



# NTNU

Kunnskap for en bedre verden

# Bacheloroppgave

IE303612 Automatiseringsteknikk

IP305012 Produkt- og Systemdesign

**Observasjons-ROV til Kystverket**

813, 834, 839, 843, 1011, 1014, 1022

Totalt antall sider inkludert forside: 255

Innlevert Ålesund, 2.Juni 2016

## Obligatorisk egenerklæring/gruppeerklæring

Den enkelte student er selv ansvarlig for å sette seg inn i hva som er lovlige hjelpemidler, retningslinjer for bruk av disse og regler om kildebruk. Erklæringen skal bevisstgjøre studentene på deres ansvar og hvilke konsekvenser fusk kan medføre. **Manglende erklæring fritar ikke studentene fra sitt ansvar.**

Du/dere fyller ut erklæringen ved å klikke i ruten til høyre for den enkelte del 1-6:		
1.	Jeg/vi erklærer herved at min/vår besvarelse er mitt/vårt eget arbeid, og at jeg/vi ikke har brukt andre kilder eller har mottatt annen hjelp enn det som er nevnt i besvarelsen.	<input checked="" type="checkbox"/>
2.	Jeg/vi erklærer videre at denne besvarelsen: <ul style="list-style-type: none"><li>• ikke har vært brukt til annen eksamen ved annen avdeling/universitet/høgskole innenlands eller utenlands.</li><li>• ikke refererer til andres arbeid uten at det er oppgitt.</li><li>• ikke refererer til eget tidligere arbeid uten at det er oppgitt.</li><li>• har alle referansene oppgitt i litteraturlisten.</li><li>• ikke er en kopi, duplikat eller avskrift av andres arbeid eller besvarelse.</li></ul>	<input checked="" type="checkbox"/>
3.	Jeg/vi er kjent med at brudd på ovennevnte er å <u>betrakte som fusk</u> og kan medføre annullering av eksamen og utestengelse fra universiteter og høgskoler i Norge, jf. <a href="#">Universitets- og høgskoleloven</a> §§4-7 og 4-8 og Forskrift om eksamen.	<input checked="" type="checkbox"/>
4.	Jeg/vi er kjent med at alle innleverte oppgaver kan bli plagiatkontrollert i Ephorus, se Retningslinjer for elektronisk innlevering og publisering av studiepoenggivende studentoppgaver	<input checked="" type="checkbox"/>
5.	Jeg/vi er kjent med at høgskolen vil behandle alle saker hvor det forligger mistanke om fusk etter NTNUs studieforskrift.	<input checked="" type="checkbox"/>
6.	Jeg/vi har satt oss inn i regler og retningslinjer i bruk av kilder og referanser på biblioteket sine nettsider	<input checked="" type="checkbox"/>



# Publiseringsavtale

Studiepoeng: 20

Veiledere: Ottar Osen, Robin Bye og Paul Steffen Kleppe

## Fullmakt til elektronisk publisering av oppgaven

Forfatter(ne) har opphavsrett til oppgaven. Det betyr blant annet enerett til å gjøre verket tilgjengelig for allmennheten ([Åndsverkloven §2](#)).

Alle oppgaver som fyller kriteriene vil bli registrert og publisert i Brage med forfatter(ne)s godkjenning.

Oppgaver som er unntatt offentlighet eller båndlagt vil ikke bli publisert.

Jeg/vi gir herved NTNU i Ålesund en vederlagsfri rett til å gjøre oppgaven tilgjengelig for elektronisk publisering:

ja  nei

Er oppgaven båndlagt (konfidensiell)?

ja  nei

(Båndleggingsavtale må fylles ut)

- Hvis ja:

Kan oppgaven publiseres når båndleggingsperioden er over?

ja  nei

Er oppgaven unntatt offentlighet?

ja  nei

(inneholder taushetsbelagt informasjon. [Jfr. Offl. §13/Fvl. §13](#))

Dato: 10.02.2016

# HOVEDPROSJEKT

**TITTEL:**

Observasjons-ROV til Kystverket

**KANDIDATNUMMER(E):**

834 843 839 813 1014 1022 1011

<b>DATO:</b> 03.06.16	<b>EMNEKODE:</b> IP305012 IE303612	<b>EMNE:</b> Bacheloroppgave	<b>DOKUMENT TILGANG:</b> - Åpen
<b>STUDIUM:</b> AUTOMATISERINGSTEKNIKK PRODUKT- OG SYSTEMDESIGN		<b>ANT SIDER/VEDLEGG:</b> 161 / 91 og 1 CD	<b>BIBL. NR:</b> ▪ Ikke i bruk -

**OPPDRAKSGIVER(E)/VEILEDER(E):**

Robin Bye, Paul Steffen Kleppe, Ottar L. Osen, Per Erik Ose

**SAMMENDRAG:**

Hensikten med denne oppgaven er å utvikle et konsept til en kompakt og bærbar observasjons-ROV som kan bistå Kystverket med å inspisere sjømerker. I dag brukes dykkere, senkekamera og undervannskikkert. Kystverket leier inn eksternt personell til å utføre dykkene, og dykkerne er avhengige av godt vær for å utføre inspeksjonen. Ved å investere i en rimelig observasjons-ROV kan inspeksjon gjennomføres på en mer effektiv og fleksibel måte, samtidig som dokumentasjon på tilstanden til sjømerkene forbedres.

Prosjektgruppa har skissert et forslag til en observasjons-ROV. ROV-en har et hydrodynamisk skrog med seks bein til å stå stødig på underlaget under filming. ROV-en kan levere rotasjonskorrigert live videostrøm som utføres i OpenCV, og har mulighet til å ta linsekorrigerte snapshots. ROV-en har et mye utprøvd tre-thrustersdesign og er konstruert slik at sluttbruker kan avgjøre om beina skal brukes. Et modulært feste gjør at beina kan tas av slik at fartøyet blir mer hydrodynamisk og manøvrerbart i flymodus. ROV-en har nøytral oppdrift uten bein og negativ oppdrift med bein. For å styre beina ble det brukt forover- og inverskinematikk til å programmere en alternerende tripod-gangart.

Det har blitt jobbet ut ifra produktutviklingsmetoden som undervises ved NTNU i Ålesund, der et verktøy til dataassistert konstruksjon (Siemens NX 10.0) ble brukt til å utvikle 3D-modeller. 3D-modellene ble eksportert til 3D-printere på FabLab på NTNU i Ålesund, slik at fysiske prototyper dannet grunnlaget for testing av funksjonsprinsipper og videre utforming av design.

## Forord

Denne bacheloroppgaven er utarbeidet av syv studenter ved NTNU i Ålesund. Oppgaven er gjennomført som en avsluttende del av ingeniørutdannelsen ved avdeling for ingeniør- og realfag (automatiseringsteknikk) og avdeling for maritim teknologi og operasjoner (produkt- og systemdesign). Oppdragsgiver er NTNU i Ålesund og Kystverket ved kontaktperson Per Erik Ose. NTNU i Ålesund og Kystverket har finansiert prosjektet, men er ikke ansvarlige for dets konklusjoner eller anbefalinger.

Å jobbe i en prosjektgruppe bestående av syv personer fra ulike studieretninger har vært utfordrende. De fleste gikk inn i prosjektet med liten forståelse for hvor mye aktiv styring og prosjektledelse som kreves for å lykkes med et tverrfaglig prosjekt av denne størrelsesordenen. Arbeidsmengden har vært stor. Dette reflekteres i prosjektresultatet, som er et konsept fremfor en ferdig prototype.

Arbeidet vi har utført med å skissere et konsept, i kombinasjon med våre anbefalinger for videre arbeid, utgjør et godt grunnlag for videre utvikling av konseptet.

Vi ønsker å takke veilederne våre: Ottar Osen, Robin Bye og Paul Steffen Kleppe fra NTNU i Ålesund. Deres kritiske blikk bidro til å drive prosjektet fremover og identifisere forbedringspotensial underveis. Vi ønsker også å takke professor Houxiang Zhang for sine innspill tidlig i prosjektet. Lars Petter Bryne takkes for å ha frest en prototype av skroget. Takk til Andre Tranvåg som hjalp oss i forbindelse med bruk av verkstedet til NTNU i Ålesund. Anders Sætersmoen, Morten Sandseth, Hans Støle og Girts Strazdins var behjelpelige i forbindelse med bestilling eller utlån av komponenter og materiale. Takk til Tommy Alnes for å ha sponset prosjektet med krabber. Takk til akvarist og arbeidsleder på Atlanterhavsparken Rune Veiseth som viste oss rundt og informerte om skalldyr.

Takk til NTNU i Ålesund for å ha støttet prosjektet økonomisk og for å ha stilt med lokaler, utstyr og materiell. Paul Steffen Kleppe takkes for å ha bidratt med et elektronisk kurs i Siemens NX10.0.

Takk til Kystverket for en spennende oppgave, for utlån av programvare, møtelokaler og store mengder kaffe.

Til slutt takker vi hverandre for støtte, oppmuntring og hyggelige stunder.

Sist, men ikke minst, ønsker vi å rette en stor og personlig takk til Per Erik Ose fra Kystverket. Hans smittende engasjement og faglige tyngde har vært avgjørende for prosjektets suksess.

God lesning!

Ålesund, 31. mai 2016

Christian Dittmann

Anders Myklebust Hellebostad

Christian Myklebust Hellebostad

Stephen Kvalsvik

Martin Strøm Pedersen

Morten Erling Skår

Linn Kristin Øwre

## Sammendrag

Hensikten med denne oppgaven er å utvikle et konsept til en kompakt og bærbar observasjons-ROV som kan bistå Kystverket med å inspisere sjømerker. ROV-en skal være lett å manøvrere, slik at inspeksjon og dokumentasjon kan gjennomføres på en effektiv og fleksibel måte. ROV-en skal kunne levere en direkte videostrøm med godt bilde. For å forflytte seg skal ROV-en bruke thrustere og/eller bein. ROV-en skal være rimelig og enkel å vedlikeholde.

Prosjektgruppa fikk et budsjett på 15 000,- NOK og en tidsramme på fem måneder.

Det har blitt jobbet ut ifra produktutviklingsmetoden som undervises ved NTNU i Ålesund. Siemens NX 10.0, et verktøy til dataassistert konstruksjon (DAK), ble brukt til å utvikle 3D-modeller. 3D-modellene ble eksportert til 3D-printere på FabLab på NTNU i Ålesund, slik at fysiske prototyper dannet grunnlaget for testing av funksjonsprinsipper og videre utforming av design.

I arbeidet med hovedprosjektet har prosjektgruppa skissert et forslag til en observasjons-ROV. Den skal styres fra en kontrollstasjon bestående av en PC og en håndkontroller. Konsept-ROV-en prosjektgruppa har utviklet har et hydrodynamisk skrog med seks bein til å stå stødig på underlaget under filming. For å styre beina ble det brukt forover- og inverskinematikk. Gangarten som ble valgt var inspirert av et insekt. ROV-en skal levere rotasjonskorrigert videostrøm som utføres i OpenCV. Det skal også være mulighet for å ta linsekorrigerede snapshots. Til forflytning er det brukt et mye utprøvd tre-thrusterdesign. ROV-en er konstruert slik at sluttbruker kan avgjøre om beina skal benyttes. Et modulært feste gjør at de kan tas av slik at fartøyet blir mer hydrodynamisk og manøvrerbart i flymodus. ROV-en har nøytral oppdrift uten bein og negativ oppdrift med bein. Ved videreføring av prosjektet anbefaler vi at det utredes en løsning med batteri som strømforsyning. Det anbefales også å vurdere andre løsninger enn bein.

Noe unøyaktighet knyttes til beregningene som er gjort på oppdrift, motstand og effektforbruk. Det tas forbehold om at de teoretiske randbetingelsene styrkeberegningssfunksjonen til Siemens NX 10.0 bruker samsvarer med virkeligheten. Det tas også forbehold om at styrkeegenskapene til 3D-filamentet som er brukt i komponenter samsvarer med det produsenten oppgir. Materialeegenskapene vil påvirkes av printerinnstillinger og miljøbetingelser som lufttemperatur, luftfuktighet, samt 3D-filamentets alder og oppbevaringsmåte.

## Innholdsliste

<b>1</b>	<b>INNLEDNING</b>	<b>1</b>
1.1	BAKGRUNN	1
1.2	OM OPPDRAGSGIVER	1
1.3	FORMÅL	1
1.3.1	<i>Problemstilling og resultatmål</i>	2
1.3.2	<i>Effekt mål</i>	2
1.4	AVGRENSNINGER	3
1.5	RAPPORTENS OPPBYGGING	4
<b>2</b>	<b>TEORETISK GRUNNLAG</b>	<b>5</b>
2.1	ROV-HISTORIKK	5
2.1.1	<i>Bakgrunn</i>	5
2.2	UTVIKLING	5
2.2.1	<i>Fase 1 – Poodle</i>	5
2.2.2	<i>Fase 2: Militærbruk</i>	6
2.2.3	<i>Fase 3: Dagens bruksområder</i>	7
2.3	ROV-KLASSIFISERING	8
2.3.1	<i>Arbeidsfartøy</i>	8
2.3.2	<i>Observasjons- og inspeksjonsfartøy</i>	8
2.4	INSPIRASJONSKILDER	8
2.4.1	<i>OpenROV</i>	9
2.4.1	<i>BluEye Robotics</i>	9
2.4.2	<i>VideoRay «Scout»</i>	10
2.4.3	<i>Espy 360°</i>	10
2.4.4	<i>Indel-Partner «Gnom»</i>	11
2.4.5	<i>Hydrovision «Hyball»</i>	11
2.5	BIOMIMETIKK	12
2.5.1	<i>Hexapoder</i>	13
2.5.2	<i>RoboLobster – en biomimetisk robot</i>	13
<b>3</b>	<b>METODE</b>	<b>15</b>
3.1	DESIGN OG KONSTRUKSJON	15
3.1.1	<i>Teoretisk bakgrunn</i>	15
3.1.2	<i>Metode</i>	18
3.1.3	<i>Materialer</i>	21
3.1.4	<i>Resultat og diskusjon</i>	25
3.1.5	<i>Delkonklusjon</i>	75
3.1.6	<i>Kildevurdering</i>	78

3.2	ELEKTRONIKK, STYRING OG BILDEBEHANDLING .....	79
3.2.1	<i>Teoretisk bakgrunn</i> .....	79
3.2.2	<i>MATERIAL OG METODE</i> .....	91
3.2.3	<i>Metode</i> .....	98
3.2.4	<i>Resultat og diskusjon</i> .....	118
3.2.1	<i>Delkonklusjon</i> .....	134
<b>4</b>	<b>RESULTATER OG DISKUSJON</b> .....	<b>136</b>
<b>5</b>	<b>KONKLUSJON</b> .....	<b>139</b>
<b>6</b>	<b>REFERANSER</b> .....	<b>140</b>
<b>VEDLEGG</b>		

## Figurliste

Figur 1. "Pegasus" .....	6
Figur 2. «Poodle» .....	6
Figur 3. U.S Navy "CURV" til venstre og Royal Navy "Cutlet" til høyre .....	7
Figur 4. OpenROV 2.8 (venstre), OpenROV Trident (midten), BluEye Robotics (høyre).....	9
Figur 5. VideoRay "Scout" og Espy 360° .....	10
Figur 6. Gnom «Baby» og Hydrovision «Hyball».....	11
Figur 7. Bakbeina til en sikade.....	12
Figur 8. RoboLobster.....	13
Figur 9. Landkrabbe og strandkrabbe.....	15
Figur 10. Strandkrabben fra Atlanterhavsparken i Ålesund. ....	16
Figur 11. Kråkebolle.....	17
Figur 12. Produktutviklingstrekanten.....	18
Figur 13. Unbrakoskruer.....	23
Figur 14. Gummipakning og asbestfritt pakningsmateriale.....	24
Figur 15. Skrogdesign 1 sett forfra og skrått fra siden.....	25
Figur 16. Skrogdesign 2 .....	26
Figur 17. 3D-print av krabbeskall-designet.....	27
Figur 18. Skrogdesign 3 .....	28
Figur 19. Skrogdesign 3 sett forfra.....	28
Figur 20. Skrogdesign 3 sett bakfra.....	28
Figur 21. Frest skalamodell av skrogdesign 3.....	29
Figur 22. Utkast av skrogdesign 4.....	30
Figur 23. Sammenstilling av skrogdesign 5 med komponenter.....	31
Figur 24. Skrogdesign 5 sett forfra.....	32
Figur 25. Skrogdesign 5 sett fra siden.....	32
Figur 26. Sammenstilling av skrogdesign 5 med thrustere og kamera.....	32
Figur 27. Skrogdesign 6.....	33
Figur 28. Snittbilde av sammenstillingen til skrogdesign 6.....	34
Figur 29. Detaljbilde som viser kamera.....	34
Figur 30. Aluminiumsplate sett fra innsiden.....	35
Figur 31. Aluminiumsplate sett fra utsiden.....	35
Figur 32. Dekselets innfestningsmetode sett bakfra.....	36
Figur 33. Snittbilde av skruerinnfestningen i deksel og skrog.....	37
Figur 34. Mutter og innskruingsmutter.....	37
Figur 35. Sammenstilling av fartøyet med symbol for massesenter.....	38

Figur 36. Sammenstilling av konstruksjonen med plassering av komponentene.....	38
Figur 37. Vannstrøm rundt ulik geometri.....	39
Figur 38. Tidlig håndtak på dekselet.....	41
Figur 39. Håndtak til skrogdesign 7.....	41
Figur 40. Prototype 1.0.....	45
Figur 41. Prototype 2.0.....	45
Figur 42. Prototype 3.0.....	45
Figur 43. Prototype 4.0.....	45
Figur 44. Prototype 1.0.....	46
Figur 45. Skalamodell.....	46
Figur 46. Prototype 2.0 sett bakfra og sett fra siden.....	47
Figur 47. Prototype 3.0.....	48
Figur 48. Kryssledd til prototype 3.0.....	48
Figur 49. Prototype 4.0.....	49
Figur 50. Støtdemper i enden av prototype 4.0.....	49
Figur 51. Servoholder på innsiden av skroget.....	50
Figur 52. Forbedret støtdemper i enden av prototype 4.1.....	50
Figur 53. Printet prototype 4.1 og servoholderen.....	51
Figur 54. 3D-print av Makerbot med 10% innfyll.....	51
Figur 55. Prototype 4.3.....	52
Figur 56. Svingledd og innfestningsbrakett til prototype 4.3.....	52
Figur 57. Prototype 4.4.....	53
Figur 58. Svingledd og innfestningsbrakett til prototype 4.4.....	53
Figur 59. Sandwichkonstruksjon (halv) av prototype 5.0.....	53
Figur 60. Deler av en flyvinge.....	54
Figur 61. Bein til skrogdesign 5.....	55
Figur 62. Skrogdesign 5 med bein.....	56
Figur 63. Tuppen på beinet.....	56
Figur 64. "True Shade"-render av beinet i NX.....	57
Figur 65. Fysiske modeller fremstilt til automatiseringsteknikk.....	57
Figur 66. Test av hydraulikk med sprøyter.....	60
Figur 67. Tilbakeslagsventil til venstre. Fra rotasjon til lineær bevegelse.....	60
Figur 68. Servo fra LMR til venstre. Servo fra LH til høyre.....	61
Figur 69. NX-Constraints til dekselet.....	63
Figur 70. Deformasjon og jevnføringsspenninger i dekselet.....	63
Figur 71. Deformasjon av skroget.....	65



Figur 72. Jevnføringsspenninger på skroget.....	65
Figur 73. «Joints» og «Actuators» på sammenstillingen.....	69
Figur 74. Nærbilde av bein med «joints» og «actuators».....	70
Figur 75. QR-kode med NX-simulering.....	70
Figur 76. Sammenstilling av ROV 1.....	74
Figur 77. Sammenstilling av ROV 2.....	74
Figur 78. Sammenstilling av ROV 3.....	74
Figur 79. Skrogets evolusjon.....	75
Figur 80. OpenCV logo.....	82
Figur 81. Illustrasjon av perspektiv.....	83
Figur 82. Parameterne.....	83
Figur 83. Kameraforvringning.....	84
Figur 84. Sjakkmonster og kamera plassering (kameraplassering, u.d.).....	84
Figur 85. Gimball for mobil (stabilisator, u.d.).....	85
Figur 86. Rolling shutter kamera.....	85
Figur 87 Rolling shutter bildetagning.....	86
Figur 88. Maestro 24-kanals.....	91
Figur 89. XBOX-One kontrollere.....	91
Figur 90. Levi Hobby servo.....	93
Figur 91. BlueRobotics T100.....	93
Figur 92. Adafruit 10DOF IMU.....	94
Figur 93. ELP 170.....	94
Figur 94. Teensy 3.2.....	95
Figur 95. ODROID XU4.....	96
Figur 96 Maestro kontrollcenter.....	97
Figur 97. Bilde til bilde transformasjon.....	99
Figur 98. X-bane.....	99
Figur 99. Y-bane.....	100
Figur 100. Bilde av sjakkmonster.....	101
Figur 101. Hjørner til sjakkmonster.....	102
Figur 102. Robotbein.....	103
Figur 103. Lengder og vinkler til ett bein.....	104
Figur 104. Klassenes hierarki.....	110
Figur 105. Filterets frekvensrespons.....	112
Figur 106. Spenning ved ROV, 30V forsyningspenning.....	115
Figur 107. Kontrollstasjonskonsept.....	120

Figur 108. Kunstig horisont.....	121
Figur 109. Kompass.....	122
Figur 110. Diverse data.....	122
Figur 111. Avansert data.....	122
Figur 112. Indikator til flymodus.....	123
Figur 113. Thrusterindikator.....	123
Figur 114. Skjerm bilde av kontrollapplikasjon.....	124
Figur 115. Programkoden til processFrameAndUpdateGUI.....	126
Figur 116. Bilde før og etter linsekorrigerings.....	127
Figur 117. Bilde med og uten stabilisering.....	128
Figur 118. Verdier av sinusgeneratoren.....	128
Figur 119. Matlab-simulering av bein.....	129
Figur 120. Verdier fra Java.....	130
Figur 121. 3D-printet demonstrasjonsbein.....	131
Figur 122. Forsyningsnett til ROV.....	132
Figur 123. Spenning ved ROV, 48V forsyningspenning.....	133

## Tabelliste

Tabell 1. Styrker og svakheter ved ABS.....	21
Tabell 2. Styrker og svakheter ved PLA.....	22
Tabell 3. Styrker og svakheter ved CPE.....	22
Tabell 4. Styrker og svakheter ved PETG.....	23
Tabell 5. Oversikt over pakningsmateriell.....	24
Tabell 6. Styrker og svakheter ved skrogdesign 1.....	26
Tabell 7. Styrker og svakheter ved skrogdesign 2.....	27
Tabell 8. Styrker og svakheter ved skrogdesign 3.....	29
Tabell 9. Styrker og svakheter ved skrogdesign 4.....	30
Tabell 10. Styrker og svakheter ved skrogdesign 5.....	33
Tabell 11. Vurdering av ulike styrker ved gitt antall bein (Zu, 2011). .....	44
Tabell 12. Styrker og svakheter ved Fireline.....	58
Tabell 13. Styrker og svakheter ved triline.....	58
Tabell 14. Styrker og svakheter ved startsnøre.....	59
Tabell 15. Styrker og svakheter ved line. ....	59
Tabell 16. Styker og svakheter ved hydraulikk i vår ROV.....	61
Tabell 17. Oppdriftstabell.....	68
Tabell 18. Maksimalt effektforbruk.....	113
Tabell 19. Oversikt over Buck konvertere for 30V løsning.....	114
Tabell 20. Oversikt over Buck konvertere 48V løsning .....	132
Tabell 21. Planlagt materiell og komponenter.....	136
Tabell 22. Medgått materiell og komponenter i løpet av prosjektperioden.....	137

## TERMINOLOGI

### **Begreper**

*Remotely Operated Vehicle (ROV):* En ubemannet og fjernstyrt undervannsfartøy som brukes til inspeksjons-, vedlikeholds- og reparasjonsarbeid av havromsinstallasjoner, samt kartlegging av havbunnen. I enkelte tilfeller brukes ROV-er også til bergingsarbeid, og de er svært viktige i forbindelse med marinbiologisk forskning. En ROV kan være fritt svømmende med kabler med nøytral oppdrift, eller veie flere tonn som krever spesialutstyr for sjøsetting og håndtering av kabler. Felles for alle ROV-er er at de er utstyrt med propeller, lys og videokameraer. Det er vanlig med instrumenter som måler vanndybde, -tetthet og -temperatur. Utover dette varierer utstyret med type arbeid ROV-en skal utføre.

*Tether Management System (TMS):* Et tether management system lagrer og fordeler kabel (tether) til ROV-en. TMS skal sikre god bevegelsesfrihet og frakoble ROV-en fra bevegelsene til overflatefartøyet. TMS kan enten være et slags «bur» som omslutter ROV-en under sjøsetting («garage TMS»), eller være plassert på ROV-ens overside («top-hat TMS»). TMS motvirker motstandskraften på tether ved å stramme og slakke den. Slik minimeres motstand forårsaket av undervannsstrømmer. (Saab Seaeye, 2016).

*Tether:* Kabel mellom TMS og ROV. Tether har nøytral oppdrift for å ikke tyngre ROV-en.

*Umbilical:* Kabel mellom TMS og overflatefartøy.

*Launch and Recovery System (LARS):* Et system for kontrollert sjøsetting og innhenting av tyngre ROV-er. Består av flere komponenter; inkludert trommel med slepering og vinsj til kabelen, vinsjmotor, A-ramme (eller kran) til sjøsetting, samt hydraulisk kraftforsyning til vinsjmotoren. (Saab Seaeye, 2016)

*Autonomous Underwater Vehicle (AUV):* En autonom undervannsrobot. En AUV har ikke tether eller umbilical og minner ofte om en torpedo i formen.

*Inshore:* Området langs med og inntil kystlinjen, inkludert bukter, viker og havneområder.

*Offshore:* «Til havs». Dypere havområder utenfor kysten. Begrepet brukes mye i forbindelse med petroleumsindustrien.

*Skvalpesone:* Området over høyvannslinjen på kystlinjer som regelmessig vaskes over av bølger. En alternativ betydning brukes i offshoreindustrien og henviser til overgangen fra luft til vann når man sjøsetter tunge objekter.

*Grunt vann:* Defineres som 0 – 25 meter (Ose, 2016)

*Rolig vann:* Dypere enn 15 meter (Ose, 2016)

*Flymodus:* Når ROV-en forflytter seg ved hjelp av thrustere.

*CAD (Computer Aided Design):* Verktøy til å lage 3D-modeller. Kjent som dataassistert konstruksjon (DAK).

*CAE Computer Aided engineering:* Verktøy til å kjøre ulike analyser på 3D-modellene.

*CAM (Computer Aided Manufacturing):* Verktøy til å realisere 3D-modellen. I vårt tilfelle vil 3D-printing være et CAM-verktøy.

*Rapid Prototyping:* Metode som brukes til å lage fysiske modeller av objekter tegnet i CAD-programmer.

*Det grønne skiftet:* En omstilling i utformingen av politikk og organiseringen av næringsliv som følge av at klimavitenskap anerkjennes. Det grønne skiftet har som hensikt å endre måten vi skaper verdier på. Målet er å sikre best mulig ressursutnyttelse og redusere klimautslipp.

*Blå økonomi:* Sikter til økonomisk vekst som kan knyttes til langsiktig bærekraftig bruk av havets ressurser.

*Biomimetikk/bionikk:* Forskningsfelt innen innovasjon og vitenskap som forsøker å identifisere prosesser og systemer i naturen og overføre dem til ny teknologi.

*Robot:* En datastyrt enhet som ved hjelp av sensorer mottar data fra omgivelsene, bearbeider disse og reagerer ved å utføre handlinger bestemt av forhåndsprogrammering. En robot består av tre hovedelementer: sensorer (data inn), enhet for behandling av data («hjernen»), og handling i form av bevegelse, lyd og/eller lys (data ut).

*Hexapod:* En robot med seks bein, oftest basert på insekters bevegelse og nervesystem. Pinnedyr og kakerlakker er mest brukt, ettersom begge arter er grundig kartlagt nevrofysiologisk og adferdsbiologisk.

*Mechatronics Concept Designer:* Verktøy i Siemens NX som anvender fysikkens lover i 3D-simulasjon.

*Embedded:* Innebygde kontrollapplikasjoner (innebygde systemer)

**Forkortelser**

ABS: Akrylnitril-butadien-styren  
CAD: Computer Aided Design  
CAE: Computer Aided Engineering  
CAM: Computer Aided Manufacturing  
CCD: charge-coupled device  
CMOS: Complementary Metal Oxide Semiconductor  
CPE: Chlorinated Polyethylene  
CURV: Cable-Controlled Underwater Recovery Vehicle  
DIY: Do It Yourself  
DOFs: Degrees of Freedom (frihetsgrader)  
EEPROM: Electrically Erasable Programmable Read-Only Memory  
EIS: Element Iterative Solver  
FPS: Frames Per Second  
GUI: Graphical User Interface  
HDMI: High-Definition Multimedia Interface  
IDE: Integrated development environment  
IK: Invers Kinematikk  
IMU: inertial-measurement-unit  
IOT: Internet of Things  
I/O: Input/Output  
IP: Internet Protocol  
IR: Infrarød  
LARS: Launch and Recovery System  
MCD: Mechatronics Concept Designer  
MIMO: multiple-input and multiple-output  
MSS: Maximum Segment Size  
MTU: Maximum Transmission Unit  
OS: operativsystem  
PC: Personal Computer  
PETG: Polyethylene terephthalate glycol (glykolmodifisert PET)  
PLA: Poly Lactic Acid  
PWM: Pulse-width modulation  
RAM: Random Access Memory  
ROV: Remotely Operated Vehicle  
SDK: Software Development kit  
SISO: Single-input single-output system  
SMPS: switched mode power supply  
TCP: Transmission Control Protocol  
TMS: Tether Management System  
TTL: transistor-transistor logic  
UART: Universal asynchronous receiver/transmitter  
UDP: User Datagram Protocol  
UVC: USB Video Class

## 1 INNLEDNING

Her presenteres bakgrunn for valg av oppgave, oppgavens formål og problemstilling, avgrensninger som er gjort, samt et avsnitt om oppdragsgiver. I tillegg presenteres prosjektgruppas argumentasjon for hvorfor bacheloroppgavens tema er viktig. Til slutt følger en gjennomgang av rapportens oppbygning.

### 1.1 Bakgrunn

Bakgrunnen for oppgaven er et ønske fra studentenes side om å jobbe med en problemstilling som krever tverrfaglig samarbeid og kreativitet. Å jobbe på tvers av studieretninger kan gi praktisk erfaring med hvordan samarbeid faktisk gjennomføres i bedrifter. Det var også et ønske om å jobbe med en eksternt gitt oppgave for å knytte kontakter og utvikle nettverk utenfor NTNU i Ålesund. Det var dessuten et ønske om å jobbe med en oppgave som er maritimt rettet. Verdier fra havet er Norges fremtid og vil utgjøre en bærebjelke for norsk økonomi (Regjeringen Stoltenberg I, 2000). All forskning som kan knyttes til den maritime næringen anses som viktig og samfunnsnyttig av prosjektgruppen, og det oppleves som meningsfullt å kunne komme med et bidrag. Mer om det samfunnsnyttige og samfunnsrelevante aspektet ved oppgaven er beskrevet i forbindelse med prosjektets effektmål senere i rapporten.

Oppdragsgiver er NTNU i Ålesund og Kystverket ved Per Erik Ose. Disse aktørene har laget oppgaven i fellesskap.

### 1.2 Om oppdragsgiver

I 1974 ble Norsk Fyrvesen, Norsk Losvesen og Norsk Havnevesen slått sammen til én nasjonal etat: Kystverket. I dag er Kystverket en del av Samferdselsdepartementet, med ansvar for å jobbe med bærekraftig utvikling av kystsonen. Dette inkluderer kystforvaltning, utvikling og bevaring av maritim infrastruktur, sjøsikkerhet og akutt forurensningsberedskap. Kystverkets visjon er å utvikle kysten og havområdene i Norge til å bli verdens rene og sikreste.

I tråd med denne visjonen har Kystverket formulert følgende hovedmål (Kystverket, 2016):

- 1. Bidra til effektiv sjøtransport*
- 2. Sikre trygg ferdsel i norske farvann og havområder*
- 3. Hindre/begrense miljøskade som følge av akutt forurensning i norske havområder eller på norsk territorium*

Hvordan dette prosjektet kan bidra til å realisere et delmål for Kystverket er beskrevet i avsnitt 1.3.1.

### 1.3 Formål

Formålet med oppgaven er å utvikle et undervannsfartøy med bein som skal programmeres til å gå på havbunnen. Rapporten skal dokumentere denne prosessen og gjøre rede for de valg som er tatt.

Studentene fra Produkt- og systemdesign ønsker å videreutvikle ferdigheter knyttet til design, designoptimalisering og konstruksjon. Studentene fra Automatiseringsteknikk vil videreutvikle ferdigheter knyttet til bildeanalyse, datakommunikasjon og motorikk.

### 1.3.1 Problemstilling og resultatmål

Den grunnleggende problemstillingen er å utvikle en kompakt og robust Remotely Operated Vehicle (ROV) med bein som kan operere i skvalpesonen. Skvalpesonen er foreløpig et svært krevende område for ROV-er, og det finnes et marked for utgaver som takler skvalpesonens utfordringer. Kystverket har behov for en slik ROV i forbindelse med vedlikehold og inspeksjon av sjømerker. Dette kan knyttes til Kystverkets delmål som sier: «*det skal ikke forekomme kollisjoner eller grunnstøtinger på grunn av svikt i Kystverkets sjøsikkerhetstjenester*» (Kystverket, 2016), samt målet knyttet til vedlikehold av marin infrastruktur og reduksjon av vedlikeholdsetterslep for fysiske navigasjonshjelpemidler (Kystverket, 2016). I dag brukes senkekamera og dykker til inspeksjon av sjømerker, men dette er svært kostbart og utgjør dessuten en unødig risiko for dykkeren. Å redusere både utgifter og risikoer er viktig for Kystverkets måloppnåelse og visjon.

ROV-en skal kunne overføre live video fra havbunnen i områder med grunt vann (0-25 meter). Etter ønske fra Kystverket vil det bli lagt vekt på å produsere gode bilder og at ROV har god fremkommelighet og er manøvrerbar. Samtidig skal det hentes inspirasjon fra naturen ved å bruke en krabbe som utgangspunkt for designet av ROV. Det vil også bli lagt vekt på å utvikle en kompakt, bærbar ROV som lett kan fraktes av en person. Dette er prosjektets konkrete resultatmål.

### 1.3.2 Effektmål

Prosjektets effektmål (verdimål) angir retning og ambisjon (Rolstadås, 2014) og begrunner hvorfor temaet er viktig. Vårt konkrete effektmål er å gjøre hverdagen til Kystverket enklere i forbindelse med inspeksjon av sjømerker i grunn sjø. I et utvidet perspektiv vil en rimelig, velfungerende og bærbar ROV dessuten kunne inspirere til et miljøengasjement hos barn og ungdom. Nettopp ved å utvikle et nyttig verktøy til å utforske livet i fjæra med, vil man kunne inspirere flere unge til å engasjere seg i den delen av Norges fremtid som ligger under havoverflaten. Olje- og gassnæringen vil ikke alltid være Norges hovedinntekt, og det grønne skiftet kommer (Bjartnes, 2015). Havet omtales som den nye kilden til økonomisk vekst og industrialisering, og vil være en ledende økonomisk drivfaktor i dette århundret (The Economist Intelligence Unit, 2015). For å bidra til overgangen til den nye hverdagen og oppdage nye inntektskilder for Norge, er det viktig å stimulere nysgjerrigheten og engasjementet til den neste generasjonen teknologer og forskere. Utforskning av havet er en måte man kan inspirere unge mennesker til å utdanne seg innen marinbiologi, teknologi, ingeniørfag og matematikk. En rimelig og enkel ROV kan fungere som «et leketøy med en hensikt» og slik stimulere interessen for å oppdage Norges neste hav-eventyr. Jo tidligere denne interessen vekkes, desto bedre for Norges fremtidige innovasjon og nyskaping.

Utforskning av havet er selvsagt ikke viktig bare av hensyn til økonomiske interesser. Grunnforskning på havet innebærer å kartlegge og beskrive de biologiske, geologiske, kjemiske og arkeologiske aspektene ved havene, og dette har en rekke potensielle praktiske formål og bruksområder. Med rundt 95% av havet utforsket (National Oceanic and Atmospheric Administration, 2014), gjenstår det mange oppdagelser. Mindre enn 0,0001% av havet har blitt tatt biologiske prøver av (Goodard, 2015). Å forske på biodiversitet i dypvannsøkosystem vil bidra til oppdagelsen av nye arter som kan komme til å spille viktige roller i forbindelse med utvikling av nye medisiner og matkilder. Forskning på havet kan også gjøre det enklere å forutsi tsunamier og jordskjelv. Forskning på havet er dessuten tett linket til forskning på klima og atmosfære og kan bidra til å løse utfordringer knyttet til global oppvarming (National Oceanic and Atmospheric Administration, 2010). Utfordringer knyttet til utforskning av havet kan også gi opphav til teknologiske innovasjoner som kan anvendes på andre områder og problemstillinger. Et relevant konsept i denne sammenhengen er den «blå økonomien». Økende verdensbefolkning legger et ekstremt press på havet som den nye kilden til mat, mineraler og energi.



Verdensbanken spår at innen 2030 vil 2/3 av all matfisk være oppdrettsfisk (The Economist Intelligence Unit, 2015). International Transport Forum la frem prognoser i 2015 som viste at offshore vindturbinparker vil vokse seg ti ganger større innen 2030, og transport sjøveien vil firedobles innen 2050 (The Economist Intelligence Unit, 2015). En rapport utarbeidet av The Economist i forbindelse med World Ocean Summit 2015 («The blue economy: Growth, opportunity and a sustainable ocean economy») undersøkte muligheter og fallgruver ved den blå økonomien, samt strategier for å forsikre at videre industrialisering av havet gjøres på en bærekraftig måte. Bekymringen er at utvinningen av ressursene i havet vil skje i et raskere tempo enn det forskere som undersøker hvor grensen går for bærekraftig drift kan holde tritt med. Avveiningen mellom økonomisk vekst og sikring av marine økosystem vil være en av vår tids største utfordringer, men forskning, innovasjon og teknologi vil garantert være en del av løsningen.

Denne hovedoppgaven tar ikke sikte på å løse verdensproblemer, men undervurderer heller ikke den potensielle betydningen av det kollektive bidraget studentoppgaver utgjør.

#### **1.4 Avgrensninger**

De mest betydningsfulle avgrensningene i prosjektet er tids- og kostnadsrelaterte. Øvrige avgrensninger er utarbeidet av Per Erik Ose fra Kystverket. Han kom med oppgavens spesifikasjoner basert på Kystverkets behov. Det ble presisert at fokuset skulle ligge på å utvikle en rimelig ROV som er enkel å frakte og manøvrere. ROV skulle dessuten være enkel å vedlikeholde og finne reservedeler til.

Tidsramme: januar – mai 2016

Budsjett fra NTNU i Ålesund: 10 000 NOK

Budsjett fra Kystverket: 5000 NOK

For ROV gjelder følgende spesifikasjoner og funksjonelle krav:

- Arbeidsområde: skvalpesonen og områder i sjøen med grunt vann (0-25 meter)
- Arbeidsoppgave: overføre live video fra havbunnen med god oppløsning og ingen betydelige forsinkelser
- Maks vekt: 15 kg

ROV skal:

- Tåle påkjenningene i skvalpesonen inkludert trykkpåkjenningene ved 25 meters dybde
- Være enkel å manøvrere
- Gå på havbunnen uten at det påvirker bildekvaliteten
- Ha mulighet for å stå i ro for å få oversikt over omgivelsene
- Være enkel å vedlikeholde
- Bestå av få og lett erstattelige deler

## 1.5 Rapportens oppbygging

Rapporten er delt inn i seks kapitler<sup>1</sup>:

Kapittel 1: Innledning

Kapittel 2: Teoretisk grunnlag

Kapittel 3: Metode

3.1: Design og konstruksjon

3.2: Elektronikk, styring og bildebehandling

Kapittel 4: Resultater og diskusjon

Kapittel 5: Konklusjon

Kapittel 1 presenterer oppgaven med formål og problemstilling i et overordnet, samfunnsnyttig perspektiv.

Kapittel 2 orienterer om historikken til moderne ROV-er og forsøker samtidig å plassere prosjektets bidrag innenfor en tradisjon i ROV-utvikling. Det orienteres også om biomimetikk som fagområde, ROV-modeller og biomimetiske roboter som prosjektgruppa hentet inspirasjon fra i designfasen.

Kapittel 3 består av to underkapitler. Del 1 beskriver design- og konstruksjonsprosessen til studentene fra produkt- og systemdesign. Del 2 gjør rede for arbeidet knyttet til automatiseringsteknikk og omhandler elektronikk, styring og bildebehandling. Begge underkapitler har egne avsnitt som beskriver teoretisk bakgrunn, metode, resultat og diskusjon, samt hver sin delkonklusjon med forslag til videre arbeid.

Kapittel 4 presenterer prosjektgruppas samlede forslag til løsning ved å oppsummere materiale og komponenter som må kjøpes inn.

Kapittel 5 består av en konklusjon med anbefalinger for videre arbeid.

---

<sup>1</sup> Rapporten tar utgangspunkt i malen som er beskrevet i heftet «Praktisk rapportskrivning» av professor Nils Olsson ved NTNU.

## 2 TEORETISK GRUNNLAG

Dette kapitlet gir en kort innføring i ROV-ens historiske bakgrunn og utvikling. ROV-en til prosjektgruppen henter designinspirasjon fra eksisterende produkter, men også fra biomimetiske roboter. Derfor vil det også refereres til noe litteratur som omhandler biomimetisk robotteknologi.

### 2.1 ROV-historikk

#### 2.1.1 Bakgrunn

Havene dekker over 70% av kloden (Oceanic Institute, 2016), men mennesker er ikke spesielt godt tilpasset forholdene i havet. Mennesket har derimot en unik kombinasjon av utforskertrang og evne til å overkomme begrensninger naturen har satt. En ROV er et konkret eksempel på å overkomme en slik begrensning.

*«Det første trinnet når det kommer til å forstå en teknologi, er å forstå hvorfor det er behov for den. Når det kommer til ROV-teknologi, er grunnen ganske enkel: det finnes ingen annen praktisk, trygg og økonomisk forsvarlig måte å utføre arbeid under vann på» (Michel, u.d.)<sup>2</sup>.*

Sitatet er fra Drew Michel, eier av ROV Technologies Inc. Han sier noe essensielt om behovet for ROV-er: mennesker ønsker å utforske og utnytte ressursene som ligger stadig dypere i havet, men kan ikke gjøre det uten utstrakt bruk av teknologi.

### 2.2 Utvikling

Grovt inndelt kan man skille mellom tre faser i ROV-ens utvikling (Marine Technology Society, 2014). Første fase knyttes til en fransk ingeniør og undervannsfotograf som utviklet ny teknologi for å øke kvaliteten på bildene sine, samt filme på større dyp. ROV-ene ble mer avanserte da marinen i USA fant nye bruksområder. Den militære påvirkningen er beskrevet i fase 2. Fase 3 innledes da offshore petroleumsindustri setter nye krav til ROV-ens kapasitet og ytelse, og teknologien har modnet nok til at et privat marked vokser frem for hobbybruk.

#### 2.2.1 Fase 1 – Poodle

Det eksisterer noe uenighet om hvilken ROV som skal regnes som den første. Den mest omtalte er «Poodle», utviklet i 1952 av Dimitri Rebikoff (Marine Technology Society, 2014). Rebikoff var en fransk ingeniør og oppfinner. Han krediteres for å ha tatt undervannsfilmning til et nytt nivå med sin forskning på undervannskameraer- og fartøy (Eyman, 1986).

Rebikoff utviklet bedre kameraløsninger til undervannsbruk, inkludert den første elektroniske undervannsblitzen (Rebikoff-Niggeler Foundation, 2016). For å øke kvaliteten på filmopptakene utviklet han en innretning kalt «Pegagsus»; en torpedoformet undervannsscooter som gjorde det enklere for dykkerne å holde kameraene stabile under vann. Pegasus ble designet etter at Rebikoff studerte haiens bevegelser (Eyman, 1986).

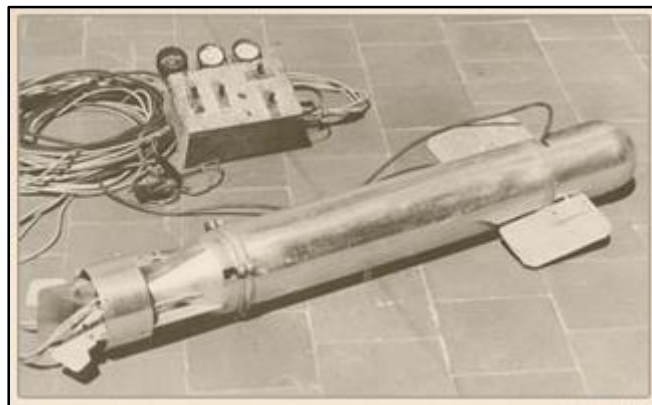
---

<sup>2</sup> Forfatterens oversettelse



Figur 1. "Pegasus".

Pegasus gjorde det enklere for dykkere å ta seg fram under vann med tungt filmutstyr, og ble forløperen til Poodle (RovMarine Technologies, 2016). Rebikoff innså at dykkeren var et unødvendig mellomledd og fordelten med å gjøre Pegasus førerløs var at man ville kunne filme på større dyp og få bilder og video man ikke ville få ved bruk av dykkere. Rebikoffs oppfinnelser oppnådde kommersiell suksess og brukt av filmindustrien, oljeselskaper og den amerikanske marinen (The Rebikoff-Niggeler Foundation, 2016).



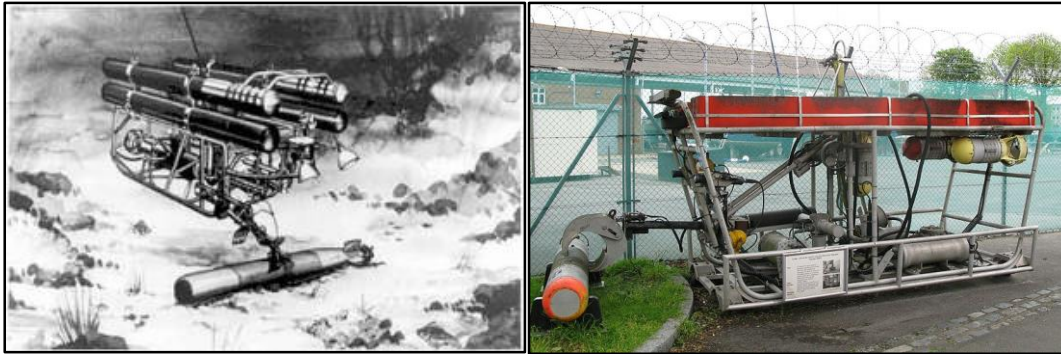
Figur 2. «Poodle».

(Chiles, 2014)

### 2.2.2 Fase 2: Militærbruk

Ubemannede undervannsfartøy ble raskt plukket opp av forskere fra «Space and Naval Warfare Systems Center Pacific» på oppdrag for den amerikanske marinen. Senteret utviklet «CURVs» (Cable-Controlled Underwater Recovery Vehicles) på 1950- og 1960-tallet. Et av de første bruksområdene var å rydde miner og torpedoer fra havbunnen etter militærøvelser holdt av den amerikanske marinen (Seaview Systems Inc, 2016).

Fartøyet CURV-I ble berømt da den assisterte i å hente opp en hydrogenbombe fra havbunnen i Middelhavet i 1966 (Paschoa, 2014), og etterfølgeren CURV-III fikk enda større omtale da den i 1973 bistod i den vellykkede redningsoperasjonen som reddet mannskapet i en havarert ubåt på 480 meters dyp (Barford, 2013). CURVen «Cutlet» ble senere utviklet av den britiske marinen og regnes også som en av forløperne til den moderne ROV-en (Bluebird Marine Systems Ltd, 2016).



Figur 3. U.S Navy "CURV" til venstre og Royal Navy "Cutlet" til høyre.

En «CURV» fra den amerikanske marinen. «Cutlet»: utviklet av Royal Navy på 50-tallet.  
(American Oil & Gas Industry Historical Society, 2011)

(Bluebird Marine Systems Ltd, 2016).

### 2.2.3 Fase 3: Dagens bruksområder

ROV-er ble videreutviklet på 1970- og 1980-tallet da olje- og gassindustrien krevde mer enn det dykkere kunne håndtere av dype dykk og utfordrende arbeidsoppgaver (Bluebird Marine Systems Ltd, 2016). I denne bransjen har ROV-er gjort dykkere overflødige og på denne måten kraftig redusert antall risikofylte oppdrag for mennesker (Bluebird Marine Systems Ltd, 2016). Før var det vanlig å bruke dykkere i alle faser offshore petroleumsvirksomhet – fra leting etter olje og bygging av plattformer til selve utvinningsprosessen (Bjerga, 2003). I dag er bruk av ROV-er så godt integrert i offshoreindustrien at selv oppdrag dykkere ville ha utført tidligere, er overlatt til ROV-er.

Marinbiologisk forskning gjør store fremskritt takket være ROV-er. Svært mye av kunnskapsbasen vi har om dypvannsøkosystemer er bygget på datamateriale innhentet ved bruk av ROV-er (National Oceanic and Atmospheric Administration, 2015), og nye arter har blitt oppkalt etter ROV-en som først oppdaget dem. Mange ROV-er har dessuten utstyr for å ta prøver av vannet slik at de kan analyseres. ROV-er har bidratt til arkeologisk forskning ved å gjøre vrak som ligger på store dyp tilgjengelige for undersøkelser. Mange historiske sunkne skip har blitt lokalisert og utforsket ved hjelp av ROV-er, deriblant RMS Titanic (Woods Hole Oceanographic Institution, 2015). Vedlikehold og renhold av fiskeoppdrettsanlegg har også blitt effektivisert ved bruk av ROV-er (Sperr ROV Technology AS, 2016).

## 2.3 ROV-klassifisering

En vanlig måte å klassifisere ROV-er er etter størrelse og effekt. Disse egenskapene vil være bestemmende for hva slags arbeidsoppgaver ROV-en kan utføre og hvor dypt den kan dykke. Den mest grunnleggende inndelingen kan se slik ut:

- Arbeidsfartøy (work class)
  - Klasse: «Light» eller «heavy duty»
- Observasjons- og inspeksjonsfartøy (observation class, inspection class)

Det er vanlig å spesifisere om ROV-en er en såkalt “frisvømmer” eller bruker TMS. En fritt svømmende ROV har en kabel med nøytral oppdrift som er direktekoblet til overflatefartøyet. Ved bruk av TMS vil det være to kabler involvert i dykket: en som går fra overflatefartøyet til TMS, og en fra TMS til ROV. Den generelle holdningen i industrien er at fordelene med å bruke TMS langt veier opp for ekstrakostnaden ved å investere i den (Saab Seaeye, 2016). Forenklet kan man generalisere og si at kommersielle ROV-er som regel er avhengige av TMS og LARS, mens hobbybaserte observasjons- og inspeksjons-ROV-er oftere kan greie seg uten.

### 2.3.1 Arbeidsfartøy

Arbeidsfartøy er mye brukt på oljeplattformer og store offshorefartøy. Dette er den mest omfangsrrike typen ROV, ettersom de kan fungere som en observasjons- og inspeksjonsfartøy i tillegg til å kunne utføre et bredt spekter av installasjons- og reparasjonsjobber. Arbeidsfartøy veier som regel mange tonn og er følgelig avhengige av LARS under sjøsetting og innhenting. Bruk av TMS er nesten regelen, og enkelte kan dykke inntil 10 000 meter. Arbeidsfartøy har stort energiforbruk og er følgelig som regel hydrauliske.

### 2.3.2 Observasjons- og inspeksjonsfartøy

Observasjons- og inspeksjonsfartøy skal hovedsakelig assistere ved eller utføre inspeksjonsarbeid i forbindelse med undervannsindustri, kartlegging av havområder og utforskning av havbunnen. Enkelte observasjons-ROV-er har såpass lite energiforbruk at de er elektriske. Eksempler på observasjons-ROV-er finnes i delkapittel 2.4.

## 2.4 Inspirasjonskilder

ROV-ene presentert i dette avsnittet besitter en eller flere egenskaper ettersom av oppdragsgiver. Det blir stadig vanligere at privatpersoner disponerer egne ROV-er til å utforske havet på egenhånd – såkalt «DIY ocean exploring» («do it yourself»). Disse ROV-ene er gode inspirasjonskilder for prosjektet ettersom de er rimelige, kompakte, lette og ukompliserte å bruke. I dette avsnittet presenteres fem ulike modeller fra fire forskjellige utviklere: OpenROV og VideoRay fra USA, BluEye fra Norge, Espy 360° fra Australia og Gnom ROV fra Russland.

### 2.4.1 OpenROV

OpenROV er et firma som designer og produserer mini-observasjons-ROV-er som egner seg godt til privat bruk. Målet til OpenROV er å senke kostnadene og øke tilgjengeligheten på ROV-er for allmenheten, noe som er i tråd med prosjektgruppas effektmål. OpenROV 2.8 og OpenROV Trident er to modeller som godt illustrerer egenskapene til en hobby-ROV. De er ultralette (2,6 kg), får plass i en ryggsekk, kan dykke ned til 100 meter og har gode manøvreringsegenskaper. Trident kalles en *undervannsdroner*, ettersom den er så lett å styre at operatøren har følelsen av å «fly». OpenROV bruker 8 LiFePO4-batterier til strømforsyning og har da en levetid på 2-3 timer (OpenROV, 2016).

OpenROV 2.8 strømmer live HD video med et webcam gjennom en to-wire tether. Den har nøytral oppdrift, en toppfart på 1,3 m/s og er enkel å manøvrere takket være tre thrustere. For å styre den trengs hverken internettforbindelse eller å laste ned software. OpenROV Trident er firmaets nyeste modell (lanseres høsten 2016), og den er både raskere (2 m/s) og mer manøvrerbar enn 2.8-modellen, sistnevnte grunnet det nye designet og en mer hensiktsmessig plassering av thrusterne. Tridents nye design er basert på tilbakemeldinger fra brukere verden over.



Figur 4. OpenROV 2.8 (venstre), OpenROV Trident (midten), BluEye Robotics (høyre).

(OpenROV, 2016)  
(Blue Robotics, u.d.)

### 2.4.1 BluEye Robotics

BluEye er en undervannsdroner utviklet ved NTNU i Trondheim. Designet er nyskapende («en prosjektor snudd på hodet» (NTNU, u.d.)), men firmaets visjon likner OpenROV sin: å gjøre områdene under havoverflaten tilgjengelig for alle. I likhet med OpenROV har teamet bak BluEye utviklet en droner for det globale forbrukermarkedet slik at flere aktivt kan drive med undervannsutforskning. BluEye ønsker å utvikle et produkt som vil få flere utenfor academia til å interessere seg for tilstanden i verdenshavene og engasjere seg for utfordringene vår generasjon står overfor knyttet til klima og miljø.

Brukeropplevelsen står sentralt i utviklingen av BluEye. Fokuset har vært på å lage en undervannsdroner med overlegen brukervennlighet, noe som har vært forsøkt realisert blant annet gjennom hvordan dronen styres og valg av oppløsning på kameraet. Ifølge daglig leder Erik Dyrkoren skal dronen styres fra en touch-skjerm og «være like intuitiv som Apple-produkter» (Stensvold, Webområde for Teknisk Ukeblad, 2015). BluEye ligner litt på en månefisk, ettersom den er svært høy i forhold til lengden. Forholdet mellom høyde, lengde og bredde gir god stabilitet.



### 2.4.2 VideoRay «Scout»

VideoRay er et amerikansk selskap som har spesialisert seg på å lage observasjons-ROV-er. «Scout» er deres mest kompakte modell og beskrives som «et svømmende kamera». Scout er ikke konstruert for å håndtere urolig vann, men har mange nok av de andre egenskapene etterspurt av oppdragsgiver til at den har vært en inspirasjonskilde.

Scout består av et lett chassis av anodisert aluminium, tre thrustere, to lyskilder og et vidvinkel forovervendt kamera. Scout er AC strømforsynt, men kan også drives ved bruk av batteripakke og inverter. Scout veier 3,6 kg og har nøytral oppdrift i ferskvann. For å oppnå det samme i saltvann (som har høyere tetthet enn ferskvann) kan ekstra vekt settes på ROV-ens «ballast retaining system» på fartøyets underside. (Videoray, 2004).



Figur 5. VideoRay "Scout" og Espy 360°.

(Underwater Inspection, 2011)

(Wakeham, James Dyson Award, 2016).

### 2.4.3 Espy 360°

En ROV med svært innovativt design er Espy 360°, utviklet av ingeniør Dale Wakeham. Utstyrt med 6 GoPro-kameraer vil ROV-en gi brukeren en form for «Google Street View»-opplevelse. Espy 360° er en aktuell inspirasjonskilde ettersom den er utviklet for å tåle påkjenningene i skvalpesonen. Vingedesignet med rotorere som kan tippes og en selvjusterende gyro gjør at Espy 360° greier seg godt selv når den utsettes for kreftene i skvalpesonen (Wakeham, James Dyson Award, 2016). Designet oppleves som originalt og spennende, noe som kan bidra gruppas effektmål om å appellere til ungdom.

Espy 360° tar sikte på å utvikle bruken av dykkere ved marinbiologiske forskningsavdelinger i Australia. Hensikten er å kunne utforske marine miljø på en skånsom måte som medfører mindre forstyrrelser for planter og dyr, samt reduserte kostnader. Espy 360° kan gjøre marinbiologisk utforskning tryggere og billigere, samt mer effektiv, pålitelig og med mindre forstyrrelser for det marine miljøet (Sistare, 2013). Dette målet ligger tett opp til Kystverkets ønske om et teknisk hjelpemiddel som kan bidra til trygg og kostnadseffektiv inspeksjon uten å påvirke det marine miljøet unødige.



#### 2.4.4 Indel-Partner «Gnom»

Indel-Partner produserer små inspeksjons-ROV-er (2-25 kg) kalt Gnom (russisk for «dverg»). Modellen «Baby» er ment for dybder ned til maksimalt 50 meter og er Indel-Partners minste, noe som gjør at den egner seg godt til hobbyutforskere. Hele systemet kun veier 8 kg, så «Baby» kan enkelt fraktes av én person. Gnom kan sjøsettes for hånd, selv fra små og ustabile fartøy som for eksempel kanoer. Fra produktbeskrivelsen kan man lese at «Baby» ligger tett opp til ønsket fra Kystverket om en ROV som er lett, kompakt, kan sjøsettes for hånd (selv i røff sjø) og som takler kreftene i grunt vann grunnet det gunstige forholdet mellom vekt og thrusterens skyvekraft (Advanced Diving Technology, 2016). Baby har to thrustere for horisontal forflytning og en for vertikal. Kameraet har lite synsfelt (66°), men er montert slik at det kan vris +/- 50°.



Figur 6. Gnom «Baby» og Hydrovision «Hyball».

(Advanced Diving Technology, 2016).

(Innovative Solutions & Technologies, 2011)

#### 2.4.5 Hydrovision «Hyball»

Hyball er produsert av Hydrovision og videreutviklet hos Ocean Systems Laboratory ved Heriot-Watt University. Hyball har to thrustere som gir bevegelse forover og bakover, samt rotasjonsbevegelse. To andre thrustere gir vertikal bevegelse. Hyball kan dreies om sin egen akse og har derfor en kamerarekkevidde som gir et 360° synsfelt (Silvercrest Submarines and ROVs, 2016). Ulempen er at den er tung (42 kg), og har et chassis som lett kan sette seg fast i tareskoger.

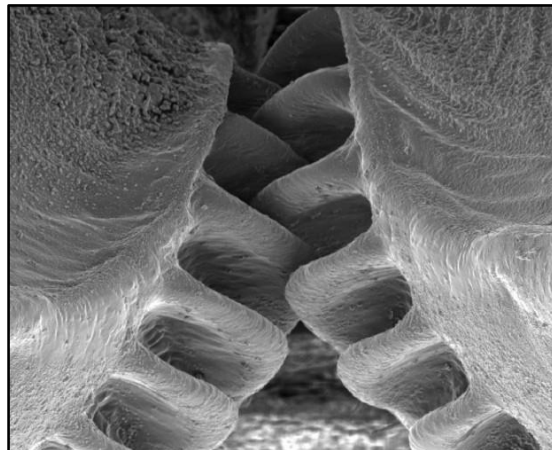
## 2.5 Biomimetikk

*“We envision a world in which people view nature not as a warehouse of goods but as a storehouse of knowledge and inspiration for solutions.” (Biomimicry Institute, 2016)*

Biomimetikk er fagområdet som tar utgangspunkt i at ingeniører, teknologer og forskere bør se til biologien etter inspirasjon. Biomimetikk går ut på å undersøke og forstå prosesser i naturen og overføre kunnskapen til innovasjon og teknologi man kan anvende på menneskeskapt problemstillinger. Bærebjelken i filosofien er å ikke se på naturen bare som et varehus med materialer, men som et bibliotek av kunnskap revidert gjennom milliarder av år. Tanken er at naturen allerede «løst» alle problemer knyttet til for eksempel motorikk, designoptimalisering og energibesparelse.

Å hente inspirasjon fra naturen i designfasen er en tradisjon med dype røtter. Leonardo Da Vinci konstruerte sin første flymaskin etter å ha studert flaggermusvinger. Borrelåsen går for å være den mest kjente og største kommersielle suksessoppfinnelsen inspirert av naturen – oppfunnet av George de Mestral på 50-tallet, etter en observasjon om at borrefruktur satte seg fast i dyrepels. Nye typer lim har blitt utviklet etter å ha studert Gekkoføtter. Det japanske lyntoget Shinkansen fikk fronten redesignet fordi en ingeniør med interesse for fuglekikking oppdaget at isfuglen kunne stupe lynraskt ned i vann med kun et minimum av forstyrrelser på vannoverflaten. Lyntogets front er i dag i praksis et 15 meter langt «nebb» som reduserer støy og energiforbruk.

I enkelte tilfeller var naturen først ute med løsninger man trodde var menneskeskapt. Et eksempel er tannhjul. Det hoppende sikadeinsektet «*Issus coleoptratus*» sparker fra med beina så raskt at nerveimpulsene ville vært for trege til å koordinere bevegelsen. Løsningen ble tannhjul, som sammenkobler bakbeina og sikrer kontrollert bevegelse. Et annet eksempel er hydraulikk, som både edderkopper og krabber (som delte stamfar for 550 millioner siden) bruker til å bevege lemmene.



Figur 7. Bakbeina til en sikade.

(Burrows, 2013)

I dag har en rekke universiteter forskningslaboratorier for biomimetikk, inkludert Harvard, Stanford og MIT. I tillegg finnes egne institutter for biomimetikk: Biorobotics research group i Frankrike og Biomimicry Institute i USA er to eksempler. Hensikten er å få til en holdningsendring ved å promotere overføringen av konsepter, design og strategier fra biologiske systemer til bærekraftig systemdesign. En kritisk problemstilling fremover er «hvordan utvikle produkter med et minimum av materialer og energiforbruk?». En planet med begrensede ressurser gjør klokt i å implementere en filosofi som forsøker å imitere naturens produktutvikling.

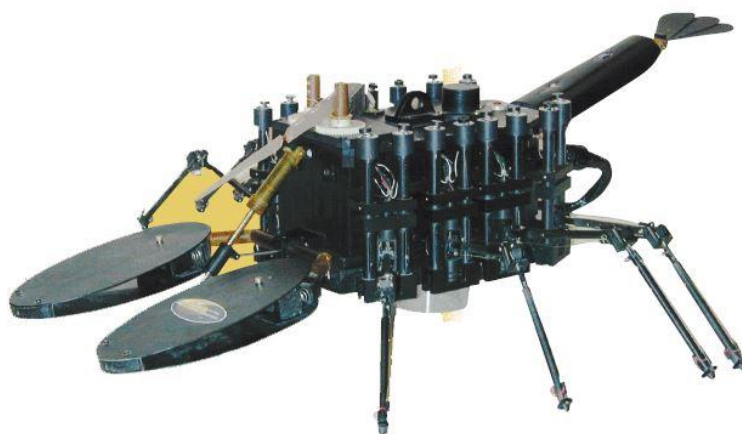
### 2.5.1 Hexapoder

Hexapoder (gresk for «seksfotinger»), bedre kjent som insekter, utgjør den vanligste typen leddyr. Over 75% av alle verdens dyr er hexapoder (Semb-Johansson, 2015), noe som bekrefter deres evolusjonære suksess. Hexapod refererer også til biomimetiske roboter med seks bein. Hexapodrobotene er ofte basert på kakerlakker eller pinnedyr, ettersom begge artene har enkle nervesystem som har blitt studert grundig.

Kjente robohexapoder inkluderer RHex og RiSE utviklet av Boston Dynamics. RHex har roterende bein som kan beveges uavhengig av hverandre og dermed produsere svært tilpassede gangarter som takler meget ulendt terreng (Boston Dynamics, 2016). RiSE kan klatre vertikalt opp vegger takket være mikrokjør på hvert av de seks beina og en «holdning» som tilpasses underlagets kurvatur (Boston Dynamics, 2016).

### 2.5.2 RoboLobster – en biomimetisk robot

RoboLobster er en robot utviklet av Northeastern University. Den er biomimetisk fordi den har et kunstig nervesystem, og fordi den er konstruert for å se ut som og oppføre seg som en hummer (arten *Homarus americanus*). Hummeren ble brukt som designinspirasjon fordi den er svært robust, effektiv til å navigere i ulendt terreng, samt perfekt hydrodynamisk tilpasset sine omgivelser (Safak & Adams, 2002). Hummeren kan dessuten stå imot sterke undervannsstrømmer (Ayers, 2004), noe som er vesentlig for denne type undervannsrobots suksess (roboten er tenkt brukt til å rydde miner på havbunnen).



Figur 8. RoboLobster.

([encyclopedia.wikia.com](http://encyclopedia.wikia.com), 2016)

Systemet bruker analoge fremfor digitale signaler til å overføre informasjon, og ligner derfor på dyrenevner. De analoge nevronene utgjør det elektroniske nervesystemet som kontrollerer robotene, og består av en rekke kretser som regner ut matematiske likninger. Disse likningene beskriver dynamikken til ionene som beveger seg inn og ut av nevroner for å produsere blant annet aksjonspotensial.

Fremfor å bruke motorer eller pneumatikk til å bevege leddene, brukes en metallisk muskel i wire-form kalt nitinol. Materialet er en legering med «hukommelse» (shape memory) som får sammentrekninger lik en muskel ved oppvarming. For å varme opp nitinolet kjøres pulser med strøm gjennom, som er drevet av nevronene i det elektroniske nervesystemet. Motstanden i wiren gjør at metallet varmes opp og får sammentrekninger.

Når pulseringen stopper, stopper også bevegelsen i leddet til roboten. På denne måten kan man si at roboten er biomimetisk styrt. Denne teknologien gjør at roboten blir helt autonom.

Utfordringene kommer når roboten møter på ukjente situasjoner og betingelser i miljøet. Robotens autonomi er basert på et nevronnettverk (nervesystem) som reagerer på kjente stimuli (sanseinntrykk). For å forsikre seg om at hummerroboten også kan reagere hensiktsmessig i ukjente situasjoner, utsettes levende hummere for motstridende stimuli i en kontrollert laboratoriesetting (Thompson, 2014).

Eksempelvis vil man kunne manipulere omgivelsene slik at hummeren får visuell input som tilsier at den beveger seg bakover, samtidig som man har en vannstrøm som treffer hummeren forfra. Under normale forhold vil det være overensstemmelse mellom hvordan for eksempel en hummers følehorn bevegges av vannets motstand og de visuelle sanseinntrykk den mottar. I laboratoriet vil man derimot kunne gi sanseinntrykk som normalt ikke oppleves sammen, studere hvordan hummeren reagerer på dette, og sammenligne reaksjonen med robohummeren. Ved utføre denne sammenligningen vil man se om det elektroniske nervesystemet fungerer slik det skal, eller om det trenger justeringer.

### 3 METODE

#### 3.1 Design og konstruksjon

##### 3.1.1 Teoretisk bakgrunn

Konseptutviklingen slutter seg til en biomimetisk tradisjon. Den bærende ideen er at naturen har jobbet med sin "produktkatalog" over en periode på 3 milliarder år, og at det er fornuftig å bla flittig i den når man forsøker å løse mekaniske problemer. Det første designet ble basert på strandkrabben *Carcinus Maenas*. Et hensyn som det ble lagt mer og mer vekt på utover i prosjektet var konstruksjonens hydrodynamikk, og dette førte til at et annet konsept ble vurdert: «kråkebolle»-designet. Senere ble designet endret igjen slik at det lignet mer på en hummer.

##### Inspirasjonskilder

Etter å ha studert utviklingen til undervannsfartøy frem til i dag samt biomimetisk produktdesign, tok studentene i produkt- og systemdesign utgangspunkt i to typer tifotkrepser og kråkebolle som inspirasjonskilder. Tifotkreps er en orden under rekken leddyr som inkluderer både krabber og hummere (van der Meeren, 2008). Felles for tifotkreps er at de har et skjelett ytterst som beskytter mot rovdyr og fungerer som muskelfeste (van der Meeren, 2008). Musklene brukes til forflytning og er plassert i grupper fordelt på hele kroppen. En forskningsrapport utarbeidet av Møreforskning og Havforskningsinstituttet beskriver det slik: «Musklene er organisert i funksjonelle grupper fordelt over hode, kropp, hale og føtter. Det er relativt få bevegelsesnerver som samspiller med hver muskel, men kombinasjonen av raske og sene synapser og sterke og svake signalsynapser gir en god kontroll over muskelfunksjonen.» (van der Meeren, 2008)

##### Strandkrabben

Strandkrabben *Carcinus Maenas* lever i skvalpe- og tidevannssonen langs hele kysten i Norge. Den trives på all slags sjøbunn (sandbunn, mudderbunn og fast bunn) ned til 6 meters dyp. Strandkrabben er svært godt tilpasset livet i fjæra og manøvrerer enkelt over hindre på bunnen og på land. Denne evnen ligger tett opp til prosjektbeskrivelsen om en ROV med bein som kan gå i skvalpesonen og i ulendt marint terreng. Dette gjorde at strandkrabben var et åpenbart sted å begynne da vi startet på designet.

##### Anatomi

Strandkrabben har en kropp som består av et «skall», et ryggskjold som kan bli 8 cm bredt, fire bein på hver side av ryggskjoldet og to klør i fronten. Strandkrabbens ryggskjold er bredere og flatere enn på landkrabben man finner i tropiske strøk. Landkrabbens kropp er formet som en litt høy «boks», hvorpå strandkrabbens form er slank og avlang. Dette er en tilpasning som gjør den mindre utsatt for påkjenninger fra bølger og undervannsstrømmer. Forskjellene i ryggskjoldets tilpasning basert på habitat er illustrert på figur 9.



Figur 9. Landkrabbe og strandkrabbe.

(Thai National Parks, 2016)

(Warren Photographic, 2016)

### *Bevegelse*

Strandkrabben har bein som stikker langt ut fra ryggskjoldet. Den beveger seg sidelengs, særlig om den skal bevege seg hurtig, men har også mulighet til å gå både baklengs og forlengs.

Strandkrabben kan gå like hurtig til høyre som til venstre, noe som gjør at den ikke trenger å snu hele kroppen ved en rask unnamanøver. Med fire bein plassert på hver side av kroppen har den til enhver tid 4 bein som skyver fra og 4 bein som drar. Resultatet blir en kraftig resultant som gjør krabben meget rask.



Figur 10. Strandkrabben fra Atlanterhavsparken i Ålesund.

### **Hummer**

Hummeren er en tiftokrebs som trives i grunt, urolig vann. Den er følgelig svært godt tilpasset til å stå imot kreftene i skvalpesonen.

### *Anatomi*

Hummere har avlang, tilnærmet sylindrisk kropp med to store klør og to mindre i front, samt en kraftig, muskuløs hale som gjør den til en rask svømmer. Amerikansk hummer kan bli opptil en meter i lengde og veie opptil 20 kg (Store Norske Leksikon, 2015).

### *Bevegelse*

Hummeren beveger seg forover og bakover ved hjelp av beina på hver side av den avlange kroppen, men svømmer baklengs ved å slå med halen dersom den må flykte raskt. Ved å bruke halen og klørne kan den innta finjusterte posisjoner som gjør at den kan stå stabilt selv i sterke undervannsstrømmer.

## Kråkebolle

Kråkebollen *Echinus Esculentus* har en hydrodynamisk utforming som gruppen hentet designinspirasjon fra. Ettersom inspirasjonen begrenset seg til kråkebollens form og ikke funksjon, anses det ikke som relevant å gjøre rede for dens anatomi eller bevegelsesmønster.



Figur 11. Kråkebolle.

(Jäntti, 2010)

## Edderkoppfamilien

En gruppe beslektede dyr til tifotkrepse er edderkoppfamilien. De åpenbare fellestrekkene i dag er at hele kroppen er leddet og dekket av et ytre skjelett. Edderkopper er bilateralt symmetriske (høyre- og venstresiden er speilbilder av hverandre) med 8 bein og to «munnbein» (følebein). Muskler på innsiden av skjelettet brukes til sammentrekninger av leddene, mens hydraulisk trykk brukes til forlengelsen av dem. Enkelte edderkopper kan hoppe en avstand som tilsvarer 50 ganger deres egne kroppslengde. Dette gjøres ved en rask blodtrykksøkning i beina.

Å bruke hydraulikk til bevegelse av beina ble vurdert som løsning. Mer om dette i delkapittel 3.1.4.

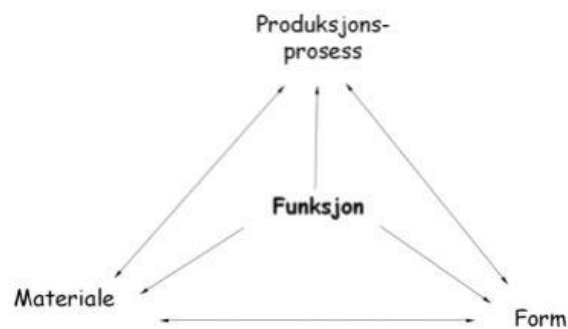


### 3.1.2 Metode

#### Produktutvikling

Produktutvikling handler om å skape, produsere og markedsføre produkter. I dette hovedprosjektet faller markedsføringsaspektet utenfor prosjektbeskrivelsen.

Produktutviklingstrekanter er vist på figur 12. Den plasserer produktets «funksjon» i fokus og ved bruk av piler representeres de ulike avhengighetsforholdene som inngår mellom funksjon, materiale, produksjonsprosess og form. Funksjon er bestemmende for alle de andre forholdene. Dette synliggjøres ved at pilene peker én vei. Form, materiale og produksjonsprosess har et gjensidig avhengighetsforhold, som illustrert ved at pilene peker begge veier.



Figur 12. Produktutviklingstrekanter.

Det finnes en rekke ulike produktutviklingsmetodikker. Felles for dem er at de skal gi en rettesnor i utviklingsarbeidet. Vi brukte metodikken som undervises ved NTNU i Ålesund. Fasene produktutviklingen går gjennom kan da oppsummeres i slik:

#### 1. Kunde og behovssituasjon

- a. Behovsundersøkelse (kunden i fokus)

#### 2. Produktet

- a. Produktprinsippfase (entydig produktdefinisjon)
- b. Produktutviklingsfase

#### 3. Produksjon

- a. Produksjonsforberedelsesfase
- b. Realiseringsfase



I begynnelsen av produktutviklingsprosessen står idéskapning i fokus. Man kommer opp med mange forslag og idéer gjennom idémyldring, som blir rikest ved bruk av divergent tenkning. Divergent tenkning kjennetegnes ved å være fleksibel, åpen, spontan og preget av frie assosiasjoner. Idévurdering er neste steg. Her blir resultatene fra idémyldringen målrettet og kritisk evaluert gjennom konvergent tenkning, som kjennetegnes ved at man vurderer all informasjon nøye og implementerer den beste løsningen.

For å lykkes med produktutvikling, trengs rikelig med tid i fase 1 og et godt definert produkt før en går over til utviklings- og produksjonsfasen (Hjeltnes, 2014). Produktet må bygges rundt kundens behov, og disse må kartlegges grundig før en tar avgjørelser om form, funksjonsprinsipper, materialer og produksjonsmetode.

Før utviklingen og utformingen av produktet starter må en entydig produktdefinisjon være på plass. Uten dette grunnlaget vil produktet koste betraktelig mer å utvikle og man risikerer å ikke møte kundens behov. Endringer i produksjonsfasen er langt mer kostbart å implementere enn i utviklingsfasen, derfor er det viktig å evaluere alle aspekter ved produktet og produksjonsmetoden kritisk underveis. Ved målbevisst og kritisk evaluering av konsepter unngår man ekstra tid- og kostnadsbruk.

### **Rapid Prototyping**

Rapid Prototyping er en prosess som bygger på en rekke teknikker som gjør det mulig å utvikle fysiske prototyper raskt. Prototypene er basert på modeller fremstilt med programvare for dataassistert konstruksjon (DAK). Rapid Prototyping gjorde det mulig å bekrefte at løsningene fungerte som tenkt, samt gjorde kommunikasjon av ideer og løsninger med andre i prosjektgruppa enklere.

### **Siemens NX 10.0**

Siemens NX10.0 er programvare som ble brukt i forbindelse med utforming av designet til konstruksjonen. Modeller og prototyper ble en sentral del av prosjektet, ettersom det var viktig å simulere og teste ideer og mulige løsninger tidlig. NX muliggjorde diverse analyser som styrke- og volumanalyser, samt utførte fysikkbaserte simuleringer. Siemens NX10.0 ble brukt som hjelp til å utføre oppdriftsberegninger på konstruksjonen. Mål av volum og simulering av bevegelsen er to eksempler.

### **Frihåndstegning**

Utover Siemens NX10.0 ble det skissert mye for hånd, ettersom frihåndstegninger er svært raske og enkle å fremstille, og har stor verdi i en tidlig fase av konseptutviklingen. Å produsere enkle og gode skisser som forklarer form og funksjon er et viktig verktøy som gir effektiv ideformidling til resten av gruppa. Frihåndstegningene dannet utgangspunkt for 3D-tegningene, som igjen ble eksportert til 3D-printere (Ultimaker 2+, Makerbot Replicator 2 og 2X) for prototypebygging.

## Litteratursøk

Det ble utført et litteratursøk med søkemotoren Google Scholar. Litteratursøket var viktig for å ikke isolere problemstillingen, men se at den inngår i en større sammenheng. Et litteratursøk bidro til bevisstgjøring rundt denne sammenheng. Litteratursøket ga et innsyn i dagens situasjon, både med tanke på hvilke problemstillinger andre er opptatt av, og hvordan utfordringer har vært løst tidligere.

Søkeord inkluderte: ROV, micro-ROV, AUV, micro-AUV, observation class ROV, biomimetic, biorobotics, biologically inspired robots, biologically inspired robotics, insect gait, tripod gait, hexapod og hexapod gait.

En viktig ressurs var nettstedet Ask Nature<sup>3</sup>, et online bibliotek som gir oversikt over mer enn 1800 naturfenomen som har praktiske anvendelsesområder for designere, ingeniører og forskere. Nettstedet tilbyr gratis informasjon via en søkefunksjon der man kan stille spesifikke spørsmål (eksempelvis «how does nature reduce drag?» eller «how does nature land softly?»).

## Direkte observasjon

Som inspirasjon kjøpte man inn og en frossen krabbe, samt hentet en levende strandkrabbe fra fjæra som ble filmet og observert grundig. Det ble arrangert en tur til Atlanterhavsparken i Ålesund for å studere krabber, hummere og kråkeboller i et kontrollert miljø. Stillbilder og filmopptak fra turen til Atlanterhavsparken ble studert og brukt i forbindelse med utviklingen av konstruksjonen.

## Intervju

I løpet av prosjektet ble det foretatt tre dybdeintervjuer med oppdragsgiver Per Erik Ose, for å sikre at produktutviklingen gikk i en retning der behovene til Kystverket ble ivaretatt. I produktutvikling er det viktig å bygge produktet rundt kundens ønsker og behov.

## Utstyr<sup>4</sup>

*Ultimaker 2+ 3D printer:* Ultimaker 2+ er en forbedret utgave av Ultimaker 2, utviklet av det nederlandske firmaet med samme navn. Firmaet er opptatt av open source bevegelsen - både hardware og software kan kopieres og modifiseres av brukerne. Printerens kan printe plasttypene PLA, ABS og CPE. ABS plast har en tendens til å krympe når den kjøles underveis i printen, men dette kan kompenseres for ved bruk av varmeplate.

*Makerbot Replicator 2 og 2X 3D-printer:* Replicator 2 printer PLA plast. Replicator 2X printer i tillegg ABS.

*Multicam 3000 fresemaskin:* Automatisert freser i tre akser (x,y,z).

*Epilog Helix:* Luftkjølt CO2-laser som kan kutte tre, akryl, plastikk, skinn og en rekke andre materialer. Kan kutte opp til 9,8 mm tykt plexiglass. Effekt: 65 watt.

*Houndsfield Tensometer:* Helmekanisk strekkprøvemaskin produsert av Tensometer Limited. Egner seg godt til å strekkteste mindre prøvestaver. Tensometeret måler kraft i kg fra 0-2000 på en kvikksølv søyle og tegner en graf med verdier underveis.

---

<sup>3</sup> <http://www.asknature.org>

<sup>4</sup> Utstyret beskrevet befinner seg på NTNU i Ålesunds verksted og FabLab med mindre annet er angitt.

*BEMATO BMT 430X1100 Dreiebenk*: Dreiebenk på NTNU i Ålesunds verksted.

*Siemens NX10.0*: Høyt avansert CAD/CAM/CAE verktøy. Anvendes ved oppgaver som design, simuleringer og styrkeberegningsanalyser.

*Luxion Keyshot*: Sanntids raytracing som bruker fysisk korrekt gjengivningsmotorikk til å generere hyperrealistiske bilder av 3D-modeller (Luxion Keyshot, 2016). Raytracing er en algoritme som regner ut optiske effekter som refleksjon og refraksjon.

### 3.1.3 Materialer

Bruksområde og pris var bestemmende for materialvalgene. For å holde kostnadene nede og gjøre reservedeler lett tilgjengelige, ble det besluttet å bruke 3D-printede komponenter i plast. De siste årene har det kommet flere typer filament til 3D-printere. NTNU i Ålesund sine 3D-printere er kompatible med ABS, PLA og CPE. Kystverkets 3D-printer har også mulighet til å printe PETG.

Ettersom konstruksjonen skal operere i et marint miljø er det viktig å bruke materialer som ikke kan forurense økosystemet. Det var nødvendig å undersøke om det lot seg gjøre å bruke materialer som fremstilles på en miljøvennlig måte for å ivareta Kystverkets visjon om å stå for bærekraftig utvikling av kystsonen.

For å finne plastmaterialet med egenskaper egnet for Kystverkets behov, ble det undersøkt fordeler og ulemper med samtlige plasttyper kompatible for de 3D-printere prosjektgruppa hadde tilgang til.

#### ABS

ABS er det eldste filamentet på markedet og det samme materialet som brukes i LEGO. ABS er en termoplast (tåler høye og lave temperaturer) og egner seg godt til mekaniske deler. ABS har en glasstemperatur på 105 °C og et smeltepunkt på ca. 230-260 °C. ABS har en tendens til å krympe når den kjøles, noe som i verste fall medfører deformasjon eller at de ulike lagene (*layers*) i printen henger dårlig sammen. Deformasjon kan unngås ved å bruke varmeplate. Krymping kan videre reduseres ved at printeren er utformet som et lukket rom slik at varmen stenges inne. Derimot egner ABS-plast seg godt til bruk i sjøen og har ønskede styrkeegenskaper.

Tabell 1. Styrker og svakheter ved ABS.

Styrker ved ABS	Svakheter ved ABS
<ul style="list-style-type: none"> <li>• forvitrer ikke</li> <li>• god formbestandighet og slagfasthet</li> <li>• olje- og fettbestandig</li> <li>• enkelt å resirkulere</li> <li>• beholder styrken ved høye temperaturer</li> </ul>	<ul style="list-style-type: none"> <li>• krymper ved avkjøling</li> <li>• petroleumsbasert</li> <li>• aceton brukes til overflatefinish</li> <li>• kjemisk ustabil</li> <li>• kan produsere skadelige avgasser under printing</li> </ul>

## PLA

PLA-filamentet er biologisk nedbrytbart og ble lansert som et brukervennlig alternativ med en miljøvennlig profil. PLA er også kjent som bioplast og brukes mye som matemballasje.

PLA har en glasstemperatur på 60-65°C og et smeltepunkt på ca. 160-220°C. Ettersom PLA fremstilles av biologiske råvarer (hovedsakelig stivelse og melkesyrer) er det mindre belastende for miljøet enn plast fremstilt av olje (Jørgensen, 2016). Med tanke på at PLA er et bærekraftig plastmateriale ville det vært ideelt til bruk i konstruksjonen.

Tabell 2. Styrker og svakheter ved PLA.

Styrker ved PLA	Svakheter ved PLA
<ul style="list-style-type: none"> <li>• miljøvennlig materiale</li> <li>• mindre utsatt for uønsket krymping</li> <li>• ingen skadelige avgasser under printing</li> <li>• gode printegenskaper</li> <li>• krever mye overflatebehandling enn ABS</li> </ul>	<ul style="list-style-type: none"> <li>• kloroform brukes til overflatefinish</li> <li>• blir sprøtt med tid</li> <li>• ikke like sterkt som ABS</li> </ul>

## CPE

CPE, også kjent som PET eller UPET, er et sterkt materiale som er mer brukervennlig å printe enn ABS. Det frigjør ingen skadelige avgasser under printing. Det er kjemisk stabilt og kan følgelig ikke overflatebehandles med kjemikalier for å forbedre overflatefinishen. CPE kan printe gjennomsiktige komponenter, men da anbefales det å bruke tykke layers. CPE trenger varmeplate under printing, ettersom printen kan miste festet til underlaget på grunn av krymping.

Tabell 3. Styrker og svakheter ved CPE.

Styrker ved CPE	Svakheter ved CPE
<ul style="list-style-type: none"> <li>• lett å printe</li> <li>• like sterkt som ABS</li> <li>• ingen avgasser under printing</li> </ul>	<ul style="list-style-type: none"> <li>• kan ikke overflatebehandles kjemisk</li> <li>• krymper under printing</li> <li>•</li> </ul>

## PETG

PET er en 100% gjenvinnbar termoplast fra polyestefamilien som brukes i alt fra røntgenfilm til brusflasker. PETG er en glykolbehandlet PET-plast. Smeltepunkt ca. 230-250 °C. Printeegenskapene er tilnærmet like gode som PLA-plast, og kan styrkemessig sammenliknes med ABS. PETG-plast lar brukeren printe gjennomsiktige produkter.

Tabell 4. Styrker og svakheter ved PETG.

Styrker ved PETG	Svakheter ved PETG
<ul style="list-style-type: none"> <li>• miljøvennlig</li> <li>• kjemisk stabil</li> <li>• gode printeegenskaper</li> <li>• høy hardhet og slagfasthet</li> <li>• fleksibelt materiale</li> <li>• 3-10 ganger sterkere enn PLA</li> <li>• krymper lite</li> </ul>	<ul style="list-style-type: none"> <li>• printer på NTNU i Ålesund printer ikke PETG</li> </ul>

Etter en gjennomgang av materialene tilgjengelige for prosjektgruppa ble det valgt å bruke ABS.

### Feste- og tetningsmidler

Skruer og rampamuffer (innskruingsmutter) kjøpes inn fra Tingstad. Tingstad er Norges største leverandør av festeteknikk og tetninger med hovedlager i Ålesund. NTNU i Ålesund har kundekonto som gir rabatter til studenter. En fordel med å velge unbrakoskruer er at de produseres med ulik utforming på skruehodet (konisk, avrundet og sylindrisk). Rampamuffer er en innskruingsmutter med både utvendige og innvendig gjenger. De utvendige er grove og de innvendige er metriske. De er mye brukt i møbelindustrien. Mer om hvordan rampamuffere ble brukt i konstruksjonen i avsnitt 3.1.4.



Figur 13. Unbrakoskruer.

(Unbrako Group, 2016)

Innskruingsmutter er en mutter med utvendige gjenger som gjerne anvendes i tremateriale for å ikke skade materialet. Vi benyttet oss av innskruingsmuttere, fordi det samme prinsippet kan gjelde for plastmaterialet, slik at man kan skru av og på de komponentene som har behov for dette.

For å gjøre konstruksjonen tett brukes pakninger og o-ringer. Pakningspapir, asbestfritt pakningmateriale og gummiduk fra Biltema var alternativer vi vurderte. Biltemas pakninger leveres i de målene som angis i tabell 5.

O-ringer kan anvendes som dynamisk og statisk tetning. Ved riktig anvendelse har o-ringer lang levetid og gir lavkostnadstetninger. Vi bruker o-ringer fra Otto Olsen, Norges største leverandør av o-ringer. Fra deres nettsider<sup>5</sup> hentet vi mål til standardiserte o-ringer.

Tabell 5. Oversikt over pakningsmateriell.

MATERIAL	PAKNINGSPAPIR	ASBESTFRITT PAKNINGSMATERIALE	GUMMIDUK
TYKKELSE [MM]	0,40	0,40	1,00
	0,75	0,80	2,00
		1,00	6,00
		1,50	

Vi gikk til innkjøp av 1 mm tykk gummiduk og 0,8 mm asbestfritt pakningsmateriale for å teste hvor lett de ulike materialene lot seg tilpasse ved hjelp av FabLabs Epilog Helix laserskjærer. Figur 14 illustrerer resultatet. Mer om prosessen i senere avsnitt.



Figur 14. Gummipakning og asbestfritt pakningsmateriale.

## Kjøling

Til våre kjølematerial kjøper vi aluminium av Smith Stål<sup>6</sup>. Smith Stål er en leverandør av ulike metall med avdeling i Ålesund. Aluminium ruster ikke og har god varmeoverføringsevne.

## Kameravindu

Pleksiglass brukes som vindusmateriale i forbindelse med front- og ryggekamera. Pleksiglass er en gjennomsiktig og knuseresistent termoplast. En plan flate foran keralinsen er dessuten å foretrekke sammenlignet med en kuppel. Mer om alternativet vi utforsket i avsnitt 3.1.4.

<sup>5</sup> <http://www.oo.no/admin/common/getImg.asp?FileId=2013>

<sup>6</sup> <http://www.smith.no/>

### 3.1.4 Resultat og diskusjon

De første utkastene til skrogdesignet ble basert på strandkrabben. Deretter ble designet endret til å ligne en kråkebolle. En senere versjon hentet inspirasjon fra en hummer. Optimal plassering av komponenter med tanke på kjøling og tyngdepunkt var bestemmende for det siste designet. Beina ble utviklet med tanke på funksjon først og form etterpå. En utfordring var at de måtte være tette.

## Skrog

### Skrogdesign 1

Det første utkastet ble utformet for å gi et overslag av konstruksjonens størrelse og danne et springbrett for videre tenkning. På dette stadiet var det uvisst hvor stor plass elektronikken ville ta i konstruksjonens skrog, men konseptutviklingen var preget av avgjørelsen fra Automasjon om å bruke vekselstrøm. To strømforsyninger utgjorde en tidlig dimensjonerende faktor med tanke på skrogets volum.

Trimaran-designet ble valgt for å lede vannstrøm inn i thrusterne for maksimal effekt. Inspirert av fartøyet i delkapittel 2.4, ble det besluttet å bruke tre thrustere. Kameraet var tenkt plassert i trimaranens «midtbjelke».

Detaljer som håndtak for bæring og kameraåpning, ble bevisst ikke modellert for å spare tid. Nytteverdien av å modellere detaljer på tidlige utkast ble vurdert som lav. Ved å følge samme resonnement ble det bestemt å ikke utvikle fysiske prototyper av hvert utkast av skroget, men vente til designet var mer ferdigstilt.



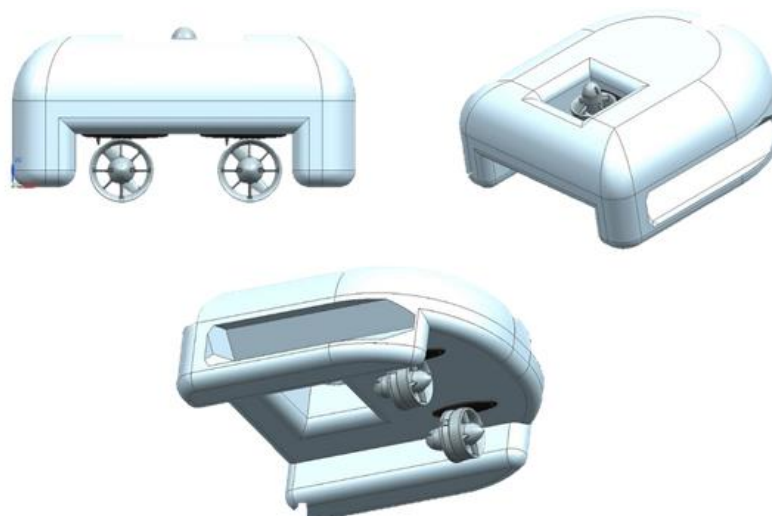
Figur 15. Skrogdesign 1 sett forfra og skrått fra siden.

Tabell 6. Styrker og svakheter ved skrogdesign 1.

Styrker ved skrogdesign 1	Svakheter ved skrogdesign 1
<ul style="list-style-type: none"> <li>• enkelt design</li> <li>• thrusterer får god vanngjennomstrømning</li> <li>• front-thruster vil kunne kompensere for eventuell «stupebevegelse» som følge av at thrusterne bak kjøres</li> <li>• danner et utgangspunkt man kan jobbe videre med</li> </ul>	<ul style="list-style-type: none"> <li>• gir tomrom det er vanskelig å utnytte</li> <li>• trolig for lite plass til å plassere alle komponentene</li> <li>• plassering av front-thruster for vertikal forflytning er ikke sentrert, noe som medfører vanskeligheter med å heve konstruksjonen</li> <li>• utarbeidet før det var etablert et godt informasjonsgrunnlag fra automasjon (kunnskap om hvilke elektroniske komponenter som skulle brukes)</li> <li>• plane flater på konstruksjonens sider gir overflater som vannet kan skyve på</li> </ul>

## Skrogdesign 2

Det generelle fokuset lå på å eliminere tomrom. Den kuppelformede oversiden er fjernet for å redusere motstandskraft fra vannet og ubenyttet tomrom. Trimarandesignet ble endret til et katamarandesign, og det ble laget skrå innfestning til beina på konstruksjonens langside, slik at beina ville stå i en vinkel. Et skrått innfestningspunkt for beina ville gjøre fartøyet mer naturtro, ettersom strandkrabbens bein står skrått ut fra krabbens kropp. Skrå plassering av servoene medførte ulike knekkpunkt på vaierne. Dette skapte uønsket friksjon og kunne dessuten medføre slitasje på strømpen som skulle tette beinkonstruksjonen. Dermed ble det besluttet å ikke ha skrå innfestning til beina på neste utkast.



Figur 16. Skrogdesign 2.



Tabell 7. Styrker og svakheter ved skrogdesign 2.

Styrker ved skrogdesign 2	Svakheter ved skrogdesign 2
<ul style="list-style-type: none"><li>• mindre ubenyttet tomrom</li><li>• færre plane flater som vannet kan skyve på</li></ul>	<ul style="list-style-type: none"><li>• store dimensjoner</li><li>• plassering av thruster for vertikal forflytning er ikke sentrert, noe som medfører vanskeligheter med å heve konstruksjonen</li><li>• uheldig med skrå innfestning til beina</li></ul>

### Skrogdesign 3

Tredje utkast er tungt inspirert av skallet til en strandkrabbe. Det er smalt i ene enden og skråner jevnt utover frem til fronten, slik det gjør på strandkrabbens ryggskjold.

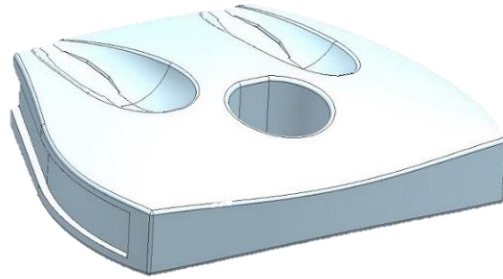
Det ble besluttet å flytte thrusterne fra undersiden til oversiden av skroget for å gjøre dem mindre utsatt for å treffe hindringer. Å flytte thrusterne frigjorde mye plass på undersiden til elektroniske komponenter og flyttet tyngdepunktet ned. En annen viktig endring som ble innført på dette stadiet var å redusere arealet som hadde plane flater.

Dette designet ble 3D-printet i en nedskalert versjon.

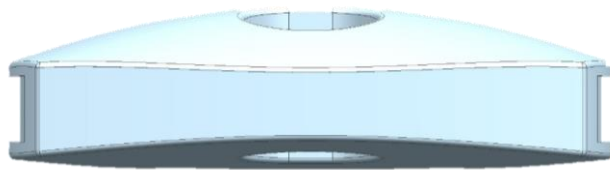
3D-modellen ble delt i to og den øvre halvdel ble frest ut i 1:1 skala. Dette fordi fresen kun freser i x-, y- og z-retning (ingen rotasjoner).



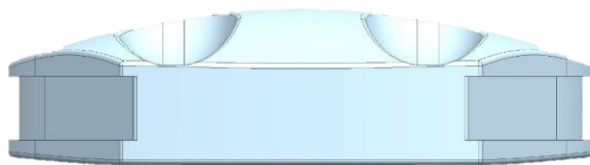
Figur 17. 3D-print av krabbeskall-designet.



Figur 18. Skrogdesign 3.



Figur 19. Skrogdesign 3 sett forfra.



Figur 20. Skrogdesign 3 sett bakfra.

Tabell 8. Styrker og svakheter ved skrogdesign 3.

Styrker ved skrogdesign 3	Svakheter ved skrogdesign 3
<ul style="list-style-type: none"> <li>• god utnyttelse av plassen i skroget</li> <li>• naturtro og estetisk tilfredsstillende utforming</li> <li>• mulighet til å plassere vekter på ulike steder i skroget for å påvirke tyngdepunkt og oppdrift</li> </ul>	<ul style="list-style-type: none"> <li>• stort skrog</li> <li>• fortsatt noen plane flater som kreftene i sjøen kan virke på</li> <li>• dårlig plassering av thrustere for horisontal forflytning. Vil kunne medføre en «stupebevegelse»</li> <li>• produksjonsmetode (støping) er tidkrevende</li> </ul>



Figur 21. Frest skalamodell av skrogdesign 3.

Måler 540 mm på det bredeste.

## Skrogdesign 4

Etter å ha utviklet et design basert på strandkrabben og skalert det slik at komponentene til Automasjon fikk plass, oppdaget vi ugunstige aspekter ved designet. Krabbedesignet hadde store plane flater som bølger kunne ta tak i. På samme tid besluttet studentene fra automasjon å gå over til likestrøm. Dette medførte bruk av elektriske komponenter som var mer kompakte, og størrelsen på skroget kunne reduseres.

Fremfor å nedskalere modellen besluttet vi å utvikle et nytt design. Til tross for at det var relativt sent i prosjektet, ga det en mulighet til å tenke nytt og angripe problemstillingen med en annen innfallsvinkel. Etter å ha jobbet med skrog med store dimensjoner var det spennende å utforske hvor liten konstruksjonen kunne lages ved å tenke helt nytt. Problemstillingen vi jobbet ut ifra var: «hvordan optimalisere designet med tanke på hydrodynamikk og funksjonalitet?».

En kule er den perfekte formen sett fra et hydrodynamisk perspektiv. Kreftene fra vanntrykket fordeles jevnt utover hele kulas overflate og det vil være minimal fare for turbulent strømning.

Konstruksjonen var inspirert av en kråkebolle og Hyball ROV. På kulas høyre og venstre side, ned mot undersiden, var det tenkt å plassere beina. Konstruksjonen hadde mål på 240 mm utvendig diameter og 200 mm innvendig diameter.

Skrogdesign 4 er formet rundt å ha innvendige batteripakker av type LiFePO-4 (brukes av OpenROV) etter ønske fra oppdragsgiver. Hovedsakelig fordi konseptet ikke forsynes med strøm gjennom batteri, og fordi det ville bli dyrt og krevende å fremstille en gjennomsiktig kuppel, kom ikke kuledesignet lenger enn til et tidlig stadie i konseptfasen. Overflaten på en gjennomsiktig kule er dessuten vanskelig å etterbehandle fordi en kule vanskelig lar seg spenne fast.



Figur 22. Utkast av skrogdesign 4.

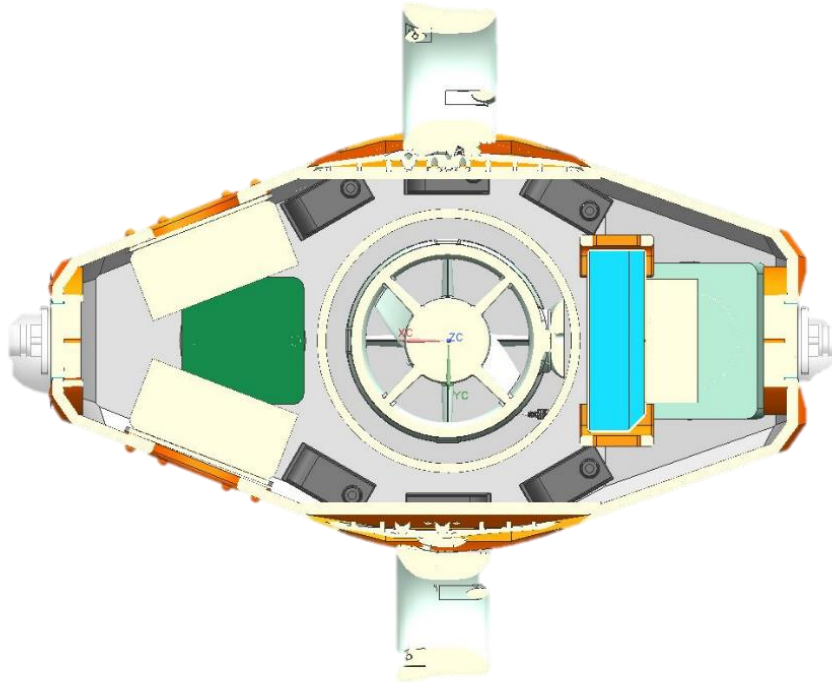
Tabell 9. Styrker og svakheter ved skrogdesign 4.

<b>Styrker ved skrogdesign 4</b>	<b>Svakheter ved skrogdesign 4</b>
<ul style="list-style-type: none"><li>• den optimale formen med tanke på hydrodynamikk og fordeling av trykk</li><li>• viste at det var mulig å lage en kompakt ROV</li><li>• inspirerte til videre utforskning av alternative design</li></ul>	<ul style="list-style-type: none"><li>• avhengig av riktig antall thrustere og god plassering av dem for å manøvrere</li><li>• har ikke nok volum til å motvirke vekten av komponentene og oppnå nøytral oppdrift</li><li>• utfordrende og dyrt å produsere</li></ul>

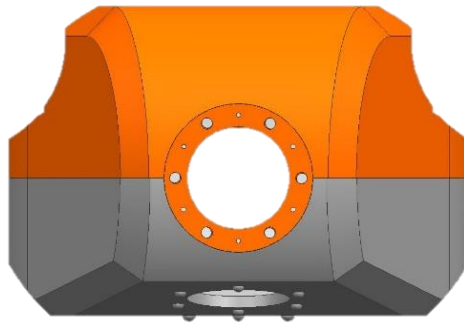
## Skrogdesign 5

Et utkast til det endelige designet ble utviklet etter å ha utført en behovsanalyse. Kystverket har først og fremst behov for en kompakt ROV. Da versjon 5 ble utviklet, startet gruppa med å plassere komponentene på en hensiktsmessig måte, for deretter å la denne plasseringen være bestemmende for skrogets utforming.

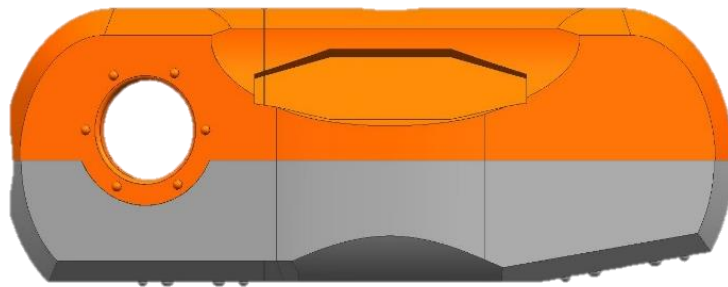
De tyngste komponentene ble plassert så langt nede i konstruksjonen som mulig for å gi et lavt tyngdepunkt. Et lavt tyngdepunkt vil bidra til å motvirke rotasjonsbevegelse (rulling) av skroget. Kameraet ble plassert ytterst for å få maksimal synsvinkel.



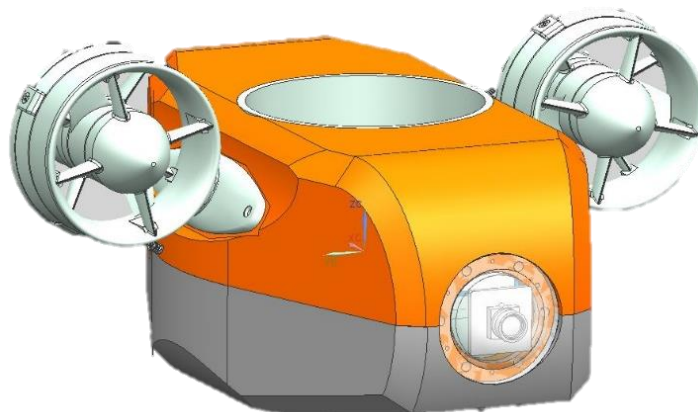
Figur 23. Sammenstilling av skrogdesign 5 med komponenter.



Figur 24. Skrogdesign 5 sett forfra.



Figur 25. Skrogdesign 5 sett fra siden.



Figur 26. Sammenstilling av skrogdesign 5 med thrustere og kamera.

Tabell 10. Styrker og svakheter ved skrogdesign 5.

Styrker ved skrogdesign 5	Svakheter ved skrogdesign 5
<ul style="list-style-type: none"> <li>• optimal bruk av plass: komponenter plassert så nært hverandre som mulig</li> <li>• god plassering av tyngdepunkt</li> <li>• god plassering av kamera</li> <li>• begynner å likne kommersiell ROV(!)</li> </ul>	<ul style="list-style-type: none"> <li>• beina står for midtstilt på fartøyet. Kan medføre ustabil gange</li> <li>• volumet gir ikke nok oppdrift</li> <li>• ikke optimalt utformet hydrodynamisk</li> <li>• kamerakuppel gir forvrengt bilde med mindre kuppelen er en oppskalert bue av kameralinsen</li> </ul>

## Skrogdesign 6

Skroget ble gjort mer hydrodynamisk ved å eliminere skarpe kanter fra forrige design. Ettersom noen komponenter utvikler mye varme og veier mye, etterlyste vi en ekstra gjennomgang av studentene fra automasjon. Vi ønsket at de utforsket muligheten for å bruke andre og/eller færre komponenter med lavere vekt. Resultatet av gjennomgangen var at én spenningsregulator ble fjernet. Dette fikk konsekvenser for effekten til horisontal-thrusterne (ble noe redusert). Komponentene ble plassert for å gi ønsket tyngdepunkt og symmetrisk utforming av skroget.

Oppdriftsberegninger på skrogdesign 5 avslørte at dimensjonene ga for lite innvendig volum. Resultatet dikterte at innvendig skrogvolum måtte tilsvare omtrent 8 liter for å gi nok oppdrift for at vertikale thrusteren skulle klare å løfte fartøyet i flymodus. Samtidig måtte det tas hensyn til at fartøyet trenger naturlig negativ oppdrift når beina er påmontert for at den ikke "svever". Mer om oppdrift i avsnitt senere.

Skroget består av to deler, en hoveddel og et deksel for innvendig inspeksjon og vedlikehold. Forsenkningene i lokket viser hvor unbrakoskruer festes. De er innfelte for å gi et mer appellerende utseende.

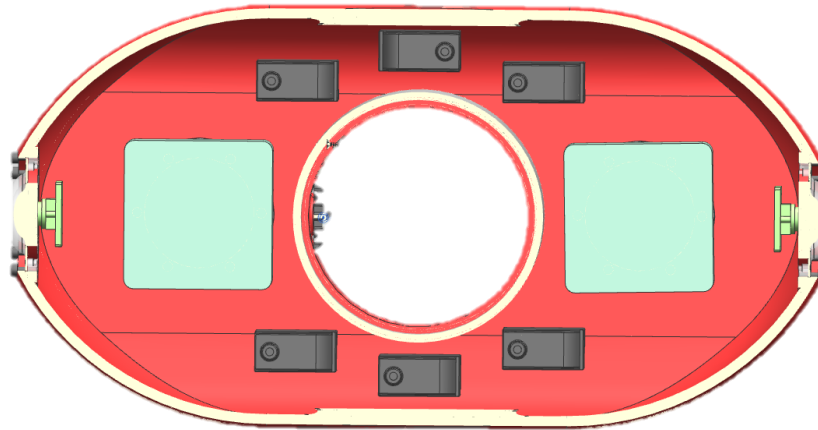
Hoveddimensjoner [mm]: L x B x H: 400 x 210 x 150



Figur 27. Skrogdesign 6.

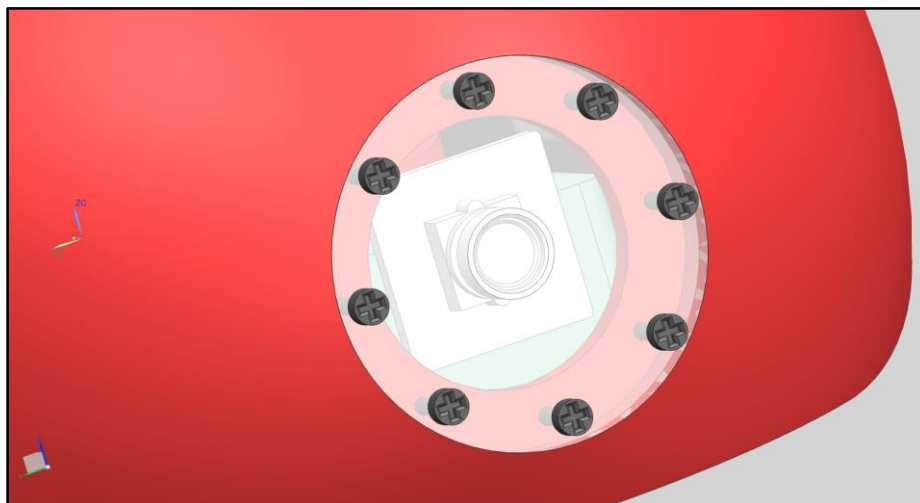
Vinklet 3D-gjengivning av skroget sett bakfra.

Den vertikale thrusteren er plassert i midten av konstruksjonen og festes direkte i skroget. Et aluminiumsrør og aluminiumsplater fungerer som et ekstra kjøleelement. I dekselet på bakenden plasseres en av kjøleplatene som skal kjøle en spenningsregulator. Åpningen bak er til et ryggekamera. På skrogets langside vises innfestningspunkt til thruster for horisontal forflytning.



Figur 28. Snittbilde av sammenstillingen til skrogdesign 6.

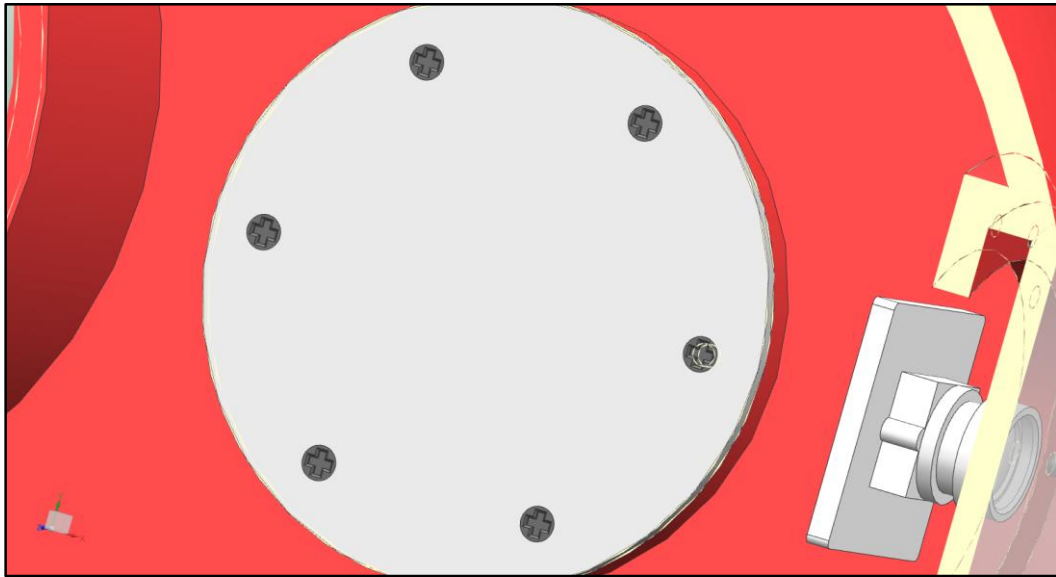
På figur 28 vises ny sammenstilling av komponentene plassert i skroget. Her synliggjøres det at skroget er romsligere enn det komponentene krever. Dette gir større handlingsrom til å kunne lage løsninger som for eksempel skruefestet til dekselet. Spenningsregulatorene (vist som de lyse kvadratiske boksene) er plassert symmetrisk for å gi lik stabilitet foran og bak i konstruksjonen.



Figur 29. Detaljbilde som viser kamera.

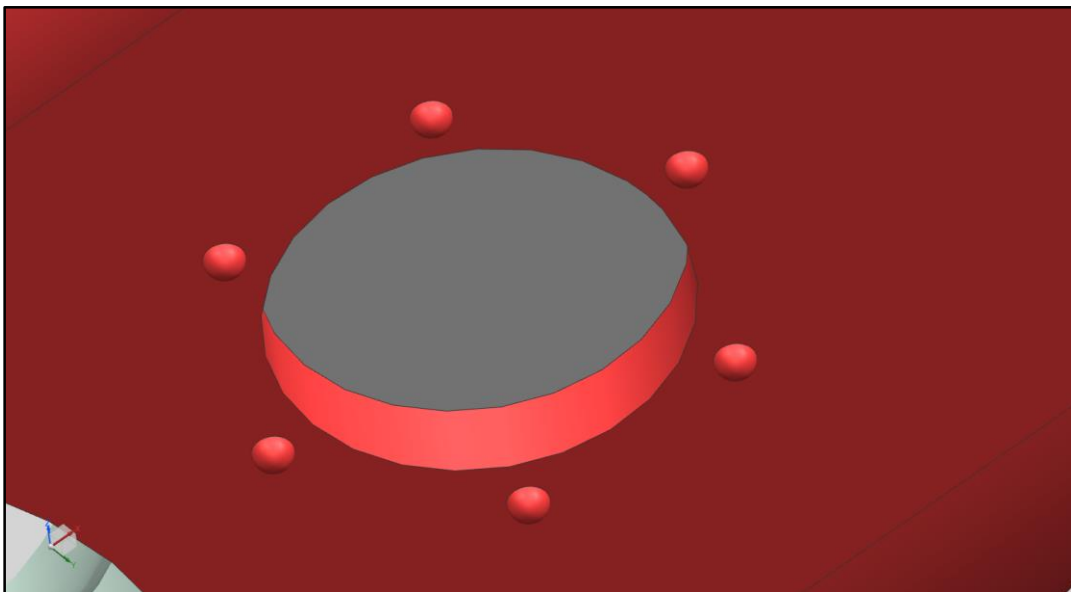
Pleksiglasset laserkuttet med en diameter på 65 mm og hull til innfestningsskruer. Pleksiglass brukes til å beskytte kameraet etter et forsøk med å benytte 3D-printe kuppel feilet.





Figur 30. Aluminiumsplate sett fra innsiden.

På utsiden av skroget hentes det inspirasjon fra skrognagler, som illustrert på figur 31. En slik utforming gir ekstra gods til skruene og gjør det tydelig fra utsiden hvor de er festet.



Figur 31. Aluminiumsplate sett fra utsiden.



Figur 32. Dekselets innfestningsmetode sett bakfra.

### Styrker ved design 6

- hydrodynamisk utforming
- symmetrisk om YZ-planet
- nok oppdrift

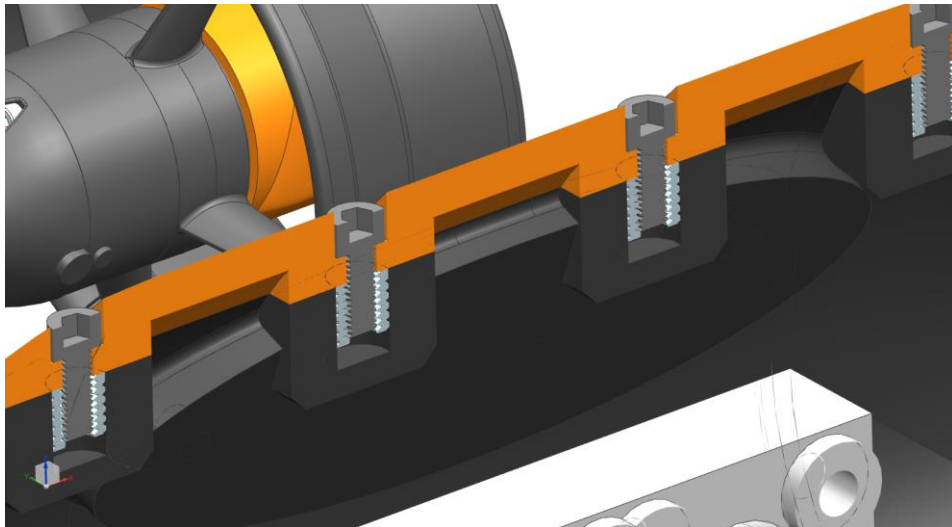
### Svakheter ved design 6

- buet skrog medfører vanskeligheter med å lage innfestning til beina
- mulighet for at konstruksjonen «stuper» når horisontal-thrustere kjøres for fullt (mulig at de trenger justering)

### Forslag til endelig skrogdesign

Det endelige designet er en forbedret utgave av skrog nummer 6. Viktige detaljer er modellert, som utformingen av innfestning til beina og inngang til strømforsyningen fra båten. På grunn av detaljendringer som hvor «umbilical-kabelen» er plassert, har vi snudd på hva som er foran og bak på konstruksjonen. Flere figurer av skroget finnes senere under avsnitt om Keyshot.

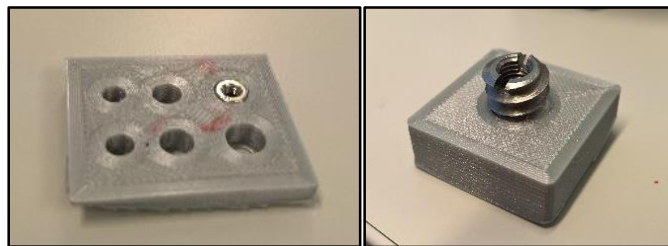
I sammenstillingene ble det montert skruer på konstruksjonen. I kameravinduet benyttes M3 unbrakoskruer med tilsvarende muttere på baksiden. Til å feste dekselet til skroget brukes innskruingsmuttere på M6 12x10 og M6 unbrakoskruer. Her brukes femten skruer for å feste, med mellomrom på 60 mm for å sikre at dekselet ikke deformerer rundt pakningen.



Figur 33. Snittbilde av skruerinnfestningen i deksel og skrog.

For å skru fast kjøleplatene og modulfestet, brukes innskruingmuttere på M4 8.5x10 og M4 innfelte skruer. Thrusterene festes med innfelte M3 skruer. I tillegg ble det lagt til 10 mm i høyden for å kompensere stupingen som kan oppstå ved høye hastigheter. Styrker og svakheter er de samme som hos skrogdesign 6.

Innskruingmuttere og sekskantmuttere ble testet i små 3D-printede klosser for å se hvilke dimensjoner de ulike skruerhullene skulle tilsvare. Dette ble gjort for å få god innfestning, samt unngå å skru direkte i skroget. Alternativet hadde vært å bruke skruer med grove gjenger, hvor man kan risikere å dra ut gjengene og måtte printe et nytt skrog. Tiltrekningsmomentet på skruene kan økes betraktelig på grunn av innskruingmutterne, som øker klemkraften fra dekslet på pakningen.



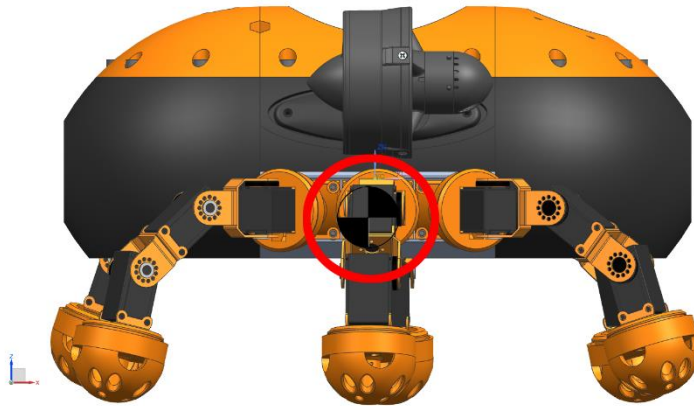
Figur 34. Mutter og innskruingmutter.

Aluminiumsrøret i midten ble fjernet for å forenkle konseptet. Det var usikkert om røret ville skape kondens eller ikke, og om det i det hele tatt var nødvendig.

Som et resultat av komponentplasseringen og den plane flaten på dørken fikk fartøyet et meget lavt tyngdepunkt (figur 35). Et lavt og sentrert tyngdepunkt vil stabilisere fartøyet, samt hindre at rotasjonskrefter overtar fartøyet. Ifølge NX «Measure Bodies», ligger tyngdepunktet (i forhold til aksene i 3D-modellen) ved:

- $x = 3 \text{ mm}$
- $y = 0 \text{ mm}$
- $z = -22 \text{ mm}$

X-koordinaten gir retningen fram- og bakover i fartøyet. Y-koordinaten tilsvarer til høyre og venstre, og z-koordinaten er høyden.



Figur 35. Sammenstilling av fartøyet med symbol for massesenter.

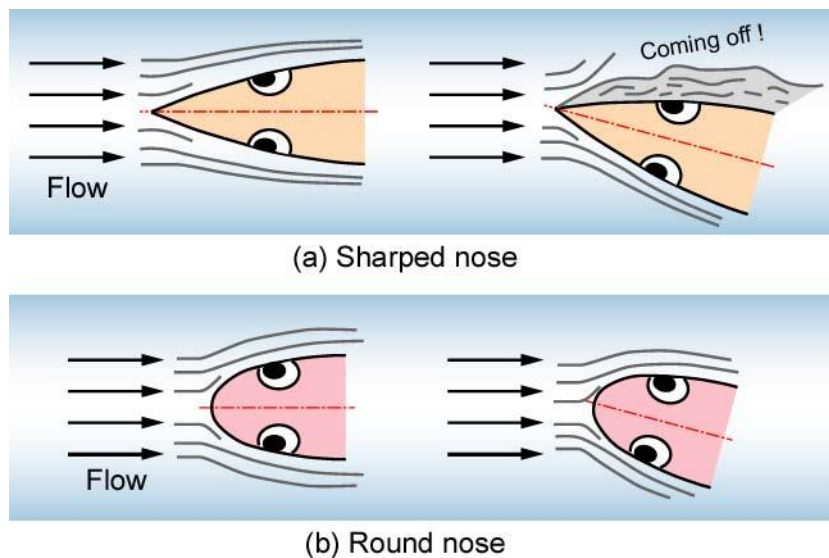


Figur 36. Sammenstilling av konstruksjonen med plassering av komponentene.

Plassering av komponentene var symmetrisk for å gi et lavt og sentrert tyngdepunkt (figur 36). Funksjonen "Measure Bodies" i NX målte skrogmaterialet (ABS) til 1.8 kg. Vekten til fartøyet ble 6.9 kg med alle komponentene, festemidlene, thrustere og bein.

## Form og sjømotstand

Vi plasserte komponentene slik at konstruksjonen fikk tilnærmet lik vekt foran og bak. Dermed ville tyngdepunktet befinne seg midt i konstruksjonen. «Measure Bodies» funksjonen, med «Associative»-egenskapen påslått, ga et resultat som forventet. Fra det hydrodynamiske synspunktet, vil en spiss front (som på figur 37) gli gjennom strømmingene, dersom strømmingene går langs skroget. En avrundet form vil medføre at strømmingene glir langs skroget og derfor skape mindre påkjenninger som skaper ubalanse.



Figur 37. Vannstrøm rundt ulike geometri.

(NMRI Japan, 2001)

En strømlinjet form gir en glatt overgang fra front til bakenden del av fartøyet, har ikke punkt hvor det skapes turbulent strømming, samt har liten drag-påkjenning. Det optimale er å ha en form som er spiss bak, men avrundet i front (dette gir minst mulig drag). Fordi oppdragsgiver ønsket kamera både foran og bak på fartøyet, ble skroget symmetrisk utformet med kameraene på samme høyde.

## Estimering av sjømotstand (drag)

### Formler

$$Re = \frac{uL}{\nu} \quad (1)$$

$$S_1 = \left(\frac{\pi D^2 L}{4}\right) \left(\frac{2}{3}\right) \quad (2)$$

$$S_2 = BH \quad (3)$$

$$F_D = C_D \frac{1}{2} \rho u^2 S \quad (4)$$

Reynolds tall (1)

Delegert areal S1 og S2 (2) (3)

Motstand i sjø (drag-kraft) (4)

Sjøtetthet: 1030 kg/m<sup>3</sup>

Reynolds tall,  $Re$ , brukes til å estimere hvordan strømninger påvirker skroget. En  $Re$  på over 2 300 betyr turbulent strømning.  $L$  [m] er den delegerte lengden,  $\nu$  [m<sup>2</sup>/s] er kinematisk viskositet og  $u$  [m/s] er farten til ROV. I Reynoldsberegningen brukes kun skroget.

Antatte fartsnivåer:

- 0.25 m/s
- 0.50 m/s
- 0.75 m/s
- 1.00 m/s
- 1.50 m/s

Den kinematiske viskositeten til sjøvann ved 0 grader angis som  $1.83 \times 10^{-6}$  m<sup>2</sup>/s. Kroppslengden på skroget er 0.4 m. For å regne ut en tilnærmet verdi for sjømotstanden til skroget, brukte vi det delegerte arealet: kameravinduet og resten (formel for S). Dragkoeffisienten var bestemt av at kameravinduets overflate var plan og at resten av skroget var strømlinjet. For beina forenklet vi overflatearealet og anså dem som kubiske overflater. Til slutt ble formelen for motstand brukt. Dragkoeffisienten for kameravinduet var 1.05 og dragkoeffisienten for resten av skroget var 0.09 (noe som tilsvarer dragkoeffisienten til en patron).

### Beregnet Reynolds tall og sjømotstand

Reynolds tall mellom  $0.5 \times 10^{-5}$  til  $3.3 \times 10^{-5}$ .

Sjømotstand på skroget varierer mellom 0.1 kg til 2 kg motstand (0.5 N – 18.4 N).

Sjømotstand på beina varierer mellom 0.2 kg til 8.8 kg.

Samlet sjømotstand på hele konstruksjonen varierer mellom 0.3 kg til 10.6 kg.

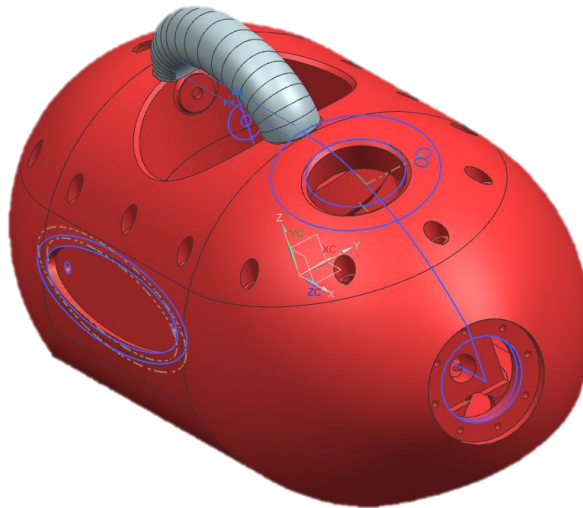
Basert på produsentens datablad for T100 har vi estimert ulike fartsnivåer fartøyet kan oppnå. Dermed kan resultatet avvike fra virkeligheten. Ut ifra overslagsberegningene, antas farten å kunne oppnå maksimalt 1 m/s.

Reynolds tall ble inkludert for å antyde hvordan strømningen vil oppføre seg rundt skroget. Vi oppnådde sammenliknbare verdier som Japans National Maritime Research Institute (NMRI Japan, 2001). Reynolds tallet forteller at det vil oppstå turbulente strømninger rundt skroget.

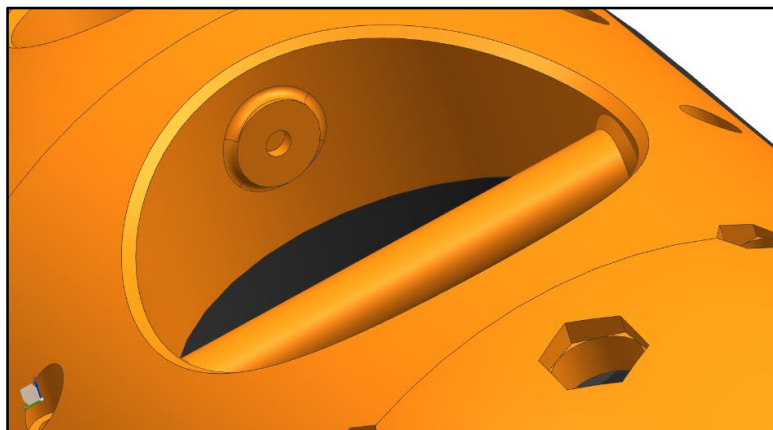
Dragkoeffisienten er avhengig av hvordan strømninger treffer skroget. En avrundet form vil gi en lav koeffisientverdi. Det var bestemt tidlig i oppgaven at man ønsket en «wide-angle fish-eye»-linse på kameraet. Derfor gikk vi for en flat overflate.

## Håndtaket

I midten av dekselet plasseres et håndtak som går tvers over skrogets midtseksjon. Grunnet lite tid ble det ikke fokusert mye på detaljene rundt håndtaket, men heller å gjøre løsningen enkel. Tidligere ble det vurdert å benytte et håndtak som strømforsyningen fra båten kunne tres gjennom. På figur 38 vises størrelsen til håndtaket, som er 30 mm. Håndtaket blokkerte for mye av thruster-tunnelen, og vi antok at dette ville hindre thrusteren å kjøre på maks effekt. Dermed endte vi opp med et tynnere håndtak som på figur 39.



Figur 38. Tidlig håndtak på dekselet.



Figur 39. Håndtak til skrogdesign 7.

## Valg av kjølemetode

Spenningsregulatorene i konstruksjonen avgir varme. Forventet arbeidstemperatur på full ytelse antas å ligge rundt 50°C (Amazon Inc., u.d.). Siden disse skal monteres i et lukket miljø, trenger regulatorene kjøling. Dermed besluttet vi med å bruke plater av aluminium, som kjøles ved direkte kontakt med sjøvann. Aluminium har god varmeoverføringsevne og er relativt lett.

Et usikkerhetsmoment knyttet til denne kjølemetoden er innvendig kondens. Kondens er svært uheldig og kan kortslutte de elektriske komponentene. Samtidig vil denne kjølemetoden medføre ekstra tetningspunkter, noe som øker sannsynligheten for at skroget lekker.

Som ekstra kjøling ble det vurdert å montere et aluminiumsrør der den vertikale thrusteren er plassert. Standarddimensjonene til røret: 110 x 3 x 100 mm

Røret medførte ekstra vekt (ca. 600 gram) og var vanskelig å implementere. Samtidig var vi usikre på hvilken nytte det ville medføre, som gjorde at vi gikk bort fra løsningen.

## Metode for å tette fartøyet

Det brukes hovedsakelig vanntette pakninger som tetningsmiddel. Disse tilpasses skroget ved å laserkattes ved FabLab på NTNU i Ålesund.

Siemens NX har en «*copy face*»-funksjon som gjør at man kan kopiere en valgt flate og eksportere den i 2D. Slik overføres helt korrekte mål til programmet laserskjærerens bruker. Uten laserskjærer hadde vi måttet bruke saks eller skalpell, samt linjal for å tilpasse pakningsmaterialet. Dette ville vært svært tidskrevende fordi man er avhengig av helt nøyaktige mål og ekstremt nøyaktig utførelse.

Innstilligene på Epilog Helix:

Funksjon: Vector cutting

Skjærehastighet: 25%

Effekt: 100%

Frekvens: 100 Hz

O-ringer ble brukt for å tette igjen deler som røret i midten og beinfesta. Størrelsene finnes i maskintegningene i vedlegg 5.

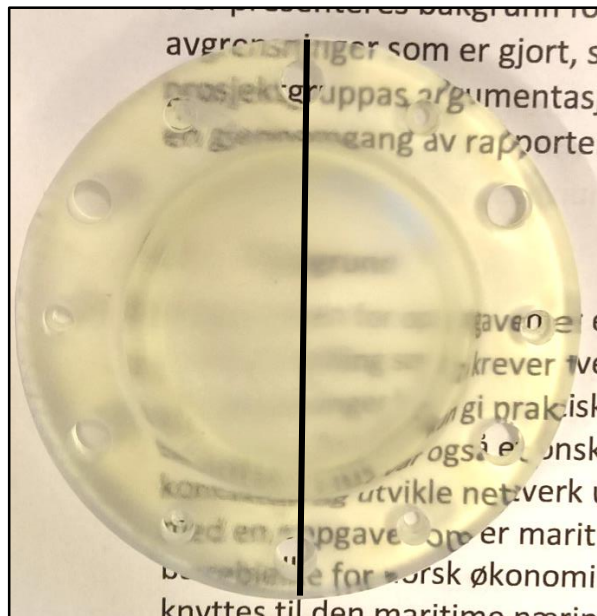


### Kamerakuppel i resin

Det ble undersøkt mulighet for å bruke egenprodusert kamerakuppel i konstruksjonen. Design og 3D-print av kuppelen ble vellykket, men etterbehandling av overflaten ga ikke et tilfredsstillende resultat. Overflateruheten ble redusert, men kuppelen ble ikke transparent nok til å kunne brukes.

Overflatebehandlingen fulgte anbefalingene til materialleverandøren Formlabs og inkluderte våtsliping med sandpapir med grovhetsgrad 600 og 1200, akrylrens og polering med mirkofiberklut. Til slutt avsluttes poleringen med klarlakk.

Venstre side er ubehandlet og høyre side er polert.



Figur 37. Kamerakuppel i resin.

## Bein

Konstruksjonen har seks bein. I tråd med biomimetiske prinsipper ble det undersøkt hva som er mest utbredt i naturen, og svaret er seksbeinte organismer (Semb-Johansson, 2015). Å ha seks bein fremfor åtte (slik strandkrabben har) ville dessuten bidra til å holde kostandene nede. Utover dette finnes det mye litteratur som bekrefter at seks bein muliggjør kontrollert, stabil bevegelse. Se tabell 11 fra Zu et al (Zu, 2011).

Tabell 11. Vurdering av ulike styrker ved gitt antall bein (Zu, 2011).

Evne til å opprettholde:	1 bein	2 bein	3 bein	4 bein	5 bein	6 bein	7 bein	8 bein
Stabil posisjon	Nei	Nei	Bra	Best	Best	Best	Best	Best
Statisk stabilitet	Nei	Nei	Nei	Bra	Best	Best	Best	Best
Dynamisk stabilitet	Ja	Ja	Best	Best	Best	Bra	Bra	Bra
Stabilitet under høyhastighets gange	Nei	Nei	Nei	Ja	Bra	Best	Best	Best
Enkelhet med tanke på DOFs	Best	Best	Bra	Bra	Bra	Ja	Ja	Ja

Som det fremkommer av tabell 11, vil en stasjonær robot være stabil med tre bein eller flere. En hexapod vil ha store variasjonsmuligheter med tanke på bevegelsen til beina. En vanlig gangart for seksbeinte organismer er imidlertid ikke å bevege alle seks bein uavhengig av hverandre, men å bevege dem parvis, tre om gangen. Slik vil tre av beina alltid har kontakt med underlaget. Denne måten å gå på er betegnet som tripod gangart («tripod gait») i litteraturen.

Prototypebygging ble satt i gang tidlig for å raskt kunne avdekke problemer med design og/eller funksjonsprinsipp. I motsetning til utvikling av skroget, var det kritisk å begynne med tidlig prototypetesting for å teste om beina og dens ledd fungerte som tenkt.

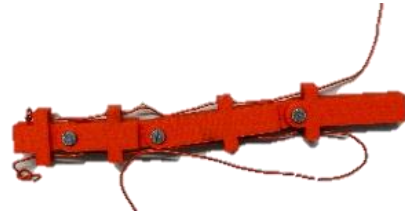
Professor Houxiang Zhang ved NTNU i Ålesund ble konsultert og hans anbefaling om å bruke vaier ble fulgt. Testing av prinsippet i praksis ble fokusert mye på for å avdekke feil.

## Prototyper fra Automatiseringsteknikk

De første prototypene brukte vaier og ble brukt som inspirasjon til videre utvikling av beina. Ved å 3D-printe ulike modeller fikk man avdekket problemer og en mulighet til å løse dem der og da ved å utvikle dem videre til mer avanserte prototyper. Leddene er montert sammen med skruer. Felles for de fire første prototypene er at de har tre innfestningsrører bolten må gjennom.

### Prototype 1.0

Den første prototypen ble utviklet ved hjelp av to kabler og Makerbot 3D-printer. Versjon 1.0 bestod av 4 komponenter (ledd). V1.0 viste funksjonsprinsippet men avdekket viktige feil. De største problemene bestod i at det ikke var mulig å kontrollere bevegelsen, og selvoppretting fungerte ikke.



Figur 40. Prototype 1.0

### Prototype 2.0

Forskjellen fra V1.0 består i utforming av mekaniske stoppere ved 45° utslag. Wireisolasjonen som ble brukt er mer slitesterk og fleksibel, men litt for tøyelig. Utfordringer som å kontrollere bevegelsen og selvoppretting er fortsatt tilstede.



Figur 41. Prototype 2.0

### Prototype 3.0

Den tredje prototypen kan låses mekanisk i vertikal retning. Det medfører at selvoppretting er mulig, gitt at beinets nøytrale posisjon er en rett linje. Det ble observert at det siste leddet starter bevegelsen slik at kontrollbarheten ikke fungerte som ønsket.

Dette på grunn av bruk av en wire. Det ytterste leddet har blitt statisk fordi rotasjonsbevegelsen den opprinnelig ga ikke bidro til å gi bedre fotfeste.



Figur 42. Prototype 3.0

### Prototype 4.0

Prototype versjon 4.0 skiller seg fra de foregående gjennom bruk av to vaier. To vaier fremfor kun en bidrar til kontrollert bevegelse av foten. Dermed kan det første og andre leddet styres uavhengig av hverandre. Wirene er plassert i en «u-sving» slik at endene kan festes til en servomotor.

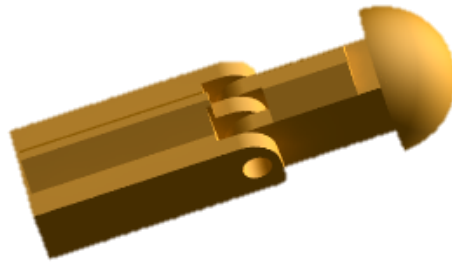


Figur 43. Prototype 4.0

## Prototyper fra produkt- og systemdesign

### Prototype 1.0

Prototype V1.0 utviklet av studentene fra POD bygger på de fire foregående prototypene med vaierprinsipp. På denne versjonen festes komponentene med en bolt gjennom fem innfestningsører, noe som gir en vridningssikker løsning. Selve foten har sfærisk utforming for at ROV skal ha fotfeste uavhengig av vinkelen til beina.



Figur 44. Prototype 1.0

### Skalamodell

Etter å ha forsikret oss om at funksjonsprinsippet fungerte, ble det laget en enkel papirmodell i 1:1 skala for å få et representativt inntrykk av den fysiske størrelsen. Formålet var å finne en lengde til beina i forhold til skroget. Dette resulterte i en lengde på 260 mm ekskludert innfestningsbrakett og fot.



Figur 45. Skalamodell.

## Prototype 2.0

Gravemaskiner har hydraulikk på oversiden av konstruksjonen. Det framstår som ryddig og samtidig beskyttende. Å ha alle endene til de ulike vaiere samlet er også positivt; det gjør det blant annet enklere å feste dem til servoene som står på innsiden av skroget. Vaiere kan ikke føres «tilfeldig» inn i skroget: det må finnes en ryddig struktur for å unngå knekkpunkt og friksjon. Derfor ble gravemaskin brukt som en inspirasjon.

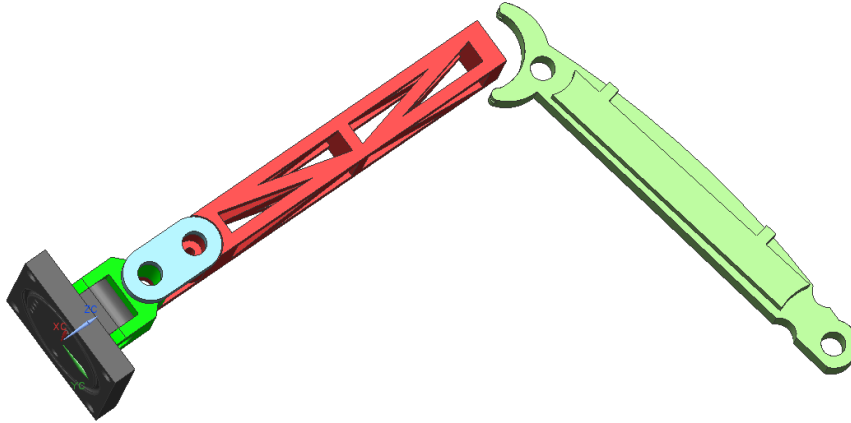
For å gjøre komponenten tett inntil skroget, ble det benyttet en o-ring. På innsiden av ringen er det seks åpninger til vaierne. De fire øverste åpningene brukes til å løfte de ytterste leddene, og de på sidene benyttes til å rotere leddet (de nederste ble brukt til å utforske horisontal eller vertikal plassering av servoer). Innfestningspunktet for vaierne i midten av beinet fungerte samtidig som en mekanisk stopper for bevegelsen. Beinets litt utslag men fungerte ellers som forventet. Prototypens lengde ble 330 mm.



Figur 46. Prototype 2.0 sett bakfra og sett fra siden.

### Prototype 3.0

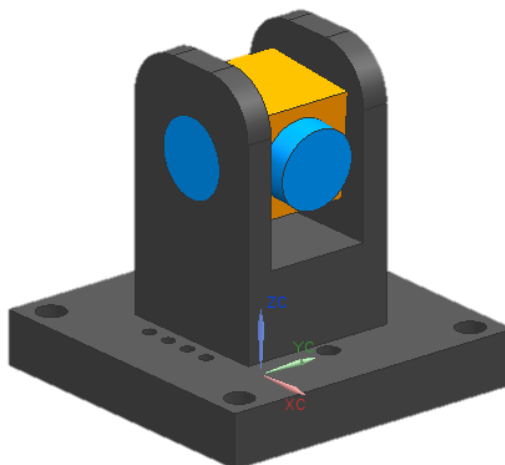
Formålet var å lage en mer hydrodynamisk utforming ved å tillate vanngjennomstrømning. Gjennomstrømningen antas å redusere friksjonen for konstruksjonen i flymodus, samt redusere vrimomenter i flymodus. I tillegg flyttes rotasjonen fra enden av beina til innfestningene. Prototypen ble aldri brukt fordi skroget endret form, noe som medførte at også beina måtte endres.



Figur 47. Prototype 3.0

### U-joint til prototype 3.0

Det lyseblå leddet på figur 47 tok mye plass. For å spare lengde ble det laget et kryssledd. Utfordringen med leddet var at vinkelen til utslaget var for lite og ideen ble forkastet.

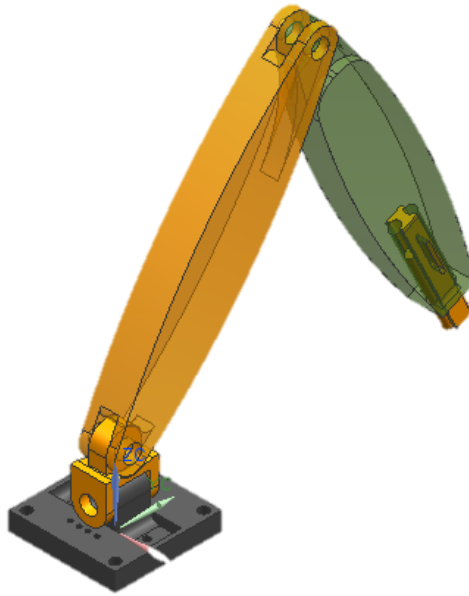


Figur 48. Kryssledd til prototype 3.0.

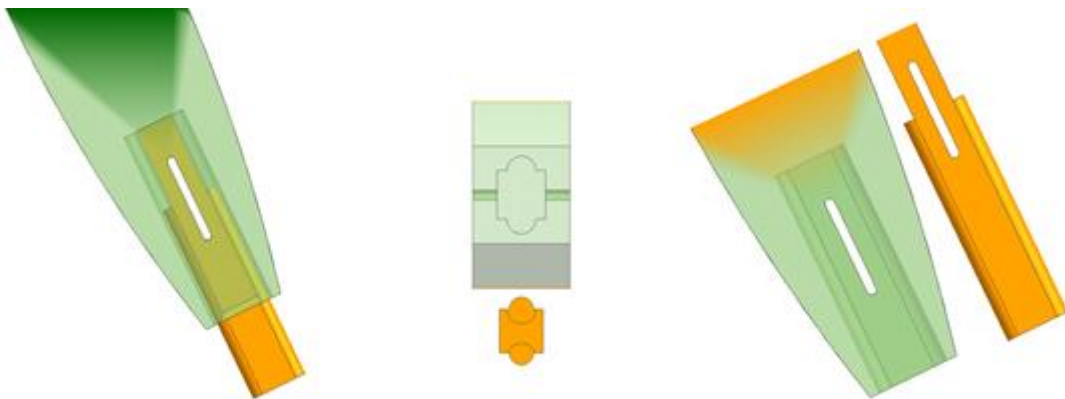
### Prototype 4.0

Bein V4.0 er mer inspirert av strandkrabben, ettersom det er ellipseformet. I enden er det en fjærbelastet støtdemper. Støtdemperen består av et kammer med to fjærer. Ved støtbelastning vil noen av kreftene bli tatt opp av disse. Planen var at beina skulle forme en spiss vinkel som illustrert på figur 49. Litt gods ble fjernet på selve innfestningen for å beholde svingradiusen.

Beinets 3D-print synliggjorde to feil. Kombinasjonen av det ovale sporet til låsesplinten og at begge fjærene lå løst medførte at støtdemperen ikke fungerte (se figur 50).



Figur 49. Prototype 4.0

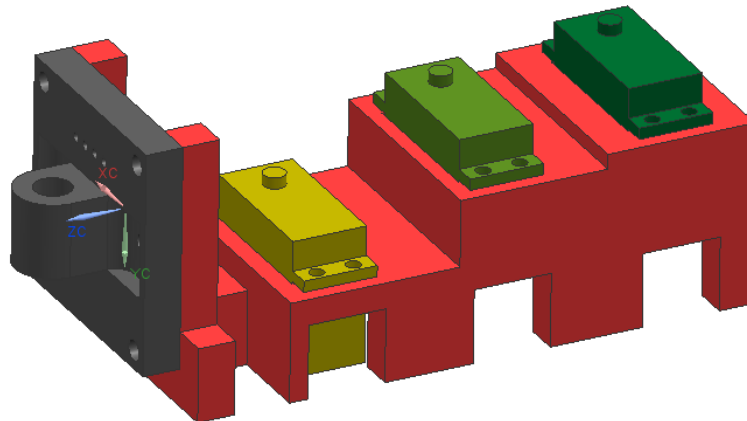


Figur 50. Støtdemper i enden av prototype 4.0.

## Servoholder

For å styre beina uten at servoene flytter seg, ble det laget en servoholder. Servoholderen ble utformet slik at friksjon ble redusert ved å unngå knekkpunkter for vaierne. Den første servoen styrte rotasjon. Den midterste løftet det innerste leddet. Den siste servoen hevet og senket ytterste leddet.

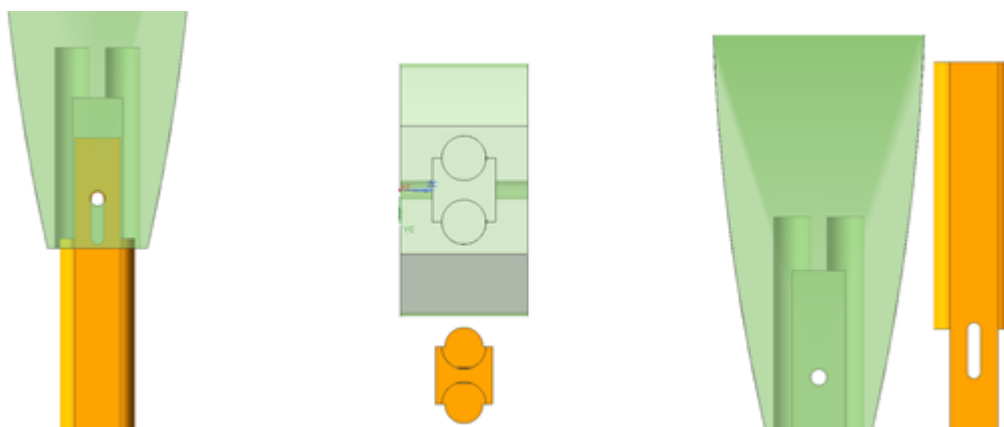
Vi printet to ulike eksemplarer. Den første viste hvordan vaierne ville oppføre seg, og den andre hadde en bedre løsning til strømkablene fra servoene.



Figur 51. Servoholder på innsiden av skroget.

## Prototype 4.1

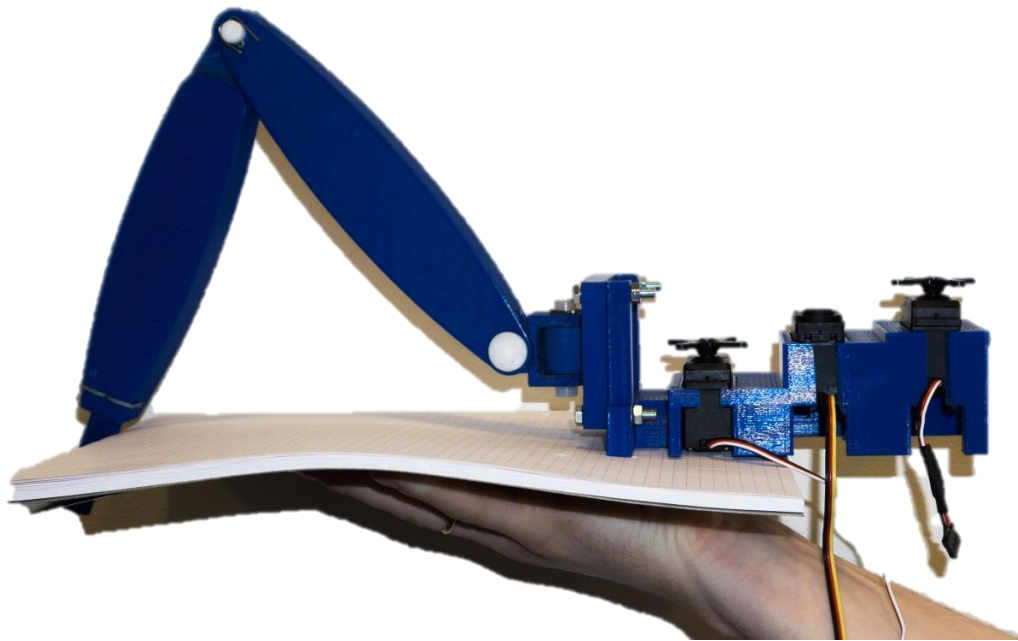
V4.1 har forbedret støtdemperfunksjon. Løsningen ble å dele kammeret til fjærene i to, slik at de stod adskilt og ikke kunne vikles sammen under kompresjon. Designet med to sylindriske spor fungerer som guide under kompresjon/ekspansjon av fjærene. Samtidig fjernet vi det ovale sporet i det siste leddet (se figur 52).



Figur 52. Forbedret støtdemper i enden av prototype 4.1.



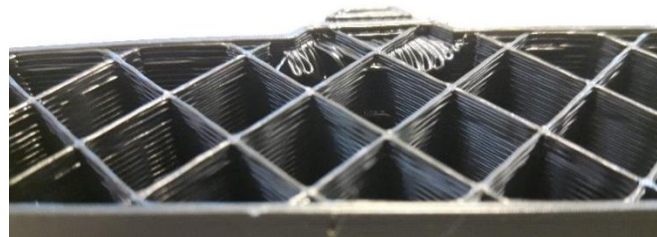
På figur 53 vises V4.1 og servoholderen. Akslingene i leddene har blitt maskinert i dreiebenk. Det ble brukt låsesplint til å holde beinet sammen.



Figur 53. Printet prototype 4.1 og servoholderen.

### Prototype 4.2

I prototype V4.2 ble det jobbet med sving- og løfteinnfestninger. Prototypen ble printet med standard innstillinger og 10% innfyll for å redusere ventetiden. 10% innfyll førte imidlertid til problemer når det skulle bores hull til innfestningspunkter. Det var for lite støttestruktur i foten og for mye tomrom.



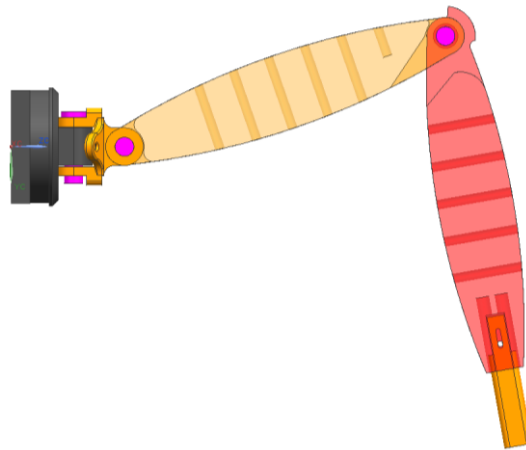
Figur 54. 3D-print av Makerbot med 10% innfyll.

### Prototype 4.3

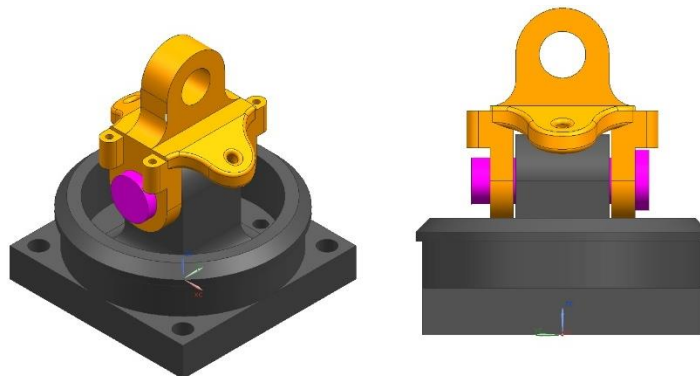
Prototype 4.3 var den første prototypen som ble testet med servoene. Testen avslørte at knekk på vaier virker uunngåelig. Beinet hadde noe hakkete og uforutsigbar bevegelse, men dette kan unngås ved at servoene kjøres saktere. Føringerne til vaierne i ledd for sving var plassert eksentrisk, og det medførte uønskede bevegelser under sving, samt uønsket heving og senkning av beinet (mekanismen som hevet beinet ble påvirket). Bruk av vaier krever at beina er vanntette, og en gummistrømpe (for eksempel sykkelsslange) var den beste løsningen med tanke på budsjettet. I den forbindelse har innfestningsbraketten fått en tetningsflate der strømpen skal festes.

Ledd 3 har fått en mekanisk stopper som hindrer fritt utslag og som fungerer som beskyttelse for servo og vaier når bevegelsen blir for stor.

Ledd 2 og 3 har ulike borer for å teste hvor innfestningspunktene bør ligge med hensyn til friksjon og vinkelen beina arbeider i.



Figur 55. Prototype 4.3.



Figur 56. Svingledd og innfestningsbrakett til prototype 4.3.

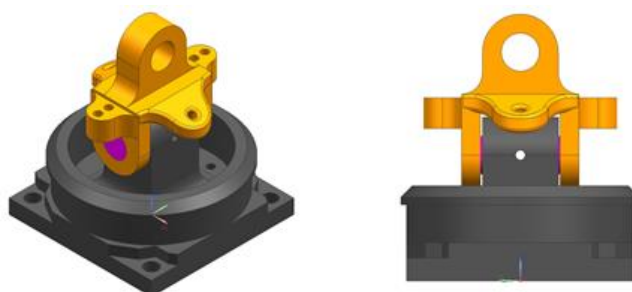
#### Prototype 4.4

V4.4 har endringer på leddet som gjør bevegelsen mer sentrisk (sving, samt heving og senkning). Akselbolten (lilla del på figur 56) måtte kortes ned for å gjøre dette mulig, som festes ved hjelp av en låsesplint til innfestningsbraketten. Skruene har blitt forsenket slik at hodet til skruene ikke påvirker tetningsflaten.

Som nevnt tidligere medførte bevegelsen til det gule leddet en påvirkning på det rosa leddet som var uønsket. Det ble vurdert å plassere en servo som styrer bevegelsen til det rosa leddet mellom det gule og det rosa. Dermed unngås eventuelle kompensasjoner fra de andre servoene når fartøyet beveges.



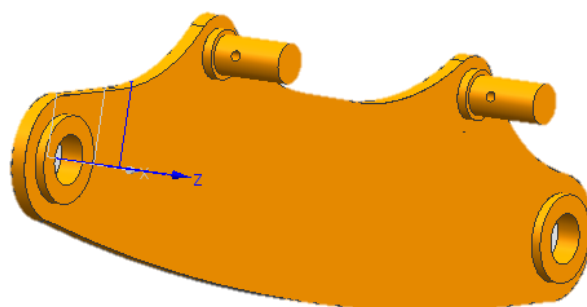
Figur 57. Prototype 4.4.



Figur 58. Svingledd og innfestningsbrakett til prototype 4.4.

### Prototype 5.0

Prototype 5.0 ble bygget opp som en sandwichkonstruksjon. En sandwichkonstruksjon har gode styrkeegenskaper samtidig som den sparer materiale. Derfor er den også mindre tidkrevende å 3D-printe. Printen av tidligere design tok opp til 10 timer per bein. Selve designet ble dessverre lagret under en «temporær»-mappe, noe som gjorde at modelleringsarbeid gikk tapt. Det finnes likevel et tidlig utkast av designet som illustrerer hovedprinsippet (se figur 59). På figuren kan man se føringer og innfestningspunktet til vaieren, samt festepunkt til leddet (de to åpningene nederst).



Figur 59. Sandwichkonstruksjon (halv) av prototype 5.0.

## Konsepter som ikke ble ferdigstilt



Figur 60. Deler av en flyvinge.

(RC Network De., u.d.)

Konstruksjoner der vannet kan strømme mest mulig fritt er å foretrekke. Inspirasjonskilden ble hentet fra luftfart, nærmere bestemt flyvinger. Flyvinger består av fagverk, noe som gir en lett og fleksibel konstruksjon.

Deler av flyvingen kan skjæres ut i for eksempel pleksiglass ved hjelp av Epilog Helix laserskjærer. Dette kan senke ventetiden betraktelig når det gjelder å lage prototyper (det er snakk om minutter kontra timer som 3D-printeren bruker).

Til å sammenkoble delene ble det vurdert lim, skruer, gummistrikker og vaier/snor. Dersom deler av beinet ødelegges er man avhengig av å ha verktøy tilgjengelig for å reparere det. Om det var mulig å bruke gummistrikker, hadde ikke det vært behov for verktøy i det hele tatt. Gummistrikken skal ikke ta opp krefter men kun låse/holde delene på plass. Det ble vurdert å bruke samme type gummistrikker som brukes til tannregulering, ettersom disse er små og svært slitesterke.

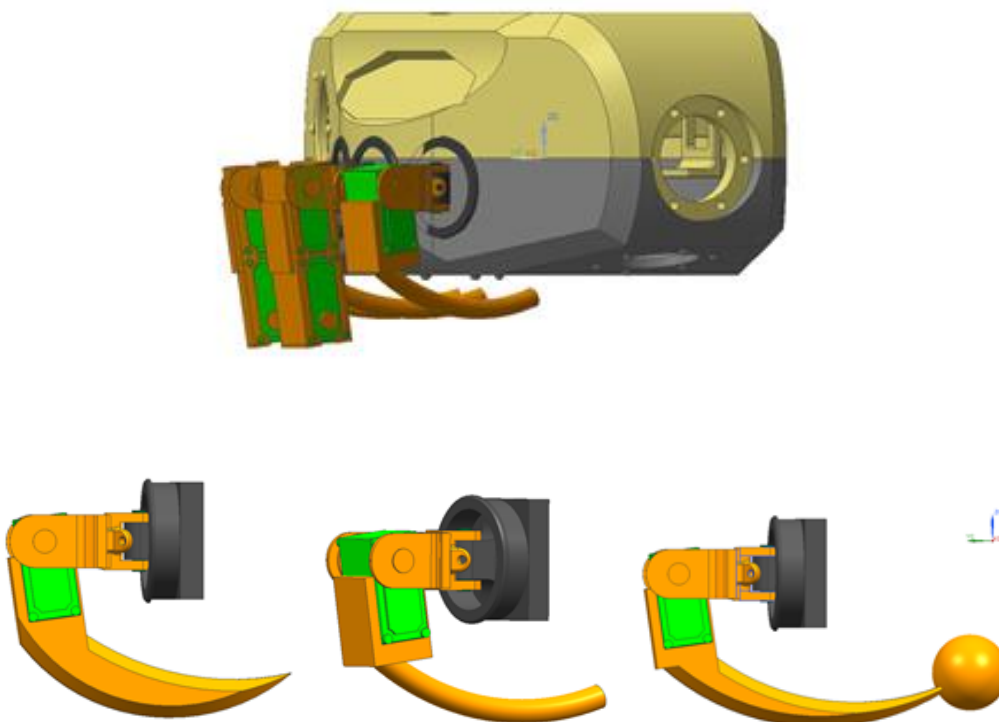
Designet ble forkastet fordi skroget for øvrig skulle reduseres. Dessuten hadde designet noen utfordringer på grunn av servoene på utsiden av skroget, som hadde trengt en ekstra tetning. Modellen ville bestått av utskjærte deler og en festebakett (samt tetningsbakett) til servoene.

## Skrogdesign 4

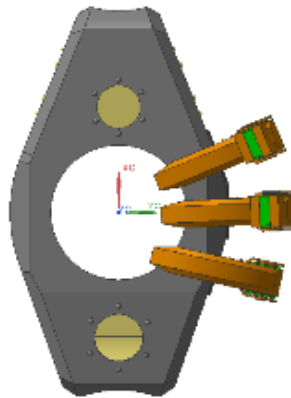
For å maksimere plassutnyttelsen i kula, var det ønskelig å ha seks servoer plassert innvendig i skroget og resten av servoene utenfor skroget. Kula hadde en kompakt størrelse og medførte utfordringer når det gjaldt utformingen av beina. Kuledesignet krevde en buet innfestningsbakett, god bakkeklaring og samtidig en hydrodynamisk utforming av beina når ROV-en er i flymodus. Det mest optimale hadde vært om det var mulig å følge kulens ytre linjer eller felle inn beina i kuledesignet. Det lyktes ikke med å finne en god løsning. På grunn av utfordringer knyttet til plass og vanskeligheter med å følge kulas utvendige linjer greide man ikke å inkorporere de valgte servoene i denne versjonen av beina. Til slutt var det nye designendringer på skroget som avsluttet prosessen med å utvikle bein til en kuleformet ROV.

## Skrogdesign 5

For å maksimere plassutnyttelsen i kula, var det ønskelig å ha seks servoer plassert innvendig i skroget og resten av servoene utenfor skroget. Kula hadde en kompakt størrelse og medførte utfordringer når det gjaldt utformingen av beina. Kuledesignet krevde en buet innfestningsbrakett, god bakkeklaring og samtidig en hydrodynamisk utforming av beina når ROV-en er i flymodus. Det mest optimale hadde vært om det var mulig å følge kulens ytre linjer eller felle inn beina i kuledesignet. Det lyktes ikke med å finne en god løsning. På grunn av utfordringer knyttet til plass og vanskeligheter med å følge kulas utvendige linjer greide man ikke å inkorporere de valgte servoene i denne versjonen av beina. Til slutt var det nye designendringer på skroget som avsluttet prosessen med å utvikle bein til en kuleformet ROV.



Figur 61. Bein til skrogdesign 5.



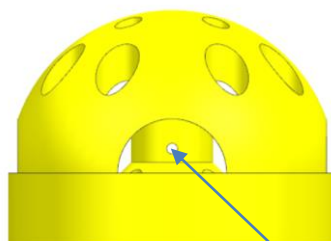
Figur 62. Skrogdesign 5 med bein.

Problemet med den utformingen var at kanalen til den vertikale thrusteren blir påvirket når ROV-en er i flymodus. Det fører til ekstra turbulens og et kontinuerlig moment som virker på servoene. Dette bidrar blant annet med å øke strømforbruket. Beina kan også vinkles slik at de står fram/bakover, men dette gir et større areal kreftene kan virke på og i verste fall medføre at ROV-en får en stupebevegelse. Med utformingen på figur 62 vil ROV-en ha en bakkeklaring på rundt 60-70 mm. Ønsket bakkeklaring fra Kystverket er satt til rundt 100 mm. En bakkeklaring på 100 mm krever at beina går stabilt.

### Endelig designforslag

Etter å ha mottatt robotservoene kunne en ny designfase starte. Alle tidligere 3D-tegninger ble forkastet for å ikke låse seg til gamle løsninger. Robotservo som ble bestilt var av type RS0610MD (mer om dette i avsnittet om "motorer" i kapittel 3.2.3). Robotservopakken inneholder også innfestningsbøyljer i aluminium (se figur 64). Det var tenkt å bruke bøyljene i videre i designfasen. Etter å ha modellert et nytt modulfeste ble det laget tre trumpetledd, som gir beina mer mellomrom. Dette gir større stabilitet til ROV-en.

Tetningsstrømpen (med Ø55 mm) skal tres over trumpetleddet og det nest siste leddet. For å beskytte den nederste tetningsflaten har selve foten fått en beskyttende kant som ligger over tetningsflaten. Foten er avrundet og blir montert med en låsesplint. Den delen av beina som er i kontakt med underlaget har hull for å synke lenger i bunnen og mer areal til å sparke fra og/eller få ekstra fotfeste. Hullene er laget slik at vann, sand eller stein kan renne ut av foten når ROV-en løftes ut av sjøen.



Figur 63. Tuppen på beinet.

Hull til låsesplint.

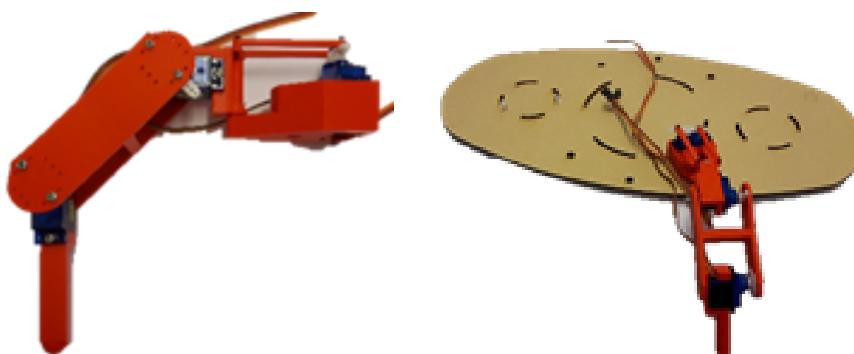
Fordeler med designet er at det begynner å ligne på kommersielle robotkonstruksjoner. Ulempen er at servoene må ta imot støt som kan redusere levetiden eller brudd på tannhulets tenner. Vi har heller ikke funnet en bedre løsning enn å tette med en gummislange (sykkelslange). Trykket på 25 meters havdybde er rundt 2.5 bar, noe som medfører at luften i gummislangene komprimeres. I verste fall gjør dette at gummislangene legger seg tett inntil servoene og skaper uønsket friksjon. Ved store utslag av beina vil det skapes ekstra friksjon som skaper knekk i slangen som følge av høyt trykk. En annen ulempe med å bruke strømpe er at luften gir ekstra oppdrift.



Figur 64. "True Shade"-render av beinet i NX.

### 3D-printede modeller til automatiseringsteknikk

For å demonstrere bevegelsen av beina og hvilken bakkeklaring de ville gi, ble det skjært ut bunnflate til skrogdesign 6 i skala 1:1. Hull til kjøleplater og rør ble antydnet for å øke modellens realisme. Nye forenklete bein illustrerte bakkeklaringen i en skala på rundt 1:1.5. Beina fikk nytt forenklet design for å redusere tiden i 3D-printeren. Servoene blir festet slik de står i originalen, i en vinkel på 30°.



Figur 65. Fysiske modeller fremstilt til automatiseringsteknikk.

## Ulike løsninger til styring av bein

### Søk etter egnet vaier

I søket etter en egnet vaier til våre prototyper har vi brukt og testet diverse produkter.

Følgende produkter ble testet:

- Diverse mangetrådet kabel (kun på våre konseptmodeller)
- Fiskesnøre: to produkter fra «Berkley» (Trilene og FireLine)
- Vaier som er brukt til å starte motorsager, plenklippere og ryddesager
- Line kjøpt hos «Ferd Båt»

Våre krav til vaier:

- Motstandsdyktig mot mekanisk slitasje
- Elastisitet (lite tøyning i vaier under belastning)
- Fleksibelt med tanke på stivhet og mulighet til å knytes
- Maks. ønsket diameter: 1.5 mm (dette på grunn av innfestningspunkter til servoene)

**Fiskesnøre:** FireLine Ø 0.25 mm (17.5 kg)

*Tabell 12. Styrker og svakheter ved Fireline.*

Styrker ved FireLine	Svakheter ved FireLine
<ul style="list-style-type: none"> <li>• svært fleksibel</li> <li>• grei elastisitet</li> <li>• riktig diameter</li> </ul>	<ul style="list-style-type: none"> <li>• tåler lite mekanisk friksjon</li> </ul>

**Fiskesnøre:** Trilene Ø 0.46 mm (20.2 kg)

*Tabell 13. Styrker og svakheter ved triline.*

Styrker ved Trilene	Svakheter ved Trilene
<ul style="list-style-type: none"> <li>• bedre egenskaper mot mekanisk friksjon</li> <li>• riktig diameter</li> </ul>	<ul style="list-style-type: none"> <li>• litt mye elastisk</li> <li>• ikke like fleksibelt som FireLine</li> </ul>



**Startsnøre** (til for eksempel motorsag), Ø3 mm.

Tabell 14. Styrker og svakheter ved startsnøre.

Styrker ved Startsnøre	Svakheter ved Startsnøre
<ul style="list-style-type: none"><li>• svært motstandsdyktig mot mekanisk friksjon</li><li>• god elastisitet</li></ul>	<ul style="list-style-type: none"><li>• stor diameter (3 mm)</li><li>• stor diameter medfører store knuter</li></ul>

**Line** kjøpt hos Ferd Båt, Ø1 mm (ca 17 kg)

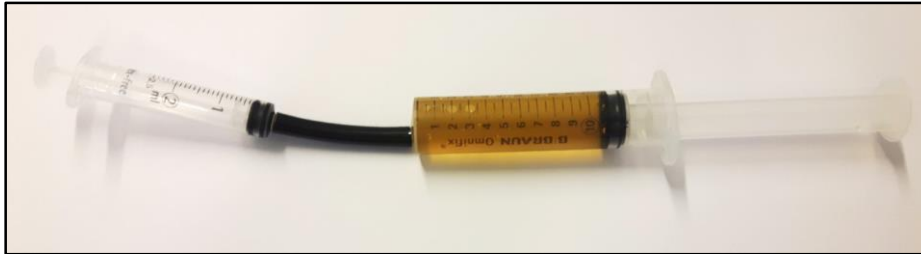
Tabell 15. Styrker og svakheter ved line.

Styrker ved Line	Svakheter ved Line
<ul style="list-style-type: none"><li>• svært motstandsdyktig mot mekanisk friksjon</li><li>• god elastisitet</li><li>• fleksibel</li><li>• diameter er innenfor</li></ul>	

Alle vaiere har positive aspekter men linen kjøpt hos Ferd Båt var definitivt det beste produktet basert på vårt behov. Linen selges i bunt og har en stykkpris på 45,- NOK. Vi har selv foretatt tester for å fastslå når linen ryker. Dette ble målt til å være rundt 17 kg belastning. Det ble fortatt fire tester og tatt gjennomsnitt av resultatene.

## Hydraulikk

Hydraulikk er kraftoverføring ved hjelp av væske og var ønskelig å utforske fordi det er et lukket system. For å unngå forurensningsproblematikk velges matolje som hydraulisk væske.

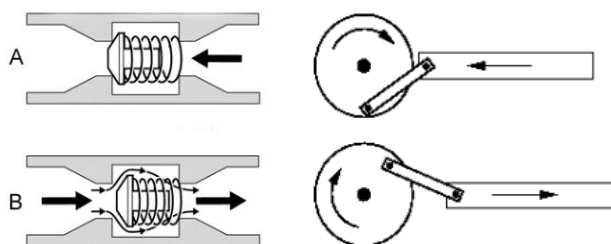


Figur 66. Test av hydraulikk med sprøyter.

Det trengs seks sprøyter for å styre et bein i x-, y- og z-retning og til sammen 36 sprøyter for alle beina. I tillegg trengs en elektrisk komponent (servomotor) til å skyve sprøytene. Samtidig var det et krav at det hydrauliske systemet skulle veie så lite som mulig. Derfor unngikk man retningsventiler, ekstra reservoar og pumper.

Forflytningen til beina krever at den utvendige sprøyten (i beinet) har et mindre væskevolum enn den innvendige. Dette medførte to valg: en liten og lang sprøyte eller økt stempeldiameter. Den sistnevnte løsningen er å foretrekke i og med at ROV-en skal være kompakt.

Den plassbesparende løsningen medførte et behov for en større kraft som skyver sprøyten på grunn av forholdet mellom kraft og areal. En uunngåelig konsekvens er at fartøyets hydraulikksystem kan påvirkes av den minste sprøyten dersom en uforutsett impuls oppstår. Derimot kan dette motvirkes ved å bruke en tilbakeslagsventil mellom sprøytene, eller at den elektriske komponenten har nok ytelse til å motstå impulsen.



Figur 67. Tilbakeslagsventil til venstre. Fra rotasjon til lineær bevegelse.

(Rice University, u.d.)

(Webområde for Reload, 2016)

Det ble undersøkt å bruke servoer som styringsenhet for sprøytene ved å gjøre rotasjon om til lineær forflytning. Med sterke nok servoer kan det spares plass, ettersom at man kan se bort fra tilbakeslagsventilen. En konsekvens av dette er at både vekt og pris økes. Samtidig er det mulig at tannhjulene i servoen ikke tåler slagbelastningene, som reduserer levetiden betraktelig. De fleste servoleverandører oppgir styrke pr cm arm. Det er mulig at beinet ikke får nok forflytningen fordi vandringsen kan være for liten.

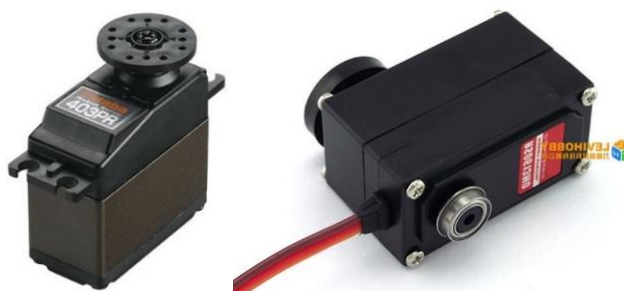
Tabell 16. Styker og svakheter ved hydraulikk i vår ROV.

Styrker ved hydraulikk	Svakheter ved hydraulikk
<ul style="list-style-type: none"> <li>• enkelt prinsipp: kraft / areal</li> <li>• kan bruke matolje som væske (slipper forurensninger og lett tilgjengelig)</li> <li>• sylindere A til B er lett å koble sammen</li> <li>• enkelt å bygge</li> <li>• komponenter som er enkle å erstatte</li> <li>• lukket system</li> <li>• muligheten for å forme beina med fri vanngjennomstrømning</li> </ul>	<ul style="list-style-type: none"> <li>• plasskrevende</li> <li>• uforutsigbare krefter kan oppstå av liten stempeldiameter på innsiden av skroget</li> <li>• vil kreve å forme skroget rundt hydraulikken</li> </ul>

Aspektet ved hydraulikk ble undersøkt videre for å se hvilke komponenter som kunne anvendes i hydraulikksystemet. En tysk leverandør<sup>7</sup> har et stort utvalg av hydrauliske stempler, som brukes i miniatyrmodeller og lignende, men prisen hindret videre vurdering.

### Valg av servo

Det var ønskelig å ha to innfestningspunkt til servoene, som også var sterke nok til å løfte fartøyet. Av den grunn undersøktes det robotservoer fremfor vanlige servoer. Festet til innfestningsbøylene i Levi Hobbys (LH) robotservo låser frihetsgradene til bevegelsen, slik at bøylene kun roterer. Dermed blir ikke beina utsatt for unødvendig vridning. Vekt- og størrelsesforskjellen til servoene fra LH og vanlige servoer var små. Samt har servoene fra LH skruinnfestninger i hvert hjørne som muliggjør et enklere feste i trumpetleddet (se figur 68). Dette gjorde robotservoene til en verdig komponent.



Figur 68. Servo fra LMR til venstre. Servo fra LH til høyre..

(Let's Make Robots, 2014)

(Levi Hobby, u.d.)

<sup>7</sup>Leimbach Modellbau + Elektronik - <http://www.leimbach-modellbau.de/>

## FEM-analyser i NX10.0

### Formler

$$\eta = \frac{\sigma_F}{\sigma_S} \quad (5)$$

Sikkerhetsfaktor – Flytegrense delt på skjærspenning (5)

### Parametere og krav i NX

Tyngdeakselerasjon: 9.81 m/s<sup>2</sup>

Tetthet til sjø: 1030 kg/m<sup>3</sup>

Hydrostatisk trykk på 2.5 bar, eller 25 meters dybde

ABS som valgt materiale

«Element Size» ved «mesh» på lavest mulig størrelse

«Pinned Constraint» for skruene og pakknippelen

«Simple Support»-begrensning for rørovergangen

Punktkraft på beina: 60 N

Punktkraft på beskyttende kant på foten: 100 N

I NX velges delen av konstruksjonen som analyseres med en «Element Iterative Solver». EIS samler og reduserer feilene gjennom iterative beregninger: derfor ønsker vi å bruke denne metoden. For å unngå feilmeldinger og redusere ventetiden ble størrelsen mellom elementnodene i "Mesh"-innstillingene satt opp til 5 mm i "3D Tetrahedral Mesh" med elementtypen "CTETRA(10)". Materialet til komponenten ble angitt som ABS. Deretter bestemmes begrensninger av frihetsgrader og det hydrostatiske trykket på komponenten. På endeledet til beina ble det plassert en punktkraft på kulen og på siden for å simulere kreftene som kan oppstå. Fordi beina er lik tas ikke alle med i analysene. Til slutt angis tyngdeakselerasjon.

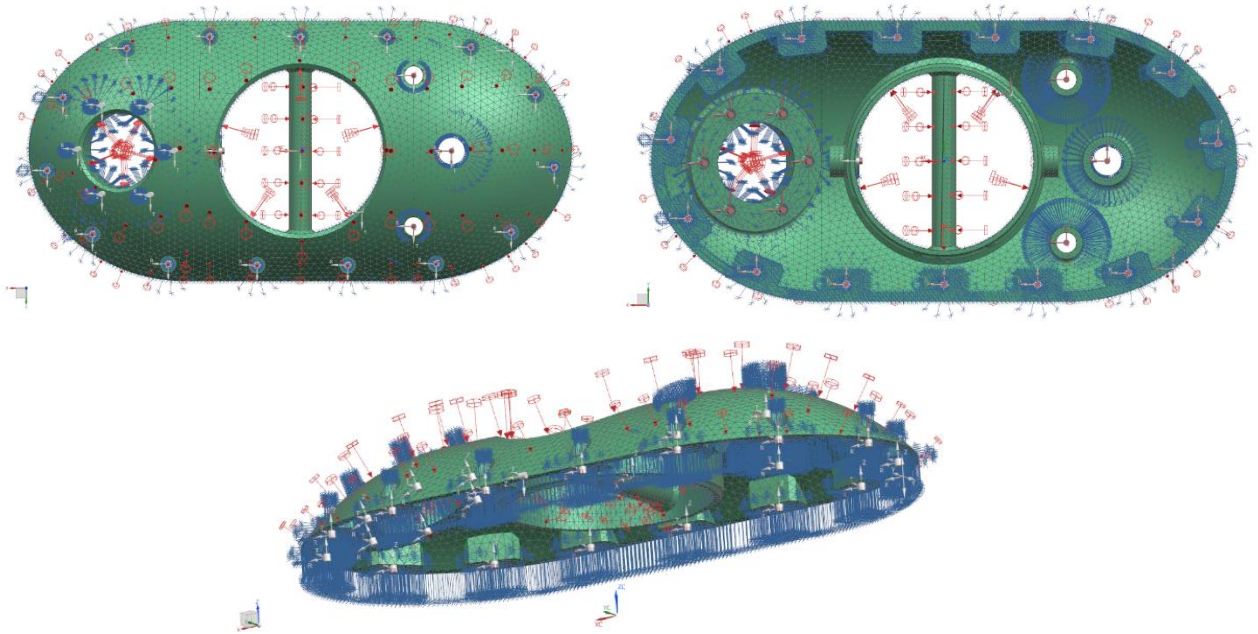
### FEM-simulering

#### Deksel

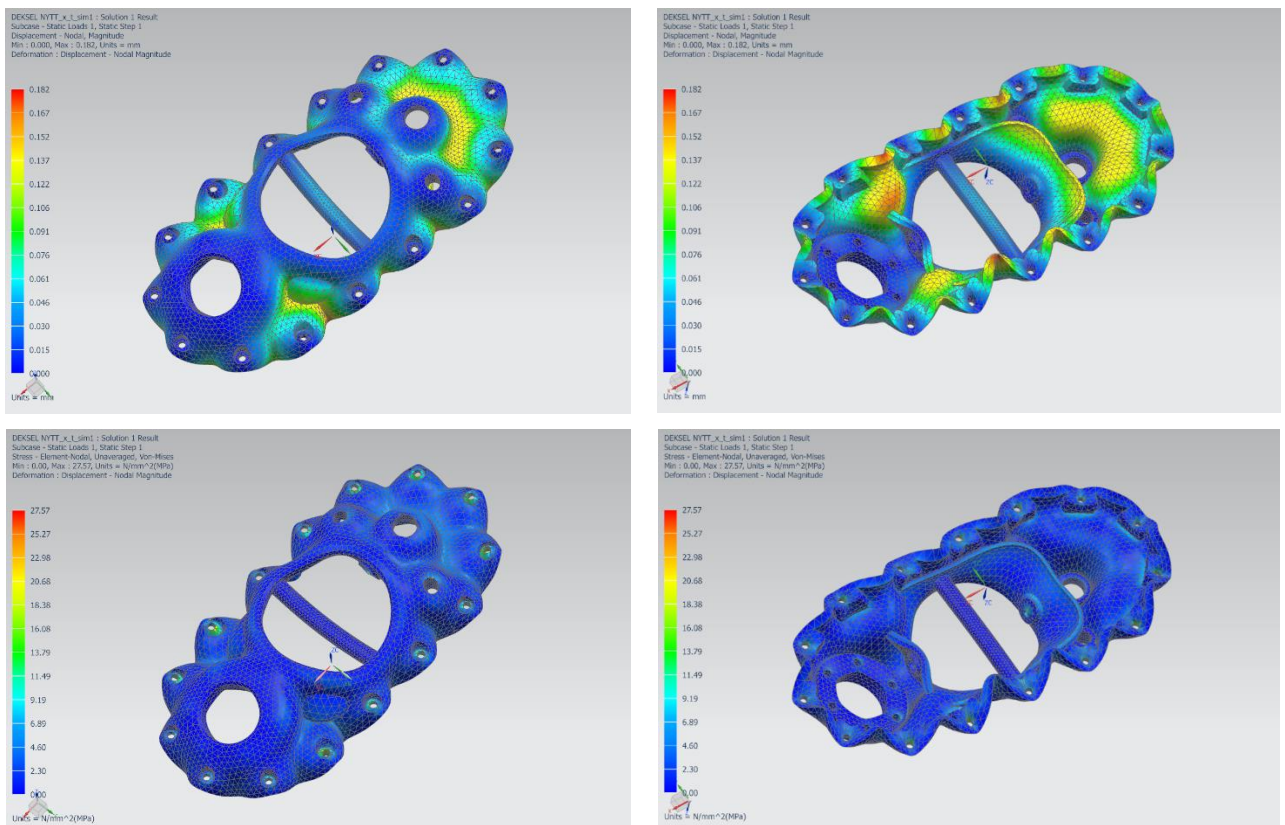
Største deformasjon = 0.2 mm

Størst jevnføringsspenning = 28 N/mm<sup>2</sup>

Sikkerhetsfaktor n = 1.4



Figur 69. NX-Constraints til dekselet.



Figur 70. Deformasjon og jevnføringsspenninger i dekselet.

Flere analysefigurer i vedlegg 4.

**Skrog**

Størst deformasjon = 0.3 mm

Størst jevnføringsspenning = 25 N/mm<sup>2</sup>

Sikkerhetsfaktor n = 1.6

**Bein****Modulfestet**

Størst deformasjon = 0.3 mm

Størst jevnføringsspenning = 18 N/mm<sup>2</sup>

Sikkerhetsfaktor n = 2.2

**Trumpetledd**

Størst deformasjon = 0.1 mm

Størst jevnføringsspenning = 15 N/mm<sup>2</sup>

Sikkerhetsfaktor n = 2.7

**Endeleddet**

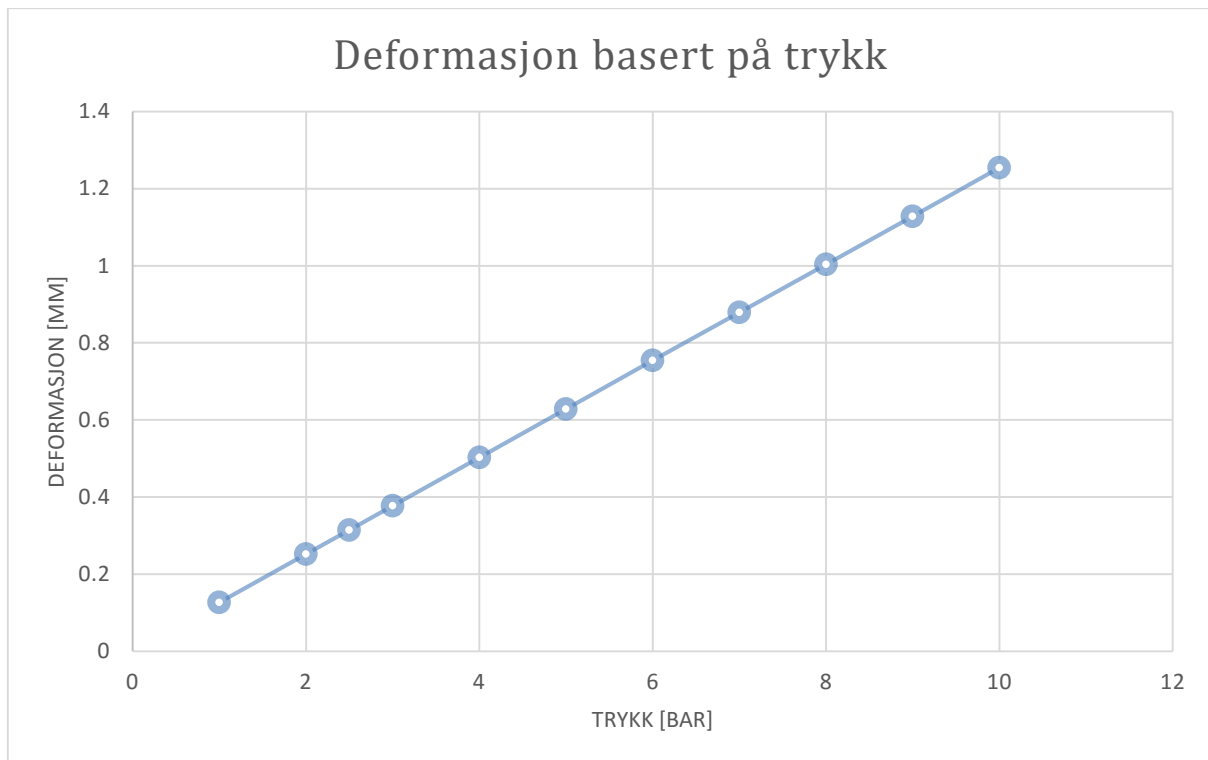
Størst deformasjon = 0.5 mm

Størst jevnføringsspenning = 19 N/mm<sup>2</sup>

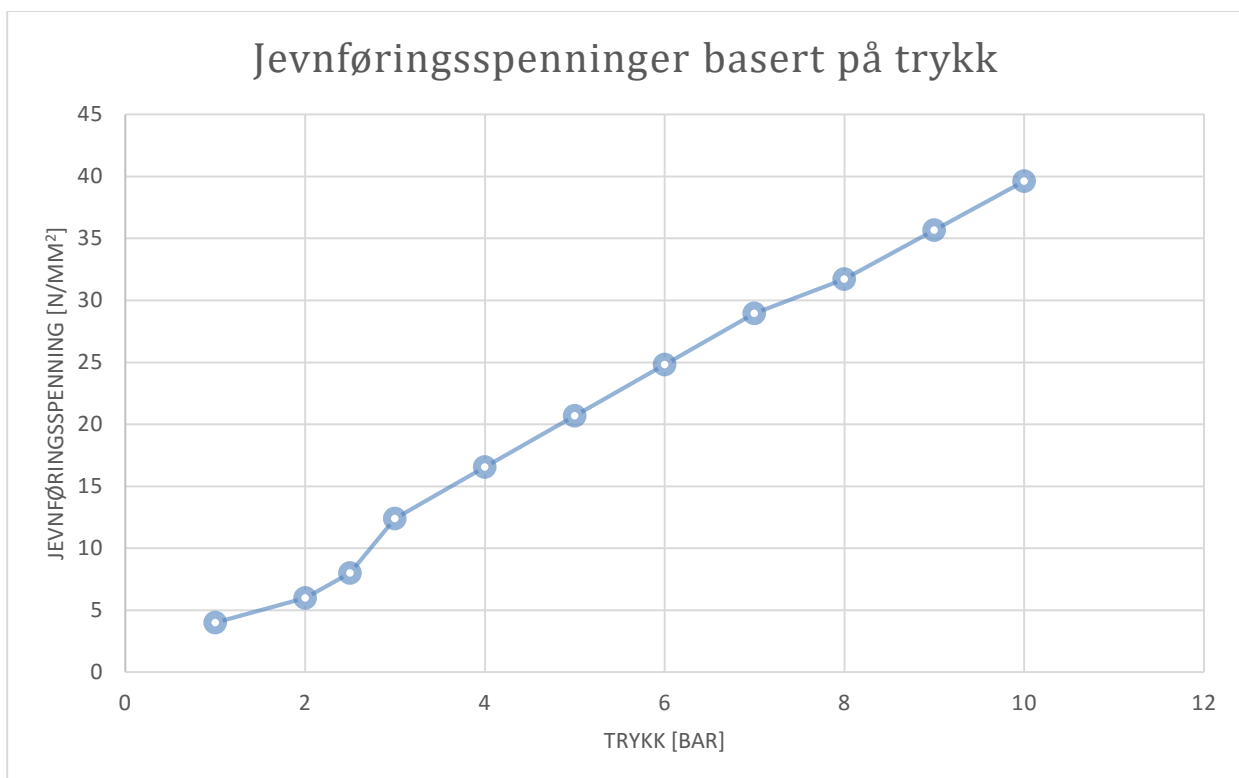
Sikkerhetsfaktor n = 2.1



Figur 72. QR-kode. Album med animasjoner som viser deformasjoner og jevnføringsspenninger i NX.



Figur 71. Deformasjon av skroget.



Figur 72. Jevnføringsspenninger på skroget.

Styrkeberegningsanalyse i NX kan gi utfordringer dersom man utøver analysen på flere deler i samme simulering. Av den grunn analyserte vi hver komponent hver for seg. Elementstørrelse i "Mesh"-egenskapene satt til opptil 5 mm. Forutsetningen for selve simuleringen var at alle komponentene printes med 100% innfyll.

For å vise deformasjonens retning overdriver NX deformasjonen i simuleringen. Skroget er analysert uten det modulære festet og av den grunn får størst deformasjon rundt festet. Dette er neglisjerbart da festet vil fungere som en avstiver.

Analysen resulterte i en deformasjon som er tilnærmet 0 mm ved 25 meters dybde av det hydrostatiske trykket. Jevnføringsspenningen som påføres av trykket medførte en spenning på opptil 30 MPa. Materialets flytegrense ligger mellom 40 MPa og 45 MPa (TestStandard, u.d.) (MatWeb, u.d.).

Sikkerhetsfaktoren tilsvarte 1.4 for den svakeste komponenten. Det røde området som vises i resultatene er i verste fall det største trykket på fartøyet og oppstår i de harde hjørnene til skruhulene. Vi ser på disse som neglisjerbare da skruer vil stive av materialet. Hovedsakelig kan ABS tåle en god del høyere trykk enn hvis vi antar den verste situasjonen, som linjediagrammene viser.

Ifølge analysen kan fartøyet dykke 25 meter uten noen betydelige deformasjoner eller påkjenninger. Anbefalt dybde er ned til 30 meter.



## Oppdriftsberegninger

### Prinsipp

Volumet av hver komponent reduserer luftrommet i skroget. For å finne innvendig luftvolum ble volumet av komponentene trukket fra det totale volumet til skroget.

### Volumformler

$$V = l * b * h \quad (6)$$

$$V = \pi * r^2 * h \quad (7)$$

$$V_{luft} = V_{total} - V_{skrog} - V_{elektriske\ komponenter} - V_{diverse} \quad (8)$$

$$V [m^3] * Miljøtetthet \left[ \frac{kg}{m^3} \right] = Masse [kg] \quad (9)$$

$$Oppdrift = Masse_{oppdrift} - Masse_{total} \quad (10)$$

$$Oppdrift\ med\ thruster = Oppdrift - Løftekraft_{T100} \quad (11)$$

$$1\ L = 1\ kg \quad (12)$$

Antas at hver elektrisk komponent er formet som en boks – det gir et større volum enn i virkeligheten. (6)

Volumet av en sylinder. (7)

Volumet av luft innvendig. (8)

Masse som tilsvarer oppdrift i gitt miljø. (9)

Oppdrift i kg. (10)

Oppdrift med vertikal thruster i betraktning. (11)

Arkimedes lov. (12)

### Parametere

Tetthet til sjøvann: 1025 kg/m<sup>3</sup>

T100-thrusterens løftekraft: 2.56 kg

Totalvekt til ROV: 6.9 kg

Vekt til bein: 1.2 kg

Alle elektriske komponenter ble sett på som bokser for å forenkle volumberegningene. Denne antakelsen medførte noe unøyaktig resultat, ettersom forenklingen gjorde at hver komponent ble sett på som litt større enn reelt. Unøyaktigheten ble imidlertid ansett som neglisjerbar. I NX10.0 benyttet vi funksjonen «Measure Bodies» til å måle skrogets volum (og andre komponenter). Skrogets totalvolum ble målt som en massiv gjenstand av skroget. Volumet av skroget ble ansett som volumet av ABS. ABS-volumet og komponentene ble trukket fra skrogets totalvolum for å finne innvendig luftvolum. Videre multipliseres skrogvolumet med en prosentkonstant for å bestemme hvor mye luft skroget printes med. Deretter brukte vi den vertikale thrusterens løftekraft til å beregne forflytningen ved maks ytelse.

## Oppdriftsverdier

«Oppdrift med komponenter» tilsvarer oppdriften fartøyet har med alle elektriske komponenter er tatt i betraktning. «Oppdrift til skroget» gjelder uten noen komponenter i beregningene.

Av NX10.0 «Measure Bodies»,

Totalvolum av skroget (inkludert innsiden) = 8748611 mm<sup>3</sup>

Volum av selve ABS-materialet til skroget = 1914792 mm<sup>3</sup>

12 servoer og 70% innfyll

Oppdrift til skroget  $\approx$  51 N

Oppdrift med komponenter  $\approx$  -12 N

Oppdrift med thrusterens løftekraft  $\approx$  12 N

18 servoer og 60% innfyll

Oppdrift til skroget  $\approx$  51 N

Oppdrift med komponenter  $\approx$  -12 N

Oppdrift med thrusterens løftekraft  $\approx$  12 N

Tabell 17. Oppdriftstabell.

	Oppdrift til skroget [N]	Oppdrift med komponenter [N]	Oppdrift med thrusterens løftekraft [N]
Ingen servoer og 100% innfyll	50.1	3.1	26.7
Tolv servoer og 70% innfyll	50.7	-11.2	12.4
18 servoer og 60% innfyll	51.1	-11.9	11.7

Fartøyet må ha negativ oppdrift for å kunne gå på havbunnen. Når konstruksjonen er utstyrt med bein vil den synke med 1 kg. Uten bein vil fartøyet ha nøytral oppdrift.

## Mechatronics Concept Designer

### Prinsipp

NX10.0 Mechatronics Concept Designer gjør at man kan bestemme posisjonen til komponentene og slik visualisere bevegelser.

### Anvendte funksjoner i Mechatronics Concept Design

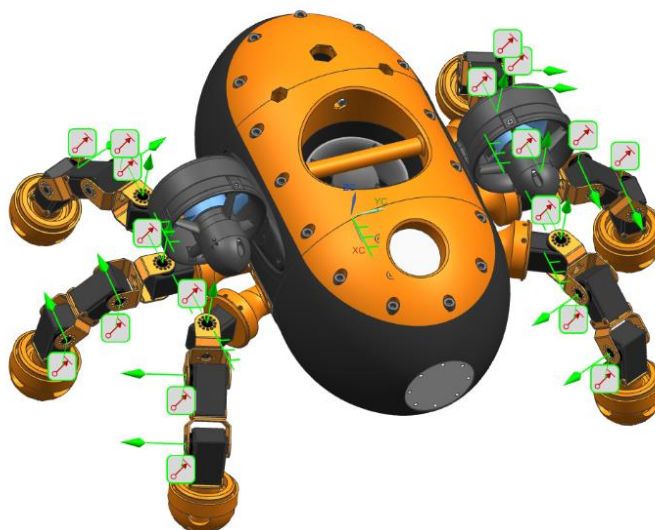
- «Rigid Body» angir hvilke komponenter som er relevante.
- «Fixed Joint» begrenser alle frihetsgrader.
- «Hinge Joint» begrenser alle translasjonsretninger mens rotasjonene er fri. Denne brukes til å gi rotasjon fra servoene til leddene. Rotasjonen vil skje på «Attachment»-verdien i forhold til «Base»-verdien.
- «Collision Body» brukes på en overflate og lager et hinder for andre komponenter med samme funksjon når de kolliderer.
- «Actuators», som «Position Control», bestemmer hastighet og posisjon til «Hinge Joint».
- I «Sequence Editor» kan man angi når en bevegelse skal skje.

«Rigid Body» ble brukt på skroget, modulfestet og det første leddet på beina. Denne funksjonen ble i tillegg brukt på den innerste servoen, på selve festet til den innerste servoen, på midterste leddet og til slutt på det siste leddet. Dette ble gjort for alle beina.

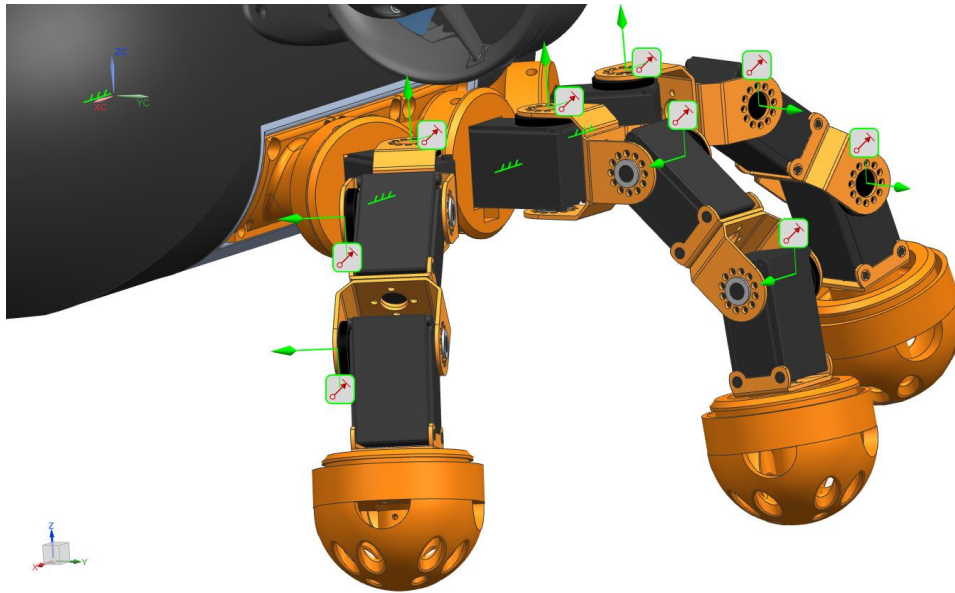
«Fixed Joint» ble valgt på alle innerste servoene for å hindre at beinene opplever et fritt fall i simuleringen. Deretter ble det brukt «Hinge Joint» der rotasjonen skjer normalt på festet.

Servoen er satt som «Base» med festet som «Attachment». Samme funksjon ble brukt på alle leddene i steg fremover. På alle «Hinge Joints» ble det anvendt en «Position Control» «Actuator». Til slutt angis verdiene til posisjon og fart i «Sequence Editor» som en operasjon basert på tid.

### Simuleringen



Figur 73. «Joints» og «Actuators» på sammenstillingen.



Figur 74. Nærbilde av bein med «joints» og «actuators».



Figur 75. QR-kode med NX-simulering.

Mechatronics Concept Designer (MCD) ble anbefalt fremfor Motion Simulation (MS), fordi det er mulig å styre flere komponenter samtidig i MCD. Vi prøvde begge deler og fant ut at MCD var mer egnet fordi man fikk større kontroll over hele simuleringen ved å bestemme posisjon til enhver tid. Av resultatet så vi at bevegelsen til konstruksjonen var lite optimalisert. På grunn av tidsmangel, og fordi vi ble rådet å vurdere flere konsepter, fikk vi ikke mulighet til å optimalisere bevegelsen.

MCD gjorde det mulig å lage en visuell fremstilling av gangarten til fartøyet. Simuleringen var et forsøk på å gjøre teori knyttet til bein og gangart litt mer praktisk, og for å få et inntrykk av om lengden og vinklene til leddene i beina kunne fungere. Dette var mulig å gjennomføre da en av POD-studentene fikk tilgang til et elektronisk kurs egentlig forbeholdt forelesere for Mastergradsstudenter.

## Keyshot

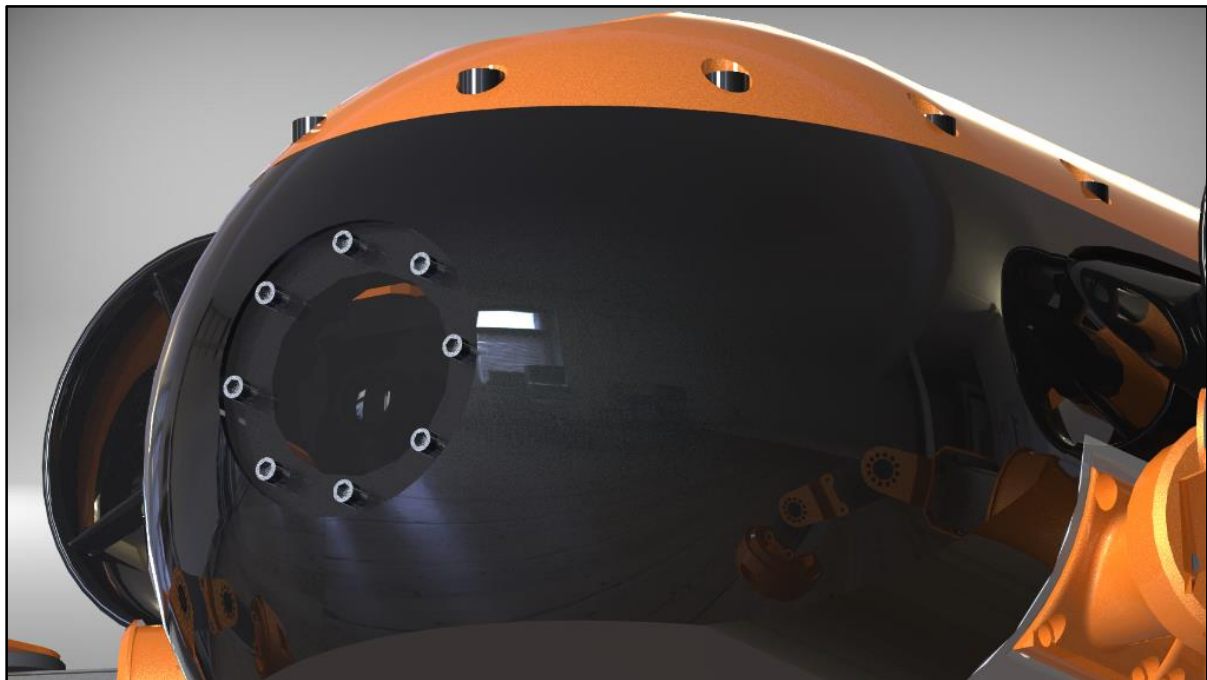
Keyshot er et verktøy som gjengir 3D-modellen med materialer og avansert lyssetting for å representere hvordan modellen vil se ut i virkeligheten.

Sammenstillingsmodellen ble importert inn i Keyshot fra IronCAD. Alle komponentene i sammenstillingen ble samlet i ulike grupper og navngitt (under "Scene") for å skape oversikt. Bakgrunnen ble satt som standard med Dosch-Studio2\_2k som "Environment". Komponenter som ikke vises fra utsiden ble skjult for å spare tid ved 3D-gjengivningen. Deretter fikk komponentene tildelt materialer fra Keyshots bibliotek. Under materialinnstillingene ble "Roughness" økt til 0.05 for å redusere lysets gjenspeiling i materialet.

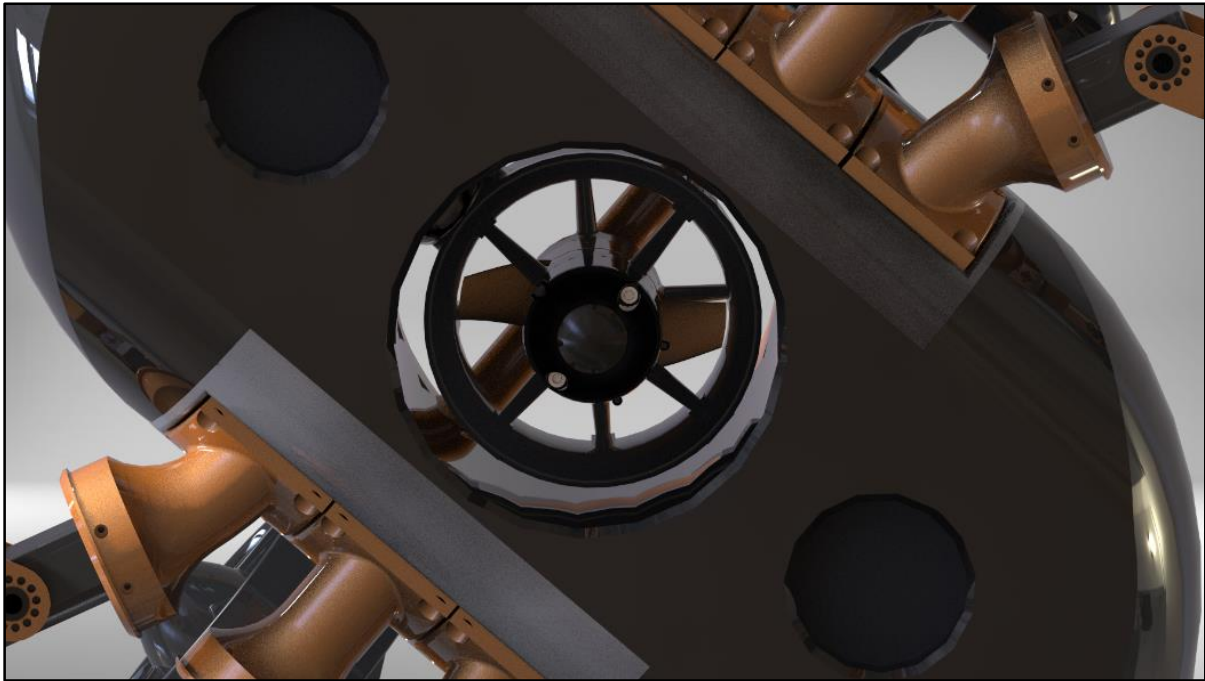
Ulike vinkler ble lagret under "Camera" som egne scener. Under "Lens Settings" ble "Orthographic" valgt for å få et realistisk perspektiv.

Under "Image" ble kvaliteten økt ved å velge krysse av "Self Shadows", "Global Illumination", "Ground Illumination" og "Caustics". For å gjøre kameravinduet gjennomskiktig (slik at "lys" slipper gjennom) ble "Ray Bounces" satt som 12, "Indirect Bounces" som 10 og "Shadow Quality" på 1. Til slutt endres belysningen fra "Dosh-studio" ved å rotere rommet i forhold til modellen under "Environment".

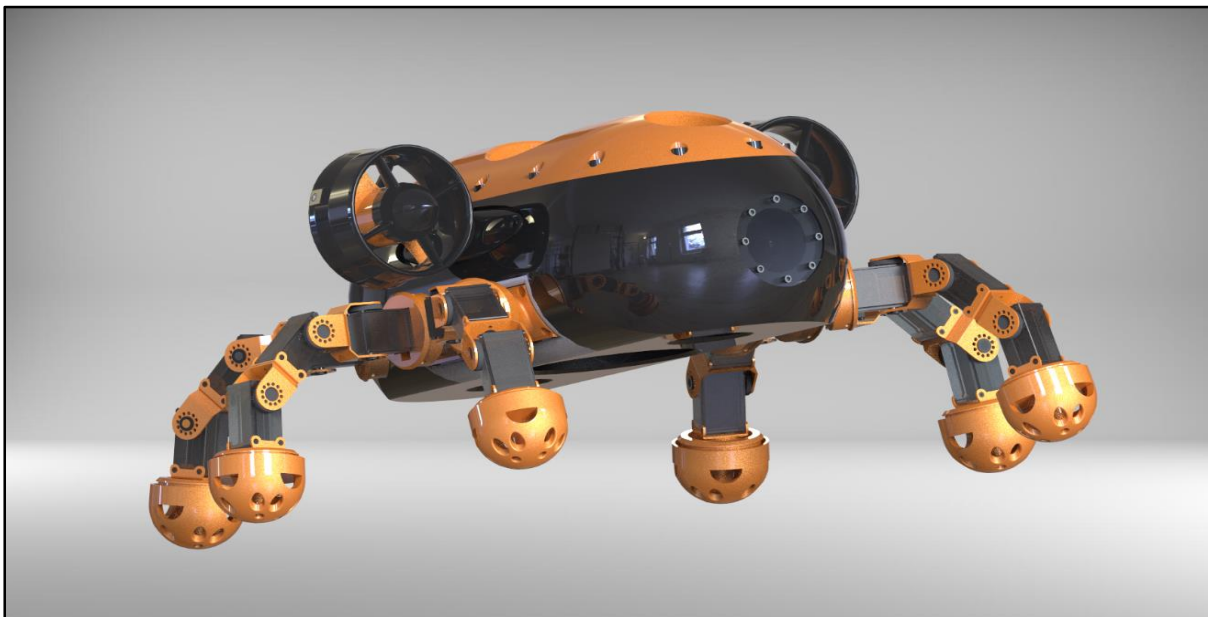
## 3D-rendering av sammenstillingen



Figur 40. Frontdel på konstruksjonen.

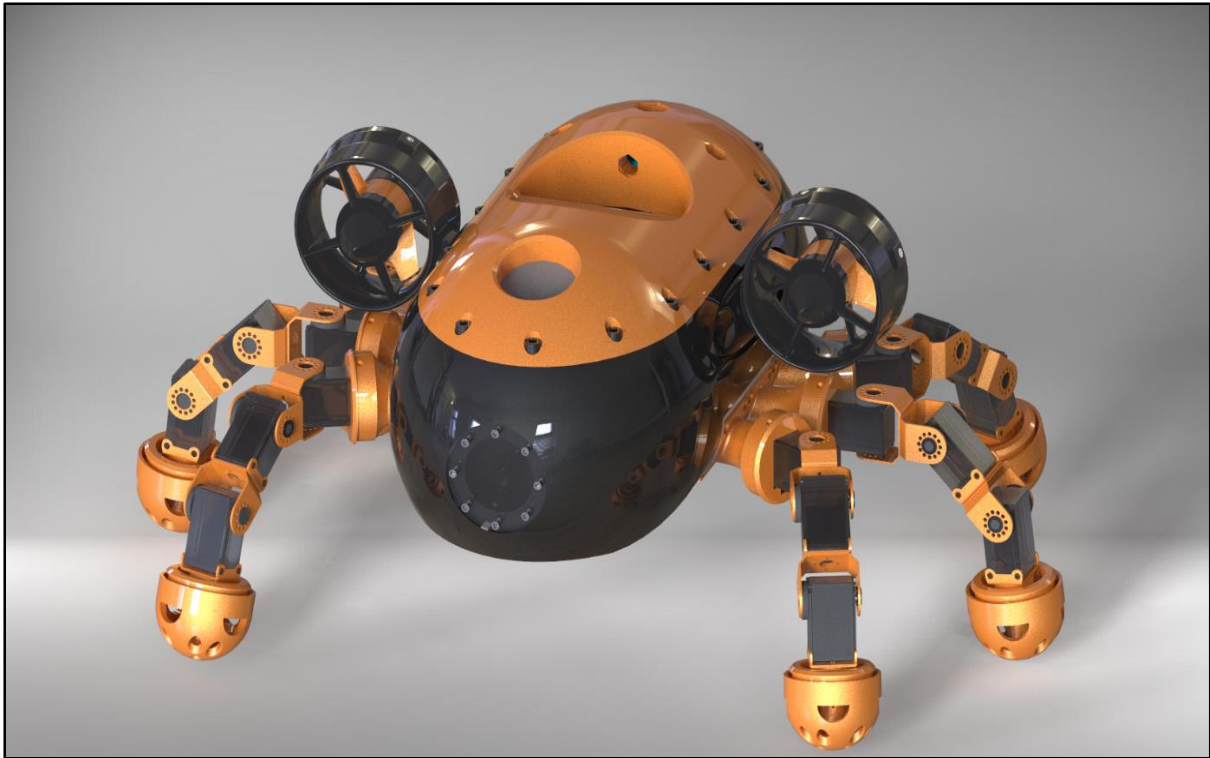


Figur 41. Undersiden av konstruksjonen.



Figur 42. Sammenstillingsbilde av konstruksjonen.





Figur 43. Sammenstillingsbilde av konstruksjonen.

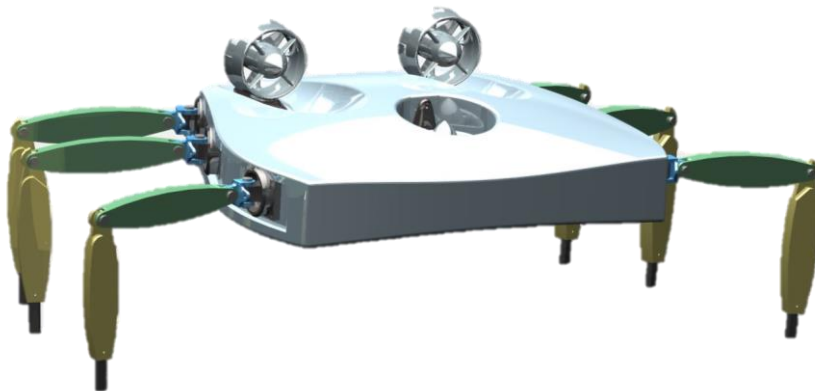


Figur 44. Sammenstillingsbilde av konstruksjonen fra siden.

## Sammenstillinger



Figur 76. Sammenstilling av ROV 1.



Figur 77. Sammenstilling av ROV 2.



Figur 78. Sammenstilling av ROV 3.



### 3.1.5 Delkonklusjon

#### *Om prosessen*

Da vi begynte var kunnskapen om ubemannede undervannsfarkoster svært begrenset og vi prioriterte å bruke tid på å orientere oss om fagfeltet, sette oss inn i dets historikk, og se på eksisterende løsninger på problemstillinger som liknet vår egen. Å stå på skuldrene til alle som har jobbet med å utvikle en ROV før oss, hjalp oss å visualisere et sluttprodukt tidlig i prosessen.

Å designe en observasjons-ROV har vært preget av jevn progresjon på noen områder og «to steg fram, ett tilbake» på andre. Evnen til å komme opp med alternative løsninger etter hvert som problemene oppstår er avgjørende, fordi motvilje til å gjøre endringer ikke hører hjemme hos en produktutvikler. Det er nettopp vilje til å avvike fra tidlige planer og utkast, samt å omfavne noe usikkerhet som er viktig for å lykkes med produktutvikling og prosjektarbeid generelt. Ofte er det nødvendig å ikke implementere en løsning man personlig er svært tilfreds med av hensyn til helheten.

#### Delkonklusjon 1:

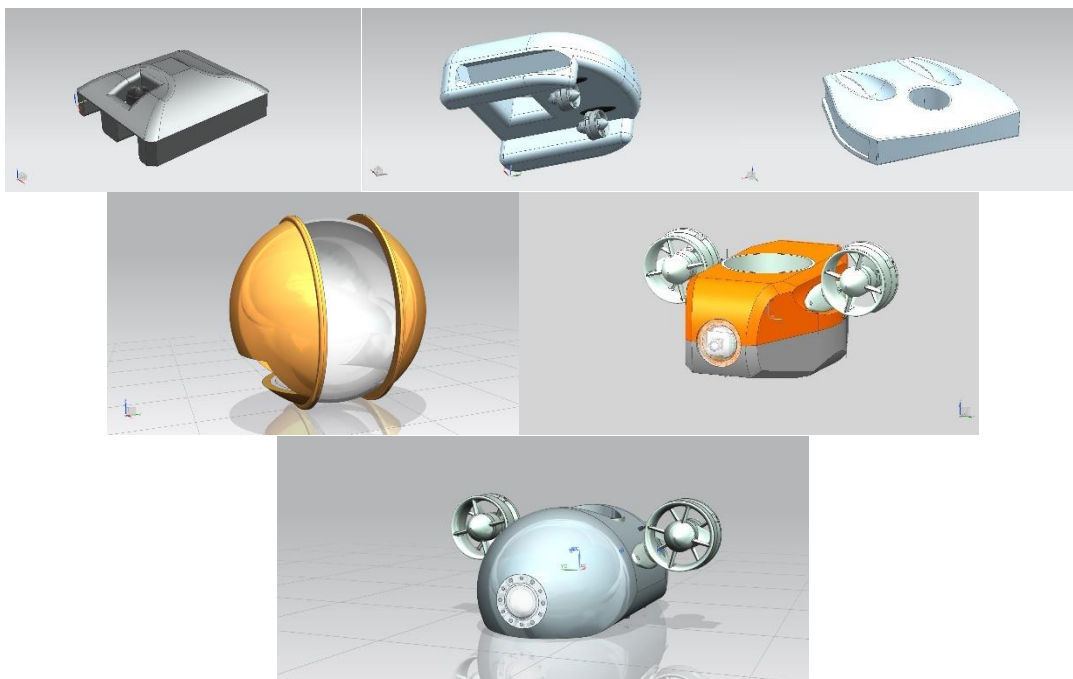
Å være villig til å henlegge design for å kontinuerlig bevege seg mot en bedre løsning er beskrivende for prosjektets arbeidsmåte.

#### *Om produktutvikling*

Tidlig i produktutviklingsfasen var det viktig å være åpen og ukritisk slik at idemyldringen ble rikest mulig. Først noe senere ble det viktigere å se kritisk på de ulike forslagene. Idemyldring preges av divergent tenkning, som er kjennetegnet ved at den er fleksibel, åpen og spontan. Senere ble den konvergente tenkningen viktig; det vil si målbevisst og kritisk vurdering av resultatene fra idemyldringen.

#### Delkonklusjon 2:

*Produktutvikling er preget av både divergent (assosiasjonsrik) og konvergent (kritisk) tenkning. Det var viktig å bruke god tid på å avdekke kundens behov og definere et entydig produkt ut ifra disse, før man skisserer et endelig design.*



Figur 79. Skrogets evolusjon.

## Anbefalinger til videre arbeid

### *Skrog*

På grunn av tidsbegrensingen endte vi ikke opp med et design vi anser som hundre prosent ferdigstilt. Dersom tidsrammen hadde tillatt det, ville vi utviklet en forbedret versjon av skroget og laget det mer avlangt i akterenden. Slik ville vi kunne redusert motstandskraft og laget en mer energieffektiv ROV. Et forslag til videre arbeid inkluderer derfor å se på utformingen av skroget for å gjøre det enda mer hydrodynamisk. Det bør likevel nevnes at skroget har en svært lav dragkoeffisient, slik det fremkommer av motstandsbergingene.

### *Bein*

Det var vanskelig å finne vanntette servoer. Løsningen medførte bruk av en strømpe for å gjøre beina vanntette, noe som gir uønsket oppdrift. En strømpe er dessuten utsatt for slitasje og punktering. Skjell, rur og skarpe steiner er vanlig i fjæra og i grunn sjø. Dette betyr at man må se mer på alternative tetningsløsninger dersom prosjektet tas videre. Dersom budsjettet tillater det anbefaler vi bruk av vanntette motorer.

For at en ROV med bein skal ha god fremkommelighet i fjæra og på ulendt sjøbunn, må den ha god bakkeklaring. Lange bein er derimot en ulempe i flymodus. Dersom kundens mest prekære behov er en ROV som kan ta gode bilder, anbefaler vi å undersøke muligheten for at beina muligens kan fjernes. Vår anbefaling er å prioritere gode egenskaper i flymodus og heller designe et slags stativ som gjør at fartøyet kan stå stødig på underlaget under filming. Dette fordi at beina, slik de er utformet nå, utgjør et svakt punkt for konstruksjonen. De kan fungere som en uønsket arm til kreftene i sjøen og vil mest sannsynlig også påvirke manøvrerbarheten i flymodus. For at en ROV med bein skal ha god fremkommelighet og takle hinder må beina være ganske lange for å gi god bakkeklaring. Lange bein er derimot en ulempe i flymodus og de kan også sette seg fast i tang og tare. Tang- og tareproblematikken ble grundig demonstrert i videomateriale vi fikk utdelt fra kontaktpersonen vår i Kystverket. På bakgrunn av videomaterialet og samtaler med kunden er det derfor uvisst hvor mye nytte beina kan tilføre Kystverket i forbindelse med inspeksjon av sjømerker. Det mest hensiktsmessige er kanskje å designe et stativ som gjør at ROV-en kan stå stødig på underlaget når den filmer, og at det er kun thrustere som brukes til forflytning.

På det første veiledermøtet i januar 2016, ble det anbefalt å se på kommersielle ROV-er og studere deres design og tekniske løsninger. OpenROV og NTNUs BluEye ROV ble fremhevet som spesielt gode inspirasjonskilder prosjektgruppa kunne lære av. Oppdragsgiver Per Erik Ose informerte om at disse dessuten hadde løst strømforsyningen på en god måte. Med tanke på at mange ulike aktører har valgt å bruke batterier til strømforsyning så anbefaler vi at neste gruppe ser nærmere på dette. Siden dette er en mye utprøvd og mye brukt løsning for ROV-er i samme klasse som vår, konkluderer vi med at denne løsningen har mye for seg og at det ikke er noe å hente ved å prøve å finne opp hjulet på nytt. Vi anbefaler bruk av LiFePO<sub>4</sub>-batteri.

Dersom bein er et krav fra kunden anbefaler vi at det bygges tidlige prototyper for å teste det optimale forholdet mellom størrelsen på skroget og lengden på beina. Trykksensor i alle bein bør inkluderes slik at konstruksjonen får vite om den har bakkekontakt.

Dersom prosjektet tas videre og en form for løsning eller erstatning til beina fortsatt er et krav, anbefaler vi å undersøke muligheten for å bruke hjul. Hjul kan bidra til at ROV står stødig, samt bidra til å finjustere posisjonen til ROV. Basert på videomateriale fra Kystverket så vi at det er mye svaberg i områdene der sjømerker er satt ned, så hjul kan fungere godt.

#### Delkonklusjon 4:

*Observasjons-ROV-en som er utviklet i dette hovedprosjektet har styrker og svakheter knyttet til designet. Selve skroget er utformet slik at det har god hydrodynamikk, men beina bør jobbes videre med. Ettersom batterier i skroget på ROV-en er en utbredt løsning blant moderne kommersielle observasjons-ROV-er, anbefaler vi at denne løsningen utredes grundig. Den er mye utprøvd og dokumentasjon på at den fungerer står blant annet OpenROV og NTNUs BluEye for.*

#### Øvrige anbefalinger

Vi anbefaler å bruke laser for å gi dybdesyn slik at avstander til ulike objekter kan bedømmes. OpenROV har en slik løsning og tilbakemeldingen fra brukerne er at den fungerer godt.

Vi anbefaler å undersøke om det er mulig å lage sin egen thruster. Ved å regne på motstand kan man finne det optimale propellbladsarealet og deretter motorytelse. Det er også mulig å gå til innkjøp av en kommersiell thruster og bruke deler av den, men modellere og 3D-printe et nytt propellhus og innfestningsbrakett. T100-thrusteren har et svært tungt propellhus sammenlignet med en tilsvarende del som 3D-printes. Det er også mulig å lage et skrogdesign som har integrert innfestningsbrakett og propellhus. Dersom vi hadde hatt mer tid ville vi laget et slik design.

Dersom prosjektet skal tas videre, anbefales det å bruke PETG i den endelige prototypen fremfor ABS. PETG har høyere flytegrense og er tilgjengelig fra Kystverket. ABS er vanskelig å 3D-printe fordi materialet krymper mye underveis i printen. Dette er spesielt problematisk for litt større deler.

Dersom vi skulle levert et helt ferdig produkt klar til bruk, ville vi støpt skroget og kun 3D-printet innvendige fester til komponenter. Da ville utformingen av skroget blitt noe annerledes, grunnet hensyn til slippvinkler til støpeformen og hvor modellen skal deles for å få den ut av formen. Støping ville vært en langt bedre egnet produksjonsmetode for masseproduksjon. Det er mulig å støpe i ABS, men det foretrekkes å bruke polystyren siden det er et vanlig støpematerial. PVC er ikke å anbefale ettersom at fartøyet er utstyrt med spenningsregulatorer som opererer under høye temperaturer. Dersom PVC brenner seg, avgir det giftige gasser som også forurenses.

#### Feilkilder

Noe unøyaktighet knyttes til beregningene som er gjort på oppdrift og motstand. Vi er likevel trygge på at forenklingene brukt i beregningene ikke er store nok til å utgjøre alvorlige feilkilder. Forenklingene er foretatt på en slik måte at unøyaktighet medfører høyere fremfor lavere sikkerhetsfaktor.

Motstandsberegningene kan ha avvik på grunn av forenklingene det ble gjort med beina og hvilken dragkoeffisient som ble brukt for alle delene på fartøyet. Dragkoeffisienten er en usikker del i beregningene, siden ulike delene må prøves for å finne en eksakt verdi. Samtidig kan de ulike parameterne som ble brukt variere.

Oppdriftsberegningene vil ha avvik fordi luftvolumet til strømpene ikke ble tatt med i beregningene. Avviket påvirkes også av forenklingene gjort ved volumberegningene til komponentene. Luftvolumet i fartøyet som komprimeres under trykk ble ikke tatt hensyn til i samtlige beregninger.

Styrkeberegningens analyse utført i NX kan ha avvik dersom komponentene begrenses feil eller dersom det hydrostatiske trykket ble plassert på feil overflate. I resultatet angir NX toppverdier hos jevnføringsspenningene. Disse toppene oppstår i utskjæringene til skruene, som medfører avvik i analysen. Deformasjonen i de samme områdene vil ikke nødvendigvis oppstå som vist i analysen på grunn av innskruingsmutterne som monteres. Mutterne vil fungere som avstivere til hullene. Det tas forbehold om at de teoretiske randbetingelsene styrkeberegningens funksjonen til Siemens NX 10.0 bruker samsvarer med virkeligheten.

3D-printede komponenter er bygget opp lagvis med materiale. For å undersøke materialstyrken hadde vi tenkt å benytte oss av tensometeret på verkstedet til NTNU i Ålesund. I og med at man sveiver for hånd egner tensometeret seg dårlig til å måle materialets elastisitet. Det finnes ingen sensor som registrerer forlengelse som funksjon av sveivens hastighet. Vi tok alle nødvendige mål av munnstykkene og lagde 3D-modeller som skulle brukes til prøvestaver. Planen var å printe fem modeller og bruke gjennomsnittet av resultatene til å bruke videre i beregninger på konstruksjonen. Dessverre hadde kvikksølvøylen en lekkasje som gjorde apparatet ubrukelig. På grunn av dette googlet vi og fant ut at Makerbot har foretatt sine egne beregninger (produktspesifikasjon). Her ble flytegrensen oppgitt til å være  $40 \text{ N/mm}^2$  (SETT INN KILDEN HER). Denne verdien ble brukt i styrkeberegningene. Det tas forbehold om at styrkeegenskapene til 3D-filamentet som er brukt i komponenter samsvarer med det produsenten oppgir. Materialeegenskapene vil påvirkes av printerinnstillinger og miljøbetingelser som lufttemperatur, luftfuktighet, samt 3D-filamentets alder og oppbevaringsmåte.

### 3.1.6 Kildevurdering

Rapporten bruker informasjon hentet fra et bredt utvalg kilder. Det er ikke til å unngå at disse er av varierende kvalitet. På den ene siden av spekteret finner vi fagfelleverderte forskningsartikler publisert i kjente og mindre kjente tidsskrifter. På den andre siden av spekteret finner vi meningsutvekslinger på diskusjonsfora. Et sted mellom disse ytterpunktene finner vi offisielle sider med gov –domene.

Vi har etterstrebet å bruke flest kilder fra Google Scholar. Felles for kildematerialet funnet utenfor Google Scholar er at hver enkelt kilde ble vurdert ut ifra et helhetsinntrykk. For å kompensere for manglende kvalitetssikring brukte vi flere kilder når en påstand uten fagfellevurdering ble undersøkt. Dette gjaldt for eksempel for erfaringsutvekslinger om bruk av ulike typer filament til 3D-printing. Dersom den samme påstanden kunne finnes igjen fremsatt av flere aktører på ulike diskusjonsfora og hjemmesider, ble den vurdert som trygg nok til å bruke.

## 3.2 Elektronikk, styring og bildebehandling

### 3.2.1 Teoretisk bakgrunn

#### Zenerdiode

Zenerdioder er dioder som er laget for å operere ved gjennomslagsspenningen i sperreretning, kalt zenerspenningen. I lederretning fungerer zenerdioden som en vanlig diode. Zenerdiodens ledeevne i sperreretning er tilnærmet lik null så lenge den påtrykte spenning ikke overstiger zenerspenningen. Når den påtrykte spenningen overstiger zenerspenningen øker diodens ledeevne eksponentielt. Denne egenskapen gjør at zenerdioder kan brukes til spenningsregulering. (Hambley, Electrical Engineering: Principles and Applications, 2014)

#### Java

Java er et objektorientert programmeringsspråk, utviklet av James Gosling og Sun Microsystems. Det er et høy-nivå språk som er plattformuavhengig. Java kompiles ikke til maskinkode slik som mange andre programmeringsspråk. I stedet kompiles det til bytekode som kjøres av «Java Virtual Machine». JVM kjører igjen på operativsystemet. Dette gjør at de som skriver Java-kode ikke trenger å tenke på maskinvarestøtte. (Oracle, About the Java Technology, u.d.) (Oracle, The History of Java Technology, u.d.)

#### Multitråding

Multitråding gjør det mulig å kjøre flere prosesser samtidig, dersom maskinen som applikasjonen kjører på har en fler-kjernet prosessor. Fordelen med en flertrådet applikasjon er at en prosess kan arbeide uten avbrudd fra tidkrevende operasjoner som kan kjøres i en egen tråd samtidig. Dette sparer tid for brukeren av programvaren som vil oppleve det som en "real-time" applikasjon.

#### Mutex

Mutex blir tatt i bruk for å unngå at flere objekter jobber med delte variabler i et multitrådet miljø. Dersom det ikke brukes Mutex er det stor sannsynlighet for at data som leses av den delte variabelen er korrumpert og ubrukelig. Bruk av Mutex kan også forsinke hele systemet ettersom et objekt kan gjøre krav på variabler som andre objekter trenger for å jobbe videre.

#### C++

Skaperen av C++ er Bjarne Stroustrup. C++ er et programmeringsspråk som har historie tilbake til 1979. Bjarne startet utviklingen av C++ etter han jobbet med Simula 67, som betraktes som det første språket som støtter objektorientert programmering. I starten ble C++ kalt "C with classes", målet med det nye språket var å gjøre C-språket objektorientert. C er et low-level og hurtig språk, derfor tok Bjarne Stroustrup utgangspunkt i C da han startet utviklingen av "C with classes". Dette nye språket inneholdt klasser, grunnleggende arv og funksjonsparameter.

Den første «compiler»-en til "C with classes" heter Cfront og er utviklet med utgangspunkt i C compileren CPre. Denne compileren var designet til å oversette "C with classes" til C kode. Cfront var skrevet hovedsakelig i "C with classes" som gjorde at det er en «compiler» som kan compilere seg selv. Dette påvirket mange fremtidige «compiler»-e og utviklingen av Unix.

C++ er et språk i utvikling og blir modernisert jevnlig med den nyeste versjonen i 2014.

## Qt

Qt er et rammeverk for applikasjonsutvikling som fungerer på tvers av mobile-, innebygde- og skrivebordsplattformer. Støttede plattformer inkluderer Linux, OS X, Windows, VxWorks, QNX, Android, iOS, BlackBerry, Sailfish OS med flere.

Qt rammeverket er skrevet i C++. En «pre-processor» kalt MOC (Meta-Object Compiler) brukes til å utvide C++ språket med særegen Qt funksjonalitet. Før kompilering av koden går MOC-en gjennom kildekoden og leser delene som er skrevet i Qt-utvidet C++, og oversetter det til standard C++ kildekode. Deretter kan kildekoden kompileres med hvilken som helst kompilering som er kompatibel med standard C++. (Qt, 2015)

## Signals og Slots i Qt

Signals og slots blir brukt til kommunikasjon mellom objekter. «Signal» oppfører seg som navnet tilsier et signal. En «slot» er en mottaker som utfører en oppgave, slik som en ordinær funksjon. Bestemte signal og slots blir tilordnet hverandre slik at de hører sammen. Når et signal blir sendt utfører den tilhørende slot-en sin oppgave. Det kan også overleveres objekter, med garantert typesikkerhet (Bennett, 2005).

## Powerline

Powerline er en nettverksteknologi som gjør at vi kan ta i bruk strømmettet i huset som nettverkskabler. Denne teknologien ble først tatt i bruk rundt 1922 på høyspent kabler i Europa med en frekvens mellom 15-500kHz (Doster).

## Switched Mode Power Supply

En SMPS er en type strømforsyning som regulerer strøm eller spenning ved hjelp brytere og energilagringselement som kondensatorer og spoler. Ideelt er disse kretsene tapsfri og gir en energioverføring på 100%. Buck konverter og Boost konverter er to typer SMPS som henholdsvis kun regulerer spenningen nedover og oppover.

## Nettverkskommunikasjon

### Ethernet

Ethernet er en standard for kommunikasjon mellom datamaskiner og er spesifisert av standarden IEEE802.3. Ut fra denne standarden er det utviklet flere typer med modifikasjoner, ofte for å øke hastigheten eller tilpasninger for nisjeområdet. En Ethernet switch støtter ofte flere av disse standardene. IEEE802.3 støtter opp til 10Mbit/s, mens IEEE802.3z takler 1000Mbit/s. (Jamfes F.Kurose, 2013)

## **TCP-Transmission Control Protocol**

På 1970 tallet eksisterte det flere former for nettverk. Vinton Cerf og Robert Kahn oppdaget behovet for en protokoll som kunne brukes på tvers av disse nettverkene. De kalte den for TCP/IP, som står for Transmission Control Protocol / Internet Protocol. Denne protokollen er under kontinuerlig oppdatering for å fortsette å støtte nye plattformer og nettverksformer. TCP kommunikasjon er internettets mest brukte kommunikasjonsprotokoll.

Når en TCP-kommunikasjon opprettes utføres det først en "Handshake". I denne oppstartsfasen av kommunikasjonen utveksles det innledende segmenter for å sette parameter før etterfølgende datatransmisjon. Segmentene som sendes inneholder tilstandsvariabler fra hver ende, server og klient. Hver påfølgende pakke etter handshake har feilsjekk og gjør forsøk på å sende på nytt dersom pakken ikke er komplett. TCP kommunikasjon er «stream»-basert, og den vil selv dele data du sender inn i deler på størrelsen til MSS (Maximum Segment Size). (Jamfes F.Kurose, 2013)

## **MTU- Maximum Transmission Unit**

MTU er den største pakken i link-laget som kan sendes av klient/server gjennom nettverket. Når MTU er satt, settes Maximum Segment Size (MSS) for at segmentstørrelsen ikke skal overskride MTU. MSS tilsvarer mengden data som kan sendes og inkluderer ikke header. Bestemmelsen for MTU blir satt i TCP handshake. (Jamfes F.Kurose, 2013)

## **GigE Vision**

GigE Vision er en grensesnitt-standard for kamerakommunikasjon over Gigabit Ethernet; protokollen standarden er basert på. GigE Vision støtter båndbredder på 125MB/s med CAT5e eller CAT6 kabler på opptil 100 meter med standard RJ-45 plugger. (Association, 2016)

## **UVC –USB Video Class**

UVC står for USB Video Class og ble først utgitt i 2003. UVC er en standard for å overføre video og lyd serielt som kan gjenskapes synkront. UVC er laget for å være en fleksibel protokoll. UVC beskriver hvilken enhet som er master og hvilken enhet som er klient, dette fordi rollene til UVC-enheter har lett for å bli blandet fordi protokollen er så fleksibel. Protokollen beskriver minimumskrav til funksjonalitet for at en enhet skal kunne kalles UVC kompatibel. (usb.org, u.d.)

## **Codex**

En Codex sin oppgave er å enkode eller dekode data. Ordene enkode og dekode kan veksles med komprimering og dekomprimering. Enkoding blir brukt for å effektivisere overføring av data, helst ved å endre strukturen på dataen som skal sendes. Resultatet ved enkoding er frigjøring av båndbredde og senking av tiden det tar for overføring av data. Enkoding kan også frigjøre lagringsplass. Dekoding av et enkodet bilde gjør det mulig å få tilgang til all informasjon av hele bildet, noe som ikke er mulig med enkodet bilde.

## DirectShow

DirectShow er designet for å takle utfordringer knyttet til audio- og videosynkronisering, hardware kompatibilitet og multi-format støtte. Målet med DirectShow er å forenkle arbeidet med å lage mediaapplikasjoner.

DirectShow bruker Direct3D og DirectSound, disse bruker grafikkort og lydkort til å effektivt prosessere data. DirectShow er laget for Windows applikasjoner og er ikke kompatibel med Linux systemer. (Microsoft, u.d.)

## FFMPEG

FFMEG står for "Fast Forward Moving Pictures Expert Group". FFMPEG er ledende innen multimedia rammeverket, med muligheter for å dekode, enkode, transkode, mux, demux, stream, filter og spille av tilnærmet alle formater laget av mennesker og maskiner. FFMPEG er kompatibelt med Linux, Mac OS X, Microsoft Windows, BSD, Solaris. (FFMPEG About, u.d.)

## Xuggler

Xuggler er laget for å lett dekode, modifisere og enkode mediafiler eller «stream» i Java. For å komprimere og dekomprimere mediafiler bruker Xuggler FFMPEG. Xuggler støtter Windows, MAC OSX og Linux. (Xuggler About, u.d.)

## GStreamer

GStreamer er et rammeverk for streaming av media. GStreamer er bygget på Video Pipeline fra Oregon Gratuat Institute og har hentet inspirasjon fra Microsoft sin DirectShow. Gstreamer er bygget for å takle hvilken som helst datastrømning i tillegg til media. Designet har lite overhead i streaming som gjør at forsinkelse blir minimal. (Gstreamer About, u.d.)

## Mikrokontroller

Enkle datamaskiner ment for innebygde kontrollapplikasjoner som kan plasseres på en enkelt silikonchip, kan kalles en mikrokontroller. Innebygde kontrollapplikasjoner er en del av et produkt, som for eksempel mobiltelefoner, printere, brødbakemaskiner og lignende elektroniske enheter. (Hambley, Electrical Engineering: Principles and Applications, 2014)

## OpenCV – Open Source Computer Vision Library

OpenCV er et maskinsyn og maskinlæringsbibliotek med over 2500 optimaliserte algoritmer. Biblioteket inneholder et omfattende sett med klassiske og "state of the art" maskinsyn og maskinlæringsalgoritmer. Algoritmene kan bli brukt for å for eksempel gjenkjenne ansikt og objekter. OpenCV er skrevet i C++. (OpenCV, 2015)



Figur 80. OpenCV logo.

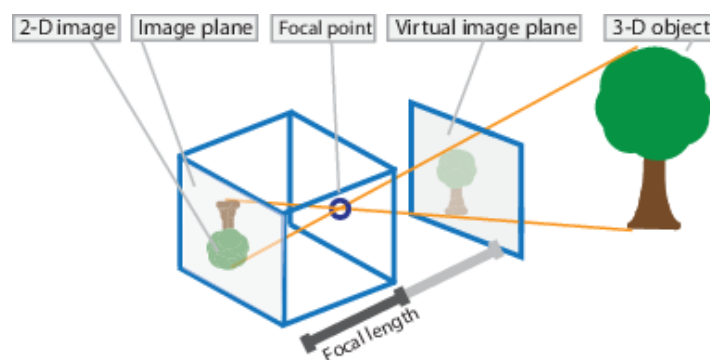


## Kamerakalibrering

Billigere type kameraer bruker noe som heter "pinhole" (også kalt "pinhole kamera"), men problemet med disse kameraene er at de har merkbar forvrengning i bildet.

Under vises hvordan et pinhole kamera fungerer.

- 2D image – bilderesultat.
- Image plane/Bildeplan – bildesensor
- Focal point/Brennpunkt – blenderåpning.
- Virtual image plane – bare en virtuell presentasjon av bildet.
- 3D - objektet i rommet.
- Focal length – lengden fra bildeplan til linsen.



Figur 81. Illustrasjon av perspektiv.

Forvrengningen i bildet forblir konstant så lenge focal length forblir konstant og kan korrigeres ved hjelp av kamerakalibrering. Kamerakalibrering (kamera kalibrering, u.d.) innebærer å estimere parameterne til linsen og bildesensoren i et kamera eller videokamera. Parameterne er gitt av en 4x3 matrise som blir kalt kameramatrikse og mapper 3D-romkoordinater til bildekoordinat. Kameramatriksen blir funnet ved hjelp av «extrinsic og intrinsic» parameterne.

$$w [x \ y \ 1] = [X \ Y \ Z \ 1] P$$

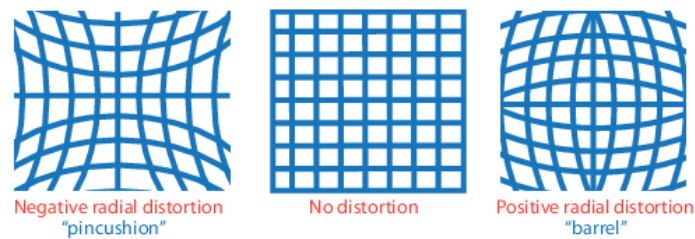
*Scale factor*     *Image points*     *World points*

$$P = \begin{bmatrix} R \\ t \end{bmatrix} K$$

*Camera matrix*     *Extrinsics*     *Intrinsic matrix*  
*Rotation and translation*

Figur 82. Parameterne.

Linseforvrengning oppstår når lysstrålene bøyes mer på kantene enn de gjør i senter. Kameramatriksen inneholder ikke linseforvrengningen og blir regnet ut i tillegg.



Figur 83. Kameraforvringning.

Den radiale forvringningen blir representert av:

$$X_{forvringt} = x(1 + k_1 * R^2 + k_2 * R^4 + k_3 * R^6)$$

$$Y_{forvringt} = y(1 + k_1 * R^2 + k_2 * R^4 + k_3 * R^6)$$

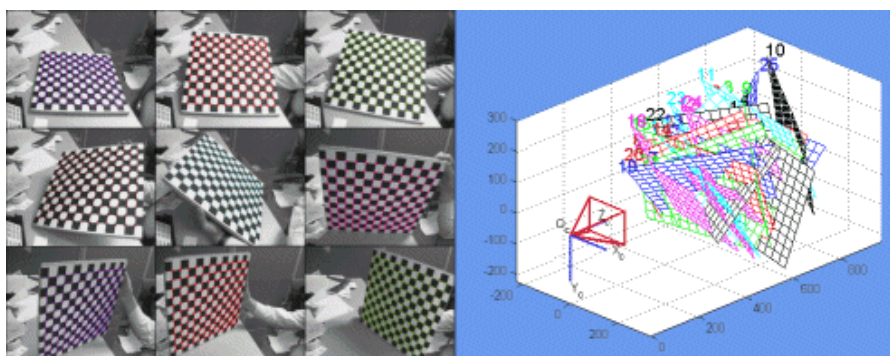
X, Y er ikke-forvringte pikselposisjoner og er normaliserte koordinater.

K1, k2 og k3 er radiale forvringningskoeffisienter til linsen. I dette tilfellet brukes alle tre forvringningskoeffisientene ettersom vi har en vidvinkellinse (med en vanlig linse så holder det med to). Den tangentielle forvringningen er når linsen ikke ligger parallelt med kamerasensoren. Det blir ikke tatt høyde for dette i denne rapporten.

Kameramatriksen og linseparameterne kan bli brukt for å korrigere linsefeil, måle fysisk størrelse på objekt i et bilde eller finne posisjonen til kameraet i romkoordinat.

### Matlab Computer Vision System Toolbox

Matlab har en toolbox som heter "Computer Vision System Toolbox" (toolbox, u.d.) og inneholder en applikasjon som heter «Camera calibrator» (CC). CC bruker sjakkmønstermetoden for å kalibrere kamera og krever minimum to bilder (helst flere) med sjakkmønster og størrelsen på sjakkrutene i millimeter for å kunne kjøres. Sjakkbrettet blir plassert foran gjeldende kamera som skal kalibreres og det blir tatt så mange bilder som mulig hvor sjakkmønsteret blir flyttet og vridt for å få flest mulige vinkler og posisjoner (se figur 83). Kalibratoren analyserer alle bildene og returnerer kameraparameterne.



Figur 84. Sjakkmønster og kamera plassering (kameraplassering, u.d.)

## OpenCV Kamerakalibrering

OpenCV har en egen del som tar seg av kamerakalibrering (Matlab, u.d.). I likhet med Matlab bruker også OpenCV sjakkmonstermetoden for å kalibrere kamera.

## Videostabilisering

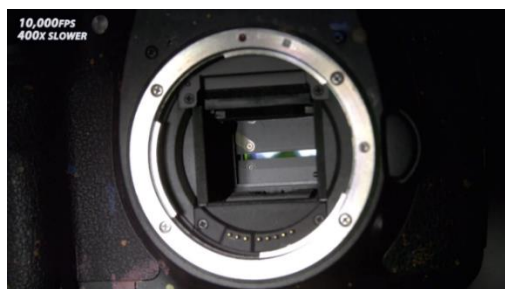
Kamera som blir utsatt for ytre påkjenning i form av risting o.l. vil vises i bildet og redusere kvaliteten. Det finnes innretninger kalt mekanisk gyrostabilisator som tar opp denne ristingen og fungerer som en demper for kameraet. I denne rapporten blir det brukt to typer stabilisering: ren bildebehandlingsstabilisering og stabilisering ved hjelp av gyroskop.



Figur 85. Gimball for mobil (stabilisator, u.d.).

## Rolling Shutter

«Rolling shutter» er en metode for å ta bilder og noe man finner i CMOS-kamera. Ved global shutter starter det med at bildesensoren ikke er utsatt for lys, for deretter å bli utsatt for lys i (for eksempel) ett millisekund. Etter millisekundet blir den tildekket igjen og på denne måten får man bildet. (DIY Photography, 2012)



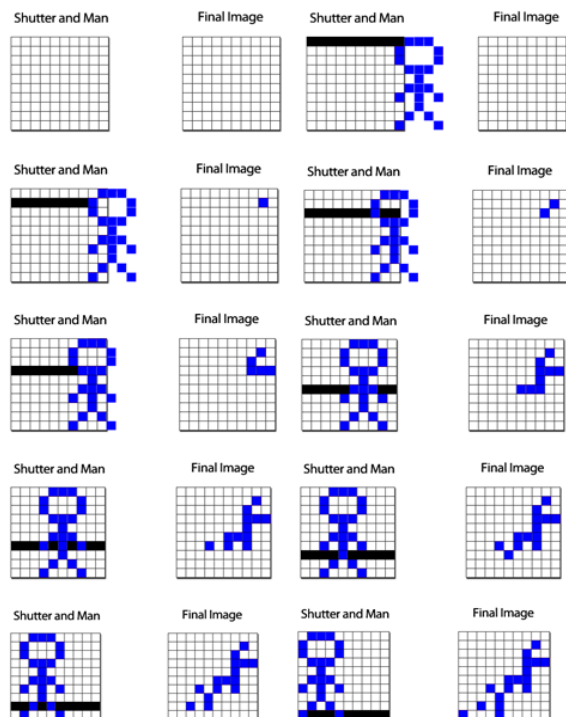
Figur 86. Rolling shutter kamera.

(kamera, u.d.)

Ved rolling shutter kan man se for seg en plate som starter på toppen av bildesensoren og ruller nedover. Bildesensoren utsettes da for lys over en gitt tid. Problemet med rolling shutter er at toppen og bunnen av bildesensoren blir utsatt for lys til litt forskjellige tidspunkt. Før brukte de fysiske luker, men nå forteller de bare bildesensoren hvilken rad av bildesensoren som skal ta til seg lys.

Fordelene med rolling shutter er at de er raskere, billigere og har mindre effektbruk. Når man skal ta video så må sensoren kanskje ta opptil 60 bilder i sekundet og med CCD så blir de for dyre. CMOS blir ofte brukt når det er snakk om video siden de er raskere.

Problemet med CMOS er når gjenstander i bildet beveger seg raskt, som for eksempel en propell, eller hvis kameraet ristes. Rolling shutter effekten oppstår når en gjenstand flytter seg så raskt at det oppstår en forvrengning i bildet. Forvrengningen oppstår fordi gjenstanden flytter seg i løpet av den tiden sensoren bruker på å forflytte seg fra første til siste rad.



Figur 87 Rolling shutter bildetagning.

(DIY Photography, 2012)

## Kinematikk

Kinematikk er studiet av bevegelse og utvikling av likninger for å beskrive objekters bevegelser. Det er en del av en større komponent av fysikken kjent som klassisk mekanikk. Kinematikk kan brukes til alt fra studier om hvordan en hest løper eller hvordan en partikkel beveger seg i rommet. To undergrupper av kinematikken er forover kinematikk og invers kinematikk (IK). Forover kinematikk gir et punkt (en posisjon) ut ifra hva som blir matet inn i bevegelseslikningen, mens invers kinematikk brukes til å regne ut hvordan objektet må bevege seg for å oppnå ønsket posisjon. En robotarm bruker invers kinematikk til å regne ut hvilke vinkler hvert ledd må ha for å kunne nå et ønsket mål (posisjon). Invers kinematikken kan ha mange forskjellige løsninger, derfor er det viktig å finne den beste løsningen. I felt som animasjon er det viktig å kunne skape realisme og dynamiske bevegelser, og kinematiske likninger ligger til grunn for dataanimasjon for film, spill og andre aktiviteter. (Not My War, u.d.)

## Kontroll av ben med sinusfunksjoner

Sinuskurvegeneratoren kan produsere veldig smidige og sømløse bevegelser i tillegg til å gjøre kontrolleren enklere. Hvert ledd i en robotarm representerer en sinuskurve. Med rett faseforskyvning og amplitude kan sinuskurvene bli satt opp slik at de styrer et bein på en robot. Vinkelen til hver ledd er gitt av likning (7) under:

$$y_i = A_i \sin\left(\frac{2\pi}{T}t + \phi_i\right) + O_i \quad (1)$$

Hvor  $y_i$  er vinkelen til ledd;  $A_i$  er amplituden;  $T$  er perioden;  $t$  er periode tiden;  $\Phi$  er fasen;  $O_i$  er offseten (Zhang, 2015).

## Jacobian

Et system hvor et sett med linker er sammenkoblet av flere ledd kalles et rigid multibody system. "Rigid Multibody System" fungerer som en robotarm, samt er grunnlaget for animasjoner innenfor spillgrafikk. For å kontrollere bevegelsen til dette multibody systemet blir det i de fleste tilfeller brukt invers kinematikk. Der er flere måter å løse et IK problem som stammer fra robotapplikasjoner. Deriblant pseudoinvers jacobian metoden og jacobian transponert metoden. Den komplette konfigurasjonen til et multibody system er spesifisert av skalarene  $\theta_1, \dots, \theta_n$  som beskriver vinklene til leddene. Endepunktet  $\vec{s}$  er en funksjon av leddvinklene. Multibody systemet vil bli kontrollert ved å spesifisere et endepunkt. Ønsket posisjon blir gitt av en vektor  $\vec{p}$ . Vi lar  $\vec{e} = \vec{p} - \vec{s}$  være forflytningen av endepunktet, en «error». Leddvinklene blir skrevet som en kolonnevektor  $\theta = (\theta_1, \dots, \theta_n)$ . Endepunktet kan uttrykkes som  $\vec{s} = \vec{s}(\theta)$ , IK problemet er å finne verdier for  $\theta$  sånn at

$$\vec{p} = \vec{s}(\theta) \quad (2)$$

Uheldigvis er det ikke alltid det finnes en løsning av likning (2). Det kan derimot brukes en iterativ metode for å tilnærme seg en løsning. Med denne metoden blir funksjonen  $s(\theta)$  lineært tilnærmet ved å bruke Jacobian matrisen. Jacobian matrisen  $J$  er en funksjon av verdiene  $\theta$  og er definert som

$$J(\theta) = \left( \frac{\partial s}{\partial \theta} \right)_{ji} \quad (3)$$

En oppdatert vinkel  $\Delta\theta$  blir funnet med hensikt å inkrementere leddvinklene  $\theta$  med  $\Delta\theta$ :

$$\theta = \theta + \Delta\theta \quad (4)$$

Ved forandring av leddvinklene kan endepunktet estimeres som

$$\Delta \vec{s} \approx J \Delta\theta \quad (5)$$

Verdien  $\Delta\theta$  må bli valgt slik at  $\Delta \vec{s}$  blir tilnærmet lik  $\vec{e}$ . Det finnes to metoder for å finne en oppdatert vinkel  $\Delta\theta$ . (i) Hvert steg i simuleringen utfører en singel oppdatering av leddvinklene slik at endepunktet omtrent følger destinasjonspunktet. (ii) Leddvinklene oppdateres iterativt inntil verdien av  $S$  er tilstrekkelig nært endepunktet. En metode for å finne  $\Delta\theta$  er å bruke den pseudoinverse Jacobian matrisen. Den pseudoinverse matrisen setter

$$\Delta\theta = J^\dagger \vec{e} \quad (6)$$

Hvor

$$J^\dagger = J^T (JJ^T)^{-1} \quad (7)$$

endepunktet tilnærmer seg det ønskede punkt  $\vec{p}$ . (ii) Leddvinklene oppdateres iterativt helt til at en verdi er tilstrekkelig nært en løsning. Hvor  $n \times m$  matrisen  $J^\dagger$  er den pseudoinverse matrisen av  $J$ , også kjent som den Moore-Penrose invers matrise av  $J$ . Denne matrisen er definert for alle  $J$  selv de som ikke er rektangulære. Den pseudoinverse matrisen gir den beste løsningen til likningen  $J\Delta\theta = \vec{e}$  med hensyn på minste kvadraters metode (Bjørnstad, 2009).

Den pseudoinverse pleier å ha stabilitetsproblem når den nærmer seg singularitet. Ved singularitet vil Jacobian matrisen ikke lenger ha «full row rank», samsvarende med det faktum at det er en bevegelse av endepunktet som ikke er oppnåelig. Dersom matrisen er nøyaktig i singularitet vil den oppføre seg. Er den nært singularitet vil den pseudoinverse metoden vil selv den minste bevegelse føre til en stor endring i ledd vinklene. For å unngå singularitet må det kontrolleres for verdier som nærmer seg null. (Buss, Webområde for University of San Diego, 2009)

## Gangarter og gangen til en krabbe og et insekt

En gangart er rekkefølgen eller måten landingen og løftingen av beina skjer på hos en mangebeinet robot. Gangarten angir en gå- eller løpeprosedyre. Hvert bein har to faser under gange: støttefasen (holdefasen) og translasjonsfasen (svingfasen). Når roboten er i bevegelse vil rekkefølgen og kombinasjonen mellom holde- og svingfasene definere gangen til roboten og en bane skal følges i svingfasen for hvert bein.

### Analyse av gangart

Det finnes et stort antall studier og litteratur om robotgange med et bredt spekter av definisjoner og applikasjoner. En gjennomgang identifiserte hovedsakelig to typer gange: fri gange (ikke periodisk) og periodisk gange.

Fri gange følger ikke en periodisk oppførsel. Hver gange som ikke er periodisk er en fri gange eller en ikke-periodisk gange. En fri gange er nyttig når roboten skal bevege seg i ukjent miljø og ulendt terreng. Å lage en fri gang er i stor grad en mer komplisert å programmere.

I periodisk gange vil hvert ben utføre samme bevegelse og oppføre seg på samme måte periodisk. Bølge, krype, krabbe, snu, tripod og tetrapodgange er forskjellige typer av periodisk gange. Den mest kjente periodiske gangen for en gående robot er bølgegangen hvor beina har en bølgebevegelse fra front til bak. Det er bevist at bølgegangen er den mest stabile gangen for en periodisk gange. En periodisk gange er en del enklere å programmere ettersom den ikke tar hensyn til omgivelsene.

Noen viktige definisjoner for en periodisk gange:

- **Standard gange** er en gange hvor alle beina har samme «duty factor»
- **Duty factor** ( $\beta$ ) er forholdet mellom støtte og translasjonsfasen i syklustiden.
- **Tripod gange** er en bølge gange med «duty factor» 0.5 ( $\beta = 0.5$ )
- **Syklustiden** er tiden det tar for å gjennomføre et steg.
- **Stride lengde** (steglengde) er lengden kroppen beveger seg etter en syklus.
- **Kinematisk syklusfase** ( $\phi$ ) er forflytningen av robotens kropp siden siste plassering av bein 1 normalisert av «stride lengden» Det kan også defineres som tid normalisert av en syklustid.
- **Relativ fase** av et bein ( $\varphi_i$ ) er «touch-down» øyeblikket normalisert av en syklustid. Med andre ord er det verdien til kinemtikfasen når et bein når bakken. Det er antatt at  $\varphi_i = 0$ .
- **Symmetrisk gange** er gangen når hvert høyre-venstre bein par er en halv syklustid ut av fase i forhold til hverandre.
- **Forover bølgegange** er en bølgegange hvor løftingen av beina begynner bak på roboten og forover som en bølge og hvert høyre-venstre bein par har en faseforskjell på  $(1 - \beta)$ . Når et av beina trekker ned på bakken vil det motsatte parbeine løfte seg samtidig.
- **Symmetrisk forover bølge gange** er forover gangen hvor den relative fasen mellom beina 1, 3, 5 og 2, 4, 6 er 0.5 (HAJIABADI, 2013).

## Fordeler med hexapod

Hexapodroboter har et stort bruksområde fra å forflytte seg over ulendt terreng til å utføre redningsoperasjoner i farlige og uforutsigbare katastrofeområder. Fordelene med en hexapodrobot over en som er utstyrt med hjul, quadruped og bipedal roboter er mange.

Roboter med hjul er raskere over jevnt terreng, men hexapodroboten den raskeste roboten med bein fordi den har det optimale nummeret bein for rask gange. Det er bevist at farten ikke øker med flere antall bein enn seks. Hexapodroboten er også overlegen hjulroboten siden hjulroboten trenger et flatt og jevnt terreng for å kunne manøvrere, i motsetning til hexapodroboten som kan gå over hindringer og velge fotfeste som gjør at den forblir stabil. Hexapodroboten har også muligheten til å snu seg rundt på stedet ved hjelp av de manøvrerbare beina.

Siden en hexapodrobot har 6 bein kan den også bruke to av beina til å utføre arbeid og reparasjoner uten å miste stabiliteten. Alle disse fordelene gjør hexapodroboten den mest vanlige roboten å bruke (Jones, 2011).

## Rotasjonsmatriser

I et  $R^2$  koordinatsystem med en matrise som roterer en vektor  $v_0$  med vinkel  $\theta$  mot klokken er

$$R_\theta = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \quad (8)$$

Sånn at

$$V' = R_\theta V_0 \quad (9)$$

I et  $R^3$  koordinatsystem hvor koordinatsystemet rotere rundt x, y og z aksene mot klokken vil matrisene se sånn ut (Weisstein, u.d.):

$$R_x(\alpha) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{pmatrix} \quad (10)$$

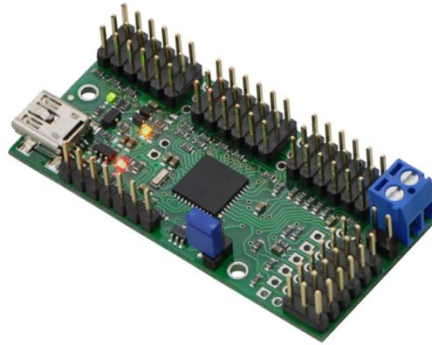
$$R_y(\beta) = \begin{pmatrix} \cos \beta & 0 & -\sin \beta \\ 0 & 1 & 0 \\ \sin \beta & 0 & \cos \beta \end{pmatrix} \quad (11)$$

$$R_z(\gamma) = \begin{pmatrix} \cos \gamma & \sin \gamma & 0 \\ -\sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (12)$$



### 3.2.2 MATERIAL OG METODE

#### Mini Maestro 24-kanal USB Servo Controller



Figur 88. Maestro 24-kanals.

(Pololu, u.d.)

Mini maestro servokontroller er et allsidig og kompakt servokontroller og generelt I/O kort. Kortet støtter 3 forskjellige styremetoder: USB for direkte kontakt med en datamaskin, TTL Serial (All About Circuits, u.d.) for å kommunisere med innebygde system og internt skript for selvkontrollerende applikasjoner. Kanalene på kortet kan konfigureres som en servo utgang for å kontrollere en RC-servo eller en ESC (electronic speed controller), som en digital utgang eller som en analog/digital input. Den ekstremt høye presisjonen og høye oppløsningen til servopulsene gir en svingning av servoene på mindre enn 200ns, som gjør denne kontrolleren svært egnet til applikasjoner med høy presisjon. Med innebygget farts- og akselerasjonskontroll for hver kanal er det enklere å oppnå en smidig og sømløs bevegelse. Kortet har en pulsvidde på 64 til 4800 mikrosekund og en oppløsning på 0.25 mikrosekund som utgjør en endring på en vanlig servo til 0.025grader.

#### XBOX ONE kontroll

XBOX ONE-kontrolleren er i utgangspunktet en kontroller til Microsofts XBOX konsoll. Den er også kompatibel med PC via USB som gjør det mulig å bruke den i dataspill. (microsoft, u.d.)



Figur 89. XBOX-One kontroll.

(Microsoft, u.d.)



### Levi Hobby RS0610MD robotic servo

RS0610MD er en digital hobby servo som har metallgir for å takle den høye kraften på 6 kg/cm. Den har en motor med børster som har en arbeidsspenning på 4.8V til 7.4V. Servoen veier kun 51 gram, noe som gjør den egnet til roboter som er kritisk til vekt. Firmaet kan levere denne servoen med et arbeidsområde på 180, 300, 360 grader.



Figur 90. Levi Hobby servo.

(Levi Hobby, u.d.)

### T100 Thruster

T100 thrusteren fra Blue Robotics er en thruster designet spesielt for ROV-er. Det er en høykvalitets thruster med over 2 kg thruster kraft. Thrusteren er godt egnet til bruk i det åpne hav og på store dyp. Forskjellige monteringsmuligheter, enkel kontrollering og lav pris gjør det til en perfekt thruster til maritime ROV-er. Thrusteren har en børsteløs motor, en arbeidsspenning på 12 volt og en effekt på 130 watt. T100 er laget av en UV-resistent, høykvalitets polykarbonatplastikk. Kjernen på motoren er beskyttet av et epoksybelegg og bruker plastikkager i stedet for stållager, da stållager rustet i saltvann (Blue Robotics, u.d.).



Figur 91. BlueRobotics T100.

(Blue Robotics, u.d.)

### Adafruit 10DOF IMU (inertial-measurement-unit)

Denne IMU'n kombinerer 3 kvalitetssensorer som til sammen gir 11 akser. 3 akser fra akselerometeret, 3 akser fra gyroskopet, 3 akser fra magnetometeret (kompass), barometrisk trykk og temperatur. Gyroskopet L3DG20H, Akselerometeret LSM303DLHC, kompasset og temperatur BMP180 bruker alle I2C som gjør at en kan kommunisere med alle sensorene med kun 2 kabler. Man leser ut data ved hjelp av seriellkommunikasjon og brettet tar fra 3-5v og er polaritetssikret slik den ikke blir ødelagt om polariteten er feil (adafruit10dof, u.d.).

- L3GD20H 3-axis Gyroskop:  $\pm 250$ ,  $\pm 500$ , or  $\pm 2000$  grader-i-sekundet skalering
- LSM303 3-axis kompass:  $\pm 1.3$  to  $\pm 8.1$  gauss magnetisk felt skalering
- LSM303 3-axis akselerometer:  $\pm 2g/\pm 4g/\pm 8g/\pm 16g$  Valgbar skalering
- BMP180 Barometrisk trykk/temperatur:  $-40$  to  $85$  °C, 300 - 1100hPa range, 0.17m oppløsning



Figur 92. Adafruit 10DOF IMU.

### ELP USB500W02M-L170

Et 5 megapiksel mini USB kamera som kommer med 170-graders fish-eye-linse (ELP Surveillance, u.d.). Bildesensoren (bildesensor, u.d.) er en CMOS OV5640 med oppløsninger fra 320x240 og 30fps til høyest 2592x1944 og 15fps. Kameraet støtter UVC.



Figur 93. ELP 170.

(ELP Surveillance, u.d.)

## Arduino UNO

Arduino UNO R3 er et kort som inneholder mikrokontrollertillaget for enkel tilkobling av utstyr. UNO R3 har en ATmega328 mikrokontroller, 32Kbyte flash minne og 16MHz klokkehastighet. Kortet har seks analoge inputs, 14 digitale I/O pins hvor seks av de er PWM utganger, kortet trenger en forsyning på 7-12V. (Arduino, u.d.)

## Teensy 3.2

Teensy 3.2 (Teensy, u.d.) er et lite utviklingskort med mange muligheter. Det gjør at det kun trengs å lodde inn og utganger, og programmere kontrolleren. Den kan programmeres i C eller med en utvidelse av Arduino programvaren. Dette gjør at kundemarkedet strekker seg fra profesjonelle til entusiaster. En av fordelene med Teensy er at det ikke er behov for egen programmerer fordi bootloader er allerede flashet på chip. Teensy 3.2 har en 32bit ARM Cortex-M4 72MHz CPU, 256K flash minne, 64K RAM og 2K EEPROM.



Figur 94. Teensy 3.2.

## Homeplug AV2

ZyXEL AV2000 HOMEPLUG AV2 Powerline 2-port Gigabit Pass-Thru Ethernet adapter starter kit inneholder to enheter. Kommunikasjonen mellom enhetene bruker MIMO (Multiple Input Multiple Output), som gjør dette til en av de raskere på markedet, flere andre adaptere bruke SISO (Single Input Single Output) som ikke klarer å levere samme ytelse. Grunnen til at MIMO er raskere er at den tar i bruk jordleder som signalbærer i tillegg til faseledere, SISO bruker kun faseledere som signalbærere (QT Co., u.d.).

## Odroid XU4

Odroid XU4 er en liten datamaskin som egner seg godt til innebygde systemer. XU4 kan kjøre med forskjellige versjoner av Linux.

XU4 tar i bruk eMMC 5.0, USB 3.0 og Gigabit Ethernet interface som gjør at produktet har rask oppstartstid og overføringshastighet.

**Spesifikasjoner:**

- Samsung Exynos5422 Cortex –A15 2GHz og Cotex-A7 åtte-kjerners prosessorer.
- Mali-T628 MP6(OpenGL ES 3.0/2.0/1.1 og OpenGL 1.1 Full profil)
- 2GigaByte LPDDR3 RAM PoP stacked
- EMMC5.0 HS400 flash lagring
- 2x USB3.0
- 1x USB2.0
- Gigabit Ethernet port
- HDMI 1.4a til display
- 8.2 x 5.8 x 2.2 cm



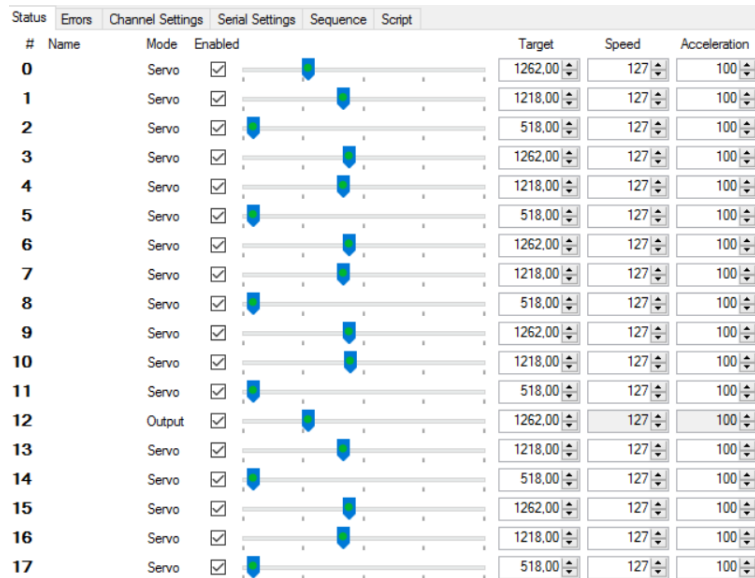
Figur 95. ODROID XU4.

**Matlab**

Matlab er et matematikkprogram som brukes til numeriske beregninger, simuleringer og visualisering. Navnet Matlab er en forkortelse for «Matrix laboratory» og er sentrert rundt matrise. Matlab er derfor best på matriseregning. (Matlab, u.d.)

## Pololu Maestro Control Center

Pololu maestro brukersnitt gir tilgang til konfigurasjonsinnstillinger i tillegg til muligheten for real-time styring, tilbakemelding og debugging. Maestro kontrollsentret er et grafisk verktøy som gjør det enkelt å bruke USB brukersnittet. Det er et nyttig verktøy for å gjøre tester med i prosjektet. Et lite utklipp av kontrollsentret er vist i figur 95 (Pololu, u.d.).



#	Name	Mode	Enabled	Sequence	Target	Speed	Acceleration
0	Servo	<input checked="" type="checkbox"/>			1262,00	127	100
1	Servo	<input checked="" type="checkbox"/>			1218,00	127	100
2	Servo	<input checked="" type="checkbox"/>			518,00	127	100
3	Servo	<input checked="" type="checkbox"/>			1262,00	127	100
4	Servo	<input checked="" type="checkbox"/>			1218,00	127	100
5	Servo	<input checked="" type="checkbox"/>			518,00	127	100
6	Servo	<input checked="" type="checkbox"/>			1262,00	127	100
7	Servo	<input checked="" type="checkbox"/>			1218,00	127	100
8	Servo	<input checked="" type="checkbox"/>			518,00	127	100
9	Servo	<input checked="" type="checkbox"/>			1262,00	127	100
10	Servo	<input checked="" type="checkbox"/>			1218,00	127	100
11	Servo	<input checked="" type="checkbox"/>			518,00	127	100
12	Output	<input checked="" type="checkbox"/>			1262,00	127	100
13	Servo	<input checked="" type="checkbox"/>			1218,00	127	100
14	Servo	<input checked="" type="checkbox"/>			518,00	127	100
15	Servo	<input checked="" type="checkbox"/>			1262,00	127	100
16	Servo	<input checked="" type="checkbox"/>			1218,00	127	100
17	Servo	<input checked="" type="checkbox"/>			518,00	127	100

Figur 96 Maestro kontrollsentret.

## Programvare

Programmer som er brukt:

- Microsoft Visual Studio
- Microsoft Word
- Microsoft Project
- Microsoft Excel
- Netbeans IDE
- Matlab
- Arduino IDE
- Qt Creator
- Pololu Maestro control center
- Draw.io
- Gimp
- CircuitLab

### 3.2.3 Metode

#### Systemoversikt

Kommunikasjonssystemet er delt inn i to deler. Det består av en kontrollstasjon og en ROV. Disse to delene er sammenkoblet via kabel og en vaier for styrke. Dette er den planlagte løsningen.

Kontrollstasjonen består av en datamaskin, kontrollapplikasjon og en XBOX ONE kontroller. Xbox kontrolleren skal brukes til å styre alt på ROV-en og kontrollapplikasjonen utenom avanserte innstillinger i kontrollapplikasjonen. Kontrollapplikasjonen blir referert til med navnene kontrollapplikasjon, GUI og klient applikasjon i denne rapporten. Kriteriene til Windows maskinen er at den kan kobles til Ethernet og USB.

ROV-ens kommunikasjon til kontrollapplikasjonen går først til en Odroid XU4, XU4 kan bli referert til som minimaskin i denne rapporten. Minimaskinen brukes som hovedenhet i ROV-en og tar seg av alt av kommunisering mellom interne komponenter i ROV og mot kontrollstasjonen. Alle kommuniserende enheter i ROV brukes seriell kommunikasjon over USB. Disse komponentene er koblet til minimaskinen som tar seg av når tid det skal skrives og leses til hver enkelt komponent. Mini maestro brukes til å styre servoer og thrustere, og kommuniserer med minimaskinen. Kamera snakker direkte med minimaskinen. Arduino brukes til å hente data fra sensorer. Arduinoen er et mellomledd mellom sensor og minimaskinen.

#### Valg av kamera

Kameraet har to formål; navigasjon og inspeksjon. For å navigere er det ønskelig med høy FPS og lav forsinkelse. Dette for at interaksjon fra brukeren skal være mest mulig synkront med bildet som vises på kontrollstasjonen. Ved inspeksjon er det ønskelig med et godt og høyoppløst bilde, for å bedre kunne tyde detaljer. For begge formål kan det være gunstig med bred synsvinkel for å kunne se mest mulig av omgivelsene.

Et annet kriterium for valg av kamera var at det benytter en kommunikasjonsprotokoll som lett lar seg implementere på en god måte med resten av systemet. Protokoller som er basert USB og Ethernet standardene er ønskelig, siden de er støttet av Odroid XU4.

Det har blitt sett på et bredt spekter av forskjellige typer kamera. Hovedsakelig fra kategoriene: ROV, Industrielt maskinsyn, overvåkning og webkamera/hobby.

Det har blitt funnet noen aktuelle kamerakandidater fra ROV- og overvåkningsbransjen, men de har blitt valgt bort. Generelt brukes det proprietære løsninger for strømforsyning og kommunikasjon, enten fysisk eller i programvare eller begge deler. Noe som vanskeligjør implementering.

Av industrielle kameraer har det blitt funnet flere aktuelle. Spesielt kameraer som benytter GigE Vision standarden til kommunikasjon. Ved å bruke GigE Vision kan kameraet kobles gjennom en switch til Ethernetforbindelsen opp til kontrollstasjonen på land. Det vil frigi Odroiden fra arbeidet å ta inn videostrømmen og sende den videre. Samtidig vil Odroiden kunne kommunisere egen data gjennom den samme switchen.

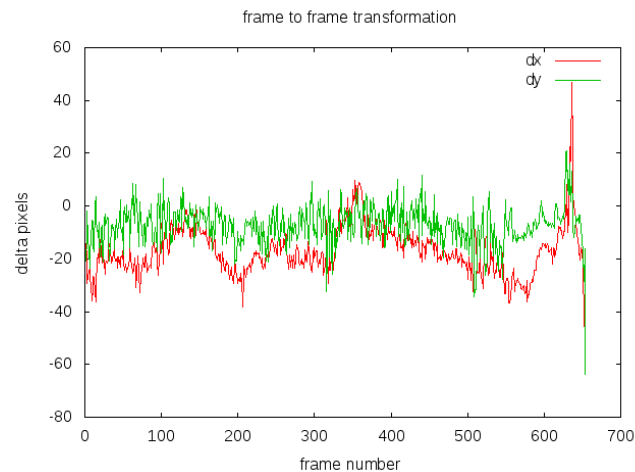
Et kamera av typen Jai GO-5000C-PGE med linsen Fujinon FE185C057HA-1 (se vedlegg på CD/mapper for mer info) ble sett på som en mulig GigE Vision basert løsning. Det vil gi en synsvinkel på opptil 185° med en oppløsning på opptil 5MP ved 22 FPS. Jai tilbyr også en gratis SDK for blant annet GigE Vision (Jai). Som kan forenkle implementering med kontrollstasjonen. Denne løsningen ble ikke valgt for utvikling av ROV-en grunnet kostnad. Kamera og linse kostet henholdsvis omtrent ti- og fire tusen (1st Vision, 2014).

For å ikke bruke unødvendig mye penger på utvikling, ble det bestemt å bruke et webkamera fra skolen. Det er av typen Logitech C920 og bruker UVC standarden til kommunikasjon. Den er støttet av blant annet OpenCV.



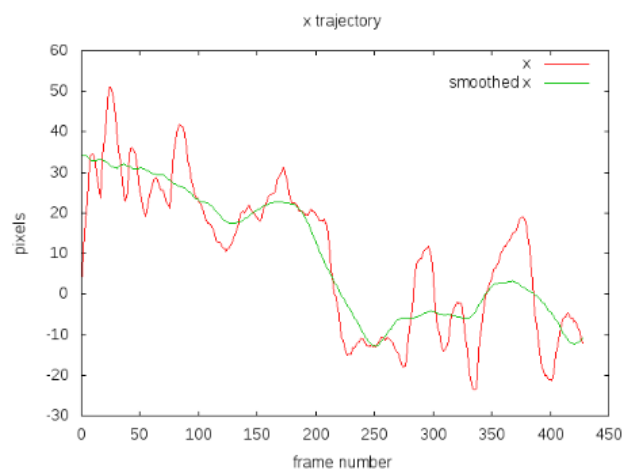
## Videostabilisering med ren bildebehandling

Med ren bildebehandling menes det at en ikke får noe input om hvor mye risting eller rotering kamera blir utsatt for. Man starter med å ta bilde 1 i en videofil og analyserer denne for å finne gode og tydelige særtrekk som kan brukes som referansepunkt. Det samme gjøres med alle bildene i videofilen. Med punktene fra bildene blir det funnet "previous to current" transformasjon som inneholder dx, dy og vinkelen.

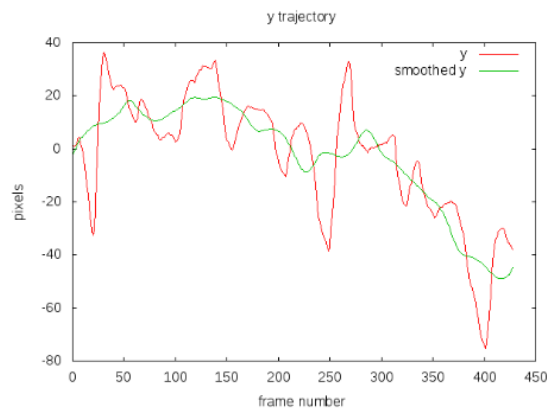


Figur 97. Bilde til bilde transformasjon.

For å stabilisere videoen glettes banen ut ved hjelp av "sliding average window". Radiusen for vinduet er antall bilderammer som blir brukt til glatting.



Figur 98. X-bane.



Figur 99. Y-bane.

Det blir laget en ny transform slik at  $ny\_transform = transform + (glattet\_bane - bane)$  og den nye transformen blir påtrykket videoen og ender opp med en stabilisert video. Det ble bygget om en eksempelkode (nghiaho, 2014) for å ta inn video i realtime.

Roll kompensering med hjelp av gyro

For å få bildet til å stå mest mulig vannrett ble det hentet roll-verdi i grader ut i fra gyroskopet. Rotasjonsmatrisen på formen:

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

blir funnet med funksjonen:

```
Mat rotx_mat = getRotationMatrix2D(Point(frame.cols / 2, frame.rows / 2), -30, 1);
```

i eksempelet blir bildet rotert med -30grader. Bildet blir så warpet med OpenCV funksjonen:

```
warpAffine(frame, warp, rotx_mat, Size(frame.cols, frame.rows));
```

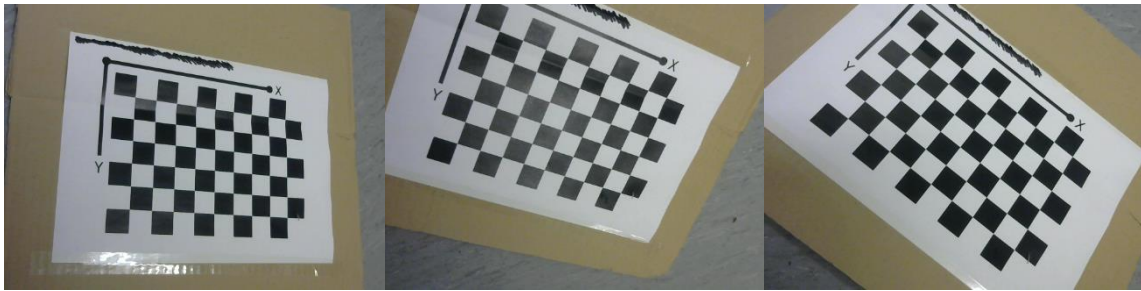
som tar inn bildet, rotasjonsmatrisen og størrelsen på bildet og roterer bildet rundt senter.

## Fisheye korrigering

### Kalibrering i Matlab

For å kunne analysere bildet bedre ble kameraet kalibrert for linsevregning og flatet ut. For å kalibrere kameraet ble det gjort en test med Matlab sin «toolboks» for å se resultatet. I testen ble det brukt et vanlig webkamera.

Først blir det tatt en rekke bilder av et sjakkbrett som vrís og flyttes rundt på.



Figur 100. Bilde av sjakkmønster.

Bildeserien kjøres i kalibrerings-"toolboxen" med rutestørrelse 23mm og får returnert intrinsic, extrinsic og en rekke andre kameraparameter. Parameterne blir satt inn i funksjonen *UndistortImage* sammen med bildet som skal korrigeres og andre valgfrie parameter. Resultatet er et bilde som er korrigert for «pinhole» og linsevregning (OpenCV, Opencv, u.d.).

### Kalibrering i OpenCV

For å kunne implementere kalibreringen i GUI-en ble det nødvendig å skrive kalibreringen i OpenCV. OpenCV har kalibreringsmetoder i likhet med Matlab og fremgangsmåten er den samme med å ta en rekke bilder av et sjakkbrett og analysere bildeserien (OpenCV, Opencv, u.d.).

Hvert bilde blir analysert med funksjonen

```
bool found = findChessboardCorners(image, board_sz, corners, CV_CALIB_CB_ADAPTIVE_THRESH | CV_CALIB_CB_FILTER_QUADS);
```

som returnerer TRUE hvis den finner sjakkmønsteret.

Input er image (bilde som skal analyseres), board\_sz (antall hjørner i horisontal og vertikal retning), corners (hvor han lagrer alle hjørnene) og resten er en rekke instillinger om bildebehandlingen. Om findChessboardCorner returnerer true fortsettes kalibreringen, om den er false så tar han et nytt bilde og leter på nytt etter sjakkmønster.

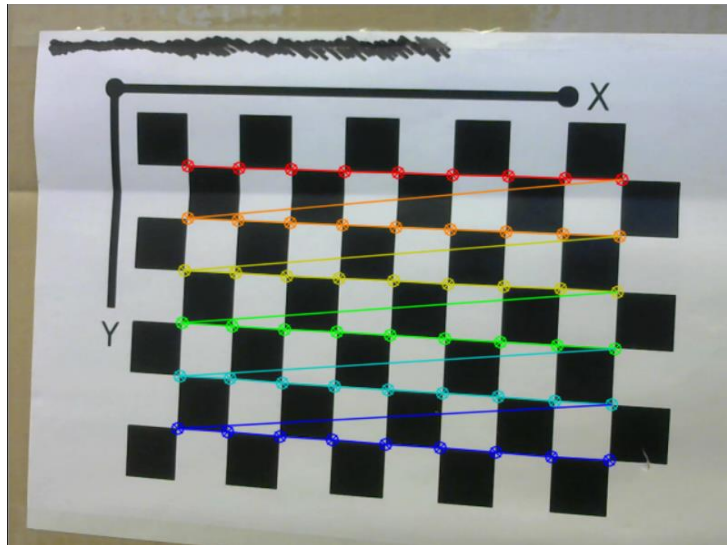
Siden Corners kun er en estimering av hjørnene så kjøres funksjonen

```
cornerSubPix(gray_image, corners, Size(11, 11), Size(-1, -1), TermCriteria(CV_TERMCRIT_EPS | CV_TERMCRIT_ITER, 23, 0.1));
```

for å få mer eksakte koordinater. Input er gråtonebilde av samme bildet som i forrige funksjon, samme med Corners og resten er parameter for søkingen. Det kan nevnes at 23 er størrelsen på rutene i sjakkmønstret. For å få ett visuelt syn på analyseringen tegnes punktene med

```
drawChessboardCorners(gray_image, board_sz, corners, found);
```

som vist på neste side.



Figur 101. Hjørner til sjakkmonster.

Kalibreringen kjøres med:

```
calibrateCamera(object_points, image_points, image.size(), intrinsic, distCoeffs, rvecs, tvecs);
```

Kalibreringen får returnert "intrinsic", distansekoeffisientene, "radial vector" og "tangential vector" matrisene. Disse blir brukt sammen med:

```
undistort(image, imageUndistorted, intrinsicOptimal, distCoeffs);
```

«Undistort» bruker matrisene til å korrigere bildene. Senere ble den samme kalibreringen gjort med ELP 170 kamera som blir brukt i ROV-en.

### Bevegelse med sinusfunksjoner

Metoden med å bruke sinusfunksjon for å styre servoene til ROV-en ble satt opp C++. Likning (1) ble brukt for å finne gradene til beina. Det ble først satt opp en timer i koden som teller millisekund. Den «timer»-en ble videre brukt for å regne ut vinkelen til servoene. Figuren under viser hvordan en sinus generator ble satt opp:

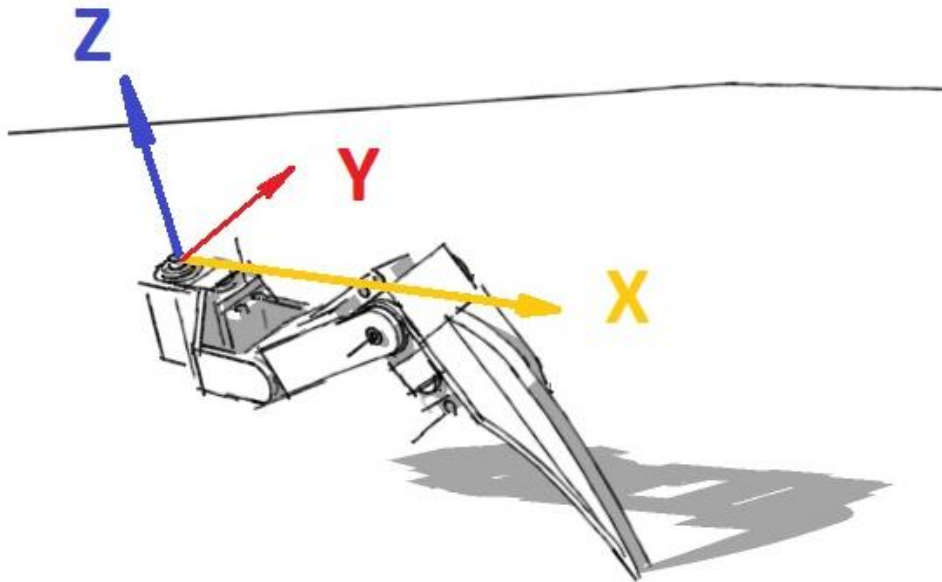
```
ben1 = amplitude*sin(w*t1 + q) + offsett;  
cout << "Grader Ben 1: " << ben1 << " Time: " << t1 << endl;
```

Hvor  $\omega$  er vinkelhastigheten omega oppgitt i rad/s,  $t_1$  er «timer»-en som er i ms og  $q$  er fasevinkelen.

### Kinematikk

### Forover kinematikk

For at ROV-en skal kunne bevege seg på sjøbunnen blir det brukt hexapod gangart som innebærer at den har 6 bein, tre på hver side av ROV-en. Disse beina er 3-linket armer med rotasjon i hvert ledd.



Figur 102. Robotbein.

(Liang, 2012)

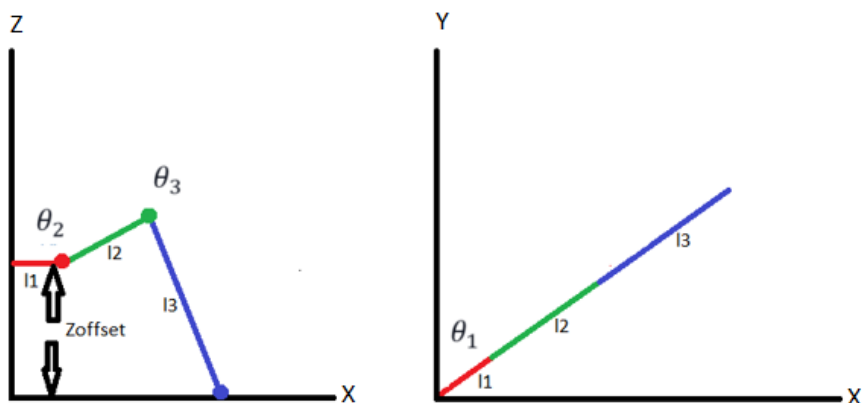
Endepunktet til armen er gitt av lengden til hver link og vinklene til leddene. Målet er å kunne sette et endepunkt sånn at beinet beveger seg til det ønskede punktet. Det første som må regnes ut er foroverkinematikken. Foroverkinematikken gir et endepunkt som en funksjon av vinklene  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ . Endepunktet til et bein på roboten er gitt av disse 3 likningene:

$$x = l_1 * \cos(\theta_1) + l_2 * \cos(\theta_1) * \cos(\theta_2) + l_3 * \cos(\theta_1) * \cos(\theta_2 + \theta_3) \quad (13)$$

$$y = l_1 * \sin(\theta_1) + l_2 * \sin(\theta_1) * \cos(\theta_2) + l_3 * \sin(\theta_1) * \cos(\theta_2 + \theta_3) \quad (14)$$

$$z = Z_{offset} * \cos(\theta_1) + l_2 * \cos(\theta_1) * \sin(\theta_2) + l_3 * \cos(\theta_1) * \sin(\theta_2 + \theta_3) \quad (15)$$

Hvor z-offsett er høyden fra bakken og opp til origo til armen,  $l_1$ ,  $l_2$  og  $l_3$  er lengdene mellom hvert ledd og  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$  er leddenes vinkler, som vist på figur 103:



Figur 103. Lengder og vinkler til ett bein.

### Invers kinematikk

I dette prosjektet blir den pseudoinverse Jacobian matrisen brukt for å finne vinklene som hvert ledd må ha for at endepunktet skal nå den gitte destinasjonen. Det er en tilnæringsmetode og den vil iterere helt til den er innenfor en gitt radius av det ønskede punktet. Nedenfor vises framgangsmåten for å finne leddvinklene:

Det er likning (6) som skal regnes ut gjentatte ganger til  $\vec{e}$  er innenfor den gitte grensen. Det første som skjer under utregningen er å finne en «error»  $\vec{e}$  som er en vektor som forteller hvor langt unna og i hvilken retning det ønskede punktet ligger i forhold til det nåværende punktet.

```
error = Pd.minus(forwardKinematic(theta1, theta2, theta3));
```

Hvor Pd (destination point) er det ønskede punktet og det nåværende punktet blir regnet ut ved hjelp av likningene (13) (14) (15).

Før den pseudoinverse matrisen blir kalkulert, må Jacobian matrisen først bli opprettet. Av likningene (13) (14) (15) finner vi jacobian matrisen ved å partiellderivere likningene inn i matrisen med hensyn på ledd vinklene  $\theta_1$ ,  $\theta_2$ ,  $\theta_3$ .

For enkelthetskyld skrives det noen forkortninger:

$$S1 = \sin(\theta_1), S2 = \sin(\theta_2), S23 = \sin(\theta_2 + \theta_3)$$

og

$$C1 = \cos(\theta_1), C2 = \cos(\theta_2), C23 = \cos(\theta_2 + \theta_3)$$

Jacobian matrisen blir da slik:

$$J(\theta) = \begin{bmatrix} -L1 * S1 - L2 * S1 * C1 - L3 * S1 * C23 & -L2 * C1 * S2 - L3 * C1 * S23 & L3 * C1 * S23 \\ L1 * C1 + L2 * C1 * C2 + L3 * C1 * C23 & -L2 * S1 * S2 - L3 * S1 * S23 & -L3 * S1 * S23 \\ -z0 * S1 - L2 * S1 * C2 - L3 * S1 * S23 & L2 * C1 * C2 + L3 * C1 * C23 & L2 * C1 * C23 \end{bmatrix} \quad (16)$$

Deretter blir den pseudoinverse Jacobian matrisen regnet ut ved hjelp av likning (). For å finne den transponerte ( $J^T$ ) og inverse ( $J^{-1}$ ) matrisen blir det brukt et Javabibliotek Jama-1.0.3.jar (math.nist.gov, 2012). Jama er et bibliotek med funksjoner som utfører grunnleggende utregninger av matriser. Likning (7) er vist under:

```
J_Pseudo = Jacobian.transpose().times((Jacobian.times(Jacobian.transpose())).inverse());
```

Vinklene blir funnet av likning (6) som multipliserer den pseudoinverse matrisen med forflytningen ( $\vec{e}$ ) av endepunktet. Resultatet er et array med de 3 nødvendige vinklene.

```
delta_Theta = J_Pseudo.times(error);
```

Siden dette er en tilnærming av de nødvendige vinklene som skal til for å nå et gitt ønsket punkt, skjer utregningen inn i en «while»-løkke som vist nedenfor:

```
while (error.norm2() > Math.pow(10, -6) & iterations <= 100)
```

Hvor norm2 er og Math.pow(10,-6) er  $10^{-6}$ . Der er også en grense på hvor mange itterasjoner while-løkken skal gjennomføre.

Det blir for hver utregning plussset på en delta\_Theta ( $\Delta\theta$ ) i hvert ledd for deretter å bruke foroverkinematikken for å finne endepunktet med de oppdaterte vinklene.

```
theta1 = theta1 + delta_Theta.get(0, 0);
theta2 = theta2 + delta_Theta.get(1, 0);
theta3 = theta3 + delta_Theta.get(2, 0);

error = Pd.minus(forwardKinematic(theta1, theta2, theta3));
```

Om det daværende punktet ikke er innenfor grensen som er gitt, vil «while»-løkken være «true» og den vil gjøre en ny utregning med de nye vinklene som parameter. Når «while»-løkken blir «false» vil det tilsi at endepunktet er innenfor den gitte grensen og programmet går videre.

### Rotasjonsmatrisen

Ett av målene for ROV-en var å kunne snu seg rundt stillestående for å inspisere et område mer nøyaktig. For å løse dette problemet ble det brukt rotasjonsmatriser som endrer endepunktet til beina. Med rotasjonsmatrisen kan dette punktet bli manipulert. Det er tre matriser (en for hver akse x,y,z), og disse matrisene blir multiplisert med hverandre for å danne en ny matrise. Den nye matrisen blir brukt for å endre endepunktet til beina. Matrisene er vist under:

```
Rx = new double[][]{{1, 0, 0}, {0, cos(pitch), -sin(pitch)}, {0, sin(pitch), cos(pitch)}};
Ry = new double[][]{{cos(roll), 0, sin(roll)}, {0, 1, 0}, {-sin(roll), 0, cos(roll)}};
Rz = new double[][]{{cos(yaw), -sin(yaw), 0}, {sin(yaw), cos(yaw), 0}, {0, 0, 1}};
```

Rotasjonene pitch, roll og yaw blir hentet fra XBOX-kontrolleren og sendt inn i matrisen «Rrot» som deretter blir multiplisert med det nåværende punktet P0. Et nytt punkt P1 vil nå være flyttet med hensyn på rotasjonene rundt x, y og z. Kinematikken bruker deretter dette punktet som nytt endepunkt (Liang, 2012).

### Visualisering av kinematikk i Matlab

Visualiseringen av kinematikken til beina ble gjort i Matlab. Matematikken ble gjort på samme måte som i Java med noen små endringer for å kunne visualisere et bein i et 3D-plot. Etter invers kinematikken er regnet ut og vinklene til hvert ledd er funnet, begynner plottingen. Siden det ikke finnes noen funksjon i Matlab for å sette en vinkel mellom to vektorer, blir det i stedet funnet posisjonen til hvert ledd. Det blir videre plottet en linje mellom punktene. Det ble satt opp to nye likninger som finner punktet til hvert ledd. Siden likningene (13) (14) (15) for endepunktet til hele robotbeinet allerede er funnet, omformuleres de likningene for å finne punktene til hvert ledd. Likningene er vist under:

```
%Point 1
P1x = x0+L1*cos(theta1);
P1y = y0+L1*sin(theta1);
P1z = z0;

%Point 2
P2x = L1*cos(theta1)+L2*cos(theta1)*cos(theta2);
P2y = L1*sin(theta1)+L2*sin(theta1)*cos(theta2);
P2z = z0*cos(theta1)+L2*cos(theta1)*sin(theta2);

% Point 3
P3x = Px_c;
P3y = Py_c;
P3z = Pz_c;
```



Siden punkt 3 er endepunktet til robotbeinet blir det satt til endepunktet som er regnet ut tidligere i scriptet. For å plote punktene og linjene blir det først satt opp en 3D-figur. Plotting av punkt skjer på denne måten:

```
plot3(x0,y0,z0,'r.','markersize',20)
plot3(P1x,P1y,z0,'r.','markersize',20)
plot3(P2x,P2y,P2z,'g.','markersize',20)
plot3(P3x,P3y,P3z,'b.','markersize',20)
```

Her blir posisjon, farge og markørstørrelse satt. Linjene blir plottet med line() funksjonen som Matlab har innlagt. Det blir for hver iterasjon plottet punkt og linjer for å kunne se hvordan beinet tilnærmer seg den ønskede posisjonen.

### Kommunikasjon med servokontrolleren

Kommunikasjonen mellom Java-applikasjonen og servokontrolleren er gjort med seriell kommunikasjon over USB. Mini maestro servokontrolleren har standard kommandoer som gjør at brukeren kan manipulere servoutgangene over seriell kommunikasjon. Leverandøren av kontrolleren har laget et eget bibliotek som er skreddersydd til kontrolleren, men dette biblioteket er kodet i C++ og C#. Siden kinematikken ble skrevet i Java, ble det derfor skrevet et program i Java som sender det samme kommandoene til kontrolleren som C++ programmet gjør. Kommunikasjonen blir skrevet ved hjelp av en eksempelkode (Skokan, Webområde for GitHub, 2014) og et serielt bibliotek som heter RXTX.jar (rxtx qbang.org, u.d.).

Det er satt opp totalt seks kommandoer i programmet: setTarget, setAcceleration, setSpeed, setMultipleTargets, getPosition, getMovingState.

Det første som skjer er at det blir opprette en port. Etter porten åpnes blir en rekke parametere satt. Parameterne som er satt er vist under:

```
serialPort.setSerialPortParams(115200, SerialPort.DATABITS_8, SerialPort.STOPBITS_1,
    SerialPort.PARITY_NONE);
```

Deretter blir en seriell «inputstream» og «outputstream» blir satt opp, hvor inputstreamen leser fra seriell porten og outputstreamen skriver til porten. Det kan nå bli sendt og mottatt informasjon fra servokontrolleren.

For å sette en posisjon til en servo må kommandoen (0x84) sendes til kontrolleren sammen med spesifisert kanal og posisjonen et byte-«array». Posisjonen er oppgitt i  $\mu\text{s}$  siden posisjonen til en servo er gitt i mikrosekund hvor  $450\mu\text{s}$  er 0 grader og  $2300\mu\text{s}$  er 180 grader. Disse verdiene kan variere da andre servoer kan gå fra 0 – 360 grader. Byte-arrayet blir satt opp slik:

```
double pos = position * skaleringsFaktor + POS_MIN;
int p = (int) pos * 8;
if (pos < POS_MIN || pos > POS_MAX) {
    throw new IllegalStateException("Wrong position " + servo + "/" + p);
}
ByteBuffer bb = ByteBuffer.allocate(4);
bb.putInt(p);
byte low = (byte) (bb.array()[3] & 0x7f);
byte high = (byte) (bb.array()[2] & 0x7f);
byte[] bbb = new byte[]{Command.SET_POSITION, (byte) servo, low, high};
```

Arrayet blir bygget opp med kommandoen først deretter servo kanalen og til slutt high og low byte.

Alle kommandoene blir satt opp i et byteArray. Hvordan hver byte-array blir satt opp for de ulike kommandoene er oppført i brukermanualen til Maestro-kontrolleren. (Pololu, u.d.)

## XBOX kontrolleren

Det blir brukt en XBOX-kontroller for å styre ROV-en. Det er XBOX-kontrollerklassen som tar seg av dette. Klassen henter ut informasjon fra kontrollen ved å bruke et bibliotek for joysticks(jinput.jar). Det ble brukt deler av et eksempelprogram som var en universal GUI for joysticks (Webområde for TheUZ0007, 2013).

Det første som skjer i koden er at det i legController klassen blir opprettet en instanse av klassen XBOX-kontroller som «extends Thread». I XBOX-kontrollerens konstruktør blir det søkt etter alle joysticks som er koblet til datamaskinen. Når den har funnet alle joystickene legger den til den joysticken som er av typen gamepad, som er XBOX-kontroller typen:

```
if (controller.getType() == Controller.Type.GAMEPAD) {
    foundController = controller;
}
```

Deretter startes tråden og funksjonen .poll() henter data fra kontrollen. Når dataene er mottatt undersøker programmet hvilken stikke eller bryter som blir operert på kontrollen.

Dersom den analoge stikken blir operert vil programmet motta verdier som går fra -1 til 1 i både y og x retning slik som på en enhets sirkel, hvor kontroller-stikken sitter i origo.

## Komprimering og dekomprimering

OpenCV gir ikke muligheten til å hente ut bilde uten at det blir dekomprimert. Å overføre et ukomprimert bilde over nettverket er ikke ideelt siden det belaster TCP protokollen og nettverket unødvendig mye. Det lønnet seg ikke å komprimere bildet igjen før sending. Derfor ble det undersøkt hvordan det er mulig å direkte hente ut komprimert bilde fra kamera. Alternativene som ble vurdert er Directshow, Xuggler, Gstreamer og FFMPEG.

Directshow er utviklet av Microsoft og er ikke kompatibelt med Linux plattformen. Directshow er godt dokumentert og kan være det beste alternativet til å hente ut videostrømming fra kamera dersom det brukes en Windows løsning.

Xuggler er et mye brukt bibliotek for komprimering og dekomprimering av bilder og videostrøm i Java. Xuggler er kompatibelt med både Windows og Linux, men kan ikke brukes sammen med C++ og ble derfor ikke lenger aktuell.

Gstreamer er kompatibelt med Linux, og det ble gjort forsøk på implementering av biblioteket. Men med vanskeligheter under installasjon og en API som ikke er tilfredsstillende ble Gstreamer valgt bort.

FFMPEG er kompatibelt med de nødvendige operativsystemene. Det ble gjort forsøk på installasjon for implementering av FFMPEG i serverapplikasjonen. FFMPEG er mye brukt som en kommandolinje verktøy og er ikke godt dokumentert for implementering til programvare.

## Server applikasjon

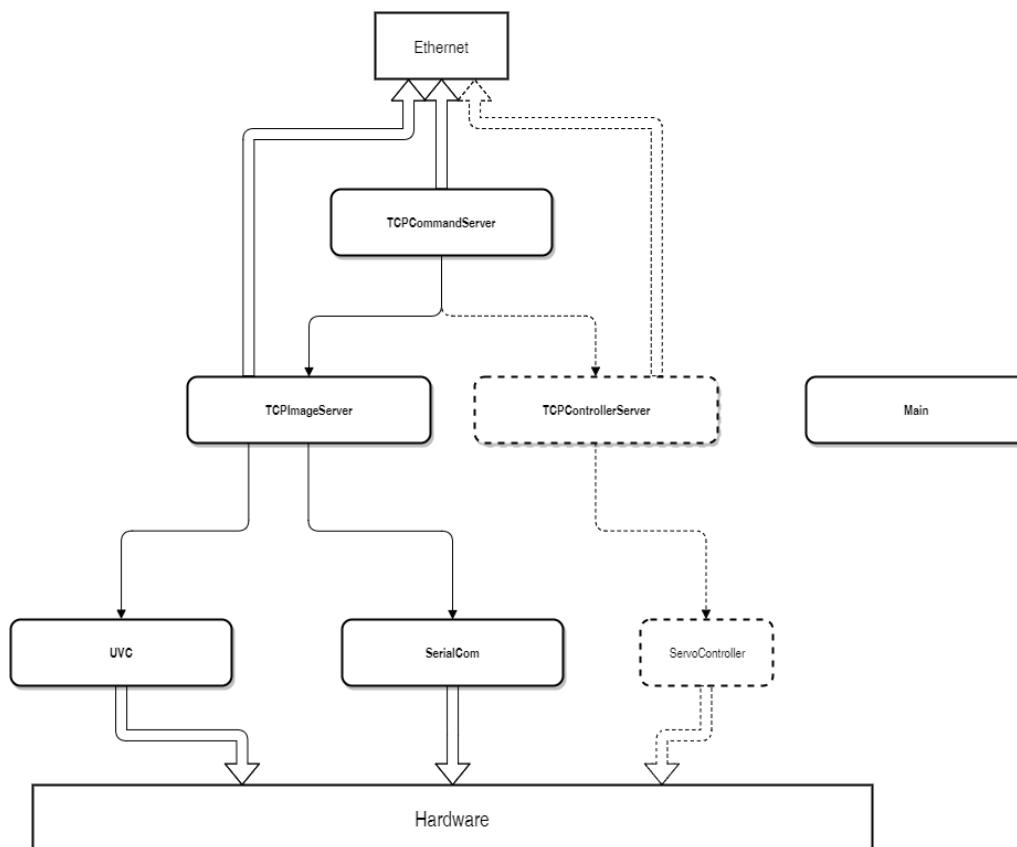
### *Plattform*

Til denne oppgaven ble det valgt at det skulle brukes en Odroid XU4. Kriteriene for valget er høy prosessorkraft og at single-board maskinen har gigabit nettverkskort. XU4 har noe høyere prosessorkraft i forhold til konkurrerende single-board datamaskin og ingen av de andre aktuelle kandidatene hadde gigabit nettverkskort. Kortet vi fikk låne av NTNU i Ålesund hadde allerede blitt installert med Lubuntu som er en lettere utgave av Ubuntu.

Dette er ideelt for denne oppgaven siden det ikke er behov for en fullversjon av Ubuntu for å kjøre den nødvendige programvaren. Med dette oppsettet kjører det minimalt med programmer under bruk. Dette gjør at applikasjonen som skal drive ROV vil ha mer ressurser tilgjengelig.

### *Klasser*

Ansvarsfordelingen til klassene i server og klient er basert på hvilke oppgaver de skal utføre. Server applikasjonen består av fem klasser. Main, UVC, SerialCom, TCPImageServer og TCPCommandServer. Under utvikling av serverapplikasjonen ble det startet med klasser som skal kommunisere med hardware ombord i ROV. Disse klassene skal ikke styre klasser oppover i hierarkiet, og skal bare holde på den nyeste informasjonen fra hardware. Klasser som er høyere oppe i hierarkiet har mulighet til å gjøre forandringer nedover, ikke oppover. Her er en oversikt over hierarkiet, TCPCommandServer er den eneste klassen som har mulighet til å foreta forandringer som kommer fra klientapplikasjonen. Objekter og linjer som er stiplet er ikke implementert i hovedprogrammet. Dette er ikke et klassediagram (neste side).



Figur 104. Klassenes hierarki.

Main klassens ansvar og arbeidsoppgave er å opprette de nødvendige objektene som trengs i applikasjonen, i tillegg til å opprette og starte trådene. Det er funksjoner i objektene TCPImageServer og TCPCommandClient som skal kjøres i tråd. Main skal også sette konfigurasjoner til trådene, og starte de objektene som skal brukes av trådene.

UVC klassen sitt ansvar er å hente ut bilde. Det ble opprettet en klasse med samme navn som protokollen for at det skal være mulig å lage andre klasser som tar i bruk andre protokoller for å hente inn bilder, som også kan hete det samme som protokollen. UVC fungerer som et mellomledd mellom kamera og TCPImageServer klassen. UVC klassen skal hente ut og holde på nødvendig informasjon om kameraet, dette gjelder hovedsakelig tilgjengelige oppløsninger. På denne måten belastes ikke kamera unødig med å sende samme informasjon flere ganger.

SerialCom klassen sitt ansvar er å opprette en seriell kommunikasjon mellom sensoravlesningen og hente ut sensordata. SerialCom klassen konfigurerer en seriell port på Odroiden og setter hvordan innkommende data skal tolkes, siden det bare skal leses og ikke skrives konfigureres det bare for innkommende data. Denne klassen leser bare på seriell port når det forekommer et funksjonskall fra TCPImageServer objektet.

TCPImageServer klassen sitt ansvar er å hente ut bilde fra UVC klassen og sende bildet over Ethernet til kontrollapplikasjon på land. Det sendes også med informasjon fra ROV-en som har med bildet å gjøre, sensordata fra gyro og gjeldende oppløsning sendes sammen med bildet. TCPImageServer kan endre egenskaper ved bildet ved at det overordnede objektet TCPCommandServer kaller på funksjoner i klassen.

TCPCommandServer klassen sitt ansvar er å endre på de tilgjengelige egenskapene i server applikasjonen. TCPCommandServer venter på kommando fra klient applikasjon som er landbasert.

Kommandoer som er implementert er omstart av nettverkskommunikasjon for TCPIImageServer og TCPCommandServer og endring av oppløsning. Etter et objekt av TCPCommandServer er opprettet skal denne ta på seg rollen som det overordnede objektet, dette objektet skal kunne ha muligheten til å styre alle egenskaper i server applikasjonen.

### *Bilde*

Bilder hentes ut ved hjelp av OpenCV. TCPIImageServer bruker et funksjonskall til UVC klassen til en funksjon som heter retrieveFrame(Mat &mat). denne funksjonen tar inn en minneplassering til et MAT objekt som UVC bruker til å legge inn det nyeste bildet tilgjengelig. Like før dette gjøres hentes det først inn den nyeste gyrodataen, like etter kalles grabNewFrame som lagrer det nyeste bildet. Dette gjør at det blir minimalt med forsinkelse mellom bildet som skal hentes ut og gyrodataen som hører til bildet. «Grab» og «retrieve» kan byttes ut med en felles funksjon som heter read, read metoden henter ut bilde fra kamera og dekoder bildet. Grab plasserer det enkoda bildet i en buffer klar for dekoding, retrieve er en mer tidkrevende metode som dekoder bildet. Ved bruk av denne kombinasjonen av grab og retrieve er vi sikrere på at gyrodata og bildet som skal stabiliseres hører sammen.

### *Sensor*

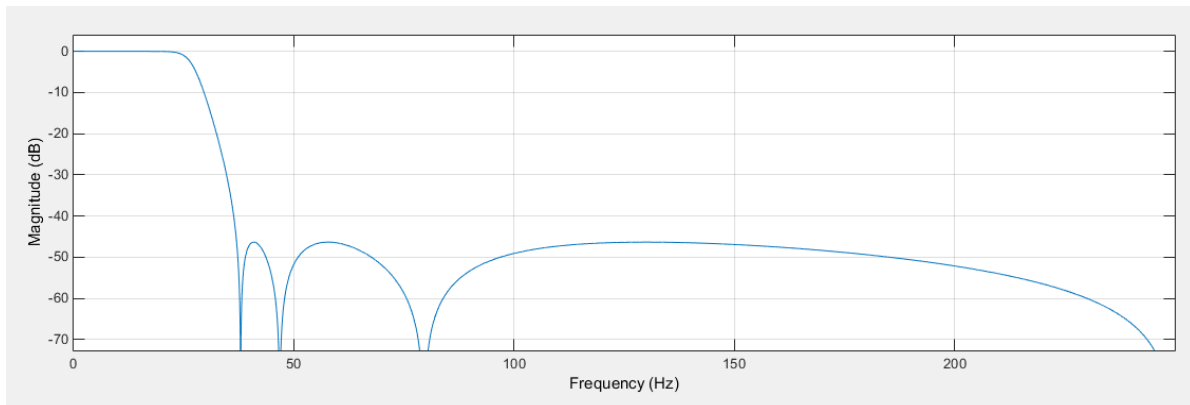
Sensoren som brukes er Adafruit 10-DOF som har muligheter til å hente ut 3-akset gyroskop, 3-akset akselerometer, 3-akset kompass, temperatur og trykk.

Verdiene som skal brukes i prosjektet er roll fra gyroskopet og temperatur, trykksensoren kan ikke brukes siden denne må være i kontakt med sjø.

Sensoren var først koblet til en Teensy 3.2. Denne har flere I/O iforhold til R3 og har generelt sett bedre spesifikasjoner. Teensyen ble koblet til Odroid via USB og blir oppdaget av operativsystemet, men når det skulle overføres data var det ikke mulig å lese fra seriellporten. Den samme koden og konfigurasjonen ble brukt på en Arduino R3 som det er mulig å lese data fra, men ikke feilfritt. Årsaken til dette problemet ligger trolig i at Arduinoen må kobles til UART via en konverter for at det skal fungere uten problemer. Slik det er nå kommer det mye feildata som ødelegger for bildestabiliseringen. Når dette kobles opp mot en Windows maskin oppstår ikke dette problemet.

Adafruit 10-DOF er nå koblet opp mot en Arduino R3. Informasjonen sendes serielt fra Arduinoen via USB type-2 kabel. Forsendelsen skjer hvert 15ms som er over det dobbelte av den høyeste bildefrekvensen systemet klarer å oppnå. I programvaren til Arduinoen er det implementert et Infinite Impulse Response Chebyshev type-2 filter for å eliminere støy fra sensoren. Filteret som brukes er av typen lavpass og påvirker minimalt frekvenser lavere enn passfrekvensen.

Passfrekvensen ligger på 25Hz, stoppfrekvensen på 37Hz og en dempning på 45dB. Filteret ble utviklet i Matlab med verktøyet «Filter Design And Analysis Tool». Det ble logget data fra gyro som ble kjørt gjennom en fast Fourier transformasjon for å finne frekvensene. Ut ifra hvor støyen lå i frekvens spekteret ble passfrekvensen også plassert. Stopp frekvensen ble plassert for å få et filter med en håndterbar orden. Filteret har en orden på 7.



Figur 105. Filterets frekvensrespons.

### Nettverk

Den mest aktuelle kommunikasjonsformen er overføring via Ethernet, dette er for kompatibilitet mot de forskjellige plattformene og har høy overføringshastighet.

Ethernet er oppsettet som brukes for å kommunisere mellom klientapplikasjonen og serverapplikasjonen. Protokollen som brukes er TCP, dette er fordi den har mulighet til å segmentere pakker større enn MTU automatisk, UDP har ikke mulighet til å segmentere pakker som gjør den ugunstig til å sende større pakker.

Odroid som er ombord i ROV har rollen som server. Server-rollen ble tildelt Odroid fordi det ikke skal være nødvendig å låse IP-adressen hos klientmaskinen. Da kan det i klient applikasjonen låses adressen til Odroid og sikre at det kan tilkobles hver gang så lenge Odroid og klient maskin er på samme nettverk. Odroid er konfigurert slik at den kan ta imot en hvilken som helst IP-adresse fra klient. Når serverapplikasjonen starter ligger den å lytter på to porter, 6000 er porten som brukes til kommando kommunikasjon, 6001 er porten som brukes til bilde kommunikasjon.

### Strømforsyning og kommunikasjon

For å forsyne ROV-en med kraft ble både vekselstrøm og likestrøm vurdert. Det ble også vurdert å bruke flere forskjellige forsyningsspenninger fra land. Avgjørende kriterier for valg av løsning var operasjonstid, vekt, plassforbruk, pris, effektleveranse og enklest mulig nettverksløsning.

Ved bruk av høy forsyningsspennning vil spenningsfallet i forsyningskabelen bli lavt, noe som gir mulighet for å bruke lavere tverrsnitt på lederne i kabelen. Men det vil bli en større differanse mellom inn-spennning og ønskede spenninger til ROV-ens komponenter. Noe som kan gjøre det mer utfordrende å finne egnede omformere. Høy spenning medfører også større fare for strømgjennomgang ved berøring.

Ved bruk av lav forsyningspenning vil spenningsfall bli et større problem, og vil føre til at det må brukes større tverrsnitt på lederne i kabelen. Men det kan bli enklere å finne egnede omformere.

For å forsyne kabelen har det blitt gått ut ifra at forsyningen vil komme fra et strømaggregat, et uttak på en båt, eller fra batterier.

### Effektforbruk ROV

En nøyaktig estimering av ROV-en har ikke latt seg gjøre. Dette fordi enkelte komponenter ikke har blitt fastslått, og fordi produsentene av enkelte komponenter har ikke oppgitt effektforbruk.

Følgende tabell viser estimert maksimalt forbruk i ROV-ens tre forskjellige tilstander:

Tabell 18. Maksimalt effektforbruk.

	Vandre	Svømme	Vandre <-> Svømme
Thrusterer	0 W	405 W	135 W
Servoer	Estimert 100 W	0 W	Estimert 50 W
Odroid	20 W		
Kamera og mikrokontroller	17,5 W		
Lys	Ikke bestemt, men vil nok ikke overstige 50 W		
Sum	137,5 W	492,5 W	272,5 W

### Vekselstrøm

En ordinær stikkontakt leverer 230V vekselstrøm. Siden slike uttak er meget utbredt ble 230V utgangspunkt for en potensiell vekselstrømsløsning.

Til vekselstrømsløsning ble det valgt en kabel med polyuretan beskyttelse som er mye brukt i oppdrettsnæringen. Kabelen som ble valgt er Roflex PUR, denne kabelen er svært bevegelig under et stort temperaturområde og tåler mye påkjenning. Det negative med denne kabelen er at den ikke har metallkappe mot mekanisk påkjenning, men polyuretan kappen har lav friksjonskoeffisient som gjør at den ikke hefter seg så lett på skarpe gjenstander. Ved bruk av så høy spenning ville det vært nok å bruke  $1\text{mm}^2$  kabel.

### Nettverk

Det gjort test på overføringshastigheten til Zyxel HOMEPLUG AV2 2000, som overfører Ethernet via strømnettet. Denne enheten gjør at det blir mulig å bruke tre ledere ned til ROV, to faseledere og jord. Det ble utført en hastighets test både på skolens strømnett og på aggregatet. Når det ble testet på skolen sitt strømnett ble den høyeste hastigheten målt til omtrent 800Mbit/s. Når powerline Ethernet ble koblet til aggregatet og den samme testen utført ble det målt en hastighet på maks 85Mbit/s. Denne testen ble utført med og uten 1000W last. Når aggregatet var ubelastet ble hastigheten målt til 85Mbit/s, med last ble hastigheten målt til 81Mbit/s. Aggregatet som testen ble utført på hadde inverter som lager støy på strømnettet. Etter en samtale med produsent ble det avgjort at dette trolig er en stor faktor for hvor høy overføringshastighet det er mulig å hente ut fra powerline Ethernet.

Det ble valgt å gå bort fra powerline Ethernet og 230V tilførsel etter det ble målt for lave overføringshastigheter.

### Landforsyning

Basert på de nevnte faktorene ble det bestemt at det først skulle undersøkes nærmere om en løsning med fasttilkobling er egnet. De to største problemstillingene med fasttilkobling var: hvordan riktige spenninger i ROV-en skal oppnås, og hvilken forsyningsspenning fra land som skal brukes. Samtidig som tverrsnittet på kablen ikke blir for stort.

For å oppnå riktige spenninger i ROV-en ble flere alternativer vurdert. Regulering med zenerdiode var den første løsning som ble vurdert. Dette fordi det er en enkel, og relativt billig måte å regulere spenning på. Men siden en spenningsregulator basert på zenerdiode regulerer spenningen ved å dumpe effekt i form av varme vil det føre til et stort effekttap og ugunstig varmeutvikling. Det neste alternativet som ble undersøkt var "Switched-mode" strømforsyninger. Som ledet videre til "Buck converter", en type switched-mode strømforsyning som kun konverterer spenningen nedover.

Neste steg var bestemme forsyningsspenning. Ved spenninger lavere enn 60V likespenning er beskyttelse mot berøring vanligvis ikke nødvendig (Norsk Elektroteknisk komité, 2002). Derfor er det ønskelig å benytte en spenning på under 60V. Den første forsyningsspenningen som ble vurdert var 30V, basert på følgende tabell:

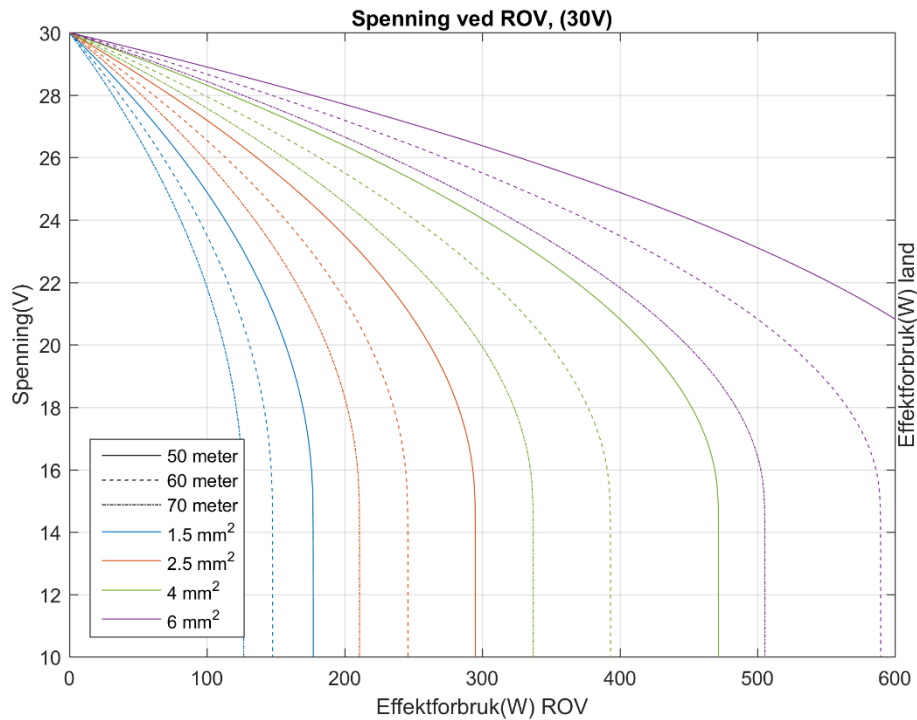
### Buck konvertere for 30V løsning

Tabell 19. Oversikt over Buck konvertere for 30V løsning.

Oversikt Buck konvertere (30V løsning)						
Antall	Ut(V)	Strøm(A)	Effekt(W)	Effektivitet(%)	Inn min.(V)	Inn maks.(V)
2	13,8	20	276	>90	20	30
1	12	5	60	< 96	17	35
1	7	15	105	>93	9	35
1	5	5	25	>90	9	30
Felles minste og maksimale inntakspenning →					20	30

Tabellen viser Buck konvertere med forskjellige spenninger og effekter med kompatible inntakspenninger. Buck konvertere er funnet basert på hvilke spenninger og effekter ROV-ens forskjellige komponenter trenger. Ut ifra tabellen kan man se at ved bruk av disse Buck konverterne vil man kunne bruke en forsyningsspenning på 30 Volt med et spenningsfall i kablen på maksimalt 10 Volt. I grafen under kan man se teoretisk spenningsfall ved forskjellige kabellengder og effektforbruk av ROV-en. Effektiviteten til ROV-en er satt til 90%, basert på Buck konverteren med størst effekt.





Figur 106. Spenning ved ROV, 30V forsyningspenning

Grafen viser at ROV-en kunne ha fungert med en kabel på 50 meter med et tverrsnitt på  $2.5\text{mm}^2$ . Den ville da kunne bruke omtrent 260 Watt. Som er mindre enn estimert forbruk i to av ROV-ens tilstander. ROV-en kunne muligens klart seg med den begrensede effekten ved å legge inn elektroniske begrensninger i programvaren. Men det vil medføre at ROV-en får utnyttet sine komponenter skikkelig, og føre til redusert ytelse.

Med en kabel på 50 meter med et tverrsnitt på  $4\text{mm}^2$ , og en tilgjengelig effekt på omtrent 418 Watt, ville nok ROV-en kunne operere ganske nært sitt fulle potensiale. Men en kabel på  $4\text{mm}^2$  er relativt tung, og straks kabellengden forlenges med få meter, øker spenningsfallet betydelig. Med andre ord vil ikke kabellengden kunne økes uten å enten gå ytterligere opp i tverrsnitt eller redusere ROV-ens ytelse.

På bakgrunn av de nevnte problemstillingene ble det besluttet å utbedre en bedre løsning på kraftfordelingen i ROV-en. Med blant annet Buck konvertere som kan ta inn høyere spenninger. Likespenningen 48V ble satt som nytt mål for landspenning. Spenningen er mye brukt, og spenningsfallberegninger viser at spenningsfall blir et betydelig mindre problem.

### Nettverk

I kombinasjon med likestrøm har det blitt vurdert å legge ved en egen kommunikasjonskabel som går ned til ROV-en sammen med spenningsstilførsel. Det har da blitt vurdert fiberkabel og vanlig 8-leder tvinnet parkabel.

Fiberoptisk kabel er et godt alternativ som takler store mengder data alt etter hvilken medieomformer som brukes. Bruken av fiber er et alternativ som ikke er begrenset av lengde aktuell for oppgaven og vil være noe lettere enn en nettverkskabel. Negative med fiberoptisk kabel er at det må brukes to mediekonvertere, en på land og en på ROV. Disse er også noe dyr i tillegg til at de vil ta mer plass i ROV. Det ble valgt å ikke bruke en fiberløsning i dette prosjektet fordi dette kan implementeres i det eksisterende systemet med en beregning på plassbruk og strømforbruk. Det vil ikke ha en innvirkning programmatisk.

Tvinnet 8-leder parkabel er ikke det ideelle alternativet, men siden ROV kabelen ikke skal overstige 100m, men heller ligge på rundt 50meter er nettverkskabel et godt alternativ som takler overføringshastigheten. Dette er den kabelen som er mest benyttet i sammenheng med Ethernet og er kompatibelt med all hardware som skal på Ethernet i denne oppgaven.

### Beregning av spenningsfall

Spenningsfallberegningene ble gjort både inkrementelt og kontinuerlig. Et Matlab skript ble skrevet for å regne ut og visualisere beregningene. Det ble først gjort inkrementelle beregninger, men for å forsikre at de var korrekte ble det utledet en kontinuerlig funksjon.

Den inkrementelle spenningsfallberegningen er gjort ved å inkrementere effektforbruket til ROV-en med en gitt steglengde. Nøyaktigheten til resultatet avhenger av steglengden, mindre steglengde gir økt nøyaktighet.

Spenningsfallet for hvert steg utregnes på følgende måte:

- Strømmen regnes ut ved å dele ROV-effekten på spenningen ved ROV-en fra forrige steg.
- Spenningsfallet blir beregnet basert på strømmen.
- Spenningen ved ROV-en blir regnet ut ved å trekke spenningsfallet fra forsyningsspenningen.
- Ny strøm regnes ut ved å dele ROV-effekten på spenningen ved ROV-en.
- Effektforbruk på land regnes ut ved å multiplisere strømmen med forsyningsspenningen.
- Deretter starter løkken på nytt og ROV-effekten inkrementeres ett steg.

For kontinuerlig beregning av spenning ved ROV ble følgende formel utledet:

$$U_{ROV} = \frac{1}{2} \left( U + \sqrt{U^2 - 4P_{ROV}R_{kabel}} \right), \quad 0 < P_{ROV} < \frac{U^2}{4R_{kabel}}, \quad U_{ROV} > \frac{U}{2}$$

Den kontinuerlige beregningen fungerer kun så lenge spenningsfallet ikke overstiger halvparten av forsyningsspenningen. Dette gjelder ikke for den inkrementelle metoden. Derfor har grafer fra den inkrementelle metoden blitt brukt til visualisering i rapporten. Den gir noe avvik fra den kontinuerlige metoden, men med en steglengde på 0.01W blir avviket ansett som neglisjerbart.

Se vedlegg 6 for utledning av funksjonen for spenning ved ROV-en og Matlabkoden.

Eksempelkoder er fra følgende kilder:

- Windows socketprogrammering (Winsock, u.d.)
- Windows trådprogrammering
- Linux socket programering (Socket, u.d.)
- Linux Seriellkommunikasjon (programming, u.d.)
- Linux trådprogrammering (Provenzano, u.d.)
- Kamerakalibrering (Matlab, u.d.)
- Stabilisering med bildebehandling (nghiaho, 2014)
- Qt dokumentasjon (QT Co., u.d.) (GitHub, u.d.)
- OpenCV dokumentasjon (OpenCV, 2015)
- Kinematikk (Cao, 2015)

### 3.2.4 Resultat og diskusjon

#### Kamera

I største delen av utviklingen ble det brukt et Logitech C920 webkamera. Det hadde godt bilde, men ikke spesielt bred synsvinkel. Det ble derfor kjøpt inn et ELP USB500W02M-L170, for å kunne teste hvordan rotasjon- og linsekorrigerer fungerer på et kamera med 170° synsvinkel. Begge kameraene brukte USV standarden så det var kompatibelt med eksisterende programvare uten modifikasjoner.

Det ble tatt noen ytelsesmålinger mot kamera via FPS målingen fra GUI kalkulasjonen. Dersom det er mye blandede farger måles det 13-14 FPS. Dersom det er mye hvitt i bildet, måles det 16-17 FPS. Disse målingene er et gjennomsnitt av FPS-en til de 25 siste bildene. Det mistenkes at hardware enkodingen til ELP kameraet er årsaken siden det yter bedre med mye hvitt i bildet. Dette var ikke tilfellet når Logitech C920 ble brukt. Oppløsningen denne målingen ble utført på er 640x480. Se vedlagt video for demonstrasjon av FPS.

#### Valg av språk

I et tidlig stadiet i prosjektet ble det utført en ytelsestest mellom Java og C++. Resultatet av denne testen ga en indikasjon på hvor hurtig et bilde kan hentes ut fra kamera, bli dekodet og deretter sendt over Ethernet med to forskjellige programmeringsspråk med tilsvarende kode. I begge tilfeller ble det sendt til en klient som ble skrevet i C++. Når det sendes en strøm med bilder i Java ble det avlest tid mellom hvert bilde på 140-160ms, dette tilsvarer mellom 6 og 7 bilder i sekundet. Når den samme testen med lik bildestørrelse ble utført i C++ ble det avlest en tid mellom hvert bilde på 47ms og 60ms, som tilsvarer mellom 21 og 15 bilder i sekundet. Dette resultatet ligger til grunn for valget om å programmere videre i C++ på både server applikasjon og klient applikasjon. OpenCV er et bibliotek som er utviklet for C++ (itseez, u.d.).

#### Server

Serverapplikasjonen er bygget opp til å kunne implementere nye funksjoner og ny hardware. Valget om å bruke Odroid XU4 baserer seg på tilgjengelighet og ytelse. XU4 er tilgjengelig på skolen og er kraftig nok til å drive serverapplikasjonen. Når det kobles strøm på XU4 starter den automatisk opp, dette tilsvarer hvordan ROV skal operere. Det skal kobles til strøm, etter maskinen har startet opp kan en kontrollstasjon koble seg til ROV. Serverapplikasjonen er ikke lagt til i listen over programmer som skal startes opp, hovedsakelig på grunn av ingen i gruppen har god nok kunnskap om C++ og Linux til å utføre det.

Det har vært en utfordring å bruke Linux som plattform, det har gjort at det må brukes forskjellige biblioteker mellom klientapplikasjonen og serverapplikasjonen. I ettertid har det blitt vurdert å bruke Raspberry pi 3 som kan leveres med Windows IOT. Med dette oppsettet kan serverapplikasjon og klientapplikasjonen kjøre med de samme bibliotekene. Da kan også det hardware enkodete bildet enklere hentes ut fra kamera uten dekoding og sendes direkte over nettverket. Dette vil gjøre at det blir mindre data som server kontrolleren må håndtere som kan gi raskere oppdatering av bildet.

#### Virkemåte

Det første serverapplikasjonen gjør er å opprette kommunikasjon med sensor og kamera, deretter opprettes TCP portene 6000 og 6001. Port 6000 brukes til å overføre kommandoer og har generelt lite trafikk, men fikk sin egen port for å enkelt skille TCP pakkene og fordi den skal ha mulighet til å starte kommunikasjon på nytt med andre porter. Når funksjonen Open() kalles på i UVC objektet

henter den inn alle relevante oppløsninger som kameraet støtter, dette lagres i et array som brukes av andre klasser. Standard innstillingen til oppløsningen er 640x480, men kan endres via TCPCommandServer.

Når kommunikasjonen er opprettet skal det komme en kommando i TCPCommandServer som ber om listen over tilgjengelige oppløsninger. Når denne listen sendes så sendes også den samme kommandoen som kom inn slik at begge parter vet hvilken informasjon som utveksles. Etter svaret er sendt vil TCPCommandServer ligge å vente på ny kommando. Når svaret er mottatt av kontroll/klient applikasjonen vil den be om tilkobling til TCPImageServer som vil starte videooverføringen. Etter den har startet er det mulig å endre oppløsningen.

TCPCommandServer er trådbasert og lytter til enhver tid på porten 6000. Når det kommer en kommando fra klient, har serveren en liste med ENUM som innkommende kommando blir sammenlignet med. Ut ifra hvilken kommando som kommer inn utføres det forskjellige arbeid. Dersom det kommer inn en kommando som påvirker et annet objekt skal den først låse mutex som sikrer at det ikke utføres en jobb fra to forskjellige objekt.

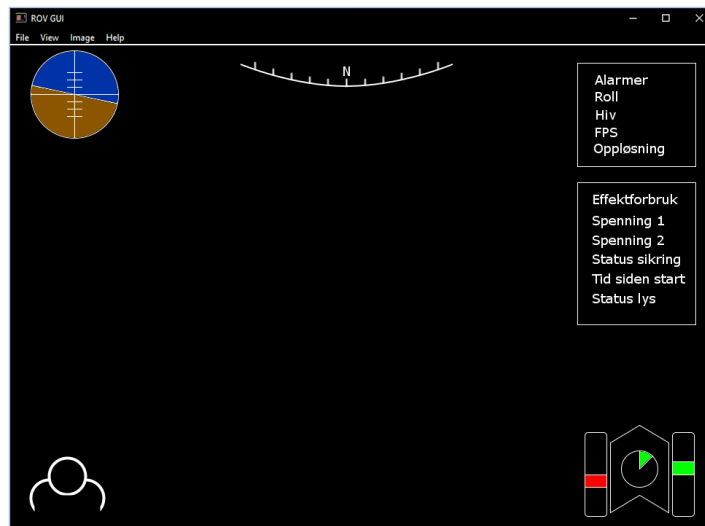
TCPImageServer bruker porten 6001 for å sende bilde. Først sendes det en pakke som inneholder gyrodata, deretter sendes to pakker som inneholder oppløsningen til bildet, til sist sendes bildet. Denne prosessen som sender bilde er trådbasert og blir kjørt så ofte som overhode mulig. Eneste avbrudd som kan forekomme er dersom kommandoserveren avbryter ved bruk av mutex, eller om det blir kommunikasjonsbrudd med kamera eller Ethernet.

Dersom det oppstår kommunikasjons svikt er det bare to måter å ordne problemet på. Det kan gjøres et forsøk på å starte kommunikasjonen på nytt via applikasjonen. Dersom det ikke ordner problemet må applikasjonen på Odroid restartes. Det kan gjøres med å koble fra strømtilførselen. Må applikasjonen startes på nytt er det et driverproblem som har oppstått og kan ikke ordnes uten å ha skrivebords tilgang på Odroid. Derfor er det lettere for brukeren å resette hele Odroiden som også ordner problemet. Driver problemet kan kjennes igjen ved at det ikke kommer bilde fra ROV, eller at gyrodataen ikke vises. Automatisk oppstart av applikasjonen er ikke implementert.

### **Kontrollapplikasjon konsept**

Den viktigste funksjonaliteten i kontrollapplikasjonen skal kunne styres fra en Xbox One håndkontroller. I tillegg skal håndkontrolleren brukes til å manøvrere ROV-en. All funksjonalitet i kontrollapplikasjonen skal kunne styres med mus og tastatur.

*Konseptbilde:*



Figur 107. Kontrollstasjonskonsept.

*Menyer:*

**File:**

- Lagre skjermdump med både originalt bilde, linsekorrigert bilde og data. Med hurtigtaster for håndkontroller og tastatur.
- Exit.

**View:**

- Muligheter for å skru av og på skjermbilde-elementer. Med hurtigtaster for håndkontroller og tastatur.

**Image:**

- Veksle mellom kamera foran og bak. Med hurtigtast for håndkontroller og tastatur.
- Rotasjonsstabilisering av bilde ved hjelp av gyro, av/på. Med hurtigtast for håndkontroller og tastatur.
- Dynamisk reskalering av bilde etter vindu, av/på. Med hurtigtast for tastatur.
- Reskalering av vindu til bildets oppløsning, trykk. Med hurtigtast for tastatur.
- Dialog for egendefinert oppløsning.
- En liste over automatisk definerte støttede oppløsninger.

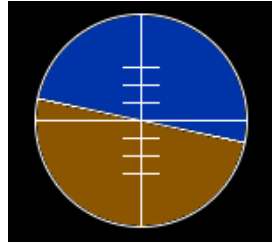
**Help:**

- Bruksanvisning for ROV-en og håndkontrolleren med hurtigtast for tastatur.
- Om programvaren

**Skjermbilde-elementer:**

*Kunstig horisont:*

Dette elementet er en etterligning av «kunstig horisont» instrumentet man finner i fly. Det indikerer «pitch» og «roll».



Figur 108. Kunstig horisont.

*Kompass:*

Et roterende kompass som indikerer retning.



Figur 109. Kompass.

*Data og alarmer/advarsler:*

Opplisting av relevant data og alarmer/advarsler. Hvis det oppstår en alvorlig feil vil en dialog sprette opp.



Figur 110. Diverse data.

*Avansert data:*

Viser avansert data som effektforbruk, spenninger, status på sikringer, tid siden ROV-en ble skrudd på og om lysene er skrudd på.



Figur 111. Avansert data.



*Beinindikator:*

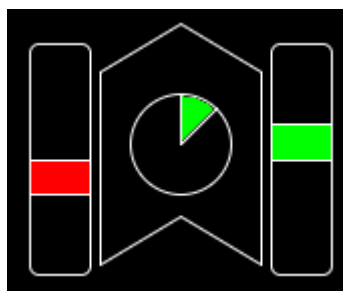
Indikerer om beinene er sammenslått eller ikke.



Figur 112. Indikator til flymodus.

*Thrusterindikator:*

De vertikale firkantene representerer de horisontale thrusterne. Senter av firkanten er nullpunkt. Grønn søyle oppover representer pådrag (i prosent) fremover. Rød søyle representerer pådrag bakover. Sirkelen i senter representerer den vertikale thrusteren. Pådrag oppover illustreres ved at sirkelen farges grønn med klokken. Pådrag nedover illustreres ved at sirkelen farges rød mot klokken. Den pil-lignende figuren i midten representerer retningen av skroget. Den skal erstattes med et omriss av det faktiske skroget.



Figur 113. Thrusterindikator.

*Videostrøm:*

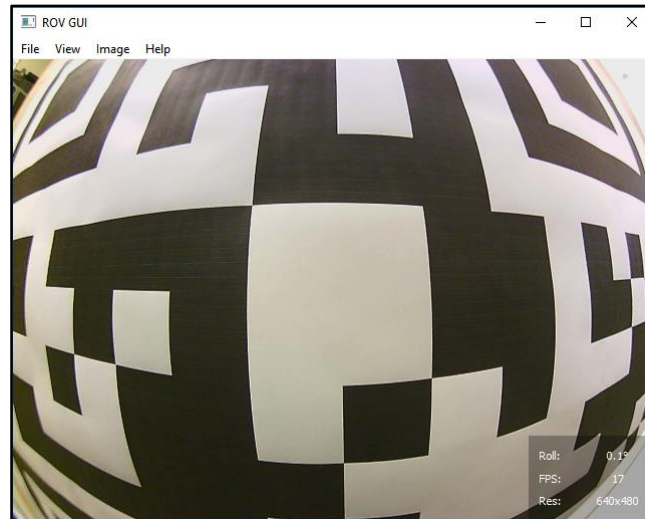
Videostrømmen skal være i senter og under skjermbilde-elementene, som blir delvis gjennomsiktig. Med hurtigtast på håndkontroller og tastatur for å veksle mellom kameraer (og lys).

**Kontrollapplikasjon***Plattform*

Kontrollapplikasjonen er laget for å kjøre på en PC med Windows operativsystem. Den er skrevet i C++ med Qt rammeverket.

### Hovedvindu

I kontrollapplikasjonens hovedvindu er all implementert funksjonalitet tilgjengelig gjennom de fire menyelementene: File, View, Image og Help.



Figur 114. Skjermbilde av kontrollapplikasjon.

Følgende funksjonalitet er implementert:

- **File:**
  - Lagre skjermdump med originalt bilde og linsekorrigert bilde. Tidspunkt lagres i filnavnet med snarveistast "S".
  - Lukking av programmet.
- **View:**
  - Data vindu med roll, FPS og oppløsning.
- **Image:**
  - Rotasjonsstabilisering.
  - Dynamisk reskalering.
  - Reskalering av vindu til bildets oppløsning.
  - Dialog for egendefinert oppløsning.
  - En liste over automatisk definerte støttede oppløsninger.

Help menyen er kun en plassholder. Styring med håndkontroller er ikke implementert.

## Klasser

Det er totalt åtte klasser i kontrollapplikasjonen: *main*, *MainWindow*, *TCPCommandClient*, *TCPImageClient*, *DataWindow*, *LensCorrection*, *SnapshotHandler* og *ImageStabilization*. *MainWindow* klassen er knyttet til alle de andre klassene i applikasjonen unntatt *LensCorrection*. Og all informasjonsflyt mellom deres objekter går via *MainWindow* objektet.

*Main* klassen oppretter et *QApplication* objekt og et *MainWindow* objekt. Deretter blir det gjort metodekall på *MainWindow* objektet som får hovedvinduet til applikasjonen til å bli synlig på skjermen.

*MainWindow* klassen sin hovedoppgave er å bygge opp og tegne hovedvinduet til applikasjonen. I tillegg skal den lese interaksjon fra brukeren og handle deretter.

*TCPCommandClient* klassen sin oppgave er å sende kommandoer fra *MainWindow* videre ned til kommandoserveren i ROV-en. For å definere kommandoer mellom server og klient er det brukt enumer. Kommunikasjonen foregår i en egen tråd.

*TCPImageClient* klassen sin oppgave er å motta bilder og gyrodata kontinuerlig fra bildeserveren og sende de videre til *MainWindow*. Kommunikasjonen foregår i en egen tråd.

*DataWindow* klassen brukes til å lage et gjennomsiktig vindu som viser informasjon om ROV-en og videostrømmen.

*LensCorrection* klassen sin oppgave er å kansellere ut forvrengningen i bildet som følge av linsen til kameraet.

*SnapshotHandler* klassen sin oppgave er å lagre et originalt bilde samt et linsekorrigert bilde. Linsekorrigeringen og lagringen foregår i en egen tråd.

*ImageStabilization* klassen sin oppgave er å rotere bildet slik at horisonten på bildet blir horisontal.

## Virkemåte

Det første som skjer når man starter applikasjonen er at alle nødvendige variabler og objekter deklarerer. Deretter blir alle «Signals» og «Slots» tilordnet sin tilhørende motpart. Det siste som skjer i konstruktøren er at *sendCommand* metoden til *TCPImageClient* blir kalt med enumer «*getSupportedResolutions*» som argument. Dette fører til at *TCPImageClient* legger kommandoen til i sendebufferen sin og starter tråden sin. Tråden starter med å forsøke å koble til kommandoserveren i ROV-en. Hvis den lykkes så hentes den siste kommandoen ut fra sendebufferen og sendes ned til serveren. Deretter sjekkes det om det skal ventes på tilbakemelding basert på hvilken enum kommandoen er. Og hva som skal gjøres med den mottatte informasjonen. I dette tilfellet skal tråden vente på tilbakemelding fra serveren.

Enumen «getSupportedResolutions» tilsier at det er støttede kamera-oppløsninger som skal returneres. De støttede oppløsningene blir sent som argument med et signal til slot-en «setSupportedCamResolutions» som ligger i MainWindow. Slot-en legger til de støttede oppløsningene i applikasjonens bildemeny. Deretter setter den hovedvinduets oppløsning lik standardoppløsningen til kameraet. Tilslutt kaller den oppstartsmetoden til TCPIImageClient, som starter objektets tråd. I mellomtiden har tråden til TCPCommandClient satt seg selv i ventemodus. Den venter helt til at den blir vekket av et nytt sendCommand kall.

Tråden til TCPIImageClient starter med å forsøke å opprette en forbindelse med bildeserveren i ROV-en. Hvis den lykkes går den inn i ei evig løkke. I denne løkken mottas gyrodata, dimensjonene til bildet, og selve bildet. Bildet og gyrodataen sendes videre til slot-en «processFrameAndUpdateGUI», før løkken starter på nytt.

Slot-en processFrameAndUpdateGUI har som oppgave å oppdatere hovedbildet med nyeste bilde og data. Bildet behandles basert på funksjonalitet som er aktivert i Image-menyen. Se tekstboksen under for en forklaring steg for steg ved hjelp av kommentarene i koden.

```
void MainWindow::processFrameAndUpdateGUI(Mat img, float gyroData) {
    if(takeSnapshot){ // Hvis "Snapshot" knappen har blitt trykket
        snapshotHandler.saveSnapshots(img); // Lagrer bildet (arbeid utføres i en annen tråd)
        takeSnapshot = false;
    }
    imageStabilization.rotationDataFilter(gyroData); // Filtreer bort ugyldige gyroverdier

    dataWindow->updatePosition(); // Oppdaterer datavinduets posisjon
    dataWindow->setResolutionLabel(img.cols,img.rows); // Setter datavinduets oppløsningstekst
    dataWindow->setRollLabel(gyroData); // Setter datavinduets "Roll" tekst
    // Regner ut og setter datavinduets "FPS" tekst (gjennomsnitt av 25 elementer)
    dataWindow->setFPSLabel(1000.0/double(t.elapsed()),true,25);
    t.restart(); // starter timeren på nytt

    // Hvis "Dynamic resizing" er merket i menyen: reskaler bildet slik at det passer vinduet
    if(ui->actionDynamic_resizing->isChecked()) dynamicResizer(img);
    // Hvis "Stabilization" er merket i menyen: roter bildet slik at horisonten blir horisontal
    if(ui->actionStabilisation->isChecked()) imageStabilization.rotateImage(img,gyroData);
    // Konverter Mat fil til QPixmap og tegn bildet
    ui->lblWebcam->setPixmap(QPixmap::fromImage(convertOpenCVMatToQtQImage(img)));
}
}
```

Figur 115. Programkoden til processFrameAndUpdateGUI.

Alle nettverksbrudd eller andre alvorlige nettverksfeil vil føre til at et feilmeldingsvindu popper opp og informerer om feilen, deretter avsluttes applikasjonen.

## Motorer

Valg av servoer var en utfordring både med tanke på størrelse, kraft og vanntetthet. Tidlig i prosjektet var dette et stort diskusjonstema om hvilke motorer som var best egnet til oppgaven. Det ble først sett på små motorer med enkodere og gir. Houxiang Zhang gav oss ett tips om noen små motorer som har gir og enkodere og har veldig høy kvalitet.

Maxonmotors heter firmaet som leverer disse motorene (Maxonmotors.uk, u.d.). Prismessig var disse motorene altfor dyr i forhold til prosjektets budsjett i tillegg til at prisen bryter med ønske fra arbeidsgiveren hvor denne ROV-en ikke skulle være noe dyr å lage og bytte deler. Derfor ble det ikke ansett som et godt valg og gruppen valgte å gå vekk fra det. Gruppen var enige om at motor og gir hadde vert en god løsning og det ble gjort flere vurderinger om motor var den beste løsningen.

Til slutt ble det valgt å gå vekk fra motor og gir på grunn at både pris og kvalitet og i stedet gå for hobby RC-servoer som har et stort marked av ulike servoer i en rekke prisklasser. I dialog med POD gruppen ble det først vurdert og gå for servoer på 16 Kg-cm med hensyn på vekten til ROV-en. Senere i prosjektet da det ble gjort store endringer i designet på ROV-en, var det ikke lenger nødvendig med servoer som hadde så høy kraft. Etter beregninger fra POD holdt det da med servoer på 6.5Kg-cm som gjorde at prisen også ble mindre. Det endte med at gruppen bestilte 12 servoer av den typen. Da servoene var motatt og testet viste det seg at disse servoene ikke kunne brukes fordi de var av typen "continuous" servoer som kan rotere kontinuerlig. Servoene ville ikke gå til gitt posisjon når de var koblet til servo kontrolleren. Det viste seg at servoen kan kun akselerasjon og fart reguleres. Dersom arbeidet skal fortsette må disse servoen byttes ut med servoer som ikke er av typen "continuous" og som kan gå til en gitt posisjon.

### Videostabilisering med gimball

Gimballstabilisering ble valgt bort da ønsket var å finne en løsning som ikke tok for mye plass, i tillegg til å redusere antall bevegelige deler. Strømforbruket ble også en faktor siden de mindre gimballenhetene bruker en spenning for å stabilisere kamera og ROV-en har begrenset med kapasitet.

### Videostabilisering med OpenCV

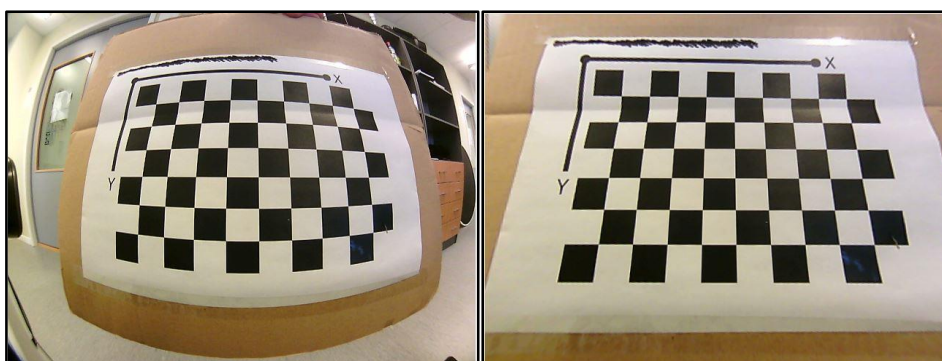
OpenCV har en seksjon som tar for seg videostabilisering, men for det første så hadde funksjonene og metodene en dårlig API og var derfor vanskelig å forstå hvordan man skulle bygge koden. For det andre, er det ikke mulig å motta video fra webkamera, da man må laste inn videofilen. Siden videostabiliseringen skal kjøres i sanntid med webkamera og at OpenCV har de begrensningene ble denne metoden valgt bort.

### Videostabilisering med ren bildebehandling

Målet med prosjektet var å kjøre videostabilisering i sanntid. Ren bildebehandling for å stabilisere en videostrøm i sanntid er ressurstungt og resulterte i en hakket video derfor ble denne metoden valgt bort som alternativ.

### Kamerakalibrering

På resultatene fra webkamerakalibreringen var det vanskelig å se om noe endring i bildet siden webkamera ikke hadde noe spesielt med linsevregning som vist under venstre er originalbildet og til høyre er korrigert bilde.



Figur 116. Bilde før og etter linsekorrigering.

Resultatene viser at med de metodene og innstillingene som ble prøvd ut ikke gir de ønske de resultatene. Øverst vises originalbildet tatt med kamera med vidvinkel linse og under vises bildet etter korrigering. Problemet som oppstår er at metodene som ble brukt ikke taklet den ekstreme vidvinkelen til kamera slik bildet ble mye mindre enn originalen og mye av detaljene ble borte. For å løse korrigeringsproblemet finnes det en egen seksjon i OpenCV for fish-eye korrigering som bruker litt andre metoder.

## Rollkompensering

Til venstre er originalbildet og til høyre er det kompenserte bildet.



Figur 117. Bilde med og uten stabilisering.

## Sinusfunksjoner

Den første metoden som ble prøvd var å bruke sinuskurver for å skape bevegelse. Sinus funksjonene viste seg å fungere sånn de burde, som vist på figur 118. Dette ble fort skrotet siden et krav var at roboten skulle kunne stå stille å se seg rundt for å inspisere området rundt seg. Å lage en sånn funksjon med en sinus generatorer ville blitt en veldig komplisert jobb. Ett annet problem var at det ble mistanke om at seriell kommunikasjonen ville slite med å overføre all dataene til servokontrolleren. Hvert eneste bein ville hatt 3 sinuskurver med faseforskyvning i forhold til hverandre. Og for at beina skulle hatt en jevn bevegelse måtte servoene i beina oppdateres ofte. Med 18 servoer er det mye informasjon som skal overføres. Dette ble ikke testet i praksis fordi gruppen ble enig om å utrede forover og invers kinematikk. Denne metoden er den mest brukte måten å styre robotarmer som har flere ledd.

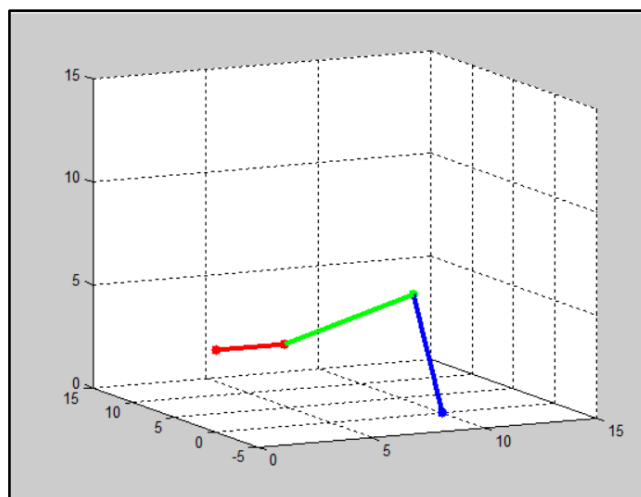
```
C:\Users\anders\Documents\Visual Studio 2015\Projects
Grader Ben 1: 173.041 Time: 0.374
Grader Ben 1: 173.149 Time: 0.375
Grader Ben 1: 173.47 Time: 0.378
Grader Ben 1: 173.68 Time: 0.38
Grader Ben 1: 173.784 Time: 0.381
Grader Ben 1: 173.886 Time: 0.382
Grader Ben 1: 173.988 Time: 0.383
Grader Ben 1: 174.19 Time: 0.385
Grader Ben 1: 174.289 Time: 0.386
Grader Ben 1: 174.388 Time: 0.387
Grader Ben 1: 174.486 Time: 0.388
Grader Ben 1: 174.679 Time: 0.39
```

Figur 118. Verdier av sinusgeneratoren.

## Kinematikk til beina

Kinematikken i prosjektet var kritisk med tanke på hvordan fartøyet skulle beveges. I starten av prosjektet ble det gjort søk etter aktuelle metoder. Etter grundig gjennomgang med hensyn på undervisningen som ble gitt i «Innføring i mekatronikk» om roboter og kinematikk, valgte gruppen å gå for et hexapod design. Hexapoder har totalt seks bein (i dette tilfellet tre bein på hver side av ROV-en). Dette designet er veldig stabilt med tanke på når den for eksempel går fremover, så har den alltid tre bein på bakken (tripod gangart). Dette gjør at programmeringen blir mye enklere og det kreves mindre kalkuleringskraft siden det blir brukt en Odroid XU4, som også skal utføre videooverføring som krever mye prosessorkraft.

En av utfordringene med kinematikken var å sette opp Jacobian matrisen. Dette var veldig nytt og var ukjent stoff for gruppen. Denne metoden ble valgt fordi invers Jacobian vil gi beinet en mer naturlig tilnærming av posisjonen den skal oppnå med tanke på vinklene. For å teste invers kinematikken ble det gjort tester i Matlab hvor endepunktet til beinet ble satt, og i en figur ble det vist hvordan beinet beveget seg for å oppnå den gitte posisjonen. Testene ble gjort i det kartesiske koordinatsystem med riktige dimensjoner på beinet sånn at det skulle bli en mest mulig autentisk simulering.



Figur 119. Matlab-simulering av bein.

Simuleringen viste at utregningene var riktige og beinet fikk en veldig naturlig bevegelse mot målet sitt. Om det ble satt en forflytning i x-retning, gikk beinet i en lineær bevegelse i x-retning og det viste tydelig at alle leddene endret vinkler for å nå målet. Siden det blir utført et nytt plot for hver iterasjon og kalkulasjonen var veldig rask, ble det satt inn en pause() i matlabskriptet for å kunne se tilnærmingen. En sekvens ble laget for å vise hvordan beinet oppfører seg når posisjonen endrer seg i et gitt mønster (se vedlegg 9). Denne sekvensen og simuleringen ble noe hakkete siden det for det første er en pause() midt i Matlab skriptet og når dette blir gjort i Java er det kun når posisjonen til beinet er funnet at vinklene blir sendt til servoene. Da vil beinet bli mer smidig siden servoene blir regulert av akselerasjonen og farten som er satt til hver utgang på i servokontrolleren.

Det ble videre gjort en test med rotasjonsmatrisen. Da roteres kroppen på ROV-en mens endepunktet til beinet da skal være på samme plass som før rotasjonen. Den vil da kunne rotere rundt sitt eget massesenter for å snu kameraet til en ønsket posisjon ved inspeksjonsområdet. Resultatet var som forventet, roterte ROV-en rundt z-aksen fulgte posisjonen til beinet etter. For mer informasjon angående matlabskript, se vedlegg 8 og 9.



## Seriell kommunikasjon med servokontrolleren

Den serielle kommunikasjonen var opprinnelig ferdig med tanke på at C++ hadde et bibliotek som fungerte opp imot Pololus servokontroller. Det som var utfordringen var at C++ var det ingen i gruppen som hadde kjennskap til og hadde brukt før. Det ble først prøvd å skrive i C++ men tiden det tok å lære seg C++ ble fort bortkastet tid i forhold til fremgang i prosjektet. Videooverføringen måtte bli skrevet i C++ på grunn av hastighetsforskjeller mellom C++ og Java. Siden kinematikken ikke hadde noe med videooverføringen å gjøre ble det tatt en beslutning å programmere kinematikken i Java. Den serielle kommunikasjonen ble derfor et annet problem siden Pololu ikke tilbyr et Java bibliotek for å styre servokontrolleren. Det ble funnet et «amatørbibliotek» (Skokan, Webområde for GitHub, 2014) som var en god start for å lage et skreddersydd program for å overføre data til servokontrolleren.

Java var et av fagene som automasjonsstudentene hadde hatt før, og dermed ble det en mindre utfordring å lage et eget program ut ifra eksempelprogrammet som ble funnet på nettet. Senere ble det oppdaget at en funksjon (`getMovingstate()`) ikke var implementert i C++ biblioteket. Den funksjonen var kritisk for å kunne få tilbakemelding om servoene er i bevegelse og dermed bruke det som et flagg for å styre en gangart sekvens. Denne funksjonen er å finne i servokontrolleren sin manual og var allerede implementert i Javabiblioteket som ble funnet. Et forenklet program ble laget og implementert i Java koden.

## Kinematikken til Javakode

Når all matematikken skulle flyttes over til Java var det noen hindringer med tanke på matriseregning. Jama ([math.nist.gov](http://math.nist.gov), 2012) var til stor hjelp da det hadde ferdige matrisekalkulasjoner som var enkle å bruke. Etter at matematikken var omskrevet til Javakode fra matlabskript ble det testet å regne ut invers kinematikken til et bein og overføre inversvinklene til servokontrolleren. Tiden det tok å regne ut og overføre dataene til et bein (3 servoer) tok under 1ms. Det ble videre testet å regne ut invers kinematikken til 6 bein (til sammen 18 servoer) og overføre alle vinklene til kontrolleren. Denne kalkulasjonen tok 1-2ms som er bedre enn forventet resultat. Et utklipp av Javakoden er vist i figur (120). Figuren viser vinklene som er regnet ut og hvor lang tid utregningen har tatt. For mer informasjon se vedlegg på CD/mapper.

```
Leg1
Theta1: 80.0
Theta2: 74.75472886942289
Theta3: 6.379355966029305
Time: 1ms
```

Figur 120. Verdier fra Java.

*Viser utregnede vinkler til ett bein og beregningstid i Java.*

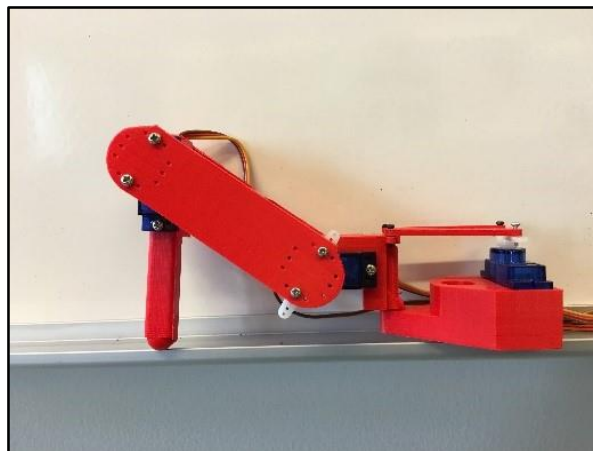


## Styring av et bein

En test ble gjort med XBOX-kontrolleren for å styre et bein som var satt sammen av POD-studentene. Testen ble gjort med et 3D-printet bein og 3 servoer som ble lånt av labben på NTNU i Ålesund. Beinets oppførte seg identisk med simuleringen som ble gjort i Matlab. Når testingen pågikk ble det oppdaget singularitet. Når beinet ikke kunne nå posisjonen som ble gitt, fikk plutselig to av vinklene for stort utslag og programmet måtte resettes.

Problemet ble løst ved å sette inn begrensninger på hvor langt beinet kunne bevege seg i x-,y- og z-retning. Beinets oppførte seg da som det skulle. Det ble en smidig bevegelse, i motsetning til Matlab-simuleringen som ikke var regulert av akselerasjonen og fart. Responstiden var også tilnærmet momentan.

Ettersom det ble brukt servoer av dårlig kvalitet til testen, ble det mye vibrering og upresise bevegelser. Dette problemet vil ikke oppstå når mer robuste og kraftige servoer brukes.



Figur 121. 3D-printet demonstrasjonsbein.

## Thrusterer

T100 var et godt valg til oppgaven og hadde egenskaper som passet til formålet. Det ble ikke fokusert så mye på thusterprogrammering og -styring. Dette fordi programmering av thrusterer ble sett på som den minste utfordringen og det ble dermed ikke gjort noe stort med. Ettersom en annen bachelorgruppe skulle bruke thrusterer, ble det bestilt inn thrusterer til begge gruppene fra samme leverandør.

## Strømforsyning

Den endelige landforsyningsløsningen med likespenning består av tre Buck konvertere: to på 12 Volt og en på 5 Volt, forsynt fra land med en spenning på 48V. Antallet Buck konvertere har blitt redusert fra fem til tre. Dette vil spare plass og vekt. En annen konsekvens er at thrusterene og servoene må kjøre på lavere spenninger enn med den forrige løsningen. For servoene sin del vil det føre til noe redusert moment. Thrusterene har ikke oppgitt ytelse i forhold til forsyningsspenning. Ved å multiplisere forsyningsspenningen 12 Volt med thrusterenes oppgitte maksimale strøm på 12,5 Ampere får man 150 Watt, som er mer enn thrusterens oppgitte maksimale effekt på 135W. Det kan tyde på at thrusterene vil yte like godt ved en lavere spenning.

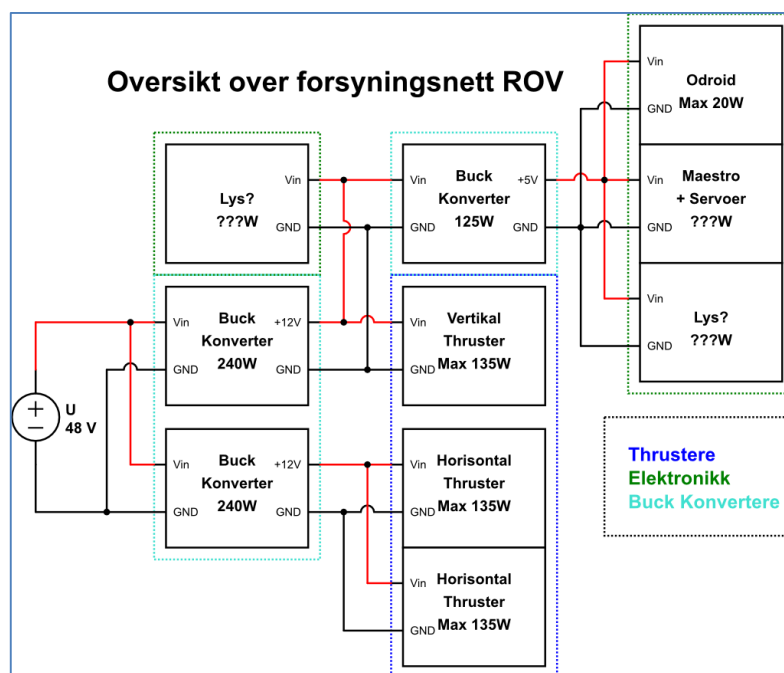
De valgte Buck konverterne har følgende spesifikasjoner:

Tabell 20. Oversikt over Buck konvertere 48V løsning

Buck konvertere for 48V løsning						
Antall	Ut(V)	Strøm(A)	Effekt(W)	Effektivitet(%)	Inn min.(V)	Inn maks.(V)
2	12	20	240	>93	28	60
1	5	25	125	>85	8	40

Forsyningsnettet er tenkt slik som vist i bildet under, der den ene 12 Volts Buck konverteren forsyner de horisontale thrusterne. De vil trolig ikke kunne kjøres på maksimal effekt siden Buck konverteren har en oppgitt maksimal effekt på 240 Watt, noe som er 20 Watt for lite. Effektbegrensningen er trolig neglisjerbar med tanke på hvilke konsekvenser det vil ha for ROV-ens manøvreringsevner.

Den andre 12 Volts Buck konverteren forsyner den vertikale thrusteren i tillegg til Buck konverteren på 5 Volt og eventuelle 12 Volts lys. Den vertikale thrusteren vil kunne yte maksimalt gitt at servoene ikke belastes samtidig, noe som ikke skal skje. Det motsatte gjelder servoene som for øvrig er forsynt av Buck konverteren på 5 Volt, sammen med resterende elektronikk.



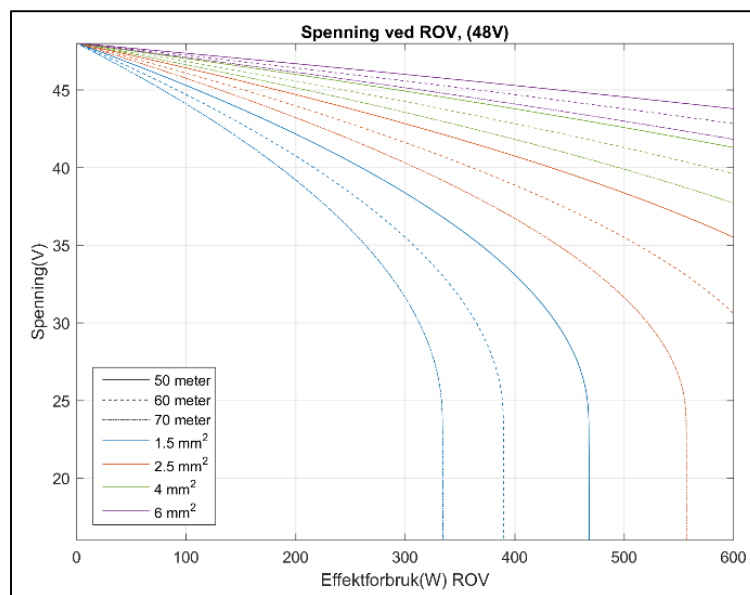
Figur 122. Forsyningsnett til ROV.

Basert på forsyningsnettet har det blitt gjort spenningsfallberegninger som vist på bildet under. Effektiviteten til 12 Volts Buck konverterne på 93% har blitt tatt med i beregningene, men effektiviteten til Buck konverteren på 5 Volt har blitt ikke blitt medberegnet. Dette fordi dens innvirkning på effektforbruket er sterkt avhengig av hvilken funksjonalitet som benyttes.

Buck konverterne på 12 Volt har en oppgitt minste inntakspenning på 28 Volt, så spenningen ved ROV-en må ikke falle under 28 Volt.

På grafen under kan man se at ROV-en kunne fungert tilfredsstillende med en  $1,5\text{mm}^2$  kabel på 50 meter. Øker man kabellengden med ti eller tjue meter må begrensninger i effektforbruket implementeres for å begrense spenningsfallet. Dersom det benyttes en kabel med tverrsnitt på  $2,5\text{mm}^2$  vil ikke spenningsfall bli et problem. Hvilket av de to tverrsnittene som egner seg best avhenger av hvor store negative konsekvenser den økte vekten til en kabel på  $2,5\text{mm}^2$  har på ROV-ens manøvreringsevner.

Effektiviteten mellom effektforbruket til ROV-en og effektleveransen fra land er direkte knyttet opp mot spenningsfallet. Se vedlegg (på CD/mapper) for grafer som viser forholdet mellom effektforbruk på land og i ROV-en, og effektivitet i prosent.



Figur 123. Spenning ved ROV, 48V forsyningspenning.

### 3.2.1 Delkonklusjon

Arbeidet som gruppen har gjennomført denne perioden har resultert i et godt grunnlag for videre utvikling. Gruppen har ikke klart å oppnå alle målene som ble satt, men det arbeidet som er utført og som ansees som ferdig er gruppen selv godt fornøyd med. Gruppen startet med lite til ingen kunnskap om programmering i C++. Dette førte til at mye av tiden i begynnelsen gikk med til å sette seg inn i C++. Effekten av valget om å bruke C++ mener vi veier opp for tiden det har tatt å lære språket, med tanke på læringsutbytte og ytelse.

Gode bilder fra havbunnen var et sterkt ønske fra oppdragsgiver. Det ble derfor satt fokus på en kombinasjon av rask overføring og god bildekvalitet. Det er ønskelig å hente ut bildet som er maskinvare-enkodet av kamera, men gruppen har ikke klart å implementere dette. Kameraet og løsningen vi endte opp med gir et middels godt bilde. Kvaliteten er nok ikke tilstrekkelig for profesjonelt bruk. Løsningen er mer et konseptbevis som kan videreutvikles.

Målet med å oppnå gode bilder fra havbunnen ble møtt med mange utfordringer. Gruppen startet med å se på videostabilisering for å få en klar og stabil videostrøm. Rolling shutter korrigering med videostabilisering ble brukt en del tid på, men ble valgt bort da programmeringen og omfanget av denne formen for stabilisering ble for krevende. Vidvinkelkorrigering er ikke optimal siden korrigeringen fjernet mye av bildet som kan være en ulempe, linsekorrigeringsmetodene som har blitt utforsket lager også en merkbar tidsforsinkelse i videooverføringen. Videostabilisering i sanntid med ren bildebehandling resulterte i en hakkete video da behandlingen var tung for systemet. Rollkompensering ble implementert og fungerer bra, men ikke optimalt på grunn av støy fra gyroen. Bildestabilisering av video er ressurskrevende.

Kinematikken til beina har vært en stor utfordring med tanke på at gruppen hadde lite erfaring med forover- og inverskinematikk. Derfor ble det brukt mye tid på å finne en løsning på kinematikkproblemet. Det finnes mange ulike måter å løse et kinematisk problem. Metoden som gruppen valgte er en Pseudoinvers Jacobian transformasjon som regner ut invers kinematikken. Dette var en god løsning med hensyn på både hurtighet og nøyaktighet. Gruppen er klar over at det finnes mange ulike løsninger på problemet som kan vise seg å være bedre. Metoden som ble valgt løste invers kinematikken på en tilstrekkelig måte som tilfredstilte våre krav til beinstyring.

ROV-en skulle også være enkel å bruke. Det var en av tingene som ble vektlagt når kontrollapplikasjonen ble utviklet. Dette har blitt delvis oppnådd. Kontrollapplikasjonen lar seg enkelt kontrollere med mus og tastatur. Den skulle også kunne styres med en håndkontroller, men det har ikke blitt oppnådd. Kontrollapplikasjonen kobler seg til og fra serveren automatisk, noe som forenkler brukeropplevelsen. Dog noen feilhåndteringsmekanismer må implementeres for at løsningen skal bli fullgod. Som nevnt tidligere har en del funksjonalitet ikke blitt implementert, men det som er implementert fungerer på en tilfredsstillende måte.

Totalt sett er gruppen fornøyd med arbeidet med tanke på tids- og budsjettbegrensninger. Det er klart for hele gruppen at løsningene som er valgt ikke trenger å være de beste. Med tanke på oppdragsgiver og videre utvikling syntes gruppen at det er et godt utviklet konsept. Løsningene som er valgt er gruppen fornøyd med og tror at noen av løsningene kan også ende opp i det ferdige produktet. Samarbeidet med Produkt- og Systemdesign har vært en god erfaring både med tanke på hvordan deres arbeidsmetode er og hvordan det er å være avhengig av hverandres valg og arbeid. Det har naturligvis vært kollisjoner og diskusjoner underveis, men ikke mer enn at hele bachelorgruppen fant en løsning på problemet.

## Anbefalt videre arbeid

Videre arbeid på serverløsningen vil være å gjøre et forsøk med Raspberry pi 3 med Windows IoT. Vi mener at en serverløsning med Raspberry pi og Windows IoT kan gi bedre resultater enn hva som har blitt oppnådd med vår løsning. Windows har egne biblioteker for å hente ut enkodet bilde fra kamera som gjør at 3.parts programmer ikke er nødvendig for enkoding og dekodning. Alternativt kan hele problemstillingen omgås ved å benytte et kamera med GigE Vision. Det vil medføre at støtte for GigE Vision må implementeres i kontrollapplikasjonen.

Serverapplikasjonen må startes manuelt hver gang Odroiden skrus på. For at serverapplikasjonen skal kunne starte opp automatisk ved oppstart av ROV må applikasjonen kompiles til en kjørbart løsning. Den kompilerte applikasjonen må legges i oppstartslisten til operativsystemet. Det burde også legges inn at dersom programmet ikke lenger fungerer, så bør serveren startes på nytt på en trygg måte.

For både serverapplikasjonen og kontrollapplikasjonen bør det implementeres støtte håndkontrolleren og de signalene den sender. Med løsninger for styring av lys, bein, thrustere, modusvalg og bakovervendt kamera.

I kontrollapplikasjonen bør det implementeres GUI-elementene som er nevnt i konseptet til kontrollapplikasjonen.

For å forsyne ROV-en med kraft ble kun en fasttilkoblingsløsning med likespenning utbedret. En batteriløsning bør også utbedres før et endelig valg av forsyningsløsning blir tatt.

Når det gjelder bildebehandlingen kan det forsøkes på andre måter å gjøre behandlingen på slik den blir mindre ressurskrevende. Fish-eye korrigeringen bør forbedres for å ivareta mer av bildet etter korrigeringen. Det anbefales montere hvitt lys og IR-lys slik at kameraet fanger opp fargene bedre for å hjelpe på inspeksjonen.

Det som videre burde vært gjort er å undersøke om punktet som er satt til et bein er oppnåelig. Det som skjer med den nåværende invers Jacobian transformasjonen dersom endepunktet er uoppnåelig, er at Jacobian matrisen blir ustabil og vil gi veldig høye inverse vinkelverdier. Dette problemet er gruppen klar over og ved videre utvikling bør dette problemet løses.

#### 4 RESULTATER OG DISKUSJON

En del komponenter ble bestilt tidlig prosjektperioden. Prosessen vi har vært gjennom gjør at vi ville ha gjort noen endringer i valg av komponenter. Under følger en beskrivelse av komponenter og materiell prosjektgruppa ville gått til innkjøp av på bakgrunn av den informasjonen vi besitter i dag. Diverse materiell i forbindelse med intern sammenkobling er ikke inkludert i tabellen. For å finne merverdiavgift er importkalkulatoren fra toll.no brukt. Se vedlegg under felles for kvitteringer.

Tabell 21. Planlagt materiell og komponenter.

Antall	Beskrivelse	Pris (ink.mva)	Total NOK
1	Håndkontroller: XBOX ONE	448,-	448,-
1	Single-board maskin: Odroid XU4	1034,-	1034,-
1	Sensortilkobling: Arduino	187,-	187,-
1	Sensor: Adafruit 10-DOF	250,-	250,-
2	Kamera: ELP 170 wide angle USB500W02M	582,-	1164,-
3	Thrustermotor fra BlueRobotics (M100)	1766,-	5298,-
1	Servokontroller: Mini-Maestro 24 kanaler	819,-	819,-
18	Robot servo: 300 graders servo	251,-	4518,-
1	Spenningsomformer: 24 til 5V: Buck converter step down	270,-	270,-
2	Spenningsomformer: 48 til 24V: Buck converter step down	158,-	316,-
	Lys: CCTV CCD 60 degree 48LEDs IR	131,-	131,-
1	Pakningsmateriell (1x500x1000 mm <sup>3</sup> )	149,-	149,-
7	O-ringer	50,-	50,-
158	42 x M4 Rampa (til kjøleplate og modul) og M4 skruer 15 x M6 Rampa (til deksel) og M6 skruer 22 x M3 Skruer og M3 muttere (til kamera og thrusterfester)	ca 150,-	ca 150,-
3	3D-filament (ABS)	299,-	897,-
1	Plate i aluminium (3x1000x3000 mm <sup>3</sup> )	25,-	25,-
1	Pleksiglassplate (4x1200x800 mm <sup>3</sup> )	350,-	350,-
		<b>Sum:</b>	<b>16 056,-</b>

Tabell 21 viser den totale planlagte kostnaden ved å bygge en observasjons-ROV med bein basert på arbeidet som er gjort hittil. Budsjettet viser en liten overskridelse basert på det samlede budsjettet fra NTNU i Ålesund og Kystverket. Dersom vi hadde hatt mulighet til å jobbe mer med prosjektet, ville vi gått gjennom komponentlisten på nytt og forhandlet rabatter med leverandører slik at budsjettet ikke overskrides.

Tabell 22 viser en oversikt over komponenter vi har kjøpt inn så langt i prosjektet. Komponenter og materiell lånt fra NTNU i Ålesund inkluderes ikke i gjennomgangen.

Tabell 22. Medgått materiell og komponenter i løpet av prosjektperioden.

Antall	Beskrivelse	Pris (NOK)	Total (NOK)	Link/butikk
1	Servokontroller: Mini-Maestro 24	819,-	819,-	<a href="https://www.pololu.com/product/1356">https://www.pololu.com/product/1356</a>
12	Robot servo: 360 graders servo	251,-	3 014,-	<a href="http://www.levihobby.com/ProdView.aspx?id=861">http://www.levihobby.com/ProdView.aspx?id=861</a>
1	Powerline Ethernet: ZYXEL PLA5456	1189,-	1189,-	<a href="https://www.netonnet.no/art/data-og-nettbrett/nettverk/powerline-homeplug/zyxel-pla5456-twin-pack/225854.11049/">https://www.netonnet.no/art/data-og-nettbrett/nettverk/powerline-homeplug/zyxel-pla5456-twin-pack/225854.11049/</a>
3	Thruster: Bluerobotics T100	1766,-	5298,-	<a href="https://www.bluerobotics.com/store/thrusters/t100-thruster/">https://www.bluerobotics.com/store/thrusters/t100-thruster/</a>
58	Festemidler	29,-	29,-	Tingstad
1	Pakningsmaterial 0,8 mm	50,-	50,-	Biltema
1	Gummiduk 1 mm	55,-	55,-	Biltema
		<b>Sum:</b>	<b>10 585,-</b>	

Som det fremkommer av tabell 22 har prosjektgruppa overholdt budsjettet på 15 000,- NOK. Enkelte komponenter (som et kamera og kameralys) ble bestilt til lab-ene til NTNU i Ålesund og havnet derfor utenfor prosjektgruppas budsjett.

- Servoene ble kjøpt inn for å visualisere programkoden. Maestro 24 var et servokontrollerkort som oppfylte kriteriene knyttet til funksjonalitet.
- Kamera ble kjøpt inn for å vise fram resultatet av rotasjonskorrigering i videooverføringen.
- Lyset skulle testes med kameraet, men ble levert defekt.
- Powerline ble bestilt for å teste ytelsen i kombinasjon med strømaggregat.
- Tingstads festemidler ble kjøpt inn for å teste innfestningsmetoden i plastmaterialet (om dette faktisk lot seg gjøre).
- Pakningsmaterial ble kjøpt inn for å teste om Epilog Helix laserskjærer kan benyttes til utforming av pakningene, samt om nøyaktigheten i skjæremetoden var høy nok.
- T100 thruster ble kjøpt inn for å teste på et tidlig skrogdesign.

Videre er det verdt å nevne at dersom vi hadde jobbet videre med prosjektet, ville arbeidet resultert i en rekke nye tekniske valg og løsninger, noe som ville resultert i en delvis eller helt ny komponentliste. Blant annet ville prosjektgruppa brukt tid på å utrede en batteriløsning som kraftforsyning.



## 5 KONKLUSJON

Prosjektgruppa har arbeidet med et konsept for en liten og kompakt observasjons-ROV. ROV-en henter inspirasjon fra eksisterende fartøy i observasjonsklassen, samt enkle biomimetiske prinsipper i forbindelse med utvikling av bein og skrog.

Vårt konsept har et hydrodynamisk skrog, seks bein til stabil gange og en samlet vekt på rundt 7 kg (eksklusiv kabelen). Dimensjonene til skroget er 40x21x16 cm. To thrustere står for horisontal forflytning, og en thruster gjennom skrogets midtseksjon brukes til vertikal forflytning. Et frontkamera og et ryggekamera sender live rotasjonskorrigert videostrøm i vidvinkel med mulighet til å ta linsekorrigerede snapshots. Dette ligger tett opp Kystverkets ønske om en lett og manøvrerbar ROV som kan produsere gode bilder. Det ble hentet inspirasjon fra naturen ved at en krabbe ble brukt som utgangspunkt for designet, som etterspurt av NTNU i Ålesund. Prosjektgruppa har også vært innom et hummer- og kråkebolledesign på veien mot det siste designforslaget.

Basert på samtaler i prosjektgruppa og tilbakemeldinger fra kontaktpersonen fra Kystverket har observasjons-ROV-en utviklet på NTNU i Ålesund både velfungerende og mindre hensiktsmessige løsninger.

De løsningene som fungerer godt inkluderer formen og størrelsen på skroget, samt rotasjonskorrigerende som gjør at man til enhver til forholder seg til en HMS-vennlig synsvinkel uavhengig av helningen til ROV-en. Kommunikasjonen mellom kontrollstasjon og fartøyet er også et velfungerende konsept.

De mindre hensiktsmessige løsningene inkluderer strømforsyningen, som er lite tilpasset Kystverkets behov, og utformingen på beina, som vil gjøre fartøyet svært utsatt for å sette seg fast i tang og tare og bidrar til å gi uønsket moment med nåværende utforming. Som beskrevet i delkonklusjonene i kapittel 3, er prosjektgruppen fornøyd med arbeidet som er gjort hittil på beina. Det bør likevel gjøres en vurdering på hvorvidt arbeidet med dem skal videreutvikles. Dette fordi det er usikkert hvor mye nytte de tilfører oppdragsgiver. Sett fra et læringsperspektiv har dette aspektet ved oppgaven likevel vært nyttig for studentene som har jobbet med det. Å ha bein på ROV-en i oppgavebeskrivelsen tilførte utvilsomt en ekstra dimensjon til læringsutbyttet.

## Referanseliste

### 6 REFERANSER

- 1st Vision. (2014). *Om oss: Webområde for 1st Vision*. Hentet fra Webområde for 1st Vision: [https://1stvision.com/store/index.php?main\\_page=product\\_info&cPath=1&products\\_id=379](https://1stvision.com/store/index.php?main_page=product_info&cPath=1&products_id=379)
- adafruit10dof. (u.d.). *Adafruit*. Hentet 02 15, 2016 fra <https://www.adafruit.com/product/1604>
- Advanced Diving Technology. (2016). *Aditech - Advanced Diving Technology*. Hentet Mars 10, 2016 fra <https://www.aditech-usa.com/en/shop/695-gnom-baby-01-m-3-050-1-01.html>
- All About Circuits. (u.d.). *Webområde for All About Circuits*. Hentet April 21, 2016 fra Webområde for All About Circuits: <http://www.allaboutcircuits.com/textbook/digital/chpt-3/logic-signal-voltage-levels/>
- Amazon Inc. (u.d.). *Webområde for Amazon Inc.* Hentet fra Webområde for Amazon Inc.: <http://www.amazon.com/DROK-Converter-Step-down-Transformer-Waterproof/dp/B00CE75K0W>
- American Lobster. (2016). *American Lobster*. Hentet April 7, 2016 fra <http://www.americanlobster.com.vn/americanlobster/>
- American Oil & Gas Industry Historical Society. (2011). *American Oil & Gas Industry Historical Society*. Hentet Februar 23, 2016 fra <http://aoghs.org/wp-content/uploads/2011/01/ROV-Nuke-AOGHS-300x201.jpg>
- Arduino. (u.d.). *Webområde for Arduino*. Hentet fra Webområde for Arduino.
- Association, A. I. (2016, 05 06). *Webområde for GigE Vision - True Plug and Play Connectivity*. Hentet fra <http://www.visiononline.org/vision-standards-details.cfm?type=5>
- Ayers, J. (2004, Juli 1). Underwater walking. *Arthropod Structure & Development*, ss. 347–360.
- Barford, V. (2013). *Pisces III: A dramatic underwater rescue: Webområde for BBC NEWS*. Hentet Februar 23, 2016 fra <http://www.bbc.com/news/magazine-23862359>
- Bennett, J. C. (2005). Practical Computer Analysis of Switch Mode Power Supplies. I J. C. Bennett, *Practical Computer Analysis of Switch Mode Power Supplies*. CRC Press, 2005 .
- bildesensor. (u.d.). *OVT*. Hentet 03 26, 2016 fra <http://www.ovt.com/products/sensor.php?id=177>
- Biomimicry Institute. (2016, April 15). *History: Webområde for The Biomimicry Institute*. Hentet fra Webområde for The Biomimicry Institute: <https://biomimicry.org/history/#.VxC8j8sz6jl>
- Biomimicry.org. (2016). *Biomimicry.org*. Hentet Mars 18, 2016 fra <https://biomimicry.org/what-is-biomimicry/#.Vuv2KssXCjl>
- Bjartnes, A. (2015). *Energi og klima*. Hentet Februar 25, 2016 fra <http://energiogklima.no/blogg/bjartnes/det-gronne-skiftet/>
- Bjerga, J. (2003). *Norsk Oljemuseum*. Hentet Februar 24, 2016 fra [http://www.norskolje.museum.no/stream\\_file.asp?iEntityId=333](http://www.norskolje.museum.no/stream_file.asp?iEntityId=333)
- Bjørnstad, J. (2009, 02 14). *Store norske leksikon*. Hentet 04 28, 2016 fra [https://snl.no/minste\\_kvadraters\\_metode](https://snl.no/minste_kvadraters_metode)
- Blue Robotics. (u.d.). *Om oss: Webområde for Blue Robotics*. Hentet fra Webområde for Blue Robotics: <https://www.bluerobotics.com/store/thrusters/t100-thruster/>

- Bluebird Marine Systems Ltd. (2016). *Bluebird Marine Systems Ltd.* Hentet Februar 22, 2016 fra [http://www.bluebird-electric.net/ROV\\_remotely\\_operated\\_vessels\\_UUV\\_underwater\\_unmanned\\_autonomous\\_drones.htm](http://www.bluebird-electric.net/ROV_remotely_operated_vessels_UUV_underwater_unmanned_autonomous_drones.htm)
- bluerobotics. (u.d.). (bluerobotics) Hentet Mai 25, 2016 fra <https://www.bluerobotics.com/store/thrusters/t100-thruster/>
- Boston Dynamics. (2016, April 15). *RHex - Devours Rough Terrain: Webområde for Boston Dynamics.* Hentet fra Webområde for Boston Dynamics: [http://www.bostondynamics.com/robot\\_rhex.html](http://www.bostondynamics.com/robot_rhex.html)
- Boston Dynamics. (2016, April 14). *RiSe: The Amazing Climbing Robot: Webområde for Boston Dynamics.* Hentet fra Webområde for Boston Dynamics: [http://www.bostondynamics.com/robot\\_rise.html](http://www.bostondynamics.com/robot_rise.html)
- Burrows, M. (2013). *Forskning.no.* Hentet 29 Februar, 2016 fra <http://forskning.no/biologi-dyreverden/2013/09/fant-forste-dyr-med-tannhjul>
- Buss, S. R. (2009, Oktober 07). *Department of mathematics.* Hentet 04 28, 2016 fra <https://www.math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf>
- Buss, S. R. (2009, Oktober 7). *Webområde for University of San Diego.* Hentet fra Webområde for University of San Diego: <https://www.math.ucsd.edu/~sbuss/ResearchWeb/ikmethods/iksurvey.pdf>
- Cao, Y. (2015, Mars 10). *Om oss: Webområde for YouTube.* Hentet fra Webområde for YouTube: <https://www.youtube.com/watch?v=3whQFs0-9Qg>
- Chiles, J. (2014). *The inhuman history of ROVs: Part 2. Webområde for Jim Chiles, personlig blogg.* Hentet Februar 23, 2016 fra <http://www.disaster-wise.blogspot.com/2014/04/the-inhuman-history-of-rovs-part-2.html>
- Coyle, T. J. (2015). *DIY Drones.* Hentet Mars 1, 2016 fra <http://diydrones.com/profiles/blogs/openrov-trident-an-underwater-drone-for-everyone>
- DIY Photography. (2012, September 24). *Webområde for DIY Photography.* Hentet fra Webområde for DIY Photography: <http://www.diyphotography.net/everything-you-wanted-to-know-about-rolling-shutter/>
- Doster, K. M. (u.d.). *Om oss: Webområde for International Symposium on Power Line Communications.* Hentet fra Webområde for International Symposium on Power Line Communications: [http://www.isplc.org/docsearch/Proceedings/1997/pdf/0563\\_001.pdf](http://www.isplc.org/docsearch/Proceedings/1997/pdf/0563_001.pdf)
- ELP Surveillance. (u.d.). *Om oss: Webområde for ELP Surveillance.* Hentet fra Webområde for ELP Surveillance: <http://www.elpcctv.com/170-degree-fisheye-lens-usb-camera-module-support-yuymjeg-p-78.html>
- elp170. (u.d.). *amazon.* Hentet fra 26
- Europakommisjonen. (2014). *Press releases: Webområde for Europakommisjonen.* Hentet Februar 26, 2016 fra [http://europa.eu/rapid/press-release\\_MEMO-14-336\\_en.htm](http://europa.eu/rapid/press-release_MEMO-14-336_en.htm)
- Eyman, S. (1986). *Sun Sentinel.* Hentet Februar 22, 2016 fra [http://articles.sun-sentinel.com/1986-01-05/features/8601010812\\_1\\_russian-navy-russian-air-force-russian-revolution/2](http://articles.sun-sentinel.com/1986-01-05/features/8601010812_1_russian-navy-russian-air-force-russian-revolution/2)
- FFMPEG About. (u.d.). Hentet fra <https://www.ffmpeg.org/about.html>
- GitHub. (u.d.). *Om oss: Webområde for GitHub.* Hentet fra Webområde for GitHub: [https://github.com/MicrocontrollersAndMore/OpenCV\\_3\\_Windows\\_10\\_Installation\\_Tutorial](https://github.com/MicrocontrollersAndMore/OpenCV_3_Windows_10_Installation_Tutorial)

- Gnom ROV. (2016). *Gnom ROV*. Hentet Mars 10, 2016 fra [http://www.gnomrov.com/\\_files/editor/image/baby3.jpg](http://www.gnomrov.com/_files/editor/image/baby3.jpg)
- Goodard, C. (2015). *Webområde for the World In 2016*. Hentet Februar 25, 2016 fra <http://www.theworldin.com/article/10625/ocean-business?fsrc=scn/tw/te/bl/ed/theworldin2016>
- Gstreamer About*. (u.d.). Hentet fra <https://gstreamer.freedesktop.org/data/doc/gstreamer/head/pwg/html/part-introduction.html>
- HABIABADI, M. M. (2013, Mai). Analytical Workspace, Kinematics, and Foot Force Based Stability of Hexapod Walking Robots. *WORCESTER POLYTECHNIC INSTITUTE*. Hentet fra <https://www.wpi.edu/Pubs/ETD/Available/etd-042413-205514/unrestricted/magheli.pdf>
- Hambley, A. R. (2014). *Electrical Engineering*. PEARSON.
- Hambley, A. R. (2014). *Electrical Engineering: Principles and Applications*. Pearson College.
- Hjeltnes, T. A. (2014, Januar 6). *Webområde for NTNU*. Hentet fra Webområde for NTNU: <https://www.ntnu.no/iie/fag/mop/Modul%202/mop-ls2-fasemodell-2014v.pdf>
- ICD Software. (u.d.). *Webområde for ICD Software*. Hentet fra Webområde for ICD Software: <http://www.icdsoftware.no/products/cdp-studio>
- Innovative Solutions & Technologies . (2011, April 12). *Under water: Webområde for Innovative Solutions & Technologies SDN BHD*. Hentet fra Webområde for Innovative Solutions & Technologies SDN BHD: [http://www.istsb.com/services-high\\_pressure\\_heat\\_page.php?pageact=under\\_water](http://www.istsb.com/services-high_pressure_heat_page.php?pageact=under_water)
- itseez. (u.d.). *OpenCV*. Hentet fra <http://opencv.org/about.html>
- Jai. (u.d.). Webområde for Jai. *Webområde for Jai, s.* [http://www.jai.com/en/support/jai\\_sdk\\_and\\_control\\_tool](http://www.jai.com/en/support/jai_sdk_and_control_tool).
- Jamfes F.Kurose, K. W. (2013). *Computer Networking: A Top-Down Approach*. Pearson Education.
- Jäntti, E. (2010, April 25). *Webområde for Wikipedia*. Hentet fra Webområde for Wikipedia: [https://commons.wikimedia.org/wiki/File:Sea\\_Urchin\\_test\\_5629\\_03\\_22.jpg](https://commons.wikimedia.org/wiki/File:Sea_Urchin_test_5629_03_22.jpg)
- Jones, T. (2011). *hexapodrobots*. Hentet Mai 03, 2016 fra <http://hexapodrobots.weebly.com/advantages-of-hexapod-gait.html>
- Jørgensen, J. (2016, April 21). *Hvad er PLA? Webområde for GreenCreative*. Hentet fra Webområde for GreenCreative: <http://www.greencreative.dk/pages/hvad-er-pla>
- kamera kalibrering*. (u.d.). (Matlab) Hentet 03 16, 2015 fra <http://se.mathworks.com/help/vision/ug/camera-calibration.html>
- kamera, r. s. (u.d.). Hentet 05 24, 2016 fra [https://iso.500px.com/wp-content/uploads/2015/01/slowmoshutter\\_cover.png](https://iso.500px.com/wp-content/uploads/2015/01/slowmoshutter_cover.png)
- kamerakalibrering, O. (u.d.). *Opencv*. Hentet 04 10, 2016 fra [http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html)
- kameraplassering, s. m. (u.d.). Hentet 05 24, 2016 fra [http://www.vision.caltech.edu/bouguetj/calib\\_doc/](http://www.vision.caltech.edu/bouguetj/calib_doc/)
- Karlsen. (2014). Prosjektledelse: fra initiering til gevinstrealisering. Bergen: Universitetsforlaget.
- Kystverket. (2016). *Mål og visjon: Webområde for Kystverket* . Hentet Mars 16, 2016 fra <http://kystverket.no/Om-Kystverket/Kva-er-Kystverket/Mal-og-visjon/>

- Let's Make Robots. (2014, Januar 13). *Om oss: Webområde for LMR*. Hentet fra Webområde for LMR: <http://letsmakerobots.com/node/39931>
- Levi Hobby. (u.d.). *Om oss: Webområde for Levi Hobby*. Hentet fra Webområde for Levi Hobby: <http://www.levihobby.com/ProdView.aspx?id=860>
- Liang, O. (2012, Januar 09). *www.oscarliang.com*. Hentet Mars 03, 2016 fra <https://oscarliang.com/inverse-kinematics-and-trigonometry-basics/>
- Lindblad, S. (2008). Hentet Februar 19, 2016 fra Forretningsprosess: <http://www.forretningsprosess.no/lean-en-filosofi/>
- LLC, T. S. (u.d.). *Test Standard Labs LLC*. Hentet fra [http://teststandard.com/data\\_sheets/ABS\\_Data\\_sheet.pdf](http://teststandard.com/data_sheets/ABS_Data_sheet.pdf).
- Luxion Keyshot. (2016, Mai 26). *Om Keyshot: Webområde for Keyshot*. Hentet fra Webområde for Keyshot: <https://www.keyshot.com/company/>
- Marine Technology Society. (2014). *ROVs - a brief history: Webområde for ROV Committee of the Marine Technology Society*. Hentet Mars 1, 2016 fra [http://www.rov.org/rov\\_history.cfm](http://www.rov.org/rov_history.cfm)
- Marx, R. (1970). The New Science of Hunting Underwater Treasure. *Popular Mechanics*, 133(6), 102.
- math.nist.gov*. (2012, November 09). Hentet 03 17, 2016 fra <http://math.nist.gov/javanumerics/jama/>
- Matlab. (u.d.). *mathworks*. (Matlab) Hentet 03 16, 2015 fra <http://se.mathworks.com/help/vision/ug/camera-calibration.html>
- MatWeb. (u.d.). *Om oss: Webområde for MatWeb*. Hentet fra Webområde for MatWeb: <http://www.matweb.com/reference/tensilestrength.aspx>
- Maxonmotors.uk*. (u.d.). (Maxonmotors) Hentet Mai 14, 2016 fra <http://www.maxonmotor.co.uk/maxon/view/content/index>
- Michel, D. (u.d.). *Webområde for Marine Advanced Technology Education*. Hentet Februar 19.02, 2016 fra Webområde for Marine Advanced Technology Education : [http://www.marinetech.org/files/marine/files/Workforce/KSG\\_ROV%20techFINAL.pdf](http://www.marinetech.org/files/marine/files/Workforce/KSG_ROV%20techFINAL.pdf)
- Microsoft. (u.d.). *Om oss: Webområde for Microsoft XBOX*. Hentet fra Webområde for Microsoft XBOX: <http://www.xbox.com/en-US/xbox-one/accessories/controllers/wireless-controller>
- Microsoft. (u.d.). *Webområde for Microsoft*. Hentet fra Webområde for Microsoft: [https://msdn.microsoft.com/en-us/library/windows/desktop/dd373389\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd373389(v=vs.85).aspx)
- microsoft. (u.d.). *www.xbox.com*. (Microsoft) Hentet April 21, 2016 fra <http://www.xbox.com/en-US/xbox-one/accessories/controllers/wireless-controller>
- MSDN About Directshow*. (u.d.). Hentet fra [https://msdn.microsoft.com/en-us/library/windows/desktop/dd373389\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/dd373389(v=vs.85).aspx)
- National Oceanic and Atmospheric Administration. (2010). *Ocean Explorer: Webområde for National Oceanic and Atmospheric Administration*. Hentet Februar 25, 2016 fra <http://oceanexplorer.noaa.gov/oceanos/explorations/10index/background/edu/media/rov.pdf>
- National Oceanic and Atmospheric Administration. (2014). *National Ocean Service: Webområde for National Oceanic and Atmospheric Administration*. Hentet Februar 25, 2016 fra <http://oceanservice.noaa.gov/facts/exploration.html>
- National Oceanic and Atmospheric Administration. (2015). *Ocean Explorer: Webområde for National Oceanic and Atmospheric Administration*. Hentet Februar 24, 2016 fra <http://oceanexplorer.noaa.gov/technology/subs/subs.html>

- nghiaho*. (2014, 02 20). Hentet 01 08, 2016 fra <http://nghiaho.com/?p=2093>
- NMRI Japan. (2001, November 15). *Om oss: Webområde for NMRI Japan*. Hentet fra Webområde for National Maritime Research Institute Japan: [https://www.nmri.go.jp/eng/khirata/fish/experiment/upf2001/body\\_e.html](https://www.nmri.go.jp/eng/khirata/fish/experiment/upf2001/body_e.html)
- Norsk Elektroteknisk komité. (2002). *Norsk elektroteknisk norm Elektriske lavspenningsanlegg* (2. utg.). Oslo: Norsk Elektroteknisk komité.
- Not My War. (u.d.). <http://www.notmywar.com/hva-er-kinematikk/>. Hentet 2016 fra <http://www.notmywar.com/hva-er-kinematikk/>
- NTNU. (u.d.). *NTNU.no*. Hentet Mai 12, 2016 fra <https://innsida.ntnu.no/wiki/-/wiki/Norsk/Matlab+for+studenter>
- Oceanic Institute. (2016). *About our oceans: aqua facts: Webområde for Hawaii Pacific University*. Hentet Februar 19, 2016 fra <http://www.oceanicinstitute.org/aboutoceans/aquafacts.html>
- OpenCV. (2015, Desember 18). *Om oss: Webområde for OpenCV*. Hentet fra Webområde for OpenCV: <http://docs.opencv.org/3.1.0/#gsc.tab=0>
- OpenCV. (u.d.). *Opencv*. Hentet 04 10, 2016 fra [http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html)
- OpenCV. (u.d.). *Opencv*. Hentet 04 10, 2016 fra [http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera\\_calibration/camera\\_calibration.html](http://docs.opencv.org/2.4/doc/tutorials/calib3d/camera_calibration/camera_calibration.html)
- OpenROV. (2016). *Webområde for OpenROV*. Hentet fra Webområde for OpenROV: <http://store.openrov.com/products/6-pack-rechargeable-li-fepo4-batteries>
- Oracle. (u.d.). Hentet 04 11, 2016 fra The History of Java Technology: <http://www.oracle.com/technetwork/java/javase/overview/javahistory-index-198355.html>
- Oracle. (u.d.). Hentet 04 11, 2016 fra About the Java Technology: <https://docs.oracle.com/javase/tutorial/getStarted/intro/definition.html>
- Ose, P. E. (2016, Januar 12). Introduksjonsmøte.
- Paschoa, C. (2014, April 12). *Marine Technology News*. Hentet Februar 23, 2016 fra Webområde for Marine Technology News: [http://www.marinetechologynews.com/blogs/pioneer-work-class-rovs-\(curv-i-iii\)-e28093-part-1-700495](http://www.marinetechologynews.com/blogs/pioneer-work-class-rovs-(curv-i-iii)-e28093-part-1-700495)
- Pololu. (u.d.). *Pololu*. Hentet Mai 13, 2016 fra <https://www.pololu.com/docs/0J40/4>
- Pololu. (u.d.). *Webområde for Pololu*. Hentet fra Webområde for Pololu: <https://www.pololu.com/picture/view/0J2336>
- Pro Marine Services. (2016). *Pro Marine Services*. Hentet Mars 31, 2016 fra <http://www.promarineservices.co.uk/rov.html>
- programming, W. s. (u.d.). *Wikibooks*. Hentet fra [https://en.wikibooks.org/wiki/Serial\\_Programming/termios](https://en.wikibooks.org/wiki/Serial_Programming/termios)
- Provenzano, C. A. (u.d.). *Pthread: Basic Routines*. Hentet fra <http://www.cs.wm.edu/wmpthreads.html>
- QRStuff. (2016). *Om oss: Webområde for QRStuff*. Hentet fra Webområde for QRStuff: <http://www.qrstuff.com/>
- Qt. (2015, November 02). *About Qt*. Hentet fra [https://wiki.qt.io/About\\_Qt](https://wiki.qt.io/About_Qt)
- QT Co. (u.d.). *Om oss: Webområde for QT Co*. Hentet fra Webområde for QT Co.: <http://doc.qt.io/>

- QT Co. (u.d.). *Om oss: Webområde for QT Co.* Hentet fra Webområde for QT Co.: <http://doc.qt.io/qt-5/signalsandslots.html>
- RC Network De. (u.d.). *Om oss: Webområde for RC Network De.* Hentet fra Webområde for RC Network: [http://www.rc-network.de/upload\\_02/forum\\_0002/1116770368.JPG](http://www.rc-network.de/upload_02/forum_0002/1116770368.JPG)
- Rebikoff-Niggeler Foundation. (2016). *Rebikoff.org*. Hentet Februar 21, 2016 fra <http://www.rebikoff.org/History/history.html>
- Regjeringen Stoltenberg I. (2000, April 12). *Webområde for Regjeringen*. Hentet Februar 3, 2016 fra [https://www.regjeringen.no/no/aktuelt/verdier\\_fra\\_havet\\_-\\_norges\\_fremtid/id242767/](https://www.regjeringen.no/no/aktuelt/verdier_fra_havet_-_norges_fremtid/id242767/)
- Rice University. (u.d.). *Webområde for Rice University*. Hentet fra Webområde for Rice University: [https://www.clear.rice.edu/elec201/Book/basic\\_mech.html](https://www.clear.rice.edu/elec201/Book/basic_mech.html)
- Rolstadås, O. J. (2014). *Praktisk Prosjektledelse: fra idé til gevinst*. Fagbokforlaget.
- Ross, K. &. (sixth edition). *Computer networking*. PEARSON.
- RovMarine Technologies. (2016). *RovMarine Technologies*. Hentet Februar 22, 2016 fra <http://www.rovmarine.it/en/component/content/article/14-not-categorized/16-the-history-of-rovs>
- Ruan, J. (2013). *Projectrhea*. Hentet April 20, 2016 fra <https://www.projectrhea.org/rhea/index.php/Jacobian>
- rxtx qbang.org*. (u.d.). Hentet Mai 11, 2016 fra <http://rxtx.qbang.org/wiki/index.php/Download>
- Saab Seaeye. (2016). *Products: Seaeye LARS, Webområde for Saab Seaeye*. Hentet Februar 19, 2016 fra <http://www.seaeye.com/lars.html>
- Saab Seaeye. (2016). *Products: TMS: Webområde for Saab Seaeye*. Hentet Februar 19, 2016 fra <http://www.seaeye.com/tms.html>
- Safak, K. K., & Adams, G. G. (2002, Juni 14). Modeling and simulation of an artificial muscle and its application to biomimetic robot posture control. *Robotics and Autonomous Systems*.
- Seaview Systems Inc. (2016). *What is an ROV?: Webområde for Seaview Systems Inc.* Hentet Februar 23, 2016 fra <http://www.seaviewsystems.com/questions/what-is-an-rov/>
- Semb-Johansson, A. (2015). *Store Norske Leksikon*. Hentet April 4, 2016 fra <https://snl.no/insekter>
- Silvercrest Submarines and ROVs. (2016, April 11). *ROVs Sales: Webområde for Silvercrest Submarines and ROVs*. Hentet fra Webområde for Silvercrest Submarines and ROVs: <http://silvercrestsubs.co.uk/hyballfulldetails.html>
- Sistare, G. (2013). *Serious Wonder*. Hentet Mars 8, 2016 fra <http://www.seriouswonder.com/espys-360-rov/>
- Skokan, L. (2014, June 13). *Github*. Hentet 05 11, 2016 fra <https://github.com/skoky/pololumaestro>
- Skokan, L. (2014, Juni 13). *Webområde for GitHub*. Hentet fra Webområde for GitHub: <https://github.com/skoky/pololumaestro>
- Socket, I. (u.d.). *linuxhowtos*. Hentet fra [http://www.linuxhowtos.org/C\\_C++/socket.htm](http://www.linuxhowtos.org/C_C++/socket.htm)
- Sperre ROV Technology AS. (2016). *Sperre AS*. Hentet Februar 23, 2016 fra <http://sperre-as.com/produkter/>
- stabilisator, g. (u.d.). Hentet 05 24, 2016 fra <http://www.aliexpress.com/item/Zhiyun-Z1-Smooth-3-Axis-Handheld-Stabilizing-Phone-Gimbal-for-iPhone-6plus-for-GoPro-H3-H4/32327128279.html>

- Stensvold, T. (2015, November 17). *Webområde for Teknisk Ukeblad*. Hentet fra Webområde for Teknisk Ukeblad: <http://www.tu.no/artikler/slik-fungerer-den-norske-ubat-dronen-til-under-20-000-kroner/276000>
- Stensvold, T. (2015, Desember 4). *Webområde for Teknisk Ukeblad*. Hentet fra Webområde for Teknisk Ukeblad: <http://www.tu.no/artikler/norsk-ubat-drone-finner-plastsoppel-i-havet/276142>
- Stewart, H. B. (2013). *Ocean Explorer: Webområde for National Oceanic and Atmospheric Administration*. Hentet Februar 26, 2016 fra <http://oceanexplorer.noaa.gov/history/quotes/explore/explore.html>
- Store Norske Leksikon. (2015, April 26). *Store Norske Leksikon*. Hentet fra Store Norske Leksikon: <https://snl.no/hummer>
- Teensy. (u.d.). *PJRC*. Hentet 02 20, 2016 fra <https://www.pjrc.com/teensy/>
- TestStandard. (u.d.). *Om oss: Webområde for TestStandard*. Hentet fra Webområde for TestStandard: [http://teststandard.com/data\\_sheets/ABS\\_Data\\_sheet.pdf](http://teststandard.com/data_sheets/ABS_Data_sheet.pdf)
- Thai National Parks. (2016, April 25). *Webområde for Thai National Parks*. Hentet fra Webområde for Thai National Parks: [www.thainationalparks.com](http://www.thainationalparks.com)
- The Economist Intelligence Unit. (2015). *The Economist Insights*. Hentet Februar 29, 2016 fra [http://www.economistinsights.com/sites/default/files/Blue%20Economy\\_briefing%20paper\\_WOS2015.pdf](http://www.economistinsights.com/sites/default/files/Blue%20Economy_briefing%20paper_WOS2015.pdf)
- The Rebikoff-Niggeler Foundation. (2016, Februar 17). *History: Milestones: Webområde for Rebikoff.org*. Hentet fra Webområde for Rebikoff.org: <http://www.rebikoff.org/History/MILESTONES/milestones.html>
- theuzo007. (2013, Oktober 26). Hentet 05 11, 2016 fra <https://theuzo007.wordpress.com/2013/10/26/joystick-in-java-with-jinput-v2/>
- Thompson, J. R. (2014). *Katie Ph.D: RoboLobsters to the rescue*. Hentet Mars 23, 2016 fra <http://www.katiephd.com/robolobsters-to-the-rescue/>
- toolbox, C. v. (u.d.). *mathworks*. Hentet 01 20, 2016 fra <http://se.mathworks.com/products/computer-vision/>
- Unbrako Group. (2016, Mai 20). *Webområde for Unbrako Group*. Hentet fra Webområde for Unbrako Group: <http://unbrako.com/images/unb-screws.jpg>
- uncyclopedia.wikia.com. (2016). *uncyclopedia.wikia.com*. Hentet April 7, 2016 fra <http://uncyclopedia.wikia.com/wiki/File:Robolobster.gif#filelinks>
- Underwater Inspection. (2011). *Underwater Inspection*. Hentet Mars 11, 2016 fra <http://www.underwaterinspection.org/videoray-products/scout/>
- usb.org. (u.d.). Hentet fra [http://www.usb.org/developers/docs/devclass\\_docs/](http://www.usb.org/developers/docs/devclass_docs/)
- van der Meeren, G. I. (2008). *Anatomi og fysiologi hos tifotkreps med vekt på nervesystem og velferd*. Ålesund: Havforskningsinstituttet og Møreforsking Ålesund.
- Videoray. (2004, Mars 1). *Videoray*. Hentet fra Webområde for Videoray: [http://shop.videoray.com/images//Scout/scout\\_v1rb.pdf](http://shop.videoray.com/images//Scout/scout_v1rb.pdf)
- Wakeham, D. (2016). *James Dyson Award*. Hentet Mars 8, 2016 fra <http://www.jamesdysonaward.org/it/projects/esp-360-rov/>
- Wakeham, D. (2016, Februar 16). *James Dyson Award*. Hentet Mars 9, 2016 fra <http://www.jamesdysonaward.org/projects/esp-360-rov/>



- Warren Photographic. (2016, April 25). *Webområde for Warren Photographic*. Hentet fra Webområde for Warren Photographic: <http://www.warrenphotographic.co.uk/13655-shore-crab>
- Webområde for Reload*. (2016, 05 31). Hentet fra Webområde for Produktwissen til Reload: [http://www.baumarktwissen.eu/Medien/mdb/data/de/24063/24069/24064/rckschlagventil\\_rckf.html](http://www.baumarktwissen.eu/Medien/mdb/data/de/24063/24069/24064/rckschlagventil_rckf.html)
- Webområde for TheUZ0007*. (2013, Oktober 26). Hentet fra Webområde for TheUZ0007: <https://theuzo007.wordpress.com/2013/10/26/joystick-in-java-with-jinput-v2/>
- Weisstein, E. (u.d.). *mathworld*. (Wolframalpha) Hentet 05 09, 2016 fra <http://mathworld.wolfram.com/RotationMatrix.html>
- Wikipedia. (2016, 05 30). *Om oss: Webområde for Wikipedia*. Hentet fra Webområde for Wikipedia: [https://en.wikipedia.org/wiki/Finite\\_element\\_method](https://en.wikipedia.org/wiki/Finite_element_method)
- Winsock, M. (u.d.). Hentet fra [https://msdn.microsoft.com/en-us/library/windows/desktop/ms741416\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/ms741416(v=vs.85).aspx)
- Woods Hole Oceanographic Institution. (2015). *Woods Hole Oceanographic Institution*. Hentet Februar 29, 2016 fra <https://www.whoi.edu/main/topic/titanic>
- www.axadd.com*. (u.d.). Hentet April 21, 2016 fra <http://www.axadd.com/products/digital-servos/robot-servos/659.html>
- Xuggler About*. (u.d.). Hentet fra <http://www.xuggler.com/xuggler>
- Zhang, H. (2015). *fronter.com*. Hentet Mai 12, 2016 fra [https://fronter.com/hials/links/files.phtml/1236135321\\$538716717\\$/Dokumenter/Presentations/Introduction+to+Mechatronics\\_Lecture+\\_Modular+robots+2015.pdf](https://fronter.com/hials/links/files.phtml/1236135321$538716717$/Dokumenter/Presentations/Introduction+to+Mechatronics_Lecture+_Modular+robots+2015.pdf)
- Zu, L. &. (2011, Desember). Study and design of an inchworm-like micro-robot walking mechanism. *Advances in Automation and Robotics*, 2, s. 71.

## Bibliografi

Aarnes, Halvor. (2003). Økologi. Sist oppdatert 23.08.2006.

Tilgjengelig fra: <http://www.mn.uio.no/ibv/tjenester/kunnskap/plantefys/okologi/okologi.pdf>

Allen, Thomas J., & Bachmann, Richard J., & Quinn, Roger D., & Ritzmann, Roy E. (2003). Abstracted Biological Principles Applied with Reduced Actuation Improve Mobility of Legged Vehicles. *Intelligent Robots and Systems*, 2, 1370-1375.

DOI: 10.1109/IROS.2003.1248835

Altendorfer, R., Moore, N., Komsuoglu, H., Buehler, M., Brown, Jr. H.B., McMordie, D., Saranli, U., Full, R., Koditschek, D.E. (2001). RHex: A Biologically Inspired Hexapod Runner. *Autonomous Robots*, 11, 207-213.

Ayers, J. (2004). Underwater walking. *Arthropod Structure & Development*, 33, 347-360.

Beer, Randall D., & Chiel, Hillel J., & Quinn, Roger D., & Ritzmann, Roy E. (1997). Biologically inspired approaches to robotics: what can we learn from insects? *Communications of the ACM*, 40, utgave 3, 30-38.

Delcomyn, Fred. (2007). Biologically Inspired Robots. *Bioinspiration and Robotics: Walking and Climbing Robots*.

Han, Baoling., & Luo, Qinsheng., & Wang, Qiuli., & Zhao, Xiaochuan. (2007). Research on Hexapod Walking Bio-Robot's Workspace and Flexibility. *Bioinspiration and Robotics: Walking and Climbing Robots*.

Huang, Qingjiu. (2007). Posture and Vibration Control Based on Virtual Suspension Model for Multi-Legged Walking Robot. *Bioinspiration and Robotics: Walking and Climbing Robots*.

Quinn, Roger D. & Ritzmann, Roy E. (1998) Construction of a Hexapod Robot with Cockroach Kinematics Benefits both Robotics and Biology. *Connection Science*, 10, 239-254. DOI: 10.1080/095400998116422

Soyguder, Servet & Alli, Hasan. (2007). Design and prototype of a six-legged walking insect robot. *Industrial Robot: An International Journal*, 34, 412-422.

Wilson, Donald M. (1966). Insect Walking. *Annual Review of Entomology*, 11, 103-122.

Gadget Gangster. Accelerometer & Gyro tutorial

Tilgjengelig fra: <http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/>

Karpenko, Alexandre., & Jacobs, David., & Baek, Jongmin., & Levoy, Marc. (2013) Digital video Stabilization and Rolling Shutter Correction using Gyroscopes. *Stanford Tech Report*.

Tilgjengelig fra: [http://171.67.77.70/papers/stabilization/karpenko\\_gyro.pdf](http://171.67.77.70/papers/stabilization/karpenko_gyro.pdf)

Ovrén, Hannes., & Forssén, Per Erik. (2015) Gyroscope-based video stabilization with auto-calibration.

Tilgjengelig fra: <http://liu.diva-portal.org/smash/get/diva2:841497/FULLTEXT02.pdf>

## VEDLEGG

### *Felles*

Vedlegg 1 Møteagendaer

Vedlegg 2 Møtereferater

### *Produkt- og systemdesign*

Vedlegg 3 Ukentlig logg

Vedlegg 4 FEM-analyser i NX10.0

Vedlegg 5 Maskintegninger av observasjons-ROV

### *Automatiseringsteknikk*

Vedlegg 6 Matlabskript for beregning av spenningsfall

Vedlegg 7 Utledning av formel for spenning ved ROV

Vedlegg 8 Visualisering av kinematikk

Vedlegg 9 Fremvisning av robotbein

Vedlegg 10 Rotasjonsmatrise-funksjon til fremvisning av robotbein

Vedlegg 11 Inverse Jacobian funksjon til fremvisning av robotbein

Vedlegg 12 Klassediagram for styring av bein

Vedlegg 13 Klassediagram for serverapplikasjon

## **Vedlegg på CD**

### *Felles*

Kvitteringer

### *Produkt- og systemdesign:*

Presentasjon av hovedprosjekt

FEM-analyser (animasjoner)

Simulering utført i NX Mechatronics Concept Designer

### *Automatiseringsteknikk*

Presentasjon av hovedprosjekt

Strømforsyning

Kildekoder

Videoer

Kamera og linse

# Vedlegg 1

## Møteagendaer

Spørsmål til Kystverket – Per Erik Ose

1. Skal prosjektet ta utgangspunkt i en ROV eller skal det lages en mindre versjon, smal nok til å utføre formålet ved trangere områder?
2. Havdybde? (ved fjøre/flosjø(?))
3. Hva er formålet? Hvilket arbeid skal utføres? /Bruksområde?
4. Krav til mobilitet?
5. Krav til sensorer?
6. Synsvinkel kamera? Roterbart (som sikkerhetskamera på buss)? Type Verktøy den må ha-tang inkl 360 rotasjon (for eksempel)
  - a. Trenger ROV utstyr til å fjerne tank og tare o.l.?
7. Maks vekt totalt (ROV)
  - a. Behov for egen heis/senkekran
  - b. Størrelse på ROV?
8. Arbeidslys
  - a. Styrke, antall, fram, bak og på sidene
9. Skal den ha støvsuger for å ta prøver / kunne blåse bort sand?
10. Plassering og utforming av fjernstyring og forsyning av ROV.
11. Kabellengde?
12. Krav til vedlikehold?
13. Budsjett?
14. Bidrag fra Kystverket? (materiell, verktøy, og andre tjenester)
15. Værforhold i.e. høyvann
16. Hva brukes i dag for slike operasjoner / oppgaver?
17. Skal vårt produkt brukes til å merke de 100 nye skjærene som er omtalt i media i det siste?
18. Mulighet for utflukt?

Spørsmål til Ottar

1. En eller to rapporter?

## Møteagenda

Tid: Mandag 8. februar 2016 kl. 09.00-09.30

Sted: A436

- Rapporten: hvordan dele den opp mest hensiktsmessig? Ok med to deler?

1. Design og konstruksjon

2. Elektronikk og styring + bilde og bildebehandling

- Kommentarer/forslag til forprosjektrapporten?
- Trykksensor: forslag til type?
- Godkjenning av handleliste

---

# Møteagenda for Krabbe-ROV

29.02.16 kl.09.00-10.00

Rom: A436

- Kan kamera kjøpes utenfor vårt budsjett?
  - Dersom ja, kan vi kjøpe et dyrere kamera med bedre kvalitet som laben kan overta ved prosjektslutt?
  - Foreløpig kandidat <http://www.elpcctv.com/5mp-hd-wide-angle-fisheye-lens-usb-camera-with-yuy-and-mjpef-output-p-84.html>
  
- Mulig ny avgjørelse angående beina: servo mellom ledd og ikke wire langs hele lengden
  - Kommentar fra veiledere?
  
- Oppdatering fra PoS om «skallet»
  - ... freses ut i skum, som vi bruker som en negativ form ved senere støyping
  
- Oppdatering fra Automatiseringsteknikk
  - Nytt programmeringsspråk på Odroid: fra C++ til Java
  - Gruppen har besluttet å bruke AC strømforsyning
    - Powerline Ethernet

---

# Møteagenda for Krabbe-ROV

14.03.2016 Kl 09:00 – 10:00

Rom: B431

## Hydraulikk

- Foreløpig ingen rimelig løsning

## Design

- Endret design
- Mindre størrrelse

## Strømkilde

- Har 2 potensielle DC løsninger.
- Buck converter (Step-down DC/DC)
- Stor bredde på input

## Kryssflettet fibertau

- Svak mot mekanisk slitasje
- Sterk og tynn

## Kamerastabilisering

- Kost nytte avveining
- Lite gevinst for translasjonkompensasjon

## CDP studio

- Tilbake til C++ i ROV



---

# Møteagenda for Krabbe-ROV

04.04.2016 Kl 09:00 – 10:00

Rom: A436

- Endelig design
    - Sfære-ROV
    - Tilpassede bein
      - Vise hydraulisk funksjon
  
  - Bildestabilisering hittil
    - Videoeksempel
    - Tar lengre tid enn forventet
  
  - Script – MatLAB-sekvens av bevegelsen til ett bein
  
  - Vise fram video-overføringen
  
  - Strømløsning
    - Spenningsfallsgraf
-

---

## Spørsmål til veiledere

- Er det nødvendig med gyrostabilisering på selve beina?

---

# Møteagenda for Krabbe-ROV

18.04.2016 Kl 09:00 – 10:00

Rom: B434

- vise fram bevegelse av bein (i Java) ved bruk av kontroller
- vise fram stabilisert kamera
- vise fram resultat av spenningsberegninger
- Demonstrere videostreaming og forklare det
- vise fram nye bein
- vise fram nytt design

## Spørsmål til veiledere

- Bruk av bilder i bacheloroppgave – holder det å oppgi kilder, eller må man kontakte hver enkelt nettside og be om lov?
  - Skal vi oppgi dimensjoner i rapporten (som størrelser på bein/skall) eller er det nok å med maskintegningene?
    - o Kan det være lurt med oppsummering i tabell og maskintegninger i tillegg?
  - Er det noen av veilederne som har motforestillinger mot å bruke rapportmalen beskrevet av Nils Olsson fra NTNU? (Vi fikk opplæring i denne)
-

- 
- 1) *Innledning*
  - 2) *Teoretisk grunnlag*
  - 3) *Metode – deles i to med egne avsnitt for teoretisk bakgrunn, metode, resultat, diskusjon og delkonklusjon*
  - 4) *Resultater*
  - 5) *Diskusjon*
  - 6) *Konklusjon*
-

---

# Møteagenda for Krabbe-ROV

02.05.2016 Kl 09:00 – 10:00

Rom: B431

- Vise real-time korrigering av video
- Vise fram videokommunikasjon og GUI
- Vise fram programmert prototype av bein
- Vise resultat av oppdriftsberegninger og medførte konsekvenser

---

# Møteagenda for Krabbe-ROV

16.05.2016 Kl 09:00 – 10:00

Rom: B434

- Vise fram GUI (direkte) og konseptet
- Vise fram analyser fra NX10.0

## Til veiledere

- Innleveringsdato
  - Invitasjon til framføring 26 og 27.05
-

# Vedlegg 2

## Møtereferater

# MØTEREFERAT

## Produkt- og Systemdesign

Tid: 12. januar, kl 09.00-10.00

Sted: Kunnskapsparken NTNU i Ålesund

- - Det legges 20 000 meter sjøkabel i året langs kysten
  - Antall båter går ned
  - Oljevernslåter med mulighet for å sette ned merker
- Vår ROV skal fungere i skvalpesonen og i grunt vann (0 – 25 m) og ha god live bildeoverføring fra havbunnen, samt være lett og kompakt - men likevel robust nok til å stå imot bølgepåkjenninger.
- Vi kan hente inspirasjon fra OpenROV. Den har ikke optimalt design ut ifra behov beskrevet i kravspesifikasjonen, men den fungerer godt på dypt vann og kan lære oss noe om de grunnleggende mekanismene.
- NTNUs BluEye fungerer godt på dypt vann, men mindre godt i urolig grunt vann.
- Naturen er et godt utgangspunkt for designet ettersom evolusjonen allerede har brukt millioner av år på å finne en optimalisert løsning på vårt problem (hvordan gå, hvordan stå imot kreftene i skvalpesonen, hvordan redusere motstand, osv). Et problem for dagens ROVer er at de lett setter seg fast i tang, tare, o.l.

Kravspesifikasjoner fra Per Erik Ose:

- ROV skal være lett og liten nok til å holde i hendene og kaste ut i sjøen. Dette er *viktig*. Den skal fungere i skvalpesonen. Maksdybde vi skal jobbe ut ifra er 25 meters dybde. Pakninger og tetninger er derfor viktig å finne ut av. Effekten av bølger er minimal på mer enn 15 meters dyp (dypere enn 10-15 meter er vannet helt rolig).
- Kameraet og dets minne må være bra, gjerne med utbyttbart minne. Full HD-kvalitet er bra nok oppløsning, så lenge FPS er høy. Det ble diskutert en GoPro med Gyballstabilisator. Gjerne få til 360 graders synsvinkel (ett eller to kamera – utred!).
- Med hensyn til oppdrift er design viktig. Avlang form er foretrukket. Høyere form medfører mindre påvirkning i skvalpesonen, men gir sterkere bølgepåkjenninger. Få ned antall bevegelige deler og bruk rimelige, standardiserte løsninger. Per Erik vil gjerne at ROV skal kunne legges sammen og ha et håndtak, gjerne med oppladbart batteri som kan byttes ut (bly), eller styres gjennom et aggregat. ROV må være utstyrt med LED-lys til kameraet slik at man kan få god bildekvalitet. Et blybatteri er svært tungt så gruppa må utrede andre løsninger enn å plassere det i ROV. Per Erik foreslår bruk av 12- eller 24volts aggregat og ett eller to batterier i ROV.
- Innmaten til ROV burde bestå av enkle komponenter som kan bestilles og leveres kjapt, slik at død-perioden til ROV er minst mulig dersom noe går i stykker. Ikke bruk spesielle materialer som må maskineres, osv. Det er derfor ønskelig å bruke hyllevarer som Arduino o.l.

Man må også være obs på at gruppen burde bruke elektrisk signal og ikke akustisk, da det finnes en god del akustisk signal i sjøen allerede som kan avbryte/kansellere ROV-signalet. Rapid Prototyping er veldig bra. Per Erik Ose har en egen 3D-printer tilgjengelig. Honeycomb-struktur å foretrekke. Fremfor å bruke propeller for bevegelse når ROV filmer, må ROV være utstyrt med bein som skal krabbe seg framover slik at grums ikke blåses opp foran kameraet. Bruk gjerne thrustere til å flytte



ROV fram og oppover.

- Størrelse: ROV skal være stor nok til å få plass i en standard koffert (helst) og trenger ikke høy effekt. Vi må vurdere størrelse mot stabilitet; hva som er viktigst når og hvorfor.
  - Konklusjon: stabilitet avgjør størrelsen, men skal helst få plass i praktisk koffert.
- Ekstra ønsker: ROV skal kunne manøvrere gjennom tang og tare, samt ta prøver av taren. I dag er dette svært kostbart, men med ROV kan det gjøres langt rimeligere. I dag leies det inn eksternt personell som foretar en visuell inspeksjon.
- Framtiden ligger under vann, og derfor er det viktig at vi er den første i kjeden til å lage et slikt produkt. Det vi gjør er basisforskning og vi skal kun bygge en prototype. Vår prototype skal være utgangspunkt til neste års kull (bachelor). Kystverket skal være med å sponse prosjektet, først og fremst bistå med utstyr (3D-printer), men også økonomisk dersom det trengs.
- Det ideelle er at ROV kan gå fra land til vann – men det er situasjonsbestemt. Den skal håndtere akkurat når en bølge bryter. Maks vekt er bestemt av hvor tungt en enkelt person kan bære. Over 15kg er ugunstig. Man kan bruke kabel med positiv eller nøytral oppdrift.
- Offisielt prosjektmøte hver 14. dag. Studentene har ansvar for å kalle inn, sende ut møteagenda 24 timer i forveien, samt booke lokale.

## MØTEREFERAT

### Produkt- og Systemdesign

Tid: 25. januar, kl 09.00-10.00

Sted: NTNU i Ålesund

- Budsjett – fra 10 000 til 15 000. Kystverket vil sponse med 5000,- NOK i første omgang.
- Bruk siste uken av bachelorperioden til å finpusse på rapporten. Dette kan veiledere hjelpe med til.
- Automasjonsgruppa oppfordres til å ta kontakt med Houxiang Zhang angående løsning til beina og motorer fra leverandøren Ali Baba, samt se på encodere med «hall-effect». Det finnes 4 stykker i tillegg til steppermotoren (3-4, men ganske store) på laben til NTNU i Ålesund.
- Designe to alternativ? Kan være lurt å bygge en primitiv prototype som kun går på land for å teste ut løsninger for bein, motor osv.
- Kjøpe: noen HDMI-overganger som tillater stream. Kan holde kameraet «utenfor» budsjettet dersom det ikke koster så mye mer enn 5000,- NOK. Dette kameraet blir kjøpt inn til labben, og vi låner det.
- Strømforsyning kan være ala pc-strømforsyning. Kjøpe TP-kabel med strøm gjennom.
- Kjøpe en BASE som lages som et anker, som senkes ned i sjøen. Her kan man feste essensielle komponenter som ROV ikke behøver å ha på.
- Sette opp to alternativer til design av thrustere (tre eller fire stk thrustere).

- NIKON KeyMission 360 er et alternativ til GoPro-kamera.
- Elektriske muskler? Automasjon oppfordres til å skissere alle alternativene i rapporten samt teste design på prototype som ikke er vanntett. Husk å skrive ned alle forslag til løsninger i rapporten: hydraulikk, elektriske muskler osv.

# MØTEREFERAT

## Produkt- og Systemdesign

Tid: 8. februar, kl 09.00-10.00

Sted: NTNU i Ålesund

- Det er bestemt å dele opp rapporten i to ulike deler, men ha felles innledning, sammendrag, forord og bakgrunn. Del 1 (design, konstruksjon) og del 2 (elektronikk, styring, bildebehandling). Da kan man avslutte med en felles konklusjon for hele rapporten. Del 1 og del 2 vil ha separate utsnitt for teoretisk bakgrunn, metode, resultat, drøfting og delkonklusjon.
- Forprosjektrapporten er godt skrevet. Per Erik sa at man ikke skal undervurdere prosjektet. Bildekvaliteten blir ganske dårlig desto lavere man dykker. Derfor kan prosjektet bli prøvende. Det blir presisert fra Per Erik at bildekvaliteten er alfa og omega for Kystverket.
- Foreløpig ingen endring i budsjett. Budsjett er imidlertid ikke "endelig" – Kystverket kan bidra med mer dersom vi kan argumentere for at man trenger mer midler.
- Ottar er usikker på om en hovedforfatter til rapporten: han ønsker at hvert medlem skriver om sin del, men at gruppen har en hovedansvarlig som sørger for at alle skriver jevnt og trutt og har progresjon med rapporten.
- Hovedrapporten er grunnlaget til karakter, men at vi bygger en prototype kan forenkle vurderingen. Veilederne har mulighet til å gi individuell karakter, men dette krever mye dokumentasjon og er som oftest aktuelt dersom en person ikke har gjort mye arbeid i løpet av prosjektperioden.
- Endelig handleliste – Thrustere og ESC er bestilt sammen med ei annen gruppe. For lys må man vurdere skjerming / refleksjon av lys i kameraet. Grums som reflekteres, er gjerne tett og ligner en vegg på bildet. Hvor man plasserer lys er mer eller mindre likegyldig i motsetning til plasseringen av thrustere (design-fase).
- Husk å dokumentere hver forandring som vi gjør på designet. Bilder og forklaring.
  - Per Erik sa at den ideelle formen er ei kule. Designet må avrundes i kantene mye mer.
- Se på en annen type vaier – telefonlinje eller noe annet. Man kan få ei sene/line som er sterk nok til formålet fremfor å bruke en vaier. Dyrkorn har en line som er billig og sterk (kryssflettet fibertau bruk for dypvannsfiske med stang).
  - Fibertvinnnet tråd versus Spectra-tau
  - Line til innhenting av ROV – sjekk ut Dyrkorn i Skarbøvika (kryssflettet fibertau)
- Per Erik Ose reagerte på spenningen til innmaten – kan hende gruppen ikke har optimalisert ROV. Vekselstrøm i sjøvann ved jordfeil kan være farlig.
- Dersom vår ROV blir for tung, vil thrustere prøve å motarbeide bevegelsen for å holde den stabil. Vi må derfor huske at vekten spiller en stor rolle når vi designer den.
- Husk å dokumente dimensjonerende faktorer (de deler som skal bestemme størrelsen – thrustere, kamera). Ta screenshots underveis - prosessen er viktigere enn resultatet.

- Servoene er i bevegelse maks halve tiden: det vil si at servoene utgjør 20% av «duty-cycle» til ROV. Da ser man på totalenergibehovet og deler det på 5 for å se reell energibehov. Dersom vi bruker et batteri, kan man kutte amplituder (topper) som er uønsket ved bevegelse. Gjennomsnittsbelastning er viktigere enn maksbelastning. Batterier kan eventuelt kompensere for topper i energiforbruket.
  - Beregne beina og moment på servoene.
  - Energiforbruk: gjennomsnittsbelastning viktigere enn maks. belastning. Batterier kan avlaste for topper i forbruk. Hovedsakelig thrustere som bruker mye, deretter servoene.
- Per Erik ønsker likestrøm og ikke vekselstrøm. Automasjon har ikke tenkt nok på sikkerhet når det gjelder elektronikk. Dette ser veilederne frem til å lese i rapporten. Per Erik er meget skeptisk til bruk av 220V.
- Vulkan-tape tåler sjømiljø ved at den trekker seg sammen.
- Redesign rundt dimensjonerende faktor (for Kystverket har batteri vært dimensjonerende faktor)
  - Utfordring: gjøre barnevennlig, kompakt og rimelig. Automasjon: Utred alternativ til vekselstrøm!
- Kamera – igjen blir Nikon Keymission nevnt. Det virker som at 360 grader synsvinkel er ønsket. Per Erik synes det hadde vært snedig å bruke to kamera med 180 grader foran og bak. Viktig å prøve og eksperimentere ute og som vi kan argumentere for i rapporten. Det viktigste for Kystverket er å få gode bilder fra ROV.
  - Designutfordring: optimal plassering av kamera

## • MØTEREFERAT

### Produkt- og Systemdesign

Tid: 29. februar, kl 09.00-10.00

Sted: NTNU i Ålesund

- Viser prototyper (frest skall, 3D-printede bein)
- Automasjon etterspør et kamera som kan gå på labens budsjett, ikke prosjektets. Veiledere ønsker demonstrasjon på algoritmer før man går til innkjøp, for å rettferdiggjøre kjøpet. Christian Hellebostad sier at en demo er langt unna. Hvor høy oppløsning greier man å prosessere (FPS)? Ottar presiserer at kamera kan kjøpes inn kun etter at han får sett demonstrasjon av at koden (prosesseringen) fungerer. Inntil videre kan vi bruke et dårligere kamera med lavere oppløsning og kvalitet til at koden fungerer.
- Per Erik si det er ideelt med 180° synsvinkel på kamera. Ønsker detaljbilder, ikke bare svarthvitt kontrastbilder. Ideelt om man kan velge et område å zoome inn på. Framerate viktigst for bildekvaliteten
- AC/DC må skrives godt om i rapporten. Oppdragsgiver må føle seg «sett» og vår ønsker DC-løsning.
- Ottar etterlyser at Automasjon tester det som kan testes tidlig, for eksempel skyvekraft til thrustere. Viktig med testing før montering. Test av powerline-ethernet. Gruppen burde teste ut alt mulig før man kjøper for å spare å kjøpe inn for mye unødvendig duppedingser. Dette innebærer å kontrollteste.
- Hydraulikk-løsning i beina? Inspirert av krabbe. Hva gjør at hydraulikk ikke lar seg gjøre? Godt rapportmateriale.

- Per Erik sa: Kameraet er innenfor – 170 grader er ganske bra. Igjen nevnte han å ha to kamera (foran og bak). Feil valg av lys kan gi svart-hvitt effekt på kameraet. Dette burde gruppen finne ut av. Hva med to kamera? Det ene kameraet kan gjerne være dårligere (bakre kamera), kan være utrolig greit når man er i «tareskog» slik at man får en oversikt både framover og bakover.
- Gruppen burde bare kjøre testing på beina ledd for ledd slik at man kan fullføre en bevegelse før den neste, og ikke ta hele i et steg.
- Ottar: Latency og delay er ikke en bekymring, i og med at under sjøen skjer ikke bevegelser for fort til å trenge lav delay. Per Erik og nevnte at de kunne leve med «problemet». Man må ikke få det problemet hvor man må vente på at signalet kommer inn. Veiledere foreslo å teste en stream for å se hvilken båndbredde som egentlig trengs. Videostreamen burde være komprimert, og man antar at MJPEG er komprimert.
- Per Erik er skeptisk til ROVens størrelse – kan den ikke gjøres mindre, spør Per Erik.
  - Grunnet momentbetraktninger er det bedre å ha thrusterne på undersiden. Størst mulig forutsigbarhet i bevegelsene til ROVen ønskelig.
  - Slik designet ser ut nå er det nødvendig med en utvendig strømppe til beina
  - Thruster-plasseringen bak kan gi dårlig manøvrering sekundet man kjører den i revers (rotering). Robin nevnte at å snu designet ikke er dumt ettersom at den kan være gunstig for å ikke sette seg fast. Per Erik svarte med at den må være forutsigbar. Å snu designet vil ikke hjelpe mye annet enn at beina kan gi et moment i motsatt retning. Flymodus vs gåmodus.
  - Egen tilføyelse: kanskje sette thrutserne et sted på midten?
- I rapport: krabben bruker hydraulikk for å bevege delene – hadde det vært mulig? Et par ord om å sammenlikne grunn for vaier mot hydraulikk og evt andre alternativ.
  - Nevn også at siden Kystverket vil ha DC strøm, må vi ha en grundig utredning for begge løsningene dersom Automasjon velger AC strøm. Hvorfor og hvordan.
- Roostock Max 2 (?) 3D-printer. 26 x 79, men vanligvis halvparten så høy. Printer i PTG. PTG er fleksibel og ganske sterkt iforhold til ABS og PLA. Ikke like sprøtt som PLA.

## **MØTEREFERAT**

### Produkt- og Systemdesign

Tid: 14. mars, kl 09.00-10.00

Sted: NTNU i Ålesund

- Gammeldags flyvingedesign kan sikre gjennomstrøming på beina
- POD foreslår å bytte design til et hummerdesign, noe Per Erik er positiv til
- Hydraulikk skal vurderes som et alternativ. Per Erik har gode erfaringer med bruk av hydraulikk. Små apotekssprøyter (kanyler) er gode for prototyping, ettersom de er laget for å ikke suge luft. Paul Steffen foreslår å bruke vann eller matolje som hydraulikkmedie, på den måten er det ikke miljøfarlig dersom lekkasjer oppstår. Foreslår også å bruke pumpe + ventil fremfor servo?
- Christian informerer om den kinesiske leverandøren av servorer og forklarer at vi har tenkt å plassere to servoeer i hvert bein for å spare plass. Servoene skal ta seg av heving av beina, og vaiere skal ta seg av sving-bevegelsen til beina. Christian informerte også om et nytt design for beina inspirert av gammeldagse flyvinger. De vil gi vanngjennomstrømming.
- Vi oppfordres til å se på to roboter fra HiTech for å hente inspirasjon (Per Erik), samt teste thrusterne så fort som mulig (Ottar).
- Linn fortalte om endring av designinspirasjon fra krabbe til hummer. Per Erik er positiv til endringen.
- Ottar: prøv å bruke en spenning som ligger tett opp til det man skal bruke – lag strømbudsjett. Lurt å bruke batterier til å shave topper i strømforbruket (feks når man skrur på thrusterne). Ifølge veiledere kan man slippe å bruke step-down converter dersom man velger batteriløsning som går opp, bruk en spenning som ligger tett opp til det man skal bruke. Lurt å bruke batterier til å shave «topper» i forbruket.
- Viktig at beina kan foldes inn under kroppen ved heving av ROven. Ottar etterlyser begrunnelser for hvordan beina beveges. Må ha fleksibilitet kun der det trengs og ikke ellers! Trenger man at hvert ledd og ben kan ha bevegelse uavhengig av hverandre? Mest sannsynlig ikke.

- Per Erik viser senkekamera og video fra Solnør. Forteller at de har 6 volts batteripakker på båten. Ønsker at vi bygger en modell fremfor fungerende prototype.

# MØTEREFERAT

## Produkt- og Systemdesign

Tid: 4. april, kl 09.00-10.00

Sted: NTNU i Ålesund

- Viser foreløpig design, diskuterer plassering av kamera. To kameraer back-to-back vil gi mye sikt uten noe leamikk. Frontkameraet har en vidvinkel som gjelder både opp og ned.
- Per Erik mente å plasttrekke to kuler ville vært det beste dersom man går for kuledesignet, fordi slik vil man kunne gjøre den gjennomsiktig gjennom hele kula.
- En kule er god fordi man ikke har noen plane flater som krefter kan virke på.
- For å oppnå stor stabilitet må det være mest mulig avstand mellom COG og COB. Tenk mer over fasongen på ROV, hent inspirasjon fra mikro-ROVer og mikro-AUV
- Ottar spør om frihetsgrader til beina. Christian D sier at det blir for mange servoer dersom vi har tre fremfor to. Det har med lengden på beina å gjøre. Ottar lurer på om bevegelsen til beinet på et løst underlag kan være problematisk i og med friksjonen vil medføre at beinet ikke sklir ut av posisjon. Han viser til å bruke stag i beinet som løfter beinet opp før det gir av rotasjonen. Ottar nevner å bruke fagverk i beina. Han er generelt bekymret på lengden fordi det skal ikke mye til for at den står fast foran en stein eller lignende. Da kan man utnytte dette til å minke motstanden på beina.
- Per Erik nevnte at hans design ville være å bruke formen til en amerikansk fotball som inspirasjon. Derfor utvide kulene med sylindere mellom, sier Ottar. Det medfører en enklere måte å gi god kameravinkel. Per Erik nevner da å minke høyden og ikke bredden. Da får den en sylinderform.
- Ottar nevnte at å bruke olje inne i ROV som lar oss fjerne både varme og oppdrift.
- Odroid kan skape problem. De mener å bytte ut Odroid. Martin vil ha rundtgående aluminiumsform til Odroid. Dette er løst ved å bruke en hoved-kjøleribbe som skal plukke opp den varme luften i konstruksjonen. Robin lurer på om Rasperry PI og Arduino bruker vifte, hvorfor ikke bare bruke disse komponentene hvis ikke? Martin disputerer med dekode-prosess. Automasjon må bruke mye krefter for å klare dekodingen av høyoppløsning av film og derfor fungerer Odroid.
- Per Erik sier at Kystverket bruker kun den kvaliteten de får (og ikke lager eller komponerer selv) for å inspisere o.l. Det er kun i Oslofjorden at sjøen havner over 20 grader. Innsiden av glasset vil ha samme temperatur som på utsiden; dette vil fungere som en egen kjøler.
- Rough/Tough PC blir brukt av Militæret. Disse er vifteløse av en grunn.
- Twins joined
- Bruke aluminiumsrør og plastkupper i hjørnet.
- Varmeledningsevne til akryl er 0.2, hvor aluminium har 2000.
- Christian D viser videoen som Per Erik har sendt oss. Per Erik mener at videokvaliteten på denne filmen er god nok for Kystverket til å inspirere med, for når ROVen kommer i ro, vil også kameraet stabilisere seg.
- Anders viser fram MATLAB-sekvensen og forteller om signaloverføringen mellom Maestro-kortet, kontrolleren og servoene.
- Christian H forteller, fremfor å vise (manglet lader), om hvordan bildestabiliseringen fungerer.



- Martin viser frem spenningsgrafene og forteller om dens verdier.
- Angående spørsmålet «kan det hende at gyrostabilisatoren ikke er kjapp nok?» Ottar sier at det må ha en hensikt dersom den skal rette opp bevegelsen.
- Siste kommentar fra Per Erik: Han vil at vi skal tenke på nytt angående design. Vårt design bruker beinbevegelsen til en hummer og ikke krabben, som vi fester på kulen. En kule er en god form ettersom den ikke har noen plane flater som bølgekrefter kan virke på. For å oppnå god stabilitet må man ha størst mulig avstand mellom COG (center of gravity) og COB (center of buoyancy).
- Gå en runde til på kjøleproblemet og unngå bruk av vifte (Ottar).

# MØTEREFERAT

## Produkt- og Systemdesign

Tid: 18. april, kl 09.00-10.00

Sted: NTNU i Ålesund

- Ottar ønsker å se en simulering av beina i kartesisk koordinatsystem. Det fungerer i matlab, men gjør det det ellers også? Koble opp servoer i en mockup for å se at bevegelsen blir riktig.
- Bruk av bilder i bacheloroppgave: holder det å oppgi kilder, eller må man kontakte hver enkelt nettside og be om lov? Svar fra veiledere: Er det lagt ut på nettet kan man bruke bildet (da dette er ute i offentligheten) så lenge man oppgir kilden.
- Skal vi oppgi dimensjoner i rapporten (som størrelser på bein/skall) eller er det nok å med maskintegningene? Kan det være lurt med oppsummering i tabell og maskintegninger i tillegg? Det ville Ottar ha gjort, men kun hoveddimensjonene som bredde, lengde osv, volum og vekt.
- Er det noen av veilederne som har motforestillinger mot å bruke rapportmalen beskrevet av Nils Olsson fra NTNU? (Vi fikk opplæring i denne) Svar: Målet er å formidle et budskap – så lenge man får dette fram går det helt fint. Om det betyr å slå sammen resultat og metode så gjør det. Man kan bruke rapportmalen på Fronter som forklarer hva som burde være i hvert kapittel. Under teoretisk grunnlag burde man ikke ha med noe annet enn hva metodikken går gjennom – inkluder gjerne framgangsmåte i dette kapittelet. Man kan forklare de ulike emnene under terminologi. *Ottar forventer ca 100 sider*
- Angående karakterskille, vil Ottar at det må være snakk om to karakterer.
- Bruk av eksempelkoder: Bare å skrive at det ble hentet resurser fra referansen med et par klammeparenteser. Så lenge de er referert, er det ikke juks.
- Anders viser fram bein i Java og forklarer dens bevegelse med kontroll. Ottar ønsker å se selve beinet i bevegelse, da kanskje i form for simulering. Han vil se at disse bevegelsene stemmer når de overføres til det kartesiske koordinatsystemet (XYZ) slik at Jakobian-matrisen fungerer.
- Christian H viser fram kompensasjonen for kameraet (stabiliseringen) og forklarer problemet han har i programmeringene og selve videooverføringen. Per Erik sier at dersom gruppen får til kamerastabiliseringen slik som vist i dag, er vi allerede på god vei. Han ønsker å se mest mulig av bildet kontinuerlig, samt ha et så stort synsfelt som mulig. Han er og litt usikker på lysfølsomheten: LED i sirkel (IR-lys)
- Paul Steffen påpeker at FABLab har tre GoPro-kamera som gruppen kan etterlyse om å låne.
- Martin viser frem både kontinuerlige og spenningsberegningene, som effektivitet (med de verst mulige tilfellene)
- Per Erik konstaterer at komponentene skaper høy varme
- Ottar begynner å snakke om oppdriften og volumet. Per Erik svarer med at vi har plassert komponentene hensiktsmessig. Stephen sier at når det er snakk om volumet, er 8.5 liter ekskludert selve røret i midten og at maks innvendig volum kanskje ligger på 3 til 4 liter. Dermed kan ikke gruppen øke bredden og høyden til tilsvarende volum (i hensyn til oppdriften). Veilederne skjønner problemet. Per Erik tenker på Honeycomb-struktur gjennom Simplify3D som lar gruppen legge til

ekstra oppdrift inni selve skallet fremfor å bruke 100% infill i 3D-printen. Ottar mener vi burde diskutere vekt og manøvrering – drøfte om thrusterne i det hele tatt vil fungere under sjøen. Per Erik sier at slik er det bare – prosjekt kan feile og forteller om hans prosjekt som en anekdote.

- Det er snakk om komponentene og vekt – nytt tema tatt opp av Per Erik. Per Erik nevner og at gruppen burde se på de største komponentene og ikke bare øke størrelsen. Christian D viser fram excel-dokumentet med vekt for hver komponent i og forklarer om gruppens komponenter. Dersom gruppen ikke kan gjøre noe med komponentene, sier Per Erik, må gruppen kun øke volumet i skallet og at man i tillegg burde sitte i lag og drøfte problemstillingen
- Ottar nevner at servoer kan ikke bruke for tynne ledninger, for da vil servoene ende opp med å skjelve. Det diskuteres om spenningene til servo, thruster og andre komponenter
- Til neste gang, burde vi se på volumet og oppdriften.
- Martin tok over for Morten og snakket om oppløsning og FPS. Ottar lurer på om gruppen har lagt opp til å kunne justere FPS selv, slik at dersom man vil ta kun bilder, så vil man kunne bruke alternativet som gir høyest oppløsning på bekostning av FPS. Martin svarer med at man skal kunne sende ned en kommando som gjør nettopp dette. Per Erik ønsker maks tre oppløsninger og en snap-funksjon.
- Stephen viste fram skallet og forklarte dette.
- Ottar sier at man ikke burde tenke for mye på manøvrering i sjøen, at den heller kan kjøre den vertikalt og ned igjen for å komme seg gjennom tareskoger og lignende. Det er også mulig å øke lengden. Per Erik mener at man burde bygge i høyden. Thrusterne er ganske store – Per Erik sier at vannet er et ganske tykt medium og så store thrustere er langt ifra nødvendig
- Kamerakuppel – Lexan (polycarbonate) – kan printe i dette materialet
- Christian D viste fram pakninga som er skjært med laserkutteren. Dette så bra ut.
- Christian D snakket om beina.

# MØTEREFERAT

## Produkt- og Systemdesign

Tid: 2. mai, kl 09.00-10.00

Sted: NTNU i Ålesund

- Christian H viser frem og forklarer korrigeringen. Han påpeker at koden ikke er tilpasset kameraet, men at med riktig kamera vil koden fungere som programmert. Per Erik sier at man må se an hvordan «delay» vil oppleves når sjøen beveger seg.
- Anders viser frem og forklarer bein-bevegelsen (prototype). Det oppstod litt problemer på grunn av servoene.
- Robin påpeker at det er positivt at Anders fant en feil i bevegelsen.
- Christian D viser frem det laserskuttete trebrettet av konstruksjonsgulvet.
- Christian D viser frem bilder av skroget. Stephen forklarer om skroget og oppdriften. Christian snakker om kamerakuppelen som er Resin-printet. Veilederne sier det ser bra ut.
- Christian D forklarer om braketten i skroget.
- Per Erik spør om oppdrift. Vi viser til oppdriften.

# Vedlegg 3

## Ukentlig logg

# Ukentlig logg

## Uke 1:

- Ble introdusert til prosjektgruppa. Diskuterte oppgaven og fikk formulert en liste med spørsmål til kontaktpersonen fra Kystverket, Per Erik Ose.

## Uke 2:

- Første veiledermøte. Diskuterte oppgaven og fikk svar på de mest overordnede spørsmål vi hadde til den.
- Begynte å diskutere designet av ROV-en i plenum; hvilken form som er mest motstandsdyktig mot bølger.
- Så litt på mekanikk- og styrkeberegning. Studerte fagverk i mekanikkboka.
- Undersøkte forholdstallet mellom de ulike leddene i beina til den europeiske strandkrabben.
- Diskuterte bruk av vaierløsning til å styre beina, samt designet på ROV-en med utgangspunkt i hva som passer til bein inspirert av en krabbe.
- Arbeid med forprosjektrapporten.

## Uke 3:

- Arbeid med forprosjektrapporten.
- Samarbeid med automasjon om prototyping av bein.
- Utarbeidet møteagenda.

## Uke 4:

- Veiledermøte.
- Frisket opp kunnskap om bølgeteori (som effekt og slagkraft fra en bølge), oppdriftsberegninger og litt statistikk (standardavvik) fra forelesningsnotater.
- Undersøkte thrusterplassering og bredde-, lengde- og høydeforhold ved å studere moderne ROV-er.
- Studerte skrog, spesielt formen på kajaker og om de kan ha noe for seg når vi skal utforme vår ROV.
- Så på dimensjonene til skroget. Basert på hva automasjonsstudentene har bestemt av komponenter, er thrustere og en vanntett boks til elektronikken dimensjonerende for skrogets størrelse.
- Undersøkte ulike typer materialer. Aluminium med 2.5% magnesium er optimalt for sjømiljø, da det er billigere enn rustfritt stål med grad 316. Vi burde bruke 3D printede materialer det det er mulig.
- Opprettet Excel-fil til å legge inn komponenter og pris, slik at vi kan lage et budsjett.
- Arbeid med forprosjektrapporten.

### Uke 5:

- Undersøkte ulike typer vaier som kan brukes som innhentningsvaier.
- Automasjonsstudentene har tatt et valg angående servoer, denne ble modellert i NX slik at man får en oversikt over størrelsen.
- Vurderte hvordan håndtaket skal se ut og dets plassering.
- Tegnet første utkast av ROV-skall i NX med grov-estimerte dimensjoner basert på de elektriske komponentene som skal få plass. Diskuterte hvordan tyngdepunktet ville være på ROV med tanke på bein, thrusterne og innmat.
- Arbeid med å utvikle bein. Gravemaskininspirert prototype.
- Diskuterte beinas design på nytt, etter å ha sett første utkast av sammenstillingstegning. Konstruksjonen oppleves som litt ustabil bare basert på det visuelle inntrykket.
- Printet prototype av bein.
- Diskuterte budsjett.
- Arbeid med rapporten.
- Utarbeidet møteagenda.

### Uke 6:

- Møte med veiledere og Per Erik Ose.
- Videreutviklet skroget i NX. Ny designavgjørelse: plassere thrusterne på utsiden av ROV-ens skrog. Står da friere til å plassere elektriske komponenter.
- Begynte å tenke på luftkapsel som skal hjelpe å styre oppdriften til ROV slik at den kan få nøytral oppdrift. Fant ut at denne må vente til slutt ettersom at vi trenger å vite selve vekten på ROV, vekten av skallet til ROV (som kommer sent i prosjektet), og vekten på resterende komponenter.
- 3D-printet ny prototype av beinet. Designet er inspirert av krabbebein.
- Kontaktet Jostein Berge angående om hjelp til eventuell støping av skroget.
- Arbeid med rapporten.

### Uke 7:

- Undersøkte plassering av thrusterne.
- Modellerte forbedringer på skroget i NX samt lagde en assembly til å vise på veiledermøte.
- Diskuterte produksjonsmetode av skroget med Paul Steffen Kleppe. Kan 3D-printe skallet og buke det som negativ støpeform.
- Diskuterte med Ola Jon Mork om å vurdere industrielle komponenter til bruk i beina.
- 3D-printet skroget i nedskalert versjon.
- Undersøkte teoretisk oppdrift til ROV.
- Videre arbeid med beina.
- Arbeid med rapporten.

### Uke 8:

- Fullskala 3D-printing av skroget.
- Undersøkte om Karl Henning Halse kunne utføre en bølgesimulering på skroget.
- Undersøkte om Lars Petter Bryne kunne frese ut skroget i 1:1. Dette sa han seg villig til å gjøre dersom vi eksporterte en parasolid-fil til ham.
- Diskuterte nytt skrogdesign. Ønskelig å gjøre skroget 100% symmetrisk.
- Designendringer på skroget: gjorde det smalere. Ny thrusterplassering. Modellerte feste til disse.
- Undersøkte plassering av elektronikken.
- Lagde en serovstasjon med «trappedesign».
- Vurderte ulike løsninger til kamerahus og dets plassering.
- Mailutveksling med Per Erik Ose.
- Utarbeidet møteagenda.
- Arbeid med rapporten.

### Uke 9:

- Møte med veiledere. Veilederene oppfordrer studentene fra automasjon til å gå over til en likestrømsløsning. Dette tok studentene til seg.
- Foretok mål på den innvendige boksen som skal inneholde all elektronikk. Målte opp og tegnet de forskjellige komponentene i NX til el-boksen. Den er foreløpig svært stor. Ønske om å gå over til likestrøm slik at komponentene kan bli mindre.
- Forsøkt å implementere designendringer basert på veiledning fra Per Erik Ose.
- Videreutvikling av beina.
- Så på skolens OpenROV-kit og dens kameraløsning.
- Undersøkte en levende og en død krabbe til inspirasjon.

### Uke 10:

- Likestrøm fremfor vekselstrøm gjør at vi kan gjøre store og grunnleggende designendringer på ROV-en. Forsøk på å designe et nytt krabbeinspirert design med mindre dimensjoner feilet. Muligens en bedre strategi å tenke helt nytt. Designet må likevel ha minst mulig slagareal og få spisse kanter (være strømlinjeformet).
- Vurderte nye inspirasjonskilder (hummer, kråkebolle).
- Undersøkte hydraulikk til å bevege beina. Løsningen virker foreløpig svært dyr.
- Undersøkte batteri som strømforsyning. OpenROV bruker batterier og har en levetid på 2-3 timer.
- Drøftet ulike løsninger til beina. Videreutviklet 3D-modeller av dem.
- Mailutveksling med Per Erik Ose.
- Arbeid med rapporten.
- Utarbeidet møteagenda.



#### **Uke 11:**

- Møte med veilederne.
- Skisserte nytt forslag til skrog i NX (kuledesign).
- Vurderte det nye designet basert på hydrodynamikk (drøftet ulemper og fordeler).
- Et viktig hensyn å ta er plasseringen av kameraet, ettersom det er ROV-ens viktigste oppgave å ta gode bilder.
- Drøftet å bruke annen propell eller å beholde den vi har og designe rundt den.
- Elektronikken trenger ikke lenger ligge inne i en boks, noe som gir en friere designprosess.

#### **Uke 12:**

- Utarbeidet et nytt designforslag basert på en hummer.
- Arbeid med rapporten.

#### **Uke 13:**

- Møte med Per Erik Ose hos Kystverket. Diskuterte bruksituasjon og arbeidsområde grundigere.
- Satte intern frist for å lage et siste skrogdesign.
- Utarbeidet møteagenda.
- Arbeid med rapporten

#### **Uke 14:**

- Møte med veilederne.
- Ekstra møte med Paul Steffen Kleppe.
- Møte med Per Erik Ose på Kystverket.
- Bestemte plassering av innvendige elektriske komponenter samt grunnleggende aspekter ved designet, som størrelse og utforming.
- Besøkte Atlanterhavsparken for å se på hummere.
- Lagde forslag til håndtak til ROV.

#### **Uke 15:**

- Laget ny assembly i NX
- Begynt på maskintegninger og FEM-analyser i NX.
- Arbeid med rapporten.

#### **Uke 16:**

- Modellert flere detaljer på ROV-en i NX
- Mailutveksling med Per Erik Ose.
- Utførte noen oppdriftsberegninger.
- Arbeid med rapporten.

#### **Uke 17:**

- Modellert flere detaljer på ROV-en i NX
- Sett på skruer til kamerafestet, thrustere, deksel osv.
- Utførte flere oppdriftsberegninger

#### **Uke 18:**

- Bestemte tetninger og pakninger.
- Fastslo hvilke skruer som skulle benyttes. Endret på skruehullene i skroget (deksel og underdel), flensene til kjøleplatene, kamerafestet og thrusterfestet.
- Arbeidet med FEM-analyser
- Arbeid med rapporten.

#### **Uke 19:**

- Mailutveksling med Per Erik Ose.
- Arbeid med å lage NX-animasjon av gangart til ROV og QR-koder.
- Arbeid med rapporten.

#### **Uke 20:**

- Møte med Per Erik Ose.
- Fortsatte arbeidet med å lage NX-animasjon av gangart til ROV.
- Arbeid med å fremstille gode bilder av ROV-en ved bruk av Keyshot.
- Arbeid med rapporten.

#### **Uke 21:**

- Arbeid med presentasjon av hovedprosjekt.
- Arbeid med maskintegninger.
- Arbeid med rapporten.

#### **Uke 22:**

- Arbeid med rapporten.

# Vedlegg 4

## FEM-analyse i NX10.0

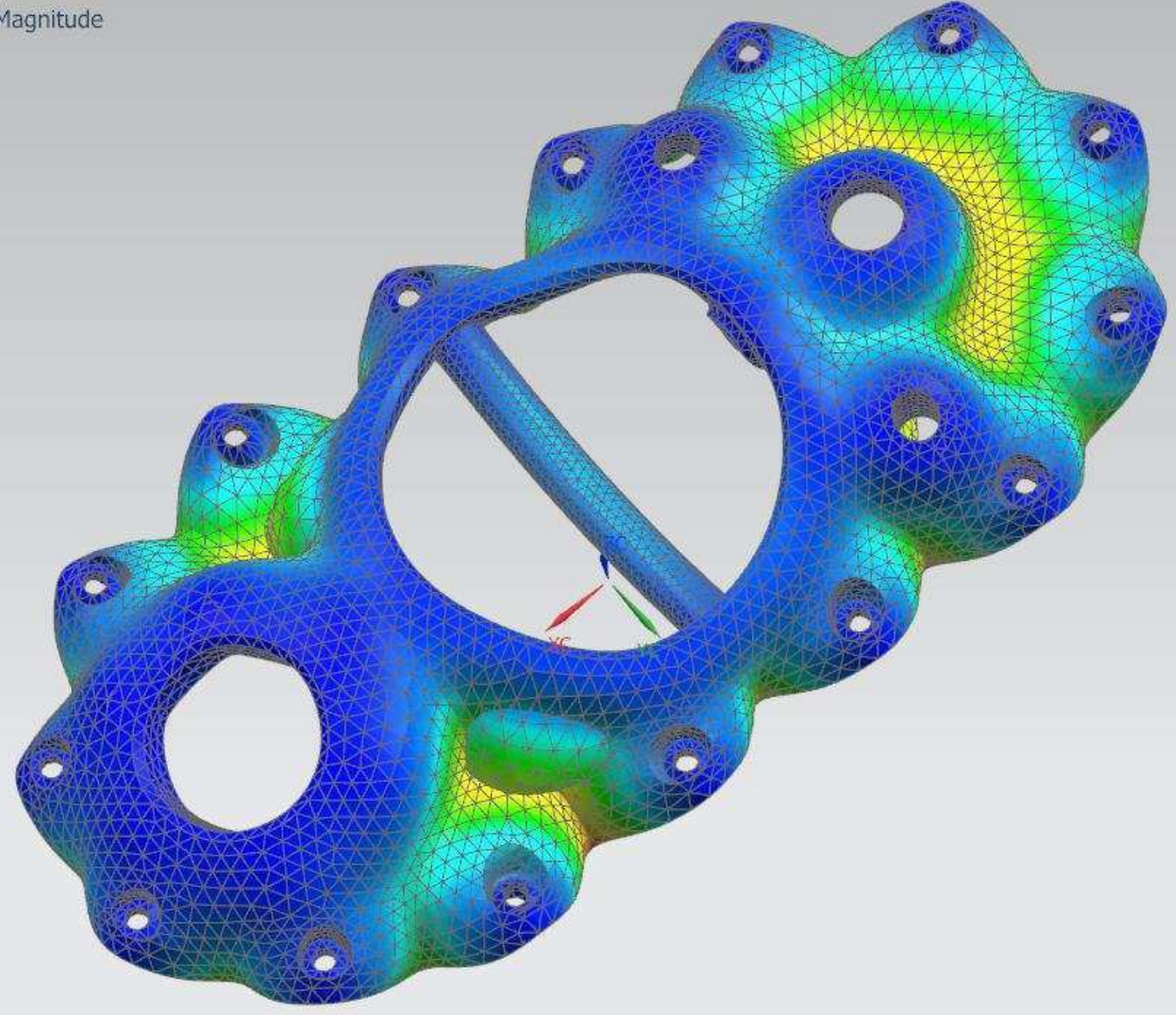
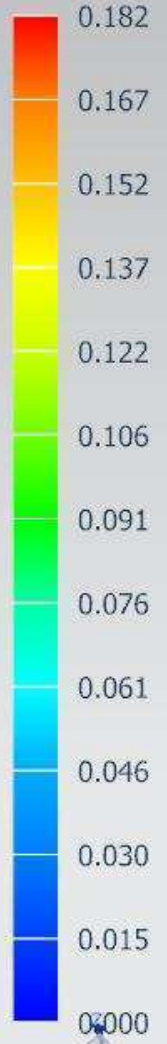
DEKSEL NYTT\_x\_t\_sim1 : Solution 1 Result

Subcase - Static Loads 1, Static Step 1

Displacement - Nodal, Magnitude

Min : 0.000, Max : 0.182, Units = mm

Deformation : Displacement - Nodal Magnitude



Units = mm

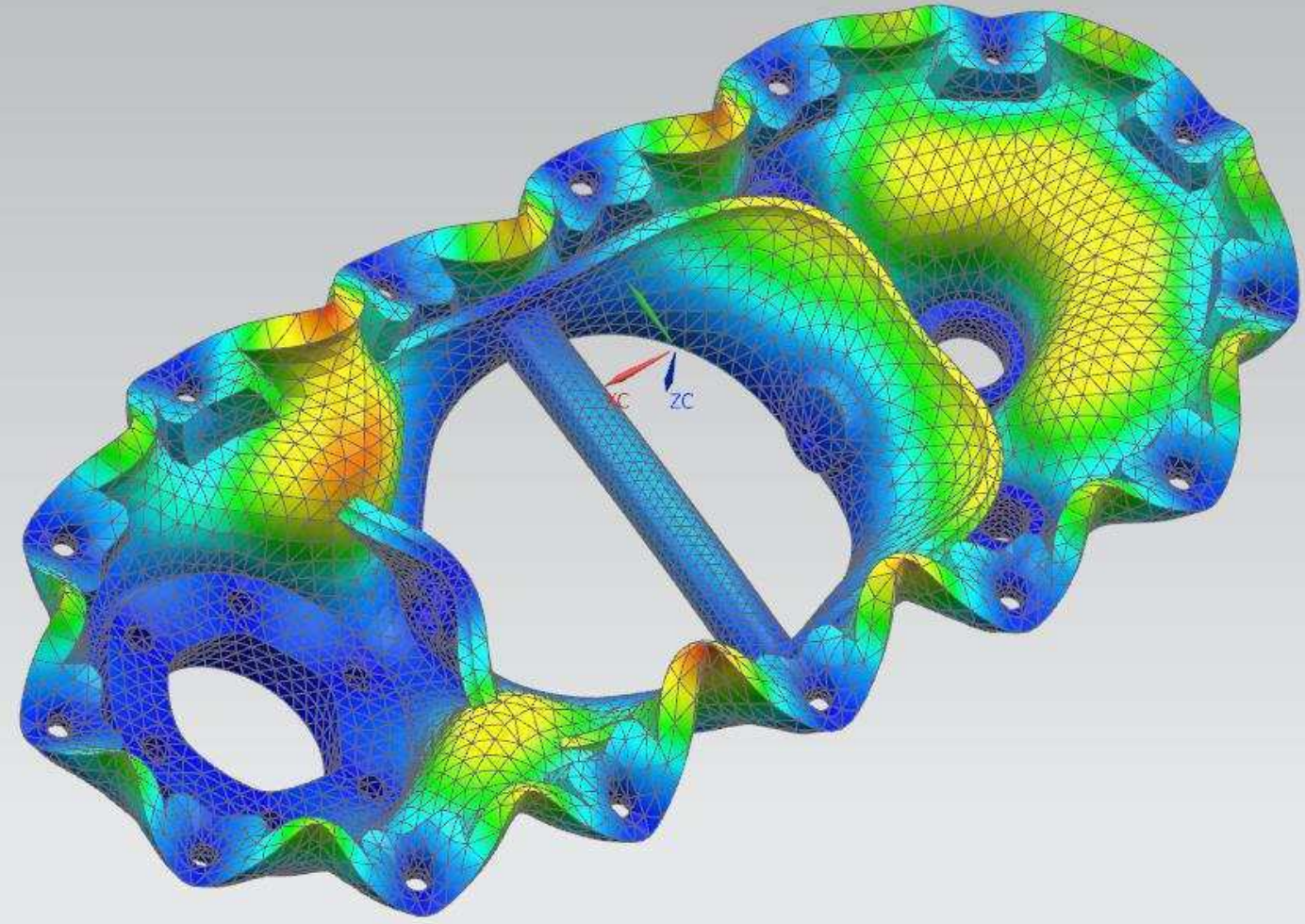
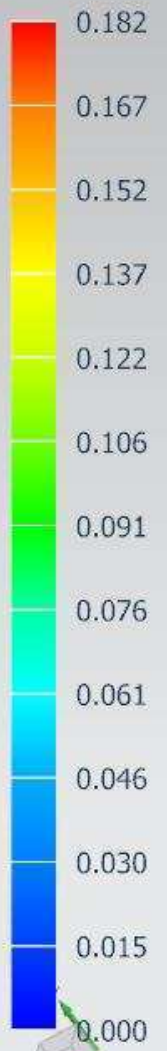
DEKSEL NYTT\_x\_t\_sim1 : Solution 1 Result

Subcase - Static Loads 1, Static Step 1

Displacement - Nodal, Magnitude

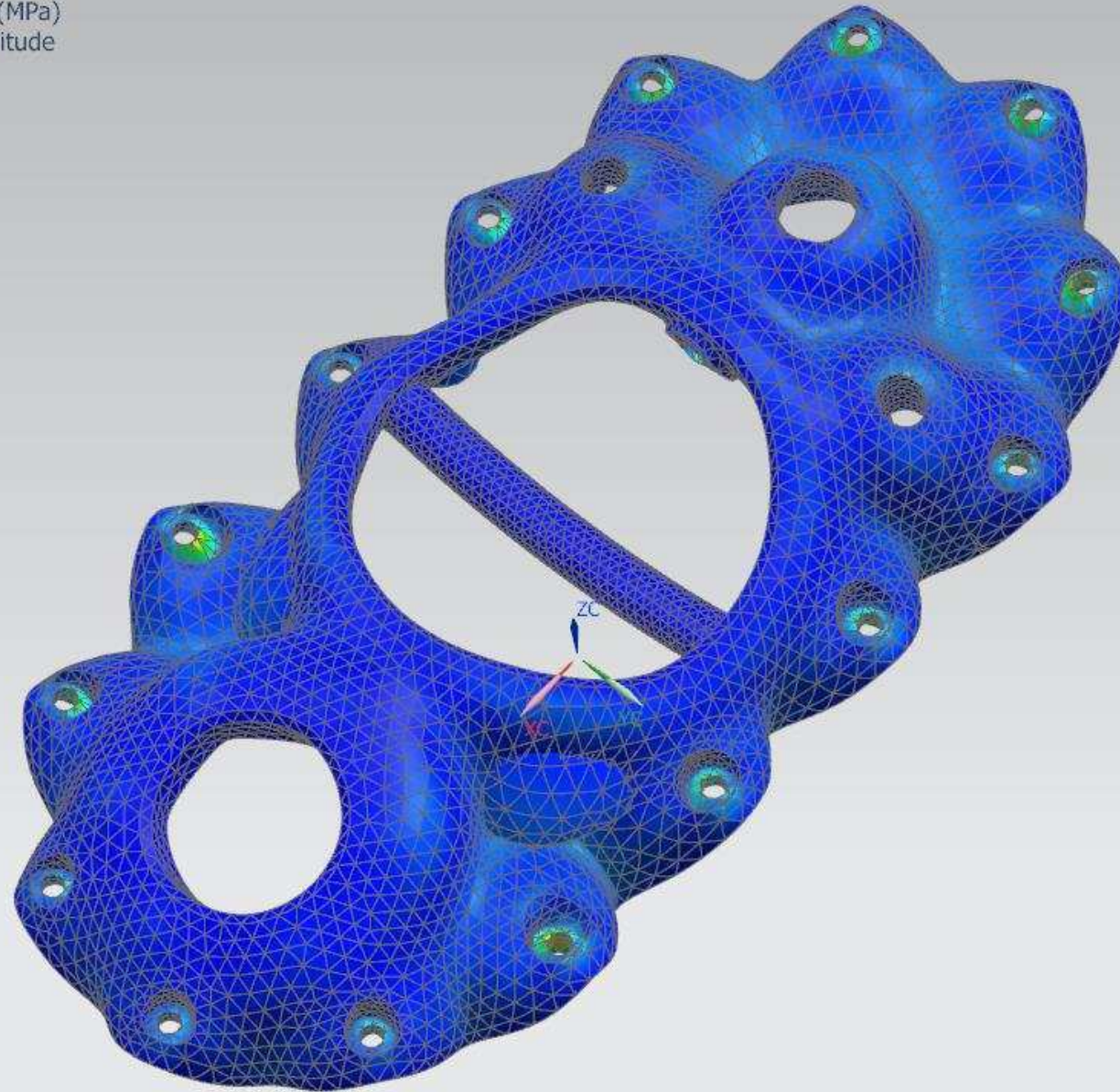
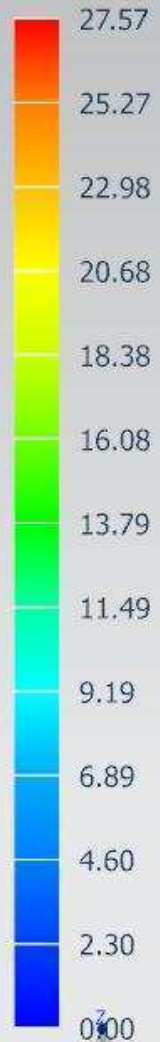
Min : 0.000, Max : 0.182, Units = mm

Deformation : Displacement - Nodal Magnitude



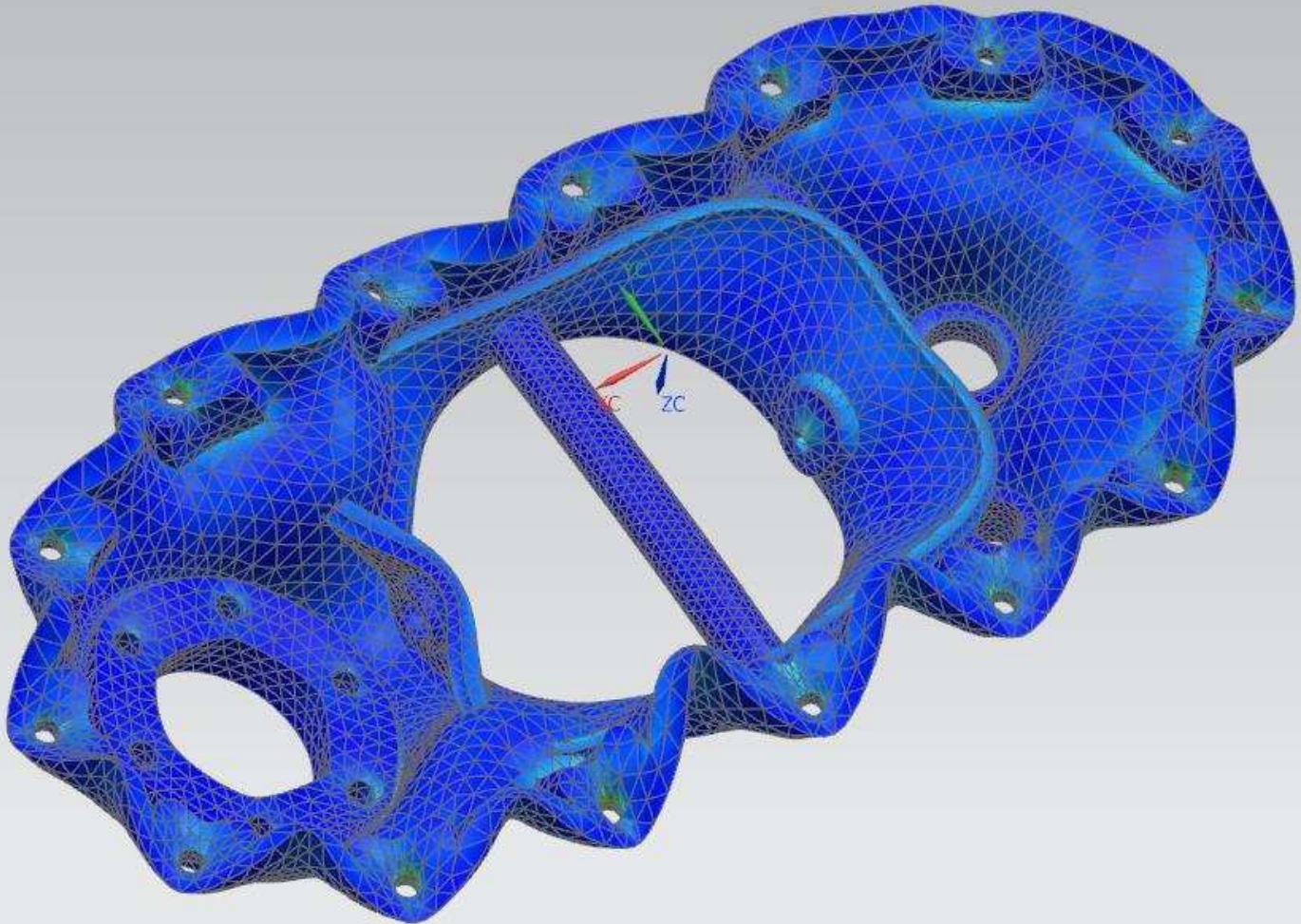
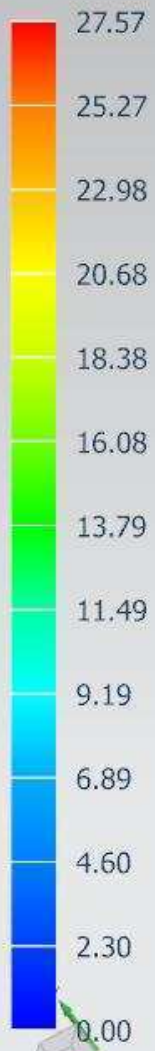


DEKSEL NYTT\_x\_t\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.00, Max : 27.57, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude



Units = N/mm<sup>2</sup>(MPa)

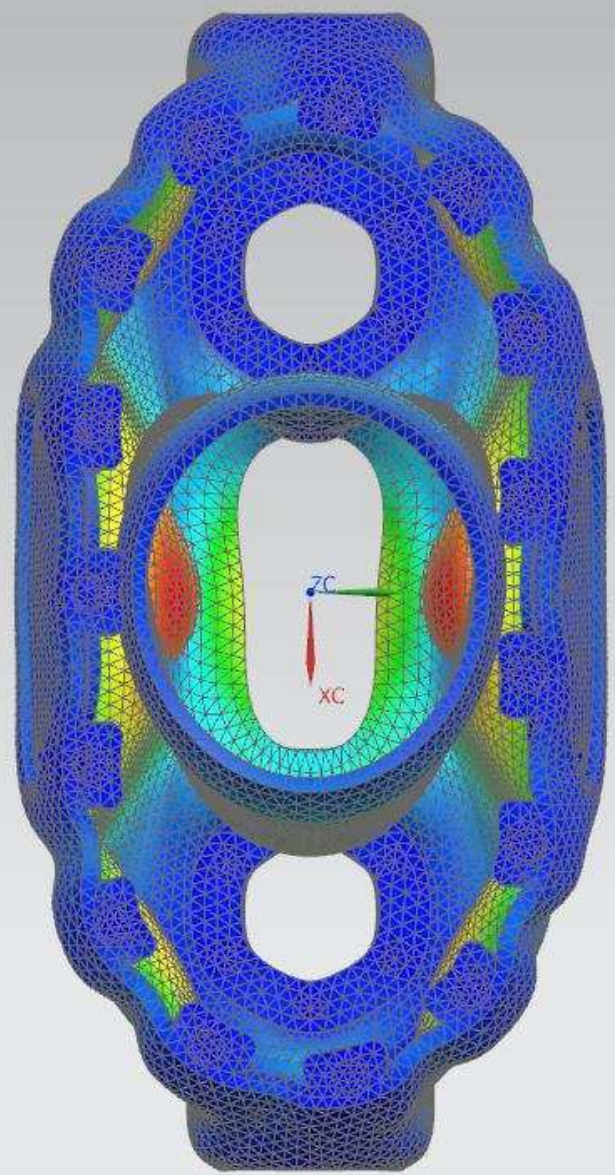
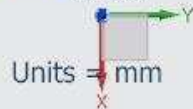
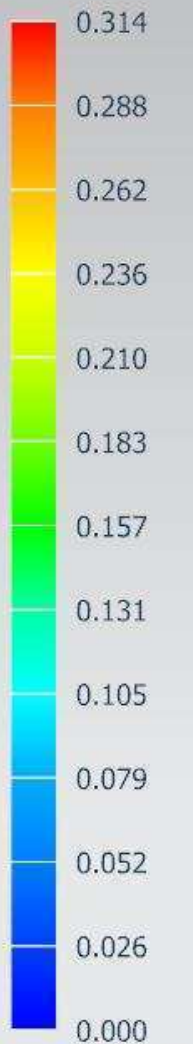
DEKSEL NYTT\_x\_t\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.00, Max : 27.57, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude



Units = N/mm<sup>2</sup>(MPa)

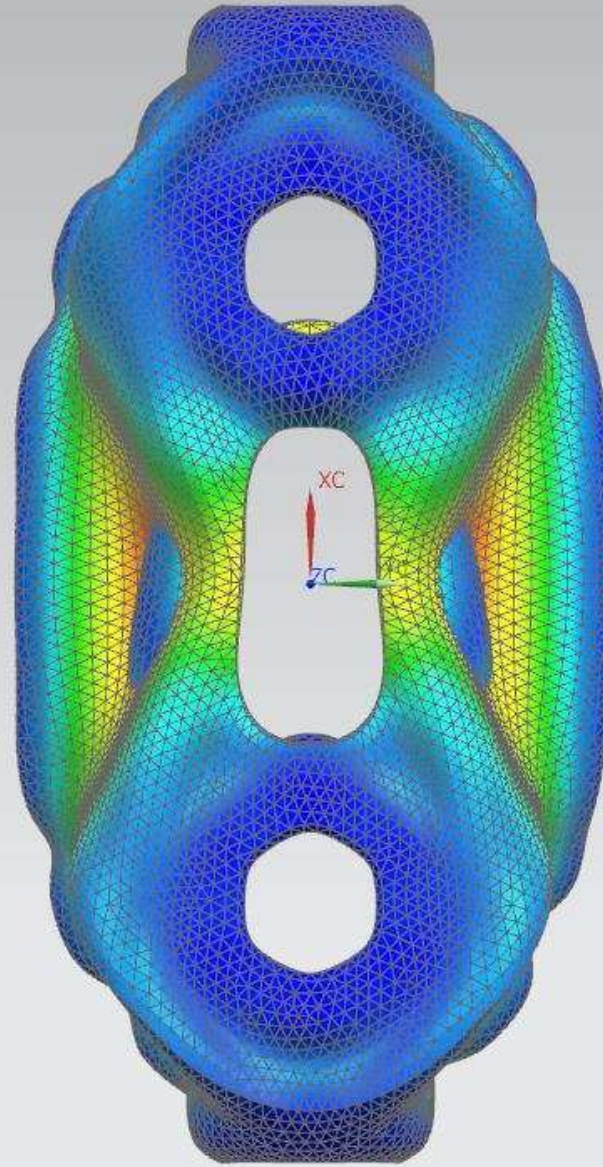
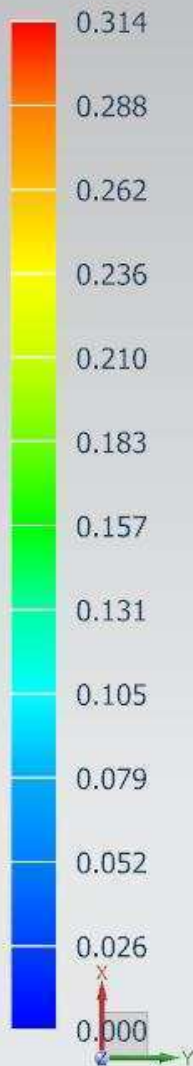


Skrog uten kamerastativ\_x\_t\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Displacement - Nodal, Magnitude  
Min : 0.000, Max : 0.314, Units = mm  
Deformation : Displacement - Nodal Magnitude



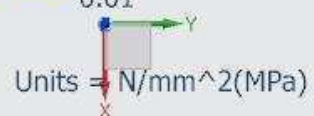
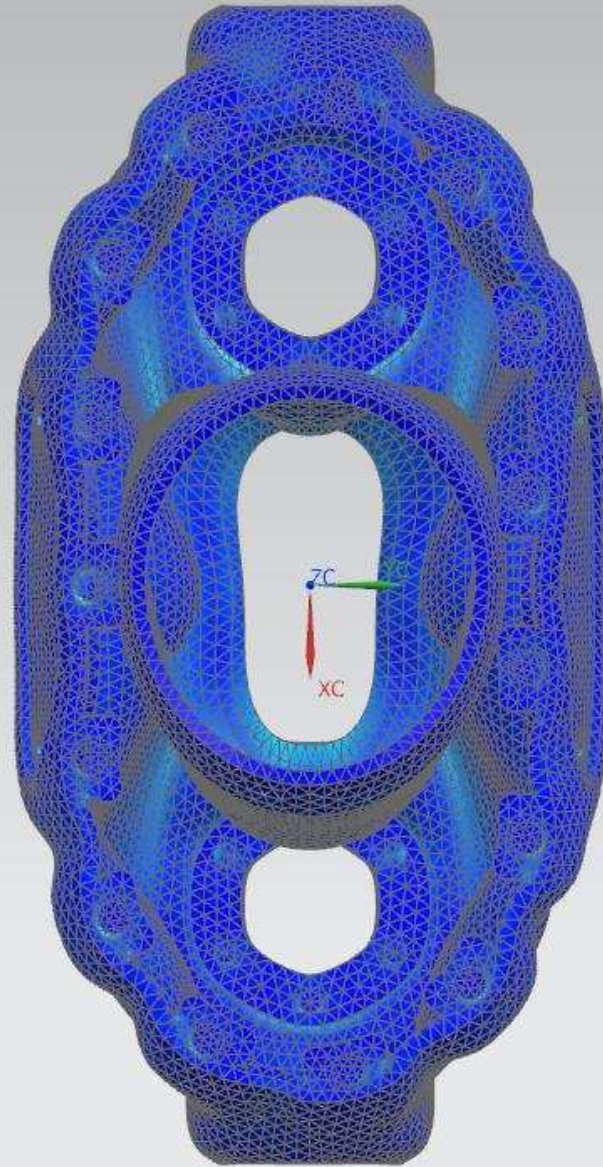
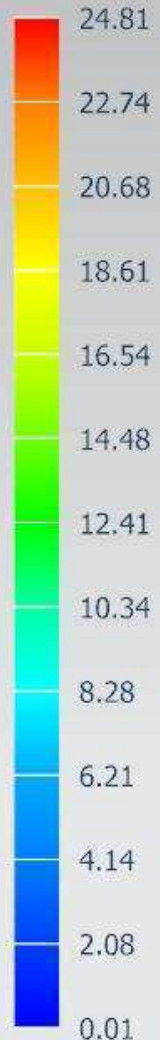


Skrog uten kamerastativ\_x\_t\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Displacement - Nodal, Magnitude  
Min : 0.000, Max : 0.314, Units = mm  
Deformation : Displacement - Nodal Magnitude



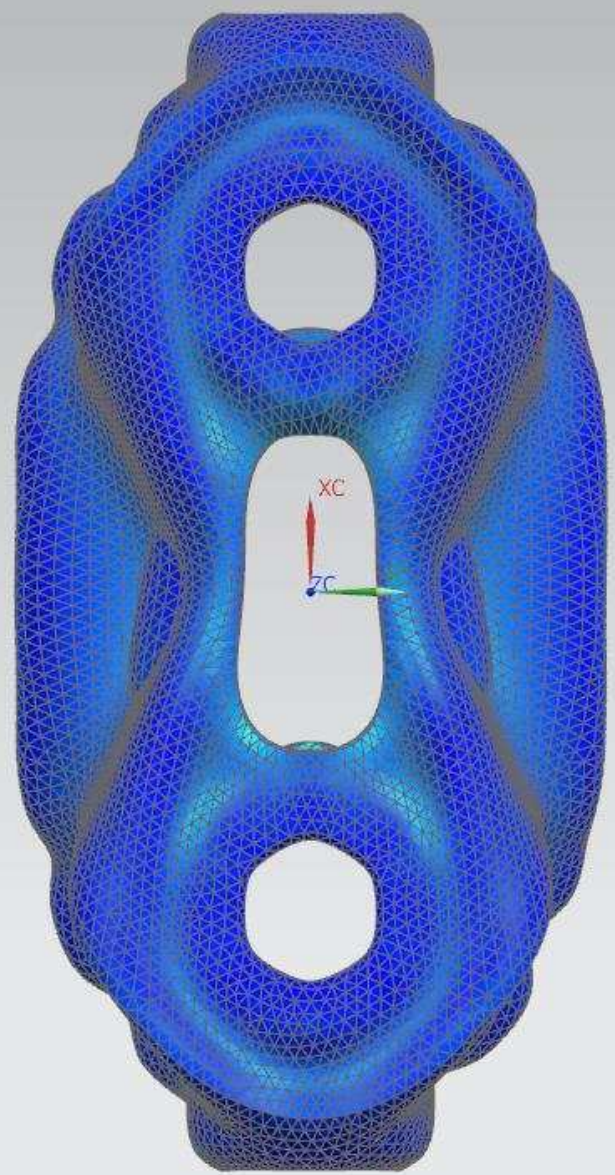
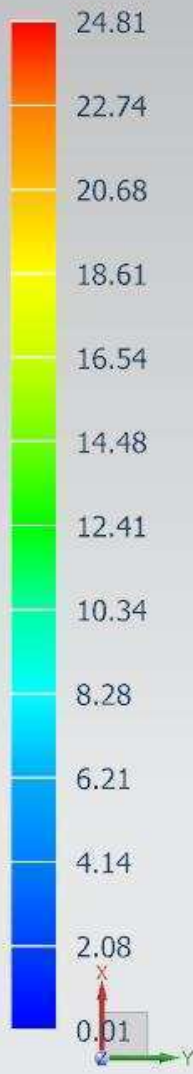
Units = mm

Skrog uten kamerastativ\_x\_t\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.01, Max : 24.81, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude



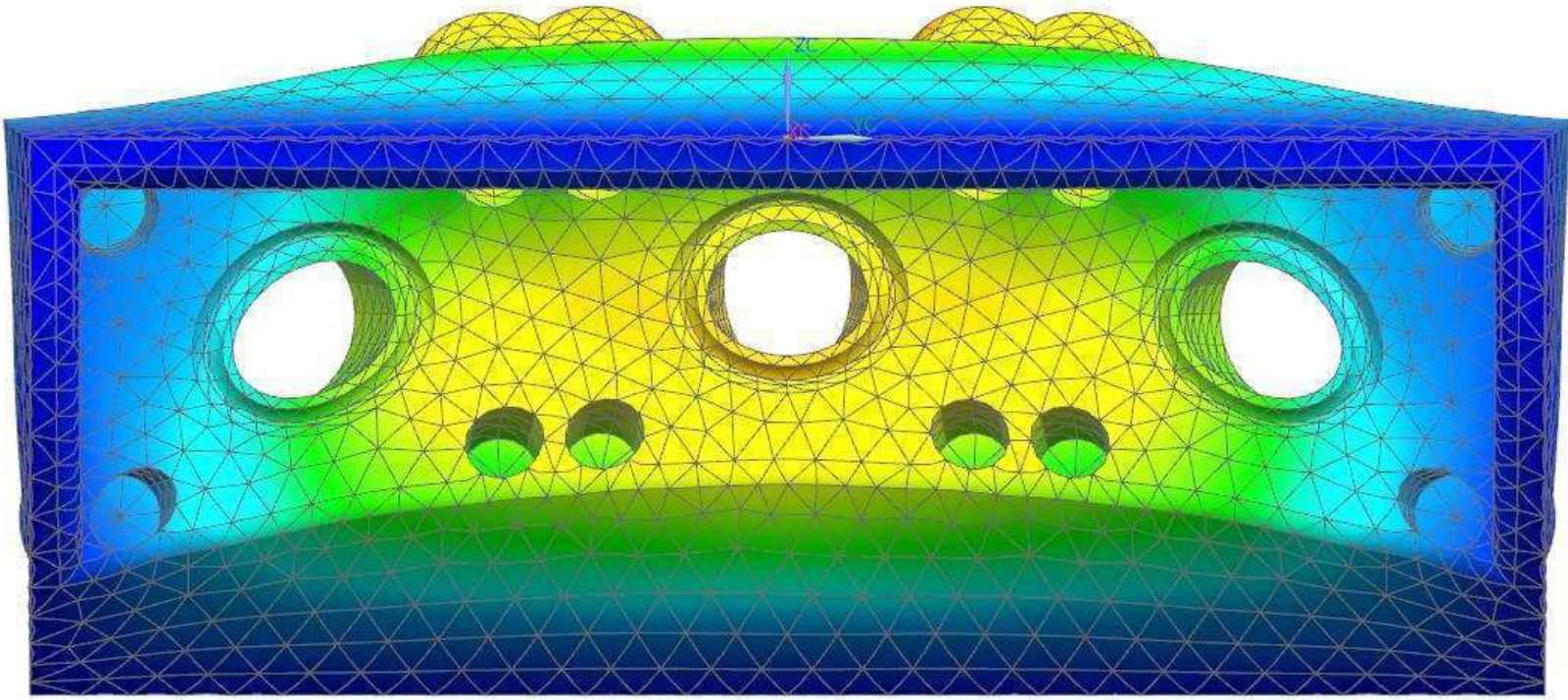
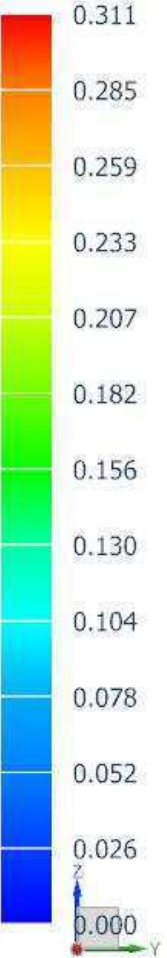


Skrog uten kamerastativ\_x\_t\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.01, Max : 24.81, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude



Units = N/mm<sup>2</sup>(MPa)

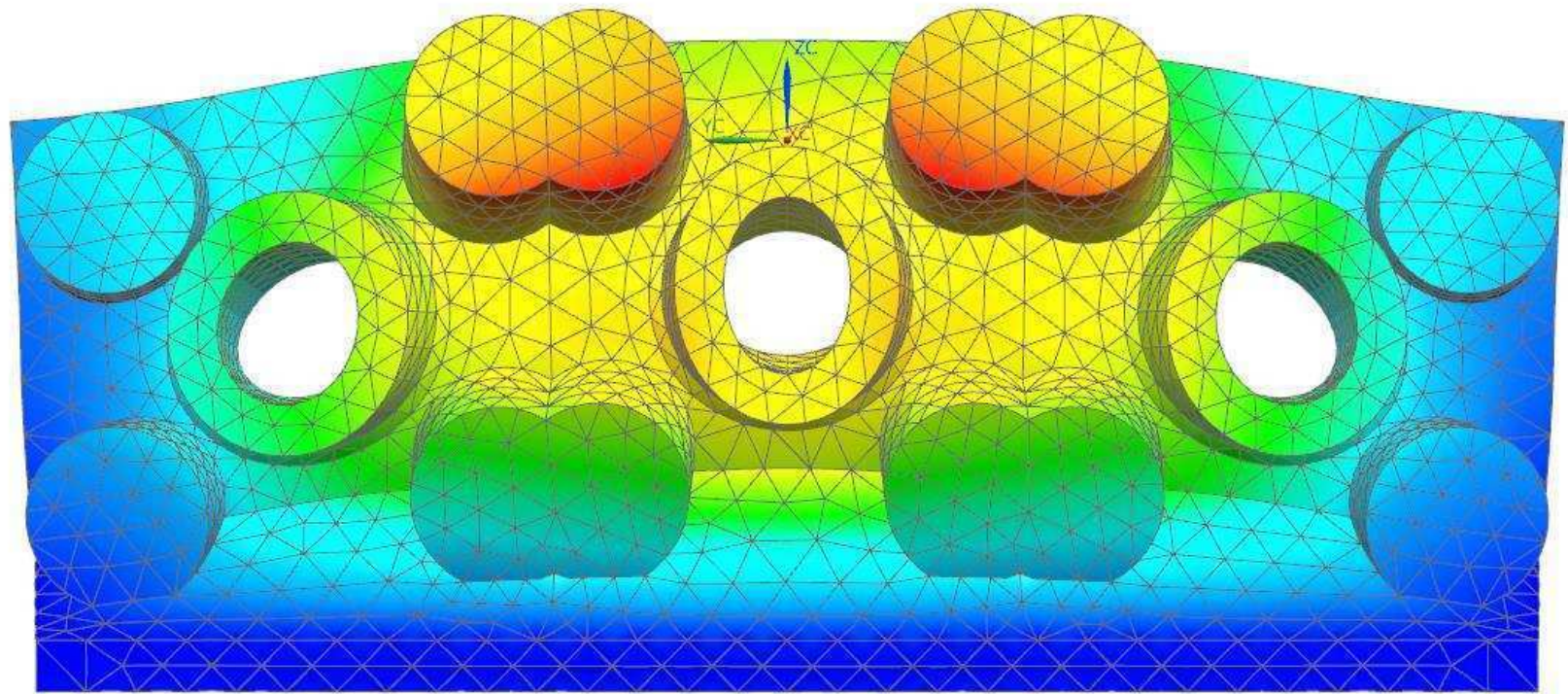
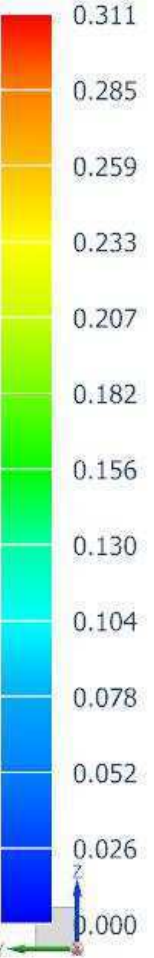
1 Connection Bein LEFT\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Displacement - Nodal, Magnitude  
Min : 0.000, Max : 0.311, Units = mm  
Deformation : Displacement - Nodal Magnitude



Units = mm



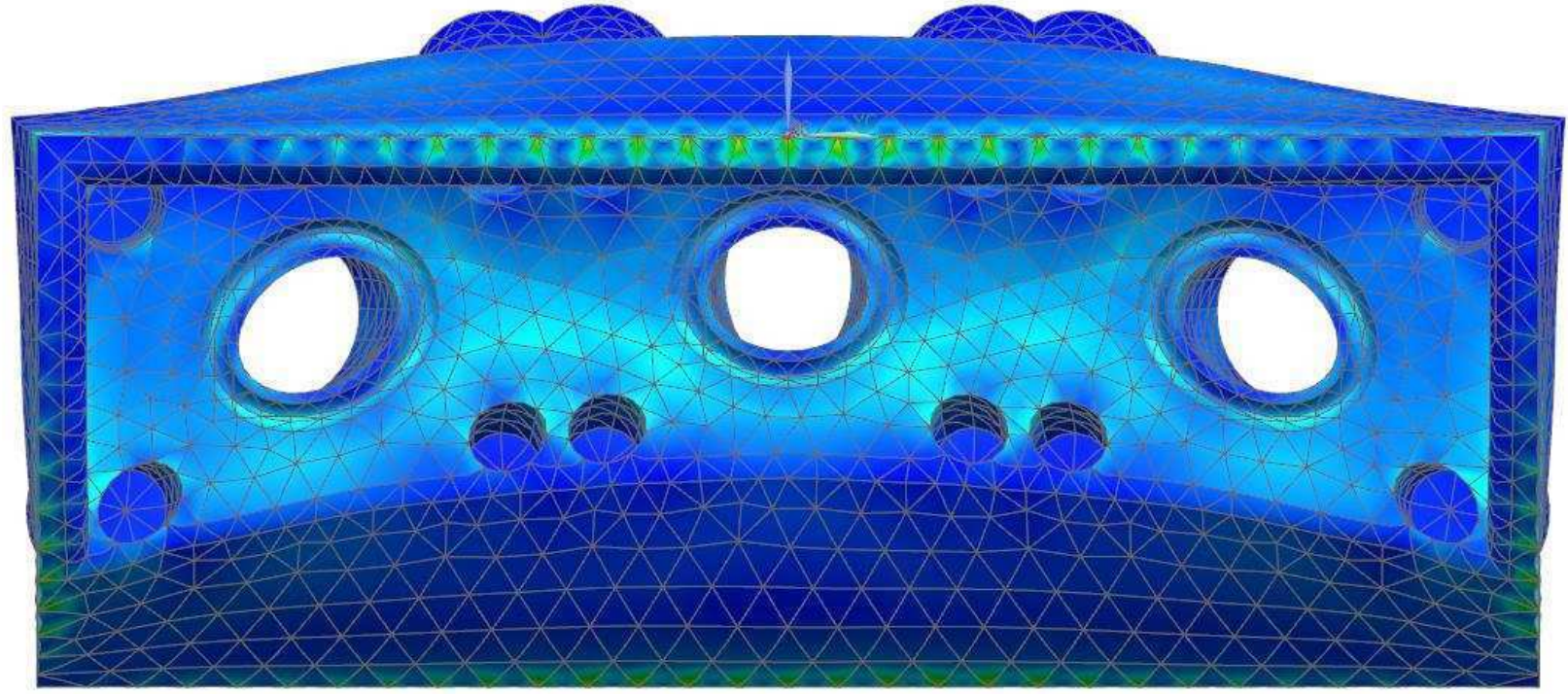
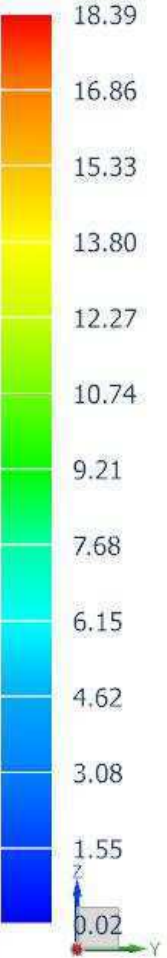
1 Connection Bein LEFT\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Displacement - Nodal, Magnitude  
Min : 0.000, Max : 0.311, Units = mm  
Deformation : Displacement - Nodal Magnitude



Units = mm



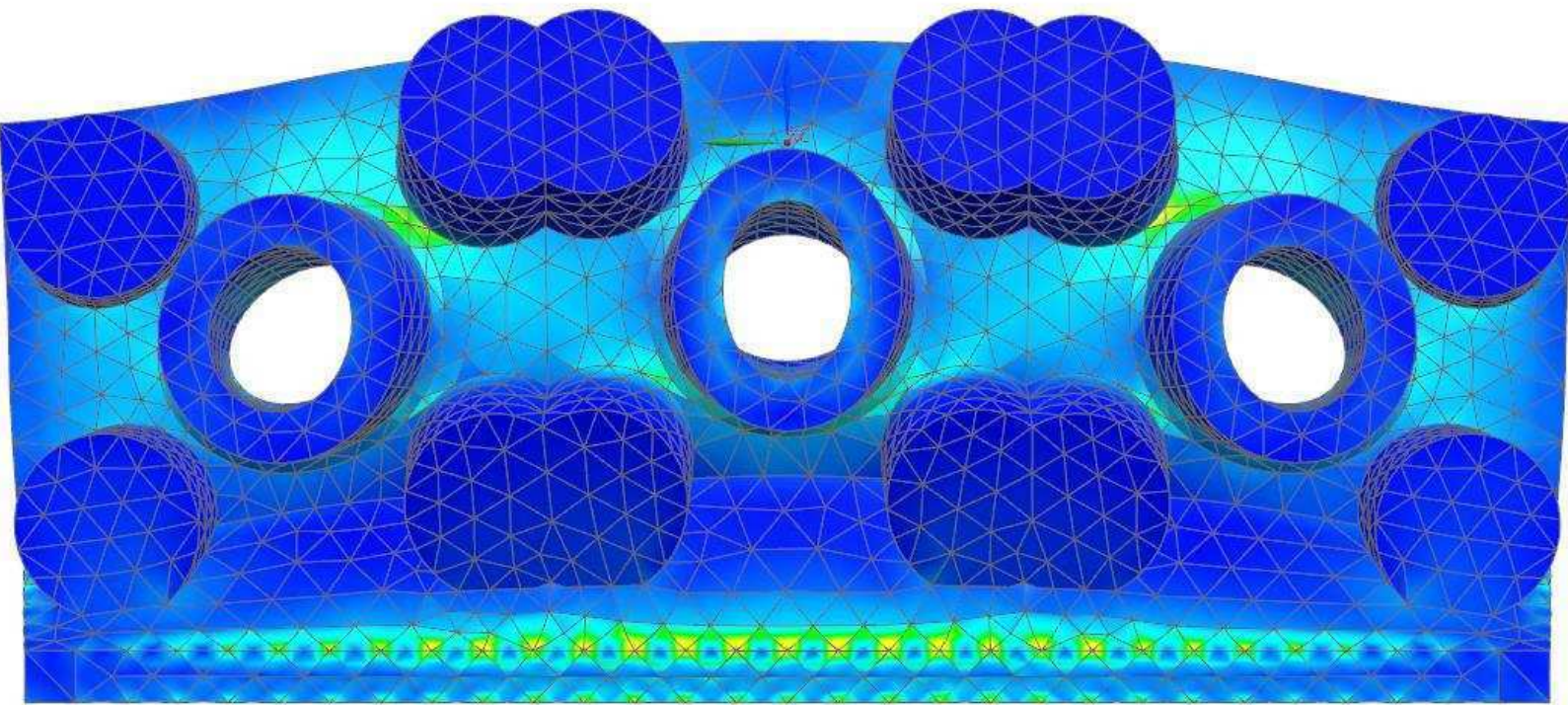
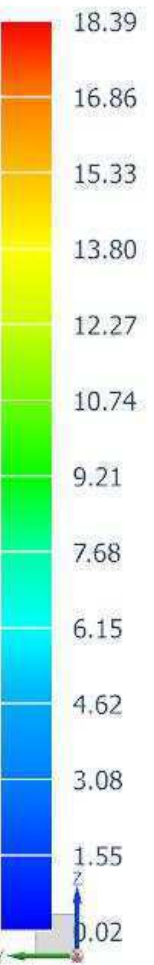
1 Connection Bein LEFT\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.02, Max : 18.39, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude



Units = N/mm<sup>2</sup>(MPa)

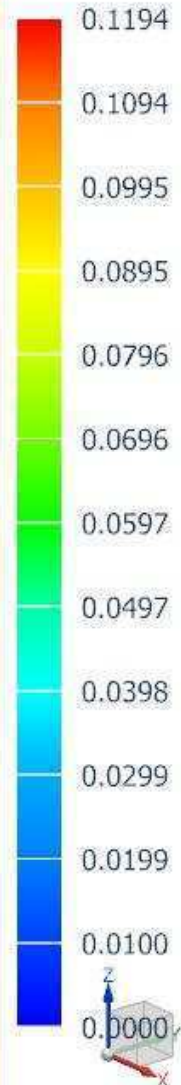


1 Connection Bein LEFT\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.02, Max : 18.39, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude

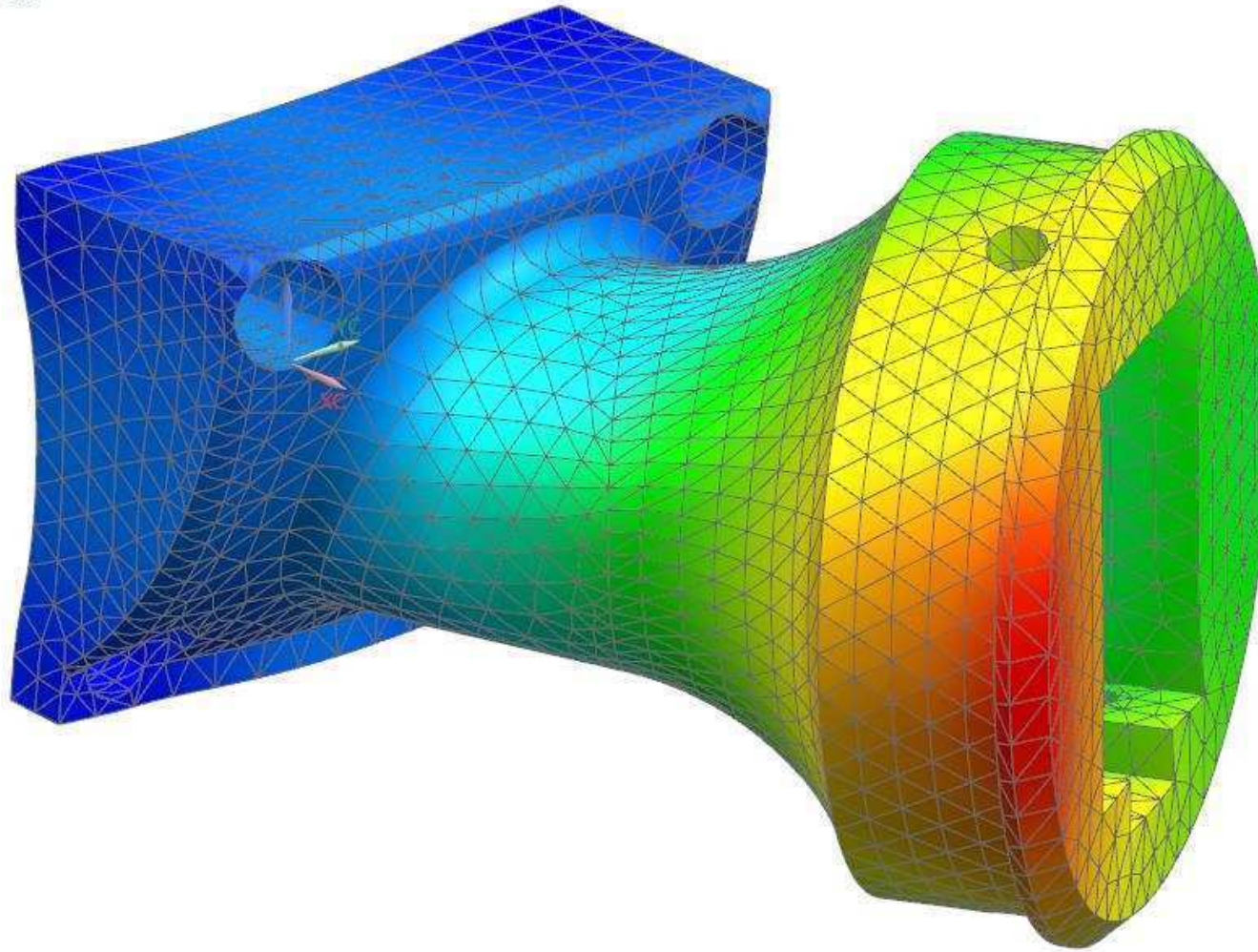


Units = N/mm<sup>2</sup>(MPa)

ROV SKALL fot ytre\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Displacement - Nodal, Magnitude  
Min : 0.0000, Max : 0.1194, Units = mm  
Deformation : Displacement - Nodal Magnitude  
Animation Frame 15 of 15

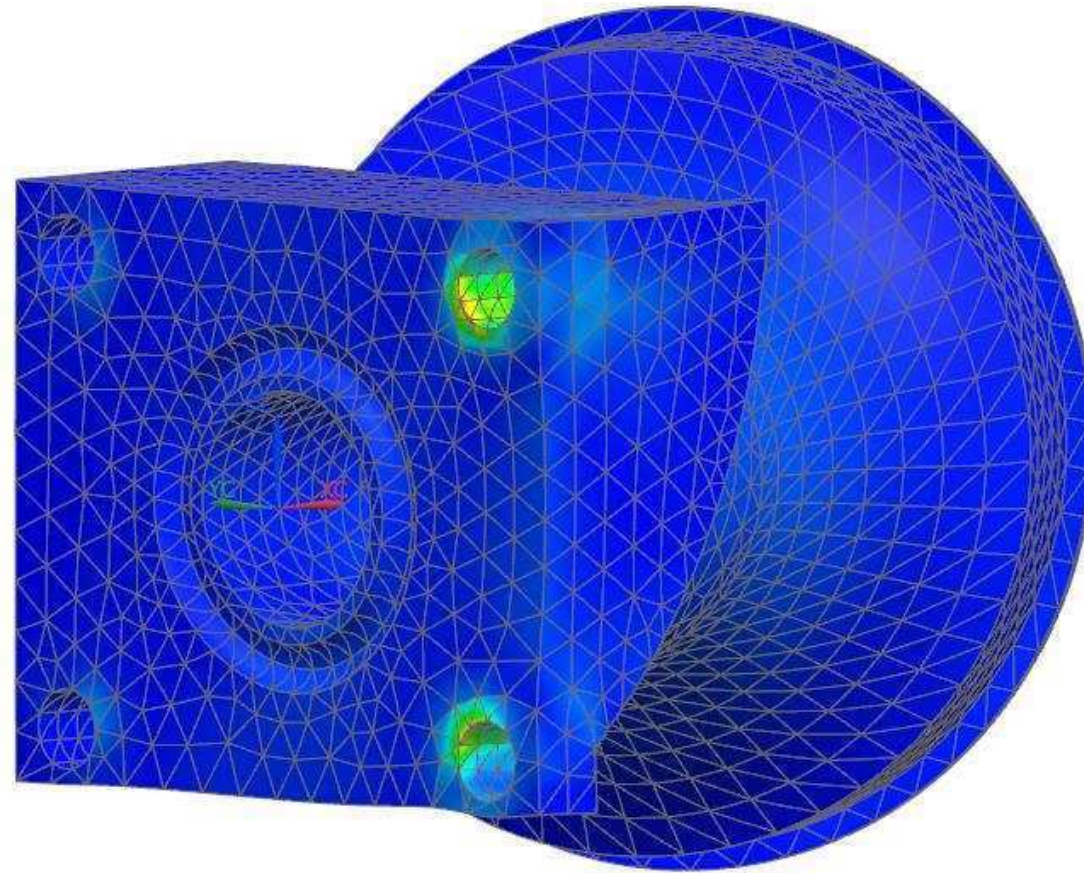


Units = mm



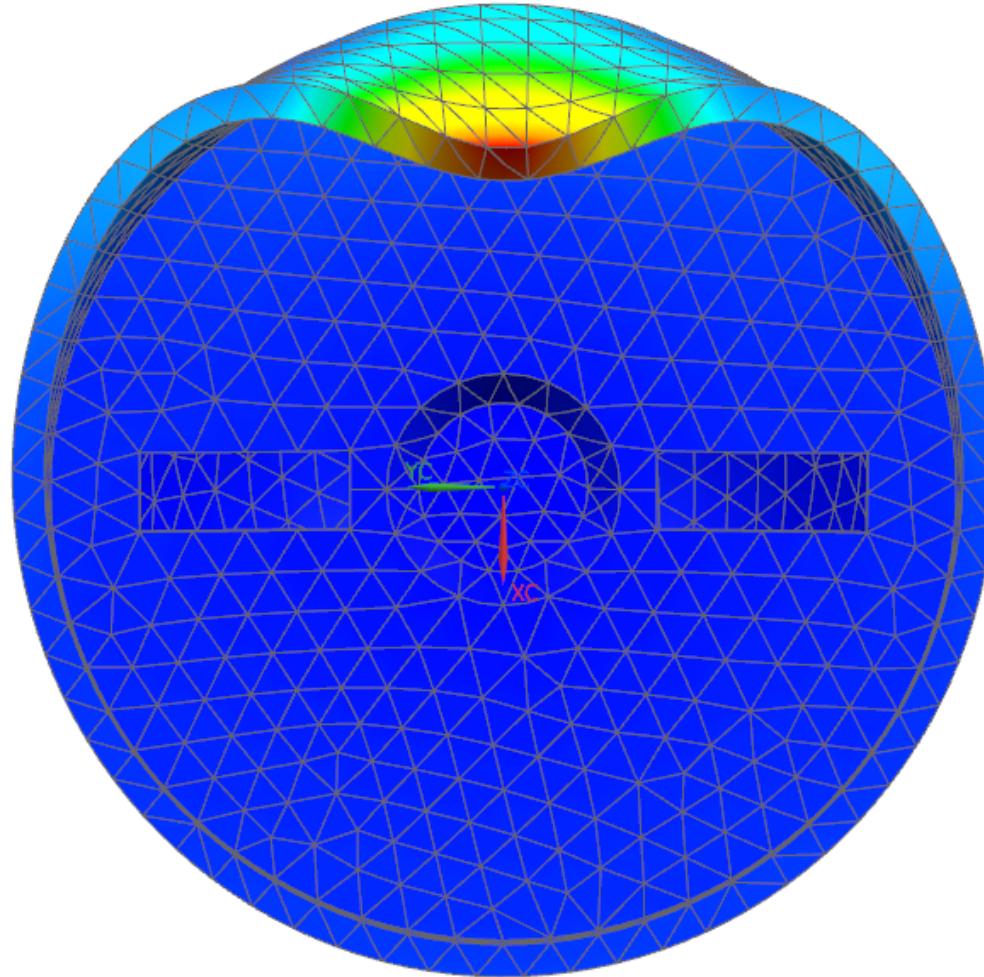


ROV SKALL fot ytre\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.01, Max : 14.70, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude  
Animation Frame 15 of 15



Units = N/mm<sup>2</sup>(MPa)

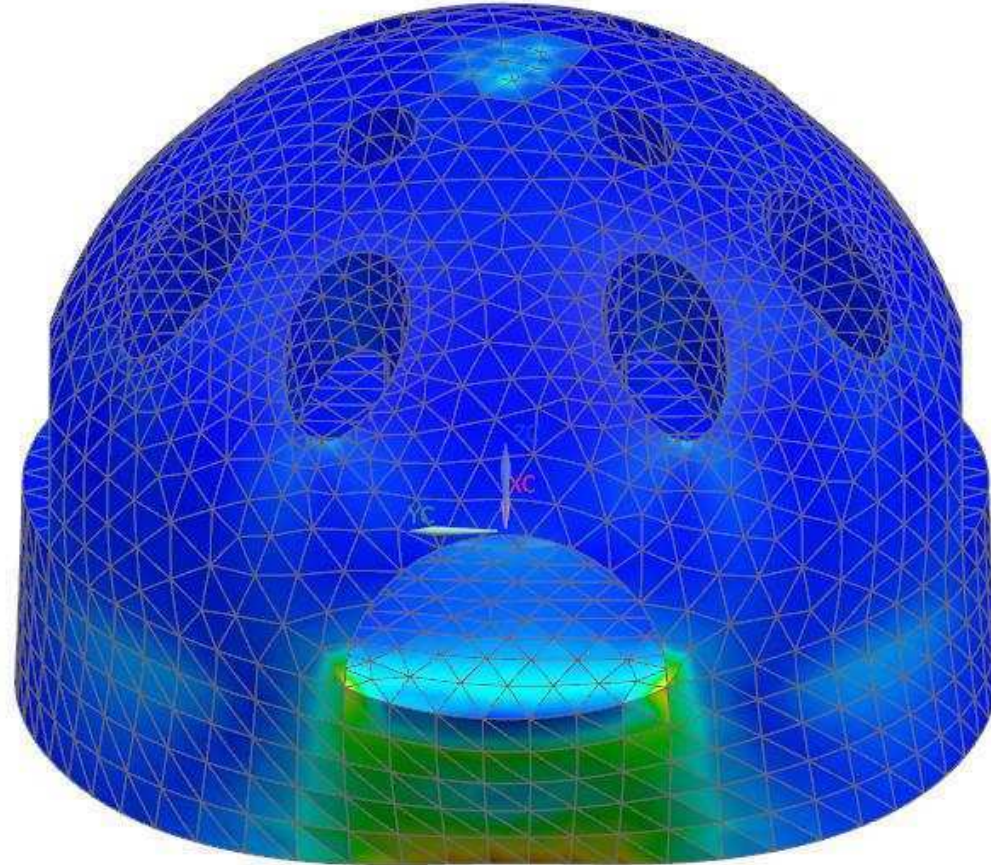
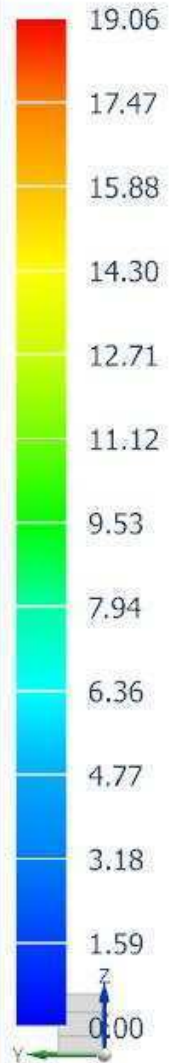
fort v1.0\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Displacement - Nodal, Magnitude  
Min : 0.000, Max : 0.480, Units = mm  
Deformation : Displacement - Nodal Magnitude  
Animation Frame 15 of 15



Units = mm

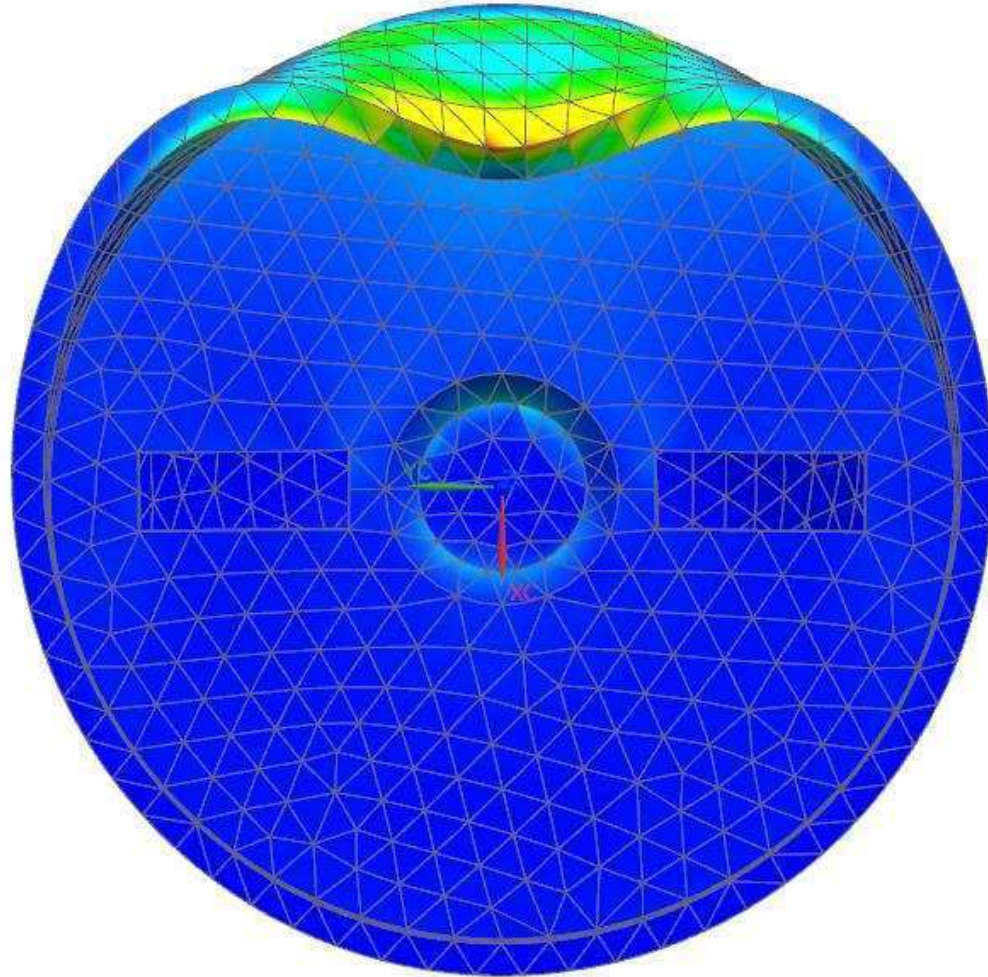
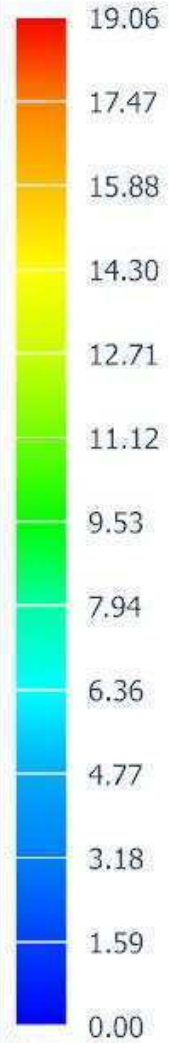


fort v1.0\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.00, Max : 19.06, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude  
Animation Frame 15 of 15



Units = N/mm<sup>2</sup>(MPa)

for v1.0\_sim1 : Solution 1 Result  
Subcase - Static Loads 1, Static Step 1  
Stress - Element-Nodal, Unaveraged, Von-Mises  
Min : 0.00, Max : 19.06, Units = N/mm<sup>2</sup>(MPa)  
Deformation : Displacement - Nodal Magnitude  
Animation Frame 15 of 15

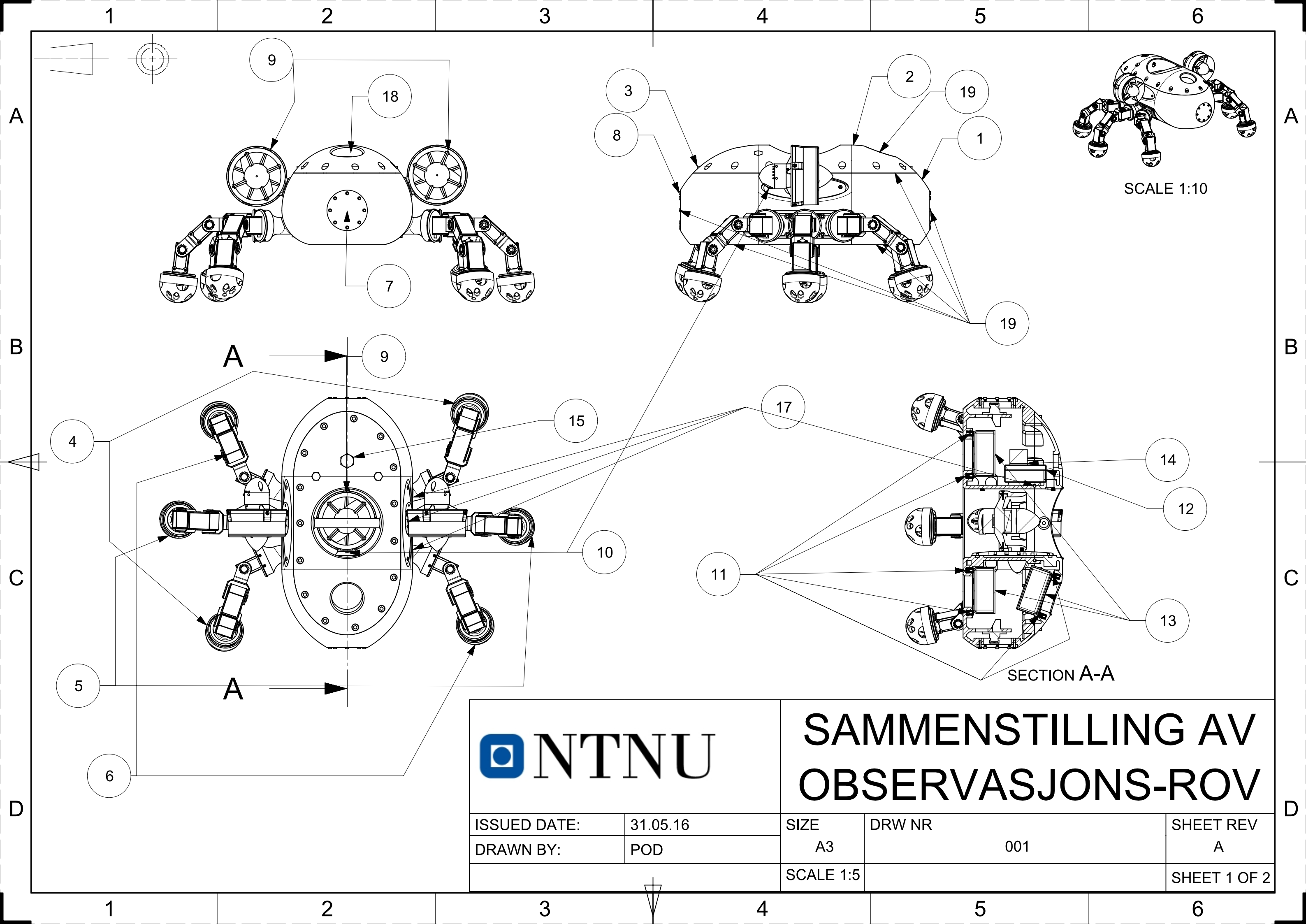


Units = N/mm<sup>2</sup>(MPa)

# Vedlegg 5

## Maskintegninger av observasjons-ROV





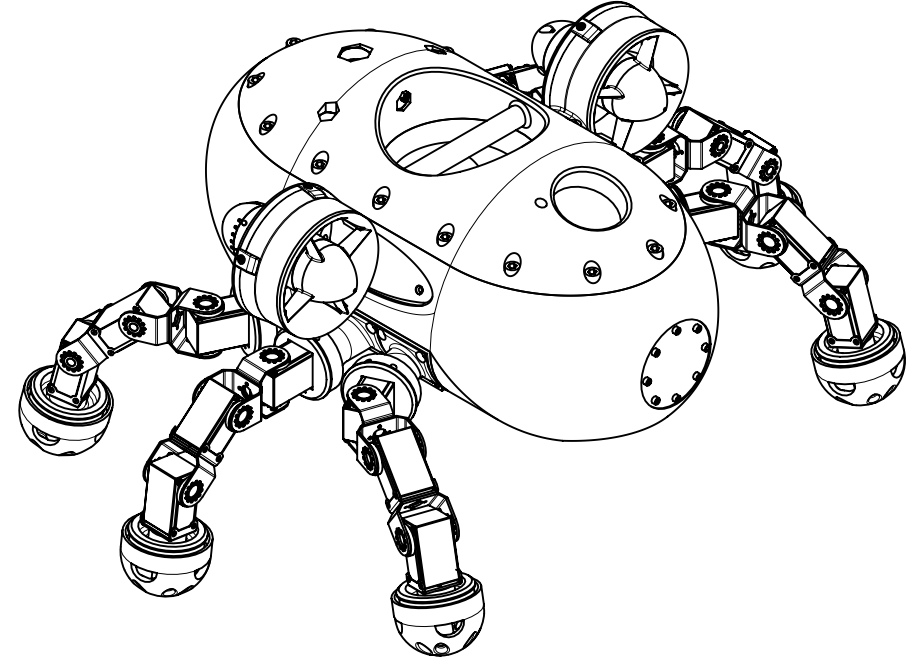
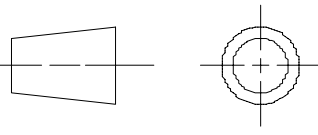
SCALE 1:10

SECTION A-A



# SAMMENSTILLING AV OBSERVASJONS-ROV

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	001	A
		SCALE 1:5	SHEET 1 OF 2	



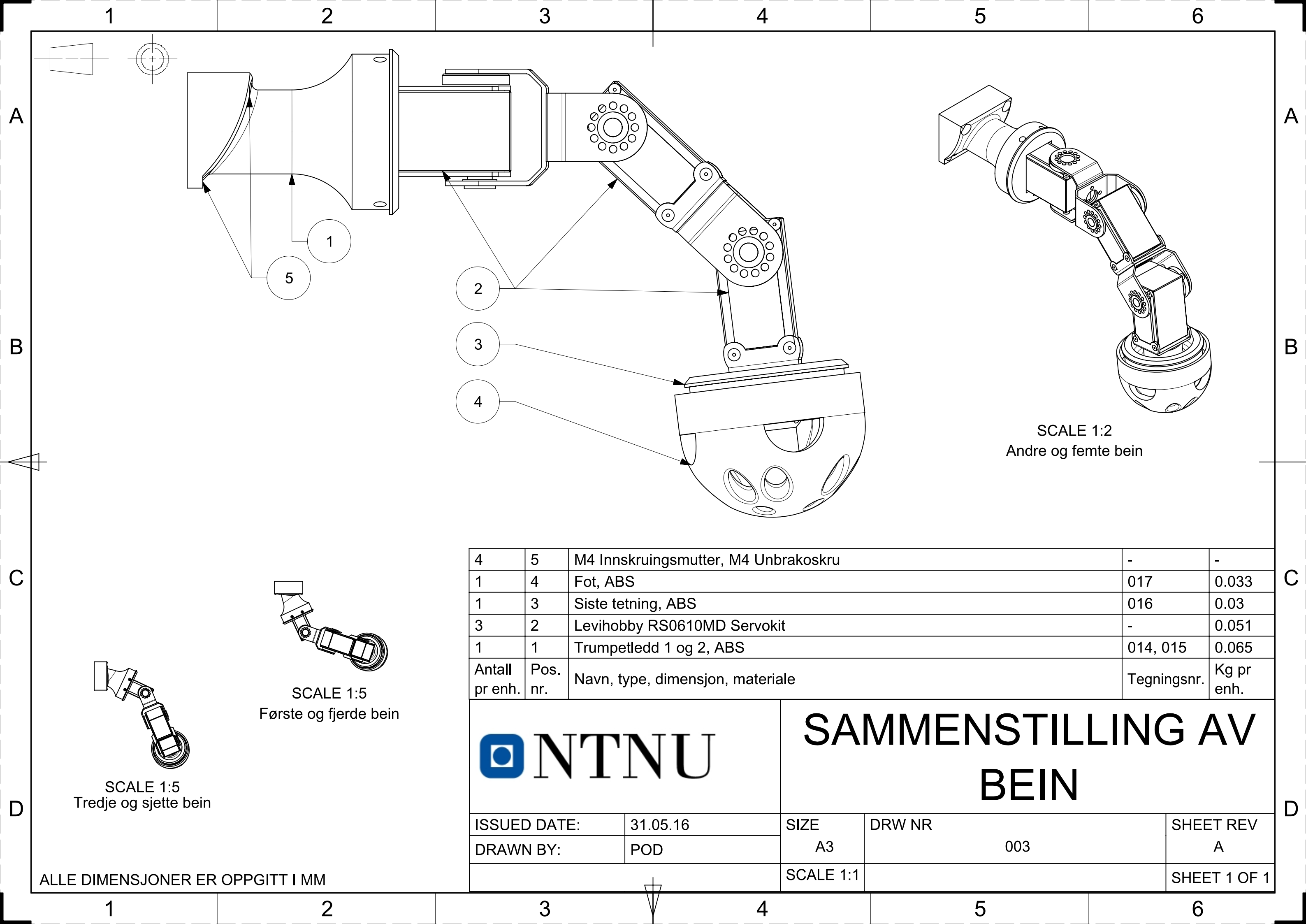
2	-	Avrundet modul	013	0.045
1	19	Pakningssett	-	-
3	18	Kjøleplate, Ø75x3, Aluminium	019	0.03
7	17	O-ringer	-	-
2	16	Pakknippel, M12	-	-
1	15	Pakknippel, M20	-	-
1	14	Maestro 24-kanals servokontroller	-	0.03
3	13	Spenningsregulator, 74x74x32	-	0.28
1	12	ODROID XU4	-	0.12
18	11	M4 Innskruingsmutter, M4 Skru	-	-
6	10	M3 Skru, M3 Mutter	-	-
3	9	Blue Robotics T100	-	0.295
16	8	M3 Unbrako, M3 Mutter	-	-
2	7	Pleksiglass, Ø65, ELP 170 Kamera, Teensy 3.2	012	0.25
2	6	Tredje og sjette bein, ABS	003	0.278
2	5	Andre og femte bein, ABS	003	0.281
2	4	Første og fjerde bein, ABS	003	0.278
15	3	M6 Innskruingsmutter, M6 Unbrakoskru	-	-
1	2	Deksel, 359.1x172.3x45, ABS	009	0.65
1	1	Skrog, 400x210x125, ABS	004	1.15
Antall pr enh.	Pos. nr.	Navn, type, dimensjon, materiale	Tegningsnr.	Kg pr enh.



# SAMMENSTILLING AV 001

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	002	A
		SCALE 1:5	SHEET 2 OF 2	

ALLE DIMENSJONER ER OPPGITT I MM



SCALE 1:2  
Andre og femte bein

SCALE 1:5  
Første og fjerde bein

SCALE 1:5  
Tredje og sjette bein

4	5	M4 Innskruingsmutter, M4 Unbrakoskru	-	-
1	4	Fot, ABS	017	0.033
1	3	Siste tetning, ABS	016	0.03
3	2	Levi hobby RS0610MD Servokit	-	0.051
1	1	Trumpetledd 1 og 2, ABS	014, 015	0.065
Antall pr enh.	Pos. nr.	Navn, type, dimensjon, materiale	Tegningsnr.	Kg pr enh.

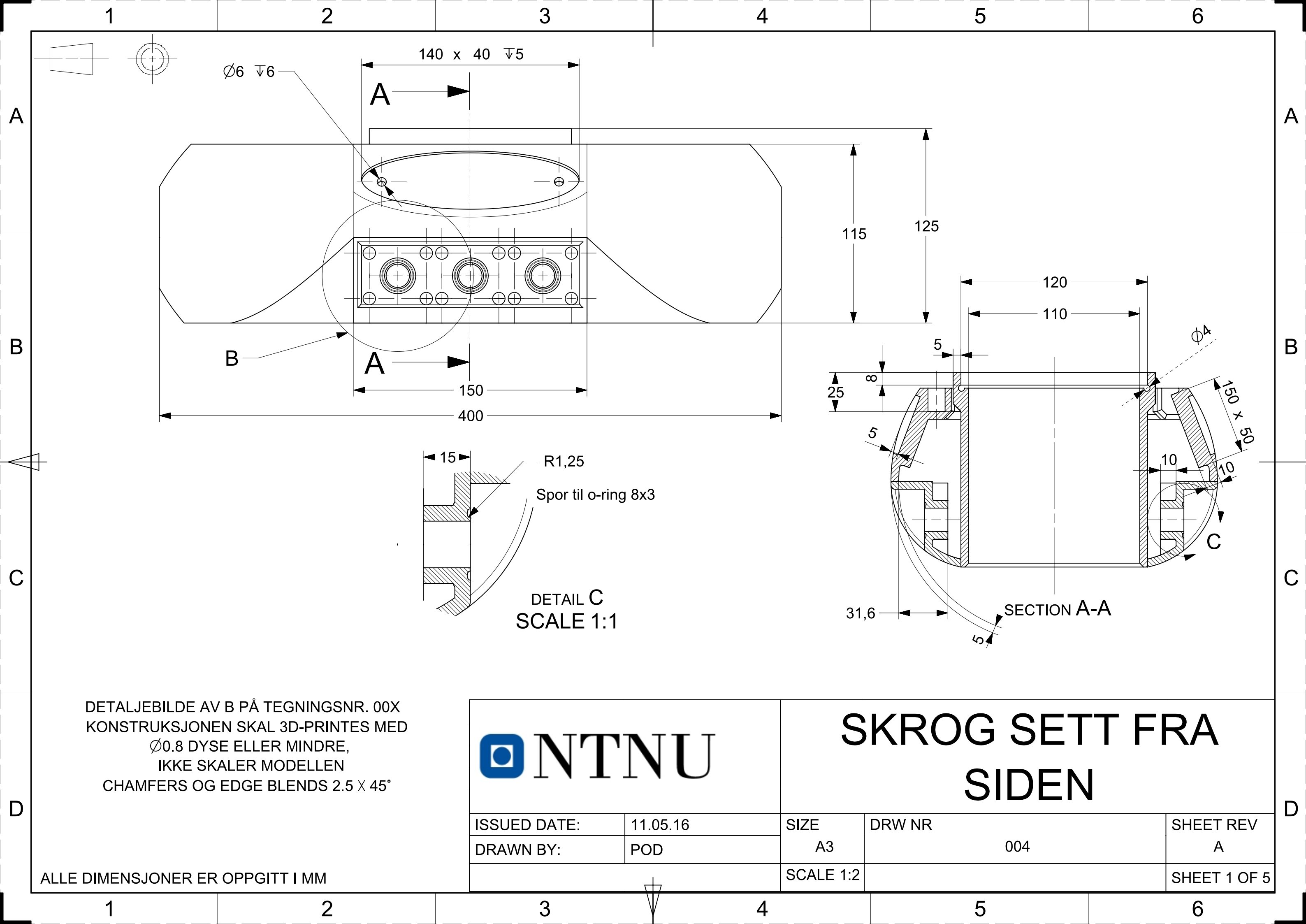


# SAMMENSTILLING AV BEIN

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	003	A
		SCALE 1:1	SHEET 1 OF 1	

ALLE DIMENSJONER ER OPPGITT I MM





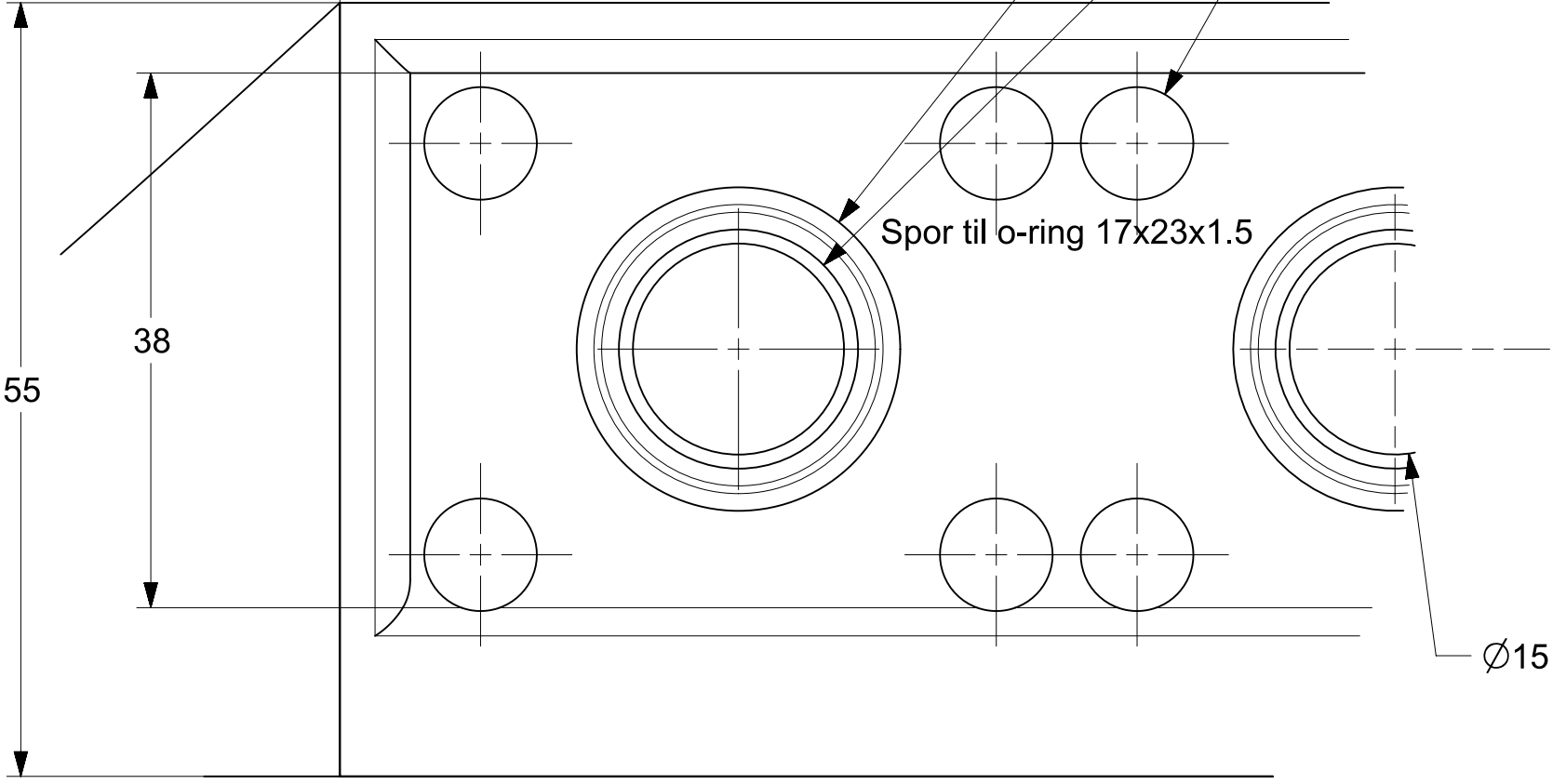
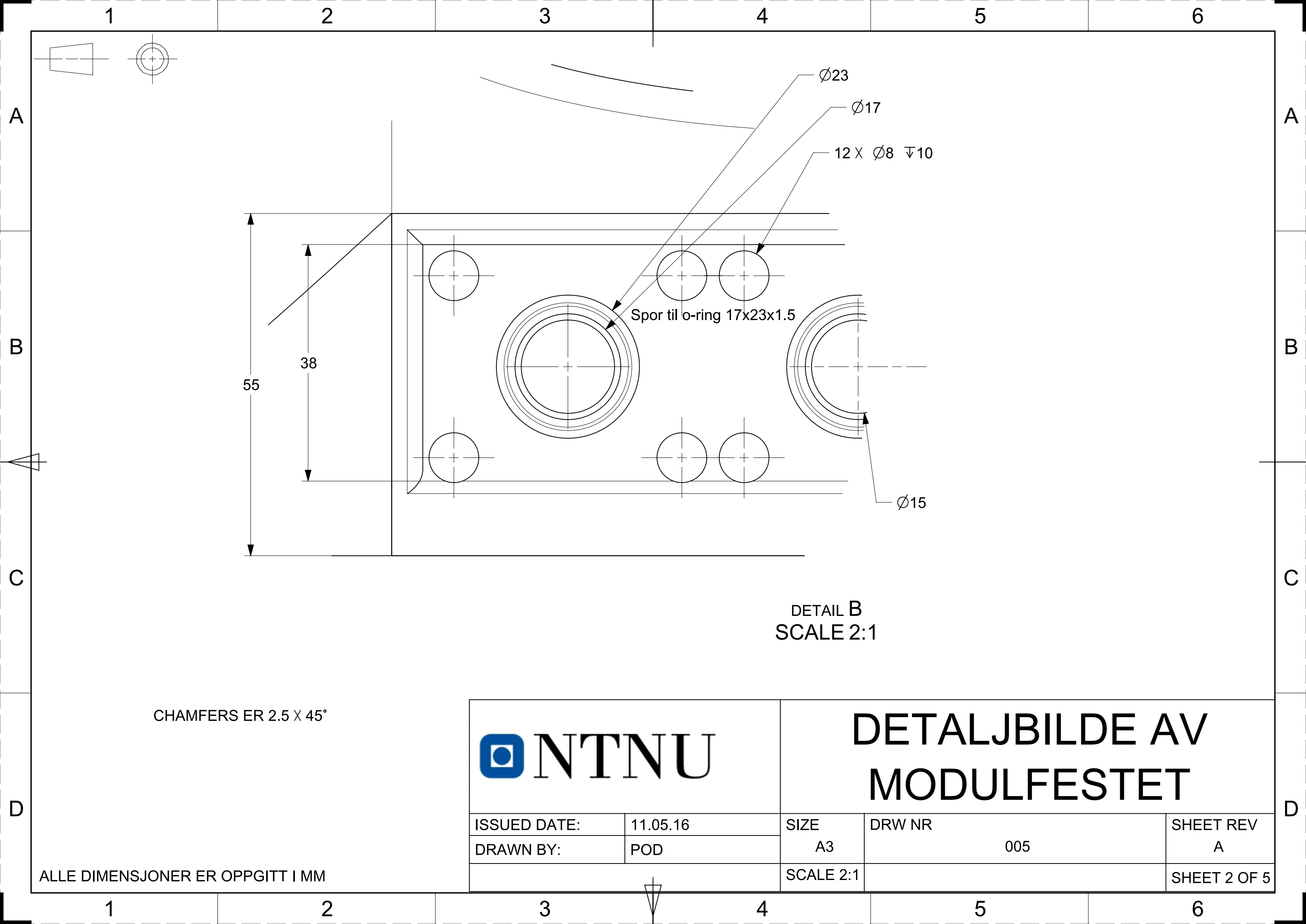
DETALJEBILDE AV B PÅ TEGNINGSNR. 00X  
 KONSTRUKSJONEN SKAL 3D-PRINTES MED  
 $\phi 0.8$  DYSE ELLER MINDRE,  
 IKKE SKALER MODELLEN  
 CHAMFERS OG EDGE BLENDS  $2.5 \times 45^\circ$

ALLE DIMENSJONER ER OPPGITT I MM




# SKROG SETT FRA SIDEN

ISSUED DATE:	11.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	004	A
		SCALE 1:2	SHEET 1 OF 5	

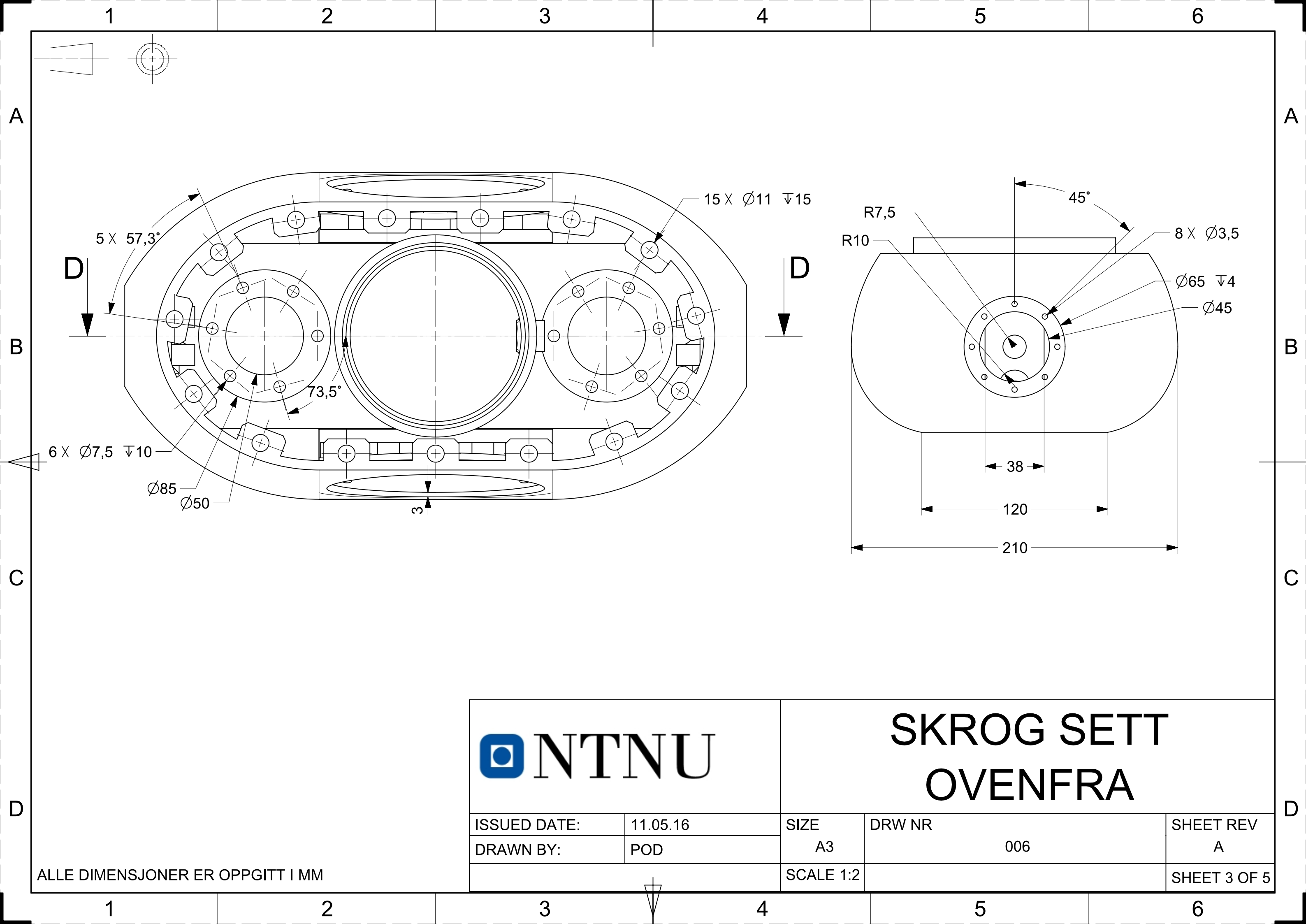



DETAIL B  
SCALE 2:1

CHAMFERS ER 2.5 X 45°

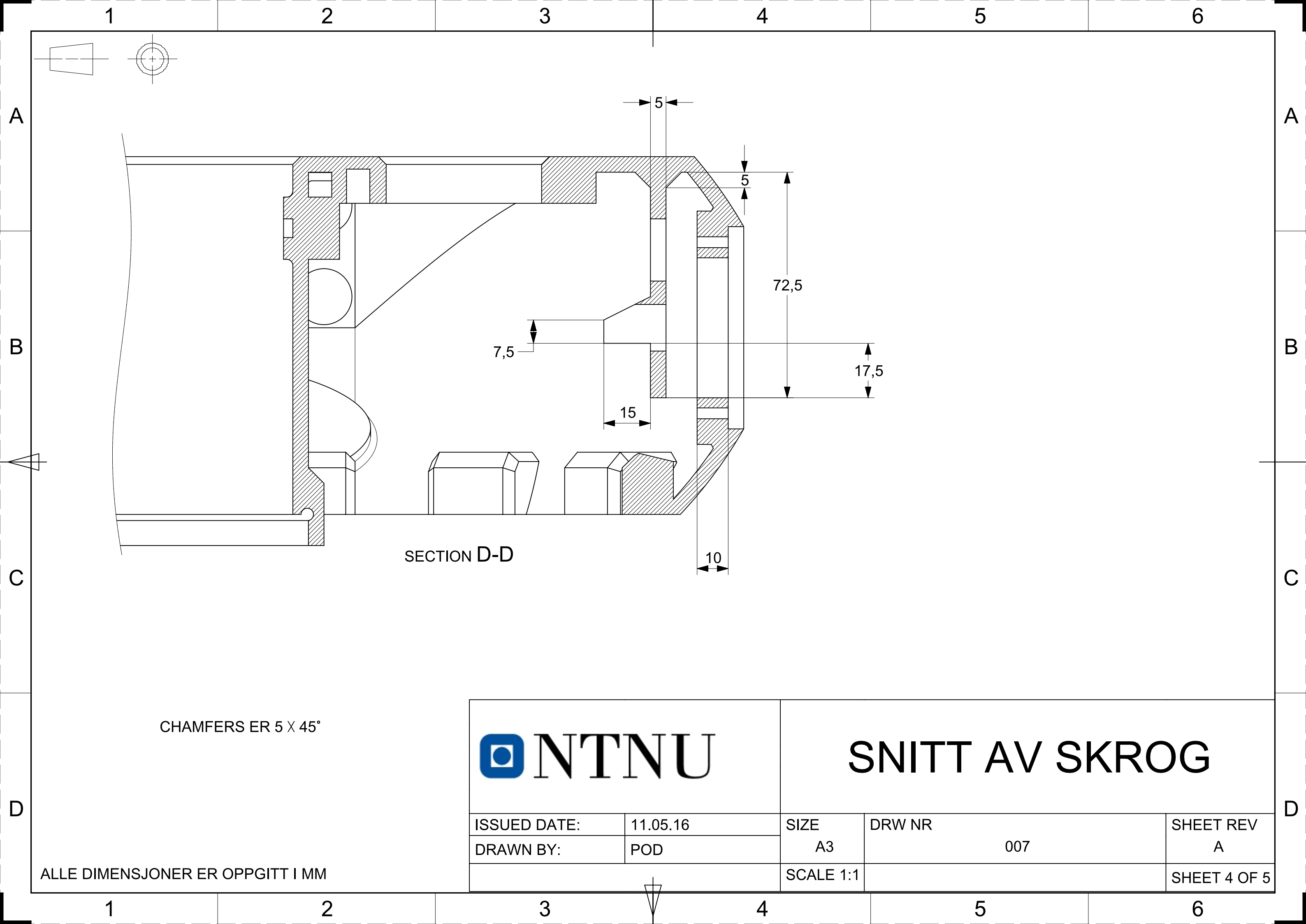
		<h1 style="text-align: center;">DETALJBILDE AV MODULFESTET</h1>			
					ISSUED DATE:
DRAWN BY:		POD	A3	005	A
			SCALE 2:1		SHEET 2 OF 5

ALLE DIMENSJONER ER OPPGITT I MM



		<h1 style="text-align: center;">SKROG SETT OVENFRA</h1>			
					ISSUED DATE:
DRAWN BY:		POD	A3	006	A
			SCALE 1:2		SHEET 3 OF 5

ALLE DIMENSJONER ER OPPGITT I MM



CHAMFERS ER 5 X 45°

SECTION D-D

# SNITT AV SKROG



ISSUED DATE: 11.05.16

DRAWN BY: POD

SIZE

DRW NR

SHEET REV

A3

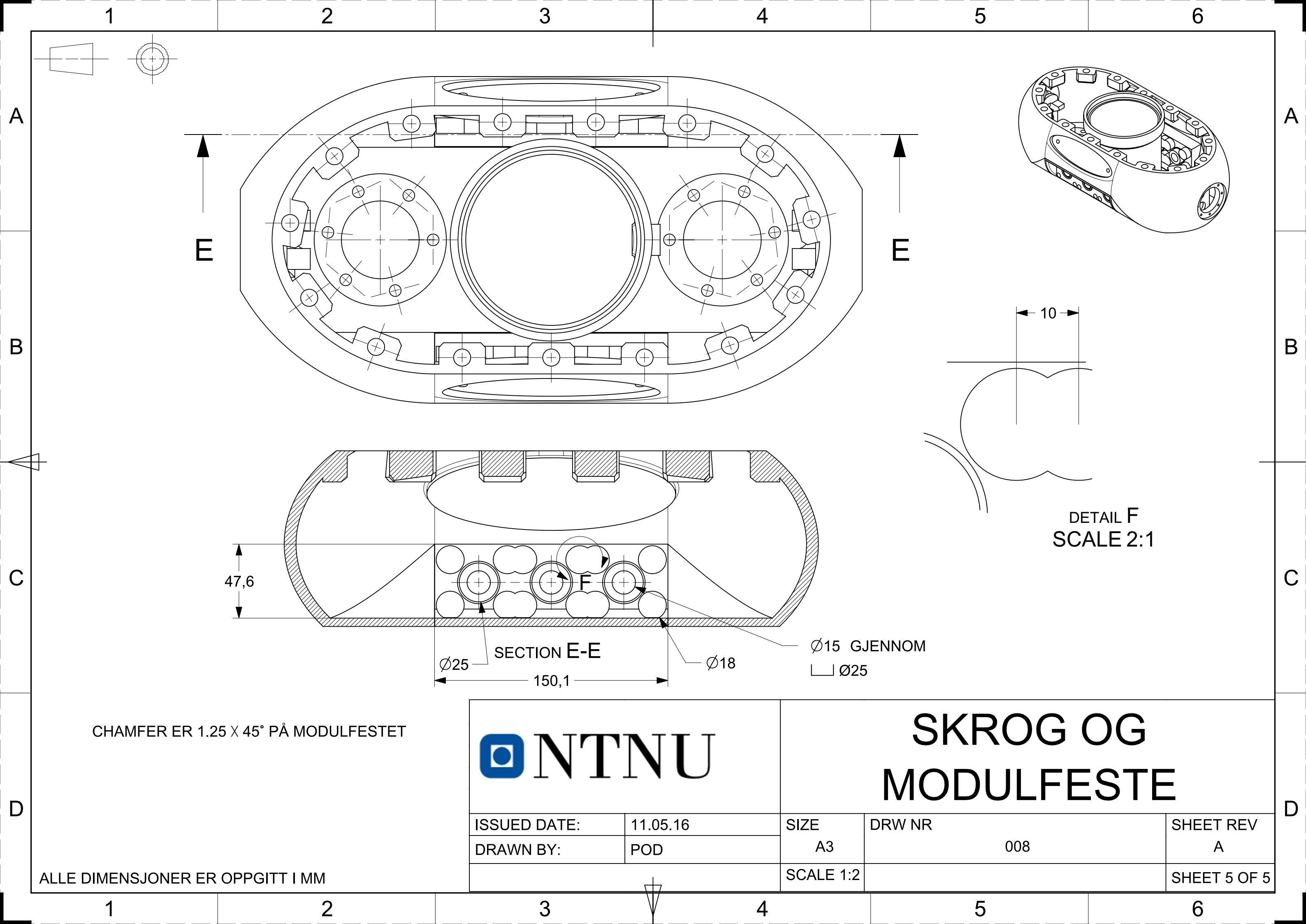
007

A

SCALE 1:1

SHEET 4 OF 5

ALLE DIMENSJONER ER OPPGITT I MM



E

E

47,6

Ø25

SECTION E-E

150,1

Ø18

Ø15 GJENNOM

□ Ø25

10

DETAIL F  
SCALE 2:1

CHAMFER ER 1.25 X 45° PÅ MODULFESTET



# SKROG OG MODULFESTE

ISSUED DATE:	11.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	008	A
		SCALE 1:2	SHEET 5 OF 5	

ALLE DIMENSJONER ER OPPGITT I MM

1

2

3

4

5

6

A

A

B

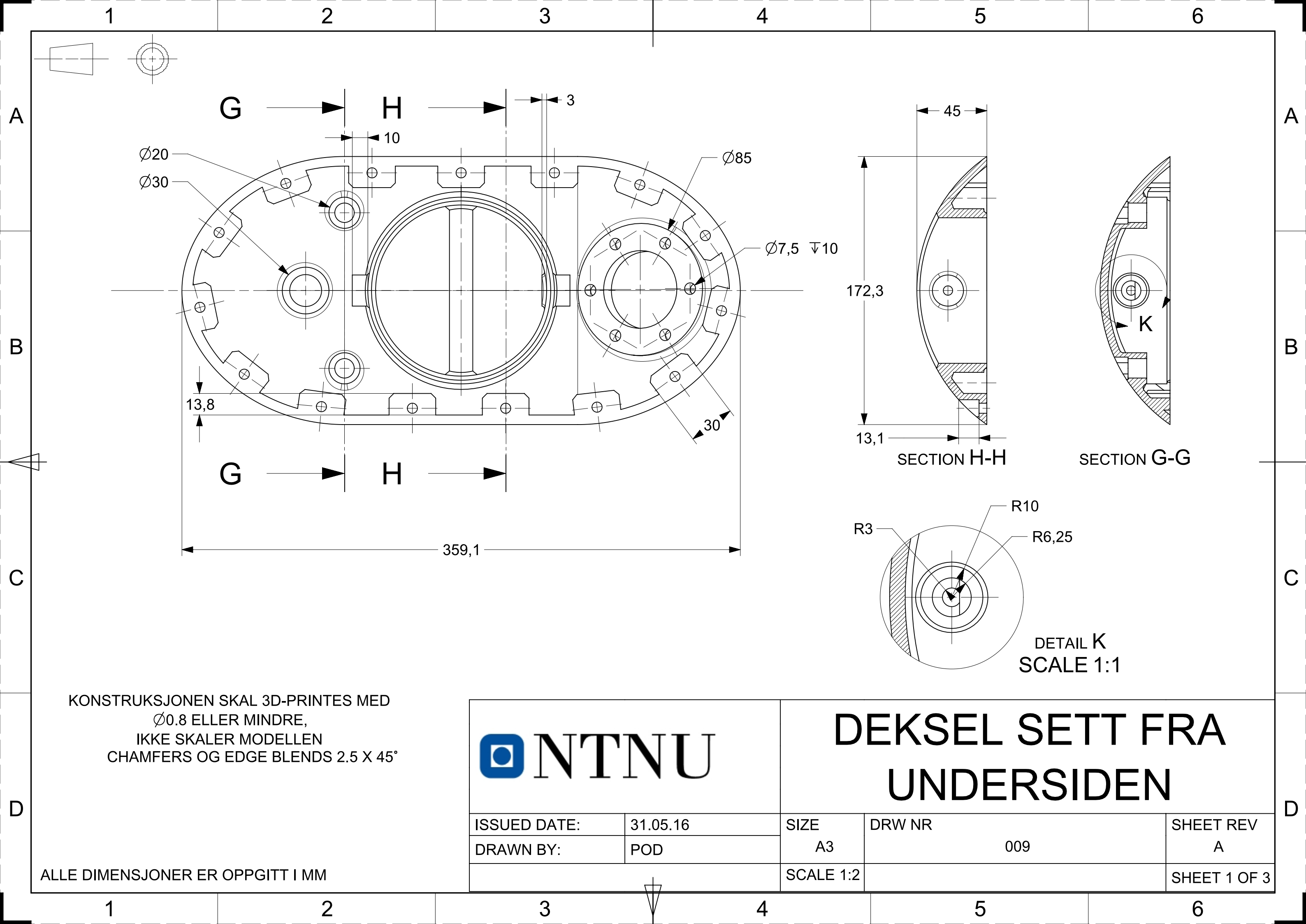
B

C

C

D

D



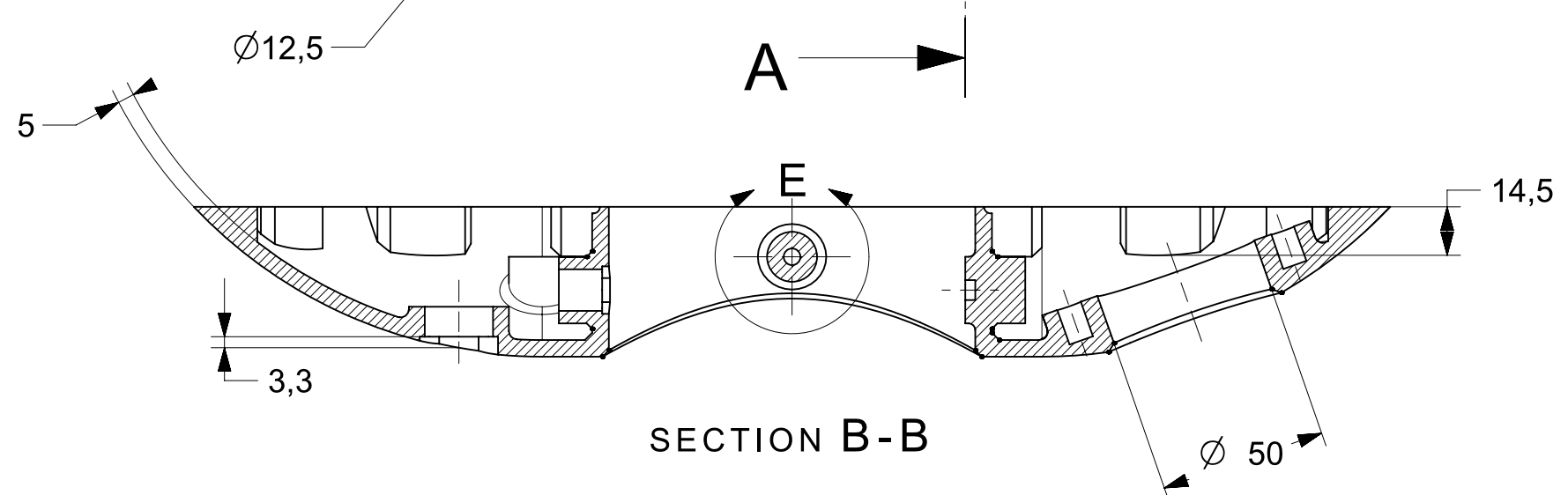
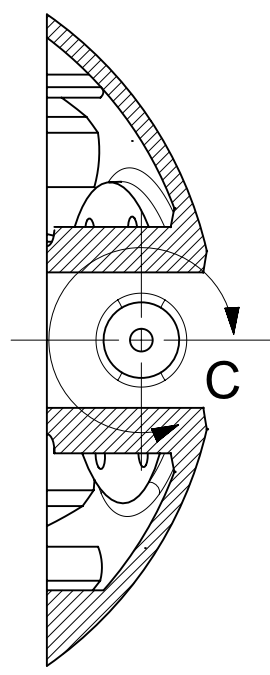
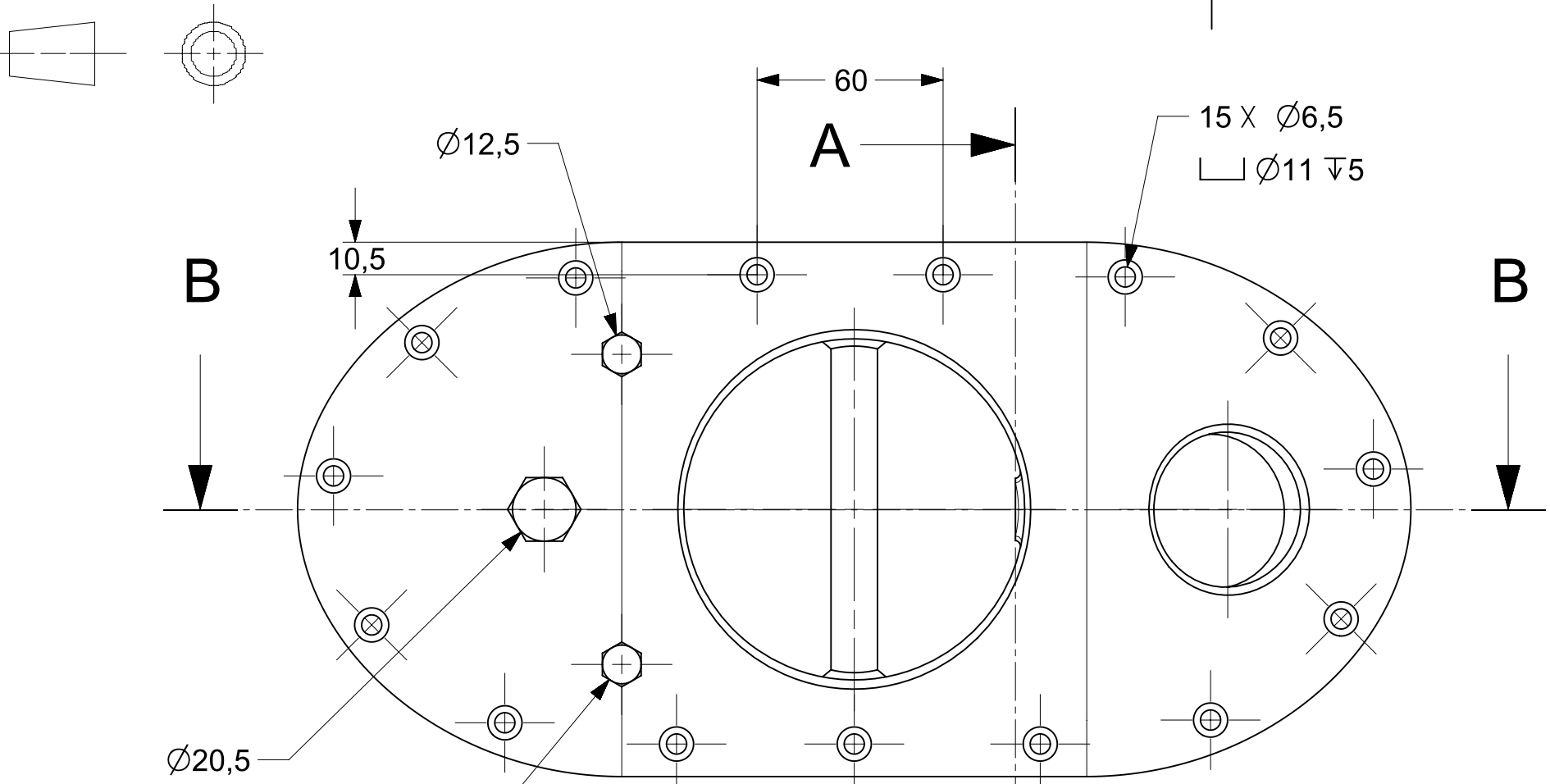
KONSTRUKSJONEN SKAL 3D-PRINTES MED  
 Ø0.8 ELLER MINDRE,  
 IKKE SKALER MODELLEN  
 CHAMFERS OG EDGE BLENDS 2.5 X 45°

ALLE DIMENSJONER ER OPPGITT I MM



# DEKSEL SETT FRA UNDERSIDEN

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	009	A
		SCALE 1:2	SHEET 1 OF 3	



DETALJEBILDER PÅ TEGNINGSNR. 00X  
CHAMFERS 2.5 X 45°

ALLE DIMENSJONER ER OPPGITT I MM

		<h1>DEKSEL SETT OVENFRA</h1>			
					ISSUED DATE:
DRAWN BY:		POD	A3	010	A
			SCALE 1:2		SHEET 2 OF 3

1

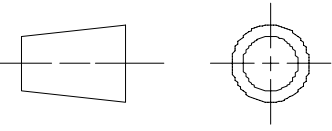
2

3

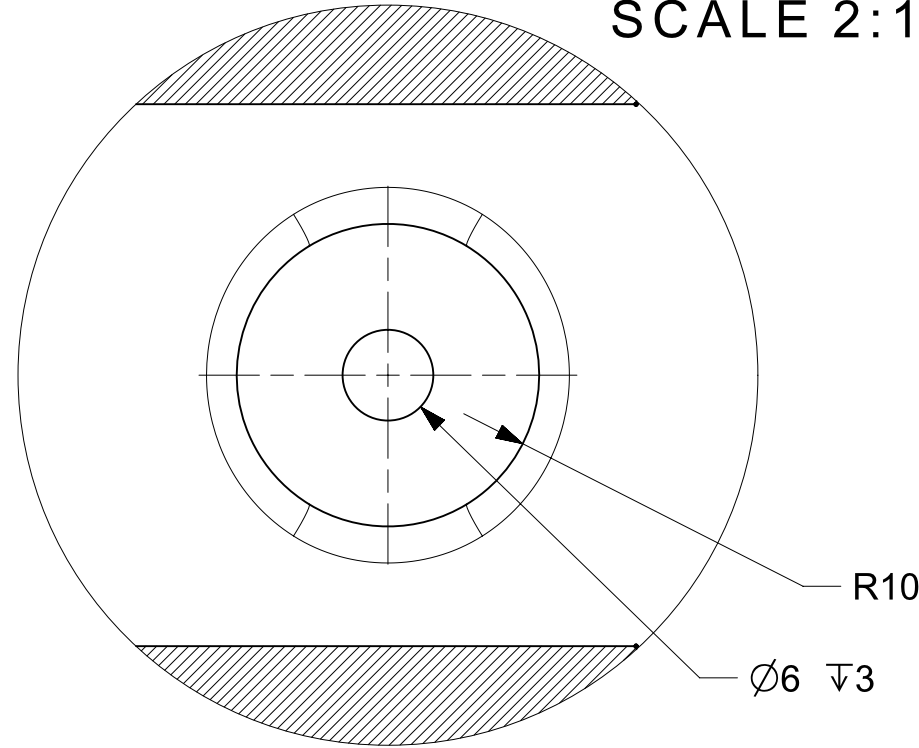
4

5

6

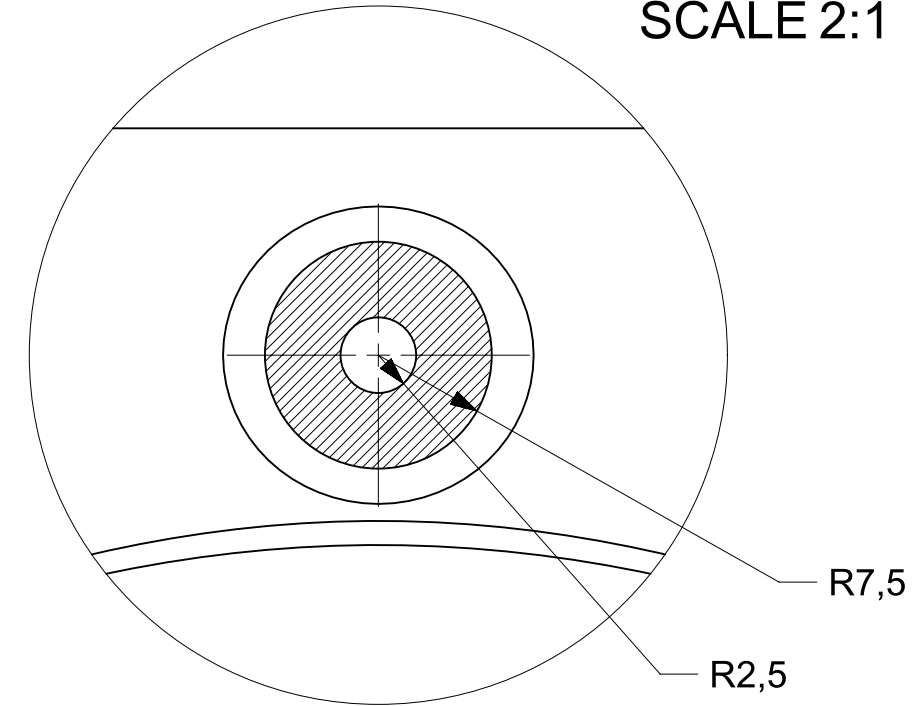


DETAIL C  
SCALE 2:1



EDGE BLEND 2.5

DETAIL E  
SCALE 2:1



R10

Ø6  $\nabla$ 3

R7,5

R2,5



# DETALJEBILDER FRA DEKSEL

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	011	A
		SCALE 2:1	SHEET 3 OF 3	

1

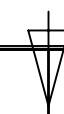
2

3

4

5

6



A

A

B

B

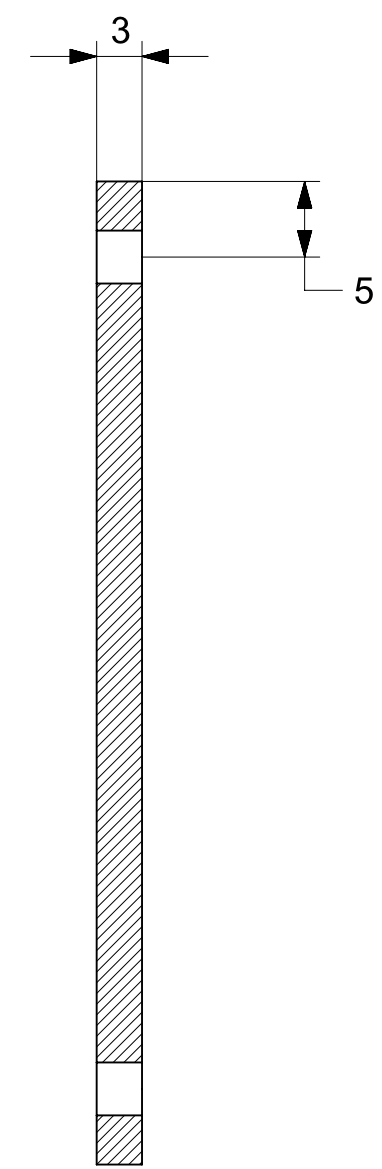
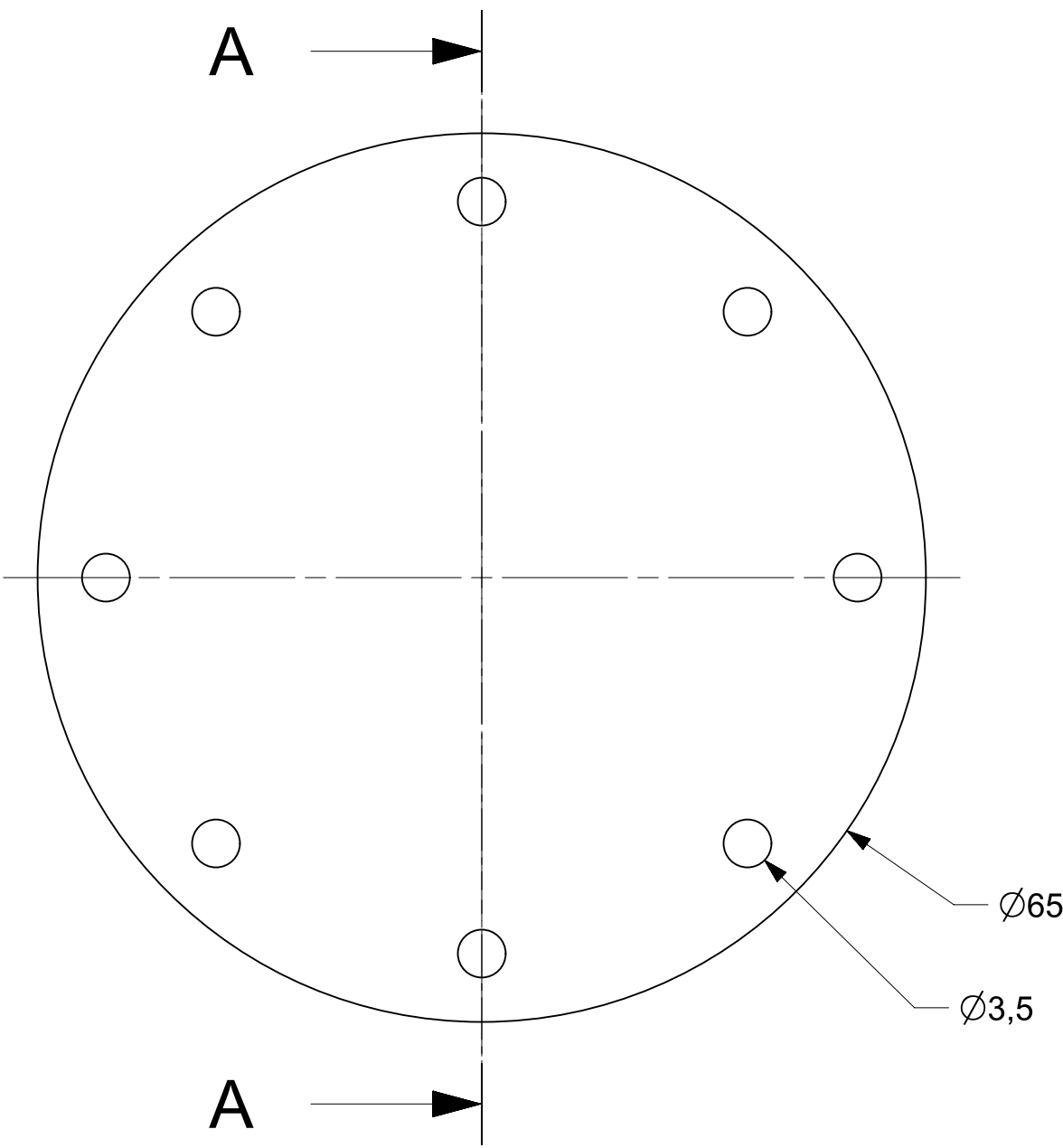
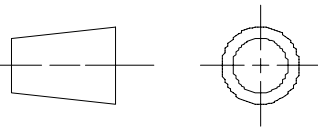
C

C

D

D





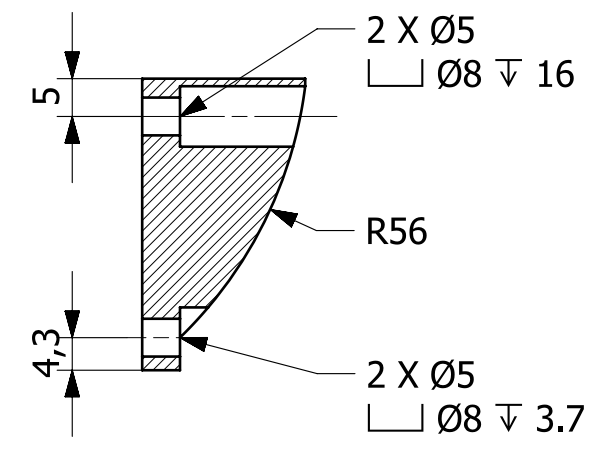
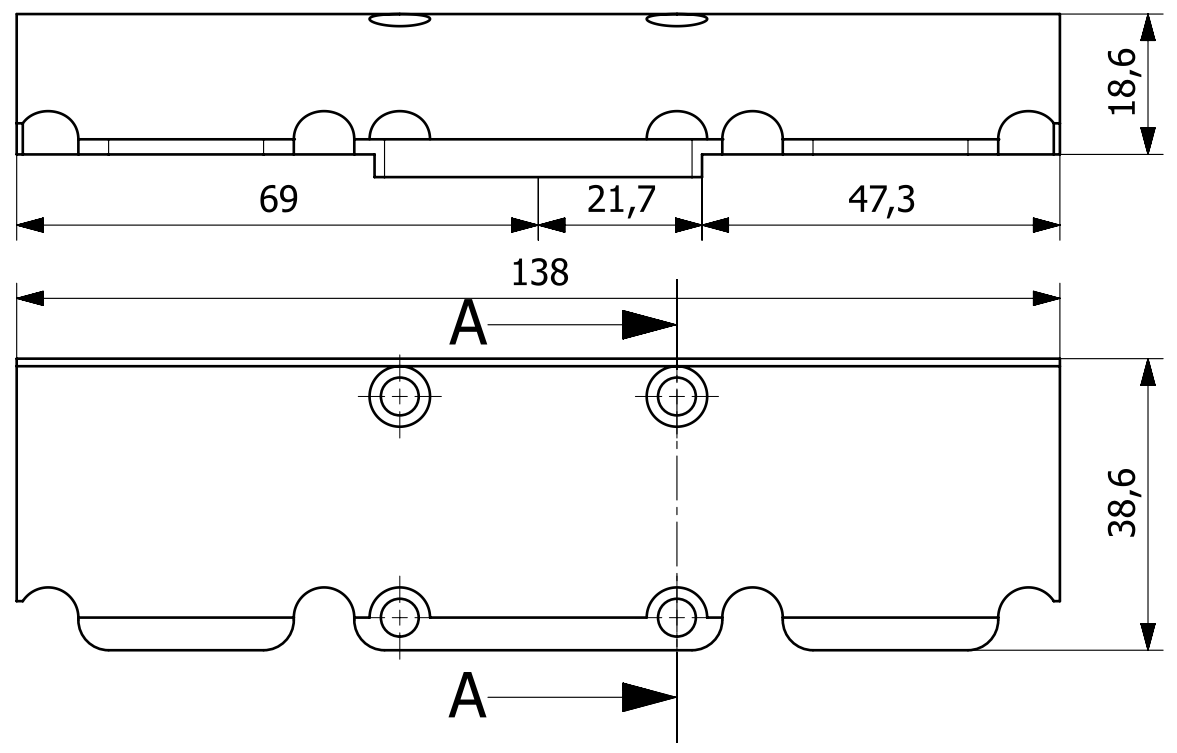
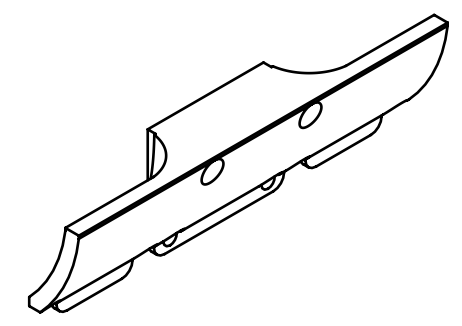
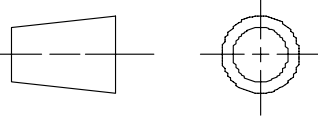
SECTION A-A



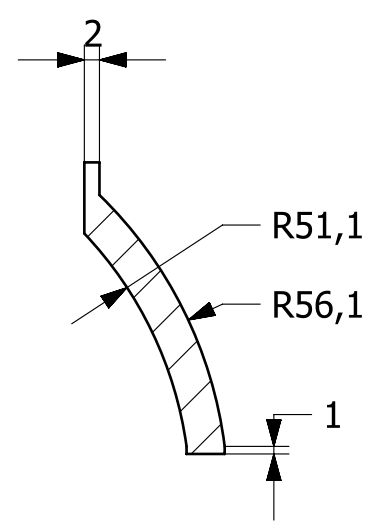
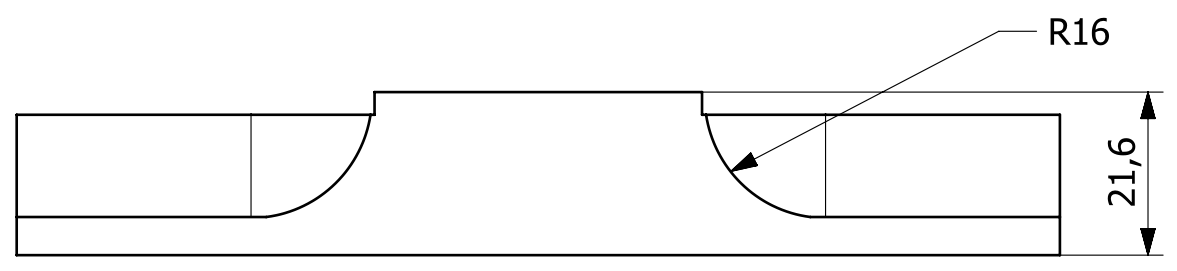
# KAMERAVINDU

ISSUED DATE:	12.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	012	A
		SCALE 2:1	SHEET 1 OF 1	

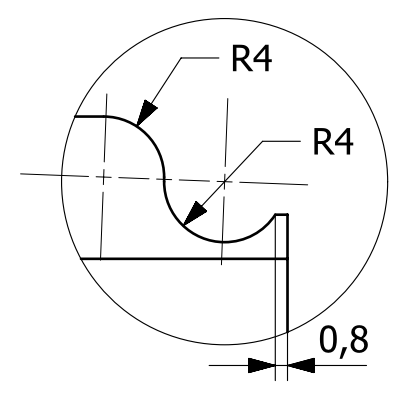
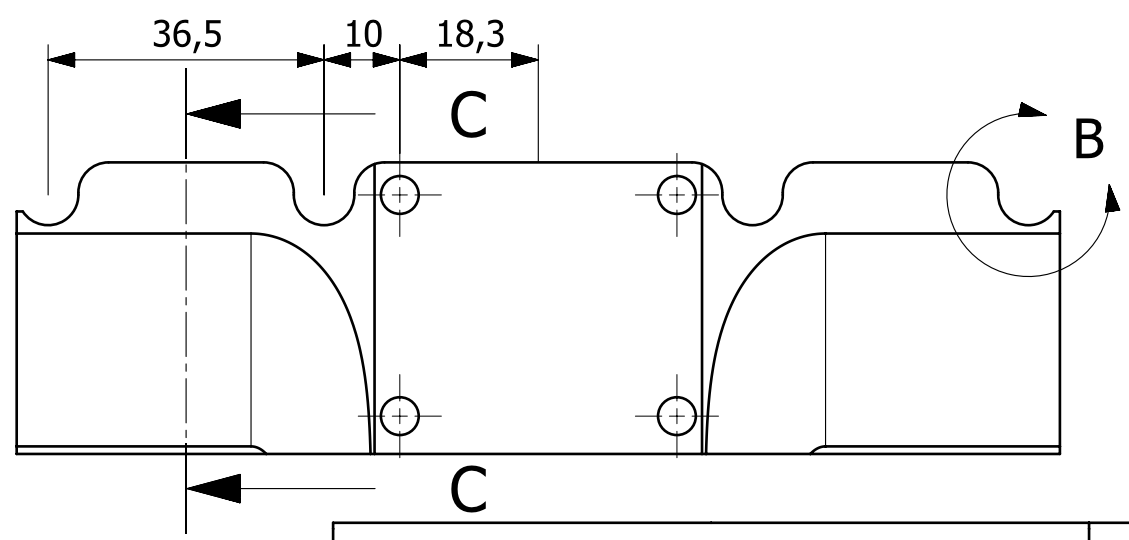
ALLE DIMENSJONER ER OPPGITT I MM



SECTION A-A



SECTION C-C



DETAIL B  
SCALE 2:1

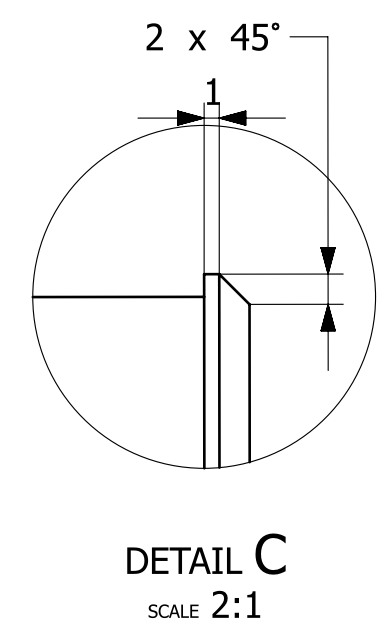
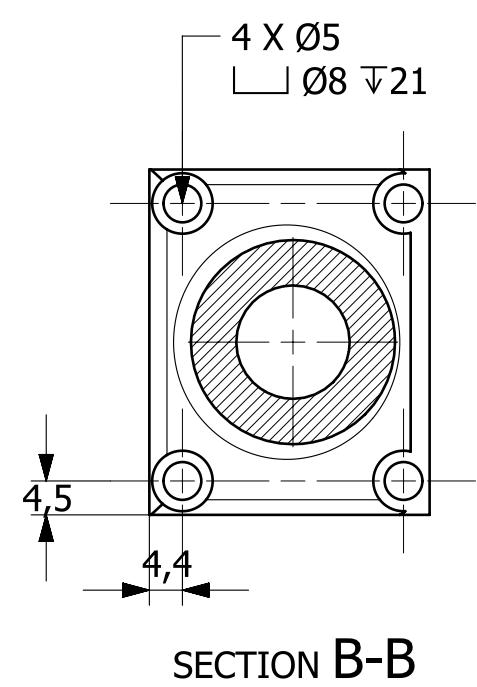
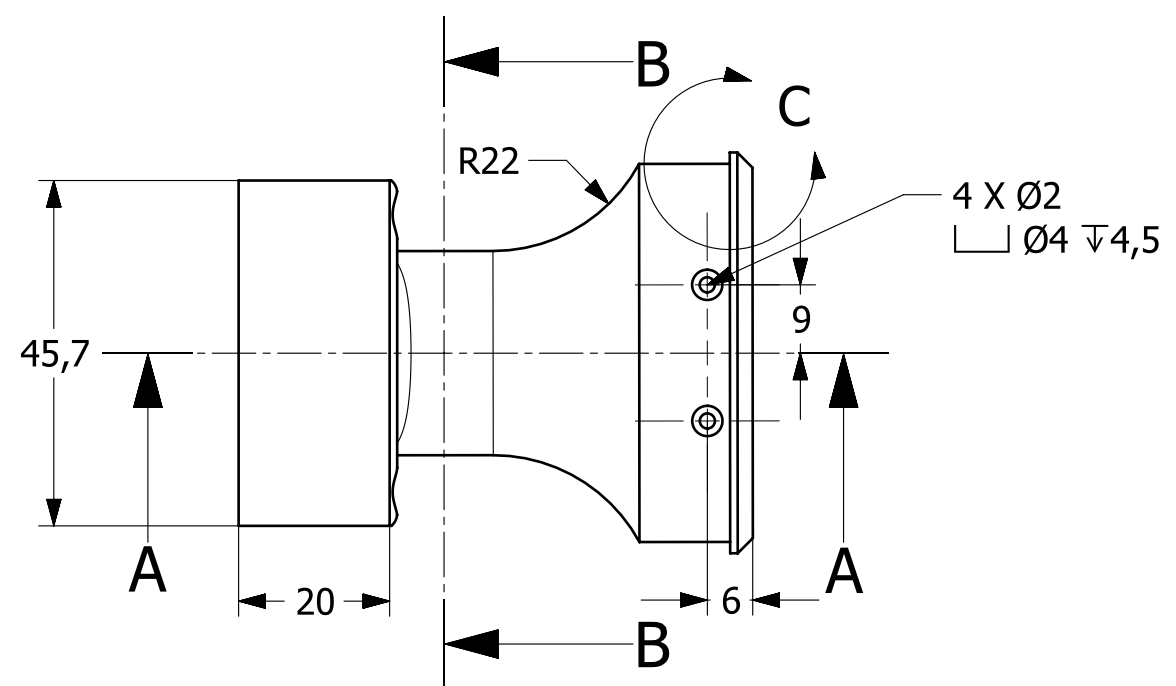
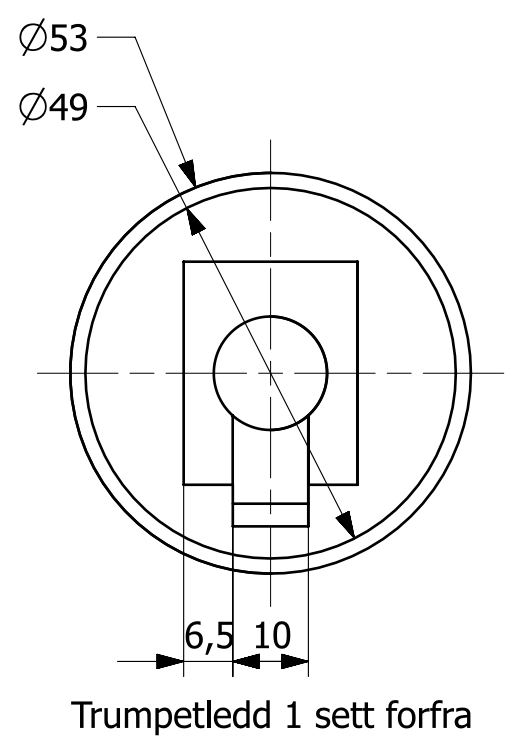
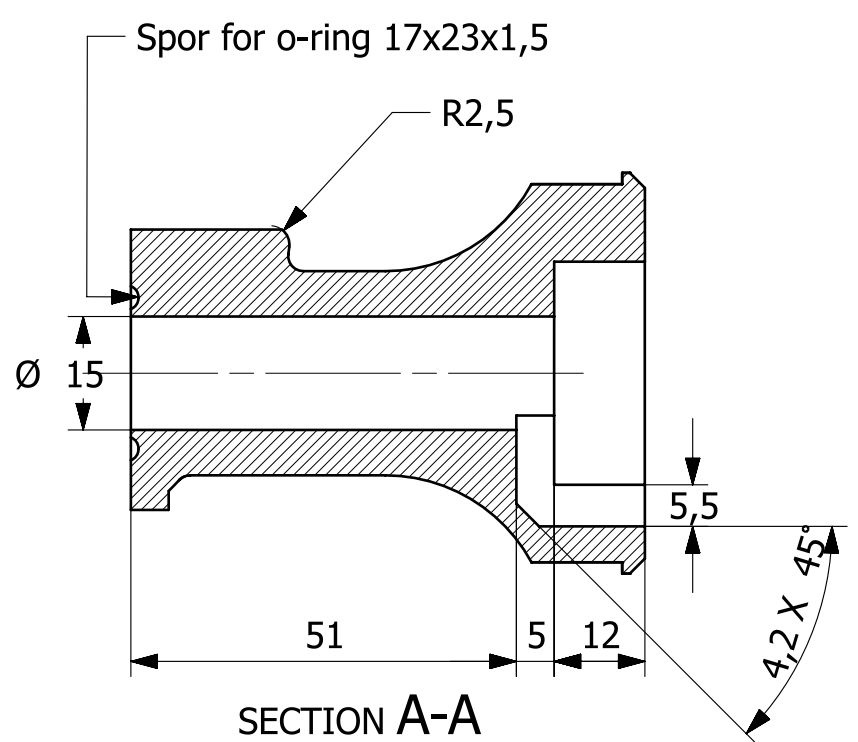
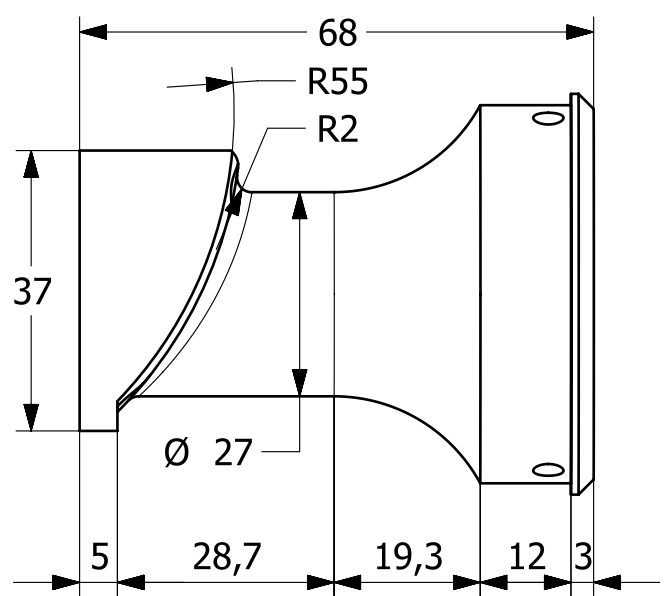
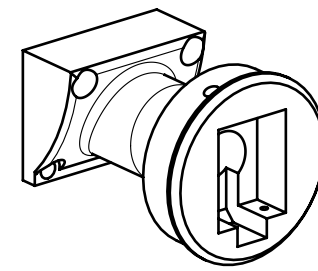
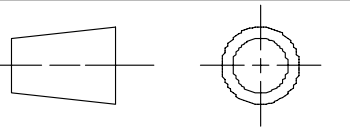
DELEN SKAL 3D-PRINTES,  
IKKE SKALER MODELLEN  
0,8 MM DYSE ELLER MINDRE

ALLE DIMENSJONER ER OPPGITT I MM



# AVRUNDET MODUL

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	013	A
		SCALE 1:1		SHEET 1 OF 1



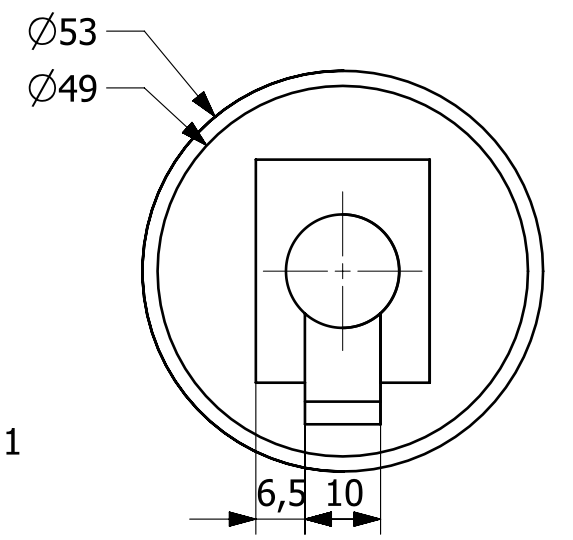
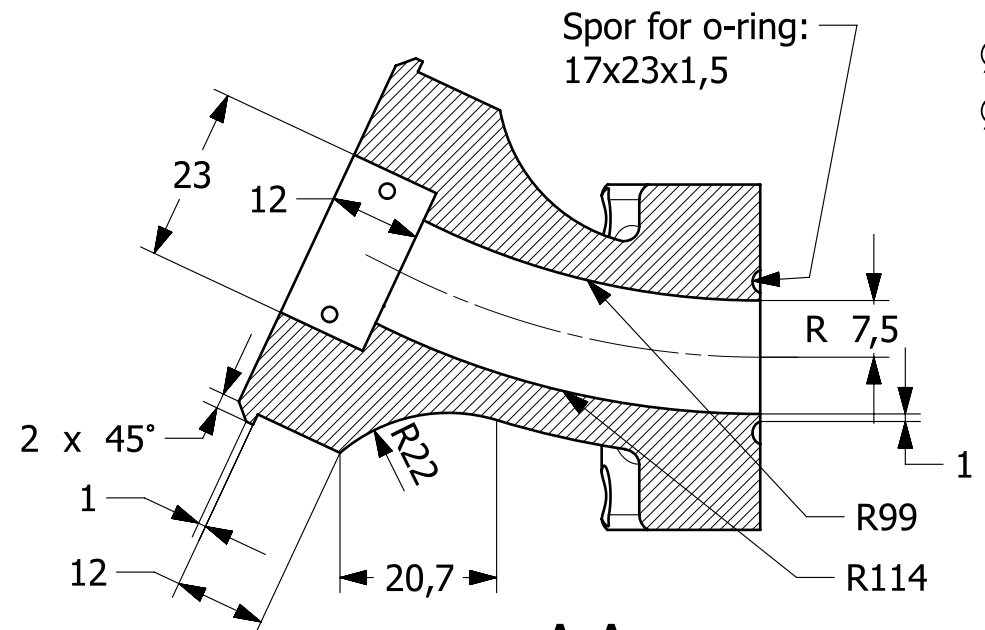
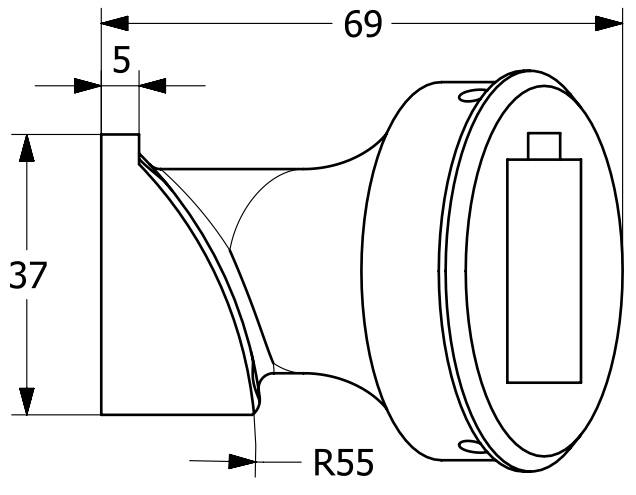
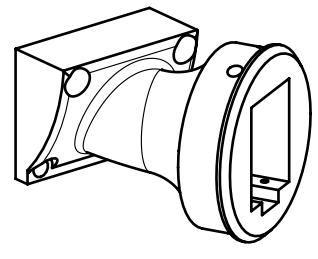
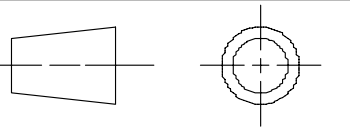
DELEN SKAL 3D-PRINTES,  
IKKE SKALER MODELLEN  
0,4 MM DYSE ELLER MINDRE



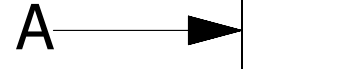
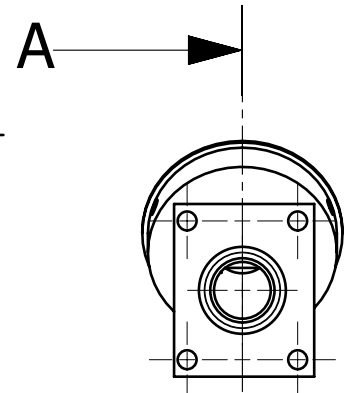
# TRUMPETLEDD 1

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	014	A
		SCALE 1:1		SHEET 1 OF 1

ALLE DIMENSJONER ER OPPGITT I MM

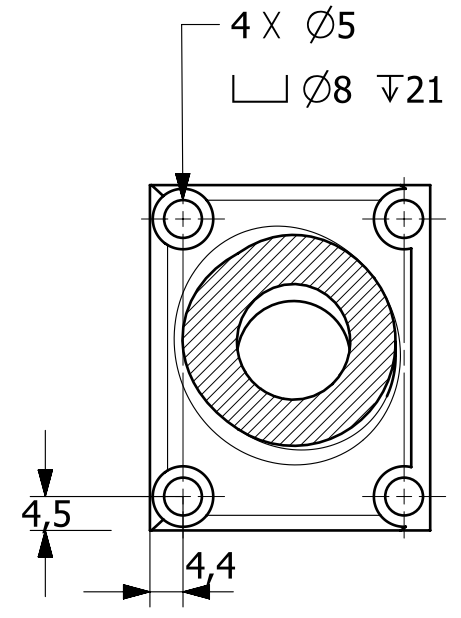
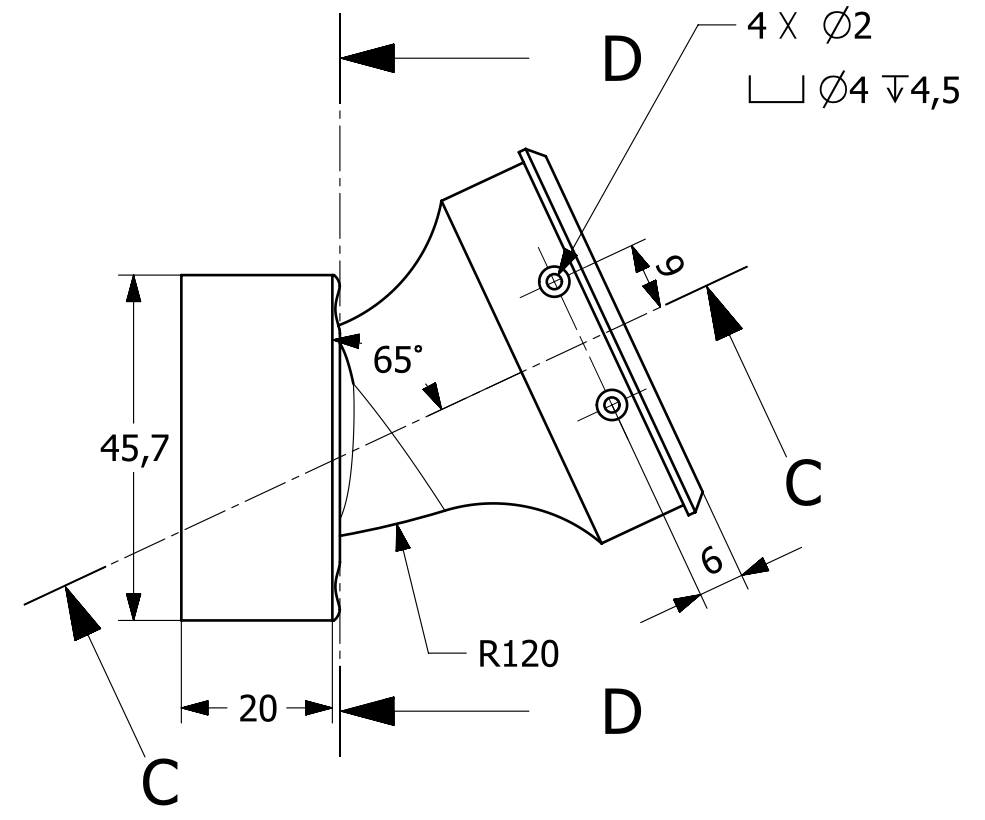


Trumpetledd 2 sett forfra.

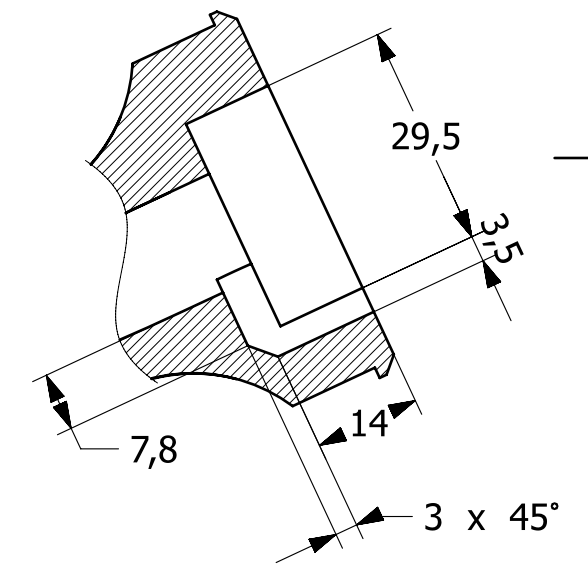


Trumpetledd 2 sett bakfra.

SCALE 1:2



SECTION D-D



SECTION C-C

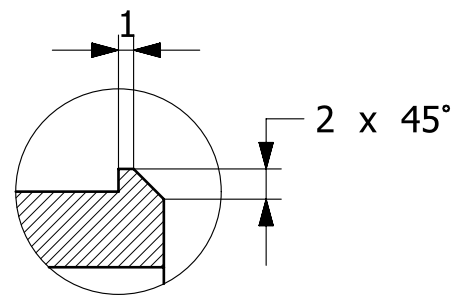
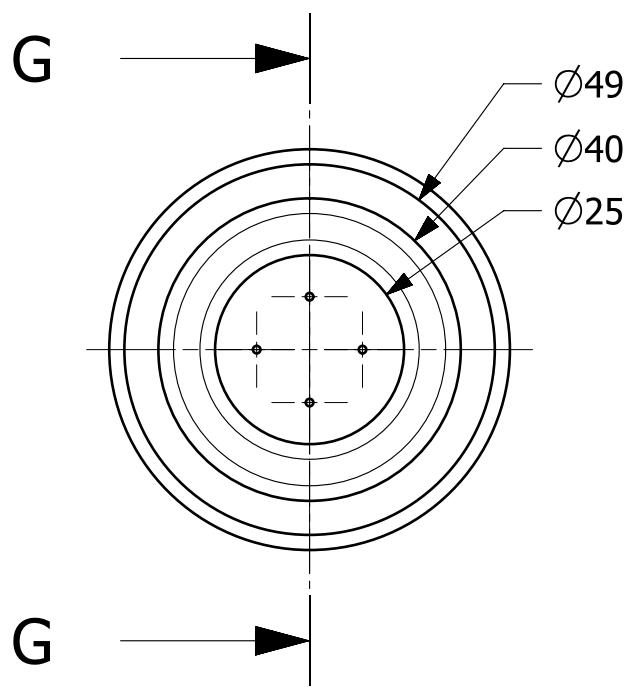
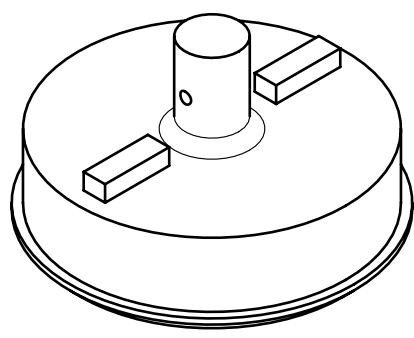
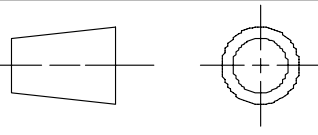
DELEN SKAL 3D-PRINTES,  
IKKE SKALER MODELLEN  
0,8 MM DYSE ELLER MINDRE

ALLE DIMENSJONER ER OPPGITT I MM

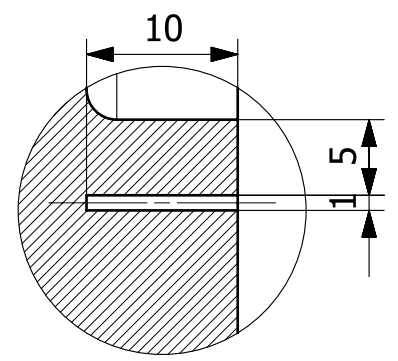


# TRUMPETLEDD 2

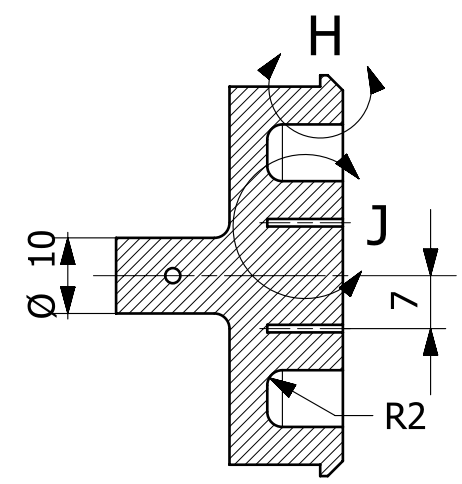
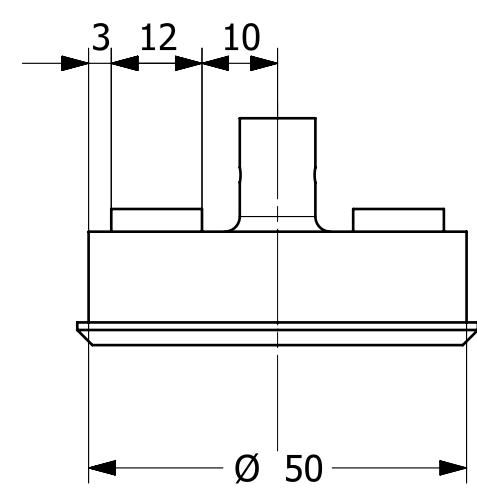
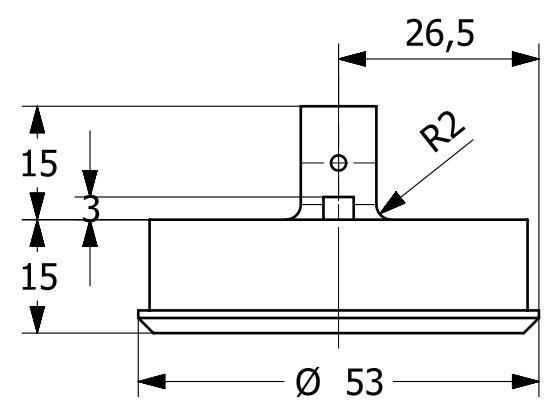
ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	015	A
		SCALE 1:1		SHEET 1 OF 1



DETAIL H  
SCALE 2:1



DETAIL J  
SCALE 2:1



SECTION G-G

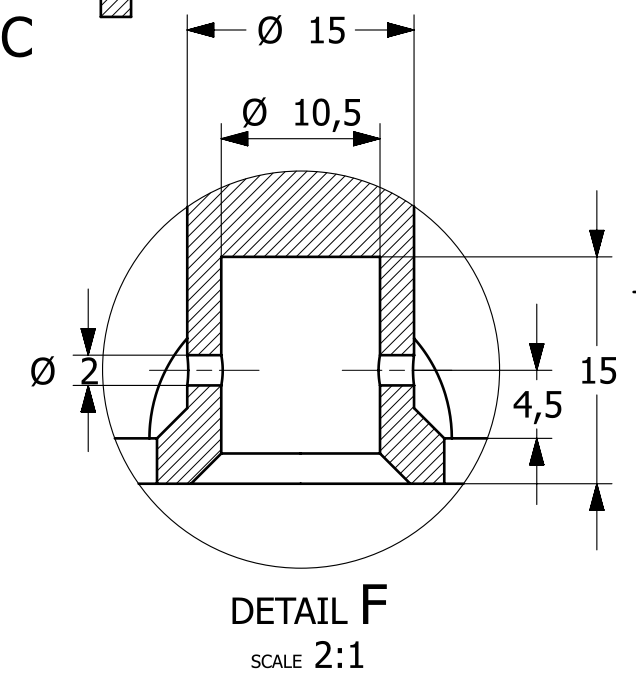
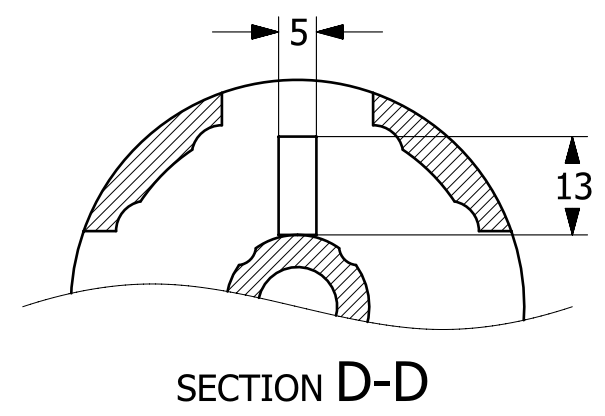
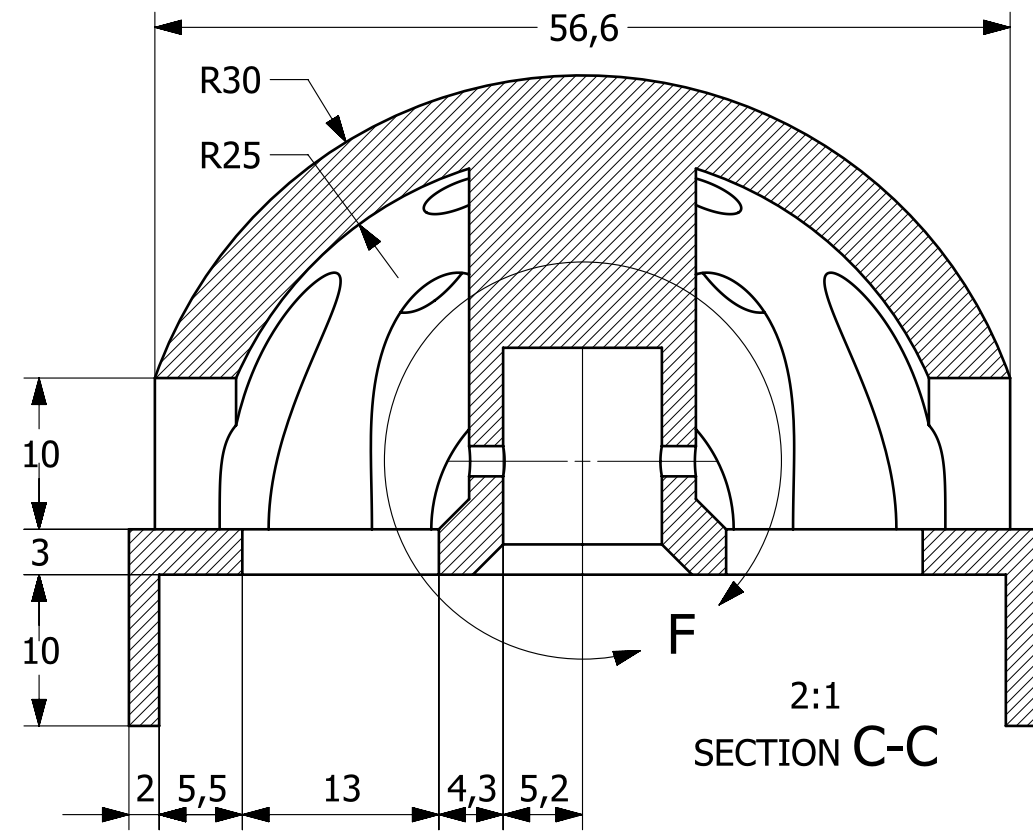
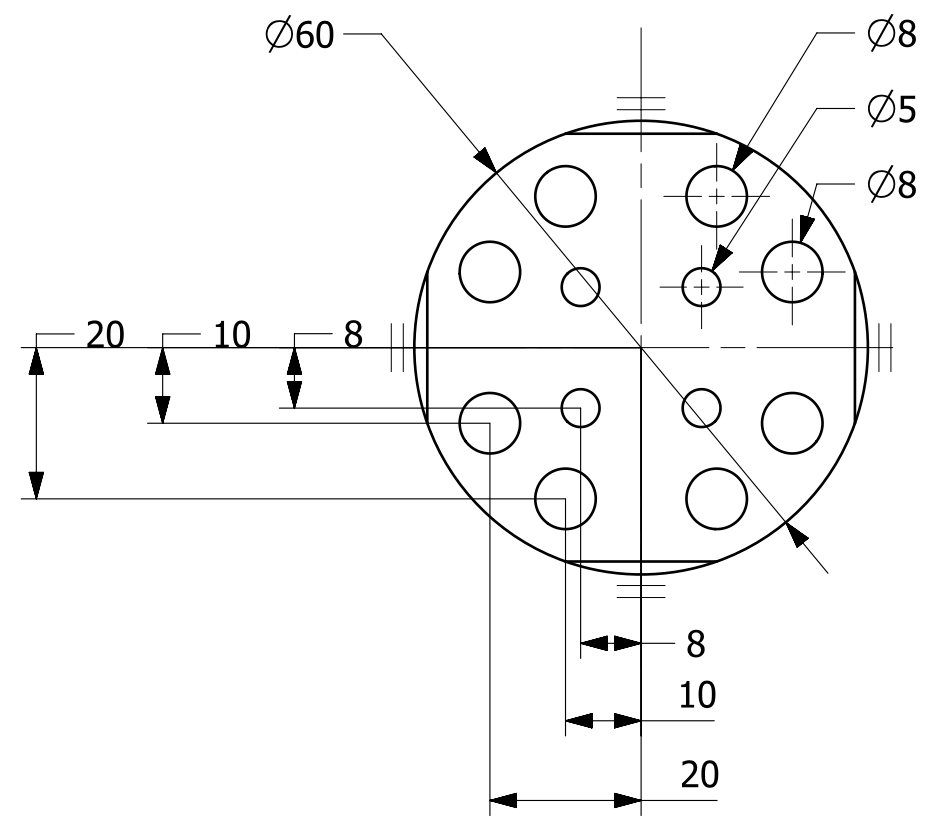
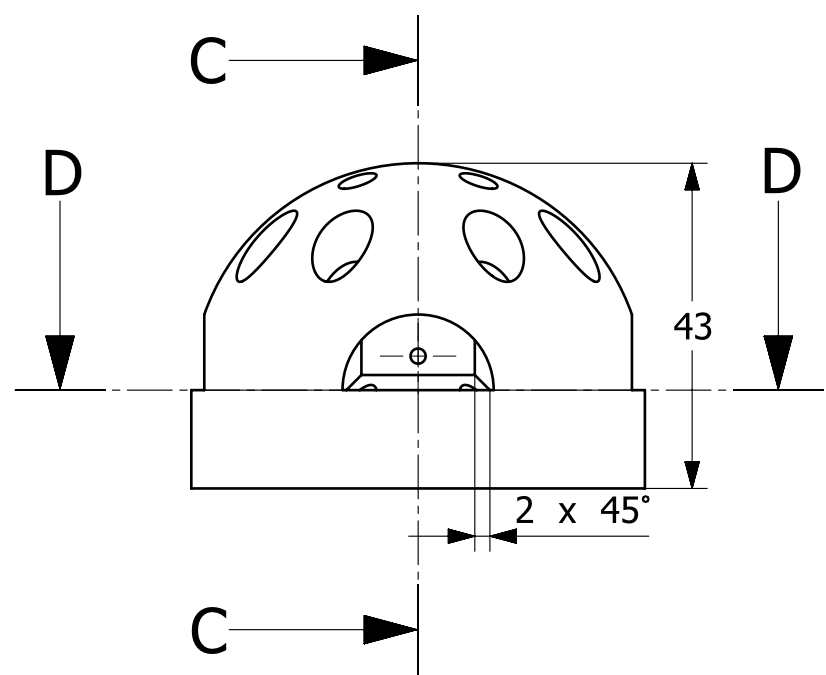
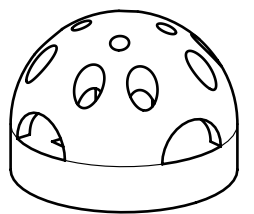
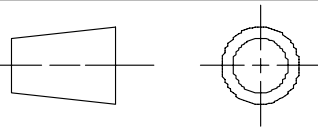
DELEN SKAL 3D-PRINTES,  
IKKE SKALER MODELLEN  
0,4 MM DYSE ELLER MINDRE

ALLE DIMENSJONER ER OPPGITT I MM



# SISTE TETNING

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	016	A
		SCALE 1:1		SHEET 1 OF 1



HULL PÅ Ø8 OG Ø5 SKAL FRESES NED TIL NESTE PLAN

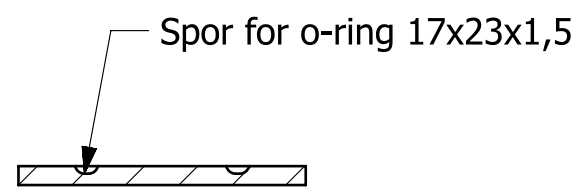
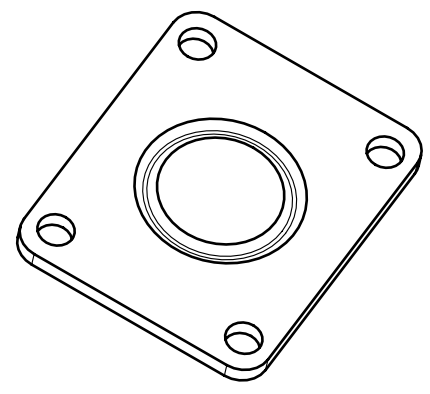
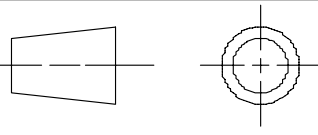
DELEN SKAL 3D-PRINTES,  
IKKE SKALER MODELLEN  
0,4 MM DYSE ELLER MINDRE

ALLE DIMENSJONER ER OPPGITT I MM

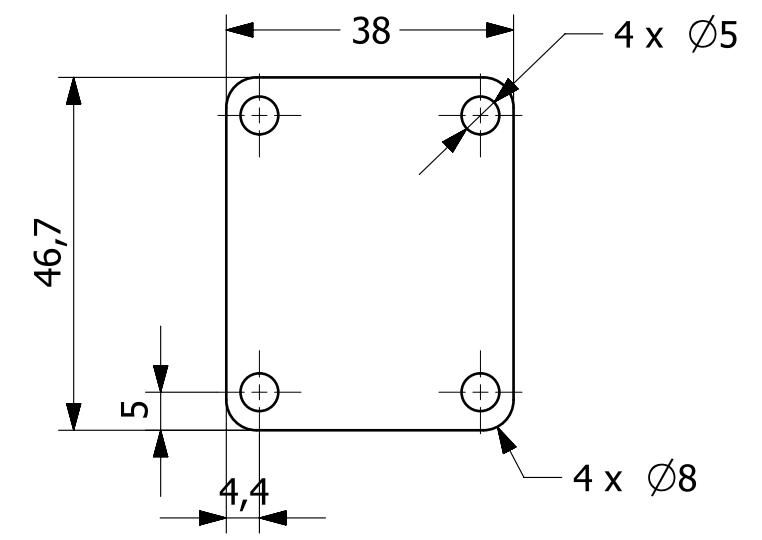
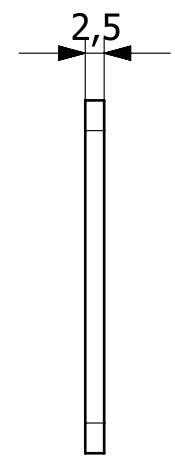
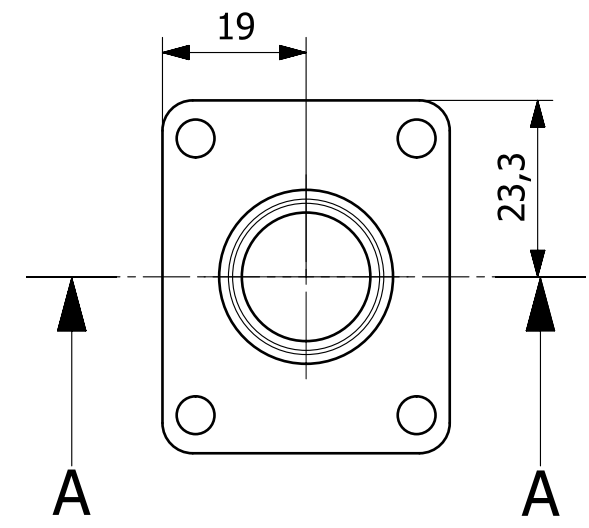


FOT

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	017	A
		SCALE 1:1	SHEET 1 OF 1	



SECTION A-A



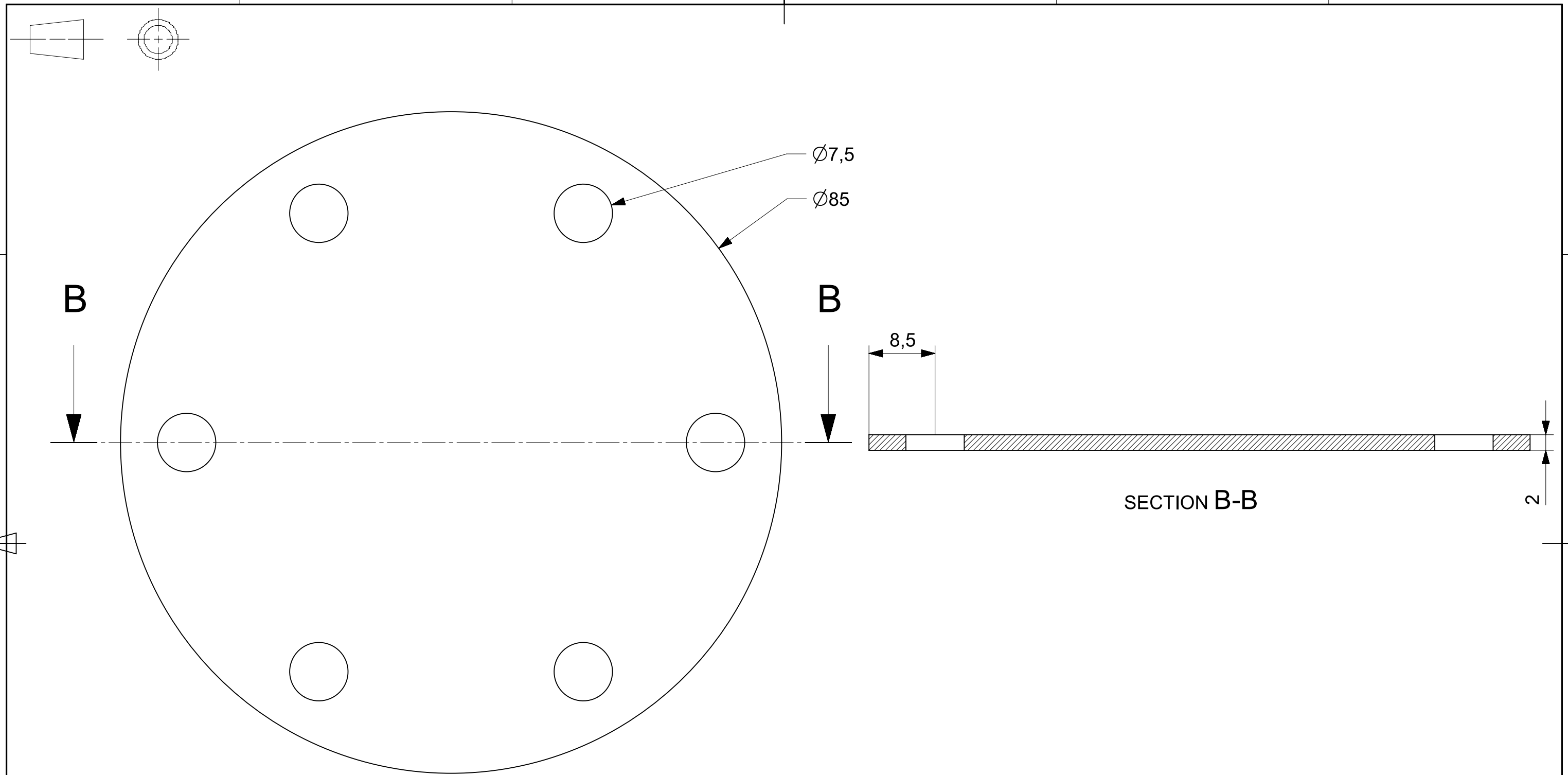
MATERIAL: PLEKSIKLAS ELLER ALUMINIUM



# TETNINGSFLATE

ISSUED DATE:	31.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	018	A
		SCALE 1:1		SHEET 1 OF 1

ALLE DIMENSJONER ER OPPGITT I MM



MATERIAL: ALUMINIUM UTEN VIDERE  
OVERFLATEFLATEBEHANDLING



# KJØLEPLATE

ISSUED DATE:	12.05.16	SIZE	DRW NR	SHEET REV
DRAWN BY:	POD	A3	019	A
		SCALE 2:1	SHEET 1 OF 1	

ALLE DIMENSJONER ER OPPGITT I MM



# Vedlegg 6

## Matlabskript for beregning av spenningsfall

```

close all; clear all; clc;

% Figur, grenser og dimensjoner -----
SVRyLimDiffF = 1/3; % Spenning ved ROV, y faktor for nedre spenningsverdi
ELyLimH      = 1000; % Effekt landforsyning, øvre y grense i watt
EyLimL      = 50;   % Effektivitet, nedre y grense i prosent
maxP        = 600;  % Øvre effektgrense
stegPInkr   = 0.01; % Steglengde på effekt ved inkrementell utregning
stegPkont   = 0.1;  % Steglengde på effekt ved kontinuerlig utregning
plotL = {'-', '--', '-.'}; % Linjetyper for plotting
% Variabler og konstanter -----
l          = [50 60 70]; % Lengde på kabel i meter
rho        = 0.0172;    % Spesifikk motstand, kobber
A          = [1.5 2.5 4 6]; % Kabeltverrsnitt
U          = [30 48];    % Inn-spenning
EffKon48V = 0.93;      % Effektivitet buck converter nede ved ROV
EffKon30V = 0.90;      % Effektivitet buck converter nede ved ROV
lgdns     = cell(1,length(A)+length(l)); % legend strings

EffKon48V = 1;
EffKon30V = 1;
for f = 1:2
    if f == 1
        stegP = stegPInkr;
    else
        stegP = stegPkont;
    end
    Prov = 0:stegP:maxP; % Effektforsyning, land
    P     = Prov;        % Effektforsyning, land
    Urov  = Prov;        % Ut spenning
    I     = Prov;        % Strøm
    titler = [{' Inkrementell spenningsfallsberegning(' num2str(l)...
              'm) med oppløsning på: ' num2str(stegP) 'w']...
              ,[' Kontinuerlig spenningsfallsberegning(' num2str(l)...
              'm), med oppløsning på: ' num2str(stegP)...
              'w. Fungerer kun så lenge spenningsfallet < U/2']];

% Figur Oppsett -----
figure(f);
set(gcf, 'Position', [60 0 1920-60 1000], 'Name', titler{f});
% Setter linjefarger for plotting
set(gcf, 'DefaultAxesColorOrder', [0 0.4470 0.7410;...
    0.8500 0.3250 0.0980;0.4660 0.6740 0.1880;0.4940 0.1840 0.5560;]);
sub = tight_subplot(2,3,[0.08 0.03],[0.05 0.03],[0.03 0.01]);
index = [1 2 3; 4 5 6];

% Utregningsløkker -----

for s = 1:length(U) % Alle spenninger
    % Lager legend for forskjellige kabellengder
    for subplots = 1:3
        for lnr = 1:length(l)
            axes(sub(index(s,subplots)))
            plot(0,0,'k','LineStyle',plotL{lnr})
            hold on
            lgdns{lnr} = [num2str(l(lnr)) ' meter']; % Lag legend string
        end
    end
end

```

```

end
% Regner ut spenningsfall ved forskjellige kabellengder =====
for lnr = 1:length(l) % Alle lengder
    U1 = U(s); % Midlertidig ROV-spenning
    for i = 1:length(A) % Alle kabeltverrsnitt
        lgdns{length(l)+i} = [num2str(A(i)) ' mm^2']; % Lag legend string
        errIdx = 0;
        Rk = (2*(lnr)*rho)/A(i); % Kabelmotstand
        for n = 1:length(Prov) % Alle effekter
            if U(s) == 48
                tempProv = Prov(n)/Effkon48V; % Effekttap i buck converter
            else
                tempProv = Prov(n)/Effkon30V; % Effekttap i buck converter
            end
        % Inkrementell beregning av spenningsfall =====
            if f==1
                I(n) = tempProv / U1; % Kabelstrøm
                sf = Rk*I(n); % Spenningsfall
                Urov(n) = U(s) - sf; % ROV-spenning
                I(n) = tempProv / Urov(n); % Kabelstrøm etter spenningsfall
                P(n) = U(s)*I(n); % Effektforbruk land

                if sf >= U(s) % Logger antall ugyldige verdier
                    Urov(n) = Urov(n - 1);
                    I(n) = I(n - 1);
                    P(n) = P(n - 1);
                    errIdx = errIdx + 1;
                end
                U1 = Urov(n); % Setter nåverende ROV-spenning lik forrige

        % Kontinuerlig beregning av spenningsfall =====
            else
                % Formel for spenning ved ROV, gitt effekt ved ROV
                % Fungerer så lenge Prov < U^2/(4*Rk)
                Urov(n) = 0.5*(U(s) + sqrt(U(s)^2 - 4*tempProv*Rk));
                I(n) = tempProv / Urov(n);
                P(n) = U(s)*I(n);
                if tempProv > U(s)^2 / (4*Rk) % Forhindring av ugyldige verdier
                    Urov(n) = NaN;
                    P(n) = NaN;
                    I(n) = NaN;
                end
            end
        end
        if errIdx ~= 0
            Urov(length(Urov) - errIdx:length(Urov)) = NaN; % Sletter ugyldige
            verdier
            P(length(P) - errIdx:length(P)) = NaN; % Sletter ugyldige
            verdier
        end
    % Plotting -----
        axes(sub(index(s,1))) % (ROV-effekt,ROV-spenning)
        plot(Prov,Urov, 'LineStyle',plotL{lnr})
        hold on
        axes(sub(index(s,2))) % (ROV-effekt,land-effekt)
        plot(Prov,P, 'LineStyle',plotL{lnr})
        hold on
        axes(sub(index(s,3))) % (ROV effekt,Effektivitet)

```

```

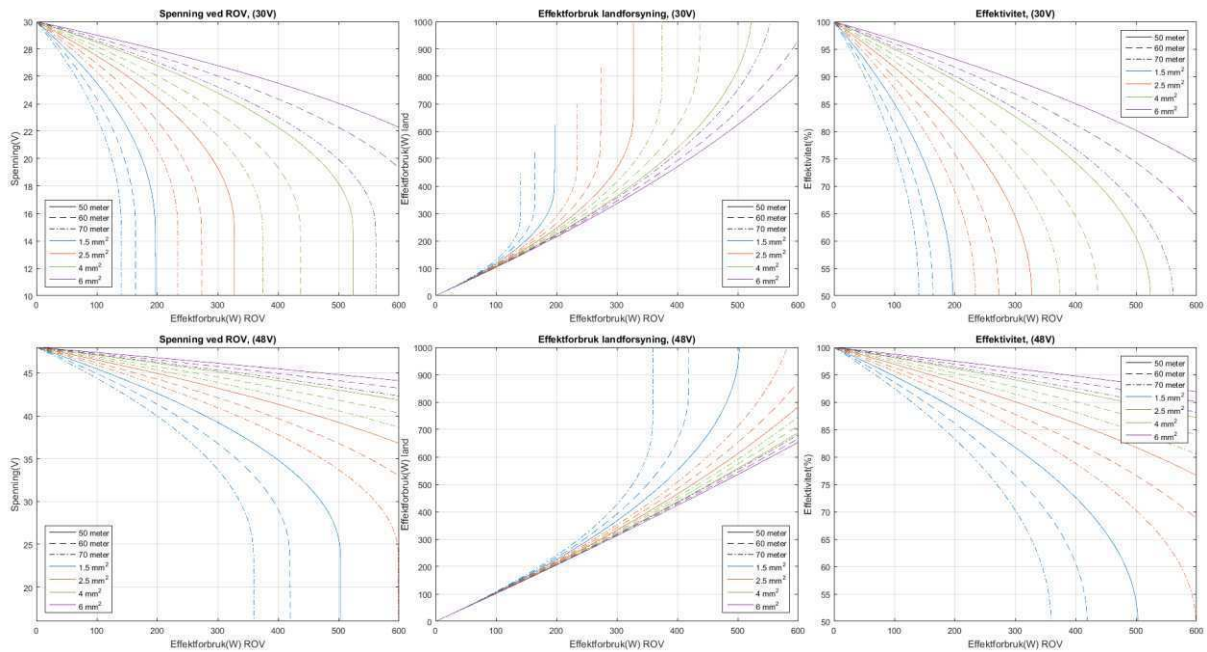
plot(Prov,Prov./P*100, 'LineStyle',plotL{1nr})
hold on
end
end
end
% Figurparametere -----
axes(sub(index(s,1)))
legend(lgdns,'Location','southwest')
title(['Spenning ved ROV, (' num2str(U(s)) 'v)'])
xlabel('Effektforbruk(w) ROV')
ylabel('Spenning(v)')
grid on
axis([0 maxP U(s)*SVRyLimDiffF U(s)])

axes(sub(index(s,2)))
legend(lgdns,'Location','southeast')
title(['Effektforbruk landforsyning, (' num2str(U(s)) 'v)'])
xlabel('Effektforbruk(w) ROV')
ylabel('Effektforbruk(w) land')
axis([0 maxP 0 ELYLimH])
grid on

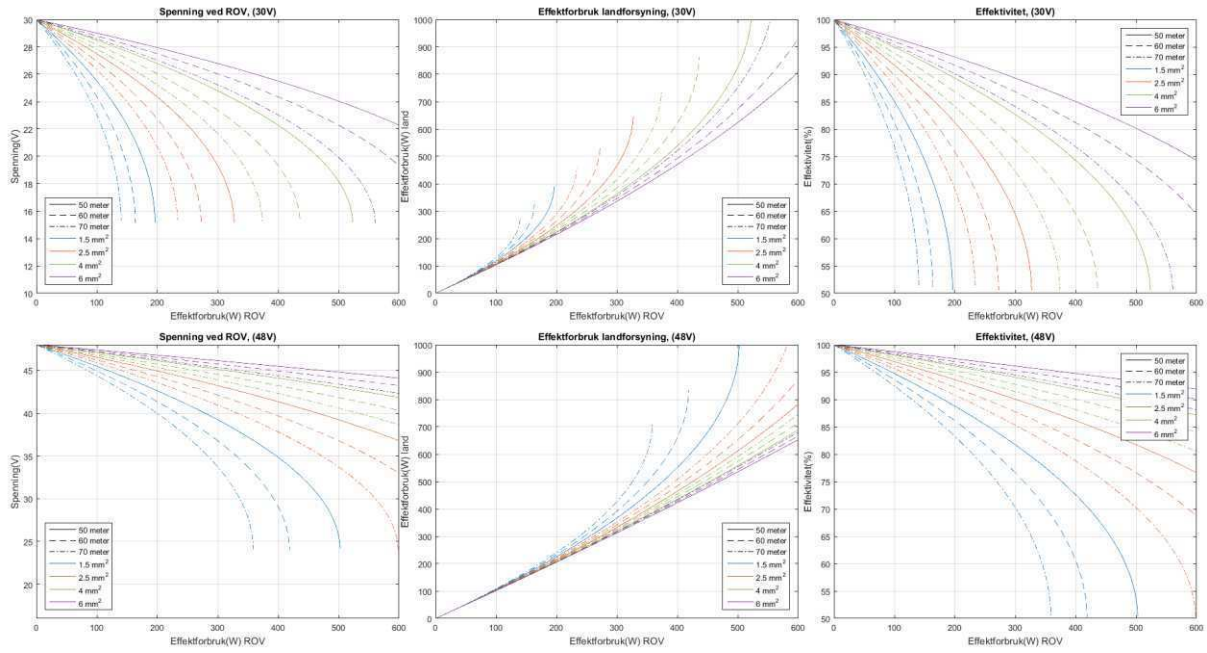
axes(sub(index(s,3)))
legend(lgdns,'Location','northeast')
title(['Effektivitet, (' num2str(U(s)) 'v)'])
xlabel('Effektforbruk(w) ROV')
ylabel('Effektivitet(%)')
axis([0 maxP EyLimL 100])
grid on
end
end
end

```

### Inkrementell utregning:

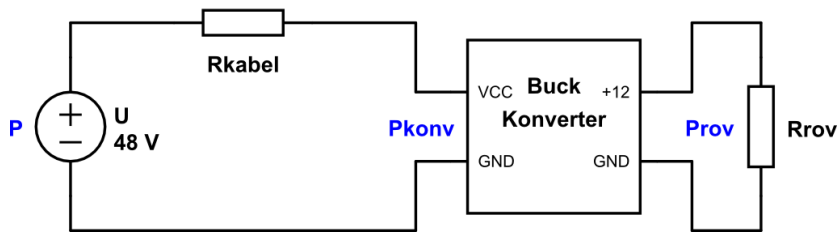


# Kontinuerlig utregning:



Published with MATLAB® R2015b

Vedlegg 7  
Utleddning av formel for spenning ved  
ROV



I bildet over kan man se et forenklet koblingskjema av forsyningsnettet til ROV-en.

$R_{kabel}$  er motstanden i kabelen og er gitt av formelen:

$$R_{kabel} = \frac{2 * l * \rho}{A}$$

Hvor «2» er antall ledere, « $l$ » er lengden på kabelen, « $\rho$ » er spesifikk motstand ( $\Omega * mm^2/m$ ), og « $A$ » er ledertverrsnitt i  $mm^2$ . [1]

For enkelhets skyld blir «kabel» referert til som « $k$ » og «konv» som « $l$ ».

$$U_l = \frac{P_l}{I}$$

$$U_l = \frac{P_l}{U(R_k + R_l)}$$

$$U_l = \frac{P_l}{U} \left( R_k + \frac{U_l^2}{P_l} \right)$$

$$U_l = \frac{P_l}{U} R_k + \frac{P_l U_l^2}{U P_l}$$

$$\frac{U_l^2}{U} - U_l + \frac{P_l R_k}{U} = 0$$

$$U_l^2 - U U_l + P_l R_k = 0$$

$$U_l = \frac{U}{2} \pm \frac{\sqrt{U^2 - 4P_l R_k}}{2}$$

Siden  $U_l$  aldri vil overstige  $U$  så vil  $U_l = \frac{U}{2} - \frac{\sqrt{U^2 - 4P_l R_k}}{2}$

$$U_l = \frac{1}{2} \left( U - \sqrt{U^2 - 4P_l R_k} \right)$$

$$P_{konv} * n = P_{ROV}, \quad n = \text{virkningsgraden til buck konverteren}$$

$$P_{konv} = P_l = \frac{P_{ROV}}{n}$$

$$U_{konv} = \frac{1}{2} \left( U - \sqrt{U^2 - 4 \frac{P_{ROV}}{n} R_{kabel}} \right), \text{ hvor } U > 0 \wedge R_{kabel} > 0 \wedge 0 < \frac{P_{ROV}}{n} < \frac{U^2}{4R_{kabel}}$$

Formel for motstand i kabel hentet fra:

Nek Kabel Norge AS

[http://nek-kabel.no/norsk/informasjon\\_1/teknisk/content\\_1/filelist\\_9b180c19-848b-4e3f-87e1-6b6c40fb7af/1395914263165/formler.pdf](http://nek-kabel.no/norsk/informasjon_1/teknisk/content_1/filelist_9b180c19-848b-4e3f-87e1-6b6c40fb7af/1395914263165/formler.pdf)

# Vedlegg 8

## Visualisering av kinematikk



```

close all; clear all; clc;
L1 = 3; L2 = 6; L3 = 6;

theta1 = 0*pi/180;
theta2 = 20.6248*pi/180;
theta3 = -114.6268*pi/180;
S1 = sin(theta1);
S2 = sin(theta2);
S3 = sin(theta3);
S23 = sin(theta2+theta3);
C1 = cos(theta1);
C2 = cos(theta2);
C3 = cos(theta3);
C23 = cos(theta2+theta3);
Pxd = 10;
Pyd = 0;
Pzd = 0;

x0 = 0;
y0 = 0;
z0 = 4;

gamma = 0*pi/180; %X-aksen
beta = 0*pi/180; % Y-aksen
alpha = 0*pi/180; % Z-aksen

P0 = [Pxd;Pyd;Pzd];

Rx = [1, 0, 0;
      0, cos(gamma), -sin(gamma);
      0, sin(gamma), cos(gamma)];

Ry = [cos(beta), 0, sin(beta);
      0, 1, 0;
      -sin(beta), 0, cos(beta)];

Rz = [cos(alpha), -sin(alpha), 0;
      sin(alpha), cos(alpha), 0;
      0, 0, 1];

Rrot = Rx*Ry*Rz;
P1 = Rrot*P0;

Pxd = Pxd + (P1(1)-Pxd);
Pyd = Pyd + (P1(2)-Pyd);
Pzd = Pzd + (P1(3)-Pzd);

Px_c = L1*cos(theta1)+ L2*cos(theta2)+L3*cos(theta1)*cos(theta2+theta3);
Py_c = L1*sin(theta1)+L2*sin(theta1)*cos(theta2)+L3*sin(theta1)*cos(theta2+theta3);
Pz_c = z0*cos(theta1)+L2*cos(theta1)*sin(theta2)+L3*cos(theta1)*sin(theta2+theta3);

error = [Pxd;Pyd;Pzd] - [Px_c;Py_c;Pz_c];
phi = 0.2;

while norm(error)>10^-3
% Jacobian matrisesen
Ja = [-L1*S1-L2*S1*C1-L3*S1*C23, -L2*C1*S2-L3*C1*S23, -L3*C1*S23];

```

```

L1*C1+L2*C1*C2+L3*C1*C23, -L2*S1*S2-L3*S1*S23, -L3*S1*S23;
-z0*S1-L2*S1*C2-L3*S1*S23, L2*C1*C2+L3*C1*C23, L2*C1*C23];

J_pseudo = Ja'/(Ja*Ja');

delta_theta = J_pseudo*error;

theta1 = theta1 + delta_theta(1);
theta2 = theta2 + delta_theta(2);
theta3 = theta3 + delta_theta(3);

Px_c = L1*cos(theta1)+L2*cos(theta1)*cos(theta2)+L3*cos(theta1)*cos(theta2+theta3);
Py_c = L1*sin(theta1)+L2*sin(theta1)*cos(theta2)+L3*sin(theta1)*cos(theta2+theta3);
Pz_c = z0*cos(theta1)+L2*cos(theta1)*sin(theta2)+L3*cos(theta1)*sin(theta2+theta3);

%Point 1
P1x = x0+L1*cos(theta1);
P1y = y0+L1*sin(theta1);
P1z = z0;

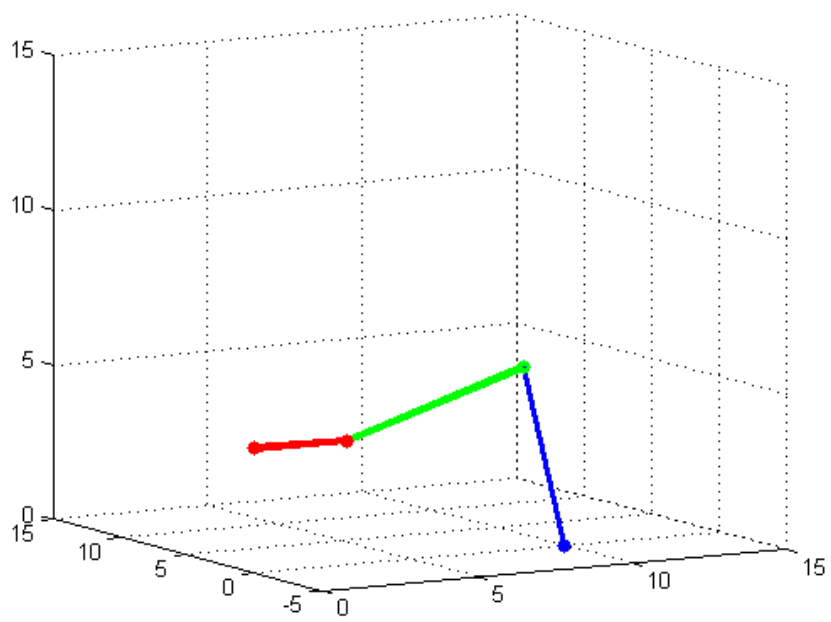
%Point 2
P2x = L1*cos(theta1)+L2*cos(theta1)*cos(theta2);
P2y = L1*sin(theta1)+L2*sin(theta1)*cos(theta2);
P2z = z0*cos(theta1)+L2*cos(theta1)*sin(theta2);

% Point 3
P3x = Px_c;
P3y = Py_c;
P3z = Pz_c;

%Plotting the leg
clf('reset')
view(-30,10)
hold on
plot3(x0,y0,z0,'r.','markersize',20)
plot3(P1x,P1y,z0,'r.','markersize',20)
plot3(P2x,P2y,P2z,'g.','markersize',20)
plot3(P3x,P3y,P3z,'b.','markersize',20)
line([x0,P1x],[y0,P1y],[z0,P1z],'linewidth',3,'color','red')
line([P1x,P2x],[P1y,P2y],[P1z,P2z],'linewidth',3,'color','g')
line([P2x,P3x],[P2y,P3y],[P2z,P3z],'linewidth',3,'color','b')
axis([0 15 -5 15 0 15])
grid on

error = [Pxd;Pyd;Pzd] - [Px_c;Py_c;Pz_c];
pause(0.001)
end

```



*Published with MATLAB® R2014a*

# Vedlegg 9

## Fremvisning av robotbein

```

close all; clear all;

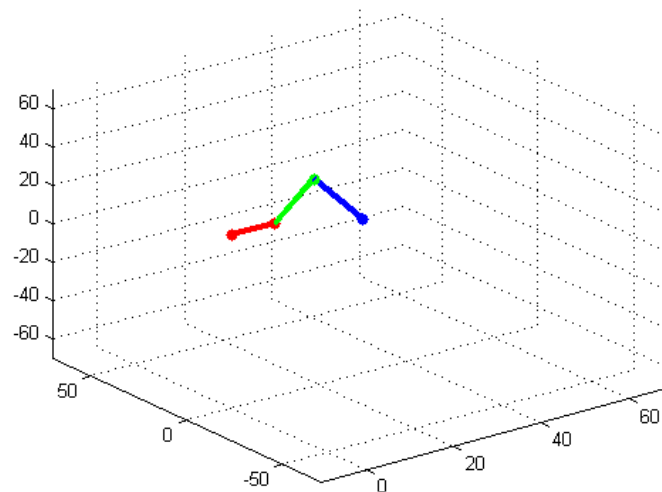
index = 0;
theta1 = 0*pi/180;
theta2 = 20.6248*pi/180;
theta3 = -114.6268*pi/180;
x = 0;
while(x<1)
    if(index == 0)
        [theta1,theta2,theta3] = InverseJacobian(30,15,0,theta1,theta2,theta3);
        index = 1;
    end

    if(index == 1)
        [theta1,theta2,theta3] = InverseJacobian(30,0,0,theta1,theta2,theta3);
        index = 2;
    end

    if(index == 2)
        [theta1,theta2,theta3] = InverseJacobian(30,-15,0,theta1,theta2,theta3);
        index = 3;
    end

    if(index == 3)
        [theta1,theta2,theta3] = InverseJacobian(30,0,10,theta1,theta2,theta3);
        index = 0;
    end
    x = x+1;
end

```



*Published with MATLAB® R2014a*

Vedlegg 10  
Rotasjonsmatrise-funksjon til  
fremvisning av robotbein

```

function [ P1 ] = RotationMatrix( Pxd, Pyd, Pzd, gamma, beta, alpha )
%UNTITLED2 Summary of this function goes here
% Detailed explanation goes here

gamma = 0*pi/180; %X-aksen
beta = 0*pi/180; % Y-aksen
alpha = 0*pi/180; % Z-aksen

P0 = [Pxd;Pyd;Pzd];

Rx = [1, 0, 0;
      0, cos(gamma), -sin(gamma);
      0, sin(gamma), cos(gamma)];

Ry = [cos(beta), 0 , sin(beta);
      0 , 1 , 0;
      -sin(beta), 0, cos(beta)];

Rz = [cos(alpha), -sin(alpha), 0;
      sin(alpha), cos(alpha), 0;
      0, 0, 1];

Rrot = Rx*Ry*Rz;
P1 = Rrot*P0;

end

```

*[Published with MATLAB® R2014a](#)*

Vedlegg 11  
Inverse Jacobian funksjon til  
fremvisning av robotbein



```

function [theta1,theta2,theta3 ] = InverseJacobian( Pxd,Pyd,Pzd,theta1,theta2,theta3 )
%UNTITLED4 Summary of this function goes here
% Detailed explanation goes here

L1 = 10; L2 = 20; L3 = 30;

% theta1 = 0*pi/180;
% theta2 = 20.6248*pi/180;
% theta3 = -114.6268*pi/180;
% theta1 = 0;
% theta2 = 0;
% theta3 = 0;
S1 = sin(theta1);
S2 = sin(theta2);
S23 = sin(theta2+theta3);
C1 = cos(theta1);
C2 = cos(theta2);
C23 = cos(theta2+theta3);
x0 = 0;
y0 = 0;
z0 = 20;

Px_c = L1*cos(theta1)+L2*cos(theta1+theta2)+L3*cos(theta1)*cos(theta2+theta3);
Py_c = L1*sin(theta1)+L2*sin(theta1)*cos(theta2)+L3*sin(theta1)*cos(theta2+theta3);
Pz_c = z0*cos(theta1)+L2*cos(theta1)*sin(theta2)+L3*cos(theta1)*sin(theta2+theta3);

error = [Pxd;Pyd;Pzd] - [Px_c;Py_c;Pz_c];
phi = 0.1;

while norm(error)>10^-1

Ja = [-L1*S1-L2*S1*C1-L3*S1*C23, -L2*C1*S2-L3*C1*S23, -L3*C1*S23;
      L1*C1+L2*C1*C2+L3*C1*C23, -L2*S1*S2-L3*S1*S23, -L3*S1*S23;
      -z0*S1-L2*S1*C2-L3*S1*S23, L2*C1*C2+L3*C1*C23, L2*C1*C23];

J_pseudo = Ja'/(Ja*Ja');

delta_theta = phi*J_pseudo*error;

theta1 = theta1 + delta_theta(1);
theta2 = theta2 + delta_theta(2);
theta3 = theta3 + delta_theta(3);

Px_c = L1*cos(theta1)+L2*cos(theta1)*cos(theta2)+L3*cos(theta1)*cos(theta2+theta3);
Py_c = L1*sin(theta1)+L2*sin(theta1)*cos(theta2)+L3*sin(theta1)*cos(theta2+theta3);
Pz_c = z0*cos(theta1)+L2*cos(theta1)*sin(theta2)+L3*cos(theta1)*sin(theta2+theta3);

%Point 1
P1x = x0+L1*cos(theta1);
P1y = y0+L1*sin(theta1);
P1z = z0;

%Point 2
P2x = L1*cos(theta1)+L2*cos(theta1)*cos(theta2);
P2y = L1*sin(theta1)+L2*sin(theta1)*cos(theta2);

```

```

P2z = z0*cos(theta1)+L2*cos(theta1)*sin(theta2);

% Point 3
P3x = Px_c;
P3y = Py_c;
P3z = Pz_c;

%Plotting the leg
clf('reset')
view(3)
hold on
plot3(x0,y0,z0,'r.','markersize',20)
plot3(P1x,P1y,z0,'r.','markersize',20)
plot3(P2x,P2y,P2z,'g.','markersize',20)
plot3(P3x,P3y,P3z,'b.','markersize',20)
line([x0,P1x],[y0,P1y],[z0,P1z],'linewidth',3,'color','red')
line([P1x,P2x],[P1y,P2y],[P1z,P2z],'linewidth',3,'color','g')
line([P2x,P3x],[P2y,P3y],[P2z,P3z],'linewidth',3,'color','b')
axis([-10 70 -70 70 -70 70])
grid on

error = [Pxd;Pyd;Pzd] - [Px_c;Py_c;Pz_c];
pause(0.01)
end

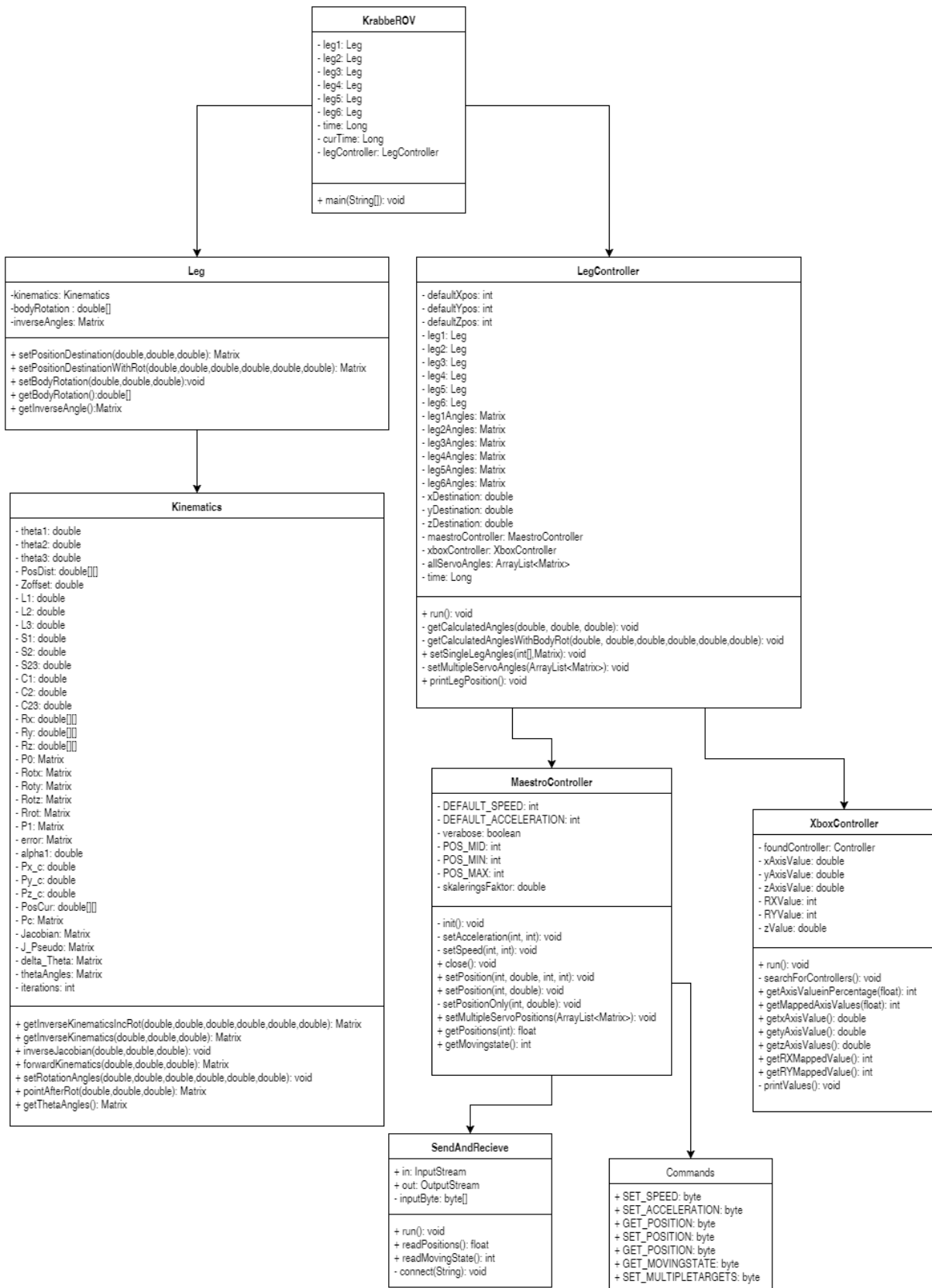
end

```

*[Published with MATLAB® R2014a](#)*

# Vedlegg 12

## Klassediagram for styring av bein



# Vedlegg 13

## Klassediagram for serverapplikasjon

