porter og styringsprotokoller. Det ble i all hovedsak benyttet tre forskjellige type porter. Noen funksjoner er bare tilgjengelig på spesifikke pinner, men de fleste er mulig å aktivere på de forskjellige pinnene.

En del porter er satt til høy eller lavspenning for aktivering og deaktivering av funksjoner. Dette utgjør majoriteten av de pinene som tas i bruk. For dette blir GPIO biblioteket benyttet. Biblioteket er det mest brukte biblioteket for styring av IO-pinns på Raspberry pi og støtter digitale signaler som standard.

For styringen av hastigheten på rotasjonsmotoren benyttes et PWM signal som sendes til motorkontrolleren. Raspberry Pi 4 har maskinvarestøttet PWM signal på pinnene 12, 13, 18 og 19 men fordelt på to generatorer, så man har kun to tilgjengelig. Det skal også nevnes at dette går utover muligheten for å spille lyd. For å skrive PWM signaler benyttes GPIO.PWM biblioteket. Dette har en nøyaktighet ned til 1u [67] og gir dermed python programerte koder en fordel over C++ koder når de kjøres på Raspberry Pi. Dette reiser også fort spørsmålet om i hvilken grad en kode som kjører på en enhet som Raspberry Pi er i stand til å nå kravene for sanntid.

Raspberry Pi er en såkalt "single board computer" altså en enhet som kjører et operativsystem hvor mange oppgaver behandles samtidig. Operativsystemet vil til enhver tid prioritere de oppgavene som er viktigst for driften av computeren. Dette fører til at programmer kan nedprioriteres uten noen måte å hindre dette på. Man kan dermed ikke vite med sikkerhet at programmet vil kjøre så fort som mulig eller med konstant hastighet. For oppgaver som krever samtidsegenskaper vil derfor denne typen plattformer være uegnet, og en regn mikrokontroller programmert i C++ ville kunne gjøre oppgaven bedre. Kravene til dette systemet er i stor grad ikke basert på hvor fort dronen

resettes, og dette er i stor grad heller ikke et ønske da kjappe bevegelser vil kunne skade dronen. Av den totale stilletiden vil også ladingen og den tiden dette tar utgjøre en mye større andel og de totale samtidskravene er derfor relativt lave. Ved sene bevegelser vil også dronen kunne lade under forflytting og optimalisere tidsbruken ytterligere.

4.7.2 Stepperdriver DVR8825

For kontroll av midtaktuatoren trengs det en stepper motor for kontroll av høyden og en DC motor for kontroll av rotasjonen. Disse blir drevet av hver sin motorkontroller. Ved valg av disse ble det sjekket et utvalg av varianter for å sørge for at riktig komponent ble valgt. Begrunnet kravet om at dette er en installasjon som skal kunne brukes som en utstilling når den er operativ er det at steppermotor driveren har mulighet for microstepping svært viktig. Dette gjør installasjonen betydelig mer stillegående, noe som vil gjøre installasjonen betraktelig bedre å stå å se på. Motoren trenger ikke å kjøres i dipol konfigurasjon og driveren trenger derfor ikke å støtte dette. Separat motorforsyning er derimot sentralt da valget om å kjøre på en Raspberry begrenser utspeisningingen tilgjengelig til 3.3v volt.

Nyere type steppermotor drivere styres vanligvis gjennom en STEP pin hvor høyt signal betyr et steg, og en DIR pin for kontroll av retningen. Dette er en mer moderne måte og styre på og vil sammenliknet med en DVR8815 hvor hver spole aktiveres med høyt signal gi mer jevn kjøring, da styringskoden allerede er implementert på chipen. Det gjør det også mulig og styre steppermotorenn ved hjelp av et PWM signal generert av Raspberry pien for rask og god styring ved hjelp av justering av frekvensen.

Det finnes et utvalg nyere drivere på markedet som tilbry "Stealth Chop", som byr på enda stillere kjøring av stepper-motorene. Grunnet installasjonens krav som utstillingenhet ble dette vurdert og testet ut ved hjelp av en TMC2209. Ved sammenlinking av kjøringen ble det funnet til å ikke ha veldig stor effekt når begge ble kjørt i microstepping modus.

Grunnet det tilbakevendende problemet med leveranse ble derfor DVR8825 valgt som det beste alternativet for valg av stepper-motor driver. Dette er en av de vanligste driverene på markedet, er gjennomtestet, god tilgjengelighet med tanke på biblioteker og kjørte stille nok til at det ikke var problemer under kjøring. Tilslutt ble tilgjengelighet den avgjørende faktoren, da DVR8825 er en komponent som var tilgjengelig på lager på universitetet i et større kvanta. Dette gjorde det mulig å starte tidlig å utvikle kode for driveren og teste aktueringen uten store forsinkelser. Skadde eller ikke fungerende drivere kan også rask bli byttet ut uten å risikere lang ventetid.

4.7.3 Motordriver TB66FNG

En motordriver er nødvendig for å kunne styre DC motoren som benyttes for rotasjon av midtaktuatoren. Dette for å kunne kontrollerer hastighet samtidig som en høyere strømstyrke benyttes. Dette er viktig med tanke på motormomentet. Giringen til rotasjonsplaten består av en tanring. Denne hjelper noe for å senke hastigheten samtidig som den valgte motoren har et innebygget gir. Dette vil kunne gi høyere presisjon på rotasjonen og muliggjøre en saktere kjøring av systemet for å få en rolig rotasjon slik at dronen ikke vil bli kastet av aktuatoren.

Undre testing ble det først benyttet en L298N, da denne var lett tilgjengelig og svært familiær gjennom en rekke andre prosjekter. Dette er en motorkontroller av gammel arkitektur og har et betydelig spenningsfall gjennom chipen. I databladet finner man stort sett et spenningsfall på 1.4 volt ved maks pwm og økende ved synkende signalspenning. Dette gjør L298N til en lite aktuell driver for mer moderne arkitektur og systemer. Dette spenningstapet bygges opp som varme og en stor kjøleribbe er nødvendig. Ved tidlige testing har problemer med varme og at driveren kutter ut på grunn av overoppheating blitt oppdaget [68]. Sammen med ønsket om å legge chipene på en PCB for enkel plugg ut, plugg inn arkitektur 4.8 egner derfor L298N seg svært dårlig.

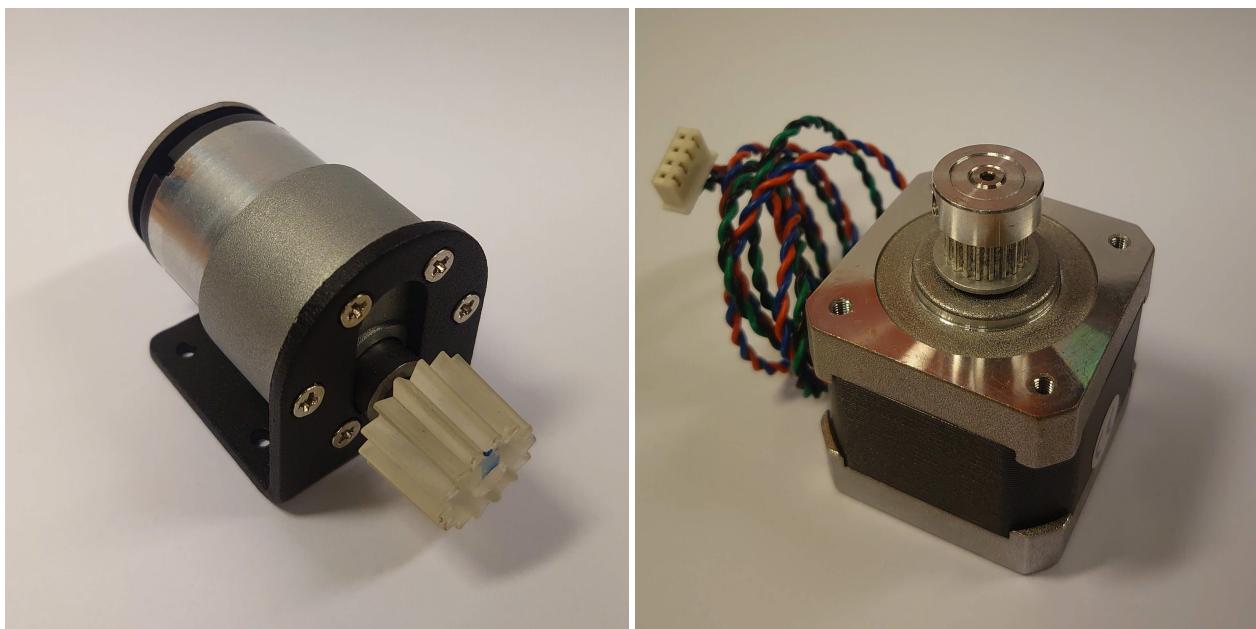
Flere motordrivere ble vurdert blant annet DRV8838 [69], da denne har mulighet for en relativt høy stabil strøm samtidig som den kommer i en kompakt størrelse. Det er heller ikke behov for mer enn en utgang på chipen. Samtidig ble leveringstider igjen avgjørende for det endelige valget. Tilgjengelig var SparkFun's versjon av TB6612FNG [70]. Denne tåler relativ høy strøm per kanal, har en nyere type arkitektur og vil passe godt inn som en pin header arkitektur på et PCB. Denne ble

derfor benyttet for videre prototype med komplett oppsett på breadboard for sjekking av systemet før bestilling av PCB.

Etter at PCB designet var komplett viste det seg at versjonen som var benyttet på kretskortdesignet var Polulu varianten som er 2.54 mm smalere og har jord og motorspenning pinnen byttet om. Dette gjorde at den intenderte SparkFun varianten ikke passet og ville potensielt sett brent hvis koblet opp direkte uten å bytte om VMOT og GND pinnene. Polulu-varianten ble derfor bestilt. Denne varianten er oppgitt med en lavere kontinuerlig strømverdi på 1 ampere. Dette burde imidlertid ikke utgjøre noen stor forskjell på grunn av last syklusen og den lave hastigheten.

4.7.4 Stepper motor og DC motor

,



(a) Nema 17 2 stack Stepper motor

(b) DC motor med encoder

Figur 4.28: Brukte motorer

Lasten for både stepper-motoren og DC rotasjons-motoren er som tidligere intendert svært lave. Disse ble derfor med tanke på kost og leveringsalternativ benyttet av deler som var lett tilgjengelig. For rotasjonsmotoren ble en giret motor fra Chihai Motor, nummer SKU434373 benyttet. Denne har et innebygget gir på 1:30 og innebygget encoder. Dette er viktig for stabil kjøring av rotasjonen og gjør at PID kontroll kan benyttes. Optimalt sett burde denne motoren byttes ut med en motor med en bedre sporbarhet da denne modellen ikke har datablad eller noe form for dokumentasjon. Det ble imidlertid nedprioritert da dette ville ført til svært lite tid til å utvikle kode og som en sentral del av designet gi liten forutsetning for å kunne lage den fysiske arenare. Festene er midlertidig laget svært modulært tenkt slik at motoren enkelt kan byttes ut hvis bedre alternativer finnes.

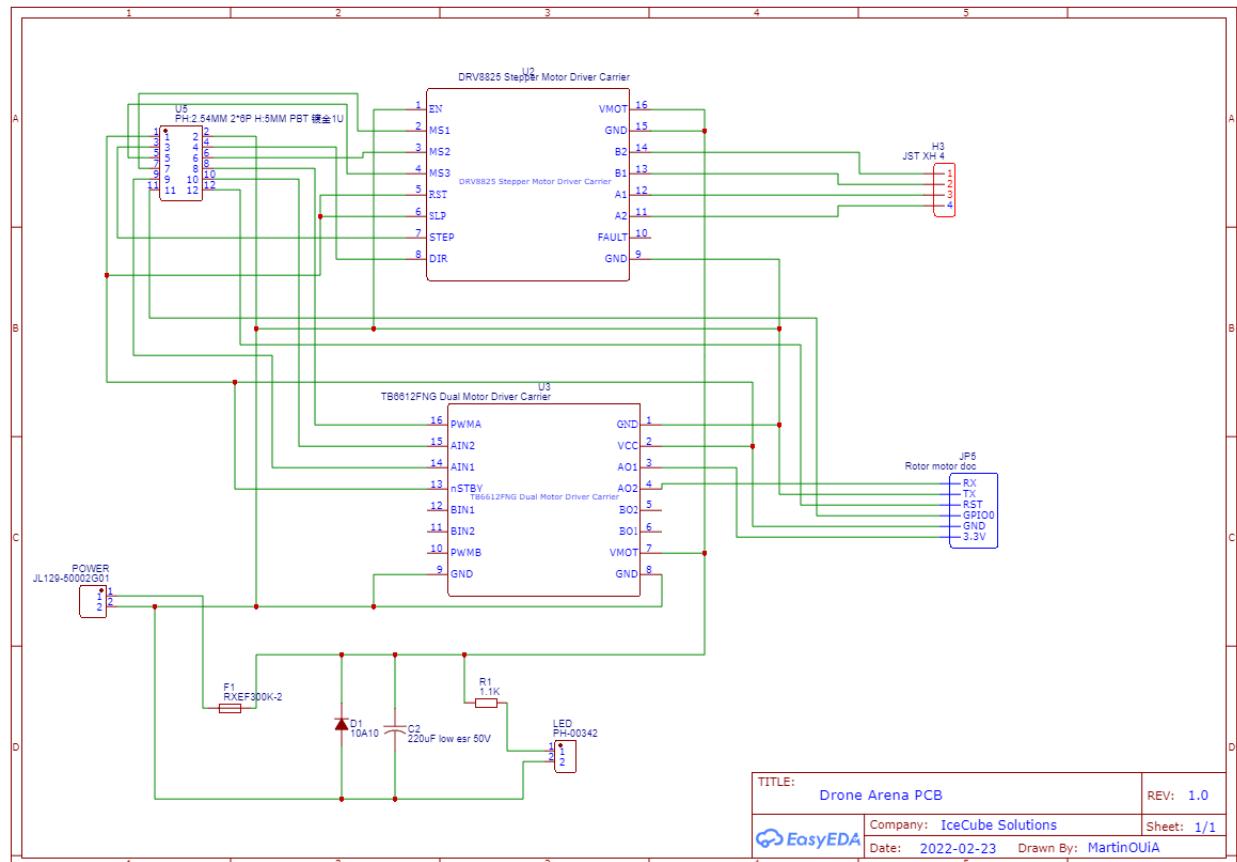
Steppermotoren ble hentet fra en gammel Ultimaker 2 Go, da det tidligere har hatt gode erfaringer med disse med tanke på kvalitet og holdbarhet. Dette er en 1.8 grader dobbel stack stepper motor. Dette er svært standard oppbygging og gjør at denne kan byttes ut med de fleste typer steppermotorer. I starten var ikke motstanden kjent og et annet oppstabiliseringsystem for midtaktuatorene var konseptert. Det ble derfor valgt å designe for en dobbel stack for å vite at det var nok kraft i systemet. Etter valget om å gå over til lineære hjullagre på ledestagene ble tatt kunne antagelig stepperen vært nedskalert til en enkel stack. Pakkingen av Ultimakerstepperen er midlertidig svært god og det er ikke mange millimeterne å vinne på å gå for en billigere enkelt stack. Ved vurdering av dette ble det funnet at dette ville gitt 5-6 mm, og ble ikke vurdert viktig nok for redesign.

4.8 PCB design

For å øke reproducertbarheten og minimere sjansen for feil grunnet oppkobling ble det bestemt at det skulle designes en PCB for styring av aktueringen i arenaen. Dette vil også føre til enklere vedlikehold og oversikt da test oppsettet ble for uoversiktlig. Dette var som følge av et utall ledninger koblet opp gjennom et breadboard, det viste seg fort at dette var uoversiktlig og dermed vanskelig å detektere feil. Raspberry Pi'en og styringssystemet er tenkt til å monteres i en av 2U rack skuffene for enkel utskifting av komponenter og tilkobling av strøm. Dette gjør at det enkelt kan skiftes motorledninger eller motorer uten noe nærmere forståelse av det elektriske systemet. Det er også tenkt at dette skal gjøre det enklere å produsere arenaen i et større kvanta da alle porter er markert og ingen nærmere forståelse kreves.

PCB'en tar utgangspunkt i at kun komponenter med pinner for på loddning skal brukes og har ingen overflatemonterte komponenter. Dette stammer hovedsakelig fra flere av lab-ingeniørene ved avdeling for Mekatronikk ved Universitetet i Agder, som vil være ansvarlige for vedlikehold av Arenaen. Etter samtal angående tidligere produserte prosjekter ved UiA hvor overflatemonterte komponenter ble benyttet kom det fram at dette hadde skapt store utfordringer med tanke på vedlikehold da kompetansen i henhold til dette var lav [71]. Flere av chipene hadde røket eller stoppet å fungere som førte til at hele enheten ble skrotet. Dette da arbeidet med å skulle finne erstatning, lese seg opp på manualene og benytte riktig utstyr kostet for mye for at arbeidet ble prioritert. Det ble derfor ytret et ønske om å ha komponenter som kunne plasseres i pin headers slik at hvis en driver røk eller liknende vil det bare være å nappet ut og sette inn en ny, og dermed unngå lange tider uten drift eller kast av store enkeltseksjoner.

For tegning av PCB ble programvaren EasyEDA benyttet. Dette er en online CAD som tilbys av JLCPCB som også var den valgte produsenten. Dette programmet har innebygget nett og DRC sjekker som er de samme som benyttes for sjekk før produksjon. Det er derfor enkelt å sikre at designet ikke har produksjonskritiske feil og minimere sjansen for at designet ikke blir godkjent.



Figur 4.29: Kretsskjema for Drone Arena PCB

I figur 4.29 er det totale kretsskjemaet for aktueringsstyring koblingene. Øverst til ventre er en hub som benyttes til å koble på signalkablene fra Raspberry Pien. Dette er laget som en male pin header for å kunne koble signalportene til Raspberry Pien med male to female dupont jumpwires. Det ble vurdert om andre former for koblinger med mer ferdig sammesatt kabel design skulle benyttes. Det ble midlertidig nedprioritert da tilkobling til en annen PCB for LED styring er ønsket og dette ville derfor sperret for disse portene.

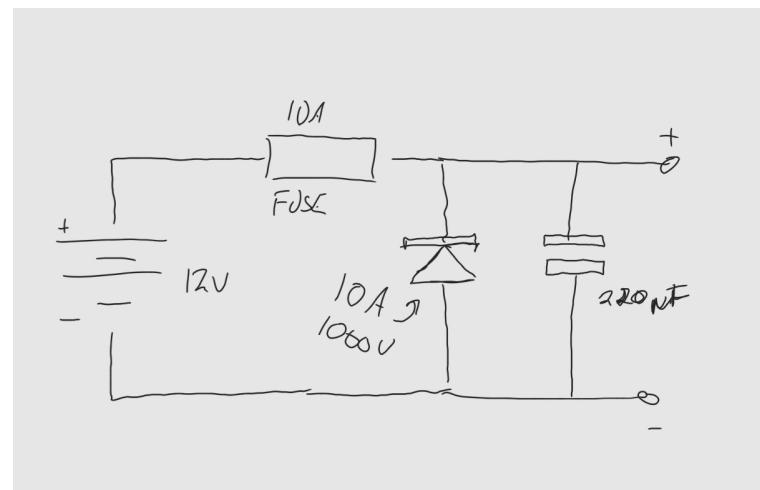
På figur 4.29 er det noen porter som ikke er koblet til. Dette er fordi de ikke blir brukt i arenaen og derfor ikke nødvendige. Alle utilkoblede pinner er sjekket for å ha pulldown resistorer koblet til for å sikre at det ikke blir stående uten tilkobling med fare for fluktuerende spenning. For TB6612FNG ble dette sjekket i databladet [70]. Dette gjelder da for BIN1, BIN2 og PWMB, BO1 og BO2 er utganger og vil ikke være i fare for denne effekten. For DRV8825 ble det ikke funnet i databladet da dette kun refererer til chip databladet. Det ble derfor målt motstand mellom FAULT og GROUND og denne ble funnet til å være tilstede. Dette betyr at det er en pulldown resistor koblet opp.

Steppermotor porten er en standard JST VH 2.54 mm, dette er derfor standard for steppermotor kabler og en male JST VH ble derfor valgt som port for stepper motoren på PCBen. Detter er likevel standard 2.54 mm avstand hull som gjør den svært kompatibelt med det meste pin headers og kan enkelt byttes ut med en pin header og dupont kabler hvis dette skulle være ønskelig. Porten på motoren er en JST VH 1.25 mm, men på grunn av kompatibilitet ble det for PCBporten valgt en 2.54 variant for dette også. Dette vil si at man må lage en kabel med disse endene eller bestille dette. 1.25 mm varianten er ikke så vanlig i utgangspunktet så det ble her vektlagt mer kompatibiliteten rundt en 2.54 mm kontakt og mulighet for pin header ovenfor å skulle legge inn en 1.25 mm variant og ha lik i begge ender av tilkoblingskabelen.

Motor og stepper-driverene støtter begge motorspenning på 12 volt og dette ble derfor valgt som den beste spenningen for å drive motoren og stepperen. Denne spenningen må leveres av en annen kilde enn Raspberry pien. En power gate ble derfor integrert i kretskort designet. For å beskytte systemet ble det også designet et tilbakekoblings-system for å beskytte driverene for byttet om polene for spenningskilden. I figur 4.30 er en skisse av selve strømbeskyttelsessystemet tegnet. En diode ligger med den åpne siden på jordplanet og med lukket mot pluss. På denne måten vil letteste vei for strømmen å gå ved feil polaritet være tilbake til

jord og gjennom sikringen. Sikringen ligger mellom pluss polen og resten av systemet. Her skal en automatsikring av typen RXEF300 installeres [72]. Denne har en triggerspenning på 6 ampere. Fordelen med denne typen sikringer er at de ved for høye strømmer vil koble ut og ved hjelp av å slippe en liten strøm gjennom vil resistansen i sikringen opprettholdes. Sikringen resettes imidlertid ved å resette strømforsyningen. Tiden før sikringen reagerer er også variabel utifra hvilken strøm som trekkes og hvor lenge denne trekkes over. Dette vil derfor gjøre det mulig for motordriverene å ha høyere strøm trekk i korte perioder hvis dette kreves.

Hver av kretsene på steppermotoren er makset til 2.0 ampere og motordriveren kan maks lever 1.6 ampere på en motor. Maks beregnet trekk som er tatt utgangspunkt i er derfor 5.6 ampere, selv om dette er lite trolig til å skje. Det ble derfor valgt en 6 ampers sikring. Dioden for polaritetsbeskyttelse må derfor være større en dette. For å unngå bestilling ble det benyttet en diode som var på lager. Tilgjengelig var et utvalg av 1N400X dioder. Denne serien har ingen verdier mellom 6 og 10 ampere og for å være sikker på at sikringen går før dioden ryker er en 10 amperes sikring nødvendig [73]. Den optimale sikringen i denne serien ville derfor antagelig vært en 10A05, altså for 10 ampere og



Figur 4.30: Strømbeskyttelseskrets for kretskortdesignet

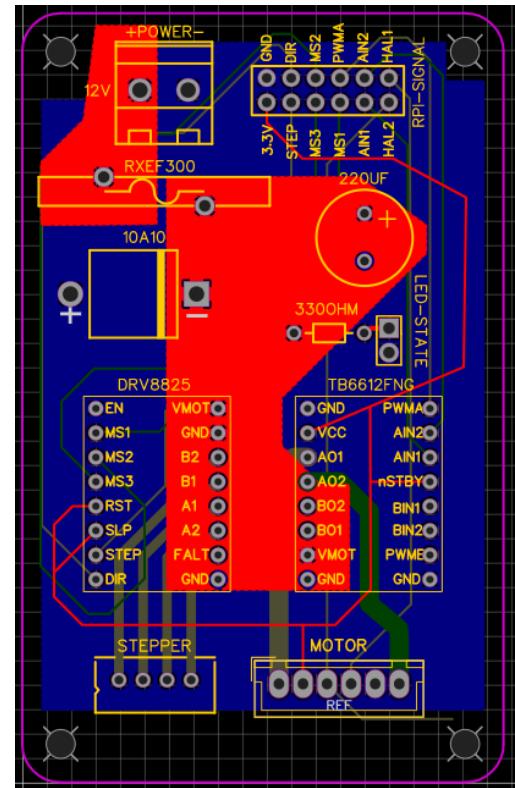
50 volt. Ratet for 10 ampere var det kun tilgjengelig 10A10, altså en 10 ampere 1000 volt. Dette er overdimensjonert, men den fysiske størrelsen er ikke veldig forskjellig da dette er primært bestemt av strømmen. Det er derfor denne (10A10) som kretskortet er designet for.

I tillegg er en kapasitator integrert i designet for å kunne jevne ut strømtopper eller ved stor indusert spenning. Tilslutt er en 300 ohms motstand med en rød ledd integrert ved hjelp av en pin header. Denne er ment til å kunne trekkes ut til frontpanelet til rackskuffen for å enkelt kunne se om strømmen på brettet stemmer. Denne vil også fungere som en drain til ground for all eventuelt indusert spenning i steppermotoren eller motorer. Dette beskytter de andre komponentene i systemet. Størrelsen på denne er litt vanskelig å estimere på grunn av at strømforbruket og tendensen til komponentene i stor grad er ukjent og kan ikke sies noe om før oppsettet er klart og analyser utføres. Det ble derfor anbefalt av veileder at en $220 \mu\text{F}$ er en ganske normal verdi i slike kretser og vil antagelig hjelpe bra her.

For å beregne linjetykkele ble en online kalkulator benyttet [39]. For signalledninger ble standardbredden på 0.254 mm beholdt da dette var det produsenten hadde som standard for produksjon. Nederste laget ble benyttet som jordingslag under hele baksiden. Det var derfor ikke nødvendig å regne på ampere for jord ledningene. For strømkilden ble det tatt til grunn 6 ampere da det er på dette tidspunktet sikringen vil slå inn så linjene burde tåle dette. Forsyningsslaget ble lagt på toppen, produsenten bruker en standard hvor øverste og nederste lag er dobbelt så standard $1 \text{ oz}/\text{ft}^2$ for topp og bunn og $0.5 \text{ oz}/\text{ft}^2$ for indre lag. Bredden på forsyningslinjen på topplaget blir derfor ved 30 graders temperaturøkning 1.83 mm. Da dette blir ganske bredt ble det valgt å også legge dette som et noe mindre fyllingsplan. Dette er markert i rødt på figur 4.31, denne ble plassert så sentrert som mulig for å lett kunne koble samtlige utganger til denne samtidig som det var mulig å legge andre poler på komponenter til et annet spenningsnivå eller liknende. Dette er blant annet gjort med LED headeren som har en 330 Ohm motstand først som fungerer som en bro fra dette strømlaget.

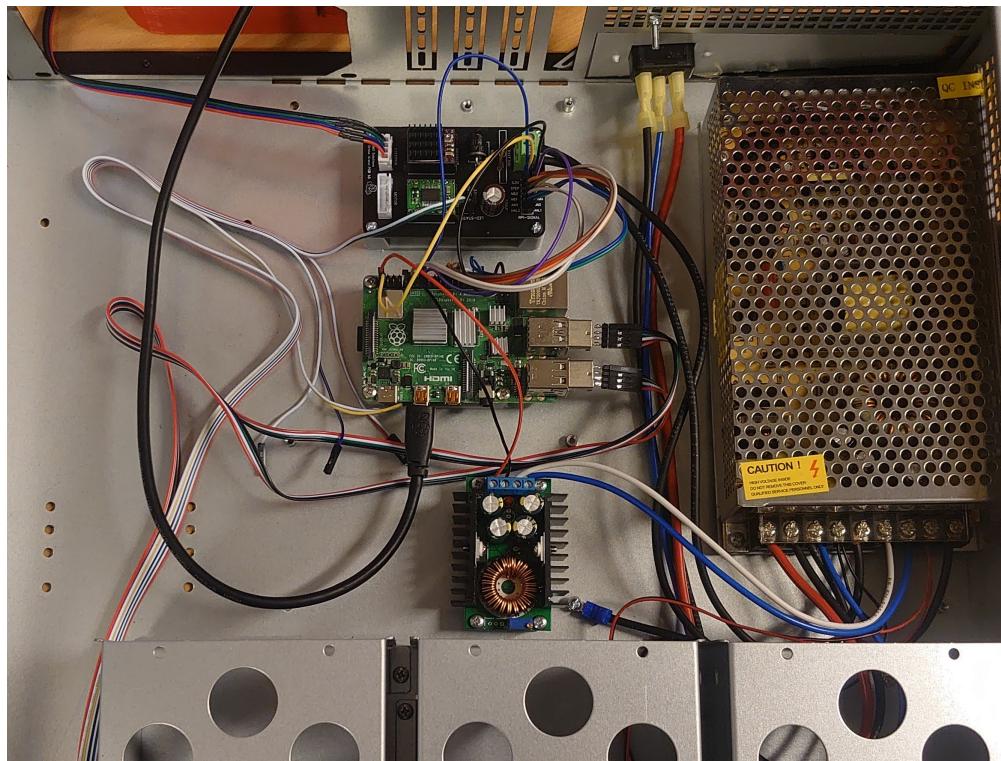
Rotasjonsmotoren vil når den kjører få en ganske jevn forsyning av strøm. Den er imidlertid ikke intendert til å kjøre svært fort og vil derfor heller ikke ha et stort behov for å være overdimensjonert i så måte. Den beregnede bredden for å ikke overgå mer en 30 graders temperaturøkning ved maks hastighet og på et indre lag med $0.5 \text{ oz}/\text{ft}^2$ er 1.53 mm. Fra veileder ble det midlertidig anbefalt å ha litt margin hvis dette lot seg gjøre og disse linjene ble derfor satt til 2 mm bredde.

Linjene til steppermotoren derimot er på dette designet noe underdimensjonert. Det er som på motoren også her antagelig ikke et stort problem da stepperen er tenkt til å kjøre i korte perioder med lite last og med lange pauser. Vekten stepperen blir utsatt for er som nevnt tidligere kun vekten av selve mekanismen pluss dronen på maks 250 gram. I tillegg er denne nedgiret ved hjelp av gjengestangen. Grunnet valget om å gå for 2.54 mm header til stepperen samt at dette er standard for driver kortet vil disse ligge relativt nærmere hverandre og for å få et godt resultat på DRC sjekken må bredden begrenses. Bredden ble beregnet ved 2 ampere til å være minimum 2.09 mm. Bredden på designet er bare 1 mm. Denne kunne vært forbedret ved blant annet og bytte jordplanet til indre plan og hatt dette som ytre for å få den ekstra tykkelsen som ytre plan har. Det ville også krevd mindre på grunn av bedre varmetap til omgivelsene. Beregnet for et ytre lag ville 0.401 vært nok og vært et minimalt problem. Disse linjene burde derfor flyttes til laget under i senere iterasjoner, men vil antagelig ikke bli et stort problem for den nåværende iterasjonen.



Figur 4.31: PCB tegning sett fra oven med top silkelag

4.9 Aktuator-skuff

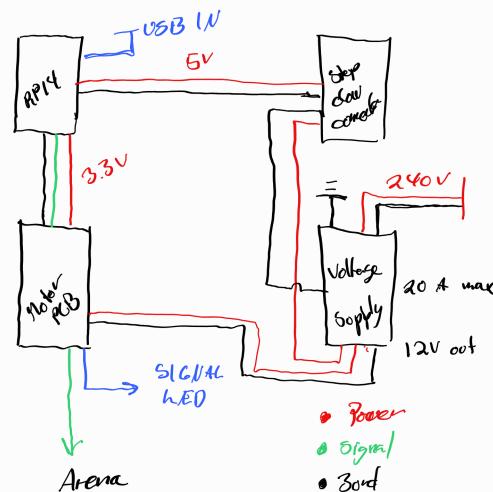


Figur 4.32: Sammenstilling av Aktuator-skuff

I den ene rack-hylla er det to 2U skuffer. Disse er tiltenkt elektronikk for styringen av arenaen. Dette henger direkte sammen med hvordan systemet er tenkt lagt opp beskrevet i kapittel 4.1 og mer i detalj angående arenaen i 4.1.1. Etter å ha bygget opp systemet ble det bestemt at kamerat skulle flyttes over på datamaskinen. I tillegg ble det bygget et motorcontroller PCB for å forenkle oppkopplingen, og på grunn av leveransen av deler er ikke QI-laderen implementert. LED styringen er også nedprioritert da dette kun er det estetiske aspektet og kun en del av "Ikke MÅ krav". Det er likevel tiltenkt at det skal inn og opplegget ligger der.

I skuffen er det lagt inn et 12 volts strøm forsyning med maksimum på 20 ampere. Tatt i betrakting at motorene i ekstreme tilfeller er begrenset til 7 ampere burde dette holde, også til å drifte andre ting, som QI ladningen senere. Strømforsyningen kan ses i bilde 4.32 og er av type S-250-12. Forsyningen er kjøpt på aliexpress og har derfor dårlig sporing, men var tilgjengelig på lageret. Denne er jordet i strømnettet og også i skuffen, slik at skulle det løsne og legge seg mot skuffen vil den kortslutte til jord og ikke gi potensielle tilskuere støt.

Raspberry Pi'en er bygget for strømforsyning på USB C porten, men har også mulighet for å bli forsynt over gpio pinnene, med et 5 volt signal på 5 volt pinnen og en på jord [74]. Dette tilbyr imidlertid ingen feilkobling beskyttelse og krever høyere aktsomhet. Siden Raspberry Pi'en krever 5 volt ble en step-down converter av typen XL4016E1 benyttet. Denne ble koblet på 12 volts forsyningen, og videre til Raspberry pien. Denne converteren har kjøleribber og vil derfor ha



Figur 4.33: Oversiktstegning av kolinger i aktuator-skuff

godt av en viss luftgjennomstrømning. Dette kan også hjelpe for de resterende komponentene og de innebygde 12 volt viftene i skuffen ble derfor koblet på forsyningen.

12 volts inngangen på PCB'en ble koblet på strømforsyningen og signalkablene koblet til Raspberry Pien som anvist på kortet. For enklere tilgang og oversikt ble en av de integrerte ledene på utsiden av skuffen koblet på PCB'en og USB kablene koblet inn i Raspberry pien. I tillegg ble en HDMI kabel trekt ut gjennom en av sideveggene for mulighet for programmering og styring for testing uten å måtte ta ut skuffen. Strømkabelen ble også trekt ut av arenaen på samme måte. For å unngå at noen av komponentene kortsluttet ved at pinnen ble lagt direkte på flatene i skuffen ble det laget tre monteringsstøtter av typen som er vist på bilde 4.34. Disse tre har egne mål for å passe de tre komponentene, tekniske tekninger i appendiks E.14, E.20 og E.18.



Figur 4.34: Monteringstøtter for aktueringskuff

4.10 Software design

4.10.1 Komunikasjonsplatform

Basert på de tidligere prosjektene er det et ønske om å benytte ROS2 som plattform for kommunikasjon mellom delsystemene da dette ble funnet til å egne seg godt i en lignende situasjon, slik som i Bacheloren ”Automatisk utskytningssystem for dronesverm” [7]. Det er også satt som et krav at Dronearenaen skal være mest mulig kompatibelt med tidligere iterasjoner av dronesverm-prosjektene 1.3, noe dette vil tilfredsstille.

4.10.2 Dybde måling

Deteksjon av dronen vil foregå ved bruk av et dybdekamera, som presentert i 4.5. Det har dog oppstått flere problemer under testingen ved bruk av dette ksystemet. Utfordringene har bestått av det å hente ut riktig modul av kameraet har vist seg krevende. Som presentert i delkapittel 2.7 består kameraet av flere sensorer, med muligheter for å uthente informasjon med hver sin hensikt. Som nevnt i delkapittel 4.5 og 5.5 er det brukt et python script for testing av hvordan kameraet håndterer det å bruke dybdefunksjonen for målinger på et relativt lite objekt som dronen. Dette scriptet er i utgangspunktet hentet fra Pysource ved Sergio Canu [75]. Det er implementert endringer for at koden skal passe best mulig til applikasjonen, herunder endring av plassering på ’Mouse event’ samt noe endring for å gjøre 5.5 Kamera testen lettere.

Enheten benyttet for dette var en hp stasjoner lab-PC, tilgjengelig gjennom Mekatronikklabben, som ble bootet med Ubuntu 20.04.4 LTS (Focal Fossa). Videre ble Pycharm lastet ned, da dette er IDE som allerede var kjent. Python scriptet for avlesning som nevnt over ble lastet inn, og de nødvendige endringene fortatt. OpenCV biblioteket er i fokus under denne testen. For en lettere bruk av RealSense kameraet har Sergio Canu utviklet en egen klasse som kan hente ut verdiene på dybdemålingene, denne var tilgjengelig gjort under samme tutorial som nevnt over. Med klassen blir selve avlesningen gjort enklere, OpenCV har også et eget bibliotek for dette, men det ble vurdert at denne tutorialen var like grei å bruke for testingen.

I koden benyttes det brukt to framer, color_frame og depth_frame, som henholdsvis representeres av RGB-modulen og IR-projectoren. Utdrag av koden blir forklart under i kode 4.1, fullstendig kode kan ses i C.1.

Listing 4.1: Python dybdemåling med D435

```
1 # Initialize the RealSense D435 Camera
2 dc = DepthCamera()
3
4 # Creating the mouse event
5 cv2.namedWindow("Color frame")
6 cv2.setMouseCallback("Color frame", mark_distance)
7
8 while True:
9     ret , depth_frame , color_frame = dc.get_frame()
10
11     # Show distance for a specific point
12     cv2.circle(color_frame , point , 3 , (0 , 0 , 255))
13     distance = depth_frame[ point [1] , point [0] ]
```

DepthCamera() initialiserer D435 kameraet, dette vil gjøre at den importerte klassen aktiveres og dermed åpner tilgangen til hele kameraet. Det er opplevd utfordringer rundt det å få D435 til å gjennomføre det koden ber om, men etter en kombinasjon med prøving og feilsøking ble resultatet tilfredsstillende.

Listing 4.2: Python dybdemåling med D435

```
1 # print(distance)
2     if distance == 0:
```

```

3         continue
4     cv2.imshow("Depth frame", depth_frame)
5     cv2.imshow("Color frame", color_frame)
6     key = cv2.waitKey(1)
7     if key == 27:
8         break

```

En if løkke er implementert for at koden ikke skal henge seg opp umiddelbart, når den syntes at detekteringen kommer for nærmee. Det er blitt erfart at om avstanden blir kort nok vil koden bare displaye 0000 mm og stoppe videre detektering, selvom avstanden egentlig er innenfor rekkevidden igjen. If løkken bryter dette ved å overkjøre med en fortsettelse, som vist i linje 2 og 3 på kode 4.2. Til slutt blir navnene på vinduene opprettet henholdsvis 'Color frame' og 'Depth frame' som kan ses på Figur 6.9 og det blir til slutt inkludert en mulighet for terminering gjennom bruk av esc-knappen.

4.10.3 ArUco

For uthenting av orientering og posisjonen skal det benyttes ArUco-markør, bakgrunnen for dette kan leses om i delkapittel 4.6. Python scriptet som benyttes for detektering av ArUco er basert på et allerede laget script av Adrian Rosebrock gjennom pyimagesearch [76]. De største forandringene er implementert for å benytte RealSense kameraet. Klassen DepthCamera generert av Sergio Canu [75] overføres hit, slik at samme metode for initialisering blir benyttet. Videre har det vært utfordringer med bibliotek for å få denne koden til å detektere ArUco-markører. Deler av koden vil bli forklart under, fullstendig script ses i C.2. Etter at alle biblioteker er implementert og RealSense D435 er initialisert, starter scriptet med en dannelse av parsere, som videre blir benyttet for behandlingen av de leste dataene.

Listing 4.3: Deteksjon av ArUco med D435

```

1 # Constructing the argument parser and parse the arguments
2 ap = argparse.ArgumentParser()
3 ap.add_argument("-t", "--type", type=str,
4                 default="DICT_4X4_50",
5                 help="type of ArUCo tag to detect")
6 args = vars(ap.parse_args())

```

Det spesifiser her at markøren som skal detekteres er av typen DICT_4X4_50, dette er en av versjonene som inneholder minimal info og er dermed lett leseelig for kameraet. Hvilken markør og størrelsen på den er forutbestemt i dette tilfellet og den er plassert på topplokket av dronen. Koden er derimot laget for å kunne detektere hvilken som helst type ArUco-markør.

Listing 4.4: Deteksjon av ArUco med D435

```

1 # Verifying that the supplied ArUco tag exists and is supported by
2 # OpenCV
3 if ARUCO_DICT.get(args["type"], None) is None:
4     print("[INFO] ArUco tag of '{}' is not supported".format(
5         args["type"]))
6     sys.exit(0)
7
8 # Loading the ArUco dictionary and grab the ArUco parameters
9 arucoDict = cv2.aruco.Dictionary_get(ARUCO_DICT[args["type"]])
10 arucoParams = cv2.aruco.DetectorParameters_create()

```

Linjene 2 - 5 i kode 4.4 går gjennom koden og ser at OpenCV støtter den, evt. de, markørene som detekteres. Videre består linjene 8 - 10 av å innhente dictionary'en som innehar ArUco og derfra henter parameterne som definerer en ArUco-markør.

Listing 4.5: Deteksjon av ArUco med D435

```

1 # Looping over the frames from the color_frame
2 while True:
3     ret , depth_frame , color_frame = dc.get_frame()
4
5     # Detecting the ArUco-markers in the color_frame
6     (corners , ids , rejected) = cv2.aruco.detectMarkers(color_frame ,
7         arucoDict , parameters=arucoParams)

```

I linjene i kode 4.5 begynner selve detekteringen, her starter letingen etter ArUco-markører gjennom color_framen på RealSense kameraet. Det betyr at color_framen, som blir displayet som et eget vindu, blir gjennomsøkt kontinuerlig etter en ArUco-markør med størrelse 4X4_50.

Listing 4.6: Deteksjon av ArUco med D435

```

1 # Verify that 'at least' one ArUco-marker was detected
2 if len(corners) > 0:
3     # Flatten the ArUco IDs list
4     ids = ids.flatten()
5
6     # Looping over the detected ArUco corners
7     for (markerCorner , markerID) in zip(corners , ids):
8
9         # Extracting the corners on the ArUco-marker (which are
10        # always returned in top-left , top-right , bottom-right , and
11        # bottom-left order)
12        corners = markerCorner.reshape((4 , 2))
13        (topLeft , topRight , bottomRight , bottomLeft) = corners
14
15        # Converting each of the (x , y)-coordinate pairs to
16        # integers
17        topRight = (int(topRight[0]) , int(topRight[1]))
18        bottomRight = (int(bottomRight[0]) , int(bottomRight[1]))
19        bottomLeft = (int(bottomLeft[0]) , int(bottomLeft[1]))
20        topLeft = (int(topLeft[0]) , int(topLeft[1]))
21
22        # Drawing the bounding box of the ArUco-marker
23        cv2.line(color_frame , topLeft , topRight , (0 , 255 , 0) , 2)
24        cv2.line(color_frame , topRight , bottomRight , (0 , 255 , 0) ,
25            2)
26        cv2.line(color_frame , bottomRight , bottomLeft , (0 , 255 , 0) ,
27            2)
28        cv2.line(color_frame , bottomLeft , topLeft , (0 , 255 , 0) , 2)
29
30        # Compute and draw the center (x , y)-coordinates of the
31        # ArUco marker
32        cX = int((topLeft[0] + bottomRight[0]) / 2.0)
33        cY = int((topLeft[1] + bottomRight[1]) / 2.0)
34        cv2.circle(color_frame , (cX , cY) , 4 , (0 , 0 , 255) , -1)
35
36        # Drawing the ArUco marker ID on the frame
37        cv2.putText(color_frame , str(markerID) ,
38            (topLeft[0] , topLeft[1] - 15) ,
39            cv2.FONT_HERSHEY_SIMPLEX ,
40            0.5 , (0 , 255 , 0) , 2)

```

```

38     # Displaying the frame from the camerafeed
39     cv2.imshow("Color frame", color_frame)
40     key = cv2.waitKey(1)

```

I kode 4.6 blir selve markeringen av detekteringen gjennomført. Seansen starter med å først se om det er en ArUco-markør tilstede. Deretter peker den ut hjørnene, slik at disse verdiene kan respektivt bli utvunnet (extracted). Disse hjørnene har alltid samme rekkefølge, og må følge denne kronologiske rekkefølgen uansett hvor de brukes. Videre tar kode 4.6 og konverterer x- og y-koordinat parene til heltall. Neste i koden er den visuelle markeringen av ArUco-markøren, her blir det tegnet en boks som omslutter hele markøren. Videre beregnes sentrum og det blir visualisert på markøren. Siste handling med markøren er å tillegge den sin egen ID, alle ArUco-koder har en unik ID 2.4 og denne vil bli visualisert på markøren. Til slutt i kode 4.6 igangsettes fremvisningen av kamerafeeden, altså color_frame hentet fra RGB-modulen. Dette vil være den viktigste delen for det visuelle, da all informasjon vil bli vist frem her og dermed får man en følelse av hva som egentlig forgår i koden.

Listing 4.7: Deteksjon av ArUco med D435

```

1 # If you press Esc, the loop will be terminated
2     if key == 27:
3         break
4
5 # Sorting every thing up
6 cv2.destroyAllWindows()

```

Siste delen av koden kan ses i kode 4.7, her skjer det samme som i dybdemålingen; altså en mulighet for terminering av programmet ved bruk av esc. Deretter brukes en innebygget kommando i OpenCV som vil rydde og avslutte alle vinduer som er åpnet.

4.10.4 Aktuator styring

En Raspberry Pi 4 er i areanaen satt opp til og styre rotasjonsmotoren og steppermotoren til midtaktuatorene. Fra systemdesignet (figur 4.1) er det bestemt at denne aktueringen kun skal skje basert på kommandoer sendt fra hoved hubben. Koden som skal kjøre midtaktuatorene trenger derfor et enkelt brukergrensesnitt som kan styre disse motorene til ønskede posisjoner. Utifra valget om at kommunikasjonen mest sannsynlig skal være ROS basert 4.10.1 var de aktuelle språkene begrenset til all hovedsak Python og C++. Raspberry Pi'en ble i utviklingsstadiet bootet i Raspien 64 bits. Dette OS'et er kvalitetssikret for Raspberry Pi og gir optimal ytelse for et desktop miljø og egner seg derfor svært godt for utvikling i desktop. Ved implementering av ROS må antagelig bootes i Ubuntu da dette er lagt opp til i tidligere arbeid [7]. Så lenge koden er i enten Python eller C++ burde dette ikke være noe problem. I Raspien er imidlertid lagt spesielt opp til Python og det er derfor dette ble valgt for å kunne teste koden raskest mulig underveis. Python har også veldig gode protokoller for å interagere med GPIO pins på Raspberry Pi'en som her benyttes for styringen. Thonny er en ferdig installert minimal IDE i Raspien for Python kode. Denne ble benyttet for utviklingen av koden.

Underveis i utviklingen ble det først laget flere små skript som ble benyttet for å teste del funksjoner ved funksjonaliteten til den endelige koden. Rotasjonsmotoren består av en nedgiret motor sammen med en encoder. Encoderen ble lest av ved hjelp av interrupts signaler og en interrupts funksjon er derfor integrert. Store deler av interruptkoden ble utviklet basert på tutorial laget av The Robotics Back-End [77]. En kode ble utviklet for testing av i hvilken grad koden fungerte. I appendix B.1 er hele koden, men et utdrag er vist i kode 4.8. For detekteringen av signalene benyttes signal biblioteket og et event for hver gang en interrupts opprettes. I dette evnetet settet koden for hvordan denne skal behandles. Her er eventet begrenset til når signalet går høyt. Dette kan settes til både synkene eller begge ettersom hvilken funksjon man er ute etter. Stigende ble her valgt for å begrense intervallene og da stigende er triggeren.

Listing 4.8: Python interrupt test

```

1 import signal

```

```

2
3 GPIO.add_event_detect(C1, GPIO.RISING, callback=button_pressed_callback
4 )  

4 GPIO.add_event_detect(C2, GPIO.RISING, callback=button_pressed_callback
4 )

```

I hovedkoden blir en subroutine opprettet for å kunne styre vinkelen til midtaktuatoren. Dette blir gjort ved å kjenne forholdet mellom antall runder for giret og hvor mange interrupts event som er per runde. Ved å gjøre om antall runder til hvor mange grader midtaktuatoren har beveget seg kan denne kjøres til en kjent vinkel. En vinkel blir gitt til programmet som en kommando og vridningen blir kalkulert. Det er derfor enkelt å kunne gi en beskjed for hoved hubben, da dette bare trengs å gir i form av en rotasjon. Rotasjonen skjer så kontinuerlig i den ønskede retningen frem til erroren er innenfor et akseptabelt område. Ved å sette et akseptabelt feil vil kjøringen lettere kunne stoppe og driftig kan lettere unngås. Det ble også oppdaget under testing at ved skifting av retning vil dette oppta såpass lang tid hvor interrupt rutinen er deaktivert at en betydelig mengde interrupts blir glemt. Det ble derfor prioritert å finne PID verdier som i så liten grad som mulig ga overshoot.

Listing 4.9: Rotasjonsfuksjon for rotasjonsmotor

```

1 def go_to_pos(self, new_ang):
2     """
3         Main control of the motor controlling the positin based on
4         feedbak from the hallsensor
5     """
6     global y_list
7     global x_list
8     error = new_ang - self.curr_ang
9
10    #if 0.00001 > error > -0.00001:
11        #    error = 0
12
13    if error > 0:
14        if self.dir == 1:
15            self.motor.stop()
16            self.dir = 0
17            GPIO.output(self.AIN1, GPIO.HIGH)
18            GPIO.output(self.AIN2, GPIO.LOW)
19        elif error < 0:
20            if self.dir == 0:
21                self.motor.stop()
22                self.dir = 1
23                GPIO.output(self.AIN1, GPIO.LOW)
24                GPIO.output(self.AIN2, GPIO.HIGH)
25        else:
26            GPIO.output(self.AIN1, GPIO.LOW)
27            GPIO.output(self.AIN2, GPIO.LOW)
28
29    new_speed, new_dt = self.PID(error, self.last_error, [20, 10, 0])
30    new_speed = abs(new_speed)
31    if new_speed > 100:
32        new_speed = 100
33    elif new_speed < 0:
34        new_speed = 0
35    self.motor.ChangeDutyCycle(new_speed)
36
37    y_list.append(self.curr_ang)

```

```

38     x_list.append(x_list[-1] + new_dt)
39
40     self.last_error = error
41     C1_now, C2_now = self.check_hall()
42
43     if error > 0:
44         self.curr_ang += 0.5 * (C1_now / self.hall_per_round) + 0.5 * (
45             C2_now / self.hall_per_round)
46     elif error < 0:
47         self.curr_ang -= 0.5 * (C1_now / self.hall_per_round) + 0.5 * (
48             C2_now / self.hall_per_round)

```

Hele koden for styring kan ses i appendix A.1. Et eget objekt blir opprettet for både rotasjonsmotoren og steppermotoren. Det er dermed enkelt å kunne benytte koden selv om motorene skulle bli byttet ut og enkoderen være et eget fysisk objekt. PIDen eksisterer som en egen funksjon i samme objekt og er derfor mulig å tune til individuelle motorer.

For steppermotoren kjører denne som et objekt som opprettes i koden. Ved testing ble det funnet at stepper kjørte stiller jo høyere grad av micro-stepping som ble benyttet. Det ble imidlertid oppdaget at ved minste setting på microsteppingen, nemlig 1/32 del, hadde motoren en tendens til å hoppe over step. Det er derfor 1/16 som er satt som standard, men dette kan lett endres om nødvendig med *set_resolution* funksjonen som tar inn hvilken step metode som skal benyttes som et string ("Full", "Half", "1/16" osv.), denne delen av koden kan ses i kode 4.10.

Listing 4.10: Stepper resolution function

```

1 def set_resolution(self, new_resolution):
2     """
3         Set a new reutution of the stepper adjusting between
4             microstepping and full
5     """
6     self.resolution_choice = new_resolution
7     for i in range(3):
8         GPIO.output(self.mode, self.resolution[self.
9                     resolution_choice])

```

For å gjøre intraksjoen enklest mulig er antallet steps som flytter aktuatoren opp i riktig høyde og ned igjen lagt inn i koden. Ved hjelp av funksjonene *go_up* og *go_down* vil det opprettede steppermotor objektet bli flyttet opp eller ned, se kode 4.11. Dette vil låsenet koden når den flytter steps. Det ville derfor vært en fordel å flytte denne delen av koden over i et eget skript som kunne vært kjørt separat. Dette burde vært helt mulig da GPIO pinsa kan brukes av flere skript på en gang. Det samme kunne vært gjort med interrupt delen av koden. Dette ville krevet en felles kommunikasjon. Denne er tiltenkt å foregå på ROS og da dette ikke er satt opp enda ble det bestemt å bygge en ferdig fungerende kode som er lett å endre og adaptiv slik at denne kan lett benyttes videre.

Listing 4.11: Stepper bevegelse fuksjoner

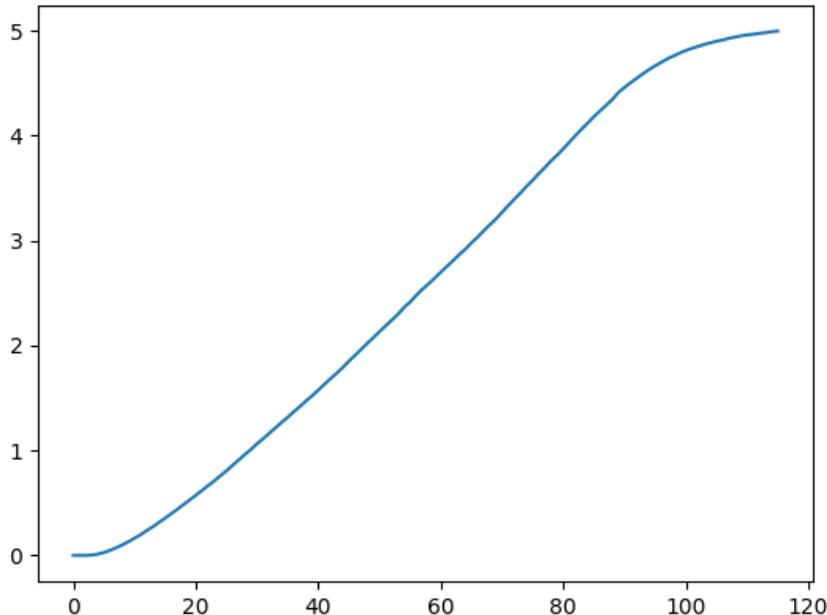
```

1 def go_up(self):
2     self.go_rounds(roundUpDown, 'Up') #Make the piston move up max
3
4 def go_down(self):
5     self.go_rounds(roundUpDown, 'Down') #Make the piston move down
6         max

```

Som nevnt benyttes en PID løkke for styringen av rotasjonsmotoren. Bevegelsene fra rotasjonsmotoren er spesielt viktig da en for rask akselerasjon av denne potensielt vil kaste av dronen eller få den til å spinne på toppen. Samtidig er en DC motor svært uberegnelig med tanke på eksakt hastighet og akselerasjon. En regn kontroller som kun kjører basert på tidsberegningen vil derfor

produsere så forskjellige resultater at denne ikke vil kunne kjøres. Encoderen er derfor absolutt nødvendig. Ved hjelp av encoderen har man mulighet til å registrere den eksakte lengden motoren har gått.



Figur 4.35: Runder per encoder per tid

en mulighet er å bruke denne lengden og stoppe eksakt etter passerte trigger events fra encoderen. Dette ville imidlertid gjøre at koden forsøkte å stoppe motoren momentant. Dette ville som nevnt tidligere ført til muligheten for at dronen spinner på toppen og eventuelt i verste fall glir av. Samtidig har motoren også en bevegelsesenergi, selv om denne er lite og giret stort vil dette ha noe å si. Siden man kun har valget om å stoppe å kjøre motoren og ingen mulighet til å aktivt stoppe den, som man har i en stepper motor, vil denne bevegelsesenergien kunne dra videre og føre til en overshoot. PID styringens sakte opp og ned akselerasjon vil derfor være helt essensielt. I figur 4.35 kan man se denne akselerasjonen i praksis.

Listing 4.12: PID funksjon implementert i DC motor klassen

```

1 def PID( self , in_val , last_val , k_list):
2     """
3         This function return the values needed
4             for calcualting th PID tuning mekanism.
5             -Økter
6             """
7     dt = time.time() - self.last_time
8     p_val = k_list[0] * in_val
9     i_mel = self.i_last + ((last_val + in_val)/2) * dt
10    i_val = i_mel * k_list[1]
11    d_val = ((in_val - last_val)/dt) * k_list[2]
12
13    val = p_val + i_val + d_val
14    self.i_last = i_mel
15    print(dt)
16    self.last_time = time.time()
17    return val ,

```

I kode 4.12 er PID funksjonene implementert i *pwm_motor* klassen vist. Funksjonen regner selv på tidsintervallet siden den selv ble benyttet for å ha best mulig forutsetninger som å kunne beregne mest presise verdier. Alle kalkulasjoner som trengs å tas vare på for neste iterasjon blir lagret som en objekt variabel.

5 Eksperimenter

5.1 Glide-test

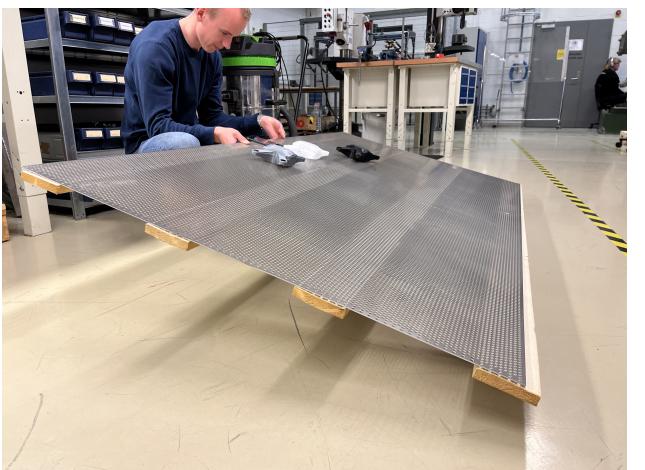
Det ble utført en glidetest basert på det endelige drone skrog iterasjonen printet ut i forskjellige plasttyper som har vært ansett som aktuelle under design prosessen. Disse skrogene ble vektet opp til maks vekten for dronen for å best mulig simulere dronene som skal brukes i arenaen. Skrogene ble vektet med muttere som ble limt med hotglue til skroget for best mulig vektfordeling. Plast filamentene som ble valgt var PLA, PLA+ og PP plastikk. Alle skrogene ble printet ut på samme type printer, en Artillery Sidewinder X2 med samme tykkelse, rotasjon og support mengde for å ha så sammenliknbare resultater som mulig.

Disse test eksemplarene ble så en etter en puttet på plater med aktuelle materialer og forskjellige overflatebehandlinger. Platene ble lagt flat på bakken og en mobil med en vater app (bubble level på android) ble festet til platen. De oppveide skrogene ble så lagt oppå platen på samme område hver gang og platen sakte hevet i en akse mens den andre ble holdt stabil.

Graden hvor skroget begynte å gli nedover platen ble notert. For størst validitet ble forsøket gjentatt tre ganger for hvert skrog på hver overflate. Et gjennomsnitt av de tre noterte vinklene ble regnet ut og benyttet for sammenlikning. Det ble testet så mange variasjoner av plater materialer som mulig, med vekt på å få testet alle materialer som var blitt foreslått eller vurdert.



(a) Glide-test utført på aluminiumsplate fra arenaen.



(b) Glide-test utført på perforert aluminium.

Figur 5.1: Glide-tester ble utført på flere materialer, to av de fremvist her.

5.2 Falltest

Fall testen av drone ble gjennomført inni arenaen for å kunne se verdiene på kreftene dronen belastes med om den krasjet inni arenaen, se figur 5.2. Det endelige drone designet ble brukt sammen med 1600mah batteri, kakute f4 og en mottaker. Slik at dronen kunne flys under testene, dermed ble den innsamlede dataen forholdsvis lik som når dronen flyr på AI. For å loggføre dataen som samles inn ble minne som er på flight controlleren brukt sammen med betaflight. I betaflight finnes det en blackbox tab hvor man enkelt kan aktivere black box, og dataen blir da loggført i samtid.

For å samle dataen ble dronen flydd og krasjet så hardt som mulig inni arenaen for å få målinger på de mest ekstreme tilfellene. Deretter kunne flight controlleren kobles til PCen og hente ut dataen som en .BBL fil. Denne filen ble lest med programmet Betaflight Blackbox Explorer. Data som akselerasjon, vinkelhastighet, motor pådrag og stikkinnput vises da på grafer og figurer. For å

kunne hente ut den ønskelige informasjonen ble filen konvertert til en .csv fil for videre behandling i matlab.



Figur 5.2: Falltest med maulet flyging i arenaen

5.3 Falltest Solidworks

For å samle data på hvordan dronen ville holde seg inni arenaen ble det foretatt en FEM analyse i Solidworks. Her ble det valgt å ta en droptest, siden dronen vil falle mellom hver flyvning i arenaen. Deretter ble materiale lagt til på alle delene i assembliet, samtidig at vekten på assembliet ble kontrollert til å være 240 gram. Delen ble meshet og simuleringen ble kjørt, men den ville ikke fulføre. Dette var grunnet problemer med slissen mellom topp- og bunn skroget. Derfor ble det laget ett nytt assembly hvor topp- og bunnskroget var festet sammen til en del istedenfor to separate.

Etter dette var fikset ble det lagt til kontaktpunkt mellom innmaten og skroget. Disse kontaktene ble satt til no penetrationsiden innmaten ikke er limt fast. Meshet ble laget og oppsetsparameterne ble satt. Her ble høyden satt til å være 1600 mm, da dette er litt over maks høyde dronen kan falle fra inni arenaen. I tillegg ble gravitasjonen satt slik at dronen ville treffe bakken med undersiden først. Dempingen i bakken ble satt til null slik at dronen blir testet så hardt som mulig.

5.4 Flygetest

En flytest ble gjennomført for å kunne teste flyge tid, flyge egenskaper og holdbarhet. Fly testen ble gjennomført ute på gress, med tre fulladede 2s batterier på 1600Mah, fem tilsvarende batterier som var ladet opp til åtte volt og ett 3 cellers batteri på 1000Mah som var fulladet. Dronen ble flydd i tre forskjellige moduser, angle mode, horizon mode og acro mode.

Til å begynne med ble dronen kjørt rolig langs bakken for å utforske manøvrerbarheten. Deretter ble ble thrusten testet i angle mode ved å gi full throttle oppover og ved å legge dronen i maks vinkel i angle mode med full throttle. Sistnevnte vil vise om drone har nok kraft til å opprettholde høyden ved maks vinkel fremover i angle mode. Senere ble det testet om dronen kunne flippe, altså rotere 360 grader i pitch eller roll uten å treffe bakken. Det er viktig at dronen kan holde seg i ro på ett lite område, for å teste dette ble det forsøkt å lande oppå en fuglekasse som var omtrentlig samme størrelse som dronen. Til slutt ble dronen kjørt så hardt som mulig for å teste holdbarheten. Det ble krasjet inn i trær både med og uten propguard. I tillegg ble dronen gjentatte ganger droppet fra ca fem meters høyde hver gang den traff ett tre.

Flyve egenskapen ble også testet inni arenaen for å kunne se hvor mye arenaen påvirker dronens fly egenskaper. Siden det ikke er større diameter enn 1.2 meter på arenaen er det begrenset med manøvrere man kan teste. Det ble testet å hovre i midten av arenaen i forskjellige høyder. Deretter ble det testet å fly langs veggene og nærmere vegger og gulv for å sjekke turbulensene disse skaper for dronen.



(a) Flyging av dronen i skogsmiljø

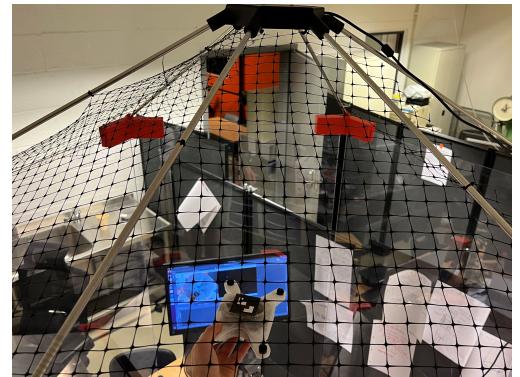


(b) Flyging av dronen i åpent miljø

Figur 5.3: Flygetest i forskjellige miljøer

5.5 Kamera test

Det ble gjennomført en test av hvordan dybdefunksjonen på Intel RealSense D435 Depth kameraet håndterte det å detektere dronen. Det ble utviklet et python script, som er nærmere forklart i delkapittel 4.10.2 for å teste hvordan dybden ble avlest på droneskroget. Testen ble gjennomført ved at dronen ble holdt ved forskjellige lokasjoner inne i arenaen samtidig som en holdt musepekeren over den gitte plassen på PC-skjermen, scriptet er laget slik at dybden blir hentet ut der musepekeren er. Dette ble gjort flere ganger med dronen plassert ved forskjellige lokasjoner inne i arenaen, samt musepekeren ble holdt over flere steder på droneskroget.



Figur 5.4: Oppsett for kameratest

5.6 Åpen dag sammenstilling

26. April 2022 ble det avholdt åpen dag ved UiA Grimstad. Det var da et ønske om å vise fram en tidlig prototype av arenaen. Det ble derfor laget en sammenstilling av arenaen hvor kun det estetiske og den fysiske avgrensingen ble satt i fokus. Oppdraget ble gitt 22. april så fristen ble derfor relativt kort, og ikke uttestedede eller beregnede løsninger ble derfor anvendt.

Bunnplatene til arenaen ble kuttet ut før dette og var klar denne uken, disse ble kuttet ut med plasmakutter av 4 mm stålplater av billigste sort, se D.1D.2. Disse ble etterbehandlet ved å slipe ned alle skarpe kanter på båndsliperen fra plasmakuttingen. Hullene for midtseksjonen ble markert med plasmaen fra den genererte koden. Hullene er for små til at plasmakutteren vil gi et godt resultat. Hullene ble derfor boret opp i etterkant med drill for hånd.

Disse ble så montert på de 3D-printede hjørnene som var produsert endel tid i forveien. Dette krevde endel finsliping for å line opp hullene perfekt. Dette ble gjort ved å sjekke hullene mot det riktige hjørnet og slipe litt og litt ned på båndsliperen. På denne måten ble det sørget for at alle plantene sitter godt innatt hverandre og riktig i hjørnene. Midtseksjonen ble så tredd på i midten fra undersiden slik at alle tre bunnplatene ble skrudd i riktig posisjon. Ved hjelp av punktsveis ble

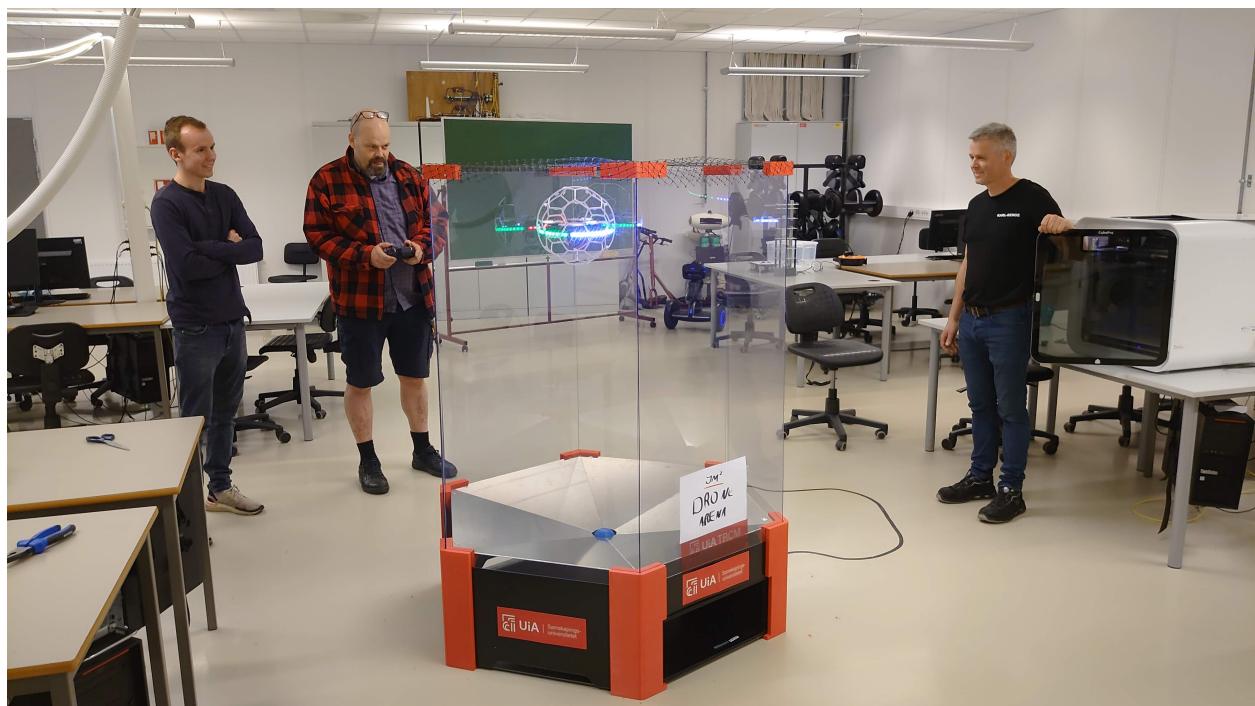
så alle platen festet med hverandre med 8-9 punkt per skjøt. Dette for å skape en solid bæreplate for trygg frakt.

Midtaktuatorene var i stor grad ferdig bygget på dette tidspunktet. Nållageret som midlertidig skal ligge mellom midtaktuatorere og midtseksjonen og muliggjøre rotasjonen av midtaktuatorene mulig har fått leveringsproblemer og var ikke mottatt for denne sammenstillingen (og heller ikke ved innlevering av rapport 27. mai). Det ble derfor laget en 3D-printet plastisk ring i PLA med samme mål som det intenderte lageret. Dette bergenser betraktelig rotasjons muligheten til midtseksjonen, men kjøring av denne var heller ikke viktig før denne sammenstillingen.

Rackkassene ble så kuttet i bakkant for å tilpasse med riktig dybde, disse ble kun plassert i riktig dybde i sine tilmalte slisser og ikke sikret på noen annen måte. Veggene ble kuttet ut med laser kutter i svart acrylplater og slidet inn i hjørnene, henholdsvis fire store og to små for racksidene. Bunnseksjonen ble så plassert på et tilmålt sted, før glideplatene ble plassert. Disse ble kappet ut ved hjelp av plasma kutteren i aluminium. På dette tidspunktet i prosessen var det ikke funnet et optimalt materiale for å sikre god glid, og valget om aluminiumsplater ble derfor tatt på grunn av den umiddelbare tilgjengeligheten og den ikke uoverkommelige prisen.

Plastglassplatene ble plassert i sporene og versjon 2 topphjørner plassert på. Etter observasjon av utglidning av platene ble det boret opp hull og tredd en M3 skrue gjennom hvert av hjørnene i hver plate. Deretter ble en hønsenetting tredd på over og stiftet fast på utsiden av topphjørnene. Tilslutt ble nettet kuttet i fasong for å passe formen best mulig. For åpning og lukking av buret ved bruk av dronen ble to topphjørnene løsnet og en side tatt av.

En drone av typen Gaupner Sweeper ble kjøpt inn for fremvisning og testing av buret, se figur 5.5. Denne kommer med en ferdig ball rundt for optimal beskyttelse av dronen. Den benytter seg av et 4S batteri på 1600 mah. Denne ble sponset av UiA Top Research Senter, derfor klistermerkene på arenaen. Det ble fløyet henholdsvis 7-8 runder inne i buret av både relativt erfarne drone piloter og helt nybegynnere. Det ble gjennomført noen harde runder med mye krasjing for testing av hvor godt den holdt sammen, rolige runder hvor egenskaper som ground vash og flygbart areal ble spesielt testet og nybegynner runder for publikum for observasjon av hvilken visuell effekt utseende hadde.



Figur 5.5: Åpen dag sammenstilling, Dekan Michael Rygaard Hansen flyr dronen

5.7 QI-test

Etter at QI coilen ble mottatt ble det gjennomført en kort test for å sjekke om konseptet lot seg gjøre. Delen ble mottatt for sent for en fullverdig implementasjon. Testen ble derfor gjennomført ved å redesigne en versjon av top-hatten med en slisse for QI coilen i toppen om mulighet for å legge elektronikken inne i toppen. Coilen drives av en strømforsyning på 12 volt og med mulighet for 5 ampere.

Mottaker coilen ble plassert inne i skroget og balansepluggen plugget i en USB Lipo lader av typen Dynamite DYNC1062 som via balanseporten sørger for lik lading av cellene [78]. En USB B micro port ble modifisert til å passe på QI mottakeren og koblet i laderen. Batteri pluggen ble koblet i Kalkute F4 flight controlleren som ble benyttet for fly testen [51].

Strømmen på mottaker siden ble målt sammen med spenningen og betaflight ble koblet opp mot FC'en for å observere batterispenningen via FC'en.



Figur 5.6: QI-lading test oppsett

6 Resultater

6.1 Glidetest

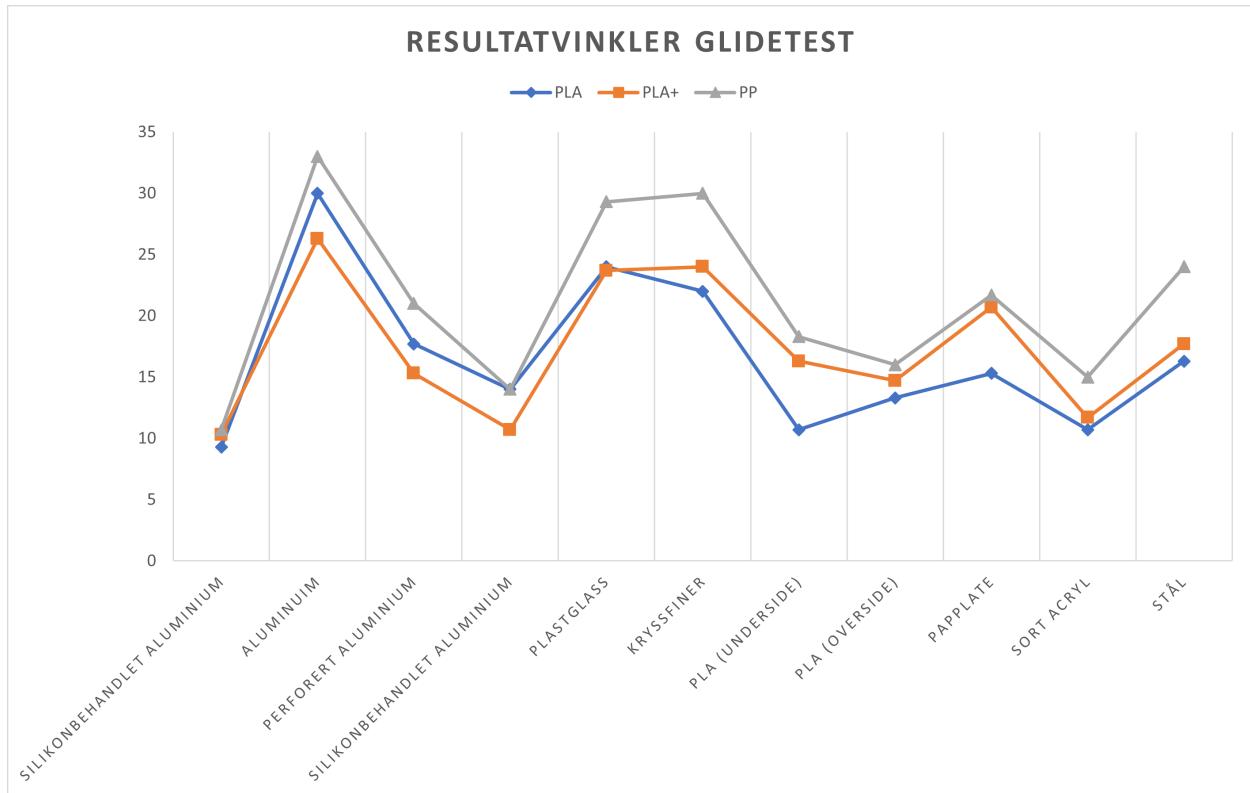
Tabell 6.1: Skrogets egenvekt og etter opplasting for test

Materiale	PLA	PLA+	PP
Skrog vekt (g)	65.8	64.6	43.7
Test vekt (g)	250.5	250.3	249.8

I tabell 6.1 er vekten til de representative testeksemplene under glidetesten vist. I tabell 6.2 er resultativinklene for når testobjektene i glidetesten 5.1 vist. Første vinkel er den gjennomsnittlige resulatvinkelen og i parentes er hele måle serien på tre forsøk vist. I figur 6.1 er vinklene framstilt som et linjediagram med gjennomsnittsvinklene. PLA plast har orange strek, PLA blå strek og PP plast grå strek.

Tabell 6.2: Resulatvinkler: gjennomsnittsvinkel (måleserie)

Plate materiale	PLA	PLA+	PP
Silikonbehandlet aluminium	9.3 (10 9 9)	10.3 (11 8 12)	10.7 (12 10 12)
Aluminium -	30 (28 33 29)	26.3 (27 25 27)	33 (33 33 33)
Perforeert aluminium	17.7 (13 21 19)	15.3 (13 15 18)	21 (15 21 27)
Silikonbehandlet perforert aluminium	14 (14 14 14)	10.7 (10 11 11)	14 (12 15 15)
Plastglass -	24 (24 24 24)	23.7 (24 24 23)	29.3 (28 32 28)
Kryssfiner -	22 (21 21 24)	24 (27 25 20)	30 (30 30 30)
PLA underside	10.7 (10 11 11)	16.3 (17 18 14)	18.3 (19 17 19)
PLA toppside	13.3 (14 14 12)	14.7 (17 14 13)	16 (17 15 16)
Pappplate -	15.3 (16 16 14)	20.7 (25 20 17)	21.7 (24 21 20)
Sort acryl	10.7 (10 11 11)	11.7 (12 12 11)	15 (15 15 15)
Stål -	16.3 (16 18 15)	17.7 (18 20 15)	24 (22 26 24)



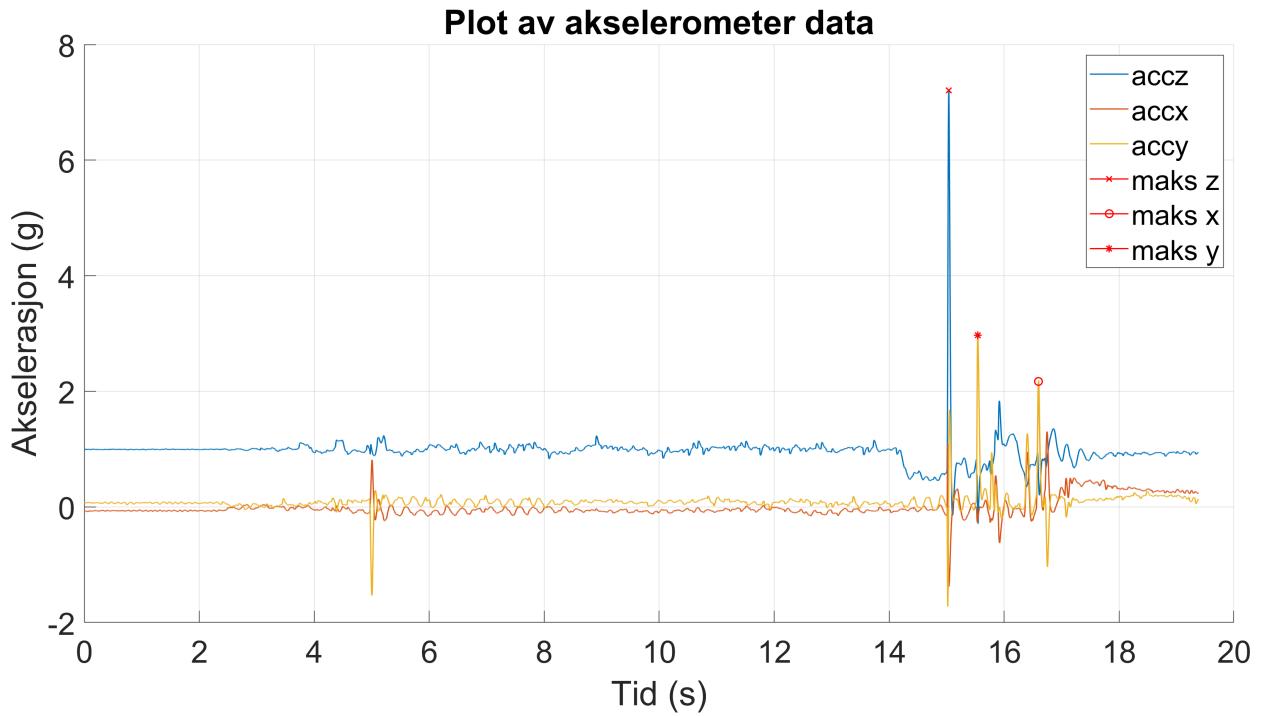
Figur 6.1: Resultatvinkler framstilt som et linjediagram

6.2 Falltest

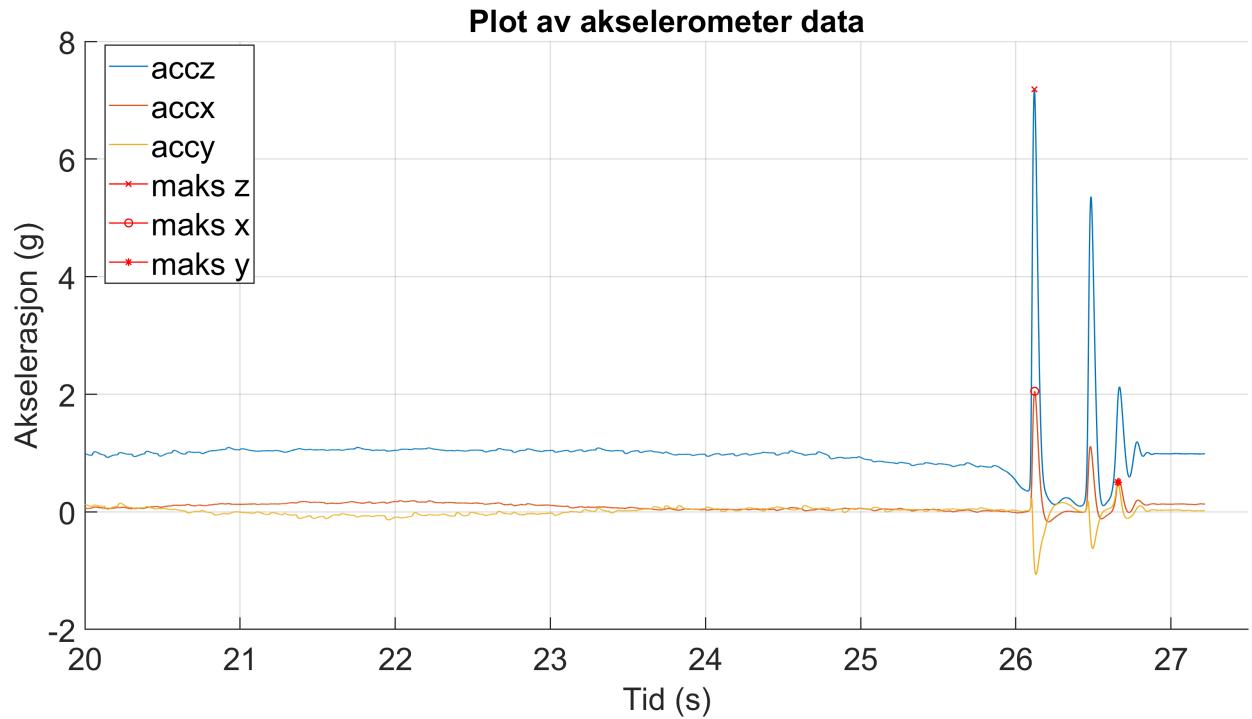
I tabell 6.3 vises den høyeste og laveste akselerasjonene i de forskjellige aksene. Dette vises for både faltesten inne og ute. Videre kan man se akselrasjonene i ett tidsrom føre fallet og frem til fallet er gjennomført i figur 6.2 6.3. Her er også den høyeste akselrasjonene markert med kryss, stjerne og sirkel.

Tabell 6.3: Maks og min akselrasjoner(g) i alle akser under falltest

Akse	Z	X	Y	Totalt
Maks i arena	7,21	2,17	2,97	8,09
Minimum arena	-0,29	-1,38	-1,72	2,22
Maks ute på grass	7,19	2,05	0,50	7,49
Minimum ute på grass	0,10	-0,40	-1,07	1,15



Figur 6.2: Falltest inni arena



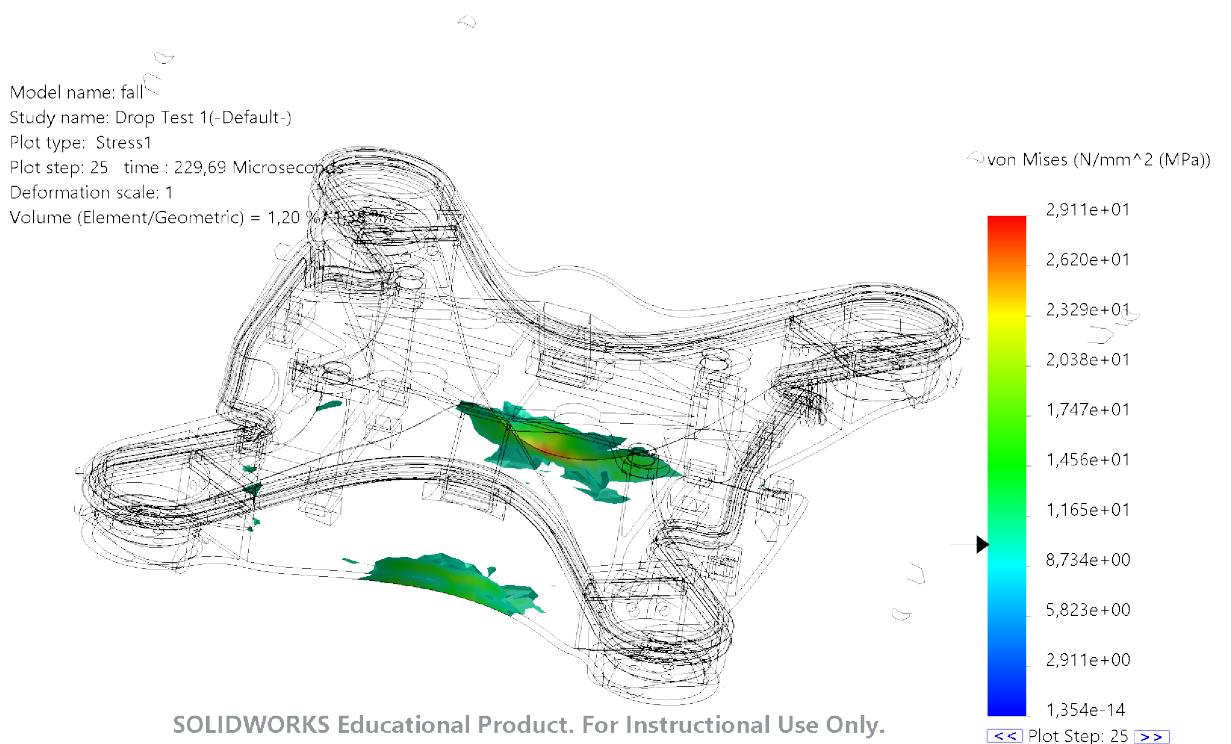
Figur 6.3: Falltest ute på grass

6.3 Falltest Solidworks

Resultatene fra falltesten i Solidworks kan sees i figur 6.4, 6.5 og 6.6. Figur 6.4 viser von Mises av hele dronen, mens figur 6.5 kun viser områder hvor von Mises er over 10 MPa. Den siste figuren 6.6 viser hvor mye flatene på dronen beveger seg i [mm].

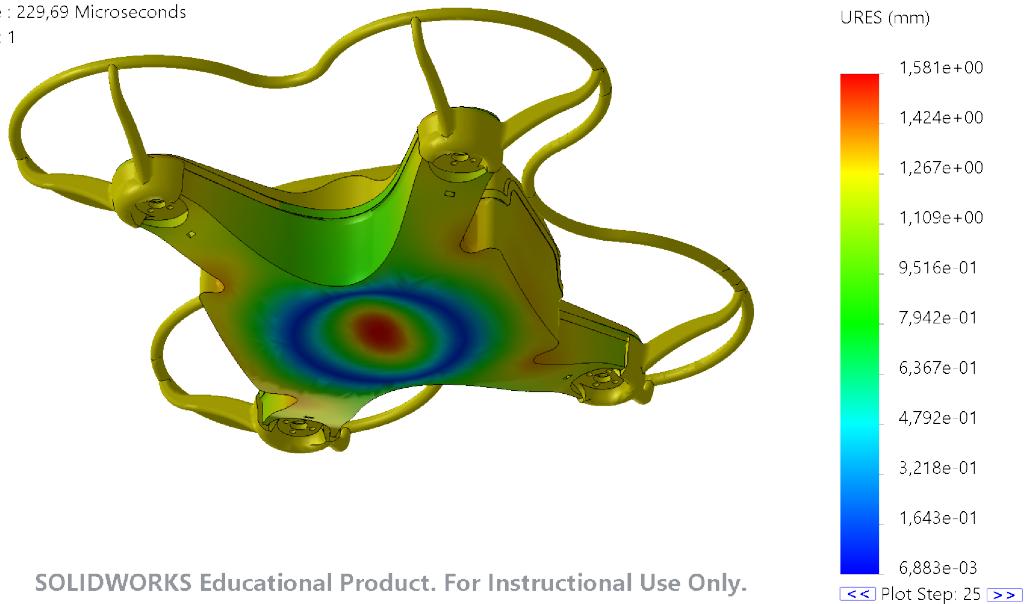


Figur 6.4: Von Mises ved fall fra 1600 mm



Figur 6.5: Områder hvor von Mises er over 10MPa

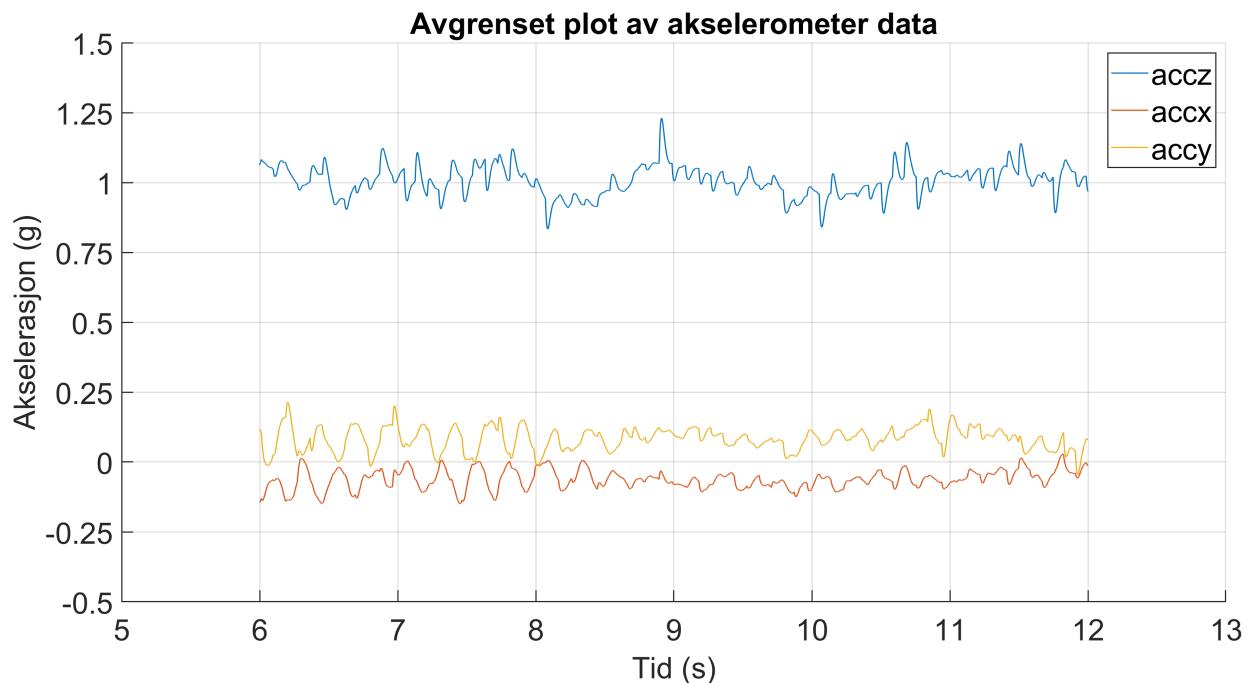
Model name: fall
Study name: Drop Test 1(Default)
Plot type: Displacement1
Plot step: 25 time : 229,69 Microseconds
Deformation scale: 1



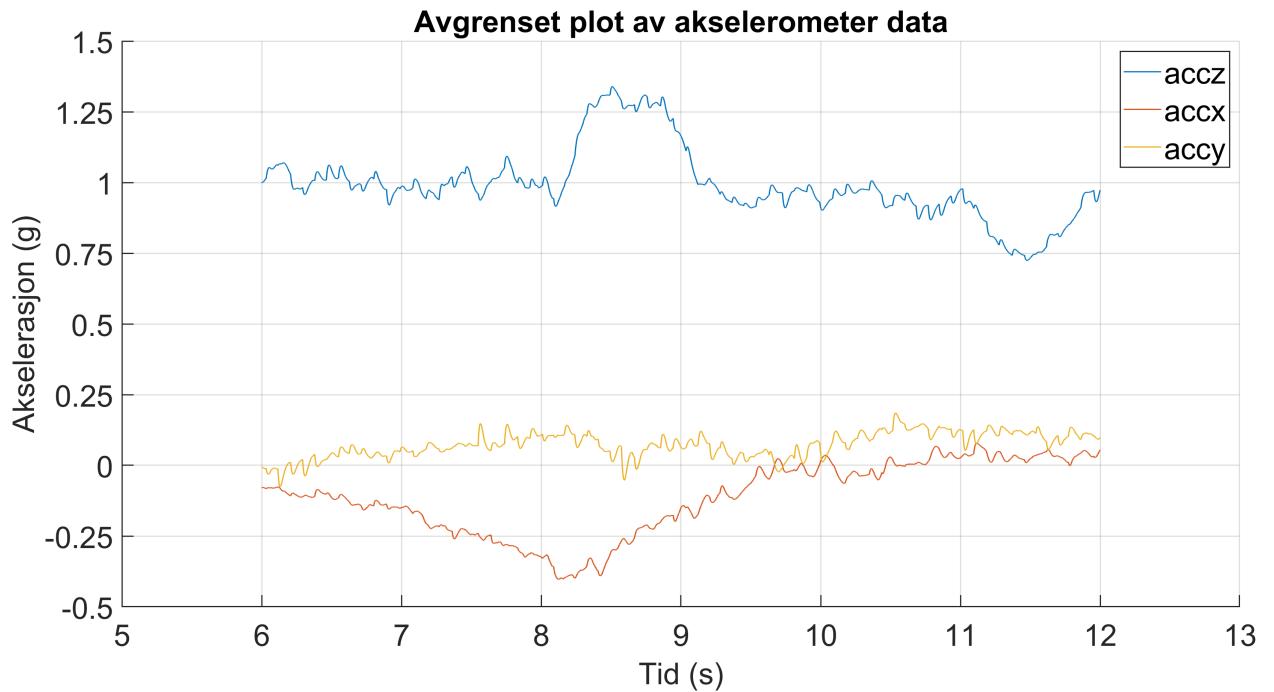
Figur 6.6: Bevegelse i skroget [mm]

6.4 Fly test

Resultatene fra flytesten er plottet i figur 6.7 og 6.8. Dette er over ett kort tidsrom hvor dronen flyges stille i luften og alle aksene er plottet.



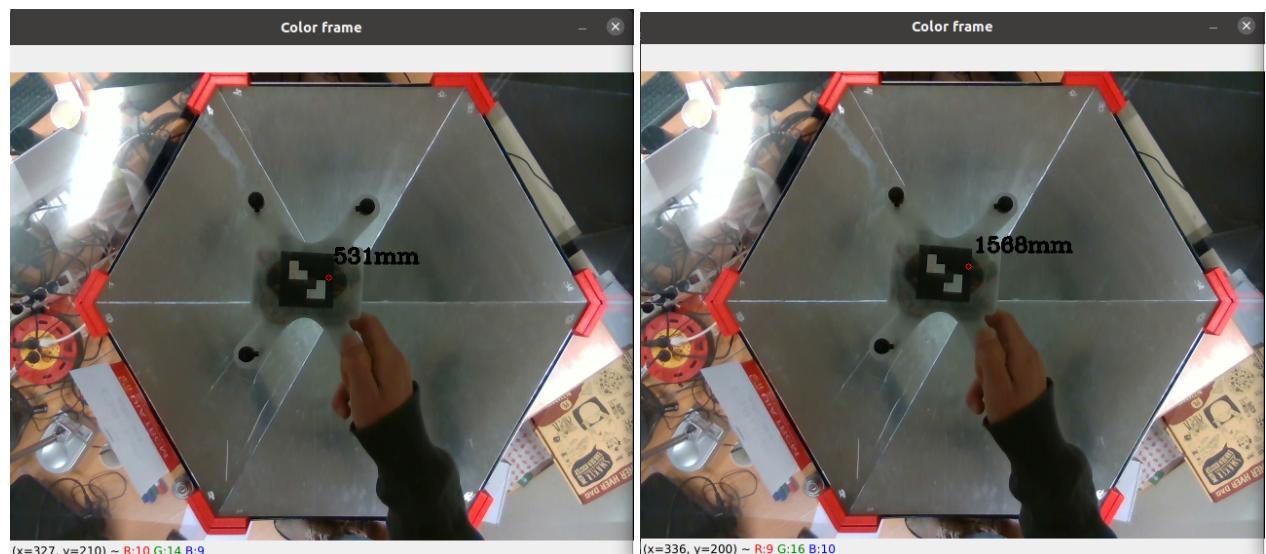
Figur 6.7: Utklipp fra flygning i arena føre kolisjon



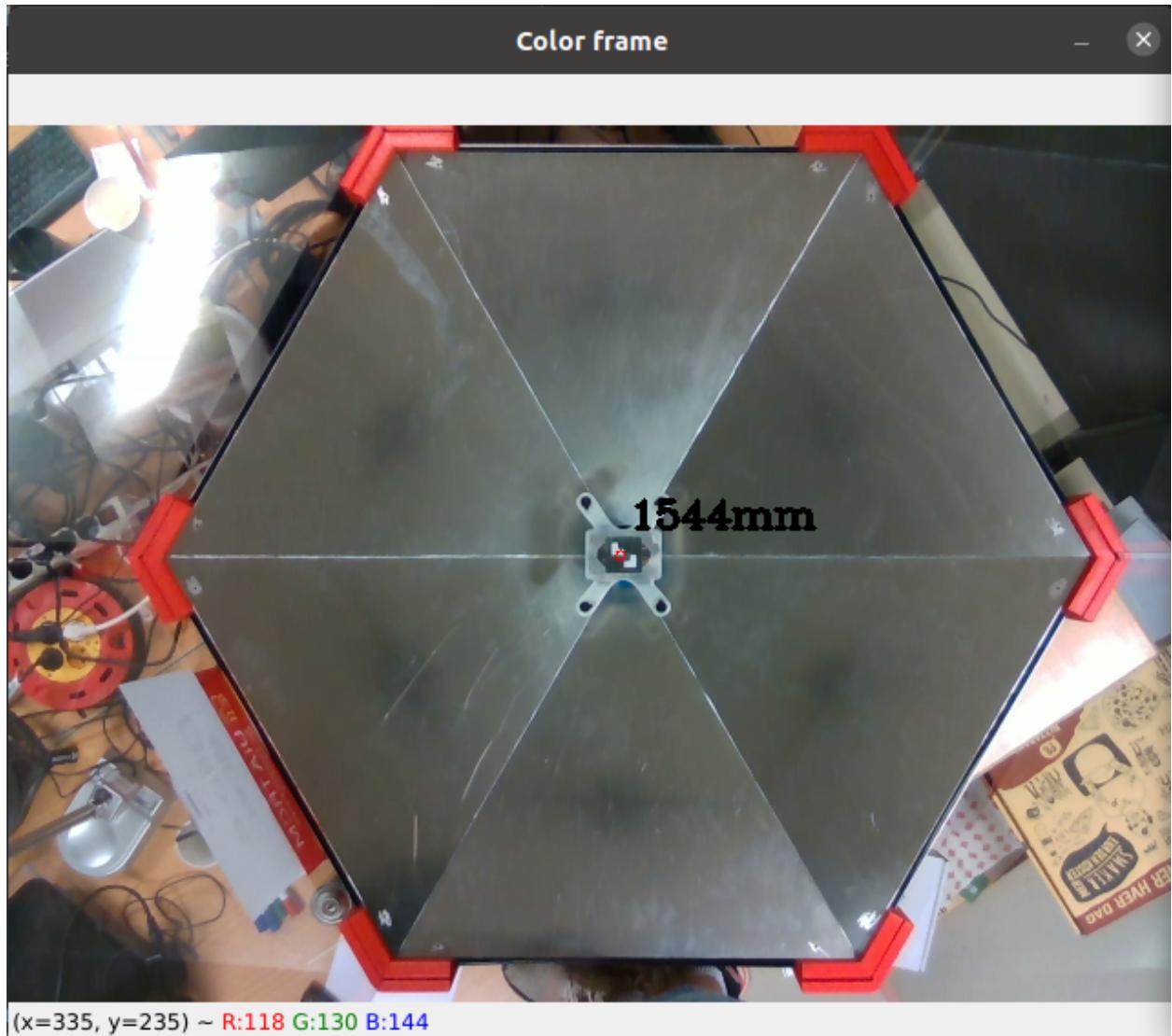
Figur 6.8: Utklipp fra flygning ute

6.5 Kamera test

Fremvist er kamera testen, i Figur 6.9a er korrekt avlesning av avstanden til dronen. I Figur 6.9b er ikke korrekt avlesning fremvist, her er det avstanden til bakgrunnen som er oppgitt. I Figur 6.10 er avstanden målt ned til dronen som står i utgangsposisjonen for å kunne lette eller lade.



Figur 6.9: Dybdemåling med RealSense D435



Figur 6.10: Dybdemåling med drone stasjonert for lading eller letting

6.6 QI-test

Tabell 6.4: Avleste verdier under QI-test

Strømforsyning spenning	12 volt
Strømforsyning maks strøm	5 ampere
QI-mottaker spenning	5 volt
QI-mottaker ladestrøm	1.5 ampere

Resultater målt med multimeter under testen av QI-systemet vises i tabell 5.6. Disse har relativt lite sikkerhet og er kun enkel målepunkter på grunn av leveringsproblemer.

7 Diskusjon

7.1 Konsept

De tre konseptene som har reellt blitt vurdert, scorer ganske likt på de fleste kategoriene. Gjennom poengutdelingen er det konsept 1 og konsept 3 som har det største potensialet, derimot er dette en tiltenkt applikasjon og Fremvisning i kombinasjon med porterbarhet er noe av det som er utslagsgivende i favør Glassbur konseptet. Alle kategoriene representerer disipliner den endelige Dronearenaen ideelt burde inneha, samtidig som det er mulig vektlegge de gode egenskapene til de andre konseptene. Denne metoden gjør det enklere å lande på et valg når det kommer til hvilket konsept som skal være grunnmuren. Det er derimot slik at det er styrker hos de andre konseptene også, det viser seg i flere av de avgjørende kategoriene at konsept 1 scorer vel så bra som konsept 2 og konsept 3. Det viser seg at Dronearenaen også trenger styrker fra de andre to konseptene, det kan i henhold til Tabell 3.1 sees hvor de to andre konseptene har styrkene sine.

Endelige løsninger rundt om på Dronearenaen har fremkommet etter hvert og det er blitt tatt i bruk løsninger utover det som initialt var inkludert i konseptene. Derimot har både blikkfang og visuell utforming overgått det som først var forespeilt, og de kategoriene som konsept 1 scoret høyt på har blitt bekreftet på endelig utforming.

7.2 Glideplater og friksjon

Glideplatene og deres funksjon viste seg for å skulle bli en sentral del av arenaens funksjon. Ut fra det valgte konseptet er disse platenes evne til å transporterer dronen tilbake til midten helt avgjørende. Prosessen ble ytterligere kompleks av arbeidet på droneskroget da dette viste seg å bli mer omfattende enn først antatt. Materialet til droneskroget er en av de største faktorene i spørsmålet om hva som vil skli nedover platene. Dette ble forandret flere ganger og det ble også derfor designet nye skrogstrukturer som ga riktig styrke og funksjon til de nye materialene. Siden utseende til skroget har endel å si for friksjonen gjorde dette samlet at prosessen ble satt noe på vent.

I seksjon 5.1 ble en test utført med det endelige skrogdesignet printet ut i 3 aktuelle plastikktyper. PLA er grunnplasten og er som standard brukes samt den som var prototypet i før. PLA+ ble oppdaget i løpet av prosessen og grunnet sin høye styrke representerer en PLA som antagelig vil tåle betydelig mer fall. PP plastikken var den plasten som var mest aktuell på grunn av sin høye duktilitet og styrke. Skrogdesignet er også svært påvirket av dette med gode støttestrukturer rundt former som kan være utsatt for vridning og bøyning. Sammenlikner man egenvekten til skrogene, sett i tabell 6.1 ser man også at PP skroget kun har halve egenvekten noe som dessuten gir mer rom for vekt til andre komponenter.

Ut ifra designet ble en vinkel på glideplatene satt til 12.76° , vinkler under dette er derfor det som vil gi et resultat som kan benyttes videre. Sånn jevnt over kan det ses at PLA+ er den plastikk typen som glir best, tett fulgt av PLA. Det var også dette som ble lett testet i begynnelsen når konseptet med glideplater ble testet. Det var i utgangspunktet PLA mot PLA underside som fra tabell 6.2 har en vinkel på 10.7. Sammenlignes det med resultatene til PP ser man at disse har jevnt over en betydelig høyere vinkel og representerer derfor en utfordring for konseptet.

For ”åpen dag sammenstillingen” 5.6, ble det produsert plater i Aluminium da dette var det som lot seg gjøre innenfor tidsrammen. Etter tips ble det senere testet å smøre platene inn med et silikon lag. Dette viste seg å love svært godt og skroget gled fint på hellingen i arenaen. Fra testen ser man at vinkelen var 10.7 samme som PLA mot underside PLA. Et av de store bekymringene for dette er holdbarheten opp mot kravet om vedlikeholdsri operasjon i opp til en måned. De første testene med dette ble gjort ved å smøre kun et tynt lag inn i platene før så endel testing og vising. Denne platen ble brukt fra og til i fire til fem uker uten noe synlige dårligere friksjon. Selv om dette burde

vært testet for en lengre tidsperiode lover disse resultatene godt med tanke på slitasje.

Av Aerodynamiske grunner diskutert i 7.3 forelå det et ønske om å bytte aluminiums platene ut med perforerte plater. Under testingen ble det funnet at disse hadde en noe dårligere og ikke tilfredsstillende glide vinkel. Det ble imidlertid observert at disse hadde en overflatebehandling med noe olje, antagelig rester etter smøring for perforering. Denne overflatebehandlingen så ut til å gi mindre vinkel ubehandlet, men høyere med forsøkt silikon behandlet. Denne overflatebehandlingen ser ut til å minske silikon virkningen og antas og kunne forbedres ved å fjerne oljelaget før silikon sprayen påføres.

Det ble også underveis i prosessen testet andre løsninger som teflonplater og teip. I undersøkelsene ble en baderomsforretning besøkt for å spørre om de hadde noen gode forslag og en teflonplate ble testet. Dette ga ikke lovende resultater for PP plast og det ble derfor ikke forsøkt videre i sammenheng med tanke på pris. Noe testing med teflonteip mot aluminium ble også utført, men på grunn av høy slitasje og potensielt skadelige avfalls materiale ble dette ikke fult opp videre.

Fra gjennomføringen av fall testen ble det også funnet ut at hvis dronen mistet kontroll var det en viss sjans for at propguarden lirket seg fast mellom glideplatene og beskyttelsesveggene. Dette er et stort problem da dette hindrer arenaens funksjon til å resette dronen tilbake til utgangspunktet. Det kan i midlertidig løses ved å legge til en firkant i profilen slik at platene går opp i dette hullet og tetter. Dette vil fjerne problemet og bør vurderes før redesign og nye plater eventuelt skal lages.

7.3 Aerodynamikk

De aerodynamiske aspektene ved arenaen var en av de mer usikre aspektene rundt konseptet. De harde platene i bunn og veggene rundt ble antatt å kun ha en vindtunnel effekt og skape betydelig turbulent luft. I hvilken grad dette ødelegger for reinforcement learning er i seg selv et vanskelig spørsmål å besvare. Det ble derimot antatt at hvis det var mulig å fly uten store komplikasjoner manuelt burde en AI-kode kunne gjøre jobben bedre og dermed kunne anses som en egn plattform.

Under ”åpen dag sammenstillingen” 5.6 ble flere interessante aspekter rundt flygbarheten bemerket. Dronen som ble benyttet var av enn modell som skapte betydelig mer luftstrømninger da motorene har betydelig mer kraft og propellene var større. Tross dette og tette bunnplater var dronen helt fint flygbar også på et nybegynner-nivå. Både sug mot veggene, vibrasjoner og uforutsigbarhet i flygingen ble observert under flygingen, men ikke på et uholdbart nivå.

Bakkeeffekten skal i teorien ikke ha mye å si på grunn av de små propellene som benyttes. Den ble så vidt registrert med Gaupner dronen, men nesten ikke under testing med UiA dronen. Tanken var å kunne heve platformen som dronen står på før take off, dette bør fjerne denne effekten helt.

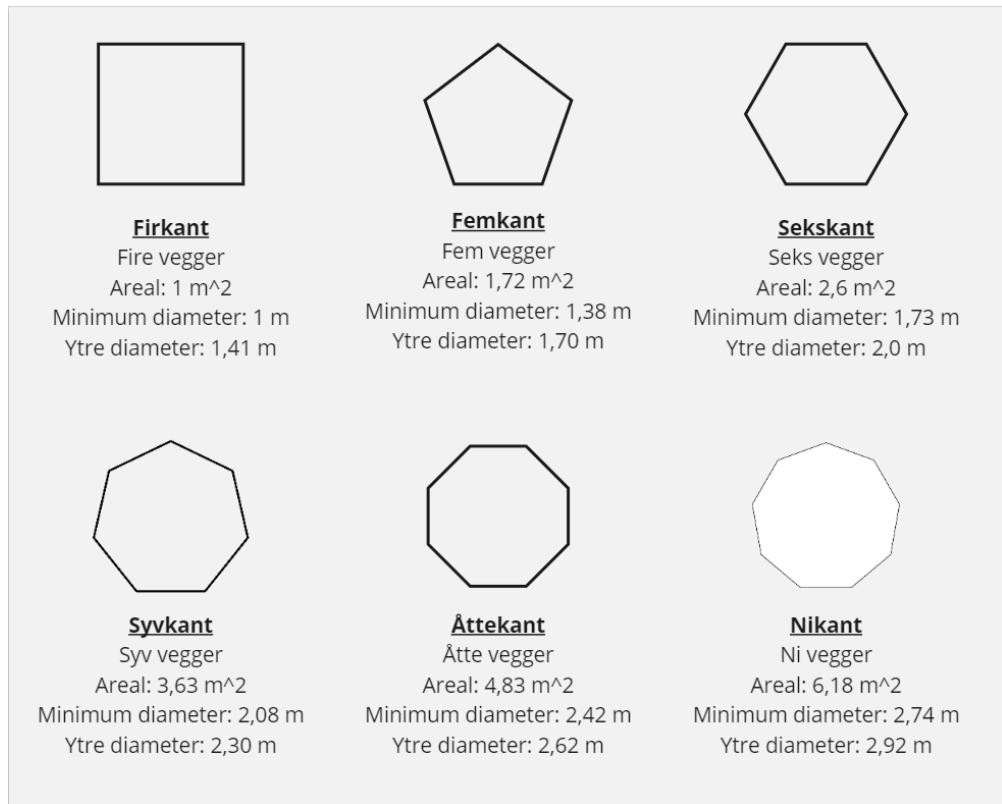
Ved flyving i lave høyder, rundt 10 til 15 cm over platene ble en effekt som antas å være en vortex ring state observert. I disse høydene var det vanskelig å holde en stabil høyde og det krevede mer kraft. Formen på platene og det faktum at de er helt tette ser ut til å presse luften opp igjen og skape samme effekt som om dronen skulle mistet høyde. Dette ble imidlertid mye mindre betydelig med UiA dronen, og ønsket er å benytte perforerte plater i istedenfor. Slike plater antas å kunne hjelpe til med å bryte opp luftstrømmen i større grad og skape mindre turbulent luft for lettere flyging. Hvilken grad dette ville hjulpet er vanskelig å si uten å forsøke da bare en avansert CDF kunne gitt gode svar. Det er likevel å anta at det kunne hjulpet i noe grad.

Suget inn mot veggene er forsatt den største utfordringen med tanke på UiA dronen. Hvis propguarden nærmer seg veggene med noen centimeter kan man betydelig kjenne et trekk. Veggene er ikke blitt perforert som foreslått i konseptet, da laser cutteren viste seg å være for liten for platene. Effekten er heller ikke veldig stor sammenlignet med det flygbare området. Selv ved å unngå sidene med flere centimeter kan man fly dronen manuelt inne i arenaen, noe som ble satt opp som et kjennemerke for flygbare forhold for en reinforcement learned kode. Dette viser også at størrelsen på arenaen er stor nok og at valget av lengden på veggene traff godt med tanke på å balansere opp de aerodynamiske effektene.

7.4 Versatilitet

7.4.1 Mekanisk mutiplisering

Under konsept fasen var en betydelig del av konsept designet basert på muligheten for å kunne plassere ut mange enheter tett stølet sammen. Utifra kravene er en arena som er visuelt pen å se på ønsket. En som er mer rettet mot bruken hvor et fåtall av disse kunne plasseres ut på et sted hvor mange kan passere dem og se på de. I en sammenheng hvor disse mer effektivt skal kunne benyttes til samling av data eller opptrening og testing av protokoller vil muligheten til å kunne benytte flere av gangen være tidsbesparende. En arkitektur som derfor gir muligheten for å sette opp flere slike uten å kreve mye tomrom vil gi mulig store fordeler i et senere stadie. Muligheten til å pakke enhetene tett vil også komme svært godt med under lagring.



Figur 7.1: Sammenlikning mellom mangekanter

Bunnsekjonens flerkantede form har fordelen at det muliggjør en arena som kan modulært bygges av flate sider, som igjen er betydelig billigere en å skulle få spesial tilpasset runde former på veggene. Flere antall kanter ble veid opp mot hverandre i konsept fasen og en oppsummering er vist i figur 7.1. Et lavere antall vegger ville betydd mindre kostnader i form av materialer, mens et høyere antall kanter vil gi bedre plass inne i arenaen til å fly på, tatt at det eksisterer en maks bredde på veggene som er realistisk og det minste tverrsnittet blir større. Tverrsnittet har mye å si for hvordan det føles å fly i arenaen da dette er maks "avvik" man kan ha uten å bli slått i bakken.

Sekskant fasongen balanserer alle disse egenskapene godt opp mot hverandre samtidig som den er særstilt til å pakke enhetene tett inntil hverandre. Sekskant ideen ble opprinnelig inspirert av bivoks plater som nettopp pakker sekskanter i et tettpakket rutemønster. Tatt arenaen sin størrelse i betrakning vil en tett pakking av disse være helt essensielt for å ikke oppta store arealer. En mulighet er å sette arenaene i doble rader, dette vil gi tilgang til alle arenaene for nødvendig vedlikehold samtidig som plassen benyttes optimalt.

7.4.2 Felles hub

I designet av det overordnende styringssystemet 4.1 ble styringen delt opp etter hvilke oppgaver det har i arenaen. Noe av baktanken for nettopp dette er muligheten for å plassere deler av systemet andre steder enn i nettopp arenaen. Dette sammen med muligheten for vedlikehold og oppgraderinger. En forespeilet mulighet for arenaen er å kunne bruke den som en reklameringsenhets, grunnet sitt visuelle uttrykk. Enheter kunne vært plassert ut på videregående skole, vitensentere og lignede institusjoner. Hvis en enhet kunne stått i et slik system og øvd hvor publikum har mulighet til å se den, ville dette kunne fungere som en god reklamere for hva UiA har muligheten til å få til. Som igjen både kan bygge omdømme og sikre fremtidige studenter.

Arenaens desidert dyreste del er imidlertid PC'en utstyrt med deler nødvendig for å foreta reinforcement learning. Denne delen er derfor tenkt slik at denne kan flyttes og eksistere et annet sted men forsatt jobbe ad-hoc fra selve arena enheten. Da vil kun en enkelt spekket maskin trenges i enheten og gjør grunnleggende bildegenkjenning, kommunikasjon med dronen og kommunikasjon med aktueringssystemet.

Selv AI maskinen kan være plassert på for eksempel universitetet med kun internett forbindelse med den eksterne enheten om dette lar seg gjøre på en tilfredsstillende sikker måte. Det kan da videre tenkes at denne maskinen snakker med flere eksterne enheter plassert utover på forskjellige lokasjoner. Dette vil også ha den effekten av at det vil synes at dronen blir betydelig bedre over natten eller over få iterasjoner. Noe lengre ventetid på opplasting vil kunne oppstå men dette bør ikke bli et problem så lenge en generasjon eksisterer på flight controlleren under flyving.

Ved å plassere de dyreste enhetene utenfor rekkevidde vil tapet av hærverk eller stjeling kunne begrenses betraktelig. Enheten i seg selv er relativ tung og litt plundrete å flytte, samt alle enheter er skrudd fast. Tyveri vil derfor være krevende og trenge noe planlegging. Dette kan ikke forhindres, men tapene vil kunne begrenses. En viss omtanke for hvor enhet plasseres vil derfor antagelig forsatt kreves.

Denne figurasjonen med en felles hub som itererer over flere enheter vil også være mulig i det tidligere scenariet hvor flere enheter er stasjonert i et rom. Kommunikasjonen internt i arenaen er tidligere tenkt til å skulle foregå per ethernet. Hvis hver arena her en switch vil disse også kunne seriekobles til neste og på denne måten lage en kjede på et lokalt nettverk.

7.4.3 Trening for nybegynnere

Selv om manuell flyving er en beskrevet del av oppgaven, kom det fram under testing at også enheten fungerer svært bra som test arena for nybegynnere. Det er i dag relativt vanskelig å starte og fly droner. Man vil ofte jobbe seg oppover fra små billige til dyrere varianter. Dette kan fort bli dyrt og krever ofte et stort område hvor man kan fly på.

Under testingen av åpen dag sammenstillingen fikk flere personer, som aldri hadde fløyet en drone, lov til å forsøke å fly inne i arenaen. Dette gjorde flere med stor suksess og klarte å holde dronen i luften i en lengre periode. Dette ga stor mestringsfølelse for nybegynner flygerene. Siden dronen fly i et lukket bur var det mulig å ha dronen flygende med publikum tett på. En mulig bruk av arenaer er derfor nettopp at personer får mulighet til å fly dronen manuelt i et kontrollert miljø, men forsatt relativt tett på mennesker. Funksjonen med at dronen faller til midten når den blir terminert gjør også at det er lett å få igang dronen hvis den har krasjet, uten å måtte bekymre seg for at den ligger tett på veggene.

7.5 Porterbarhet

Som en del av kravene til arenaen og som en sentral del av konseptet er graden av porterbarhet en viktig faktor. Konseptet bygger på muligheten for å kunne demontere de største delene på en enklast mulig måte for å kunne flytte arenaen. Dette er igjen viktig med tanke på eventuelt bruk som framvisning objekt hvor den må kunne fraktes og monteres til nye lokasjoner uten å kreve for mye verktøy eller ressurser.

Den sammenstilte arenaen møter dette kravet godt på veg. Veggene, sideveggene og toppen monteres ved å slide ned i slisser som holder komponentene på plass. Glideplatene legges oppå og kan

presses ned for å sette sammenstillingen i press. Disse kan festes ved å skru fast som vil kreve mer montering, men fortsatt minimalt med utstyr. Skuffene med aktueringssystemet og PC'en kan trekkes ut og kobles av med hurtigkoblingene på ledningene. I disse skuffene vil de dyreste komponentene ligge og muligheten for å frakte disse på en sikrere måte, men forsatt beskyttet av boksene gir trygg frakt.

For de gjenværende delene som ikke kan demonteres er det lagt vekt på å feste godt. De tre bunnplatedelene er sveiset sammen og danner en solid flate. Hjørnene er boltet på med M8 umbraco skruer som gjenger i 3D-printen. Dette er en monteringsmetode som holder godt så lenge de kun er skrudd fast og ikke skrus ut igjen. Midtsekjonen er boltet på gjennom bunnen og bunnringen, med M6 skruer som har skive og mutter på toppen. Midtaktuatoren er designet for en skrue gjennom rotasjon senter som vil med to nållager skrus fast i midtsekjonen. Disse delene danner en enhet.

Denne enheten har forsatt en for stor radius til å passere gjennom en dør horisontalt, men har kun en høyde på 40 cm. Hjørnene danner også gode håndtak som tåler at seksjonen løftes i disse på høykant. Det ble i konseptet diskutert å ha håndtak på tre av sidene for frakt. Under konstruksjonsprosessen ble dette ikke prioritert, og når frakten blir dette ofte gjort på høykant, noe de opprinnelige håndtakene ikke ville hjulpet med. Sammenstillingen ble også funnet til å være for ustabil for frakt med vegger og topp og håndtaket ville heller da ikke hjulpet og potensielt bare gjort enhetene vanskelige å sette tett inntil hverandre.

Håndtakene ble derfor forkastet, men er en mulighet å legge på hvis en bedre løsning skulle dukke opp. Arenaen har generelt en god grad av porterbarhet tatt størrelsen og vekten i betraktning og innfriar dette kravet godt.

7.6 Flygbarhet av dronen

Flygbarheten av dronen har blitt vurdert utifra testene i kapitel 5.4. Når det ses på flybarheten kommer det helt an på hva man vurderer det opp imot og hva formålet til dronen er. Ut ifra fly testene ute med 2s batteri var dronen godt manøvrerbar. Dronen hadde nok kraft til å kunne flys opp til ønskelig høyde, men den gikk ikke fort oppover. For å få den til å stige må man omtrentlig bruke full throttle, og for å hovre dronen var det nødvendig å holde en throttle på mellom 50% til 75% avhengig av strømnivået på batteriet. Dette gjør at dronen bruker mye energi på å holde seg oppe i luften og spesielt for å stige oppover. Samtidig som det vil belaste batteriet hardt ettersom dette må sende over 50% konstant. Fly tiden med disse batteriene var også varierende ettersom kvaliteten i batteriene varierte, men på de beste batteriene oppnådde man en flytid på åtte til ti minutter.

Mangelen på thrust fremkom også unner flyging i maks vinkel i angle mode. Om man prøvde å kjøre med full gass fremover ville dronen falle nedover mens den akselererte og tilslutt treffe bakken om den ikke ble rettet opp. Uten om mangelen på thrust var dronen lett å styre og bevegde seg fint etter stikkmandoene. Selv når man fløy høyt oppover og slapp den i fritt fall var det lite oscilleringer i drone uten at PID kontrolleren ble tunet. De få oscilleringene som oppstod kunne antageligvis også blitt fjernet om det hadde vært tid til å tune PIDen under flyingen. Til tross for at det var endel kraftige vindkast under flyvningen klarte dronen å holde seg relativt stabil i disse vindkastene. Det største problemet var at dronen ikke alltid hadde nok kraft til å stå imot vinden.

Dronen hadde en overraskende god evne til å hente seg inn i krevende situasjoner. Blant annet etter at dronen ble kastet hardt ut i luften klarte den nesten alltid å hente seg inn og stabilisere seg i angle mode. Den hadde problemer noen få ganger hvor den endte opp på hodet og klarte da ikke å snu seg rundt føre den traff bakken. Dette kan være grunnet med den manglende kraften i batteriet. Dronen fløy også veldig fint i horizon mode og acro mode, den klarte fint å ta flipp i horizon mode. Dette krevde derimot mye høyde føre fluppen ble utført, for å unngå å treffe bakken. Dette ville nok også vært lettere om maks rotasjonshastighet hadde vært stilt opp høyere, men disse innstillingene ble først funnet etter testen. PID kontrolleren fungerte også godt i horizon og acro mode, men det var vanskelig å vurdere PIDen i disse modusene grunnet vindforholdene. Landingen oppå fuglekassen som ble gjennomført var krevende, men dronen klarte å holde en stabil høyde som gjorde at man rolig kunne manøvrere dronen over fuglekassen og senke trottelen for å lande. Dette gikk overraskende nok fint, men gikk ikke hver gang og tokk litt lenger tid enn nødvendig.

Fra testen med 3s batteriet ble dronen naturligvis mye sprekere, og flygeegenskapene minnet mere om en racing drone. Med dette batteriet var det ingen problemer for dronen å holde maks vinkel i angle mode. Som et resultat av det kraftige batteriet ble dronen vanskeligere å styre, det ble også lagt merke til at dronens PID innstillingar var endel dårligere enn med 2s batteriet. Dette kan komme av at dronen nå har mere krefter og det fremkommer oscillasjoner i situasjoner hvor 2s batteriet ikke hadde krefter nok til å lage oscillasjoner. Oscillasjonene kan også ha kommet av at batteriet hadde andre dimensjoner og vekt, enn 2s batteriet. Når dronen ble fløyet høyt og trottelen ble kuttet fremkom det store og kontinuerlige oscillasjoner i dronen mens den falt, dermed hadde det vært nødvendig å tune PID kontrolleren nøyaktig for å få et bedre bilde av PID kontrolleren og flygeegenskapene med dette batteriet. Selv om batteriet var 600Mah mindre en 2s batteriet så ble flygetiden på 12 til 15 minutter som er litt lengre enn med 2s batteriet. Det vil også være fint mulig å sette inn ett 3s batteri med større kapasitet. Da vil antagelig flytiden øke til opp mot 20 minutter.

Når dronen ble fløyet inni arenaen var flygeegenskapsene noe annerledes enn utendørs. Den var fortsatt grei å manøvrere, men ble vanskeligere å styre når man var nærmere veggene eller gulvet i arenaen. Dette var antakeligvis grunnet med den reflekterte turbulensen fra platene. Disse gulvplatene skulle også være perforerte i utgangspunktet, som ville minsket denne effekten betydelig. Inni arenaen ble det kun kjørt med 2s batteri, siden dette behandler dronen og arenaen bedre og det var heller ikke noe behov for en sprekere drone i arenaen. Oscilleringene som fremkommer av turbulensene er også mulig å se i figur 6.7. Her kan man se at oscilleringene i både x og y aksen er betydelig høyere enn oscilleringene som fremkom når det ble flydd ute. Om man sammenligner ute flyvningen med flyvningen i arenaen 6.7 6.8, så ser man at ute flyvningen også har oscilleringer, men disse har bare halvparten av amplituden til arena flyvningene. Inni arenaen er det oscilleringer rundt 0.2g mens ute testen viser at oscilleringene ligger på 0.1g.

Totalt sett kan dronen manøvrere slik som man forventer fra en drone, men om flygeegenskapene er best med ett 2s batteri eller ett 3s batteri kan diskuteres. Til hovedformålet som er arenaen, vil det helt klart være en fordel å bruke 2s batteriet. Dette vil gjøre det lettere for AI'en å trenere opp dronen og man slipper ekstremt harde krasjer som er en fordel for hele systemet. Samtidig vil det være mye bedre for dronen å bruke ett 3s batteri om den skal fly autonomt ute, da vil den ha mere krefter til å kunne motstå vind og andre værforhold i naturen. I tillegg vil den kunne fly lengre føre den trenger en lade stopp. Optimalt sett vil det nok være best å benytte både 2s og 3s batteri i dronen. Da kan den starte AI læringen med ett 2s batteriet, og deretter gå over til 3s batteriet, etter at den har lært grunnprinsippene rundt flyvning. Dermed vil dronen kjøre rolig under starten av AI læringen, men når den blir brukt ute har den kreftene til 3s batteriet.

7.7 Dronens holdbarhet

Dronens holdbarhet ble satt på prøve under flyvetestene både inni arenaen og ute. Fra flyvetestene ute kan man se at maks akselerasjon er under 8g i hver av aksene. IMUen tåler 16 g i hver av aksene 2.10 som betyr at de største støtene ikke er for store for IMUen. Det gjør at det ikke er nødvendig å resette IMUen etter en krasj. Videre klarer dronen seg godt inne i arenaen, med unntak av propguarden. Denne ble lett dyttet inn i propellene og ble etterhvert kuttet av på forskjellige steder. Dette kommer nok av at den er for myk, men den nyeste varianten rakk ikke å bli testet inni arenaen. Denne varianten ville antakelig holdt seg bedre siden den både er stivere og har større avstand til propellene.

Dronen fikk testet seg hardere når den ble flydd ute i naturen. Resultatene av dette var varierende, noe tålte den veldig godt og annet dårlig. Her var igjen propguarden det svake punktet til dronen. Dette var den forsterkede versjonen, men ved harde støt i bakken og inn i trær ble den allikevel dyttet inn i propellene. Dette er nok fordi PP plasten er såpass bøyelig i tynne strukturer, og hvis den skal lages robust vil den ødelegge for aerodynamikken. Det kan være en ide å lage denne i PLA istedenfor PP. Det vil gjøre den mye stivere, og støtene inni arenaen vil nok være for svake til at den sprekker.

Resten av dronen klarte seg svært godt også ute i naturen. Det kunne fremkomme skader på skroget ved krasj i stor fart eller fall fra store høyder. Dette oppstod ved bruk av 3s batteriet,

og når den ble kastet ut i luften og dunket ned i bakken. Dette er tilfeller som ikke vil oppstå i arenaen eller ved vanlig flyvning. Det svakeste punktet på skroget var i hjørnet mellom gulvet og kant veggene til armene. Dette hjørnet kunne sprekke ved fall fra store høyder. Skaden kommer av at det er en stress konsentrasjon i dette hjørnet, som også fremkom på droptesten i Solidworks. I figur 6.5 kan det ses at von Mises er over 10 MPa i hjørnet og øker jo nærmere man kommer den skarpe kanten. Derved var det mulig å endre dette ut fra resultatet av flyvetesten og droptesten i solidworks. En fillet ble lagt langs hele kanten som fjerner store deler av stress konsentrasjonen.

Topplokket på dronen kunne løsne i krasjene under testen. Om dette skjer under AI treningen vil det påvirke utviklingen. Grunnet at batteriet måtte byttes flere ganger under flytesten ble ikke lokket til dronen limt fast i skroget, og problemet med at lokket faller av vil antagelig forsvinne når det blir limt fast. Innmaten i dronen overlevde de fleste krasjene med dronen. Den taklet vekten av batteriet når den falt rett ned på bakken, dette kan også ses på analysen fra Solidworks 6.5. I denne figuren vises det ikke noe farge på innmaten til dronen, som betyr at von Mises er under ti MPa. Dette er godt under PLA sin strekkfasthet som ligger på 63 MPa 2.3.2. Derimot når dronen ble truffet hardt inn i sideveggene oppstod det at innfestningen til innmaten sprakk. Dette skjedde spesielt uten propguarden. Løsningen på dette kan være ett redesign av innfestningen til innmaten eller bruke ett annet materiale på innmaten. Det ble testet litt med PLA+ på innmaten, og det så ut til å fungere bra, men det ble ikke testet hardt nok med til å kunne si at dette er en endelig løsning.

Som det kan sees i figur 6.6 beveger bunngulvet i dronen seg opptil 1.5 mm ved støtet. Den største bevegelsen ligger i midten og er på 1.581 mm. Dette vil være uheldig for QI coilen som ligger rett ovenfor denne flaten. Resultatet av dette kan være at coilen får permanente deformasjoner, dette vil da påvirke ladeevnens. Det kan derfor vurderes å stive opp bunnen litt mere for å minimere deformasjonen. Om coilen festes godt nok vil allikevel ikke denne deformasjonen gjøre nok til at dronen ikke vil lade.

7.8 Kameradeteksjon

Ved bruk av Intel RealSense D435 Depth camera får man klare bilder med høy nok oppløsning til at det vil fungere med OpenCV og detektering ved bruk av ArUco-markøren. Det har i et eksperiment blitt sjekket hvorvidt det har en betydelig innvirkning hvordan IR-projectoren er plassert på kameraet i forhold til RGB-modulen. Eksperimentet er gjennomført i henhold til beskrivelsen i delkapittel 5.5, og resultatene er presentert i delkapittel 6.9. Resultatene ga en oppsiktsekkende avklaring på hvorvidt avstanden mellom modulene på kameraet har en innvirkning på hvor nøyaktig dybdemålingene er. Som fremvist i delkapittel 2.7 er det betydelig avstand mellom modulene, denne avstanden viser seg å ha innvirkning på nøyaktigheten på dybdemålingen. Det vil si at om arealet av det kameraet skal måle er for lite, er det en mulighet for at kameraet heller detekterer avstanden til bakrunnen, og i tilfellet Drone Arenaen, er avstandene så små at det vil være utslagsgivende.

Opphavet til denne feilavlesningen vil muligens være fordi kameraet vises- og musepeker 'eventet' er generert i RGB-modulen, mens det er IR-projectoren som detekterer den faktiske avstanden. Basert på resultatene er det en betydelig forskjell på hvor musepekeren er og hvor deteksjonen av avstanden foregår. Dette kan medføre store feilavlesninger hvis kameraet skulle tro at dronen er lengre ned enn det den egentlig er, altså at det er et avvik mellom nå-verdien og referanse-verdien.

Dette problemet gjelder imidlertid kun hvis horisontal retning er for smal i forhold til avstanden mellom de to modulene på kameraet. Med et relativt stort skrog som dronen, vil dette nødvendigvis ikke være en faktor, om avstandsmålingen kan holdes kontinuerlig i senter av ArUco-markøren, noe som burde være mulig å oppnå. Da vil feilkildene med at dybdemålingen hopper frem og tilbake elimineres, siden IR-projectoren ikke vil falle utenfor siden på skroget, slik den har gjort i Figur 6.9b. Her ser man at selv om musepekeren holdes godt inne på skroget, mener fortsatt målingen et det er 1568 mm ned til dronen. Den reelle avstanden er 531 mm som fremvist i Figur 6.9a når IR-projectoren treffer slik den intendert.

7.9 QI-ladingen

QI-lading systemet ble i liten grad utviklet på grunn av deler som kom for sent til å bygge et godt system. Den ene testen som ble gjort lovet svært godt for systemet da batteriet ladet på 1.5 ampere. Vanlige LiPo batterier skal lades iforhold til sin C-rate og i de fleste tilfeller er lading ved 1C det optimale. De batteriene som ble benyttet var derimot mulig for lading opp til 2 ampere og 1.5 er derfor et svært godt resultat.

1.5 ampere var ved nær perfekt oppkobling av coilene med vanlig papir teip over senderen og 1 mm PP plast under mottakeren. Det er derfor ikke sannsynlig at den vil lade like bra i arenaen, grunnet en offset på laderen når dronen sklir ned. Derimot skled dronen relativt rett til senter, og vil stort sett oppnå en god ladning. En bedre og egen laget PCB for ladingen hadde vært svært bra da det kunne gitt bedre kontroll, mer effektiv pakking og lagt til andre funksjonaliteter som en killswitch for aktivering av esc'er etter lading.

8 Konklusjon

Sluttproduktet av arbeidet, kalt Drone Arenaen, er en plattform utviklet for at en drone kan fly i et eget miljø for utvikling av Reinforcement Learning basert kode. Miljøet tillater dette uten at dronen skader seg selv, observatører eller omgivelsene. Det er implementert løsninger for datainnsamling ved bruk av et RealSense D435 dybde kamera. Ved hjelp av ArUco-markør og dybdemålinger genereres et posisjonsreferansesystem som er essensielt for RL koden. Kombinasjonen arenaen og dronen er omfattende testet ved bruk av den tiltenkte dronen, samt en kraftigere drone. Det var ikke under noen omstendigheter mulig for dronene å komme seg ut av arenaen, som betyr at den gir god beskyttelse for observatører. Designet på arenaen består av et visuelt pent og ryddig design, hvilket vil si at den egner seg til utstilling. Det er fortsatt potensial for videreutvikling både med tanke på mekanisk design og software. Eventuelle utbedringer kan bestå av implantasjon av LED-strips, og andre applikasjoner som kan føre til mer blikkfang.

I senter av Drone Arenaen er det designet en midtseksjon. Denne har to hovedfunksjoner bestående av å nullstille dronen i både posisjon og orientering etter hver iterasjon, samtidig som den innehar en QI-lader hvor ladingen av dronen vil foregå. Disse systemene fungerer godt hver for seg, men har ennå tilgode å bli testet som et helhetlig system hvor alle komponentene jobber sammen. Kommunikasjonen mellom dronen og styringsenheten er ikke ferdigstilt, men det er lagt opp til videre arbeid for dette. Det er i den sammenheng installert to 4U Rackmount kabinetter, det ene kabinettet er dedikert til implementasjon av datamaskinen som skal trenne AI og dermed kommunikasjon med dronen. I det andre kabinettet er all styrings- og aktueringselektronikk plassert.

Arenaen er designet til å være modulær, den kan bæres av to personer, og være ferdig montert på under en halvtime. Om den skal fraktes med bil vil det dreimot kreves et bagasjerom på $1.2\text{ m} \times 1.2\text{ m}$.

Hele dronen er blitt redesignet og oppfyller kravet om å veie under 250 g. Selv om det er blitt gjort et redesign, er det brukt samme grunnmål som på tidligere versjoner. Den er dermed fortsatt kompatibel med tidligere iterasjoner av det overordnede drone-prosjektet på UiA. En QI coil og en lader har blitt implementert og testet for at systemet fungerer. Denne vil muliggjøre at dronen kan lade trådløst fra tårnet i arenaen og dermed operere uavbrutt over en lengre periode.

Dronen har ikke blitt testet for kravet om sitt fulle potensiale, altså operere i Arenaen over en måned uavbrutt på egen hånd. Derimot er det hold i at dronen burde klare dette, da skroget har taklet harde tester godt, og arenaen ser ut til å kreve lite vedlikehold. Dette vil gjøre at systemet er godt egent for Reinforcement Learning, som vil kreve operasjon over noe tid. Det er avdekket noen stresskonsentrasjoner som krever utbedring, samtidig må armene gjøres rundere for å oppnå en bedre aerodynamikk. Med det nye designet vil dronen også ha mulighet for å gjøres vanntett, som vil muliggjøre at den kan operere utendørs i regnvær.

9 Videre arbeid

9.1 Arenaen

Etter testene er det flere aspekter som kunne vært jobbet mer med for å få arenaen operativ og forbedret. I utgangspunktet gir det mekaniske designet en fungerende plattform, men noe justering trengs hovedsakelig med tanke på glideplatene. Motor og nett er tatt raske løsninger for å få et komplett design og kunne vært revurdert komponent valg og LED baren ble nedprioritert for sammenstilling.

9.1.1 Glideplateter

Glideplatene trengs å optimaliseres i form av ny profil, og optimalisere glidningen. Å benytte de perforerte platene ville vært det optimale med tanke på aerodynamikk men dette krever at de får riktig overflatebehandling slik at dronen glir like godt som på de hele platene. Det bør da lages plater som tetter det diskuterte hullet mellom glideplatene og beskyttelsesveggene. Det kan også i denne prosessen vurderes om plate støtten på hjørnene kan heves litt mer for litt økt vinkel.

9.1.2 Bedrer motor valg

Motorene ble valgt utifra tilgjengelighet for rask utvikling og pris. Før eventuelt produksjon av flere arenaer burde det mulig tas en overgang spesielt med tanke på rotasjonsmotoren om det burde vært funnet et bedre sporbart alternativ med datablad.

9.1.3 LED

For det visuelle var det tiltenkt at LED strips skulle integreres inni hele bunsekssjonen slik at disse ville lyse opp og inn i arenaen. Dette ble nedprioriter, men er tenkt å styres med Raspberry Pi'en fra aktuatoreskuffen. Hvis de perforerte platene benyttes vil dette unne gi en svært kul effekt inne i arenaen og kunne ha forskjellige moduser etter som hvilken tilstand dronen er i. Dette bør integreres før arenaen blir brukt som utstillings objekt.

9.1.4 Nett

Nettet er laget foreløpig av en hønsenetting som er strekt opp. Det kan undersøke om det finnes bedre løsninger for å være helt sikker på at propellene ikke kan sette seg fast her samtidig som det i minst grad hindrer luftgjennomstrøming for å ikke forverre aerodynamikken.

9.2 Drona

Dronen bør ha litt redesign rundt aerodynamikken. Det neste steget er å redesigne armene til å bli litt tynnere og ha en rundere profil. Nåværende profil er litt for firkantet og klumpete. I tillegg skal det være mulig å gjøre dronen enda flatere nå som alle komponentene er bestemte. Uten om dette trenger også propguarden noen forsterkninger for at den skal bli mindre sannsynlig at den blir trykket inn i propellene.

9.2.1 Innmat

Innmatten til dronen bør ha noen endringer på innfestningene, både til cm4 og kakute f4V2. Det finnes egne dempere til FC kort. Det bør legges inn slik at disse kan brukes, slik at alt kan monteres på innmatten uten verktøy. Samtidig bør holderen redesignes slik at den passer til ett aktuelt 3s batteri.

9.3 Kommunikasjon

Arenaen mangler den nødvendige kommunikasjonen mellom styringsdatamaskinen og de perifere kontroll enhetene i flightcontrolleren til Drona og i Raspberry Pi enheten for kontroll av Arena aktuatorer. På et senere tidspunkt kan også kommunikasjon fra kontroll enheten ut på internett for kommunikasjon over lengre avstander vurderes, men dette bør komme på et mye senere tidspunkt.

9.3.1 Kommunikasjon mellom RPI og data

I Drone Arenaens nåværende konfigurasjon er det ikke noen kommunikasjon mellom Raspberryen og styringsdatamaskinen. Det må her bygges et nettverk ved bruk av ROS. Dette er en viktig del som innebærer å sende informasjonen hentet ut fra kamera og videre oppgi de nødvendige handlingene til Raspberryen. Denne kommunikasjonen er tenkt til å foregå per ethernet.

9.3.2 Kommunikasjon mellom drone og data

Kommunikasjonen mellom drona og dataen er nødt til å være stabil, her skal viktig informasjon kommuniseres. Dette inkluderer også termineringen, altså en killswitch ved overstigning av en gitt høyde. Det må her bygges et nettverk ved bruk av ROS basert på tidligere arbeid gjort med tanke på drone svermen. Denne kommunikasjonen må også være trådløs.

9.4 Innebygget Skjerm

En skjerm til å ha på siden av hubben er ønskelig å få implementert. Det er lagt opp til at denne skal bygges inn på motsatt side av Rackmountene, slik at det blir en applikasjon på annen hver side. Det er tiltenkt at denne skjermen skal ha et kult brukergrensesnitt som kan visualisere utviklingen til dronen, altså se Reinforcement learning i praksis. Tatt dimensjonene i betraktning kan størrelsen være opp mot 23", men type skjerm og alternativer må sees videre på.

Bibliografi

- [1] Shinpei Kato mfl. «An Open Approach to Autonomous Vehicles». eng. I: *IEEE MICRO* 35.6 (2015), s. 60–68. ISSN: 0272-1732.
- [2] *ABOUT US*. Ocean Infinity. Mai 2022. URL: <https://oceaninfinity.com/about-us/>.
- [3] *Verdens første autonome skip i drift skal erstatte 40.000 vogntogturer i året*. Teknisk Ukeblad. Mai 2022. URL: <https://www.tu.no/artikler/verdens-forste-autonome-skip-i-drift-skal-erstatte-40-000-vogntogturer-i-aret/382717>.
- [4] Derek Gregory. «From a View to a Kill: Drones and Late Modern War». eng. I: *Theory, culture & society* 28.7-8 (2011), s. 188–215. ISSN: 0263-2764.
- [5] Mehryar Mohri. *Foundations of machine learning*. eng. Cambridge, MA, 2018.
- [6] Sang-Yun Shin, Yong-Won Kang og Yong-Guk Kim. «Obstacle avoidance drone by deep reinforcement learning and its racing with human pilot». eng. I: *Applied sciences* 9.24 (2019), s. 5571. ISSN: 2076-3417.
- [7] Martin Mæland ; Martin D. Hermansen ; Jan-Henrik Skogstad. «Automatisk utskytningssystem for dronesverm». I: (2021). Unpublished Paper by University of Agder.
- [8] Ola Christoffer Våge Jørgen Mikal Benum ; Simon Hus ; Jan Petter Ottesen. «Modulbasert Drone Tilpasset Autonom Bruk i Motion Capture System». I: (2018). Unpublished Paper by University of Agder.
- [9] *Forskrift om luftfartøy som ikke har fører om bord mv*. [Online]. 2015. URL: <https://lovdata.no/dokument/SF/forskrift/2015-11-30-1404>.
- [10] Kristoffer Hansen Kruithof; Marius Egeland. «Modulbasert Drone For Svermekspimenter i UiA Motion Lab». I: (2019). Unpublished Paper by University of Agder.
- [11] Martin Holen, Kristian Knausgård og Morten Goodwin. «Development of a Simulator for Prototyping Reinforcement Learning-Based Autonomous Cars». I: *Informatics* 9 (apr. 2022), s. 33. DOI: [10.3390/informatics9020033](https://doi.org/10.3390/informatics9020033).
- [12] Lars Mæhlum. *3D-printing*. Store Norske Leksikon. Mai 2022. URL: <https://snl.no/3D-printing>.
- [13] Matt Jani. *Comparing FFF, SLA, and SLS technologies*. Ultimaker. Mai 2022. URL: <https://ultimaker.com/learn/comparing-fff-sla-and-sls-technologies>.
- [14] L. Jyothish Kumar, Pulak M Pandey og David Ian Wimpenny. *3D Printing and Additive Manufacturing Technologies*. eng. Singapore: Springer Singapore Pte. Limited, 2018. ISBN: 9789811303043.
- [15] *FFF vs FDM: Difference and Best Printers*. Digital manufacturing experts. Mai 2022. URL: <https://top3dshop.com/blog/fff-vs-fdm-difference-and-best-printers>.
- [16] *Reliable 3D printers that simply work for you*. Ultimaker. Mai 2022. URL: <https://ultimaker.com/3d-printers>.
- [17] *Creality 3D CR-10S Pro V2*. Creality. Mai 2022. URL: <https://www.creality3dshop.eu/collections/3d-printers/products/creality-3d-cr-10s-pro-v2>.
- [18] *Artillery SW-X2*. Artillery. Mai 2022. URL: <https://artillery3d.com/products/artillery-sidewinder-x2-upgrade-version-abl-auto-calibration-3d-printer-550405640mm-larger-printed-size-118111811575-inches-high-precision-dual-37>.
- [19] *Core-XY VS Cartesian*. Creality. Mai 2022. URL: <https://creality3d.shop/blogs/choose-your-3d-printer/core-xy-vs-cartesian>.

- [20] PRIMACREATOR SELECT NYLONPOWER CARBON FIBER. PRIMA. Mai 2022. URL: <https://primacreator.com/collections/special/products/select%5C%E2%5C%84%5C%A2-nylon-power-carbon-fiber>.
- [21] PRIMACREATOR EASYPRINT FLEX TPU. PRIMA. Mai 2022. URL: <https://primacreator.com/products/primacreator-easyprint-flex-tpu?>.
- [22] Resin. The Free Encyclopedia. Mai 2022. URL: <https://en.wikipedia.org/wiki/Resin>.
- [23] FDM Printers. stratasys. Mai 2022. URL: <https://www.stratasys.com/en/3d-printers/printer-catalog/fdm-printers/?filter=FDM>.
- [24] Plastic 3D Printers. 3D SYSTEMS. Mai 2022. URL: <https://www.3dsystems.com/3d-printers/plastic>.
- [25] 3D printing: Doing its part to save the environment. LEXOLOGY. Mai 2022. URL: <https://www.lexology.com/library/detail.aspx?g=84abb734-0bec-428c-9077-35aaeb73488f>.
- [26] PLA - 3D Printing's Biggest Lie. Makers Muse. Mai 2022. URL: <https://youtu.be/jCsnVp6mEbk>.
- [27] Are Moon Lamps REALLY 3D Printed? Makers Muse. Mai 2022. URL: <https://youtu.be/q0ndWCWwUV8>.
- [28] J. Borge ; L. Hilton ; M. S. Wad ; M. Okter. «MA223-STATISTIKK PROSJEKT». I: (mai 2021). Unpublished Paper by University of Agder.
- [29] PRIMACREATOR™ EASYPRINT PLA. PRIMA. Mai 2022. URL: <https://primacreator.com/products/primacreator%E2%84%A2-easyprint-pla?>.
- [30] PRIMASELECT™ PLA PRO. PRIMA. Mai 2022. URL: <https://primacreator.com/collections/pla-pro/products/primaselect%E2%84%A2-pla-pro?>.
- [31] PRIMASELECT™ PP POLYPROPYLENE. PRIMA. Mai 2022. URL: <https://primacreator.com/products/primaselect%E2%84%A2-pp-polypropylene?>.
- [32] PRIMASELECT™ PETG. PRIMA. Mai 2022. URL: <https://primacreator.com/products/primaselect%E2%84%A2-petg?>.
- [33] Rafael Muñoz Salinas. «ArUco: An efficient library for detection of planar markers and camera pose estimation». Unpublished Paper. Mar. 2022.
- [34] Adrian Rosebrock. AprilTag with Python. pyimagesearch. Mar. 2022. URL: <https://pyimagesearch.com/2020/11/02/apriltag-with-python/>.
- [35] Øyvind Grøn. elektrisk strøm. Store Norske Leksikon. Apr. 2022. URL: https://snl.no/elektrisk_str%C3%B8m.
- [36] Paul Bjørn Andersen. ledningsbånd. Store Norske Leksikon. Apr. 2022. URL: https://snl.no/ledningsb%C3%A5nd_-_fysikk.
- [37] John Haugan. Formler og tabeller. 3. utg. Kanalveien 51, 5068 Bergen: Fagforlaget, 2017. ISBN: 978-82-562-7437-6.
- [38] IPC-2221A Generic Standard on Printed Board Design. ASSOCIATION CONNECTING ELECTRONICS INDUSTRIES. Apr. 2022. URL: <https://www.ipc.org/TOC/IPC-2221A.pdf>.
- [39] Advanced Circuits. PCB Trace Width Calculator. URL: <https://www.4pcb.com/trace-width-calculator.html>. (accessed: 11.04.2022).
- [40] Image resolution. The Free Encyclopedia. Mai 2022. URL: https://en.wikipedia.org/wiki/Image_resolution.
- [41] Intel RealSense Depth Camera D435. Intel Corporation. Mai 2022. URL: <https://www.intelrealsense.com/depth-camera-d435/>.
- [42] Intel RealSense Product Family D400 Series. Versjon 012. Intel RealSense. Mar. 2022. URL: <https://www.intelrealsense.com/wp-content/uploads/2022/03/Intel-RealSense-D400-Series-Datasheet-March-2022.pdf>.

- [43] Sherif Hekal. *Compact Size Wireless Power Transfer Using Defected Ground Structures*. eng. 1st ed. 2019. Energy Systems in Electrical Engineering. Singapore: Springer Singapore : Imprint: Springer, 2019. ISBN: 981-13-8047-3.
- [44] Hugh D Young og Roger A Freedman. *Sears and Zemansky's University physics with modern physics*. eng. Pearson Education, 2015. ISBN: 9781292100319.
- [45] Tor Halvor Johansen. *aerodynamikk*. Store Norske Leksikon. Mai 2022. URL: <https://snl.no/aerodynamikk>.
- [46] *Aerodynamikk*. The Free Encyclopedia. Mai 2022. URL: <https://no.wikipedia.org/wiki/Aerodynamikk>.
- [47] Erik Tandberg. *vindtunnel*. Store Norske Leksikon. Mai 2022. URL: <https://snl.no/vindtunnel>.
- [48] Erik Tandberg. *bakkeeffekt*. Store Norske Leksikon. Mai 2022. URL: <https://snl.no/bakkeeffekt>.
- [49] Wayne Johnson. *Helicopter theory*. eng. New York, 1994.
- [50] *Vortex ring state*. The Free Encyclopedia. Mai 2022. URL: https://en.wikipedia.org/wiki/Vortex_ring_state.
- [51] *Kakute F4 AIO (V2)*. Holybro. Apr. 2022. URL: http://www.holybro.com/manual/Holybro_Kakute_F4_AIO_V2_Manual.pdf.
- [52] *High Performance 6-Axis MEMS MotionTracking™ Device*. TDK InvenSense. Apr. 2022. URL: <https://3cfeqx1hf82y3xcou1108ihx-wpengine.netdna-ssl.com/wp-content/uploads/2021/03/DS-000143-ICM-20689-TYP-v1.1.pdf>.
- [53] *About*. Betaflight. Mai 2022. URL: <https://betaflight.com/>.
- [54] Henrik H. Togba ; Eirik Helle ; Fridtjof H. Lunde ; Jakob U. Finnvold. «Flight controller design and implementation». I: (mai 2022). Unpublished Paper by University of Agder.
- [55] *RS PRO Electric Linear Actuator*. RS PRO. Jan. 2022. URL: <https://docs.rs-online.com/f505/A70000007396917.pdf>.
- [56] Brian Stone. *Motherboard Sizes Explained*. PC Guide. Mai 2022. URL: <https://www.pcguide.com/motherboard/faq/sizes-form-factor-explained/>.
- [57] *GEFORCE RTX 3090-SERIEN*. NVIDIA Corporation. Feb. 2022. URL: <https://www.nvidia.com/nb-no/geforce/graphics-cards/30-series/rtx-3090-3090ti/>.
- [58] *NVIDIA DGX Station A100 System Architecture*. Versjon 01. NVIDIA Corporation. Apr. 2022. URL: <https://images.nvidia.com/aem-dam/Solutions/Data-Center/nvidia-dgx-station-a100-system-architecture-white-paper.pdf>.
- [59] *NVIDIA DGX STATION A100*. NVIDIA Corporation. Feb. 2022. URL: <https://www.nvidia.com/en-us/data-center/dgx-station-a100/>.
- [60] *ASUS Motherboards-Components*. ASUSTeK Computer Inc. Mai 2022. URL: <https://www.asus.com/no/Motherboards-Components/Motherboards/All-series/>.
- [61] *ASUS PRIME Z690-P WIFI Hovedkort*. KOMPLETT. Mai 2022. URL: <https://www.komplett.no/product/1198165/datautstyr/pc-komponenter/hovedkort/intel-socket/asus-prime-z690-p-wifi-hovedkort#>.
- [62] *Hvordan velge en CPU*. Hitech Glitz. Mai 2022. URL: <https://hitechglitz.com/norway/hvordan-velge-en-cpu/>.
- [63] *Intel Core i9-12900K CPU*. KOMPLETT. Mai 2022. URL: <https://www.komplett.no/product/1197860/datautstyr/pc-komponenter/prosessorer/intel-core-i9-12900k-cpu#productinfo>.
- [64] V. M Passegger mfl. «The CARMENES search for exoplanets around M dwarfs – A deep learning approach to determine fundamental parameters of target stars». eng. I: (2020).

- [65] Dongkyu Lee, Woong Gyu La og Hwangnam Kim. «Drone Detection and Identification System using Artificial Intelligence». I: *2018 International Conference on Information and Communication Technology Convergence (ICTC)*. Okt. 2018, s. 1131–1133. doi: [10.1109/ICTC.2018.8539442](https://doi.org/10.1109/ICTC.2018.8539442).
- [66] Igor Lebedev, Aleksei Erashov og Aleksandra Shabanova. «Accurate Autonomous UAV Landing Using Vision-Based Detection of ArUco-Marker». I: *Interactive Collaborative Robotics*. Red. av Andrey Ronzhin, Gerhard Rigoll og Roman Meshcheryakov. Cham: Springer International Publishing, 2020, s. 179–188. ISBN: 978-3-030-60337-3.
- [67] *Raspberry Pi PWM Generation using Python and C*. ElectronicWings. Mar. 2022. URL: <https://www.electronicwings.com/raspberry-pi/raspberry-pi-pwm-generation-using-python-and-c>.
- [68] J. Borge ; M. S. Wad ; M. Okter. «MAS240 - Mechatronics Technical Report». I: (2021). Unpublished Paper by University of Agder.
- [69] *DRV8825 Stepper Motor Controller IC*. Texas Instruments. Mar. 2022. URL: <https://www.ti.com/lit/ds/symlink/drv8825.pdf>.
- [70] *TB6612FNG Driver IC for Dual DC motor*. Teoshiba. Mar. 2022. URL: <https://www.sparkfun.com/datasheets/Robotics/TB6612FNG.pdf>.
- [71] Jan Christian Bjerke Strandene. Personlig samtale. 2022.
- [72] *PolySwitch Resettable Devices*. Littlefuse. Mar. 2022. URL: https://www.elfadistrelec.no/Web/Downloads/_t/ds/polyswitch-radial-leaded_eng_tds.pdf.
- [73] *1N400x general-purpose diode*. The Free Encyclopedia. Mar. 2022. URL: https://en.wikipedia.org/wiki/1N400x_general-purpose_diode.
- [74] The PiHut. *How do I power my Raspberry Pi?* URL: <https://thepihut.com/blogs/raspberry-pi-tutorials/how-do-i-power-my-raspberry-pi>. (accessed: 20.04.2022).
- [75] Sergio Canu. *Distance detection with Depth Camera (INtel Realsense d435i)*. URL: <https://pysource.com/2021/03/11/distance-detection-with-depth-camera-intel-realsense-d435i/>. (accessed: 23.03.2022).
- [76] Adrian Rosebrock. *Detecting ArUco markers with OpenCV and Python*. URL: <https://pyimagesearch.com/2020/12/21/detecting-aruco-markers-with-opencv-and-python/>. (accessed: 23.03.2022).
- [77] The Robotics Back-End. *Raspberry Pi GPIO Interrupts Tutorial*. URL: <https://roboticsbackend.com/raspberry-pi-gpio-interrupts-tutorial/>. (accessed: 01.03.2022).
- [78] HORIZON HOBBY. *USB LiPo Charger*. URL: <https://www.horizonhobby.com/product/usb-lipo-charger/DYNC1062.html>. (accessed: 16.04.2022).

A Raspberry Pi kode

Listing A.1: Python grunnkode for styring av Arena actuatorer

```
1 #Martin Økter
2 #Main controll of mid actuators in the Drone Arena
3 # 10/2/2022
4
5 import signal
6 import sys
7 import RPi.GPIO as GPIO
8 import time
9 import matplotlib.pyplot as plt
10
11 #Setup GPIO
12 GPIO.setmode(GPIO.BCM)
13 GPIO.setwarnings(False)
14
15 y_list = [0]
16 x_list = [0]
17
18 #Class setup
19 class stepper_motor:
20     def __init__(self, direc, step, EN_pin, mode, base_wait_time,
21                  stepPerRot, roundUpDown):
22         self.direc_pin = direc
23         self.step_pin = step
24         self.EN_pin = EN_pin
25         self.mode = mode
26         self.base_wait_time = base_wait_time
27         self.stepPerRot = stepPerRot
28         self.roundUpDown = roundUpDown
29
30         self.resolution = {'Full':(0,0,0), 'Half':(1,0,0), '1/4':
31                           :(0,1,0), '1/8':(1,1,0), '1/16':(0,0,1),
32                           '1/32':(1,0,1)}
33         self.resolution_multiplier = {'Full':1, 'Half':2, '1/4':4,
34                                       '1/8':8, '1/16':16, '1/32':32}
35         self.resolution_choice = '1/16' #Best movement for the stepper
36           in use
37
38     GPIO.setup(self.direc_pin, GPIO.OUT)
39     GPIO.setup(self.step_pin, GPIO.OUT)
40     GPIO.setup(self.EN_pin, GPIO.OUT)
41     GPIO.setup(self.mode, GPIO.OUT)
42
43     #Standard pins set
44     GPIO.output(EN_pin, GPIO.LOW)
45     for i in range(3):
46         GPIO.output(self.mode, self.resolution[self.
47                     resolution_choice])
```

```

43     def go_rounds( self , num_of_rounds , direction ):
44         """
45             Make the stepper go compete rounds for basic manuvering
46             with more easy to use number
47             May handle float number
48             """
49             #Determine direction
50             if direction == 'Up':
51                 GPIO.output( self .direc_pin , GPIO.LOW)
52             elif direction == 'Down':
53                 GPIO.output( self .direc_pin , GPIO.HIGH)
54
55             mode_multi = self .resolution_multiplier[ self .resolution_choice ]
56             adjuster = 10
57
58             #Calculate number of steps
59             number_of_steps = round( num_of_rounds* self .stepPerRot*
60                                     mode_multi )
61
62             #Makes steps
63             for x in range( number_of_steps ):
64                 GPIO.output( self .step_pin , GPIO.HIGH)
65                 time.sleep( self .base_wait_time/(mode_multi*adjuster) )
66                 GPIO.output( self .step_pin , GPIO.LOW)
67                 time.sleep( self .base_wait_time/(mode_multi*adjuster) )
68
69             def go_up( self ):
70                 self .go_rounds( roundUpDown , 'Up' ) #Make the piston move up max
71
72             def go_down( self ):
73                 self .go_rounds( roundUpDown , 'Down' ) #Make the piston move down
74                 max
75
76             def set_resolution( self , new_resolution ):
77                 """
78                     Set a new reutution of the stepper adjusting between
79                         microstepping and full
80                 """
81
82
83             class pwm_motor:
84                 def __init__( self , motor_pin , freq , C1_pin , C2_pin , AIN1_pin ,
85                             AIN2_pin ):
86                     self .C1 = C1_pin
87                     self .C2 = C2_pin
88                     self .AIN1 = AIN1_pin
89                     self .AIN2 = AIN2_pin
90                     self .C1_count = 0
91                     self .C2_count = 0

```

```

92     self.hall_per_round = 300
93
94     self.curr_ang = 0
95     self.last_error = 0
96     self.i_last = 0
97
98     self.last_time = time.time() #Set a start time as a setpoint
99     self.dir = 0
100
101    GPIO.setup(motor_pin, GPIO.OUT)
102    GPIO.setup(self.AIN1, GPIO.OUT)
103    GPIO.setup(self.AIN2, GPIO.OUT)
104
105    #Setup hall interrupt
106    GPIO.setup(self.C1, GPIO.IN, pull_up_down=GPIO.PUD_UP)
107    GPIO.setup(self.C2, GPIO.IN, pull_up_down=GPIO.PUD_UP)
108    GPIO.add_event_detect(self.C1, GPIO.RISING, callback=self.
109                           hall_callback)
110    GPIO.add_event_detect(self.C2, GPIO.RISING, callback=self.
111                           hall_callback)
112
113    self.motor = GPIO.PWM(motor_pin, freq)
114    self.motor.start(0)
115
116    def hall_callback(self, channel):
117        """
118            For retriving number of passes from the hallsensor
119        """
120        if channel == self.C1:
121            self.C1_count += 1
122        if channel == self.C2:
123            self.C2_count += 1
124
125    def pwm_stop(self):
126        self.stop() #Stop all movement
127
128    def PID(self, in_val, last_val, k_list):
129        """
130            This function return the values needed
131            for calcualting th PID tuning mekanism.
132            -Økter
133        """
134        dt = time.time() - self.last_time
135        p_val = k_list[0] * in_val
136        i_mel = self.i_last + ((last_val + in_val)/2) * dt
137        i_val = i_mel * k_list[1]
138        d_val = ((in_val - last_val)/dt) * k_list[2]
139
140        val = p_val + i_val + d_val
141        self.i_last = i_mel
142        print(dt)
143        self.last_time = time.time()
144        return val, dt

```

```

144     def check_hall(self):
145         C1_now_count = self.C1_count
146         C2_now_count = self.C2_count
147
148         self.C1_count = 0
149         self.C2_count = 0
150
151     return C1_now_count, C2_now_count
152
153     def go_to_pos(self, new_ang):
154         """
155             Main control of the motor controlling the positin based on
156             feedbak from the hallsensor
157         """
158
159         global y_list
160         global x_list
161         error = new_ang - self.curr_ang
162
163         #if 0.00001 > error > -0.00001:
164         #    error = 0
165
166         if error > 0:
167             if self.dir == 1:
168                 self.motor.stop()
169                 self.dir = 0
170                 GPIO.output(self.AIN1, GPIO.HIGH)
171                 GPIO.output(self.AIN2, GPIO.LOW)
172             elif error < 0:
173                 if self.dir == 0:
174                     self.motor.stop()
175                     self.dir = 1
176                     GPIO.output(self.AIN1, GPIO.LOW)
177                     GPIO.output(self.AIN2, GPIO.HIGH)
178             else:
179                 GPIO.output(self.AIN1, GPIO.LOW)
180                 GPIO.output(self.AIN2, GPIO.LOW)
181
182             new_speed, new_dt = self.PID(error, self.last_error, [20, 10, 0])
183             new_speed = abs(new_speed)
184             if new_speed > 100:
185                 new_speed = 100
186             elif new_speed < 0:
187                 new_speed = 0
188             self.motor.ChangeDutyCycle(new_speed)
189
190             y_list.append(self.curr_ang)
191             x_list.append(x_list[-1] + new_dt)
192
193             self.last_error = error
194             C1_now, C2_now = self.check_hall()
195
196             if error > 0:
197                 self.curr_ang += 0.5 * (C1_now / self.hall_per_round) + 0.5 * (
198                     C2_now / self.hall_per_round)

```

```

197     elif error < 0:
198         self.curr_ang == 0.5*(C1_now/self.hall_per_round) + 0.5*(
199             C2_now/self.hall_per_round)
200
201
202 #Function setup
203 def signal_handler(sig, frame):
204     global y_list
205     global x_list
206     fig, ax = plt.subplots()
207     ax.grid()
208     ax.set_xlabel("Iteration")
209     ax.set_ylabel("Angle")
210     ax.plot(x_list, y_list)
211     plt.show()
212     GPIO.cleanup()
213     sys.exit(0)
214
215 #Initialize motors
216 #Initial basetime 0.0208
217 mainStepper = stepper_motor(22,23,24,(8,7,1),0.0416,200,2.8)
218 mainRotMot = pwm_motor(12,2000,25,16,27,17)
219
220 #mainStepper.go_rounds(1,'Down','Up',1234567890)
221 while True:
222     print('Initiate run')
223     mainStepper.set_resolution('1/16')
224     mainStepper.go_rounds(0.3,'Down')
225     print('Run compleat')
226     #mainRotMot.go_to_pos(1)
227     #signal.signal(signal.SIGINT, signal_handler)
228     #signal.pause()
229     time.sleep(10)

```

B Python grunnkode styringsmaskin

Listing B.1: Python grunnkode for testing av interrupt

```
1 import signal
2 import sys
3 import RPi.GPIO as GPIO
4 from time import sleep
5 import matplotlib.pyplot as plt
6
7 C1 = 25
8 C2 = 16
9
10 C1_count = 0
11 C2_count = 0
12
13 GPIO.setmode(GPIO.BCM)
14 GPIO.setup(C1, GPIO.IN, pull_up_down=GPIO.PUD_UP)
15 GPIO.setup(C2, GPIO.IN, pull_up_down=GPIO.PUD_UP)
16
17 x_list = [0]
18 c1_list = [0]
19 c2_list = [2]
20
21 def signal_handler(sig, frame):
22     global C1_count
23     global C2_count
24     print("C1 count: ", C1_count, "    C2 count: ", C2_count)
25     GPIO.cleanup()
26     sys.exit(0)
27
28 def button_pressed_callback(channel):
29     global C1_count
30     global C2_count
31     if channel == C1:
32         C1_count += 1
33         c1_list[-1] = 1
34         #print("Pin ", channel, " count is: ", C1_count)
35     elif channel == C2:
36         C2_count += 1
37         c2_list[-1] = 3
38         #print("Pin ", channel, " count is: ", C2_count)
39
40 def wait_for_edge():
41     CW_count = 0
42     edge = 4
43     for i in range(10):
44         if GPIO.input(C1) == GPIO.HIGH:
45             if GPIO.input(C2) == GPIO.LOW:
46                 CW_count += 1
47                 sleep(0.01)
```

```

48     if CW_count > edge:
49         print("CW -" + str(CW_count))
50     elif CW_count <= edge:
51         print("CCW -" + str(CW_count))
52
53
54 #print("Initsialiserer")
55 #GPIO.wait_for_edge(C2, GPIO.BOTH)
56 print("Start")
57
58 GPIO.add_event_detect(C1, GPIO.RISING, callback=button_pressed_callback
59 )
59 GPIO.add_event_detect(C2, GPIO.RISING, callback=button_pressed_callback
60 )
60
61 while True:
62     #x_list.append(x_list[-1] + 1)
63     #c1_list.append(0)
64     #c2_list.append(2)
65     #if len(x_list) > 1000: break
66     wait_for_edge()
67     sleep(0.1)
68
69
70 fig, ax = plt.subplots()
71 ax.grid()
72 ax.set_xlabel("Iteration")
73 ax.set_ylabel("Angle")
74 ax.plot(x_list, c1_list)
75 ax.plot(x_list, c2_list)
76 plt.show()
77
78 print(c1_list)

```

C Dybde kode styringsmaskin

Listing C.1: Python kode for måling av dybde med musepeker

```
1 # Importing the necessary packages
2 import cv2
3 from realsense_depth import *
4
5 point = (325, 250)
6
7 def mark_distance(event, x, y, args, params):
8     global point
9     point = (x, y)
10
11 # Initialize the RealSense D435 Camera
12 dc = DepthCamera()
13
14 # Creating the mouse event
15 cv2.namedWindow("Color frame")
16 cv2.setMouseCallback("Color frame", mark_distance)
17
18 while True:
19     ret, depth_frame, color_frame = dc.get_frame()
20
21     # Show distance for a specific point
22     cv2.circle(color_frame, point, 3, (0, 0, 255))
23     distance = depth_frame[point[1], point[0]]
24
25     cv2.putText(color_frame, "{}mm".format(distance), (point[0] + 5,
26                                                       point[1] - 15), cv2.FONT_HERSHEY_COMPLEX, 0.7, (0, 0, 0), 2)
27
28     # print(distance)
29     if distance == 0:
30         continue
31     cv2.imshow("Depth frame", depth_frame)
32     cv2.imshow("Color frame", color_frame)
33     key = cv2.waitKey(1)
34     if key == 27:
35         break
```

Listing C.2: Python kode klargjort for detektering av dronen

```
1 # Importing the necessary packages
2 import argparse
3 import time
4 import cv2
5 import cv2.aruco as aruco
6 import sys
7 import pyrealsense2
8 from realsense_depth import *
9
10
```

```

11 dc = DepthCamera()
12
13 # Constructing the argument parser and parse the arguments
14 ap = argparse.ArgumentParser()
15 ap.add_argument("-t", "--type", type=str,
16                 default="DICT_4X4_50",
17                 help="type of ArUCo tag to detect")
18 args = vars(ap.parse_args())
19
20 # Defining the names to the different ArUco-markers OpenCV supports
21 ARUCO_DICT = {
22     "DICT_4X4_50": cv2.aruco.DICT_4X4_50,
23     "DICT_4X4_100": cv2.aruco.DICT_4X4_100,
24     "DICT_4X4_250": cv2.aruco.DICT_4X4_250,
25     "DICT_4X4_1000": cv2.aruco.DICT_4X4_1000,
26     "DICT_5X5_50": cv2.aruco.DICT_5X5_50,
27     "DICT_5X5_100": cv2.aruco.DICT_5X5_100,
28     "DICT_5X5_250": cv2.aruco.DICT_5X5_250,
29     "DICT_5X5_1000": cv2.aruco.DICT_5X5_1000,
30     "DICT_6X6_50": cv2.aruco.DICT_6X6_50,
31     "DICT_6X6_100": cv2.aruco.DICT_6X6_100,
32     "DICT_6X6_250": cv2.aruco.DICT_6X6_250,
33     "DICT_6X6_1000": cv2.aruco.DICT_6X6_1000,
34     "DICT_7X7_50": cv2.aruco.DICT_7X7_50,
35     "DICT_7X7_100": cv2.aruco.DICT_7X7_100,
36     "DICT_7X7_250": cv2.aruco.DICT_7X7_250,
37     "DICT_7X7_1000": cv2.aruco.DICT_7X7_1000,
38     "DICT_ARUCO_ORIGINAL": cv2.aruco.DICT_ARUCO_ORIGINAL,
39     "DICT_APRILTAG_16h5": cv2.aruco.DICT_APRILTAG_16h5,
40     "DICT_APRILTAG_25h9": cv2.aruco.DICT_APRILTAG_25h9,
41     "DICT_APRILTAG_36h10": cv2.aruco.DICT_APRILTAG_36h10,
42     "DICT_APRILTAG_36h11": cv2.aruco.DICT_APRILTAG_36h11
43 }
44
45 # Verifying that the supplied ArUCo tag exists and is supported by
46 # OpenCV
47 if ARUCO_DICT.get(args["type"], None) is None:
48     print("[INFO] ArUCo tag of '{}' is not supported".format(
49         args["type"]))
50     sys.exit(0)
51
52 # Loading the ArUCo dictionary and grab the ArUCo parameters
53 arucoDict = cv2.aruco.Dictionary_get(ARUCO_DICT[args["type"]])
54 arucoParams = cv2.aruco.DetectorParameters_create()
55
56 # Looping over the frames from the color_frame
57 while True:
58     ret, depth_frame, color_frame = dc.get_frame()
59
60     # Detecting the ArUco-markers in the color_frame
61     (corners, ids, rejected) = cv2.aruco.detectMarkers(color_frame,
62             arucoDict, parameters=arucoParams)

```

```

63 # Verify that 'at least' one ArUco-marker was detected
64     if len(corners) > 0:
65         # Flatten the ArUco IDs list
66         ids = ids.flatten()
67
68         # Looping over the detected ArUco corners
69         for (markerCorner, markerID) in zip(corners, ids):
70
71             # Extracting the corners on the ArUco-marker (which are
72             # always returned in top-left, top-right, bottom-right, and
73             # bottom-left order)
74             corners = markerCorner.reshape((4, 2))
75             (topLeft, topRight, bottomRight, bottomLeft) = corners
76
77             # Converting each of the (x, y)-coordinate pairs to
78             # integers
79             topRight = (int(topRight[0]), int(topRight[1]))
80             bottomRight = (int(bottomRight[0]), int(bottomRight[1]))
81             bottomLeft = (int(bottomLeft[0]), int(bottomLeft[1]))
82             topLeft = (int(topLeft[0]), int(topLeft[1]))
83
84             # Drawing the bounding box of the ArUco-marker
85             cv2.line(color_frame, topLeft, topRight, (0, 255, 0), 2)
86             cv2.line(color_frame, topRight, bottomRight, (0, 255, 0),
87                         2)
88             cv2.line(color_frame, bottomRight, bottomLeft, (0, 255, 0),
89                         2)
90             cv2.line(color_frame, bottomLeft, topLeft, (0, 255, 0), 2)
91
92             # Compute and draw the center (x, y)-coordinates of the
93             # ArUco marker
94             cX = int((topLeft[0] + bottomRight[0]) / 2.0)
95             cY = int((topLeft[1] + bottomRight[1]) / 2.0)
96             cv2.circle(color_frame, (cX, cY), 4, (0, 0, 255), -1)
97
98             # Drawing the ArUco marker ID on the frame
99             cv2.putText(color_frame, str(markerID),
100                         (topLeft[0], topLeft[1] - 15),
101                         cv2.FONT_HERSHEY_SIMPLEX,
102                         0.5, (0, 255, 0), 2)
103
104             # Displaying the frame from the camera feed
105             cv2.imshow("Color frame", color_frame)
106             key = cv2.waitKey(1)
107
108             # If you press Esc, the loop will be terminated
109             if key == 27:
110                 break
111
112             # Sorting every thing up
113             cv2.destroyAllWindows()

```

D Bill of Materials

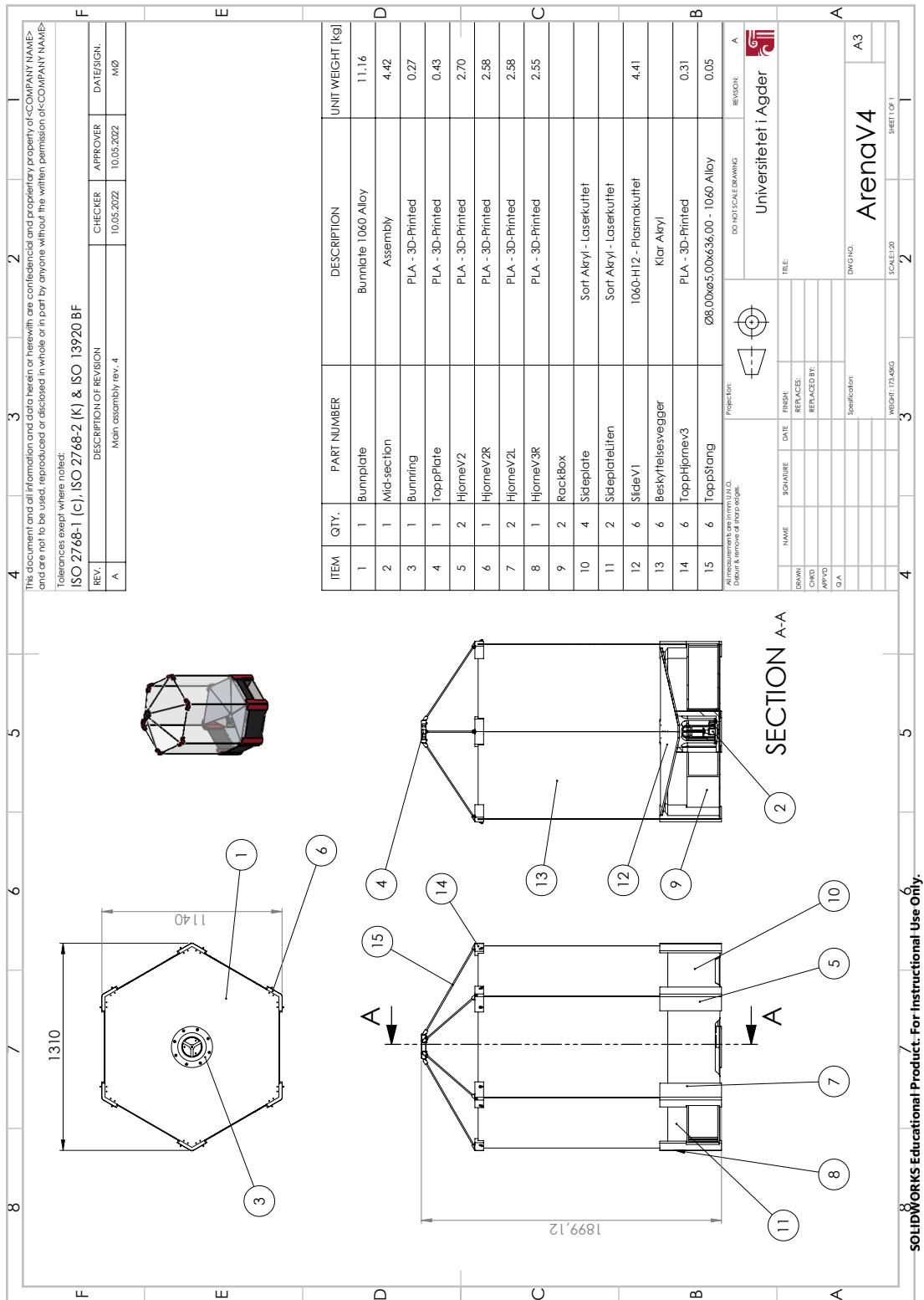
Tabell D.1: Bill of Materials for full assembly

Delenr.	Navn	Materiale	Antall	Subsystem	Kommentar
1	convsand	PLA	1	Aktueringsskuff	
2	RPIstand	PLA	1	Aktueringsskuff	
3	PowerStand	PLA	1	Aktueringsskuff	
4	PowerBack	PLA	1	Aktueringsskuff	
5	ToppHjornev3	PLA	6	Topp assembly	
6	MidSuport	PLA	1	Arena assembly	
7	platform	PLA	1	Aktuator assembly	
8	MovingStickV2	PLA	1	Aktuator assembly	
9	Top-ring	PLA	1	Aktuator assembly	
10	Flexledd	Aluminium	1	Aktuator assembly	
11	ToppPlate	PLA	1	Top assembly	
12	HjorneV2L	PLA	2	Arena assembly	
13	HjorneV2R	PLA	2	Arena assembly	
14	HjorneV3R	PLA	2	Arena assembly	
15	SlideV1	Perforert aluminium	6	Arena assembly	
16	TopStang	Aluminium Rør	6	Topp assembly	
17	Bunnplate-del1	4 mm stål	1	Arena assembly	
18	Bunnplate-del2	4 mm stål	1	Arena assembly	
19	Bunnplate-del3	4 mm stål	1	Arena assembly	
20	foring	Aluminium rør	3	Aktuator assembly	
21	SideplateLiten	Sort akryl	2	Arena assembly	
22	Sideplate	Sort akryl	4	Arena assembly	
23	RotTannhjulV3	Resin	1	Arena assembly	
24	Beskyttelsesvegger	Klart plastglass	6	Arena assembly	
25	top-stick	PLA	1	Aktuator assembly	
26	Stepper motor		1	Aktuator assembly	17" Nema dobbel stack
27	Kuleskrue	Stål	1	Aktuator assembly	
28	Ledeskrue	Messing	1	Aktuator assembly	
29	Rotasjonsmotor		1	Arena Assembly	SKU434373
30	M6 x 20 mm		6	Topp assembly	
31	M6 x 40 mm		31	Arena assembly	
32	M6 skive		8	Arena assembly	
33	M6 bolt		8	Arena assembly	
34	M6 hex x 30 mm		1	Aktuator assembly	
35	Thrust bearing 12 x 20 x 2 mm		1	Aktuator assembly	RS Pro
36	Thrust bearing 120 x 150 x 4 mm		1		RS Pro
37	Raspberry Pi 4 model B		1	Aktuator skuff	

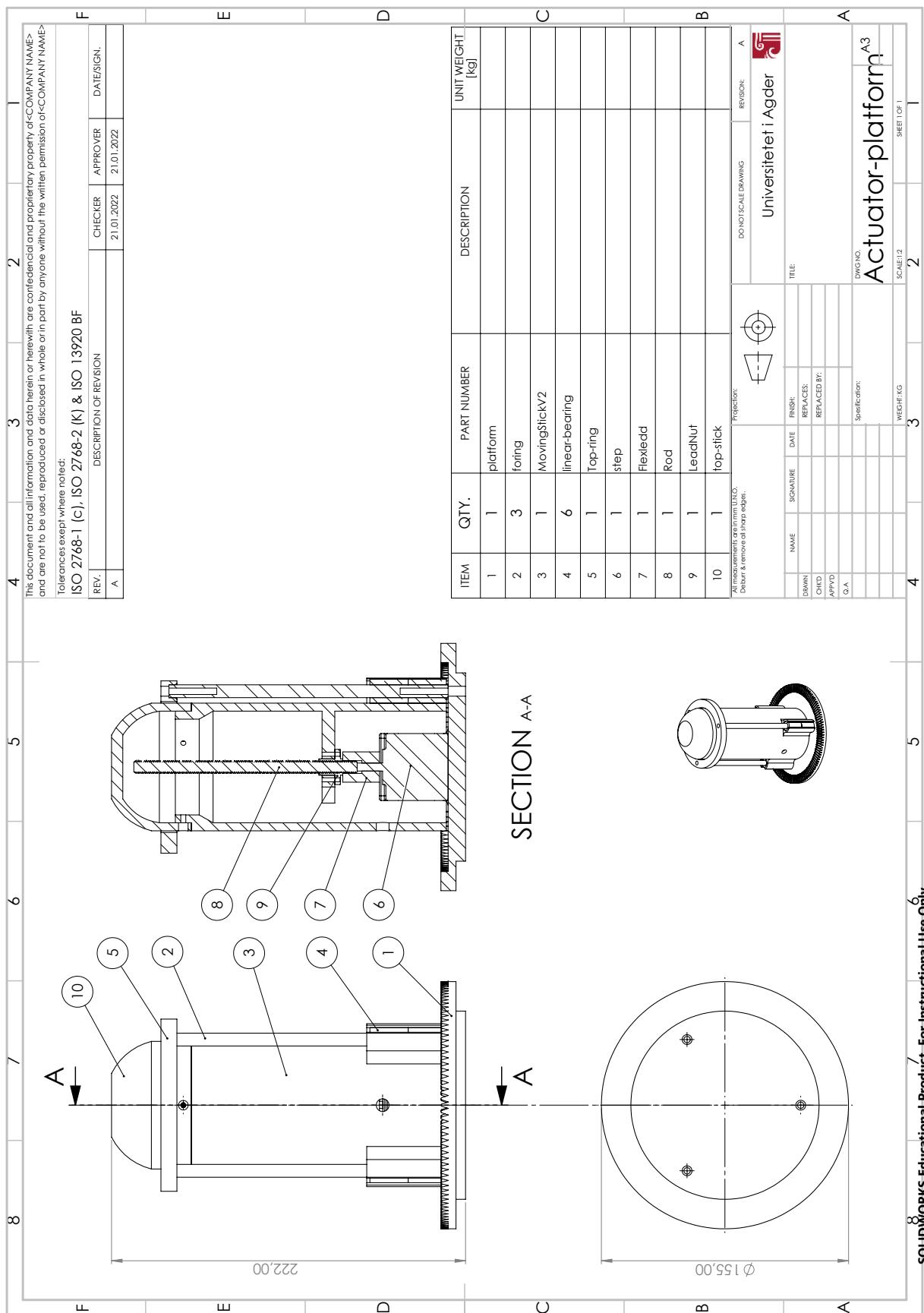
Tabell D.2: Bill of Materials for full assembly 2

Delenr.	Navn	Materiale	Antall	Subsystem	Kommentar
38	Step down conerter XL4016E1		1	Aktuaor skuff	
39	Motor controller PCB		1	Aktuator skuff	
40	DRV8825		1	Motor PCB	
41	TB6612FNG		1	Motor PCB	Polulu versjon
42	USB B to Pinout		2	Aktuator skuff	
43	Diode		1	Motor PCB	10A10
44	RXEF300		1	Motor PCB	RS Pro Skiring
45	Kondensator		1	Motor PCB	220 μF
46	Power supply S-250-12		1	Aktuator skuff	
47	Power plug C14		1	Aktuator skuff	
48	JM2 Drone Arena PCB		1	Aktuator skuff	Selv designet
49	Intel RealSense Depth Camera D435		1	Topp assembly	
50	Milenium Steel Box 4 45	Stål	2	Arena assembly	19" Studio Racks
51	2U Server Enclosure		2	Arena assembly	2U 19" Server Skuff
52	Toppskrogmk2	PP	1	Drone assembly	Lokk som dekker toppen på dronen
53	Bunnskrogmk2	PP	1	Drone assembly	Bunndelen av skroget
54	Midtplatemk2	PLA	1	Drone assembly	Plate som binder komponenter i dronen
55	Innmatmk2	PLA	1	Drone assembly	Holder rundt batteri og lader
56	Innmatspeilmk2	PLA	1	Drone assembly	Speilendt del av Innmatmk2
57	Propguardmk2	PP	1	Drone assembly	Beskyttelse til propellene
58	Kakute f4V2	PP	1	Drone assembly	Flight controller til dronen
59	Turnigy 1600Mah		1	Drone assembly	Batteri til dronen
60	BLheli_S 7A ESC		4	Drone assembly	Motorkontrollere til hver motor
61	SPM4650		1	Drone assembly	Reciver til dronen
62	Emax rs1106ii		4	Drone assembly	Motorer til dronen
63	GEMFAN 3025		2	Drone assembly	Propeller til dronen medurs
64	GEMFAN 3025		2	Drone assembly	Propeller til dronen moturs

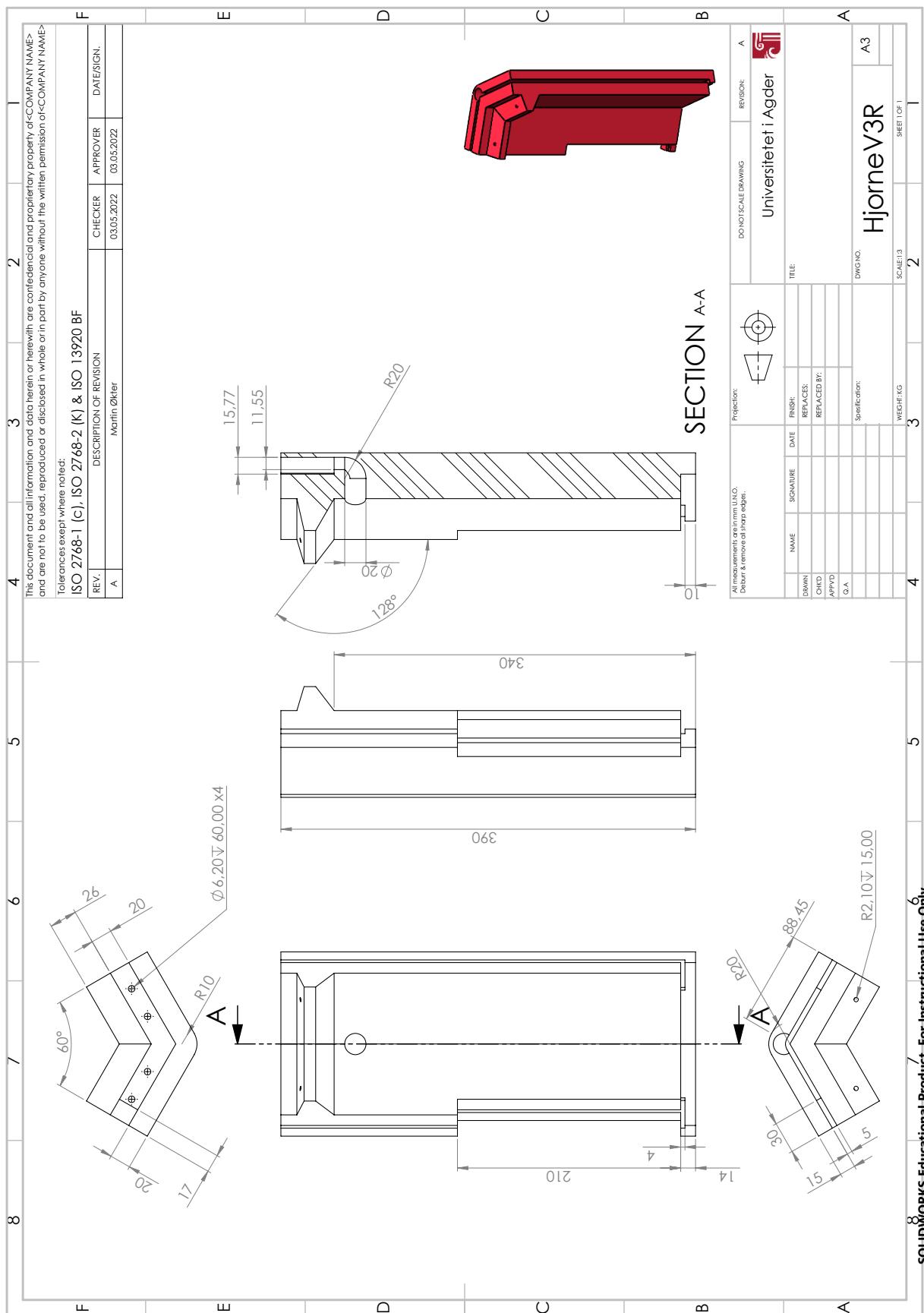
E Tekniske tegninger



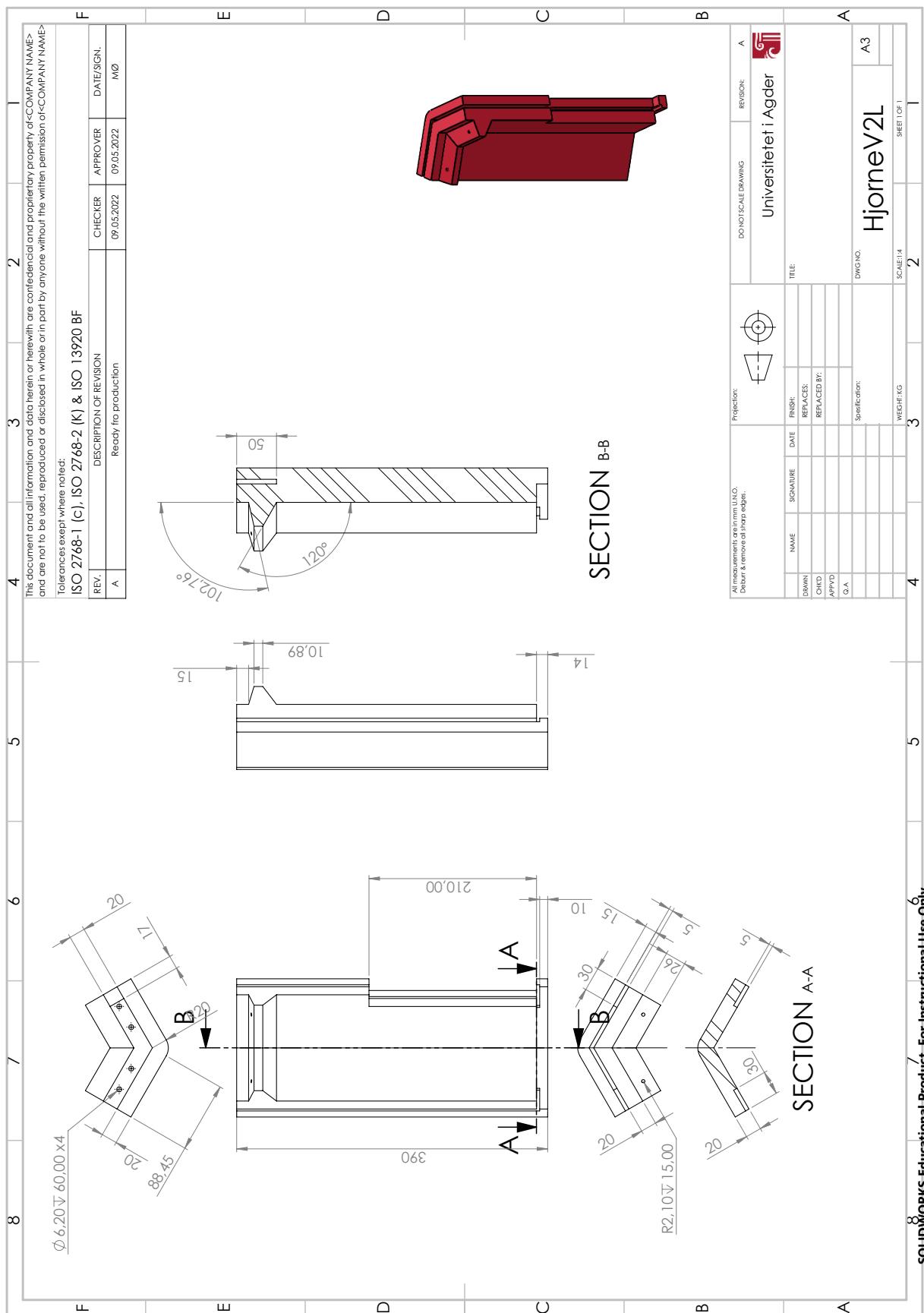
Figur E.1: Teknisk tegning av ArenaV4



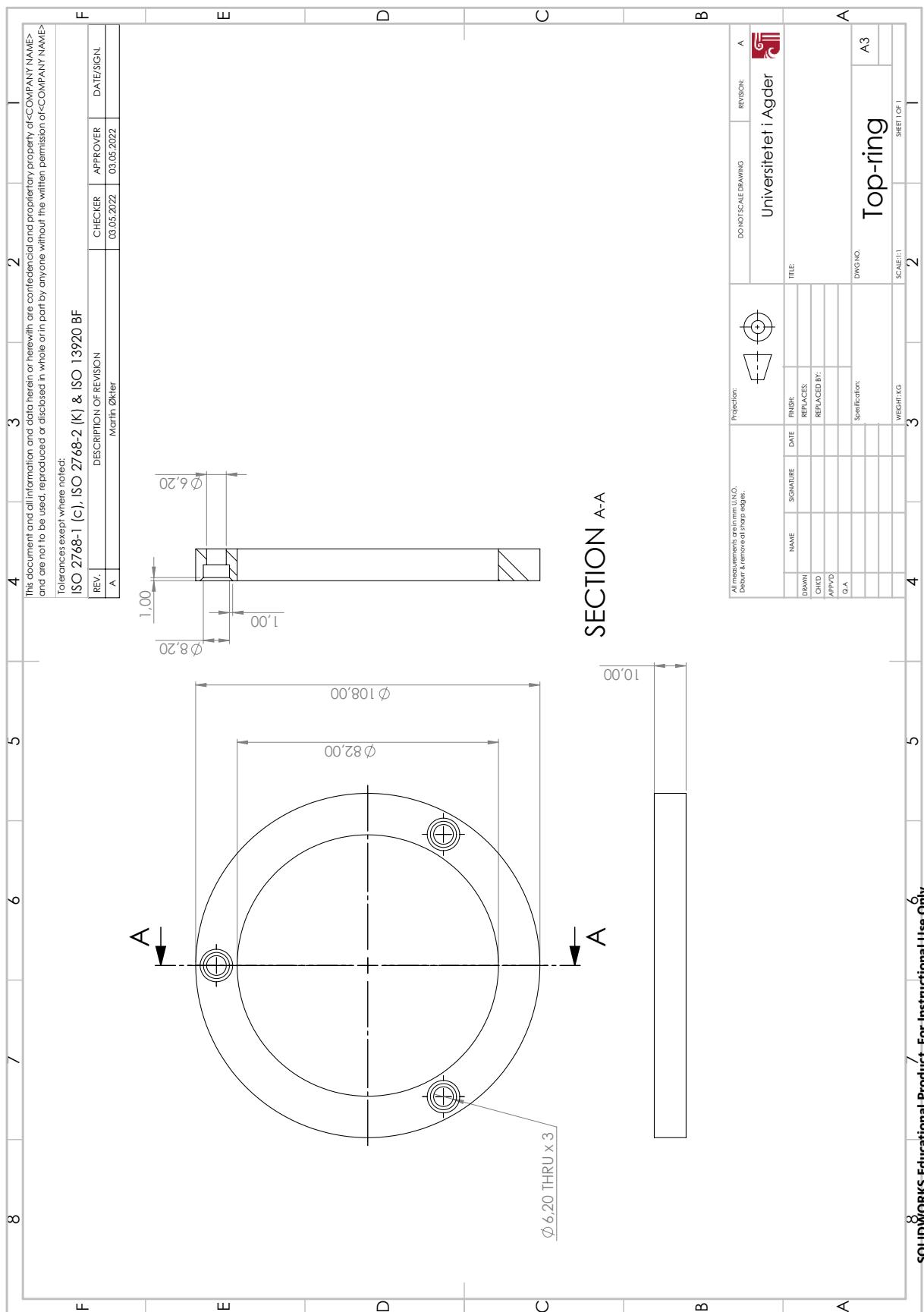
Figur E.2: Teknisk tegning av Actuator platformen assembly



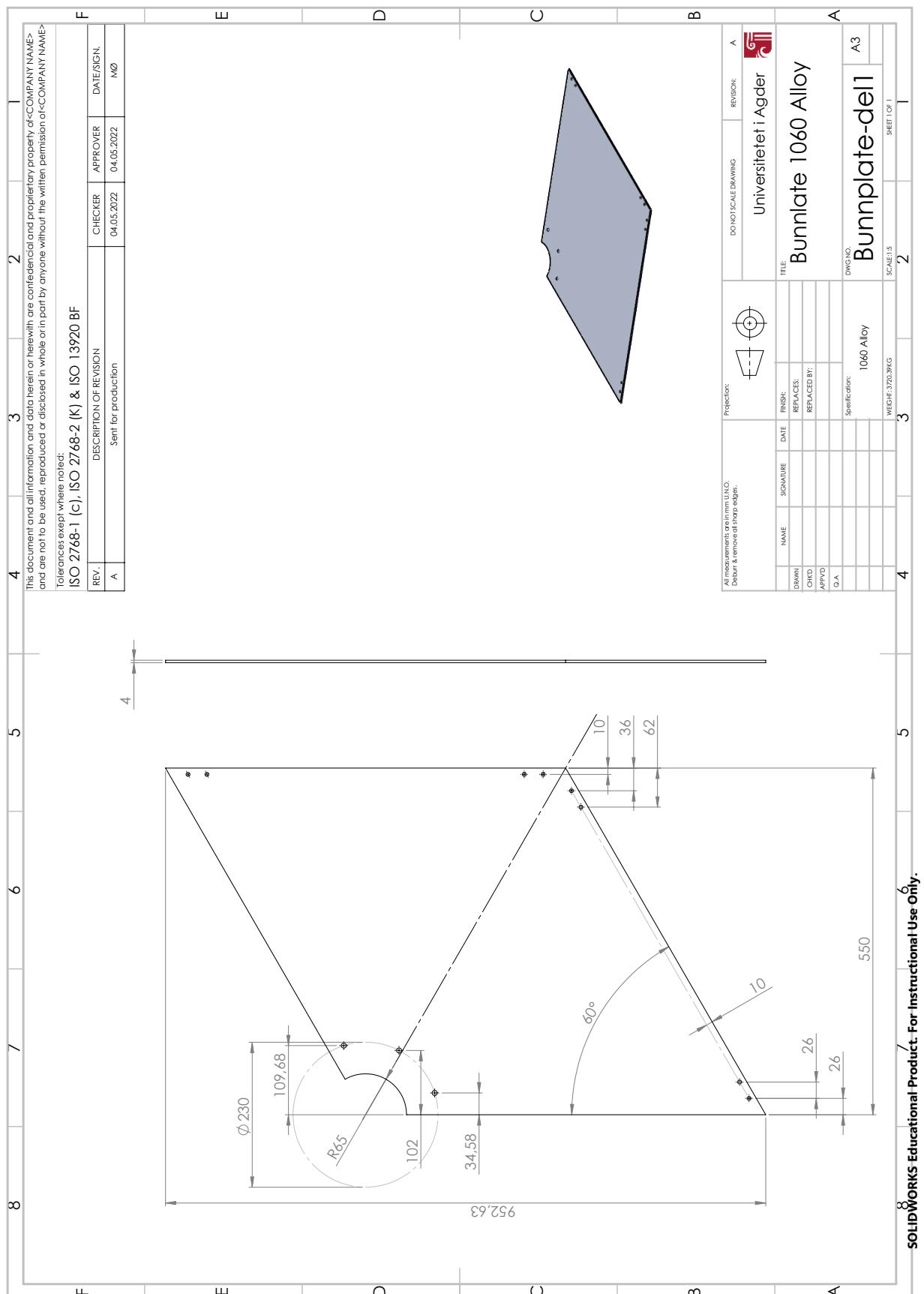
Figur E.3: Teknisk tegning av HjorneV3R



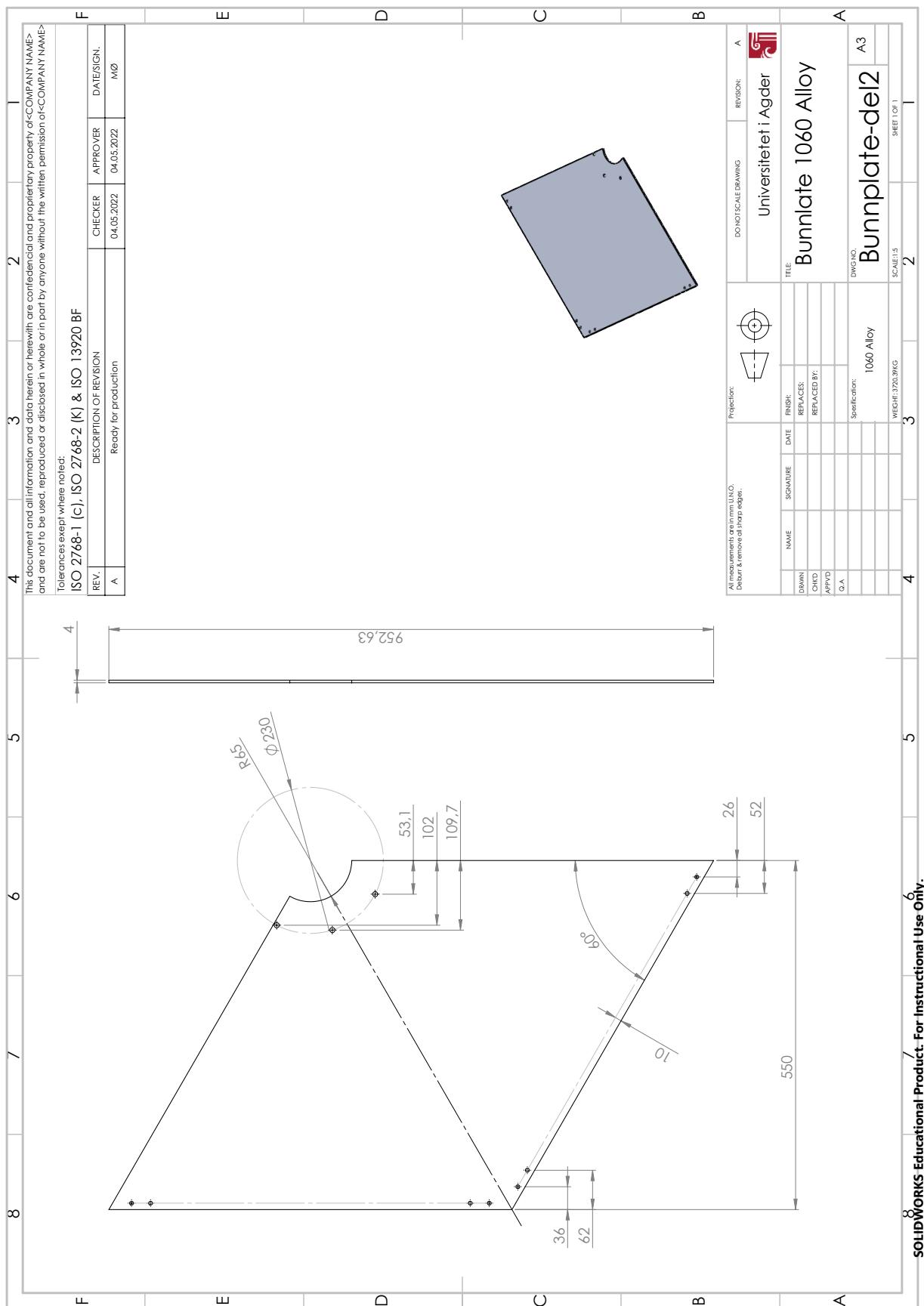
Figur E.4: Teknisk tegning av HjorneV2L



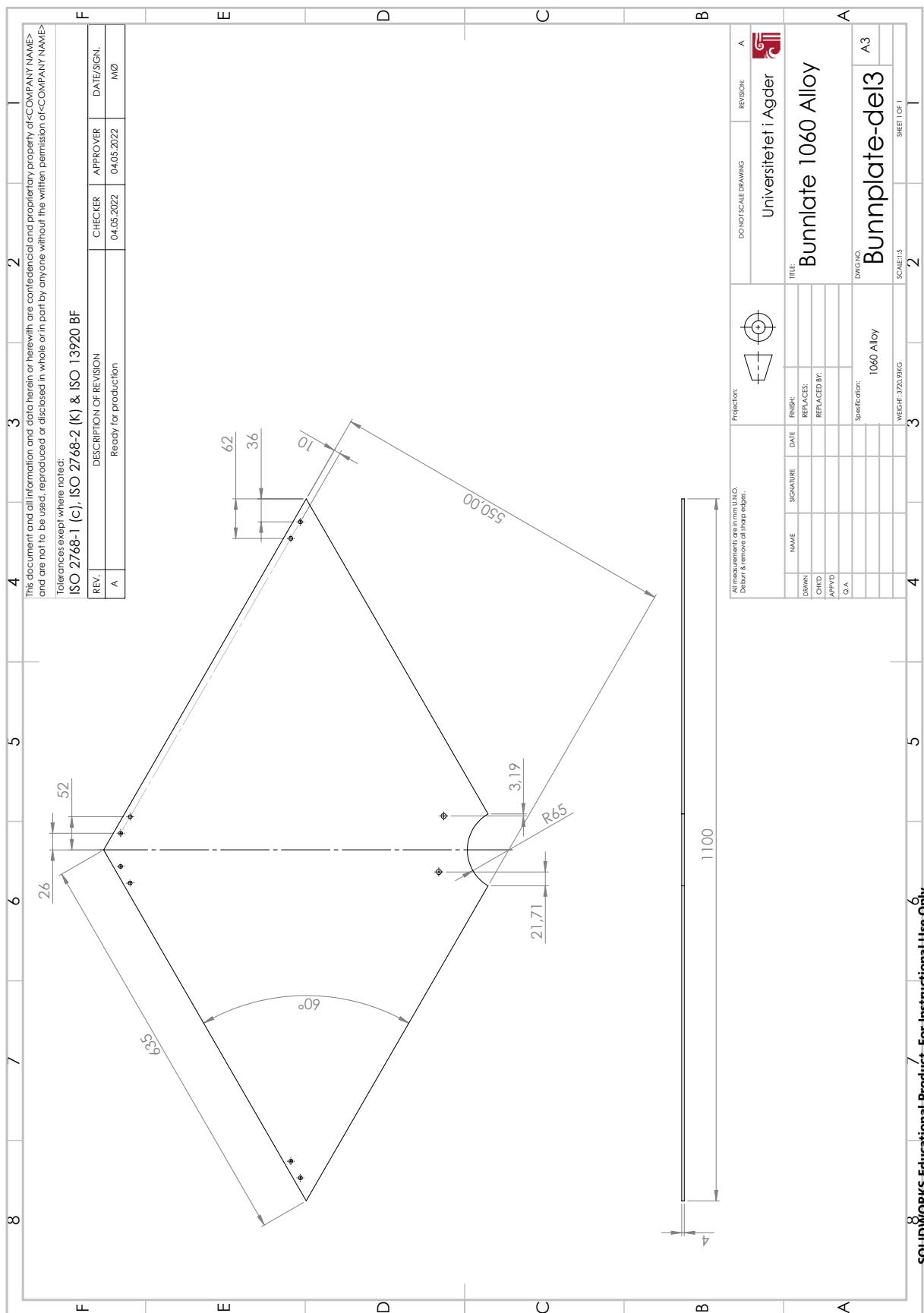
Figur E.5: Teknisk tegning av Top-ring



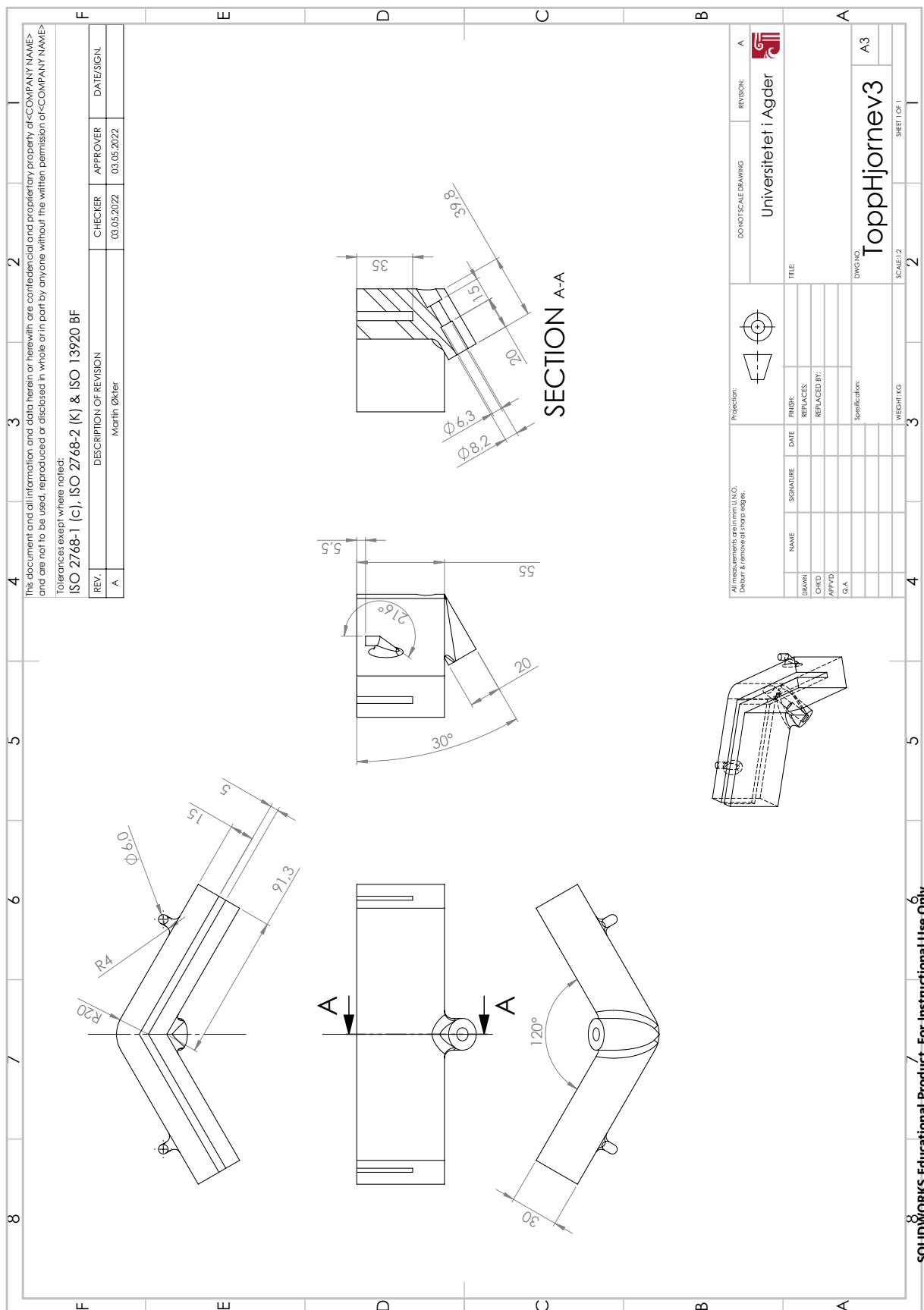
Figur E.6: Teknisk tegning av Bunnplate-del1



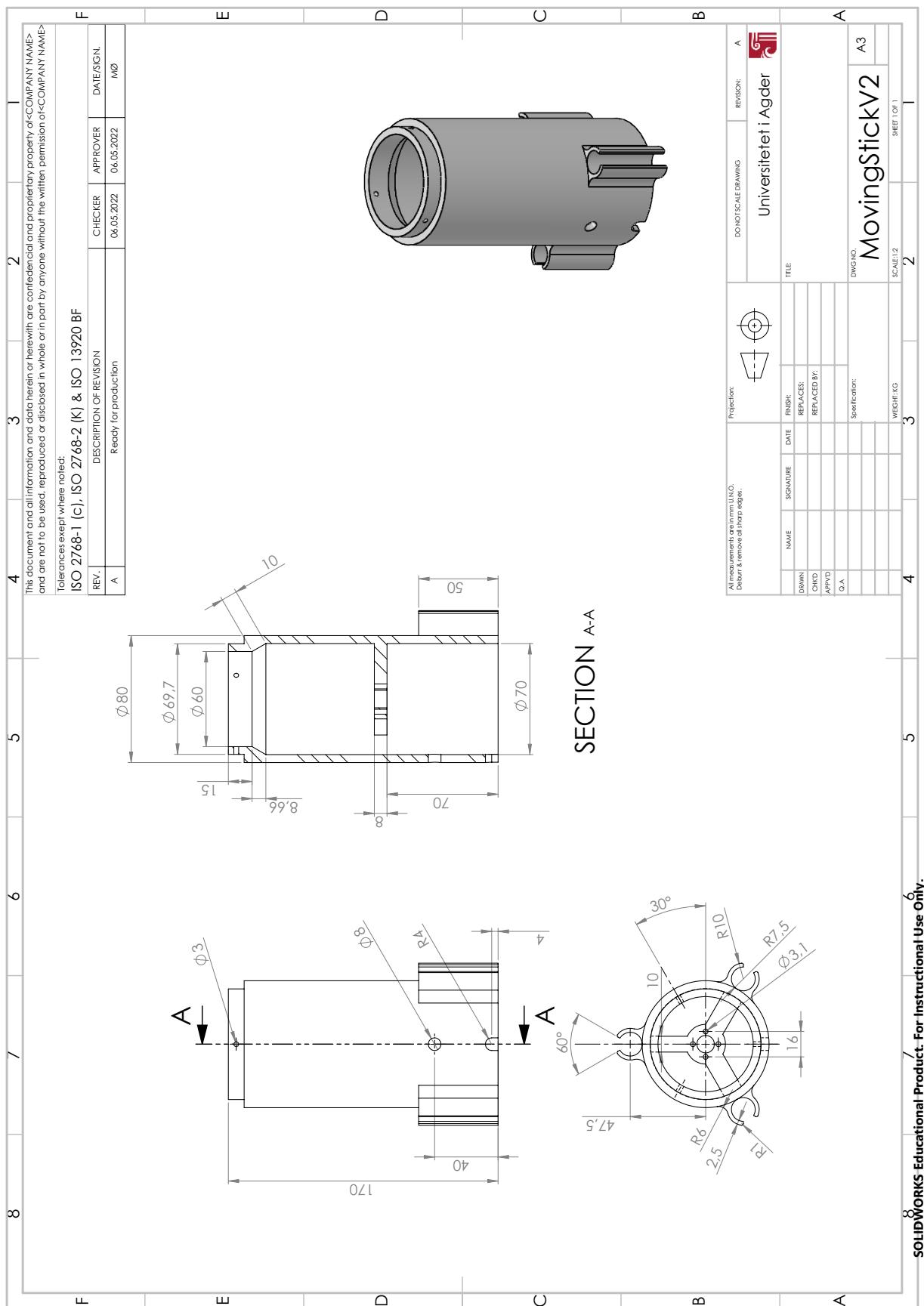
Figur E.7: Teknisk tegning av Bunnplate-del2



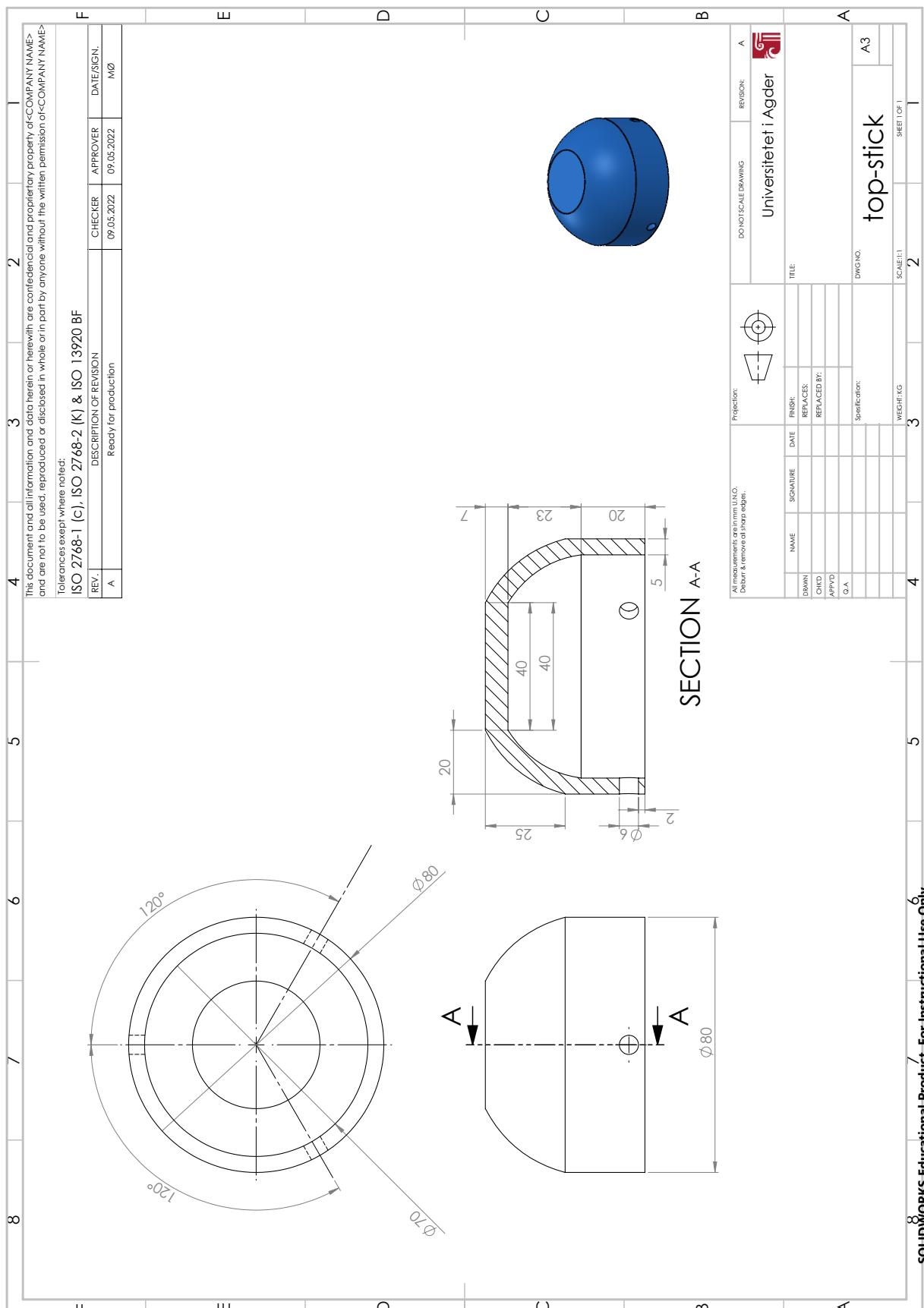
Figur E.8: Teknisk tegning av Bunnplate-del3



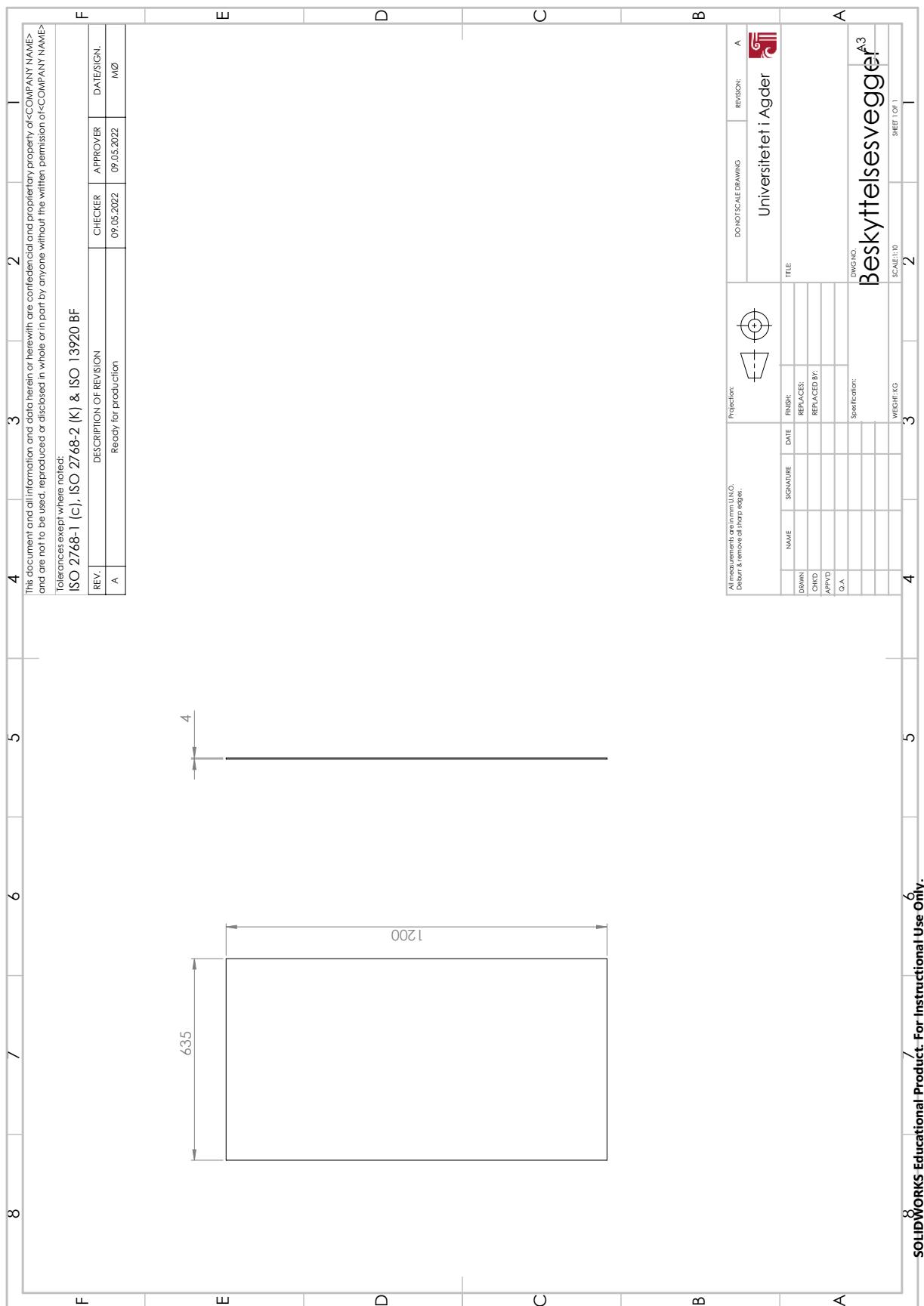
Figur E.9: Teknisk tegning av ToppHjorneV3



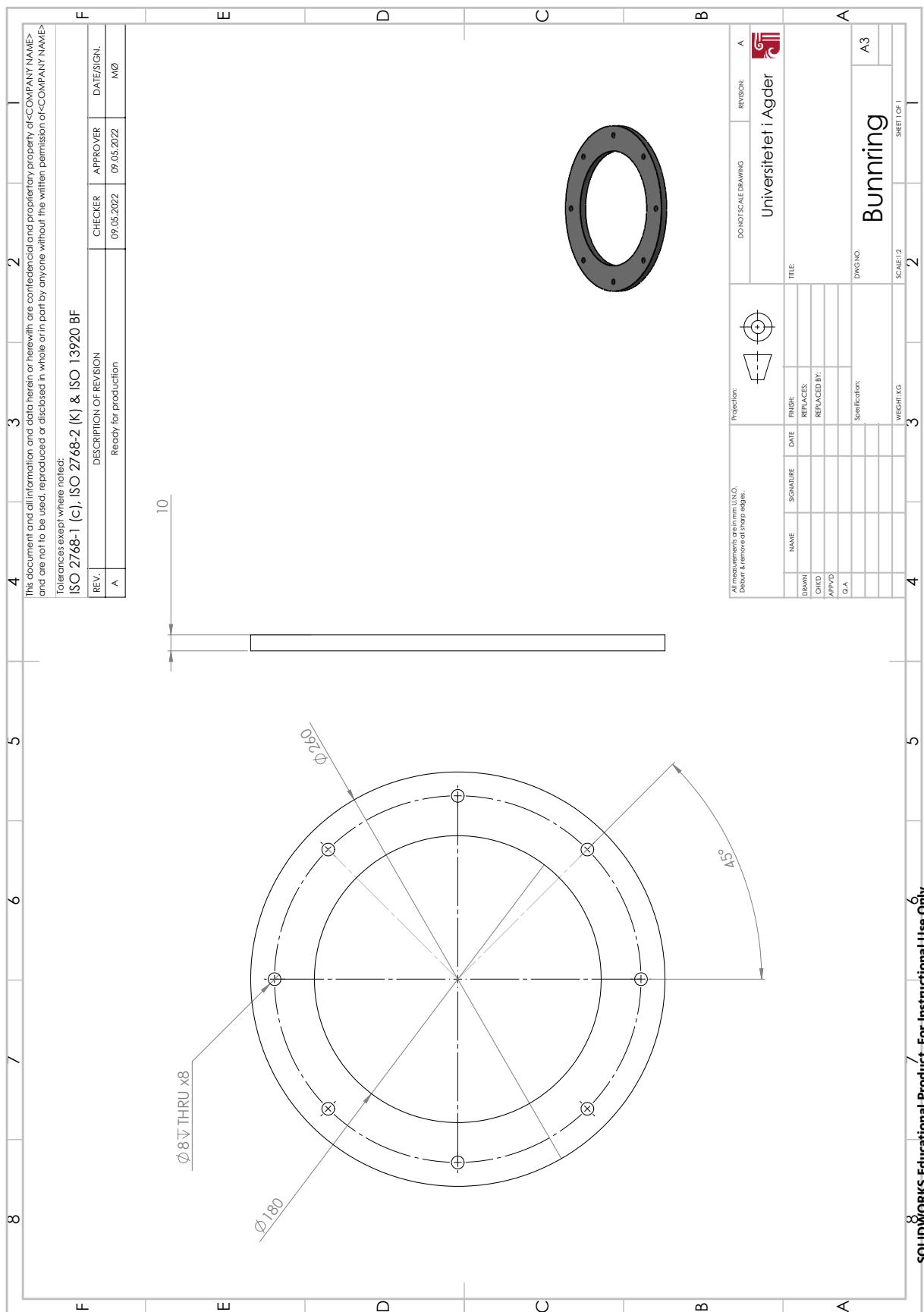
Figur E.10: Teknisk tegning av MovingStickV2



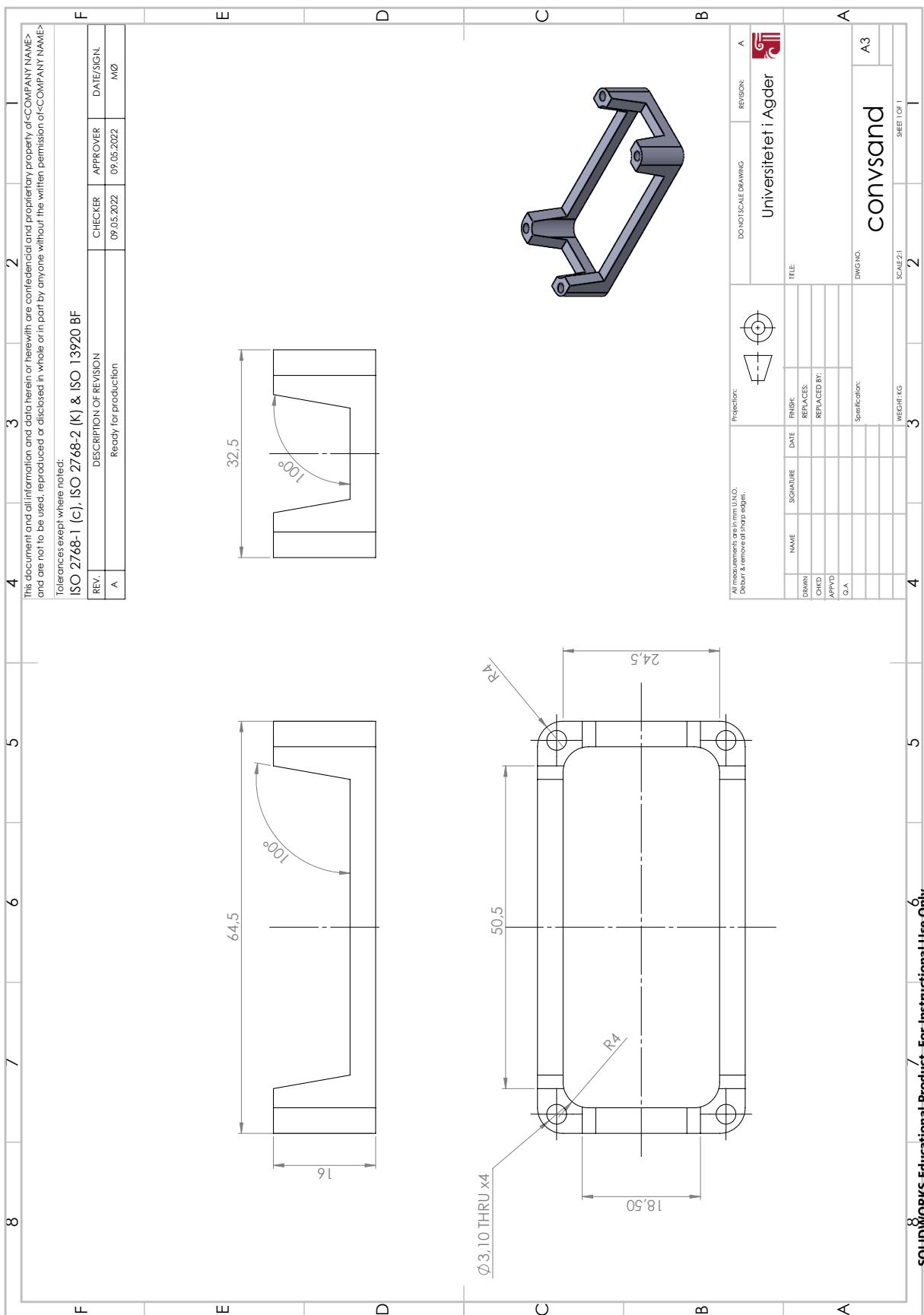
Figur E.11: Teknisk tegning av top-stick



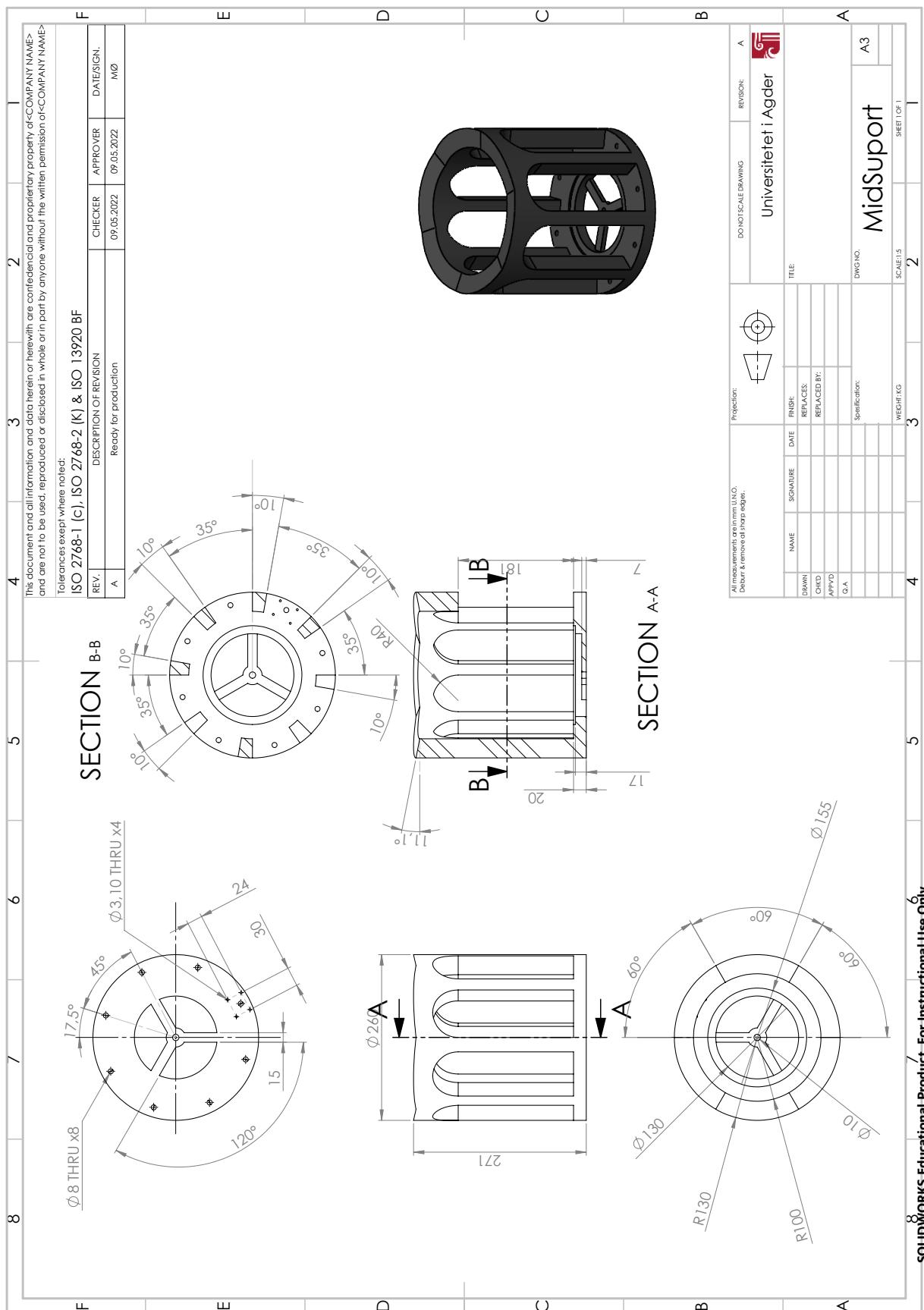
Figur E.12: Teknisk tegning av Beskyttelsesvegger



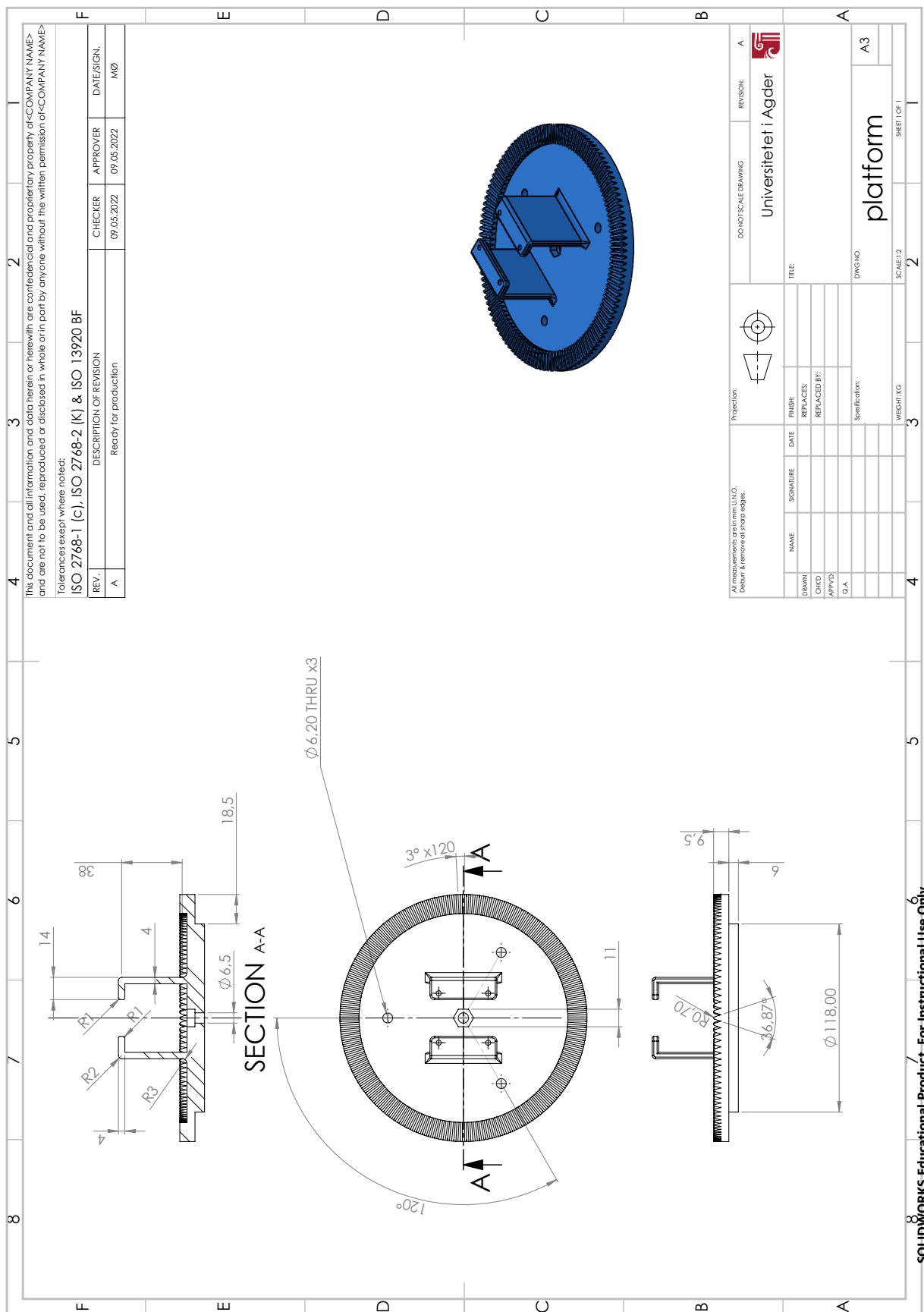
SOLIDWORKS Educational Product. For Instructional Use Only.



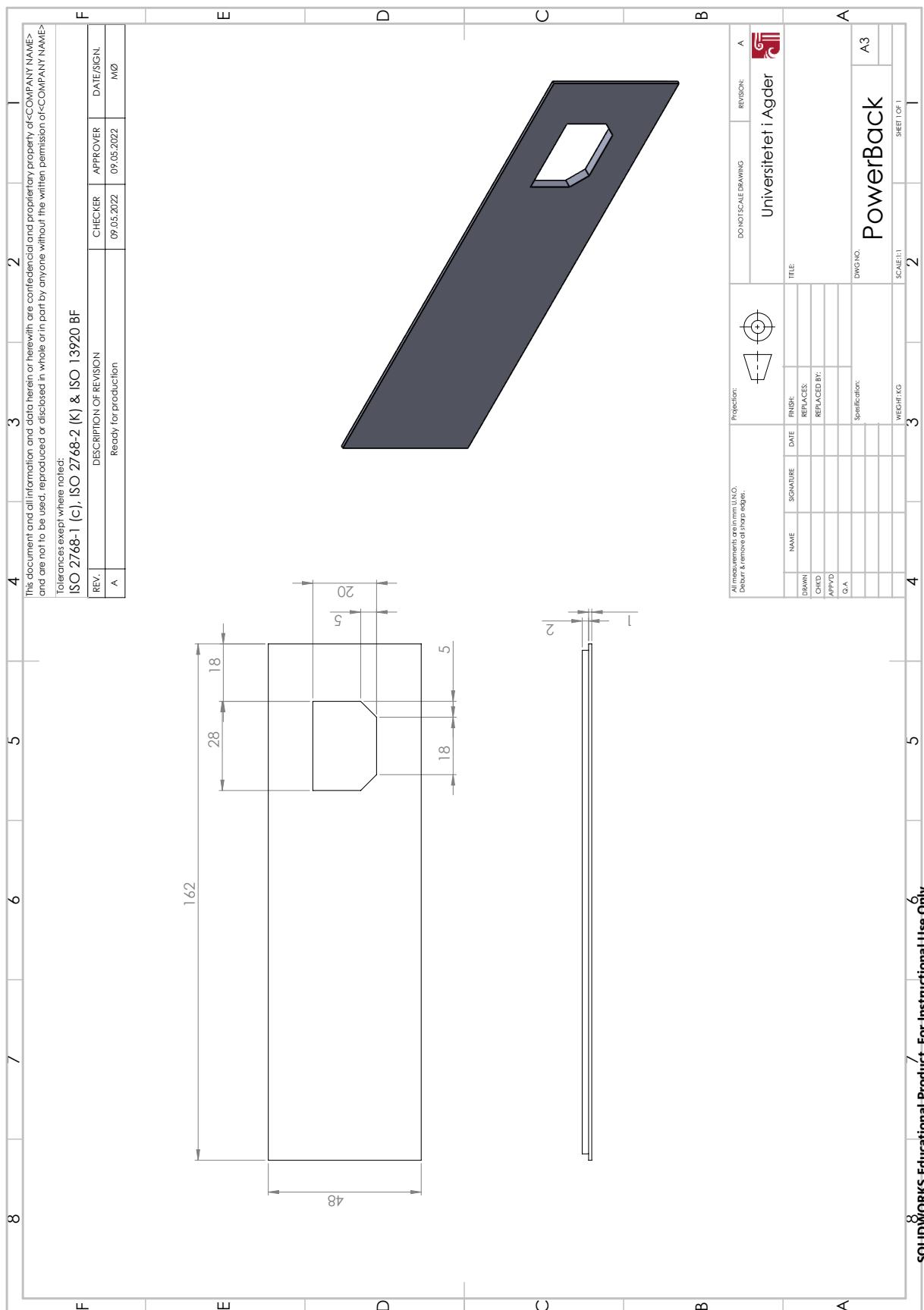
Figur E.14: Teknisk tegning av convstand



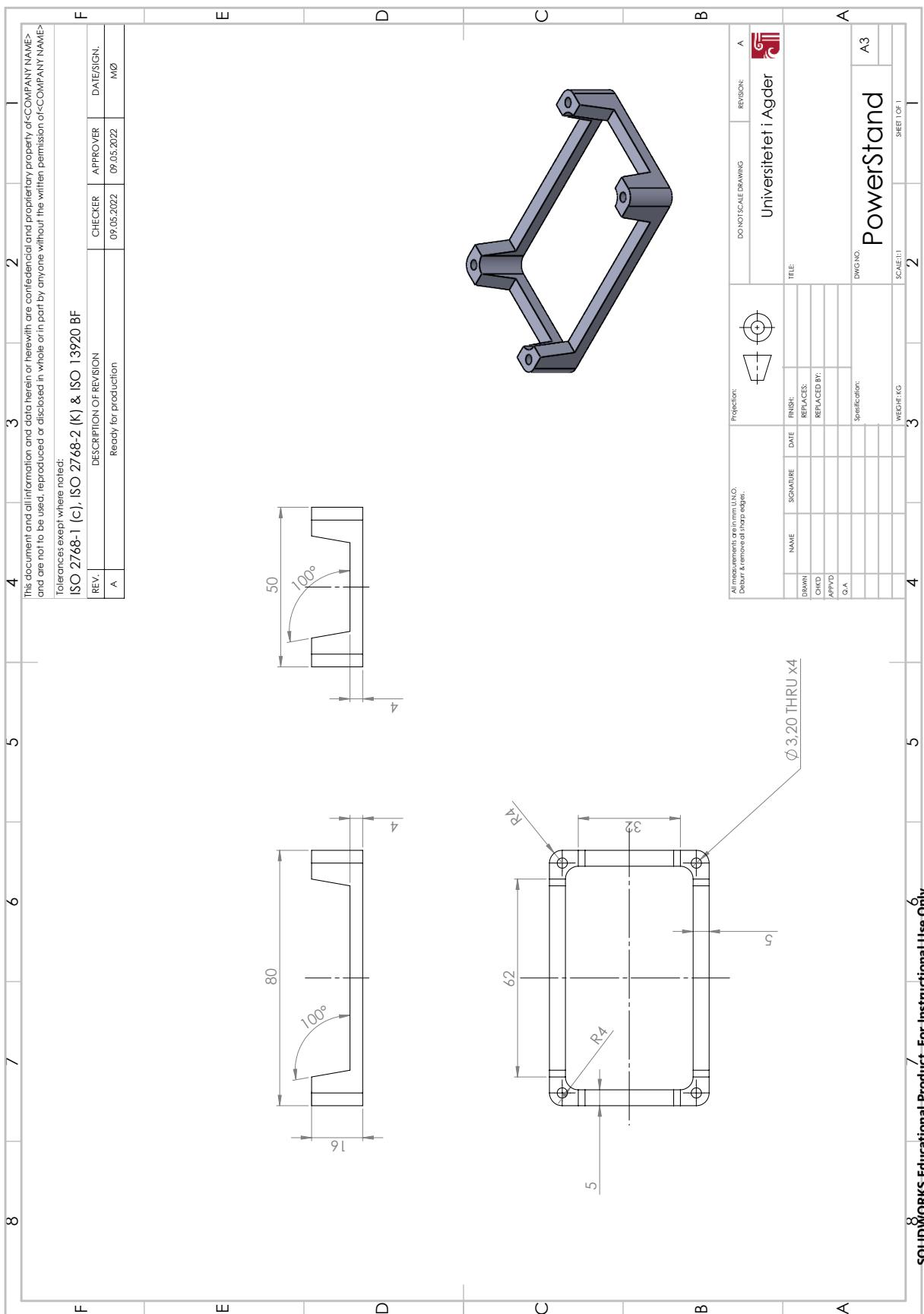
Figur E.15: Teknisk tegning av MidSuport



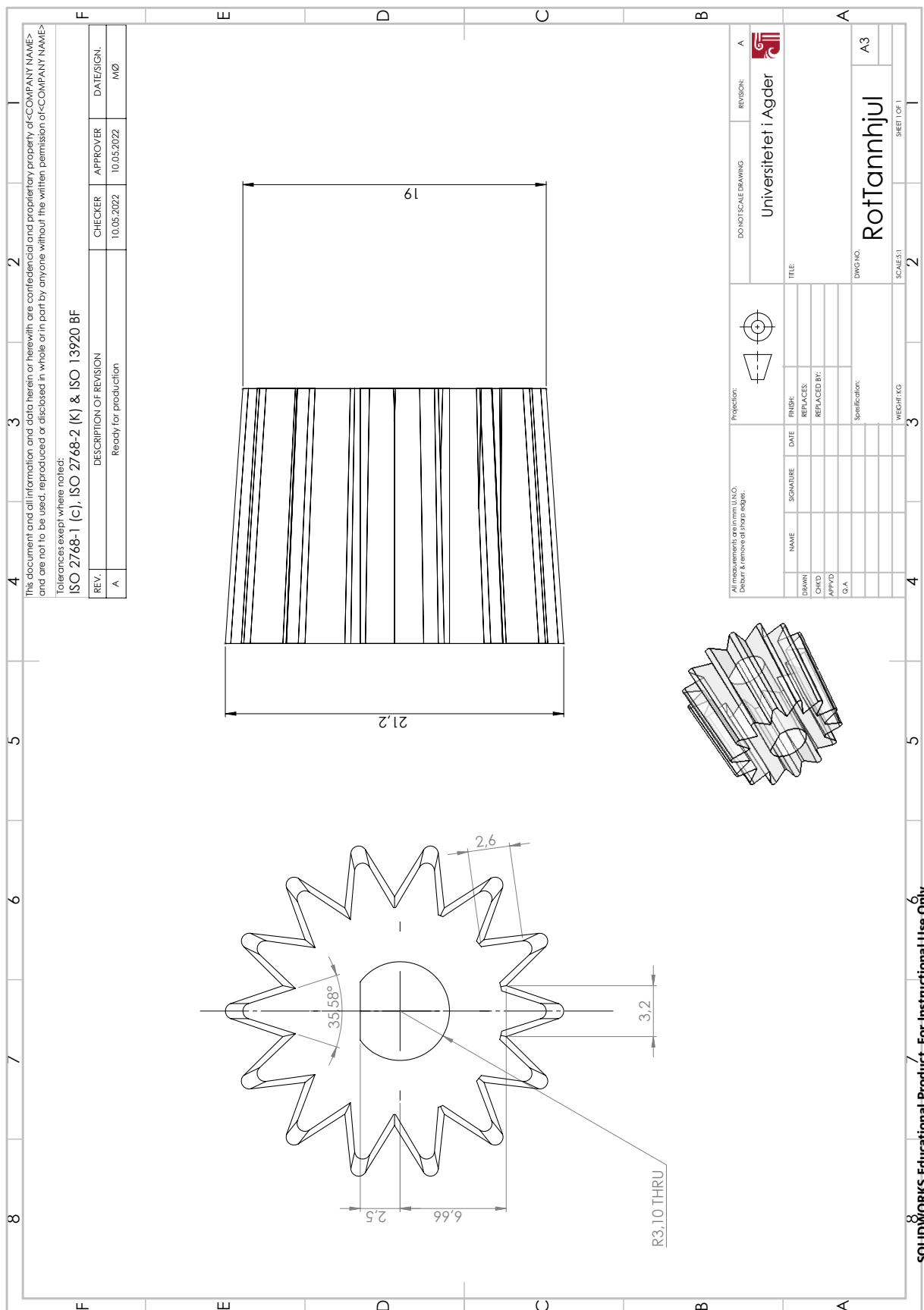
Figur E.16: Teknisk tegning av platform



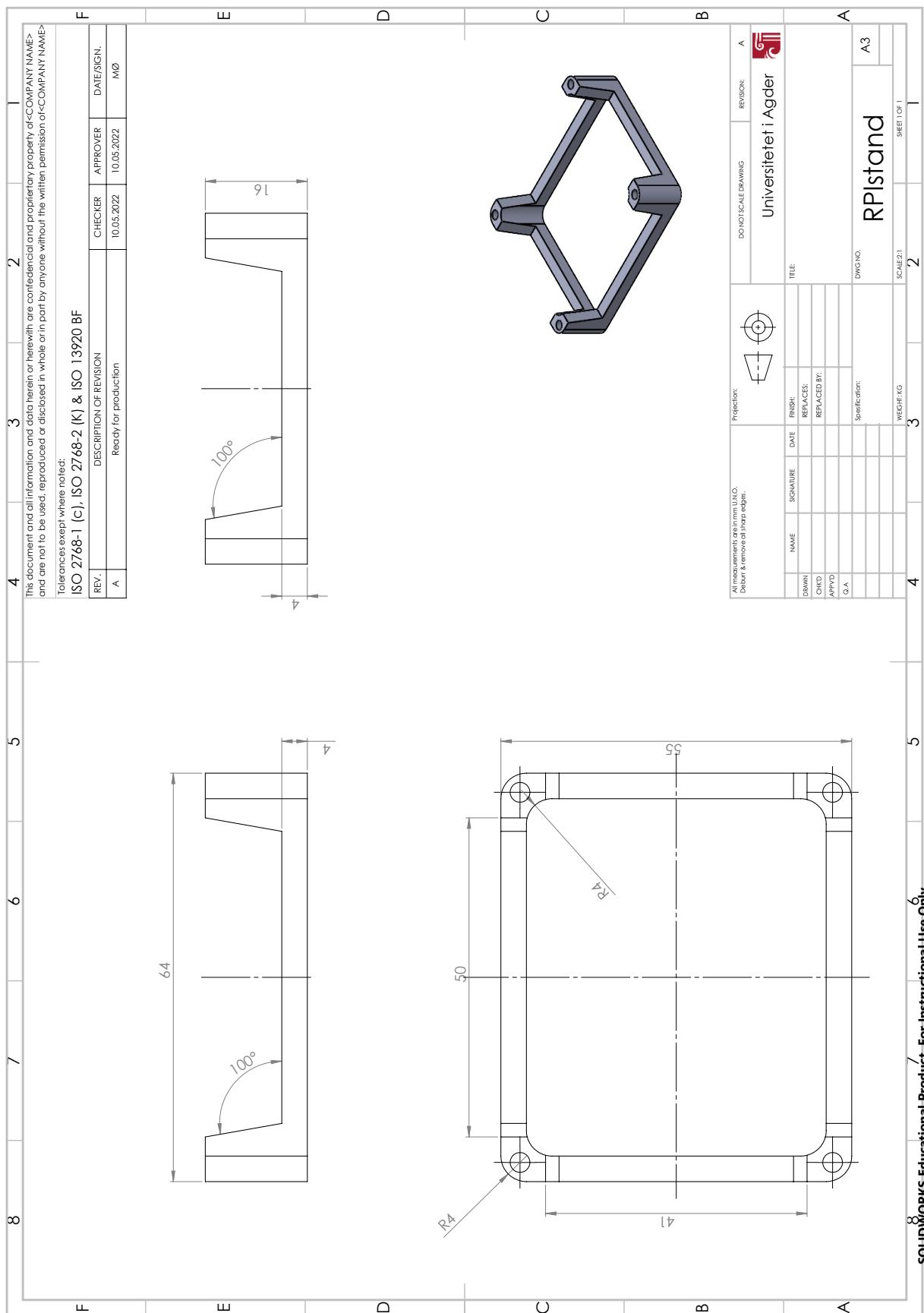
Figur E.17: Teknisk tegning av PowerBack



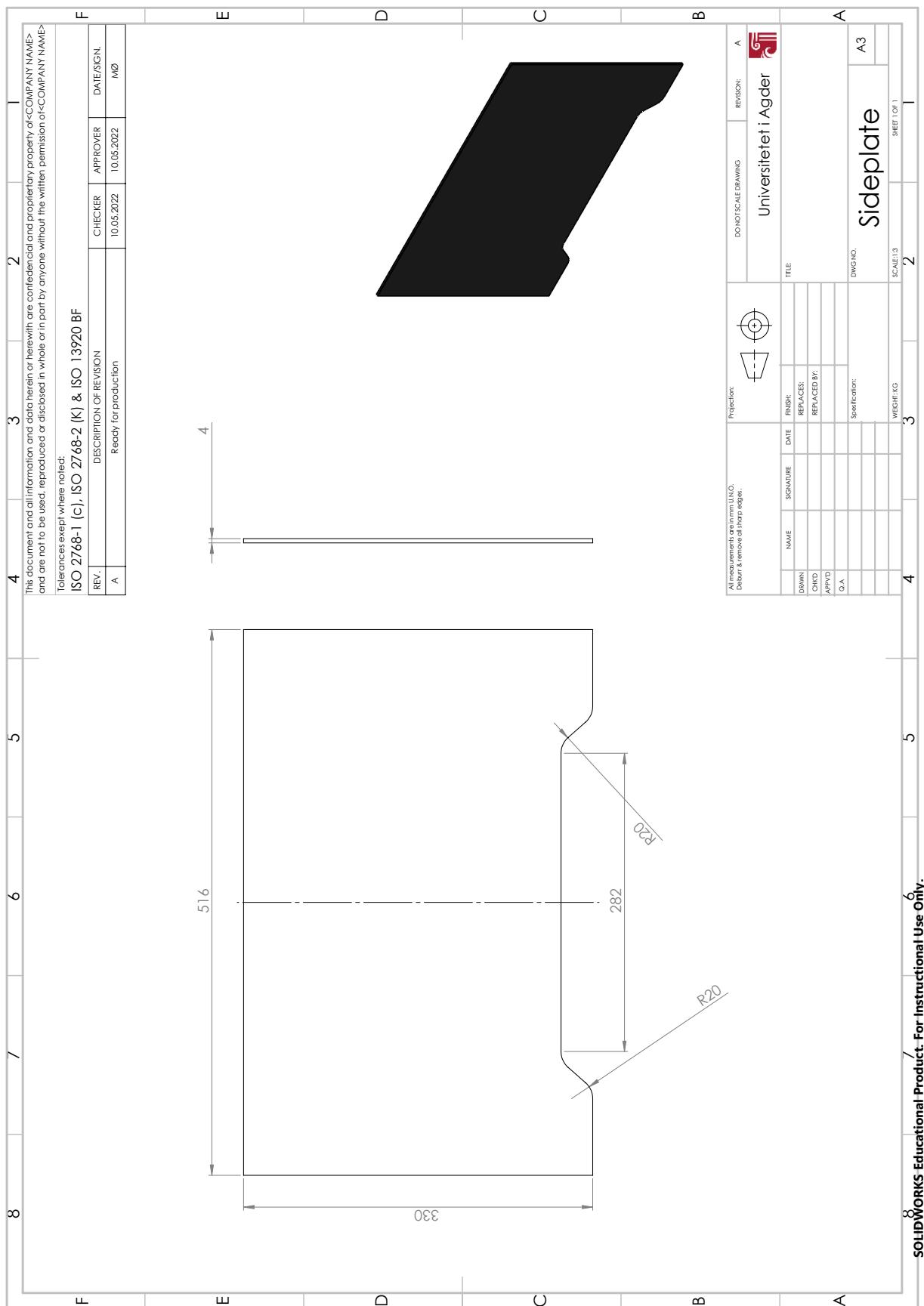
Figur E.18: Teknisk tegning av PowerStand



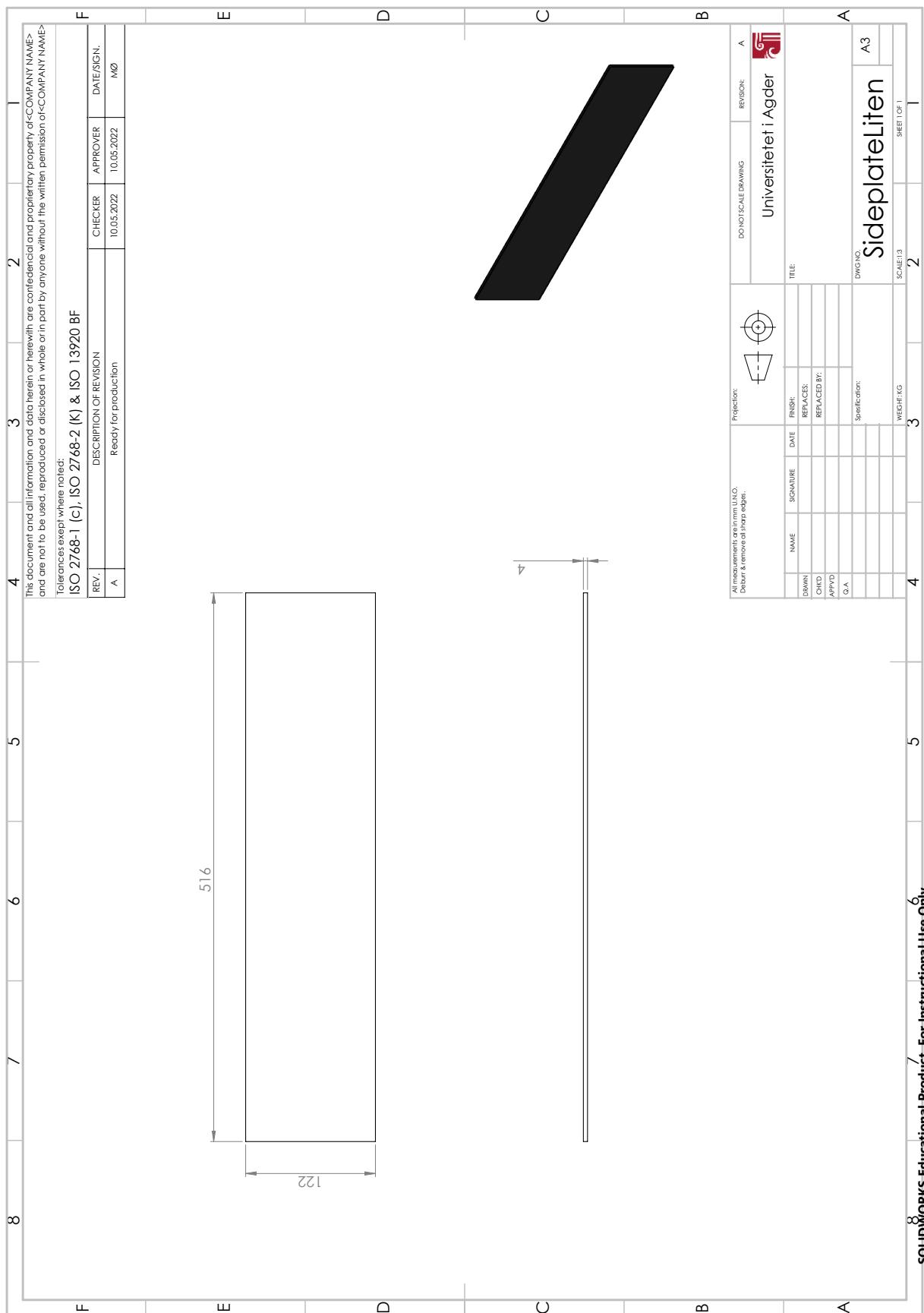
Figur E.19: Teknisk tegning av RotTannhjul



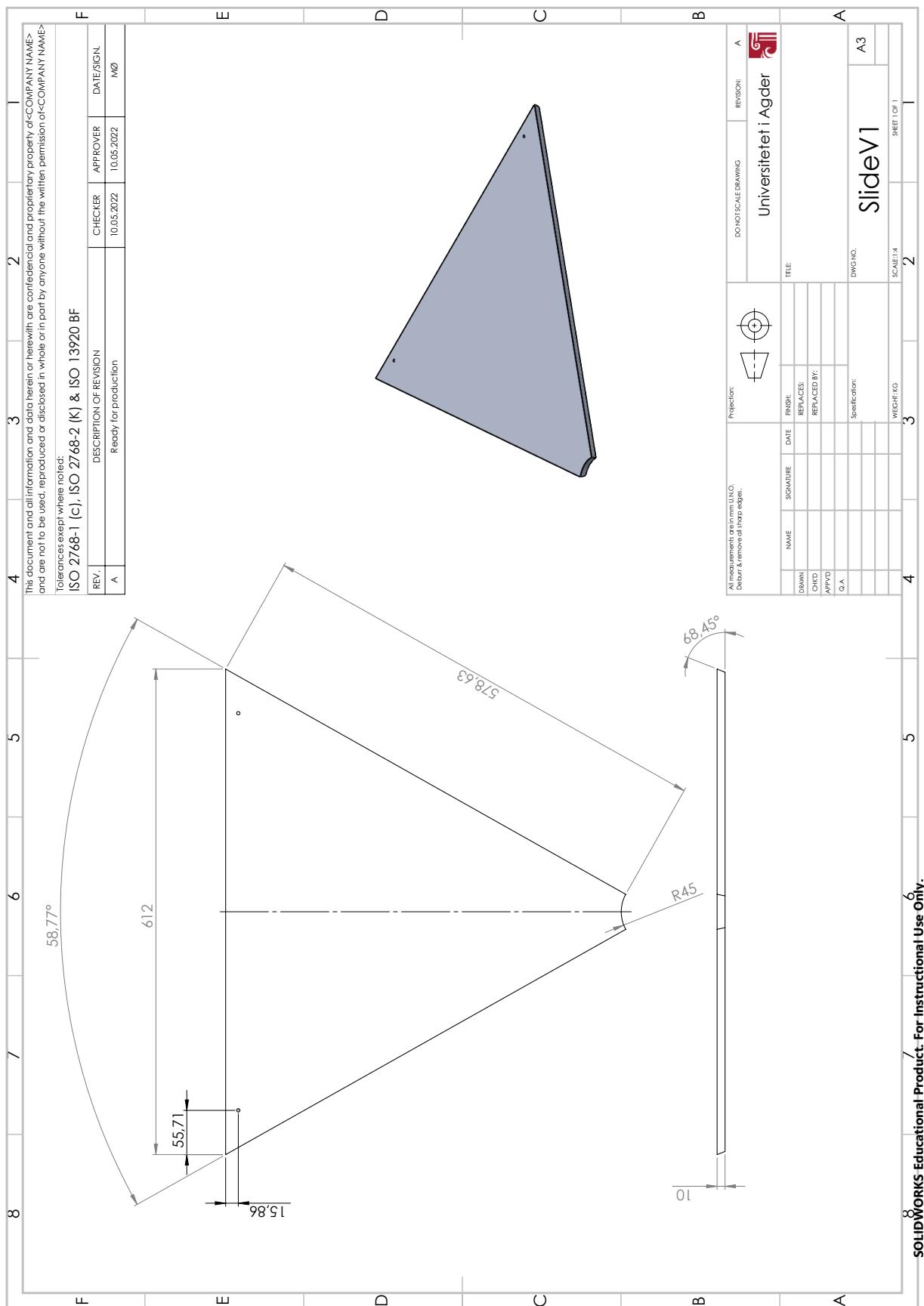
Figur E.20: Teknisk tegning av RPIstand



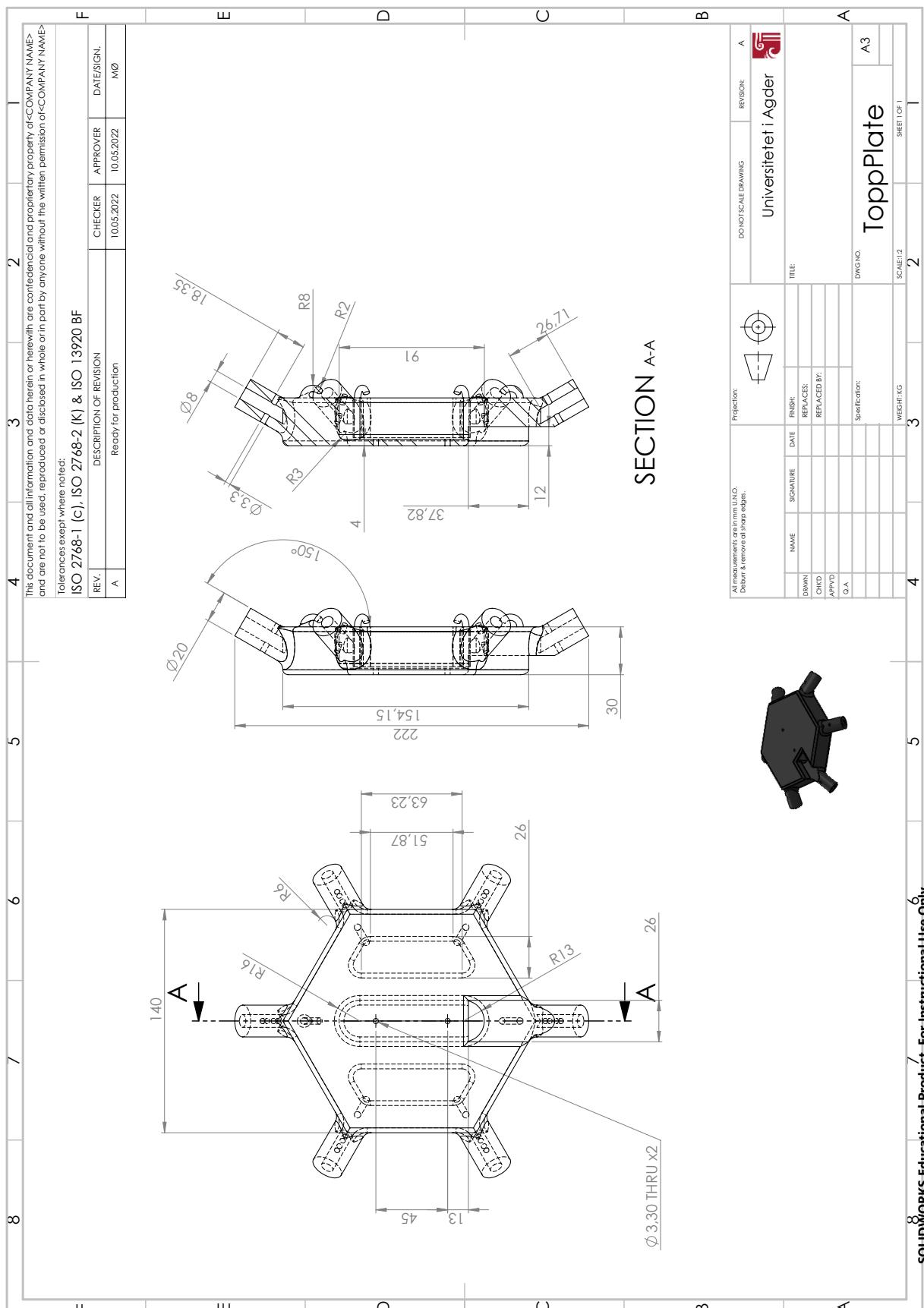
Figur E.21: Teknisk tegning av Sideplate



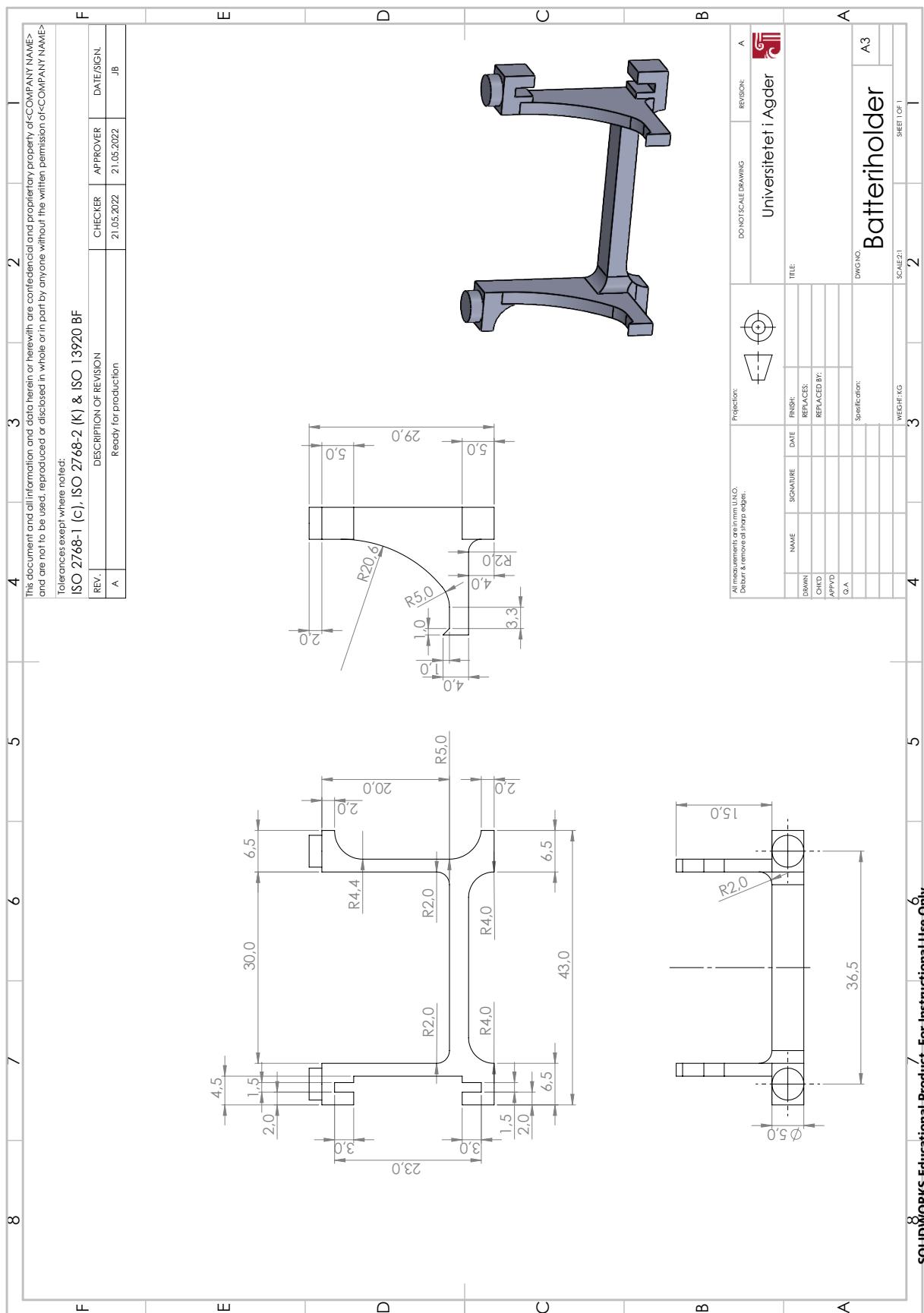
Figur E.22: Teknisk tegning av SideplateLiten



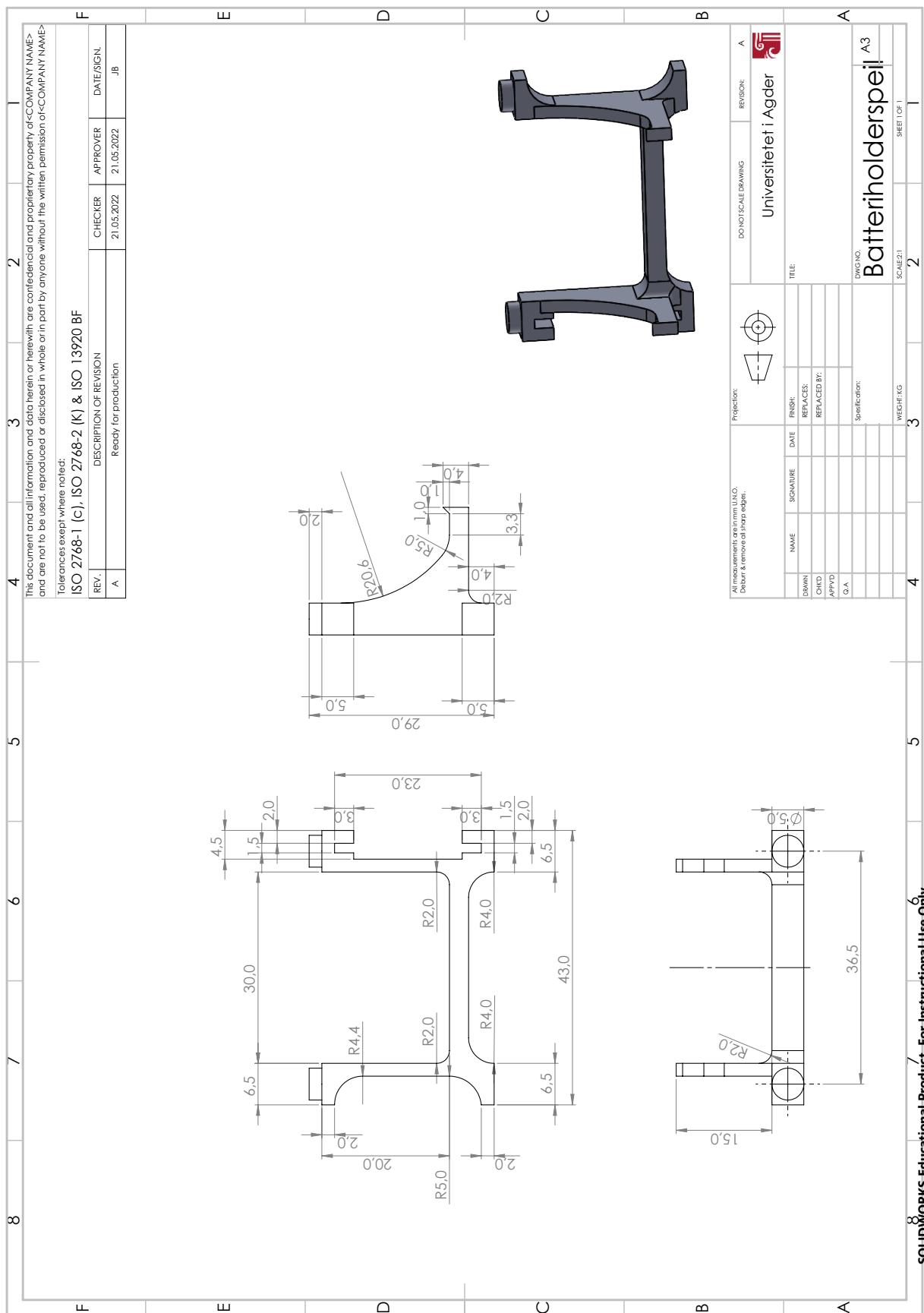
Figur E.23: Teknisk tegning av SlideV1



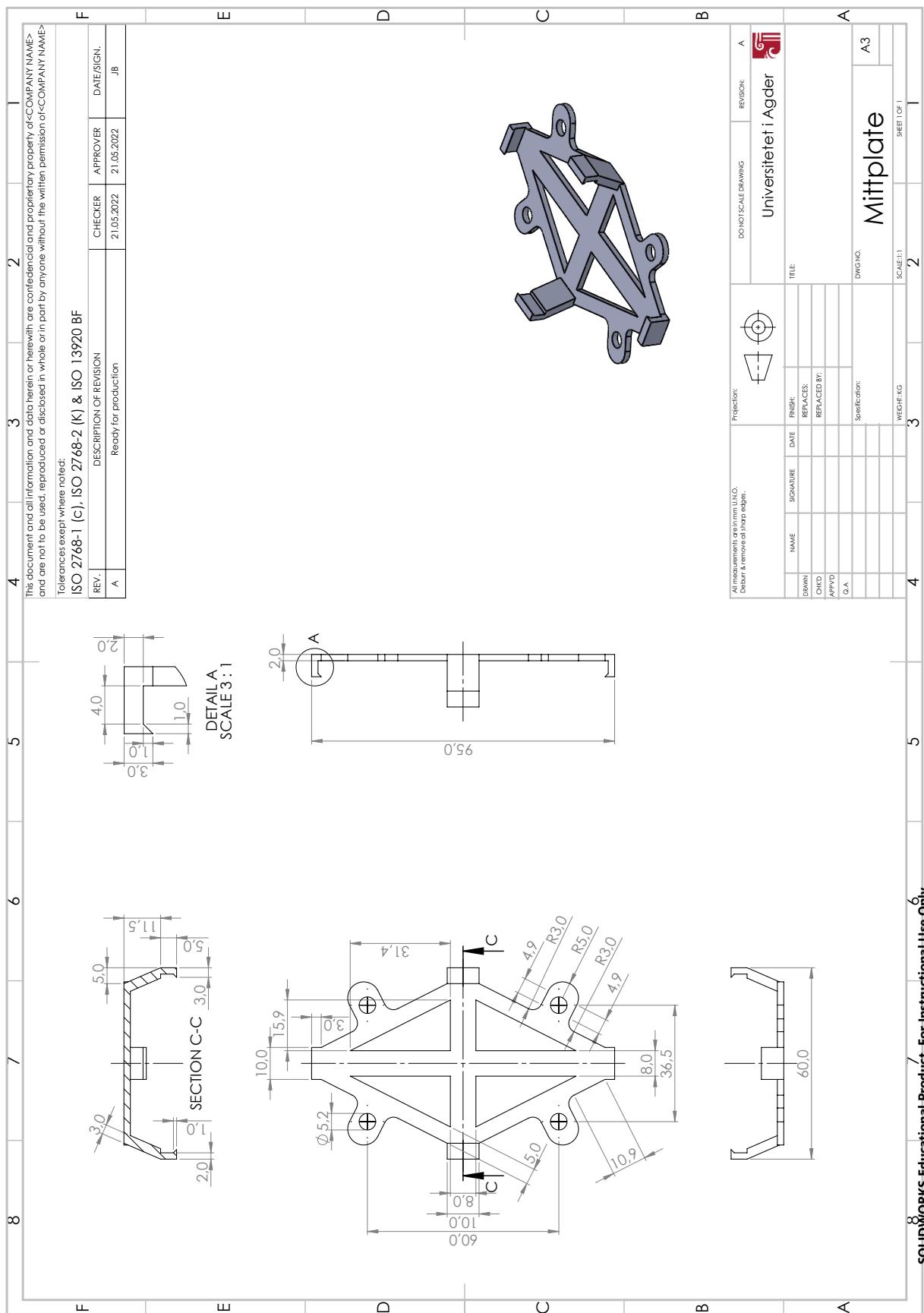
Figur E.24: Teknisk tegning av ToppPlate



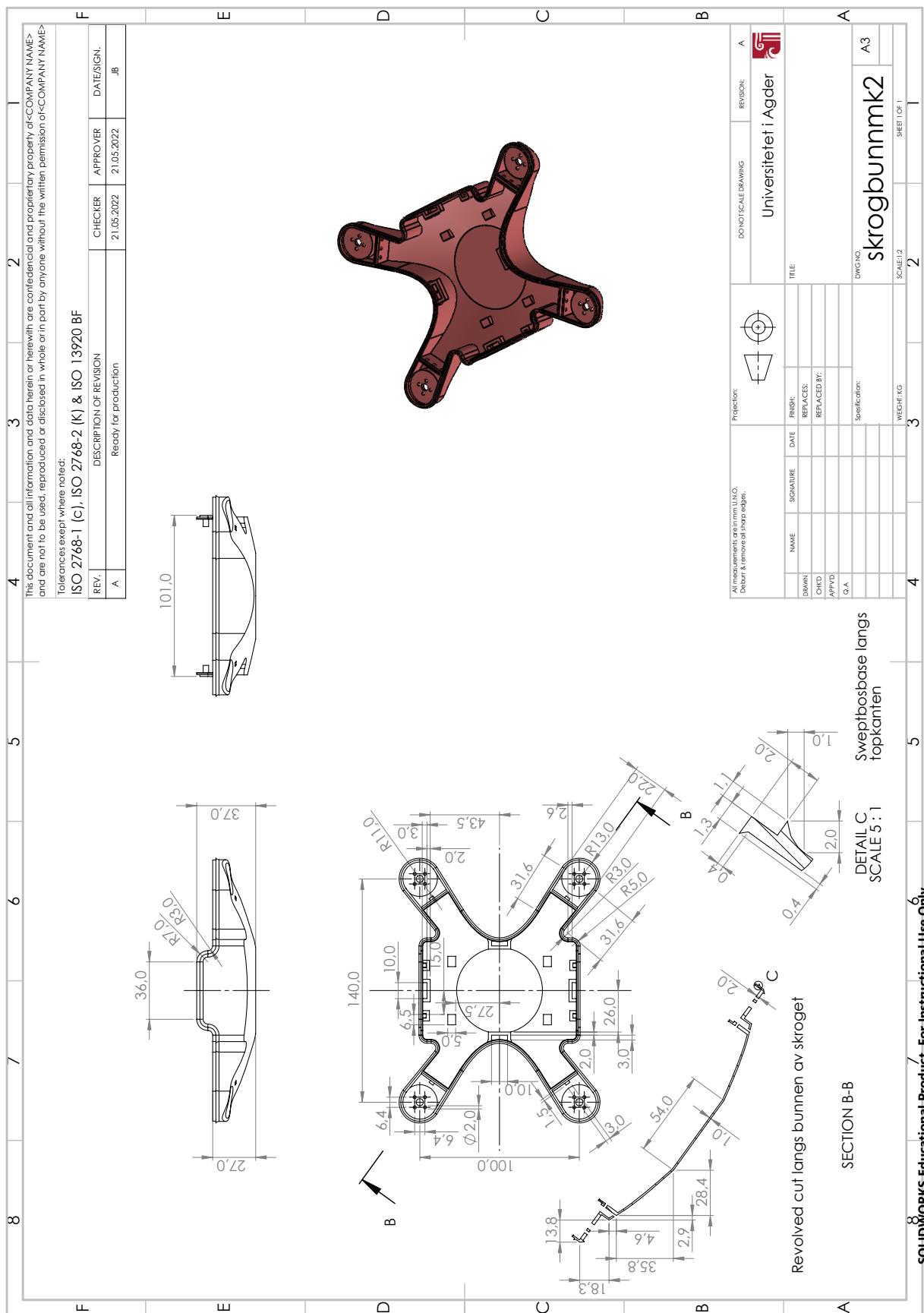
Figur E.25: Teknisk tegning av batteriholder



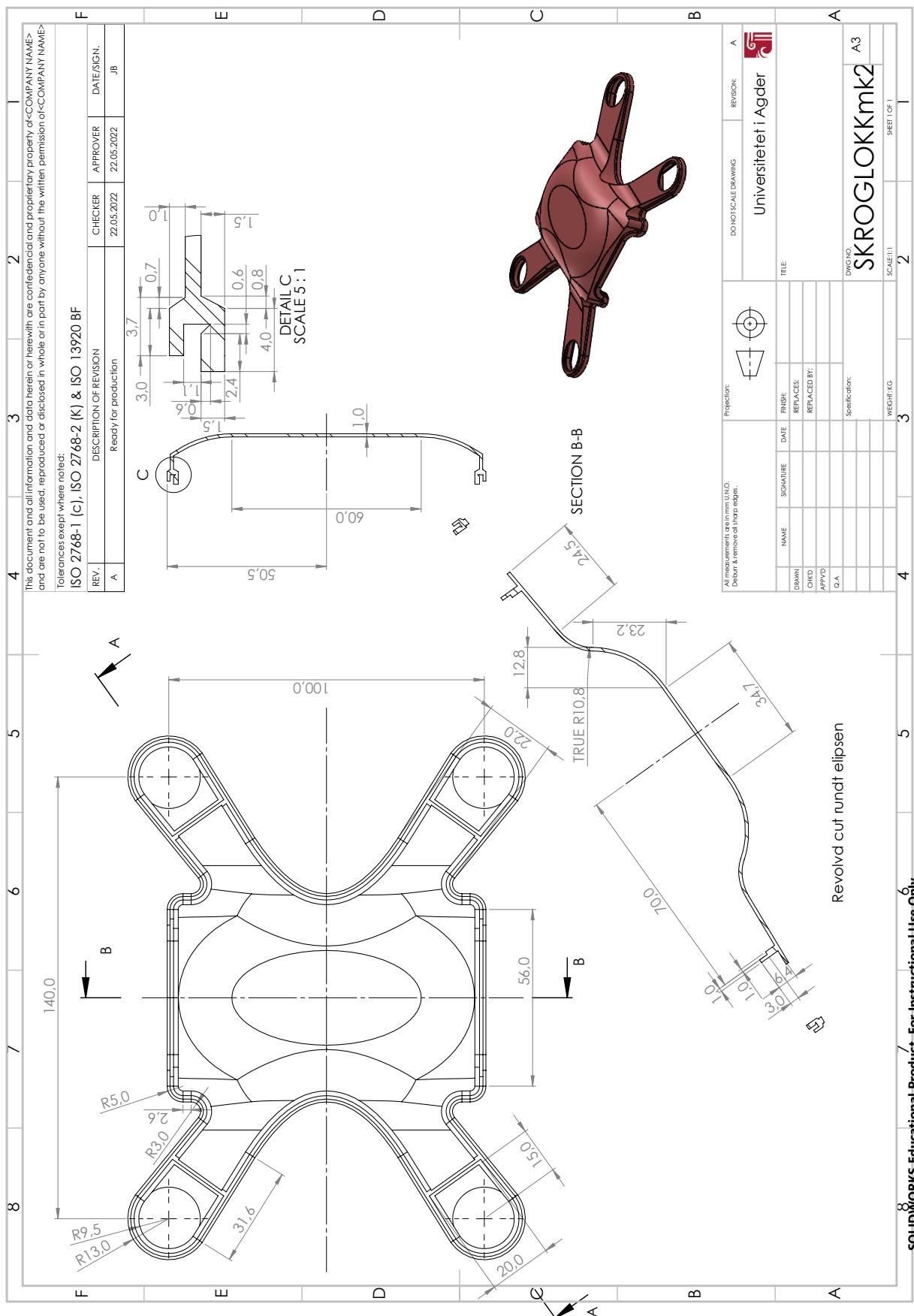
Figur E.26: Teknisk tegning av speilet batteriholder



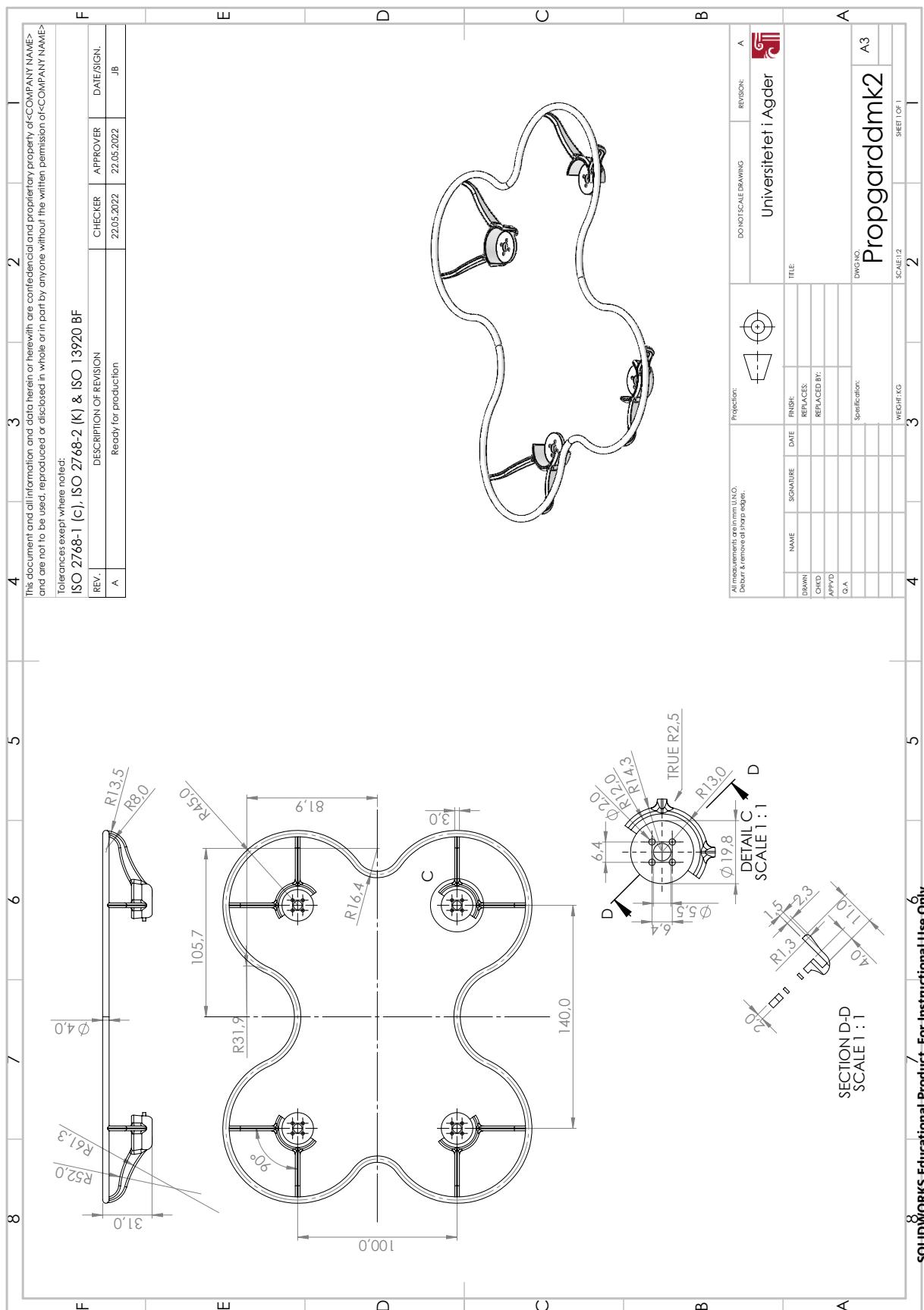
Figur E.27: Teknisk tegning av Midtplate



Figur E.28: Teknisk tegning av skrogbunn



Figur E.29: Teknisk tegning av skroglokk



Figur E.30: Teknisk tegning av propgard