



CHALMERS



Fjärrstyrning av RescueRunner

I samarbete med Svenska Sjöräddningssällskapet

Kandidatarbete EENX15-18-43

Daniel Bäck, Max Christensen, Jonathan Hansson, Andreas Henriksson, Olav Ravndal Skjølingstad och Filip Ödeen

Institutionen för Elektroteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2018

KANDIDATARBETE 2018

Fjärrstyrning av RescueRunner

I samarbete med Svenska Sjöräddningssällskapet

Daniel Bäck
Max Christensen
Jonathan Hansson
Andreas Henriksson
Olav Ravndal Skjølingstad
Filip Ödeen

Handledare: Boaz Ash
Examinator: Petter Falkman

Omslagsbild
Bilden på framsidan skildrar en RescueRunner med fjärrstyrning



CHALMERS

Kandidatarbete EENX15-18-43
Institutionen för Elektroteknik
System- och reglerteknik
CHALMERS TEKNISKA HÖGSKOLA
Göteborg, Sverige 2018

Sammanfattning

Svenska Sjöräddningssällskapet startades som en ideell verksamhet för att rädda liv till havs, och har idag över 70 stationer i Sverige med personal redo att rycka ut inom 15 minuter. För att öka flexibiliteten vid olycksplatser har RescueRunnern utvecklats, en liten och smidig vattenskoter, som lätt kan ta sig mellan hinder i vattnet. Problemet med vattenskotern är transporten från utryckningsstationen till olycksplatsen, ett problem som detta projekt syftar till att bidra med en del lösning på.

Målet med projektet är att sammanställa lösningarna från tidigare projekt och vidareutveckla samt förbättra fjärrstyrning av en RescueRunner. Till stor del har målen som sattes upp för projektet uppfyllts. Systemet, vilket reglerar styrvinkeln samt växlingen mellan manuellt och fjärrstyrt läge, uppfyller majoriteten av kraven som ställdes. Hastighetsregleringen uppfyllde däremot inte alla krav vilket delvis beror på svår genomförd testning av lösningar.

Abstract

The Swedish Sea Rescue Society was founded as a non-profit organisation to save lives at sea. Today it has more than 70 stations around Sweden with volunteers ready to depart for rescue missions within 15 minutes. The RescueRunner, a small and adaptive jet-ski, was developed to increase flexibility at the scene of the accident. The problem with this jet-ski is the transportation from the station to the scene of the accident, a problem that the project aims to help find a partial solution to.

The goal of the project is to compile solutions from earlier projects and to develop and improve remote control functionality of a RescueRunner. Overall, the goals of the project have mostly been achieved. The system that controls the steering angle and switching between manual mode and remote control mode fulfills a majority of the requirements set. The acceleration control did not achieve all the goals, which is partly due to challenging testing conditions.

Tack till

Tack till Fredrik Falkman på SSRS för utlåning av och hjälp med båten. Vi vill även tacka Reine Nohlborg på Chalmers prototypplabb för hjälp med tillverkning av komponenter. Slutligen stort tack till handledare Boaz Ash, REVERE och Chalmers Robotförening för konsultation, verktyg och material.

Innehåll

1	Inledning	1
1.1	Syfte	1
1.2	Avgränsningar	1
1.3	Tidigare projekt	2
1.4	Projekt mål	3
2	Reglering av styrvinkel	4
2.1	Kravspecifikation för styrning	5
2.2	Konceptval	5
2.2.1	Koncept	6
2.2.2	Val av koncept	7
2.2.3	Återkoppling av styrvinkeln	8
2.3	Mätning av vridmoment på styret	8
2.4	Val av komponenter	10
2.4.1	Val av elmotor	10
2.4.2	Val av kraftelektronik	11
2.4.3	Val av koppling	12
2.5	Realisering	12
2.5.1	Montering av motor och cykelbroms	13
2.5.2	Inställning av motorns styrkort	14
2.5.3	Återkoppling av styrvinkel	15
3	Hastighetsreglering	17
3.1	Kravspecifikation	17
3.2	Konceptval	18
3.2.1	Koncept	18
3.2.2	Val av koncept	19
3.3	Uppmätning av APS	20
3.4	Uppmätning av digital potentiometer	21
3.5	Matematisk modell för gaspådrag	22
3.6	Realisering	23
4	Elektronik	24
4.1	Mikrokontrollerkort	25
4.2	Radiokommunikation	25
4.3	SD-kort	26

4.4	PS2-kontroller	27
4.5	Motorstyrkort	28
4.6	Relä	28
4.7	Digital potentiometer	29
4.8	Skapandet av sköld	30
4.9	Strömförsörjning och nödstopp	30
5	Mjukvara	32
5.1	PS2-kontroll	32
5.2	SD-kort	33
5.3	Styrvinkelgivare	34
5.4	PID-regulator	34
5.5	Radiokommunikation	37
5.6	Debug och loggning	39
6	Resultat	40
6.1	Reglering av styrvinkel	41
6.2	Hastighetsreglering	42
7	Diskussion	44
7.1	Reglering av styrvinkel	44
7.2	Hastighetsreglering	45
7.3	Elektronik	46
7.4	Vidareutveckling	46
7.4.1	Reglering av styrvinkel	46
7.4.2	Hastighetsreglering	47
7.4.3	Dataöverföring	47
7.4.4	Radoräckvidd	48
7.4.5	Dataloggning	48
7.4.6	Följningsfunktion	48
	Referenser	50
A	Stora figurer	I
B	Komponentlista	X

Ordlista

Allmänt

SSRS Svenska Sjöräddningssällskapet

Styrning

Dysa Enhet som riktar vattenstrålen för att kunna ändra riktning på RescueRunnern

Linjär aktuator En elektromekanisk komponent som producerar en linjär rörelse, till skillnad från en elektrisk motor, vilken producerar en cirkulär rörelse

Hastighetsreglering

APS Accelerator Position Sensor

ECM Electronic Control Module, även kallad Enginge Cotrol Unit, är den enhet som styr alla signaler på RescueRunnern, det vill säga dess huvuddator

Elektronik

CS Chip Select – används av Arduinon för att bestämma vilken av periferienheterna som ska köras på SPI-bussen. Kan på andra håll även kallas för SS (Slave Select) eller CE (Chip Enable)

NC Normally Closed, normalt stängd. Är inom elektronik sluten krets vid opåverkat läge

NO Normally Open, normalt öppen. Är inom elektronik bruten krets vid opåverkat läge

Pull-up-resistans Ett motstånd som säkerställer att en signals binära logikvärden är väldefinierade mellan hög och låg. Pull-up håller signalen hög när den inte används.

RX Reciever, mottagare

Sköld Ett lager av komponenter och kopplingar som ansluts över en Arduino

SPI Serial Peripheral Interface – En speciell buss på Arduino som behövs för många periferienheter. Kommunicerar via MISO-, MOSI- och CLK-pinnarna på Arduino

TX Transmitter, sändare

1

Inledning

1907 grundades organisationen som idag heter Svenska Sjöräddningssällskapet (SSRS). Den startades som en ideell verksamhet för att rädda liv till havs. Längst med landets kuster, inklusive vid de största sjöarna, finns över 70 stationer med personal redo att rycka ut inom 15 minuter [1].

Vid utryckning är större båtar bättre på att klara av grövre sjöväder, medan mindre båtar, med enklare manövrering och låga fribord, är bättre på att vara en del i det faktiska räddningsarbetet. Med anledning av detta har SSRS utvecklat RescueRunnern, en liten och flexibel vattenskoter. Att få ut en mindre båt till olycksplatsen i oväder innebär dock en styrkemässig utmaning för dess förare; krafter som hade kunnat sparas till själva livräddningen.

De flesta av SSRS:s större båtar är inte stora nog att rymma en RescueRunner ombord. Dessutom riskerar RescueRunnern att under påfrestningar av för stora krafter slås sönder vid eventuell bogsering. Med ovanstående anledningar i åtanke, används den därför i stort sett enbart i utbildningssyfte. Att bygga in stöd för autonom följning av den större ledarbåten skulle möjliggöra att besättningen kan spara sina krafter, samtidigt som de har nytta av en liten och flexibel båt under räddningsoperationerna. För att kunna utveckla denna följningsfunktion måste RescueRunnern först kunna styras elektroniskt på ett snabbt och pålitligt sätt.

1.1 Syfte

Syftet med rapporten är att beskriva och diskutera projektets lösningar på en fjärrstyrningsfunktion för RescueRunnern. Rapporten beskriver i detalj varje dellösning på ett sådant sätt att det ska vara lätt för ett efterkommande projekt att ta efter och vidareutveckla systemet.

1.2 Avgränsningar

För att projektet ska vara genomförbart under den planerade tidsramen har följande avgränsningar gjorts:

- Projektet fokuserar endast på fjärrstyrning av RescueRunnern och inte en komplett följningsfunktion.
- Systemet testas endast under goda väderförhållanden utan kraftig nederbörd, hinder eller yttre störningar.
- Inga genomgående stresstester av systemet utförs för att testa exempelvis resistans mot vatten eller slitage.
- Projektet kommer inte undersöka eller implementera styrning av backfunktionen på RescueRunnern.

1.3 Tidigare projekt

Tidigare har det bland annat skett två kandidat- och två mastersarbeten inom området [2]–[5]. Det ena kandidatarbetets lösning för styrningen innefattade att ansluta en motor på det befintliga mekaniska systemet med en remkoppling [2]. Enligt rapporten gav lösningen fördelar så som möjligheten att kunna koppla bort det nya systemet och enbart köra RescueRunnern manuellt. Nackdelen var att systemet var långsamt och utan särskilt hög precision [2].

Det andra kandidatarbetets lösning gick å andra sidan ut på att den mekaniska kopplingen mellan styret och dysan togs bort, och istället kopplades de elektroniskt ihop med en vinkelgivare och en linjär aktuator som vrider dysan [3]. En linjär aktuator är en elektromekanisk komponent, vilken producerar en linjär rörelse, till skillnad från en elektrisk motor som producerar en cirkulär rörelse. Detta innebar att styrningen blev mjuk och snabb, men med nackdelen att om det nya elektriska systemet av någon anledning inte skulle fungera, skulle RescueRunnern inte gå att manövrera varken i autonomt eller i manuellt läge [3].

Två tidigare lösningar har även tagits fram för att reglera gasen. Den första skapades i kandidatarbetet från 2015 [2]. Lösningen använde en digital potentiometer som kontrollerades med en Arduino och gav ett snabbt och finjusterbart resultat [2]. Lösningen blev beskriven som välfungerande och uppfyllde deras kravspecifikation [2].

Den andra lösningen togs fram under kandidatarbetet 2016 [3]. RescueRunnern som användes i detta projektet var baserad på en äldre vattenskoter där gasspjället öppnas med vajer, som är kopplad direkt till gashandtaget. Lösningen använde en elmotor, vilken drog i en vajer som var kopplad på gasspjället [3].

Medan kandidatarbetena fokuserade på hårdvaran och elektroniken för manövreringen av RescueRunnern, fokuserade masterarbetena på algoritmerna och mjukvaran för att få den att följa efter ledarbåten [4], [5]. Masterarbetena behöver därför inte detaljstuderas för detta projekt. Utöver dessa arbeten utfördes ett ytterligare, mindre, projekt 2017, som mestadels fokuserade på systemidentifiering och modellering av RescueRunnern [6].

1.4 Projektmål

Målet med projektet är att förbättra växlingen mellan manuellt och fjärrstyrt läge. Detta innebär att skapa en extern styrning som är snabb och exakt nog, samtidigt som den manuella styrningen inte berörs. För att kunna testa systemet ska en lösning för fjärrstyrning tas fram. Då ett robust system är önskvärt, kommer komponentval och realisering göras med hänsyn till den tuffa miljön systemet kommer utsättas för. Efter att detta projekt har avslutats kommer ett mastersarbete att pågå som syftar till att implementera följningsalgoritmen.

Kravspecifikationen i tabell 1.1 skapades för att lösningen som utvecklas i projektet ska stämma överens med uppdragsgivarens förväntningar. Specifikationen används under projektets gång för att motivera val av lösningar och slutligen för att undersöka projektets resultat.

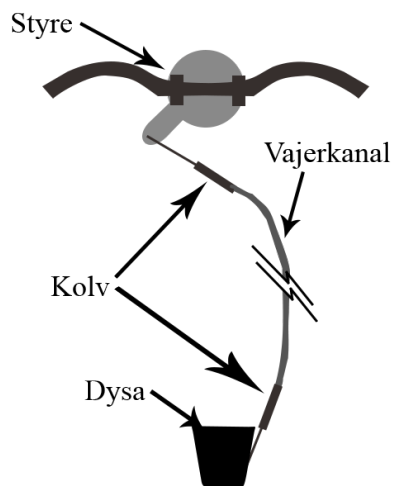
Tabell 1.1: Generell kravspecifikation för projektet

Kriterier	Kontrollmetod	Målvärde	Krav/ önskemål
Växling mellan manuell körning och fjärrstyrning	Test och slutligen godkännande av Fredrik Falkman (SSRS)	Ja	Krav
Ej påverka manuell körning märkbart	Test och slutligen godkännande av Fredrik Falkman (SSRS)	Ja	Krav
Komponenter med kapslingsklassning	Komponenterna som köps in skall uppfylla denna kapsling	Minst IP67	Önskemål
Räckvidd på trådlös kommunikation	Mätning	≥ 50 m	Krav
Räckvidd på trådlös kommunikation	Mätning	≥ 200 m	Önskemål

2

Reglering av styrvinkel

Styrningen regleras genom en styrvajer vars ena ände är fäst på höger sida av styrhandtaget på RescueRunnern och den andra änden är fäst i dysan. Dysan riktar vattenstrålen som driver RescueRunner framåt, och genom att rikta denna svänger RescueRunnern. På insidan av skotern är en del av styrvajern exponerad utanför skyddshöljet och istället fäst över en metallstång. Här finns möjlighet för yttre påverkan på styrningen. Se figur 2.2 nedan.



Figur 2.1: Illustration av styrningsmekanism. Från [7]. Återgiven med tillstånd.



Figur 2.2: Exponerad styrvajer

Detta kapitlet beskriver lösningsgången för att reglera styrvinkeln elektroniskt och hur en växling mellan elektroniskt och manuellt läge kan se ut.

2.1 Kravspecifikation för styrning

Utifrån projektmålet i avsnitt 1.4 och de tidigare kanidiatrapporterna togs kravspecifikationen i tabell 2.1 fram [2], [3]. Målvärdena för kraven och önskemålen bestämdes genom att överväga hur en människa kör en vattenskoter, och vilka hastigheter och egenskaper som krävs för att manövrera den.

Tabell 2.1: Kravspecifikation för styrning av dysa

Kriterier	Kontrollmetod	Målvärde	Krav/önskemål
Styra vinkeln i hela dess register	Mätning mot manuell styrning	95% av manuell	Krav
Tid mellan ändlägen	Tidtagning	≤ 1 s	Krav
Tid mellan ändlägen	Tidtagning	$\leq 0,5$ s	Önskemål
Noggrannhet	Mätning	± 5 procentenheter	Krav
Noggrannhet	Mätning	± 1 procentenhet	Önskemål

2.2 Konceptval

För att komma fram till en bra och realiserbar lösning har tre olika delar beaktats: reglering av styrvinkel, växling mellan manuellt och fjärrstyrt läge och återkoppling av styrvinkeln.

Tre sätt undersöktes för att med en roterande kraft påverka styrvinkeln. Det första sättet innebär att en elmotor med koppling placeras i anslutning till dysan och påverkar denna direkt. Det andra alternativet är att en motor kopplas på under styret. Det tredje alternativet går ut på att en kuggstång eller lindat band eller liknande påverkar den exponerade vajern inuti skotern.

På samma sätt undersöktes tre sätt för att med en linjär kraftkälla påverka styrvinkeln. Det första sättet innebär att dysan påverkas direkt av en linjär kraft, liknande hur vajern vrider på denna. Det andra alternativet är att placera en hävarm under styret som sedan påverkas med en linjär rörelse. Det tredje alternativet går ut på att en linjär aktuator monteras på den exponerade vajern mellan styret och dysan.

För att det på ett pålitligt sätt ska gå att koppla till och från styrsystemet, det vill säga att växla mellan manuellt och fjärrstyrt läge, krävs det någon form av hakning eller koppling. Det finns några olika sätt att lösa detta på.

Inbyggd koppling

En motor eller linjär aktuator med inbyggd frångkoppling används. Exempel på sådana produkter är drivenheter till autopiloter på fritidsbåtar [8].

Magnetkoppling

En elektromagnet monteras på både den fjärrstyrda manövreringskraften och valt styrningsobjekt.

Hasp eller sprint

En enkel mekanisk lösning låser ihop den fjärrstyrda manövreringskraften och styrningsobjektet genom exempelvis en införd stång.

Remkoppling

En rem kopplas mellan motorn och det den ska driva, som kan spännas antingen via en spak eller en aktuator. Denna lösning fungerar enbart för roterande påverkan.

Skivbroms

En enkel mekanisk lösning låser ihop den fjärrstyrda manövreringskraften och styrningsobjektet.

2.2.1 Koncept

Det är inte alla tidigare listade dellösningförslag för styrvinkelsreglering som är genomförbara i praktiken. Till exempel att påverka dysan direkt hade krävt mycket arbete och systemet hade i slutändan ej varit robust på grund av saltvattnet. Ovansidan av skotern är redan relativt skyddad från väder och vågor så att koppla sig på styrningen på vajern eller styret är därmed både lättare och mer robust. Nedan listas ett antal lösningförslag.

1. “Drive by wire”

“Drive by wire” är en lösning där det elektriska systemet aldrig kopplas bort. Även i det manuella läget mäts den önskade styrvinkeln på styret och systemet reglerar själv vinkeln med till exempel en aktuator. Denna lösning kommer från 2016 års arbete [3].

2. Linjär aktuator

Ett alternativ vilket kommer ganska naturligt och är relativt lättimplementerat är att koppla in sig på vajern under huven. Aktuatorskoppling till och från med någon form av hasp eller krok som fästs på vajern. En till- och frångkoppling kan i detta fallet bara ske i ett fast läge.

3. Elmotor under styret

En roterande kraft placeras direkt under styret på RescueRunnern. Kraftkällan kopplas till och från med någon form av koppling. Denna lösning kan kopplas till och från i fler lägen, beroende på vilken typ av koppling som väljs.

4. Drivenhet från autopilot

En färdig lösning tas från en autopilot till större båtar. Sådana autopilot-enheter kommer oftast i formen av en linjär aktuator med inbyggd bortkoppling [8].

5. Hydraulik

Eftersom en aktuator är komplicerad att koppla bort och har högt motstånd, uppkom denna liknande lösningen. En hydra cylinder har ett mycket lågt motstånd i öppet läge och behöver inte kopplas av och på för att byta mellan manuell och automatisk styrning. En magnetventil kan användas för att reglera trycket i hydra cylindern. Om cylindern då är kopplad på styrvajern kan denna kontrolleras helt via en magnetventil.

2.2.2 Val av koncept

Med hjälp av ovanstående lösningsförslag skapades Pughmatrisen i tabell 2.2.

Tabell 2.2: Pughmatris styrreglering

Kriterier	Viktning	Koncept				
		1*	2	3	4	5
Påverkan i manuellt läge	5	0	+1	+1	+1	+1
Antal komponenter	3	0	-1	-1	+1	-1
Genomförbarhet	5	0	-1	-1	+1	-1
Noggrannhet	4	0	0	0	0	0
Tid mellan ändlägen	5	0	0	0	-1	0
Pris	1	0	0	0	-1	0
Hållbarhet	5	0	-1	0	0	0
Summa		0	-8	-3	7	-3

*: Referenskoncept

Enligt Pughmatrisen i tabell 2.2 var det fjärde lösningsförslaget, drivenhet från autopilot, det bästa konceptet. Denna lösningen är väldigt åtråvärd i sin simplicitet. Dessa produkter är dock gjorda för att hålla större båtar på rak kurs och behöver därför inte kunna svänga speciellt hastigt. Systemet på RescueRunnern måste däremot, enligt kravspecifikationen 2.1, kunna svänga väldigt snabbt och då är denna lösningen ej tillräckligt snabb [8].

Den näst bästa lösningen var referenskonceptet “Drive by wire”. Denna lösningen är inte tillåten enligt kraven från SSRS, som kan ses i kravspecifikationen enligt tabell 1.1, då den manuella styrningen måste vara helt mekanisk. “Drive by wire”-förslaget valdes därför bort.

Det femte konceptet, hydraulik, valdes bort på grund av bristfällig kunskap om hydraulik och liknande system då det på grund av detta ej hade varit genomförbart inom projektets tidsram.

Därför valdes istället det tredje konceptet, elmotor under styret, eftersom det anses vara det mest praktiska att implementera. Monteringen av en sådan kraftkälla är svårare än för en aktuator, medan en till- och fränkoppling av ett sådant system däremot är lättare.

Till en början var det en industrikoppling som skulle koppla till och från motorn. En IP-klassad sådan hade varit det absolut mest robusta och tillförlitliga systemet. Industrikopplingar är dock både dyra och svåra att få tag på. Därför prövades en billigare ersättningslösning, nämligen en skivbroms tillverkad för en cykel.

2.2.3 Återkoppling av styrvinkeln

För att mäta styrvinkeln valdes en lösning från ett tidigare år [6]. En fiskelina lindades upp på en fjädrad rulle som satt på en potentiometer. Rullen monterades sedan på den exponerade delen av styrvajern inuti båten som avbildas i figur 2.2. När styret roterar, drar vajern i fiskelinan och potentiometerns läge ändras.

Denna lösningen var inte robust tidigare år, och fiskelinan kunde därför lätt hoppa ur rullens spår enligt Alexander Lydh, som arbetade med projektarbet 2017. Lösningen var att tillverka en större rulle och göra spåret djupare än på den förra.

Ett annat problem som fanns med denna lösningen var att fjädern vilken valts till potentiometern var av dålig kvalitet. Konsekvensen blir att fjädern töjs ut om båten lämnas i ett läge där fjädern är utsträckt. Det leder till att fiskelinan inte kommer att rullas ihop vid svängning om båtens styre lämnats riktat åt vänster under en längre tid. Följaktligen kommer systemet då inte heller längre kunna känna av högersvängar. Lösningen på detta var att byta fjäder.

2.3 Mätning av vridmoment på styret

För att beräkna styrets vridmoment mättes kraften vilken krävdes för att vrida på det. Kraften mättes både vid höger- och vänstersväng fem gånger vardera för att få ett representativt mått. En enklare dynamometer i form av en bagagevåg användes. Därav är krafterna angivna i kilogram istället för newton. Ett medelvärde beräknades sedan både för sväng åt höger och vänster, se tabell 2.3.

Tabell 2.3: Uppmätning av kraft för att vrida på styret

Mätning nr	Vänster→höger [kg]	Höger→vänster [kg]
1	7	2,5
2	6	2,4
3	4	2,3
4	6	2,3
5	6	2,4
Medel	5,8	2,4

Då det var stor skillnad på kraften som krävdes för att svänga åt höger gentemot vänster, användes krafterna uppmätta för sväng åt höger vid dimensionering av delarna i systemet, då dessa var större. Klarar systemet att svänga åt höger kommer det därmed även klara att svänga åt vänster. För att omvandla krafterna till newton användes formel (2.1).

$$F = m \cdot g \quad (2.1)$$

där m är massan (alltså det vågen visade) och g är tyngdaccelerationen, 9,81 N/kg. Den maximala kraften som mättes upp var $m_{max} = 7$ kg och medelkraften för sväng åt höger var $m_{medel} = 5,8$ kg enligt tabell 2.3. Detta ger maximala kraften

$$F_{max} = m_{max} \cdot g = 7 \cdot 9,81 = 68,67 \approx 69 \text{ N} \quad (2.2)$$

och medelkraften för sväng åt höger

$$F_{medel} = m_{medel} \cdot g = 5,8 \cdot 9,81 = 56,898 \approx 57 \text{ N} \quad (2.3)$$

Kraften uppmättes längst ut på styret. För att beräkna momentet som krävs för att vrida på styret mättes därför längden på styret till 0,68 m. Då mätningen genomfördes längst ut på styret blir alltså hävstången, d , hälften av det.

$$d = \frac{0,68}{2} = 0,34 \text{ m} \quad (2.4)$$

Momentet kunde då beräknas med formel (2.5).

$$M = d \cdot F \quad (2.5)$$

Detta gav det maximala momentet

$$M_{max} = d \cdot F_{max} = 0,34 \cdot 68,67 = 23,3478 \approx 23 \text{ N m} \quad (2.6)$$

och medelmomentet för sväng höger

$$M_{medel} = d \cdot F_{medel} = 0,34 \cdot 56,898 = 19,34532 \approx 19 \text{ N m} \quad (2.7)$$

2.4 Val av komponenter

I följande kapitel presenteras arbetsgången för valet av komponenter till lösningen som valts.

2.4.1 Val av elmotor

Vid valet av elmotor till styrningen behövde ett flertal av kraven och önskemålen beaktas. Från den generella kravspecifikationen i tabell 1.1 var det enbart önskemålet om kapslingsklass IP67 som direkt påverkar valet av motor.

Från kravspecifikationen för styrning av dysan i tabell 2.1 påverkade kravet och önskemålet för "Tid mellan ändlägen" direkt valet av motorn då de kunde översättas till ett krav på varvtalet för den. För att omvandla detta krav och önskemål från ett tidskrav till ett varvtalskrav användes formel (2.8).

$$n = \frac{\varphi \cdot (60 \text{ s/min})}{t \cdot 360^\circ} \quad (2.8)$$

där n är varvtalet i rpm, φ är utslaget på styret i grader och t är tiden i sekunder det får ta att svänga mellan ändlägena. φ uppskattades till 90° och kravet gav $t_{krav} = 1 \text{ s}$ medan önskemålet gav $t_{önskemål} = 0,5 \text{ s}$. Denna data gav ett varvtalskrav på

$$n_{krav} = \frac{\varphi \cdot (60 \text{ s/min})}{t_{krav} \cdot 360^\circ} = \frac{90 \cdot 60}{1 \cdot 360} = 15 \text{ rpm} \quad (2.9)$$

och ett önskat varvtal på

$$n_{önskemål} = \frac{\varphi \cdot (60 \text{ s/min})}{t_{önskemål} \cdot 360^\circ} = \frac{90 \cdot 60}{0,5 \cdot 360} = 30 \text{ rpm} \quad (2.10)$$

Utöver önskemålet på IP67-klassning och hastighetskraven behövde dessutom motorn ha ett maximalt vridmoment på minst $M_{max} \approx 23 \text{ Nm}$ och ett nominellt vridmoment på minst $M_{medel} \approx 19 \text{ Nm}$ enligt avsnitt 2.3. Därtill krävdes det att den kunde drivas av 12 V för att batteriet som finns på RescueRunnern skulle kunna användas.

En sammanfattning av kraven och önskemålen på motorn finns i tabell 2.4.

Tabell 2.4: Kravspecifikation för motor till styrning

Kriterier	Målvärde	Krav/önskemål
Kapslingsklass	Minst IP67	Önskemål
Varvtal	≥ 15 rpm	Krav
Varvtal	≥ 30 rpm	Önskemål
Maximalt moment	≥ 23 N m	Krav
Nominellt moment	≥ 19 N m	Krav
Nominell spänning	12 V	Krav

Med hjälp av dessa krav valdes elmotorn med artikelnummer 259.3710.20.00 från Doga. Dess specifikationer och verifiering mot kravspecifikationen finns i tabell 2.5 [9, s. 14-15]. Där syns att motorn som valts klarar alla kraven, dock inte önskemålen.

Tabell 2.5: Specifikation för elmotor Doga 259.3710.20.00 och verifiering mot krav

Specifikation	Målvärde	Verkligt värde	Krav/önskemål	Klarar krav
Kapslingsklass	Minst IP67	IP53	Önskemål	Nej
Varvtal	≥ 15 rpm	22 rpm	Krav	Ja
Varvtal	≥ 30 rpm	22 rpm	Önskemål	Nej
Maximalt moment	≥ 23 N m	130 N m	Krav	Ja
Nominellt moment	≥ 19 N m	20 N m	Krav	Ja
Nominell spänning	12 V DC	12 V DC	Krav	Ja
Maximal ström	-	60 A	-	-
Nominell ström	-	12 A	-	-

2.4.2 Val av kraftelektronik

För att driva elmotorn i olika hastigheter krävs kraftelektronik i form av ett motorstyrkort. Kortet behöver klara spänningen och strömmen som motorn kräver och gå att styra från en mikrokontroller. Ett önskemål är även att styrkortet har inbyggd strömbegränsning för att kunna begränsa momentet som motorn kan generera. Med dessa krav i åtanke valdes motorstyrkortet EM-243C från tillverkaren Electromen. Dess specifikationer och verifiering mot kraven finns i tabell 2.6, och där kan ses att det uppnår både kraven och önskemålet [10].

Tabell 2.6: Specifikation för motorstyrcort Electromen EM-243C och verifiering mot krav

Specifikation	Målvärde	Verkligt värde	Krav/ önskemål	Klarar krav
Drivspänning	12 V DC	12 V-48 V DC	Krav	Ja
Nominell ström	≥ 12 A	35 A	Krav	Ja
Maximal ström	≥ 60 A	60 A	Krav	Ja
Styrsignal	Styrbar från mikro-kontroller	0 V-5 V	Krav	Ja
Inbyggd strömbegränsning	Ja	Ja, 0 A-100 A	Önskemål	Ja

2.4.3 Val av koppling

Kopplingen som ska koppla motorn, och därmed styrsystemet, till och från styraxeln, var från början tänkt att vara en industriellt standardiserad koppling med IP67 eller högre. Sådana kopplingar visade sig vara mycket svåra att få tag på. Därför valdes istället en annan komponent med liknande egenskaper, nämligen en skivbroms till en cykel.

Skivbromsen monteras på elmotorn som centreras direkt under styrets axel. På styrets axel monteras skivbromsens bromsok. När elmotorn ska rotera styret dras bromsen åt och styret roterar tillsammans med skivan. När styret ska rotera fritt, det vill säga när RescueRunnern ska köras manuellt, kopplas skivbromsen från och styraxeln roterar då friktionsfritt gentemot skivan och därmed också elmotorn. Fördelen med detta är att det går att koppla till och från i alla styrvinklar. Eftersom skivbromsen är designad för användning på en cykel, är den anpassad för tuffa förhållanden.

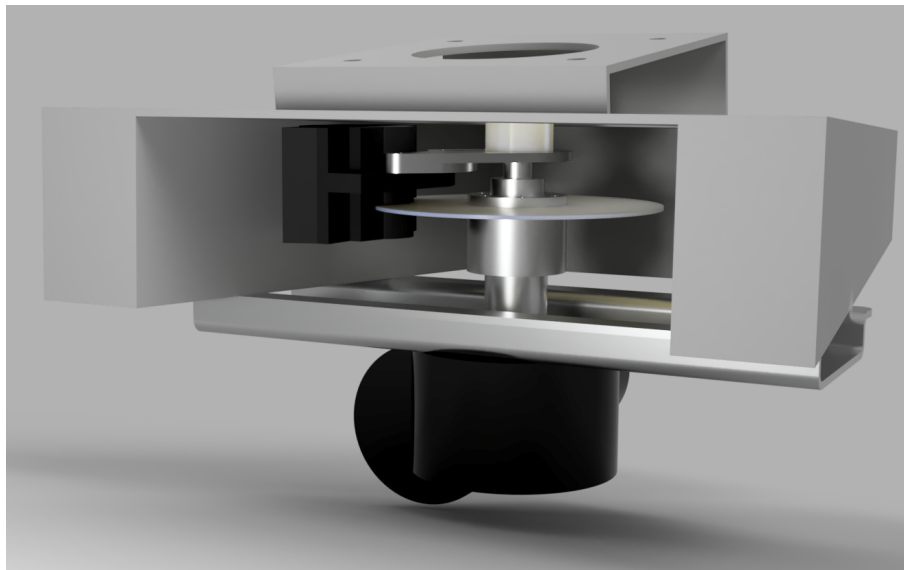
2.5 Realisering

I detta avsnitt beskrivs hur komponenterna för regleringen av styrvinkeln har monterats och ställts in.

2.5.1 Montering av motor och cykelbroms

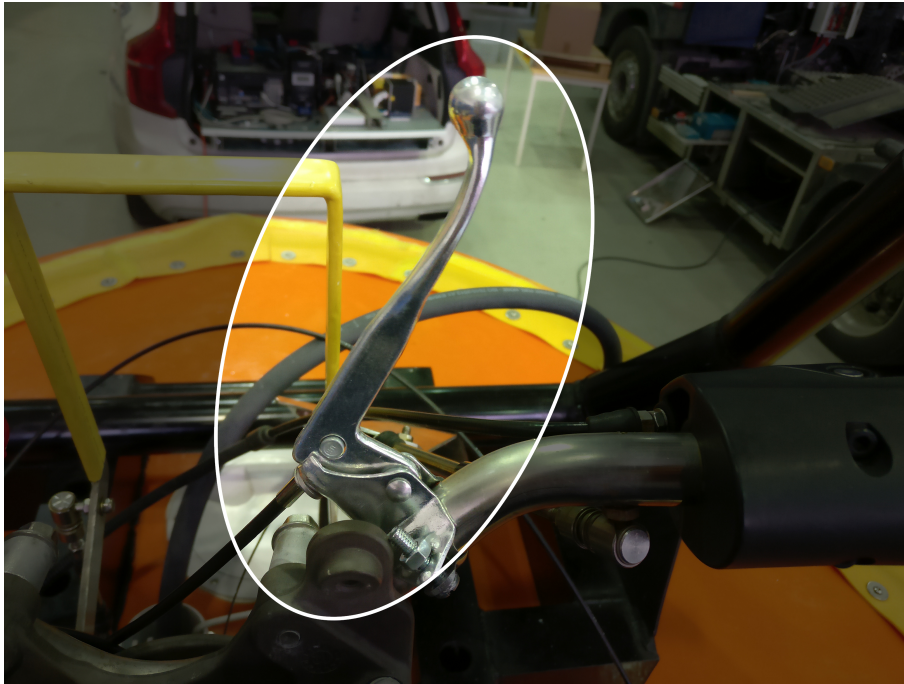
För att montera styrmotorn under styret och skivbromsen som koppling mellan motorns drivaxel och styraxeln, designades och tillverkades flera delar. Först mättes området runt styret på RescueRunnern upp och en CAD-modell skapades. Detsamma gjordes för skivbromsen och motorn. Med hjälp av dessa modeller designades en första version av komponenterna, som sedan skrevs ut i 3D-skrivare för att testmonteras då det ej var möjligt att mäta tillräckligt exakt på RescueRunnern för att göra en slutgiltig design direkt. Med hjälp av testmonteringen gjordes vissa mindre modifikationer av designen. Med samma anledning designades dessutom samtliga komponenter så att det skulle gå att justera in dem vid montering utan att göra om dem, exempelvis genom att göra avlånga hål där det var möjligt. För att skivan och motorn skulle centreras under styret användes ett kullager som monterades ovanför skivbromsen och passade i styraxeln.

Komponenterna tillverkades sedan i aluminium och stål och monterades på RescueRunnern. En rendering av motorn, skivbromsen och de tillverkade komponenterna monterade under styret på RescueRunner kan ses i figur 2.3, och ritningar på komponenterna finns i bilaga A.



Figur 2.3: Rendering av styrningskomponenter monterade på RescueRunner

För att aktivera skivbromsen, och på så vis koppla in styrmotorn på styret, användes ett bromshandtag anpassat för cykelbromsar. Handtaget monterades på styret, och ett spännband användes för att låsa handtaget i draget läge. En bild på hur detta är monterat finns i figur 2.4.



Figur 2.4: Bromshantaget monterat på RescueRunner

2.5.2 Inställning av motorns styrkort

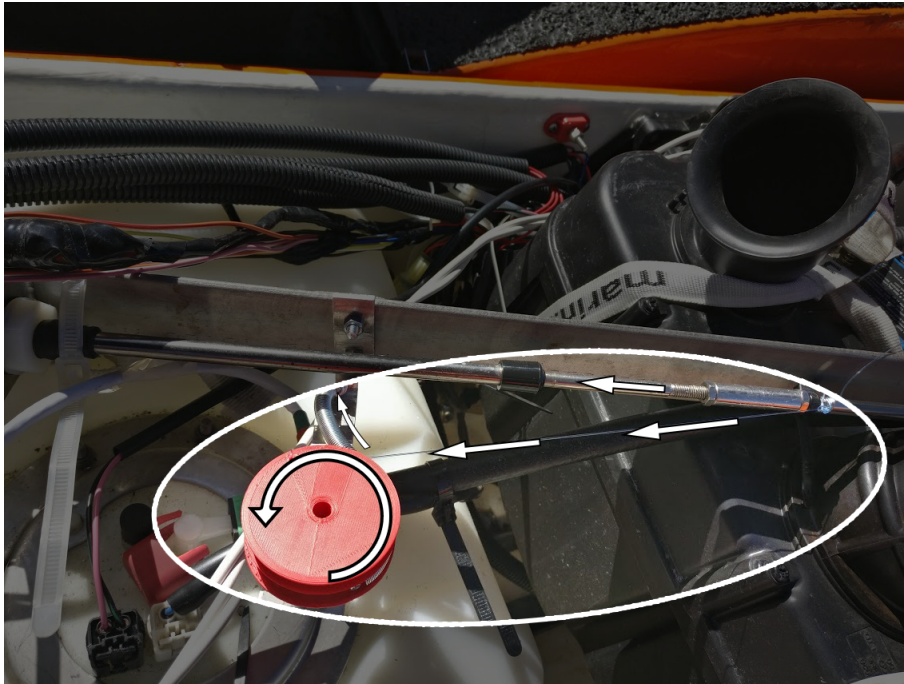
Motorstyrkortet som användes har 21 parametrar som kan programmeras med USB-adaptern EM-328 [10]. De parametrar som ändrades för att styrkortet skulle kunna användas i detta projekt finns i tabell 2.7. Resterande parametrar lämnades på deras standardvärden. För att komma fram till vad strömbegränsningen (parameter 6 och 7) skulle ställas in på, testades styrningen med olika strömbegränsningar. 10 A valdes slutligen som strömbegränsning då den aldrig löste ut under normal drift vid testerna. Dock löste strömbegränsningen ut vid denna inställning om motorn försökte vrida styret utanför ändlägena. På så vis förhindras att styrmekanismen bryts sönder. Start- och stoppramparna (parameter 15 och 16, tiden som det minst måste ta för motorn att gå från 0-100% respektive 100-0% hastighet) ställdes lågt för att det inte skulle kunna uppstå problem när motorn skulle positionsregleras. Anledningen att parametrarna inte stängdes av helt var att det ansågs säkrare att motorn inte kunde starta eller stoppa direkt.

Tabell 2.7: Parametrar som ändrades i motorstyrkortet

Parameter nr.	Namn i datablad	Beskrivning	Inställt värde	Original-värde
5	running speed-2	Hastighet 2 (0=analog 0 V-5 V ingång)	0	50%
6	current limit FW	Ström-begränsning framåt	10 A	10 A
7	current limit REV	Ström-begränsning bakåt	10 A	10 A
15	start ramp	Startramp	100 ms	1000 ms
16	stop ramp	Stoppramp	100 ms	1000 ms

2.5.3 Återkoppling av styrvinkel

Lösningen från projektarbetet 2018 kunde i stort sett replikeras [6]. En cylinder med hål för en potentiometer tillverkades och monterades sedan vinkelrätt längsmed den exponerade delen av styrvajern. När styrvinkeln ändras, drar vajern i fiskelinan, som rullas ut från rullen och roterar på potentiometern. När styrvinkeln förändras i motsatt riktning, drar fjädern i cylindern åt andra hållet. I det senare fallet roterar potentiometern då i motsatt riktning och fiskelinan lindas upp. Se figur 2.5 för att se styrvinkelgivarn monterad med en illustration av funktionen.



Figur 2.5: Färdigmonterad styrvinkelgivare med illustration av funktionen

3

Hastighetsreglering

Gasen i RescueRunnern regleras genom att ett handtag på styret drags åt och släpps. Handtaget drar i sin tur en vajer som är kopplad på en modul, vilken refereras till som Accelerator Position Sensor (APS) i manualen [11, Kap. 7, s. 26]. I APS:en sitter två potentiometrar, APS1 och APS2, som varierar när vajern drags åt [11, Kap. 7, s. 28]. APS:en är kopplad med en kabel till RescueRunners motorstyrenhet, som kallas Electronic Control Module (ECM) [11, Kap. 7, s. 26]. ECM:en mäter kontinuerligt motståndet och kontrollerar gasventilen beroende på APS:ens värde och miljöförhållanden i motorn [11, Kap. 7, s. 31]. Det ursprungliga gassystemet illustreras i figur 3.1 som ett flödeschema.



Figur 3.1: Flödesschema för ursprungligt accelerationssystem

3.1 Kravspecifikation

Utifrån problembeskrivningen och de tidigare kandidatrapporterna [2], [3] togs kravspecifikationen i tabell 3.1 fram.

Tabell 3.1: Kravspecifikation för styrning av gas

Kriterier	Kontrollmetod	Målvärde	Krav/önskemål
Styra gasen i hela dess register	Mätning mot manuell styrning	95% av manuell	Krav
Tid mellan ändlägen	Tidtagning	≤ 1 s	Krav
Tid mellan ändlägen	Tidtagning	$\leq 0,5$ s	Önskemål
Upplösning	Mätning	≥ 10 steg	Krav
Upplösning	Mätning	≥ 50 steg	Önskemål
Noggrannhet	Mätning	± 5 procentenheter	Krav
Noggrannhet	Mätning	± 3 procentenhet	Önskemål

3.2 Konceptval

För att ta fram ett koncept gjordes flera olika lösningar som bedömdes vara realiserbara. För att välja vilket koncept som skulle fungera bäst för detta projekt, användes en Pughmatris.

3.2.1 Koncept

Det finns två huvudsätt att påverka bränsleinsläppet till motorn. Antingen påverkas den vid att elektroniskt efterlikna RescueRunners ursprungliga system eller vid att mekanisk påverka gasspjället. Utifrån kravspecifikationen, tidigare arbeten och diskussion gjordes tre alternativ till lösningar på problemet.

1. Efterliknad av APS med motstånd

Ursprungligen mäter ECM:en kontinuerligt av APS:en, vilket är två potentiometrar. En lösning skulle vara att ECM:en istället mäter en digital potentiometer, som styrs av en Arduino. Det ursprungliga systemet bibehålls i stor del, vilket är önskvärt. Lösningen är redan testad och är dessutom billig och lätt att implementera. En relämodul används för att växla mellan manuellt och fjärrstyrningsläge.

2. Efterliknad av APS med spänning

Det andra konceptet är en variant på det tidigare konceptet. Skillnaden är att spänningen, istället för motståndet, som ECM:en läser av, varieras. Mätningar som genomfördes visar att spänningen över potentiometern i APS:en är ~ 5 V, och i manualen beskrivs vilka spänningar som motsvarar inget respektive fullt gaspådrag, något som skulle kunde efterliknas med en Arduino eller

annan spänningskälla som kan styras elektronisk. Dock är ECM:n i stor grad odokumenterad, vilket gör att det är svårt att säga hur den är jordad och hur en spänningskälla skulle kunna kopplas för gemensam jord. Växling mellan manuell och autonom styrning sker på samma sätt som i det första konceptet.

3. Mekanisk reglering av gasspjäll

En tredje lösning är att mekaniskt öppna och stänga gasventilerna. För denna lösning skulle det behövas ett noggrant och snabbt sätt att få en linjär rörelse, förslagsvis med en servomotor, som kopplas via en vajer direkt till gasventilerna. Växling mellan manuellt och fjärrstyrt läge sker genom på- och avkoppling av servomotorn.

3.2.2 Val av koncept

Med hjälp av konceptförslagen i avsnitt 3.2.1 togs Pughmatrisen i tabell 3.2 fram.

Tabell 3.2: Pughmatris hastighetsreglering

Kriterier	Viktning	Koncept		
		1*	2	3
Påverkan i manuellt läge	5	0	0	-1
Antal komponenter	3	0	+1	-1
Monterbarhet	4	0	-1	-1
Upplösning	3	0	+1	+1
Noggrannhet	4	0	0	-1
Tid mellan ändlägen	4	0	0	-1
Pris	1	0	+1	-1
Hållbarhet	4	0	0	-1
Testat tidigare	4	0	-1	0
Summa		0	-1	-22

*: Referenskoncept

I ovanstående Pughmatris i tabell 3.2 visas att det första konceptet, med efterliknad av APS med motstånd, har flest och störst fördelar. Detta koncept var också det som valdes för vidareutveckling.

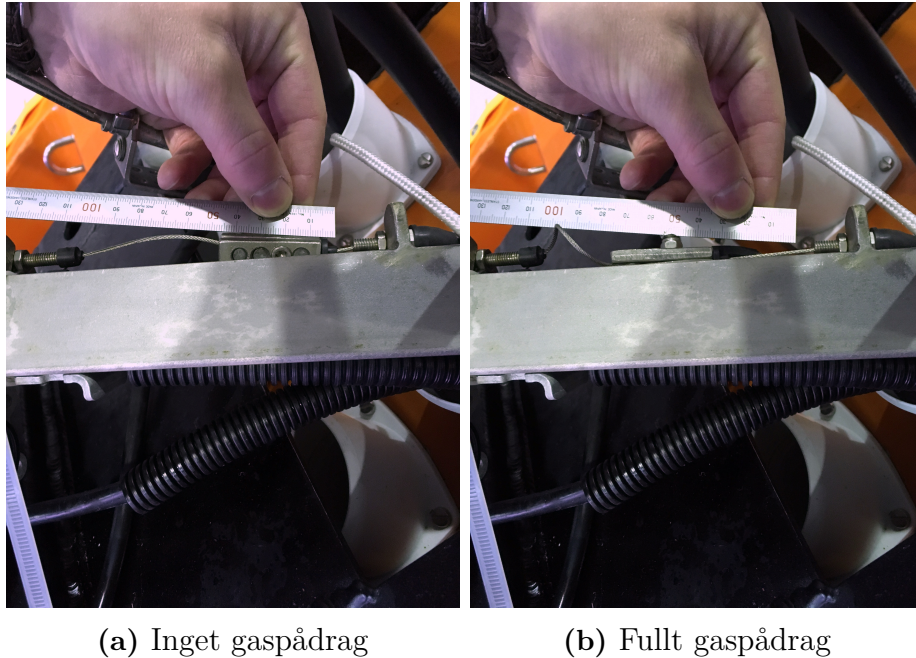
Det andra konceptet, som efterliknar APS:en med spänning, är ett simpelt och snyggt sätt att lösa problemet på. Trots detta är det den enda lösningen som inte är testad innan. På grund av detta finns en stor osäkerhet eftersom den kräver gemensam jord mellan spänningskällan och ECM:ens mätningsskrets. Ingen dokumentation av ECM:en kunde hittas, vilket gör det svårt att bedöma genomförbarheten innan

testning genomförs. Efter diskussion med Fredrik Falkman uppmärksammades det att RescueRunnern endast kan köras på land under kort tid och på låga varvtal. Därför måste gasen, till skillnad från andra delar av projektet, testas i vattnet. Detta gör testproceduren tidskrävande.

Det tredje och sista konceptet, med mekanisk påverkan av gasspjället, har många nackdelar och få fördelar, vilket syns i tabell 3.2, och valdes därför bort.

3.3 Uppmätning av APS

För att avgöra hur det ursprungliga systemet skulle efterliknas, mättes resistansen på APS:en. Mätningarna gjordes genom att använda en linjal för mätning av gasvajerens position, som illustreras i figur 3.2, och en multimeter för att logga resistansen.

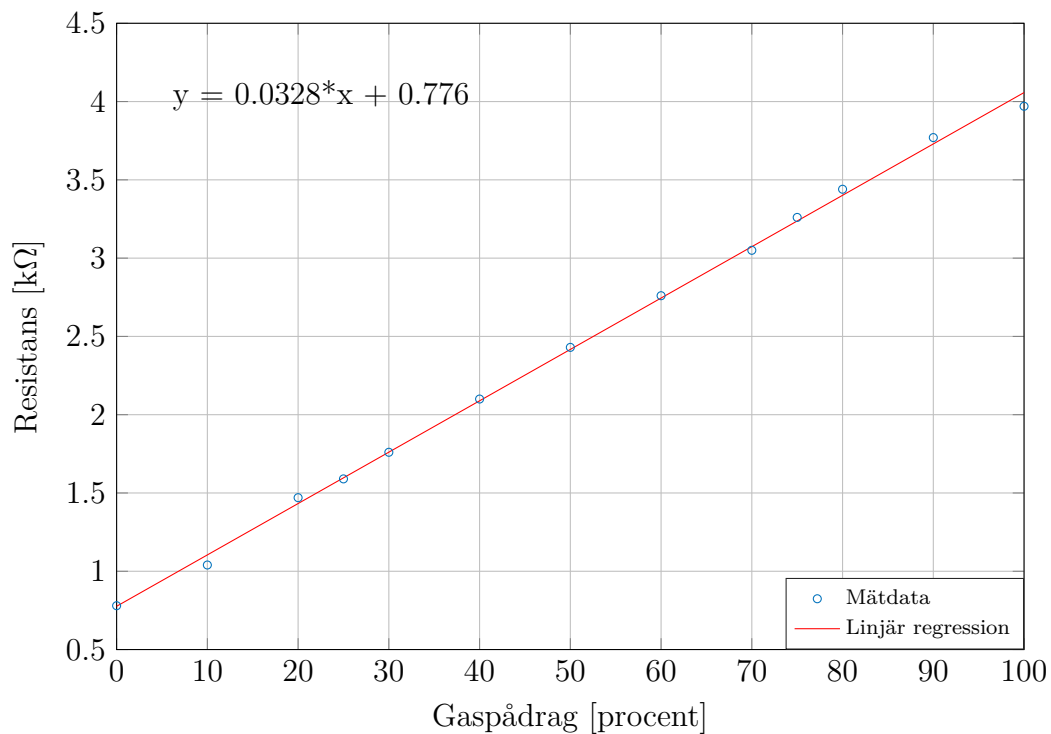


Figur 3.2: Mätning av gasvajerposition

Utifrån 13 mätpunkter kunde det antas att APS:en är linjär. MATLABs `polyval`- och `polyfit`-funktioner användes därför för att approximera funktionen med linjär regression:

$$R_{APS} = 0,0328 \cdot p_{gas} + 0,776 \text{ k}\Omega \quad (3.1)$$

där R_{APS} är resistans och p_{gas} är procent.



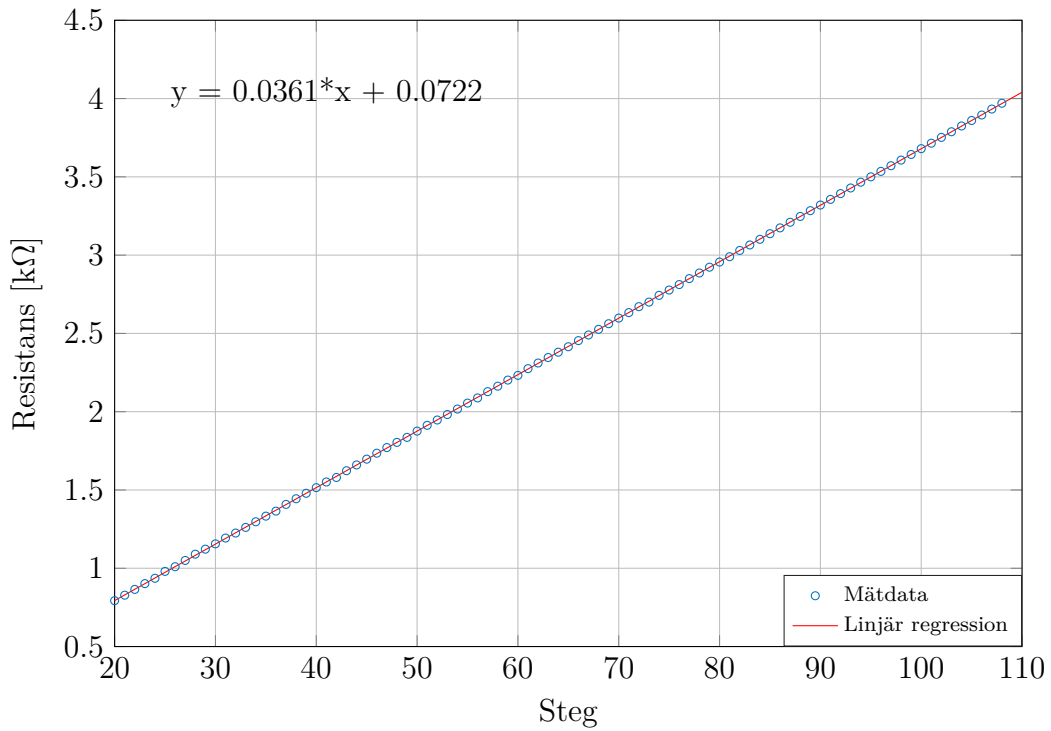
Figur 3.3: Mätning av potentiometer

3.4 Uppmätning av digital potentiometer

En MCP41010-potentiometer används för att efterlikna APS:en, då denna har en resistansräckvidd som går ihop med värdena som ska imiteras. Dess upplösning är på 256 diskreta steg, där 0 är nära 0Ω och 255 är nära $10\text{ k}\Omega$ [12]. Resistansen mellan olika potentiometrar kan skilja så mycket som 20% [12, s. 2], och det måste därför göras mätningar för att kontrollera vilka värden just denna potentiometer har. Mätningarna gjordes genom att använda en multimeter för att logga resistansen. En Arduino används för att kontrollera potentiometern och stega den från 0 till 255. Värden som ligger över eller under det intressanta resistansintervallet $[0,78\text{ k}\Omega; 4,06\text{ k}\Omega]$ förkastas, då modellens noggrannhet borde maximeras innanför just detta området. De resulterande mätpunkterna gav ett fint linjärt samband, och MATLABs `polyval`- och `polyfit`-funktioner användes igen för att approximera funktionen med linjär regression:

$$R_{pot} = 0,0361 \cdot S_{pot} + 0,0722\text{ k}\Omega \quad (3.2)$$

där R_{pot} är resistans och S_{pot} är ett diskret steg.



Figur 3.4: Mätning av gaspådrag

3.5 Matematisk modell för gaspådrag

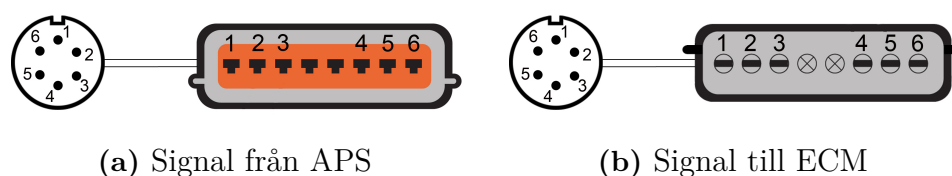
För att kontrollera gasen på ett lätt och intuitivt sätt i koden, var det önskvärt att göra en funktion som tar in procent gas och räknar ut motsvarande steg på den digitala potentiometern. De linjära regressionerna från ekvation (3.1) och (3.2) används därför för att ta fram en funktion som sedan kan skrivas in i koden.

$$\begin{aligned}
 R_{APS} &= 0,0328 \cdot p_{gas} + 0,776 \text{ k}\Omega \\
 R_{pot} &= 0,0361 \cdot S_{pot} + 0,0722 \text{ k}\Omega \\
 R_{APS} &= R_{pot} \\
 &\iff \\
 0,0328 \cdot p_{gas} + 0,776 &= 0,0361 \cdot S_{pot} + 0,0722 \\
 &\vdots \\
 S_{pot} &= 0,91 \cdot p_{gas} + 19,5
 \end{aligned}
 \tag{3.3}$$

3.6 Realisering

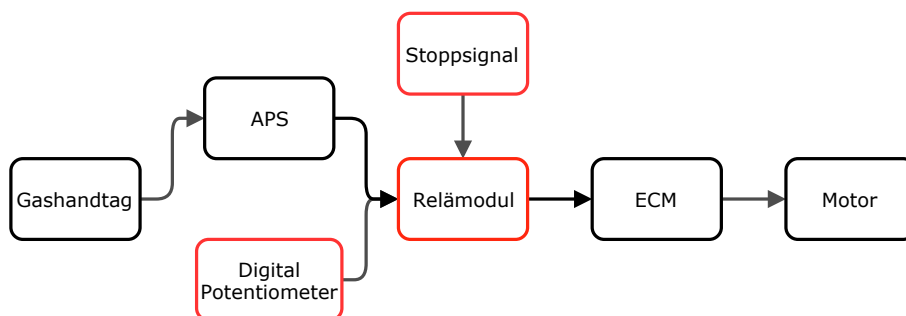
Grunderna för konceptet finns redan beskrivna i kandidatrapporten från 2015 [2], men för att uppfylla målet om att göra en robust lösning som inte kan störa manuell styrning krävdes ett vattentätt, tåligt och enkelt system för att koppla den digitala potentiometern mellan APS:en och ECM:en.

Valet föll på att tillverka egna kablar, vilka kan kopplas direkt på APS:en och ECM:en och använder samma kontaktdon som det ursprungliga systemet så att klippning av kablar undviks. Detta illustreras i figur 3.5. Kontaktdonen som används i RescueRunnern är mycket svåra att få tag på. Det är möjligt att beställa hon-kontakten från en tredjepart, men han-kontakten kunde däremot inte hittas någonstans. En modell fick därför designas i CAD och skrivas ut med en 3D-skrivare.



Figur 3.5: Kablar och kontaktdon som används i konceptet

Växlingen mellan signalen från den digitala potentiometern och signalen från APS:en sker genom ett relä. På så sätt kan det säkerställas att det vid stoppsignal eller strömbrott kopplas tillbaka till signalen från APS:en och manuellt läge erhålls. Det färdiga konceptet illustreras i figur 3.6.



Figur 3.6: Flödesschema för modifierat accelerationssystem. Blocken markerade i rött är utvecklat i samband med projektet

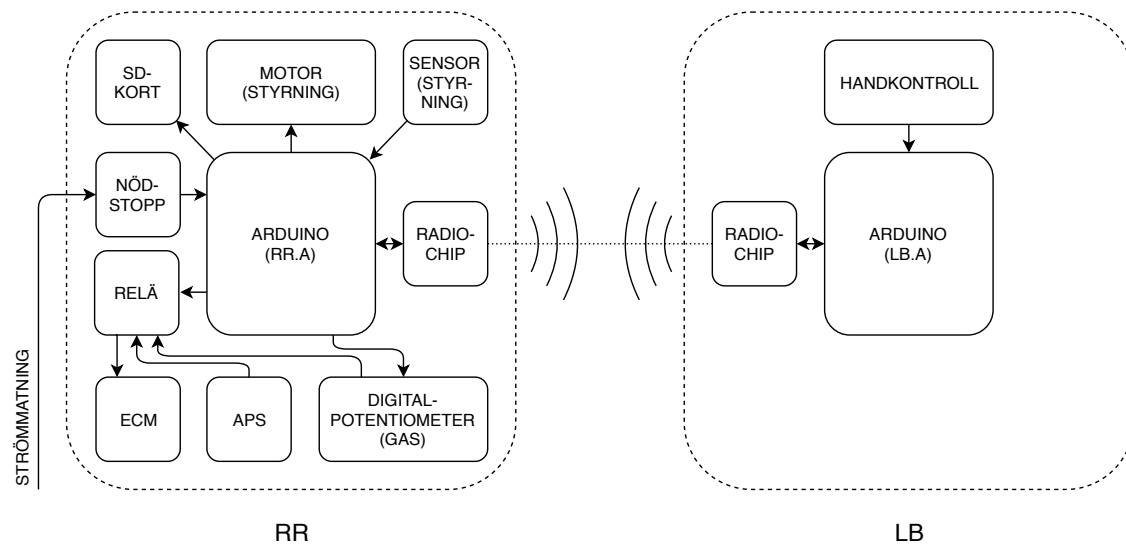
Komponenter och implementering beskrivs mer utförligt i avsnitten 4.5, 4.6 och 4.7.

4

Elektronik

I följande avsnittet kommer de val som gjorts gällande elektroniken att förklaras. Det undersöktes bland annat hur RescueRunnern elektroniskt fjärrstyrs, hur kommunikationen sker, hur strömförsörjningen sker, hur data kan loggas och hur allt kopplas samman.

Det översiktliga blockschemat av komponenter var, baserad på funktioner och redan valda komponenter, tänkt att vara enligt figur 4.1, och det slutgiltiga kopplingsschemat kan ses i figur A.2. RR.A syftar på den Arduino som är tänkt att vara på RescueRunnern, medan LB.A på den som är tänkt att vara på ledarbåten, alltså den plats som RescueRunnern fjärrstyrs ifrån. Varför just Arduino valdes förklaras närmare i avsnitt 4.1.



Figur 4.1: Översikt komponenter

4.1 Mikrokontrollerkort

Arduino är en hårdvara med öppen källkod, det vill säga att ritningarna ligger ute för vem som helst att tillverka [13]. Det har lett till att användandet av och kunskapen om Arduinos har spridits. Det finns olika varianter av Arduino, men gemensamt för dem alla är att de har ett antal analoga och digitala in- och utgångar, även kallade pins [13]. Arduino är enkelt att integrera och komma igång med. Alternativet hade varit att införskaffa en Raspberry Pi, men då hade ett operativsystem behövts väljas och installeras. Arduino kommer å andra sidan med ett enkelt utvecklingsverktyg som finns tillgängligt för Mac, Linux och Windows [13].

Från tidigare projekt fanns dessutom tre Arduino Mega 2560 och en Arduino Due. Duen har en snabbare processor och kör på 32-bitars minne, medan Megan har en långsammare processor och 16-bitars minne [14]. En annan skillnad är att Duen kommunicerar med 3,3 V istället för Megans 5 V [14]. Snabbheten på Megans processor antas räcka gott och väl för det som projektet ämnar att göra. Då motorstyrkortet kommunicerar via 5 V, föll valet på Megan.

4.2 Radiokommunikation

Några olika val fanns med i beslutsprocessen för vilket system som skulle användas:

- Bluetooth
- WiFi
- Radio

Kandidatarbetet 2015 använde sig av WiFi och mastersarbetet 2016 använde radio [2], [5]. WiFi har hög datahastighet och strömförbrukning. Dessutom har WiFi många protokollslager som paketet behöver ta sig igenom vid varje ny sändning [15]. Bluetooth har likvärdig räckvidd som WiFi med 802.11b-teknologin (100 m) [16]. Radio kör inte lika många protokoll som WiFi utan borde således gå snabbare än WiFi, och dessutom ska räckvidden vara likvärdig eller bättre än WiFi [5], [17]. Med ovanstående resonemang i åtanke valdes kommunikation via radio.

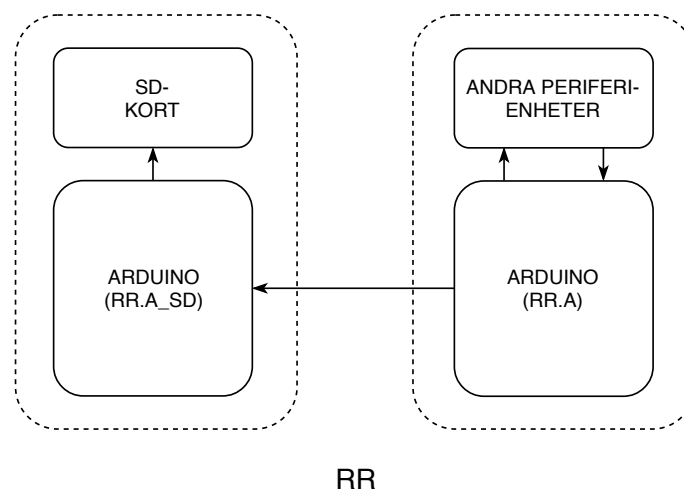
Kortet som används är av modellen nRF24L01 och valdes delvis eftersom det är ett standardkort och är lättillgängligt. Enligt tester ska kortet komma upp i 500 m räckvidd vid fri sikt, vilket räcker gott och väl för projektets ändamål [5, s. 55-56]. Kortet använder SPI-bussen [18, s. 7]. SPI står för Serial Peripheral Interface och är ett gränssnitt för att skicka data mellan en mikrokontroller och dess periferi-enheter.

Eftersom radiokortet ibland stängde av sig, placerades en kondensator på $10\mu\text{F}$ mellan **GND** och **VCC** för att stabilisera spänningsmatningen. Hur kondensatorn och resterande pins är kopplade syns i figur 4.3 samt i kopplingsschemat A.2.

4.3 SD-kort

För att kunna samla in data behövde ett SD-kort kopplas till **RR.A**. Samtliga SD-läsare kommunicerar via **SPI**-bussen [19, s. 209]. De SD-läsare som fanns tillgängliga under projektet var felaktigt designade – så länge som de var inkopplade, släppte de aldrig **MISO**-pinnen och kunde därför inte användas samtidigt som radiokortet, den digitala potentiometern eller andra **SPI**-baserade moduler [20]. Genom att modifiera SD-läsaren kan problemet lösas så att **MISO**-pinnen släpps när den inte används [20]. Modifieringen gjordes, och möjliggjorde samtidig användning av övriga **SPI**-enheter. Dock valdes denna lösning bort då det inte kändes pålitligt att ha en tunn och utsatt löttråd i luften på kretsarna i en kraftigt vibrerande vattenskoter, som modifieringen resulterat i.

Det fanns även sköldar med inbyggd SD-läsare anpassade för Arduino. En sköld är ett lager av komponenter och kopplingar som ansluts över en Arduino. SD-sköldarna var inte felkonstruerade likt SD-kortsadaptarna, men kunde inte användas ovanpå projektets egenkonstruerade sköld. Den tillfälliga lösningen som därför skapades gick istället ut på att sätta in en till Arduino Mega i lådan (hädanefter benämnd som **RR.A_SD**) med SD-skölden på, och koppla samman dessa Arduinos seriellt via **TX1**- och **RX1**-pinnarna. Det enda som **RR.A_SD** gör är att ta in logmeddelanden från **RR.A** och spara dem på SD-kortet, som kan ses i figur 4.2.



Figur 4.2: Översikt SD-lösning

4.4 PS2-kontroller

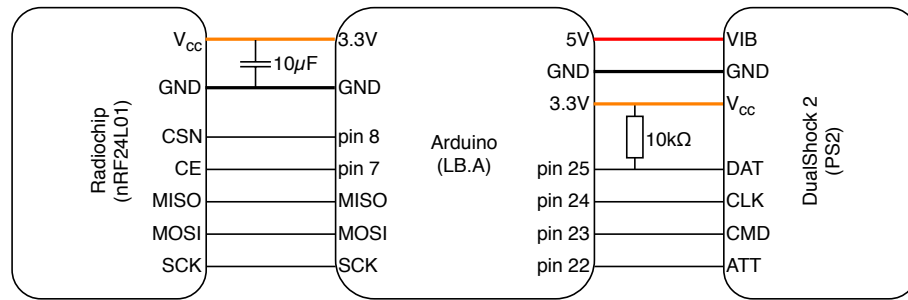
Sedan tidigare fanns det en joystick-sköld till Arduino. Med hjälp av ett färdigt bibliotek kunde joystickens x- och y-värde enkelt avläsas. Dock var det inte möjligt att få max utslag åt alla håll, och värdena ökade inte heller linjärt åt alla håll. Dessutom krävs två joysticks för att kunna svänga max med fullt gaspådrag.

Det stod därför mellan att köpa två nya joystick-moduler och hoppas på att de hade högre kvalitet än den som fanns, eller att finna en handkontroll med två inbyggda joysticks. Det senare alternativet valdes dels då det innebar fler knappar som kan programmeras för att bland annat byta läge mellan manuellt och fjärrstyrt, dels då det endast behövs en sladd till Arduinon istället för två separata kort.

Som handkontroll valdes en DualShock 2, vilken är kontrollen tillhörande spelkonsolen PlayStation 2, och hädanefter benämnd PS2-kontrollen. Ett flertal olika handkontroller övervägdes, men då vissa krävde Bluetooth-kommunikation, och därmed också en Bluetooth-modul, eller behövde egna proprietära kommunikationsprotokoll eller kopplingar för att fungera, föll valet på PS2-kontrollen. Dessutom finns det ett färdigt bibliotek och utförliga instruktioner för hur kontrollen sammankopplas med en Arduino-enhet, vilket förenklade processen [21].

PS2-kontrollen har åtta pins, varav endast sju används här, vilket åskådliggörs i tabell 4.1. Anledningen till att **ACK** inte används är att den endast behövs av PlayStation 2-konsolen för att avgöra om kontrollen är inkopplad [22]. Då detta projekt inte använder en PS2-konsol, behövs inte **ACK**. De fyra pinsen **DAT**, **CMD**, **ATT** och **CLK** kommunicerar via ett **SPI**-liknande protokoll [22]. Dock kräver protokollet inte att Arduino-kortets **SPI**-pins används, utan dessa kan kopplas till valfria digitala pins så länge de specificeras i koden. En pull-up-resistans på $10\text{ k}\Omega$ kopplas enligt figur 4.3 mellan **DAT** och **VCC** [22]. Dessa sju pins kopplas till **LB.A**, som tar emot styrsignalerna och kommunicerar dessa med **RR.A**.

PS2-kontrollen har en inbyggd vibrationsfunktion, vilken enligt specifikation ska drivas med mellan $7,2\text{ V}$ och 9 V , men som efter test visade sig fungera även med den begränsade 5 V -källan på Arduino-kortet [22]. Med vibrationen kan återkoppling av kommunikationen mellan **LB.A** och **RR.A** ske. Detta förklaras vidare i avsnitt 5. Knapparna som finns tillgängliga på PS2-kontrollen kan alla användas av koden, och vilka knappar som används kan ses i tabell 5.1.



Figur 4.3: Kretsschema LB

Tabell 4.1: PS2-kontrollens pinnar

Pin	Förkortning	Förklaring
1	DAT	Dataöverföring från kontrollen till Arduino
2	CMD	Kommandoöverföring från Arduino till kontrollen. Används vid initiering.
3	VIB	Vibration (5 V)
4	GND	Jord
5	VCC	Spänningsmatning (3,3 V)
6	ATT	Uppmärksamhet
7	CLK	Klocka
8	ACK	Bekräftelse – används inte

4.5 Motorstyrkort

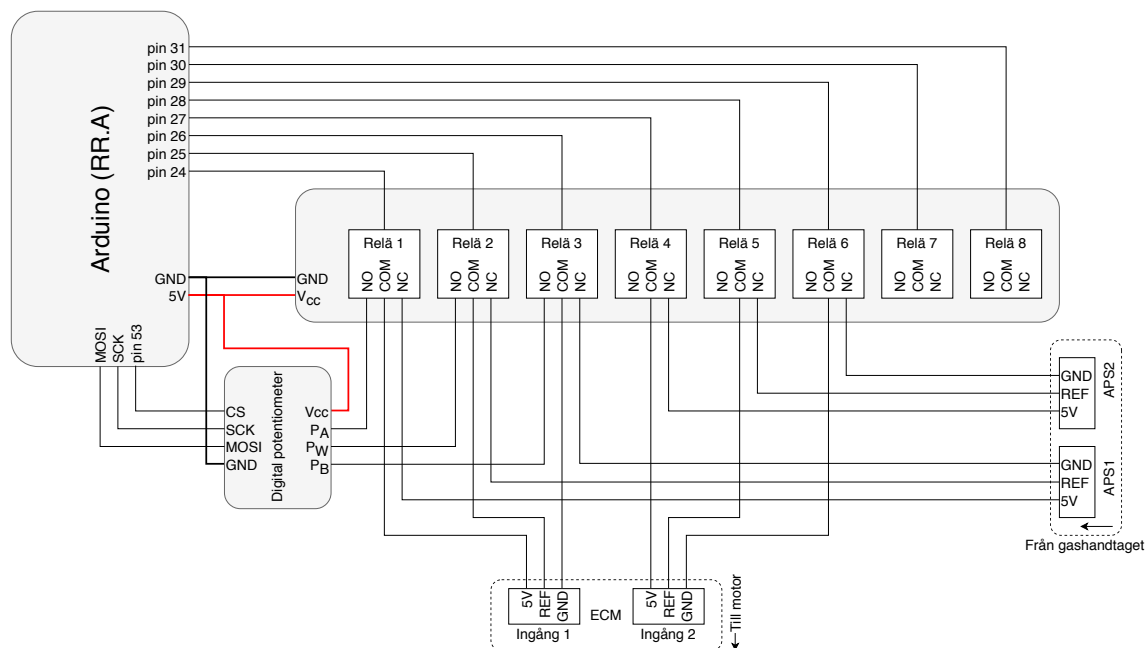
Det valda motorstyrkortet EM-243C, beskrivet i avsnitt 2.4.2, förser styrningens elmotor med ström. Kortet kontrolleras av RR.A via tre pins: CWPIN och CCWPIN som bestämmer riktning, samt SPEEDPIN som bestämmer hastighet genom att matas med ett analogt värde mellan 0 V-5 V [10]. Portarna för spänningsmatningen kopplas till batteriet via en säkring och nödstoppet, och utgångarna går direkt till elmotorn, vilket åskådliggörs i bilaga A, figur A.2.

4.6 Relä

Som nämnts i avsnitt 3.2, bygger relä-lösningen på en tidigare lösning från kandidat-arbetet från 2015 [2]. Lösningen kan enkelt och säkert koppla till och från signalerna från APS:en och den digitala potentiometern till ECM:en. Ett relä av det slag som finns på den använda modulen har två ingångar och en utgång, nämligen NC, NO respektive COM. NC står för Normally Closed, eller normalt stängd, och är standard-ingången som är tillkopplad då relät är av. NO, som står för Normally Open, eller normalt öppen, kopplas inte till förrän systemet aktiveras. Motorns ursprungliga signaler från APS:en kopplas till reläns NC-ingångar, vilket innebär att om ingen

ström tillförs, faller systemet tillbaka på dessa kopplingar. Då nödstoppet bryter hela systemets strömtillförsel innebär detta alltså att manuellt motorläge är garanterat vid ett eventuellt nödstopp då fjärrstyrningssystemet kopplas ifrån och motorn får signalerna från båtens inbyggda ECM.

Systemet använder sex av kortets åtta relän vilket illustreras i figur 4.4, och lämnar därmed två relän fria för till exempel eventuell framtida implementation av start- och stoppsignaler för motorn. Båda APS-kontakterna har tre kopplingar vardera, nämligen 5V, REF och GND, som ger ett referensvärde på REF relativt 5V och GND baserat på önskat gaspådrag. Enligt kopplingsscheman från kandidatarbetet 2015, kopplar reläna ihop APS1 till ECM ingång 1 och APS2 till ECM ingång 2 i standardläge, medan systemet i aktivt läge kopplar systemets simulerade APS-signal till ECM ingång 1 och ECM ingång 2 kopplas ifrån [2]. APS1 och APS2 kopplas helt ifrån i detta läge då de inte fyller någon funktion.



Figur 4.4: Reläkoppling

4.7 Digital potentiometer

När fjärrstyrningsläge aktiveras, och signalerna från gashandtagets APS-signal kopplas ifrån med hjälp av reläna, kopplas istället systemets simulerade APS-signal, det vill säga signalerna från den digitala potentiometern MCP41010, till. MCP41010 sitter monterad på skölden för RR.A, kontrolleras via SPI, och är kopplad med MOSI, SCK, och CS, vilket åskådliggörs i figur 4.4. Som nämnts i avsnitt 3.4 har denna en resistansräckvidd vilken går ihop med värdena som ska imiteras, och detta kontrolleras i mjukvaran i RR.A enligt formeln (3.3).

4.8 Skapandet av sköld

Då allt på RescueRunnern till havs kommer att utsättas för otaliga stötar och tuffa förhållanden riskerar lösa sladdar och komponenter att lossna. För att minimera risken för detta konstruerades en så kallad sköld till Arduinon, som kopplar samman de mindre komponenterna samt har kontakter för kopplingar till de övriga komponenterna. På skölden monterades hållare för den digitala potentiometern och radiomodulen med tillhörande kondensator. Kontaktier för kopplingar till motorstyrkortet, vinkelsensorn och reläna lades till. Även en hållare för SD-kortsmodulen lades till, men används ej efter att den visade sig vara feldesignad (se 4.3). Skölden kan ses i A.3.

4.9 Strömförsörjning och nödstopp

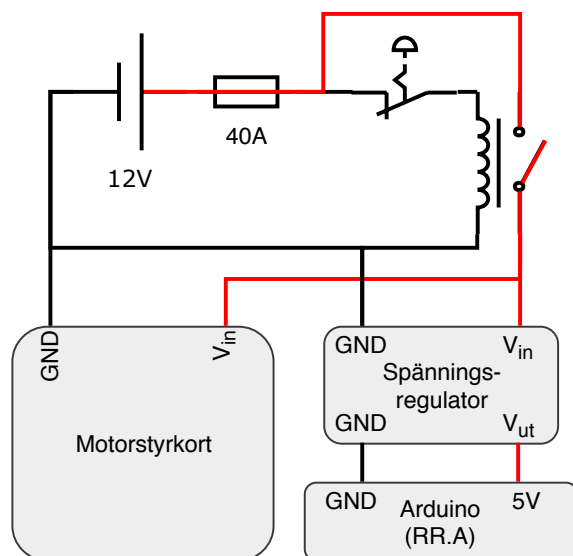
På RescueRunnern drivs all elektronik av det befintliga 12 V-batteriet, och då olika komponenter drivs av olika spänningar behövs spänningsregulatorer. Motorstyrkortet, vilket får sin ström direkt från batteriet, reglerar spänningen som matas till motorn. Motorn drivs på upp till 12 V, och är därmed den komponent som behöver högst spänning, medan de två Arduinos, reläna, den digitala potentiometern, och vinkelsensorn alla drivs på 5 V. Radiochippet drivs på 3,3 V. En säkring samt en nödstoppslösning är installerat mellan batteriet och elektroniken, och vid eventuellt fel slås strömmen helt ifrån. Nödstoppslösningen, som kan ses i figur 4.5, bygger på att en nödstoppsknapp slår till eller ifrån ett relä som i sin tur matar vidare strömmen från batteriet till elektroniken.

Arduinon har inbyggda spänningsregulatorer som kan mata ut stabila spänningar på 3,3 V och 5 V, och kortet kan i praktiken förses med ström på tre sätt, som alla har för- och nackdelar. Om Arduinon kopplas via USB används de inbyggda spänningsregulatorerna och spänningsskydden för att förse utgångarna med stabila 3,3 V- och 5 V-spänningar, och strömförbrukningen begränsas av vad USB-källan kan leverera. Dock är det fördelaktigt att lämna USB-ingången fri för att enkelt kunna koppla till en dator för felsökning och omprogrammering.

Ett annat alternativ är att använda den befintliga strömingången, som kan drivas med upp till 12 V. Spänningen regleras ner till de önskade 3,3 V och 5 V, och skyddar även Arduinon mot eventuell överspänning. Det är alltså möjligt att driva Arduinon med ström direkt från batteriet, men vid test kunde de inbyggda regulatorerna inte mata ut tillräckligt mycket ström för att driva alla komponenter.

Det tredje alternativet är att koppla till en stabil 5 V-strömkälla direkt på någon av Arduinons 5 V-utgångar. Detta förser Arduinon med ström samtidigt som spänningen även regleras ner till 3,3 V. Dock förbikopplar detta de inbyggda spänningsskydden, så om den tillförda spänningen skulle överstiga tillåtna värden riskeras kortet att skadas. Strömmen begränsas i detta fall av strömkällan.

Baserat på ovanstående valdes det tredje alternativet, och för att undvika över-spänning samt att reglera ner spänningen till 5 V kopplades en separat spänningsregulator på. Alla komponenter som drivs på 5 V och 3,3 V tar sin ström från Arduinon, och spänningsregulatorn samt motorkortet kopplas, genom nödstoppet och säkringen, till batteriet.



Figur 4.5: Översikt över strömtillförsel genom säkring och nödstopp

5

Mjukvara

Mjukvaran är uppdelad i tre Arduino-projekt (`RR.A`, `RR.A_SD`, `LB.A`) och ett gemensamt bibliotek. Deras respektive översiktliga exekveringsloopar syns i bilaga A, figur A.1.

5.1 PS2-kontroll

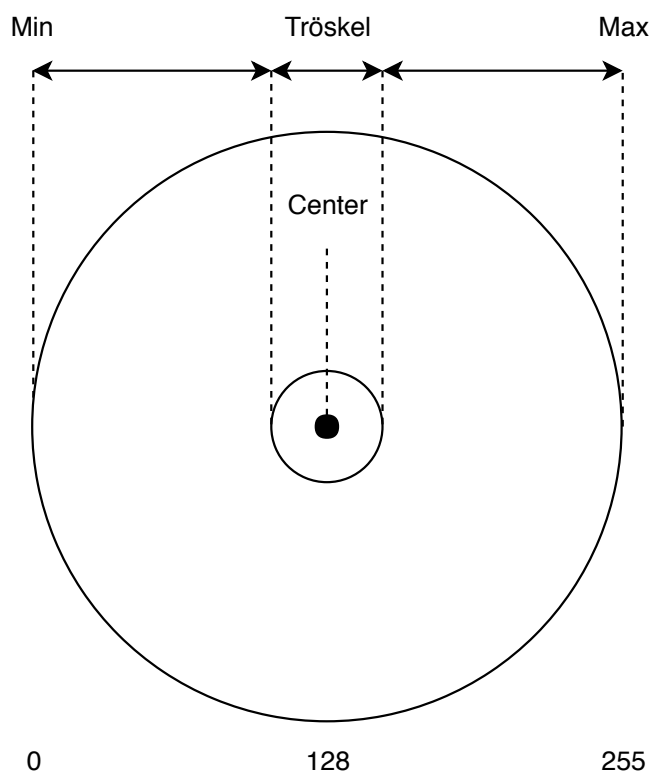
I mjukvaran är PS2-kontrollens knappar bundna enligt tabell 5.1.

Tabell 5.1: PS2-kontrollens använda knappar

Knapp	Förklaring	Nedhållningstid
SELECT	Byt läge	0,4 s
R1	Initiera benchmark (höger→mitten) eller avbryt	1,0 s
L1	Initiera benchmark (vänster→mitten) eller avbryt	1,0 s
R2	Initiera benchmark (höger→vänster) eller avbryt	1,0 s
L2	Initiera benchmark (vänster→höger) eller avbryt	1,0 s
TRIANGLE	Starta loggning	1,0 s
CIRCLE	Stoppa loggning	1,0 s

För att bekräfta ett byte av läge kommer kontrollen att vibrera i pulser – lika många gånger som det nya läget kodas som (byte till manuellt läge resulterar alltså i en vibration medan byte till fjärrstyrningsläge resulterar i två) vid bekräftelse av ett meddelande, hädanefter benämnt **ACK**. Vid övriga **ACK** vibrerar kontrollen i en sekund.

PS2-kontrollens joysticks har ett intervall mellan 0 till 255, där mittenvärdet är 128. För att förhindra att `RR.A` gasar när joysticken inte har gått tillbaka helt och hållet (om den till synes är i mitten men värdet till exempel är 134), har en tröskel implementerats enligt figur 5.1 i skapandet av nya meddelanden. Om absolutvärdet av mittenvärdet (128) subtraherat med det faktiska värdet är mindre än tröskelvärdet, tolkas det som att joysticken är i mittenläget.



Figur 5.1: Joystickens intervall

Styrning sker med den högra joysticken. Så länge som värdet är större än tröskelvärdet, avbildas alla kvarvarande värden om så att övergången blir mjuk (höger: center + tröskelvärde upp till max, motsvarande för vänster). Ett accelerationsmeddelande skapas genom att den vänstra joystickens övre y-koordinater avbildas till ett värde mellan 0 och 100.

5.2 SD-kort

Det sker seriell kommunikation till `RR.A_SD` från `RR.A`. Det uppstod dock problem om `RR.A` skickade data snabbare än vad `RR.A_SD` kunde läsa, då bufferten aldrig minskade och det till slut ledde till att allt minne blev förbrukat. En timer har därför lagts in för att komma runt problematiken. Om `RR.A`:s nästa uppgift är att skicka ett loggmeddelande, kommer det meddelandet aldrig att skapas om tidskravet inte är uppfyllt. Sändningen av viktiga meddelanden, så som byte av loggnamn, sker dock i en `while`-loop tills de skickats.

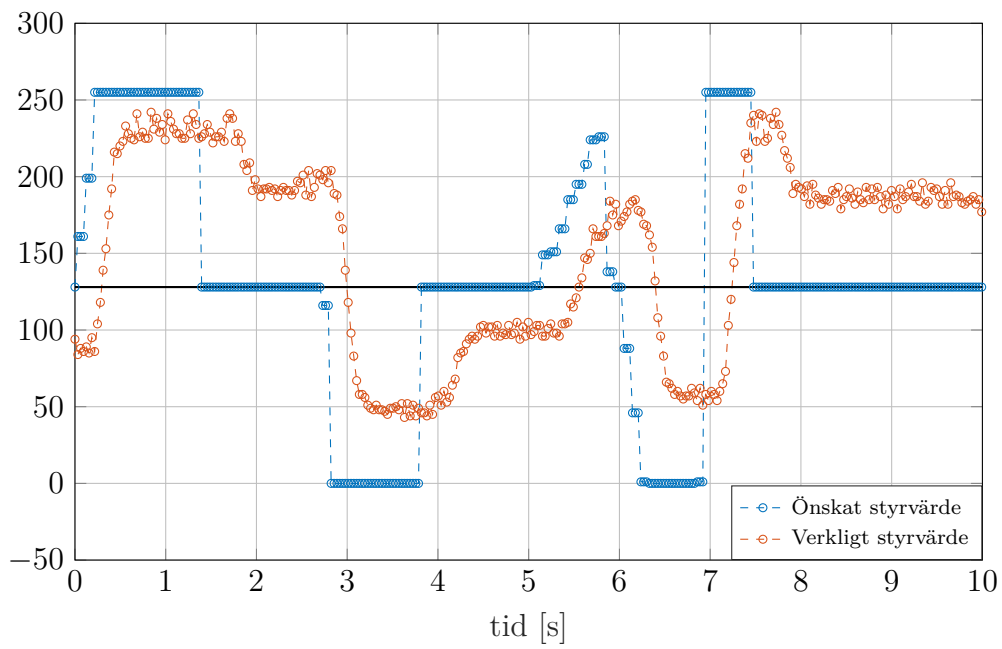
5.3 Styrvinkelgivare

Från styrvinkelgivaren mottogs ett värde från 0 till 1023. Värdet behövde kalibreras och skalas om till ett värde mellan 0 och 255, då det var denna skala som användes för önskat styrvärde. För att göra detta mättes vad givaren visade vid full vänstersväng respektive full högersväng. Värdena noterades och användes för att avbilda max sväng höger till 255 och max sväng vänster till 0. Av säkerhetsskäl valdes att lämna 10 värden utanför intervallet för att inte riskera att systemet skulle försöka köra mot ändlägena. Vid max sväng vänster uppmättes värdet 670, och vid max sväng höger uppmättes 0. Därför avbildades 660 till 0, och 10 till 255. Med avbildningen antas att det går att svänga lika mycket höger och vänster för att centrum ska ligga på 128.

5.4 PID-regulator

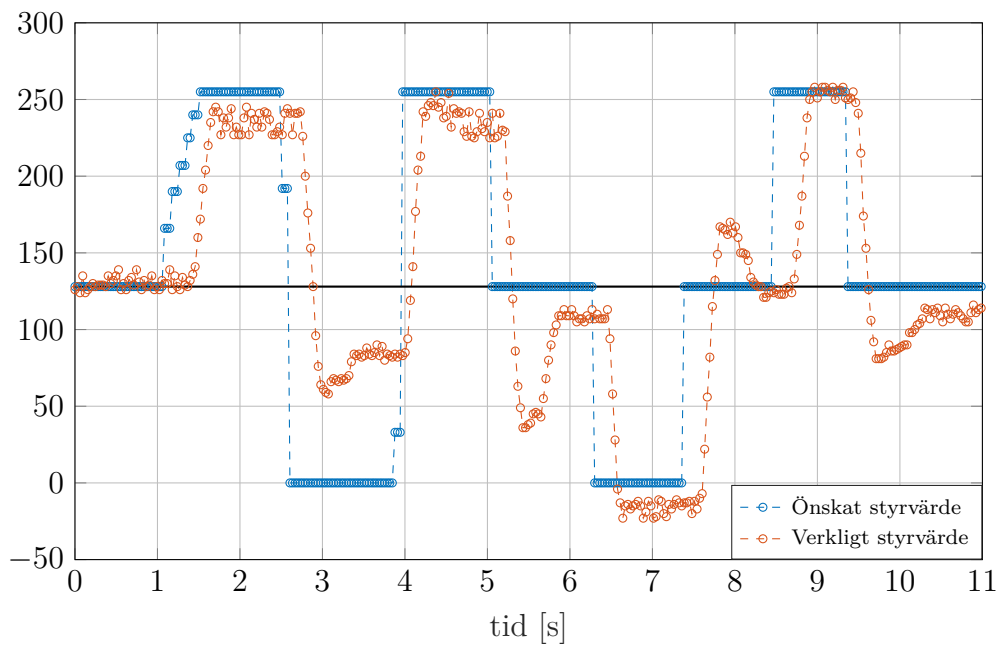
För att kunna styra dysan så snabbt som möjligt utan att svänga för mycket krävs en regulator. En PID-regulator valdes för att den är snabb och enkel att implementera, samt att det är en standardlösning som fungerar tillfredsställande i de flesta fall. Parametrarna har dock endast testats fram då det inte finns någon matematisk modell på RescueRunnern. Tillvägagångssättet var att öka K_p tills systemet var snabbt men utan översläng, och därefter öka K_i med liknande krav. K_d har även testats men utan framgång. Värt att notera är att det är lättare att svänga från höger till vänster än tvärt om. Även värt att notera är att sensorn inte är exakt och att värdena därför hoppar något i följande grafer.

För det första testet valdes $K_p = 1$, $K_i = 0,1$ och $K_d = 0$. Resultatet av detta test kan ses i figur 5.2, där regulatorn har för låg förstärkning då det verkliga styrvärdet aldrig når fram till det önskade.



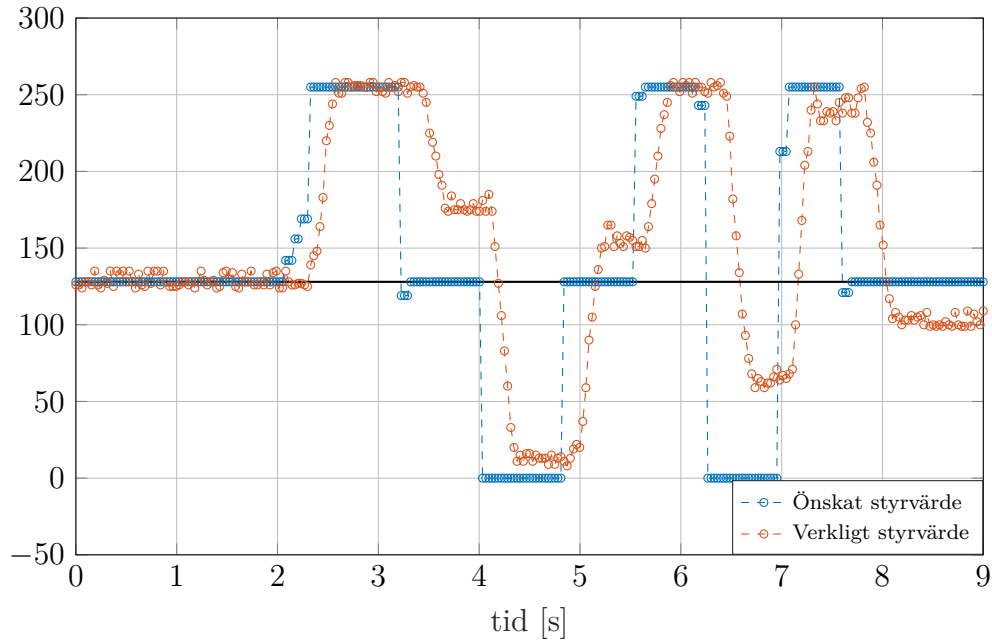
Figur 5.2: Loggad data av reglering för styret med ($K_p = 1$, $K_i = 0.1$, $K_d = 0$)

K_p höjdes därför sedan till 2, med de andra parametrarna oförändrade. Som kan ses i figur 5.3, får systemet dock en översläng med dessa värden när det önskade styrvärdet sätts till 128 (rakt fram); K_p är därmed för hög. Överslängen syns ej vid max sväng höger då det finns ett mekaniskt stopp som gör att styret ej kan vridas längre.



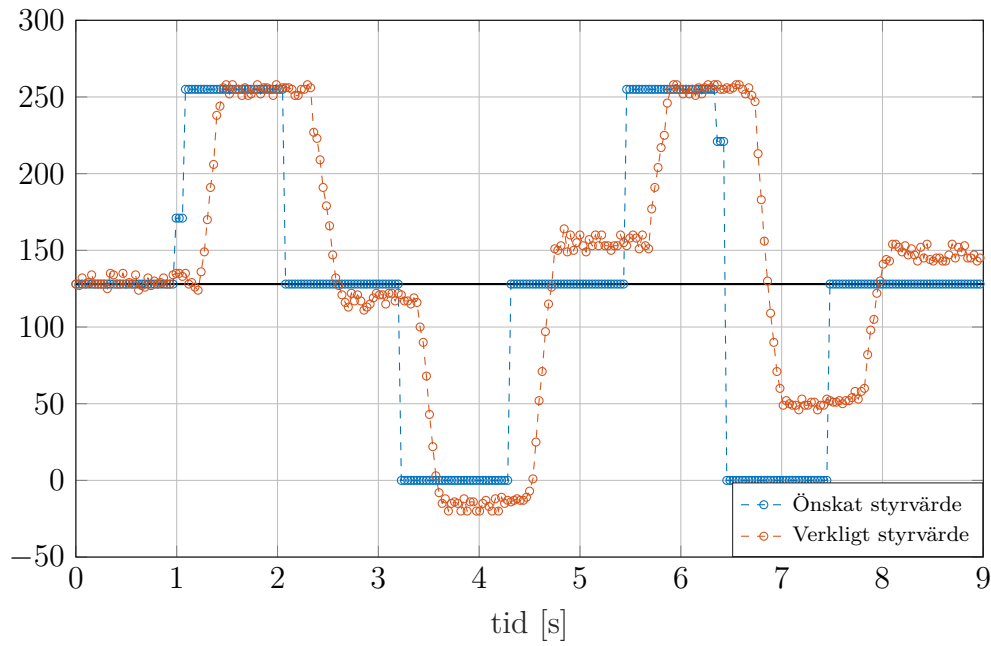
Figur 5.3: Loggad data av reglering för styret med ($K_p = 2$, $K_i = 0.1$, $K_d = 0$)

På grund av överslängarna i föregående test sänktes K_p till 1,5. Då erhöles datan i figur 5.4, där överslängen vid styrning rakt fram fortfarande finns kvar, men inte är lika stor. Detta värde på K_p ansågs därför godtagbart.



Figur 5.4: Loggad data av reglering för styret med ($K_p = 1,5$, $K_i = 0,1$, $K_d = 0$)

K_i ökades sedan i steg tills att överslängen vid styrning mot mitten återigen blev för stor. Resultatet med $K_i = 2$ kan ses i figur 5.5, där markanta överslängar både vid tiden $t = 5$ s och $t = 8$ s syns.



Figur 5.5: Loggad data av reglering för styret med ($K_p = 1,5$, $K_i = 2$, $K_d = 0$)

På grund av dessa överslängar minskades K_i igen till 1,5. Dessa värden är de som valdes att användas, och resultatet kan ses i avsnitt 6.1.

5.5 Radiokommunikation

Kommunikationen mellan RR.A och LB.A sker, som tidigare nämnts, via två radiomoduler. Det finns ett färdigt kodbibliotek, vilket används för att skicka och ta emot meddelanden [23]. Koden som körs på RR.A lyssnar hela tiden efter nya meddelanden från LB.A. Ett meddelande byggs upp av 16 bitar; en byte header och en byte data. Det passar bra eftersom en integer på Arduino Mega är 16 bitar [24]. Vilka meddelanden som finns att skickas kan ses i tabell 5.2.

Tabell 5.2: Radiomeddelanden att skickas mellan RR.A och LB.A

Namn	Header	Data	Förklaring
SetMode	0b00000001	Mode-data (se tabell 5.3)	Byter läge
Steer	0b00000010	Styrningsvärde [0; 255]	Sväng
Accelerate	0b00000011	Gas i procent [0; 100]	Accelerera
SetLogging	0b10000000	true eller false	Skifta logg-läge
NewLog	0b10000001	Nytt loggnummer [0; 255]	Skriv data till ny logg
Benchmark	0b10000010	Benchmark-data (se tabell 5.4)	Benchmark start/stopp
ACK	0b10000011	Meddelandet som mottogs	Bekräfta meddelande
PIDP	0b11000000	10· nytt K_p -värde [0; 255]	Sätt K_p i runtime
PIDI	0b11000001	10· nytt K_i -värde [0; 255]	Sätt K_i i runtime
PIDD	0b11000010	10· nytt K_d -värde [0; 255]	Sätt K_d i runtime

Tabell 5.3: Mode-data – vad de olika lägena kodas som

Namn	Data	Förklaring
ManualMode	0b00000001	Byter läge till manuellt läge
RemoteMode	0b00000010	Byter läge till fjärrstyrningsläge

Tabell 5.4: Benchmark-data – vad de olika benchmark-meddelanden betyder

Namn	Data	Förklaring
BenchmarkLeftToRight	0b00000000	Börja benchmark vänster→höger
BenchmarkRightToLeft	0b00000001	Börja benchmark höger→vänster
BenchmarkLeftToMid	0b00000010	Börja benchmark vänster→mitten
BenchmarkRightToMid	0b00000011	Börja benchmark höger→mitten
BenchmarkCancel	0b00000100	Avbryt benchmark

Meddelandena skickas via ett UDP-liknande protokoll, det vill säga att det inte finns några garantier att de faktiskt går fram. Däremot uppnås lägre fördröjningar då handskakningar inte sker och då tidigare meddelanden inte behöver skickas om. Beslutet att använda ett UDP-liknande protokoll framför ett TCP-liknande togs eftersom meddelanden snabbt blir inaktuella när nya behöver skickas så fort PS2-kontrollens joystick ändrar position och att det behövs snabb överföring. För att säkerställa att meddelanden, som skapas då joysticksen är i mittenlägena (0% gas och styrvinkel rakt framåt), inte försvinner, skickas regelbundna meddelanden om önskad gas och styrning. Alternativet hade varit att implementera återskickning från TCP och sluta skicka när **RR.A** bekräftat mottaget meddelande om 0% gas eller ingen styrvinkel.

När ett meddelande att byta till manuellt läge tas emot, skickas signaler att sluta gasa och svänga, samt signaler till reläerna till gasstyrningen att slå ifrån. Om **RR.A** körs i fjärrstyrningsläge och det blir radiotystnad för mer än fem sekunder slås manuellt läge automatiskt på.

Viktigare mottagna meddelanden som byte av läge, aktivering och avaktivering av logg, ny logg, benchmark och byte av PID-parametrar skickar dock bekräftelse genom ett **ACK**-meddelande från **RR.A** till **LB.A**. Det sker alltså ingen automatisk återskickning av bortfallet meddelande, utan detta får användaren själv göra.

En nackdel som upptäcktes med radiokorten är att de inte tycks klara av att skicka data hur frekvent som helst. Enligt specifikationen för radiokortet, får det inte vara i sändningsläge för mer än 4 ms [18]. Om radiokortets kö för meddelanden fylls snabbare än vad som skickas, kommer det aldrig att lämna sändningsläget. Som en konsekvens stänger Arduinon av sig. För att lösa detta har en timer lagts in som förhindrar att meddelanden skickas oftare än 40 ms. Problem kan uppstå där Arduinon i värsta fall kör i en sådan hastighet att enbart en typ av meddelande sammanfaller

med det bestämda tidskravet. Därför skickas varannat styrmeddelande och varannat accelerationsmeddelande. Övriga meddelanden ligger dock i en `while`-loop som körs tills att meddelandet har skickats, då dessa meddelandena enbart skickas en gång och är således viktigare än de vanliga fjärrstyrningsmeddelanden.

När `RR.A` har slagit om till manuellt läge efter en önskad radiotystnad, sänder `LB.A` fortfarande ut fjärrstyrningsmeddelanden. Vid första mottagna styrnings- eller accelerationsmeddelande kommer `RR.A` automatiskt att slå om till fjärrstyrningsläge igen. Detta sker för att förhindra problem när `LB.A` och `RR.A` är i olika lägen.

5.6 Debug och loggning

För att enklare kunna utvärdera PID-regulatorn och byta loggfil lyssnar `LB.A` på nya seriella meddelanden från en inkopplad dator. Genom att skriva `NEW_LOG` följt av en siffra inom intervallet $[0; 255]$, skickar `LB.A` ett `NewLog`-meddelande till `RR.A`. `RR.A` vidarebefordrar sedan detta till `RR.A_SD` som ändrar vilken fil som loggning ska sparas till. Liknande kommunikationskedja sker vid meddelandena `PID_P`, `PID_I` och `PID_D` följt av en siffra för att ändra PID-regulatorns parametrar medan programmet körs.

6

Resultat

Stora delar av målen som sattes för projektet har uppfyllts. Systemet som reglerar styrvinkeln och växlingen mellan manuell och fjärrstyrt läge fungerar enligt majoriteten av kraven som ställdes på systemet. Hastighetsregleringen uppfyllde däremot inte alla krav, vilket delvis beror på svår genomförd testning av lösningar till hastighetsregleringen, som i sin tur leder till att det är få krav i den generella kravspecifikationen som uppfylls. Exempelvis är “Ej påverka manuell körning märkbart” uppfyllt för styrvinkeln, men ej för hastighetsregleringen.

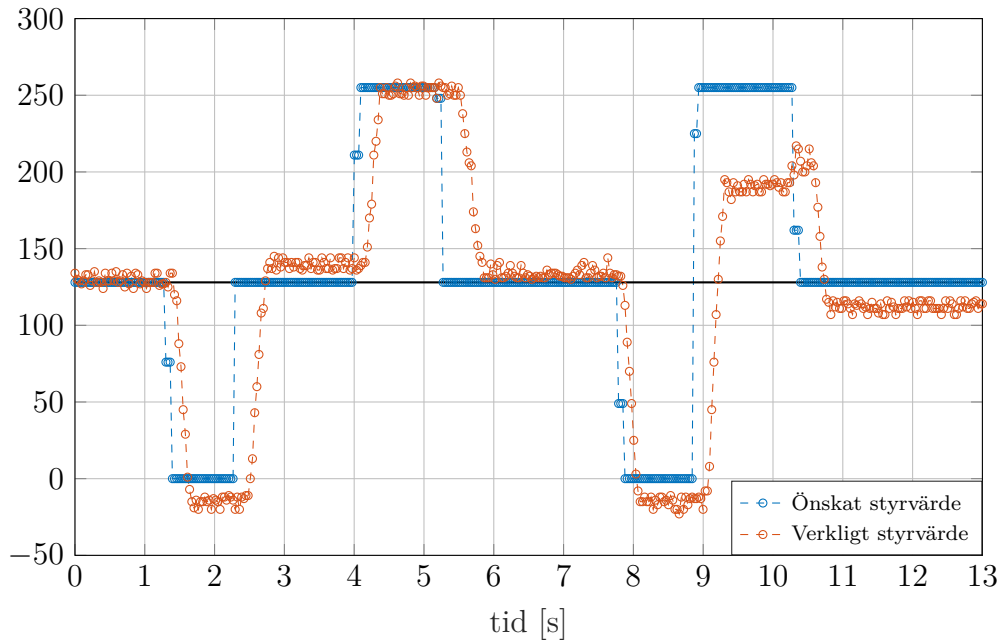
Tabell 6.1: Generell kravspecifikation och resultat

Kriterier	Målvärde	Krav/ önskemål	Uppfylls
Växling mellan manuell körning och fjärrstyrning	Ja	Krav	Ja
Ej påverka manuell körning märkbart	Ja	Krav	Nej
Komponenter med kapslingsklassning	Minst IP67	Önskemål	Nej
Räckvidd på trådlös kommunikation	≥ 50 m	Krav	Nej
Räckvidd på trådlös kommunikation	≥ 200 m	Önskemål	Nej

Vid test av fjärrstyrningen upptäcktes att radiokorten börjar tappa meddelanden efter 12,5 m. Det blev dessutom avsevärt värre täckning när en förare satt på RescueRunnern. Projektets mål gällande radoräckvidd uppfylldes därför ej.

6.1 Reglering av styrvinkel

För att kunna utvärdera kravspecifikationen för styrningen (tabell 2.1) loggades önskat styrvärde och aktuellt styrvärde vid en styrsekvens som innehöll fullt utslag åt både höger och vänster, samt styrning rakt fram, se figur 6.1.



Figur 6.1: Loggad data av reglering för styret med ($K_p = 1,5$, $K_i = 1,5$, $K_d = 0$)

En sammanställning av kraven för styrningen jämfört med kravspecifikationen finns i tabell 6.2.

Tabell 6.2: Kravspecifikation och resultat för styrning

Kriterier	Målvärde	Krav/önskemål	Uppfylls
Styra vinkeln i hela dess register	95% av manuell	Krav	Ja
Tid mellan ändlägen	≤ 1 s	Krav	Ja
Tid mellan ändlägen	$\leq 0,5$ s	Önskemål	Nej
Noggrannhet	± 5 procentenheter	Krav	Nej
Noggrannhet	± 1 procentenhet	Önskemål	Nej

Styra vinkeln i hela dess register

För att utvärdera kravet ”Styra vinkeln i hela dess register”, kan det ses i avsnitt 5.3 att värdet från givaren avbildades från intervallet $[660; 10]$ istället för $[670; 0]$ som givaren gav när styret vreds manuellt. Detta ger att styrsystemet maximalt kan använda $660 - 10 = 650$ steg, medan i manuellt

läge kan $670 - 0 = 670$ steg användas. Dessa steg ger att styrsystemet kan nå $\frac{650}{670} \approx 97\%$ av det manuella styrutslaget. Dessutom kan det ses i figur 6.1 att ändlägena (0 och 255) nåddes. Därmed drogs slutsatsen att kravet på $\geq 95\%$ av manuell styrvinkel klarades.

Tid mellan ändlägen

Tiden för ändläge till ändläge mättes ej upp direkt, däremot kan tiden för sväng från centrum till ändlägena, respektive från ändlägena till centrum tas fram ur figur 6.1. Vid verifiering av detta krav används därför att tiden för att rotera från ändläge till ändläge är mindre än tiden för att svänga från ena ändläget till mitten adderat med tiden för att svänga från mitten till andra ändläget. Genom att studera mätdata som illustreras i figur 6.1 togs tabell 6.3 fram, där tiden för vänster→höger samt för höger→vänster är beräknade enligt ovan. Den långsammaste rörelsen var alltså höger→vänster som tog 0,96 s. Kravet på detta var ≤ 1 s, vilket systemet klarar. Däremot var önskemålet $\leq 0,5$ s, vilket inte uppnåddes.

Tabell 6.3: Uppmätt och beräknad tid för styrningsrörelser

Rörelse	Tid [s]
Vänster→centrum	0,46
Centrum→höger	0,40
Höger→centrum	0,62
Centrum→vänster	0,34
Vänster→höger	0,86
Höger→vänster	0,96

Noggrannhet

Även detta krav utlästes från figur 6.1 genom att finna den maximala skillnaden mellan det önskade styrvärdet och det verkliga styrvärdet efter insvängningen. Detta inträffade vid $t = 9,4$ s där det önskade styrvärdet var 255, medan det verkliga styrvärdet var 182. Ett maximalt fel på $255 - 182 = 73$ med en skala på 0-255 kan översättas till en noggrannhet på $\pm \frac{73}{255} \approx \pm 29$ procentenheter. Resultatet möter inte kravet på ± 5 procentenheter eller önskemålet på ± 1 procentenhet.

6.2 Hastighetsreglering

Vilka krav samt önskemål som har uppfyllts och ej uppfyllts visas i tabell 6.4. Eftersom det inte fanns GPS eller annat sätt att logga hastigheten elektroniskt, fick personen på RescueRunnern jämföra hur hastigheten upplevdes gentemot i manuell läge. I kombination med vattenskoterns inbyggda hastighetsmätare kunde det bildas en uppskattning av hur fjärrstyrt och manuellt läge var gentemot varandra.

Tabell 6.4: Kravspecifikation och resultat av styrning av gas

Kriterier	Målvärde	Krav/önskemål	Uppfylls
Styra gasen i hela dess register	95% av manuell	Krav	Nej
Tid mellan ändlägen	≤ 1 s	Krav	Ja
Tid mellan ändlägen	$\leq 0,5$ s	Önskemål	Ja
Upplösning	≥ 10 steg	Krav	Ja
Upplösning	≥ 50 steg	Önskemål	Ja
Noggrannhet	± 5 procentenheter	Krav	Nej
Noggrannhet	± 3 procentenheter	Önskemål	Nej

Styra gasen i hela dess register

Gasen kunde uppskattningsvis utnyttjas från 5% till 10% gaspådrag, vilket gör att kraven ej uppfylls, med avseende på dess register.

Tid mellan ändlägen

Eftersom systemet är helt elektroniskt kommer tiden mellan ändlägena endast vara den tid från det att Arduinon på RescueRunner skickar signalen till den digitala potentiometern, till dess att gasventilerna tar emot signalen från ECM:en. Elektriska signaler skickas oerhört fort, och det var ingen märkbar fördröjning vid testning av systemet. Utifrån detta kan det antas att tiden mellan ändlägena är mycket mindre än både kravet och önskemålet, på 1 s respektive 0,5 s.

Upplösning

Ekvation (3.3) beräknar stegen mellan 0% och 100% gaspådrag. Detta visar att det används $0,91 \cdot 100\% + 19,5 - (0,91 \cdot 0\% + 19,5) = 91$ steg på potentiometern, vilket både är mer än både kravet och önskemålet på 10 respektive 50 steg.

Noggrannhet

Av samma anledning som att kravet för att styra gasen i hela dess register inte uppfylls, nås heller inte kravet om noggrannhet på systemet, då gaspådraget uppskattningsvis inte överstiger 10%, även om den önskade gasen gör det.

7

Diskussion

Trots att det är många krav i den generella kravspecifikationen som inte uppfylls, har projektet nått stora delar av de mål som sattes. Problemen som förhindrar att kraven uppfylls är varken många eller stora, och potentiella lösningar har utretts. Det fanns inte tillräckligt mycket tid för att implementera och testa dessa lösningar, men om detta skulle göras borde de flesta kraven bli uppfyllda.

7.1 Reglering av styrvinkel

Under senare delen av projektet fanns möjlighet att undersöka en RescueRunner av standardmodell. Det noterades då att styret på denna gick betydligt lättare att vrida på än det på prototypen. Tack vare detta är det möjligt att en mindre motor skulle ha kunnat användas för att manövrera styrningen. Något som ej har tagits hänsyn till då är att styret troligtvis går betydligt tyngre att vrida på under full gas på grund av vattenströmmen genom dysan. Det är därför möjligt att en stor motor kan krävas även fast styret går lättare på standardmodellen än på prototypen.

Bromshandtaget spänns som tidigare nämnts med ett spännband för att koppla in styrmotorn på styret för att gå över till fjärrstyrt läge. Denna lösning är inte optimal, då föraren måste komma ihåg att lämna båten med spännbandet draget för att det inte ska behöva dras innan avfärd vid en uttryckningssituation. Dessutom är handtaget väldigt omständigt för föraren att koppla till respektive från då båda händerna krävs, vilket kan orsaka problem vid grov sjö. Önskvärt vore att trådlöst ha möjlighet att växla mellan manuellt och fjärrstyrt läge. Detta måste dock kunna ske på ett mycket pålitligt och säkert sätt, så ingen risk finns att RescueRunnern låses i fjärrstyrt läge och inte kan manövreras av föraren.

Elmotorn som valdes är bara IP53-klassad och möter därmed inte önskemålen. Samma modell av elmotorn går att få IP66-klassad men att beställa hem en sådan hade tagit för lång tid. Dessutom uppfyller även en sådan motor inte önskemålet på IP67-klassning. Motorn som valts är dock rejält stänktålig och håller därför så länge RescueRunnern inte välter eller hamnar under vattnet trots den lägre klassningen.

Att kravet om noggrannhet för styrningen, som enligt avsnitt 6.1, inte uppfylldes beror på att PID-regulatorn inte lyckas driva styret till den önskade vinkeln. Detta kan grunda sig i ett flertal faktorer, varav den troligaste är att parametrarna för regulatorn inte är optimalt inställda. Parametrarna togs fram experimentellt. En orsak till att de framtagna parametrarna inte fungerade bra skulle kunna vara att styret går olika tungt att vrida på åt höger och vänster. Den dåliga noggrannheten beror alltså inte på hårdvarulösningen utan på regulatorn. Värt att notera är även att noggrannheten är betydligt högre (ca ± 9 procentenheter) vid sväng från ändlägen till centrum, och tvärt om. Den låga noggrannheten som syns i resultatet uppstår alltså enbart vid sväng från ändläge till ändläge, en rörelse som inte tros användas ofta vid normal framföring av RescueRunnern.

Problemet med att fjädern till återkopplingen på styret töjdes ut i en tidigare lösning, som nämndes i 2.2.3, visade sig bli ett problem även i detta projekt. Fjädrarna som används för att vrida tillbaka potentiometern töjs nu ut om styret lämnas i fel läge för länge, och sensorn slutar då att fungera. Detta trots att nya, mer passande fjädrar användes.

Potentiella lösningar till nämnda problem och vidareutveckling av regleringen av styrvinkeln diskuteras i 7.4.1.

7.2 Hastighetsreglering

Lösningen fungerar, men inte bra. När APS:en kopplas bort, och den digitala potentiometern tar över, leder det till att gaspådraget är uppskattningsvis 5% när den ska vara i nolläge, och 10% när den ska vara i maxläge. Detta trots att lösningen som användes baseras på den som tidigare har använts med gott resultat enligt rapport. Vad det beror på är inte helt säkerställt – kopplingarna är korrekta, vilket har kontrollerats flera gånger. Det kan bero på att ECM:en reagerar på att den endast matas med en potentiometer, i kontrast till det befintliga systemet som har två. Om detta är fallet borde det ha kommenterats i rapporten från 2015, vilket det inte gjorts. I det projektet begränsades hastigheten till 20%, och det kan vara därför felet aldrig upptäcktes.

Efter att felet upptäcktes kontrollerades APS1 och APS2. Den totala resistansen över dessa är 4,60 k Ω respektive 4,68 k Ω . Detta kan vara anledningen till det oväntade resultatet, eftersom den digitala potentiometern som används har en total resistans på 10 k Ω .

Föraren av RescueRunnern noterade att det kändes som att maxhastigheten var mycket lägre efter att fjärrstyrningsläget var igång, men att det återställdes efter att motorn stoppades och startades om. Skillnaden i maxhastighet kan bero på att vattenskoterns elektroniska styrsystem upptäcker att det endast är en potentiometer

kopplad till ECM:en, eller att den totala resistansen över potentiometern är för hög, och uppfattar det som fel på det elektriska systemet. Det kan vara så att maxhastigheten blir begränsad av säkerhetsskäl tills att systemet startar om och inget fel upptäcks.

Potentiella lösningar till nämnda problem och vidareutveckling av hastighehetsregleringen diskuteras i 7.4.2.

7.3 Elektronik

Vad gäller radions räckvidd är det förvånande att resultatet som uppmättes endast blev 12,5m då mastersarbetet 2016 fick 500m med exakt samma kort ute på vattnet. Testning av radioräckvidden skedde dock där det fanns över 80 WiFi-nätverk. Visserligen var inte alla av 2,4GHz-sort, men det är rimligt att anta att resultatet hade blivit högre utan störningar. Det kan också vara så att radiokorten eller antennerna har blivit skadade mellan projekten.

7.4 Vidareutveckling

För att SSRS ska kunna använda det framtagna systemet behöver en del arbete göras för att lämna prototypstadiet. I detta avsnitt presenteras några av dessa vidareutvecklingsförslag.

7.4.1 Reglering av styrvinkel

En lösning på problemet med bromshandtaget, vilket spänns för att aktivera fjärrläget, skulle kunna vara att använda en mindre linjär aktuator. Aktuatoren skulle dra i bromsvajern med någon sorts nödkoppling så att den skulle kunna kopplas bort. Nödkopplingen skulle i så fall inte användas under normal drift, utan enbart användas ifall något oväntat sker så att systemet ej kopplar ifrån.

Problemet med att fjädrarna till återkopplingen för styrvinkeln töjs ut skulle kunna lösas genom att omarbete sensorn för att använda en givare med inbyggd fjäder. Ett annat alternativ är att byta till en fjäder med ännu högre kvalitet. Ett tredje alternativ är att ta fram en helt ny givare för styrvinkeln som fungerar på något annat sätt, eller monteras direkt på styret istället för via styrvajern.

Något annat som borde förbättras är monteringsanordningen till elmotorn. Plåten är just nu tillverkad i ett stål som inte är rostfritt och kommer följaktligen att börja rosta om den inte behandlas med något rostskydd. Ett annat alternativ är att tillverka en ny plåt i exempelvis aluminium eller rostfritt stål. Dessutom är

motorn, som nämnts ovan, inte vattentät. Ett alternativ för att lösa detta skulle vara att välja en annan motor med liknande egenskaper med minst IP67-klassning. Ett annat alternativ skulle vara att montera något runt motorn som gör att vattnet inte kommer åt den.

För att ta fram mer passande parametrar till PID-regulatorn behöver en matematisk modell av RescueRunnern (från styret till dysan) tas fram. Med hjälp av denna modell skulle parametrarna kunna beräknas och systemet simuleras. Modellen skulle dock vara mycket olik den för standardmodellen av RescueRunnern då styret gick betydligt tyngre på prototypen som projektet utfördes på. Om detta ska göras bör det alltså göras för standardmodellen istället för prototypen för att vara meningsfullt.

En lösning på att styret kräver olika stort vridmoment vid höger- och vänstersväng skulle kunna vara att använda olika parametrar för respektive håll. En bättre lösning skulle dock vara att göra någonting åt att styret går olika tungt, alternativt använda standardmodellen av RescueRunnern istället för prototypen.

7.4.2 Hastighetsreglering

Som tidigare nämnts används en digital potentiometer med en total resistans på 10 k Ω medan det i efterhand har uppmäts att APS1 har en total resistans 4,60 k Ω och APS2 har en total resistans 4,68 k Ω . Detta kan vara en av anledningarna till att det inte fungerar som det ska. Optimalt sätt borde potentiometern ha exakt samma egenskaper som APS:ens potentiometrar, men genom att använda en digital potentiometer med en total resistans på 5 k Ω skulle detta kunna förbättra hastighetsregleringen eftersom det är närmre det ursprungliga värdet.

En annan möjlig lösning till problemet med hastighetsregleringen är att ersätta både APS1 och APS2 med två separata digitala potentiometrar, istället för endast en som det är i dagsläget. Detta är fullt möjligt, men skulle kräva att skölden delvis görs om.

7.4.3 Dataöverföring

Datan som skickas mellan RR.A och LB.A sker helt öppet och okrypterat, vilket kan leda till att vem som helst kan läsa av meddelandena och sedan skicka egna och på så sätt kapa RescueRunnern. I framtiden bör varje meddelande antingen krypteras med en nyckel eller avslutas med en hemlig kontrollsumma för att verifiera att ingen har modifierat datan.

Dessutom skickas inget ID med i meddelandet, något som behövs när systemet vidareutvecklas för att stödja flera RescueRunner. Visserligen skickas datan på en viss radio-adress som kan ändras, men då denna i nuläget är hårdkodad måste koden kompileras om för varje ny RescueRunner. Önskvärt hade varit att snabbt kunna justera adressen när RescueRunnern behövs i ett räddningsuppdrag.

En annan sak som kan vara bra att implementera är att använda TCP med automatisk omsändning istället för UDP. I ett mer effektivt system skickas bara de meddelanden som inte gått fram igen, och det skickas inget mer efter att `RR.A` har läst det senaste meddelandet.

7.4.4 Radioräckvidd

För att höja räckvidden för radiokommunikationen skulle RescueRunnern kunna byggas om så att antennen ingår i chassit. På nyare modeller skulle det utbytbara undre skalet kunna designas med en metallring ovanpå som kan fungera som antenn. Genom denna modifiering skulle inte positionen av en eventuell förare störa radiokommunikationen. Ett enklare alternativ skulle kunna vara att montera en antenn längst upp på RescueRunnern, alternativt byta till en annan typ av radiokort med lång räckvidd.

7.4.5 Dataloggning

Lösningen som skapades för att logga till SD-kortet är temporär och bör fixas till. Att ha en egen Arduino endast för loggning är ineffektivt, både till processorkraft och plats i lådan. Vidare är den nuvarande lösningen inte särskilt robust eftersom `RR.A` och `RR.A_SD` ansluts via två lösa kablar. Ett förslag är att införskaffa en annan typ av SD-kortsadapter som inte är felkonstruerad från fabrik och kan anslutas direkt på skölden till `RR.A`. Ett annat förslag är att konstruera en egen SD-kortsadapter.

7.4.6 Följningsfunktion

Ambitionen var från början att använda den givna kodbas för FollowMe-algoritmen som fanns sedan tidigare och anpassa denna till den mekaniska lösning som byggdes [6]. Denna tanke slopades dock då koden inte var komplett; filer och metoder saknades. Även om projektet begränsade detta utvecklingsområde, har fortfarande informationssökning skett för att lösa andra problem relaterade till följningsfunktionen. Förhoppningen är att detta kan användas av framtida projekt.

Ett problem med den tidigare koden var att den använder sig av enkelpunktsnavigering (single point) och har därför ± 5 m i precision [25]. För att få position med hög precision brukar realtidsrörelse (RTK) användas. Den låga noggrannheten kommer av störningar när satelliternas data färdas genom luften [25]. RTK går ut

på att använda sig av en basstation som är ordentligt inmätt så att felets storlek där blir känt. Informationen om felet skickas därefter från basstationen till det fordon där positionen är önskad, och på så sätt kan precision på centimeternivå uppnås [26].

RTK passar dock inte användningsområdet för följning av ledarbåten; för att det mottagna felet ska vara motsvarande på både basstationen och RescueRunner måste de vara som mest 20 km ifrån varandra [27]. Visserligen har SSRS över 70 stationer utspridda längst kusterna, men att en olycka alltid skulle ske just inom 20 km är inte troligt [1]. Dessutom skulle det behövas internetuppkoppling på både ledarbåten och RescueRunnern. Andra lösningar undersöktes därför.

Broadcom utvecklar ett nytt GPS-kort med namn BCM47755 för smarttelefoner som kommer ge noggrannhet på under 1 m genom att använda två frekvensband [28]. Till skillnad från existerande flerbandslösningar kommer kortet användas i konsumentprodukter, och blir därför billigt och lätt att få tag på. Dock är det inte ute på marknaden förrän om ett par år [28].

En annan kandidat är den så kallade rörlig bas (moving base)-lösningen. Denna lösning tillåter istället att basstationen rör på sig [29]. En relativ position tillhandahålls medan den absoluta positionen är oanvändbar [29]. Rörlig bas-lösningen skulle i teorin passa användningsområdet perfekt då basstationen kan placeras på ledarbåten.

Referenser

- [1] Sjöräddningssällskapet, “Om oss,” 2018. [Online]. Tillgänglig: <https://www.sjoraddning.se/om-oss>, hämtad: 2018-05-14.
- [2] A. Andersson, V. Bäckman, O. Grankvist, A. Hjelmström Sarvik och O. Pantzare, “Autonom eskortering av Rescuerunner,” Kandidatarbete, Signaler och system, Chalmers Tekniska Högskola, Göteborg, Sverige, 2015.
- [3] H. Berg, G. Lindberg och J. Skötte, “Elektronisk manövrering av Rescuerunner,” Kandidatarbete, Signaler och system, Chalmers Tekniska Högskola, Göteborg, Sverige, 2016.
- [4] A. Bergholtz och C.-A. Hernvall, “Waypoint-based path following system for a jet ski,” Examensarbete, Signaler och system, Chalmers Tekniska Högskola, Göteborg, Sverige, 2015.
- [5] B. Hallhagen och E. Persson, “Adapting a small watercraft to autonomously follow a lead boat,” Examensarbete, Signaler och system, Chalmers Tekniska Högskola, Göteborg, Sverige, 2016.
- [6] A. Lydh, N. Hellberg och M. Dabarera, “Documentation and implementation for the Rescue runner,” Projekt, Signaler och system, Chalmers Tekniska Högskola, Göteborg, Sverige, 2018.
- [7] A. Andersson, V. Bäckman, O. Grankvist, A. Hjelmström Sarvik och O. Pantzare, “Skiss över befintligt styrsystem. Dysan är mynningen från vattenjetmotorn, Rescuerunnern svänger när dysans vinkel ändras,” 2015. [Elektronisk bild].
- [8] *Linear drive installation instructions*, England: Raymarine, 2016. [Online]. Tillgänglig: <https://raymarine.app.box.com/s/nv3eue7rj4dzuvhm4j2d/file/56058597782>, hämtad: 2018-05-14.
- [9] *DOGA Drive Systems*, Barcelona, Spanien: DOGA, 2017. [Online]. Tillgänglig: http://www.doga.es/assets/doga_drive_systems_17.pdf, hämtad: 2018-05-14.
- [10] *EM-243C DC-Motor Controller 12-48V 50A*, Åbo, Finland: ELECTROMEN, 2018. [Online]. Tillgänglig: https://electromen.com/files/1915/1867/6344/EN_em-243C.pdf, hämtad: 2018-05-14.
- [11] *WaveRunner VX110 Sport VX110 Deluxe SERVICE MANUAL*, USA: YAMAHA MOTOR CORPORATION, 2005.

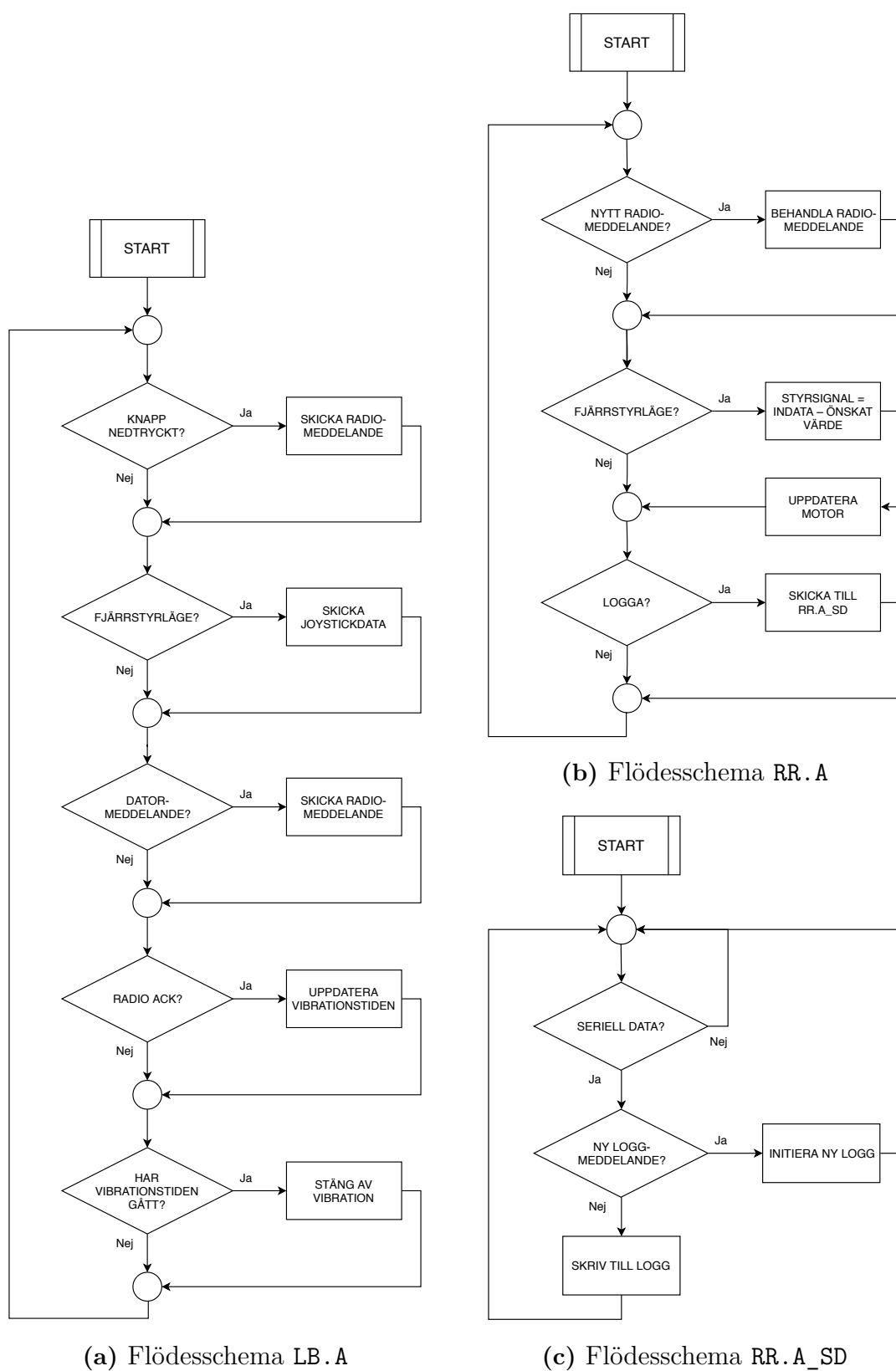
- [12] *MCP41XXX/42XXX*, USA: Microchip Technology Inc., 2003. [Online]. Tillgänglig: <http://ww1.microchip.com/downloads/en/DeviceDoc/11195c.pdf>, hämtad: 2018-05-14.
- [13] Arduino, "Introduction," [Online]. Tillgänglig: <https://www.arduino.cc/en/Guide/Introduction>, hämtad: 2018-05-14.
- [14] —, "Compare board specs," [Online]. Tillgänglig: <https://www.arduino.cc/en/Products/Compare>, hämtad: 2018-05-14.
- [15] G. Duffy, "What is network latency and how do you use a latency calculator to calculate throughput?," [Online]. Tillgänglig: <https://www.sas.co.uk/blog/what-is-network-latency-how-do-you-use-a-latency-calculator-to-calculate-throughput>, hämtad: 2018-05-14.
- [16] J. Wright, "Dispelling Common Bluetooth Misconceptions," [Online]. Tillgänglig: <https://www.sans.edu/cyber-research/security-laboratory/article/bluetooth>, hämtad: 2018-05-14.
- [17] J. Mrázek, "Testing the WiFi range of ESP32," [Online]. Tillgänglig: <https://blog.honzamrazek.cz/2017/08/testing-the-wifi-range-of-esp32/>, hämtad: 2018-05-14.
- [18] *nRF24L01+ Single Chip 2.4GHz Transceiver Product Specification v1.0*, Trondheim, Norge: NORDIC SEMICONDUCTOR, 2008. [Online]. Tillgänglig: https://www.nordicsemi.com/eng/content/download/2726/34069/file/nRF24L01P_Product_Specification_1_0.pdf, hämtad: 2018-05-14.
- [19] *SD Specifications Part 1 Physical Layer Simplified Specification Version 6.00*, USA: SD Card Association, 2017. [Online]. Tillgänglig: https://www.sdcard.org/downloads/pls/pdf/index.php?p=Part1_Physical_Layer_Simplified_Specification_Ver6.00.jpg&f=Part1_Physical_Layer_Simplified_Specification_Ver6.00.pdf&e=EN_SS1, hämtad: 2018-05-14.
- [20] V. M. Aiea, "Interfacing Catalex Micro SD Card Module with Arduino," [Online]. Tillgänglig: <http://www.vishnumaiea.in/projects/hardware/interfacing-catalex-micro-sd-card-module>, hämtad: 2018-05-14.
- [21] B. Porter, "PlayStation 2 Controller Arduino Library v1.0," 2010. [Online]. Tillgänglig: www.billporter.info/2010/06/05/playstation-2-controller-arduino-library-v1-0/, hämtad: 2018-05-14.
- [22] rhydoLABZ, "Interfacing PS2 Wireless Controller With Arduino," [Online]. Tillgänglig: <http://www.rhydolabz.com/wiki/?p=12663>, hämtad: 2018-05-14.
- [23] "Optimized High Speed NRF24L01+ Driver Class Documenation," 2016. [Online]. Tillgänglig: <http://tmrh20.github.io/RF24/>, hämtad: 2018-05-14.
- [24] Arduino, "Ardunio Reference (int)," [Online]. Tillgänglig: <https://www.arduino.cc/reference/en/language/variables/data-types/int/>, hämtad: 2018-05-14.
- [25] GPS.gov, "GPS Accuracy," [Online]. Tillgänglig: <https://www.gps.gov/systems/gps/performance/accuracy/>, hämtad: 2018-05-14.

- [26] Novatel, “Real-Time Kinematic (RTK),” [Online]. Tillgänglig: <https://www.novatel.com/an-introduction-to-gnss/chapter-5-resolving-errors/real-time-kinematic-rtk/>, hämtad: 2018-05-14.
- [27] A. Rietdorf, C. Daub och P. Loef, “Precise Positioning in Real-Time using Navigation Satellites and Telecommunication,” Tyskland, 2006. [Online]. Tillgänglig: http://wpnc.net/fileadmin/WPNC06/Proceedings/34_Precise_Positioning_in_Real-Time_using_Navigation_Satellites_and.pdf, hämtad: 2018-05-14.
- [28] S. Riley, H. Landau, V. Gomez, N. Mishukova, W. Lentz och A. Clare, “Positioning with Android: GNSS observables,” 2018. [Online]. Tillgänglig: <http://gpsworld.com/positioning-with-android-gnss-observables/>, hämtad: 2018-05-14.
- [29] T. Everett, “Exploring moving-base solutions,” 2016. [Online]. Tillgänglig: <https://rtklibexplorer.wordpress.com/2016/12/29/exploring-moving-base-solutions/>, hämtad: 2018-05-14.

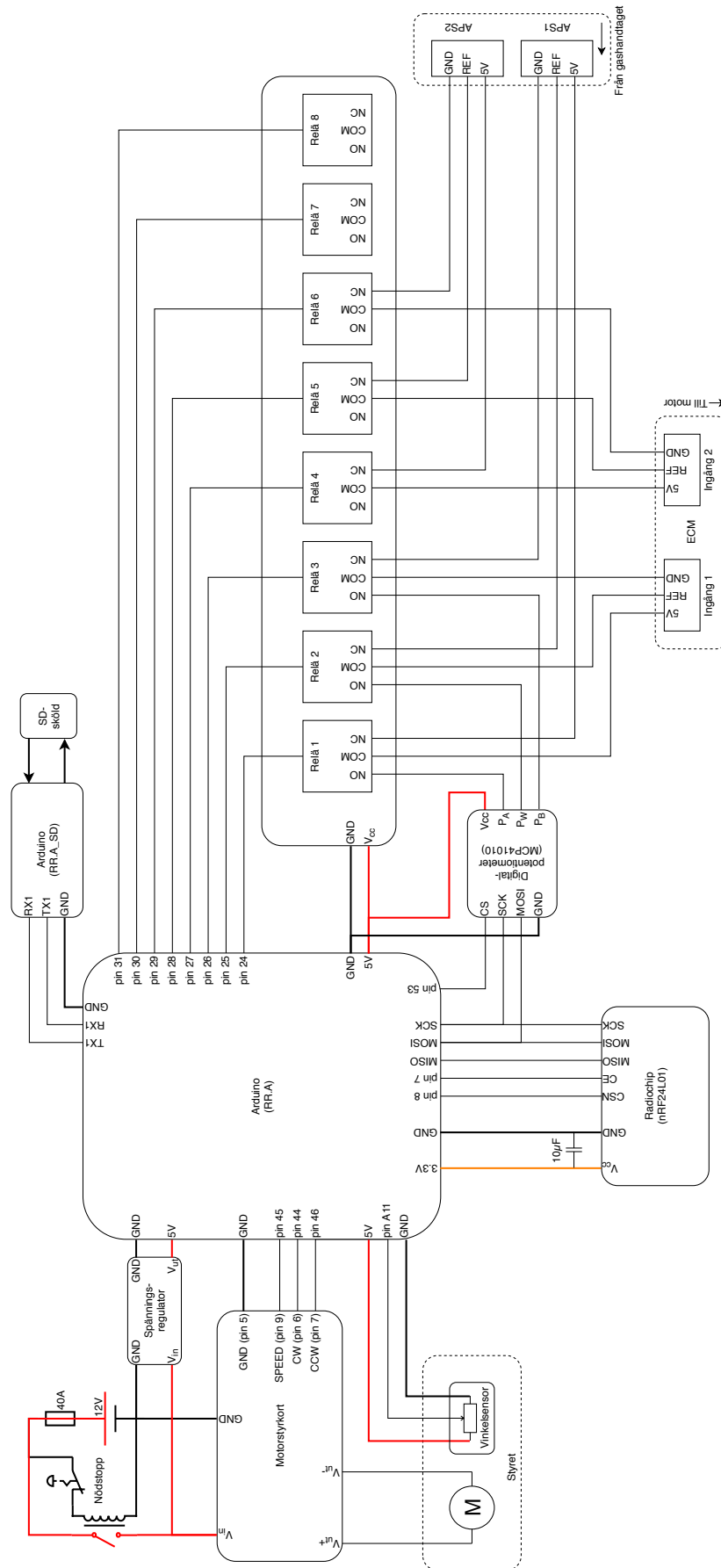
A

Stora figurer

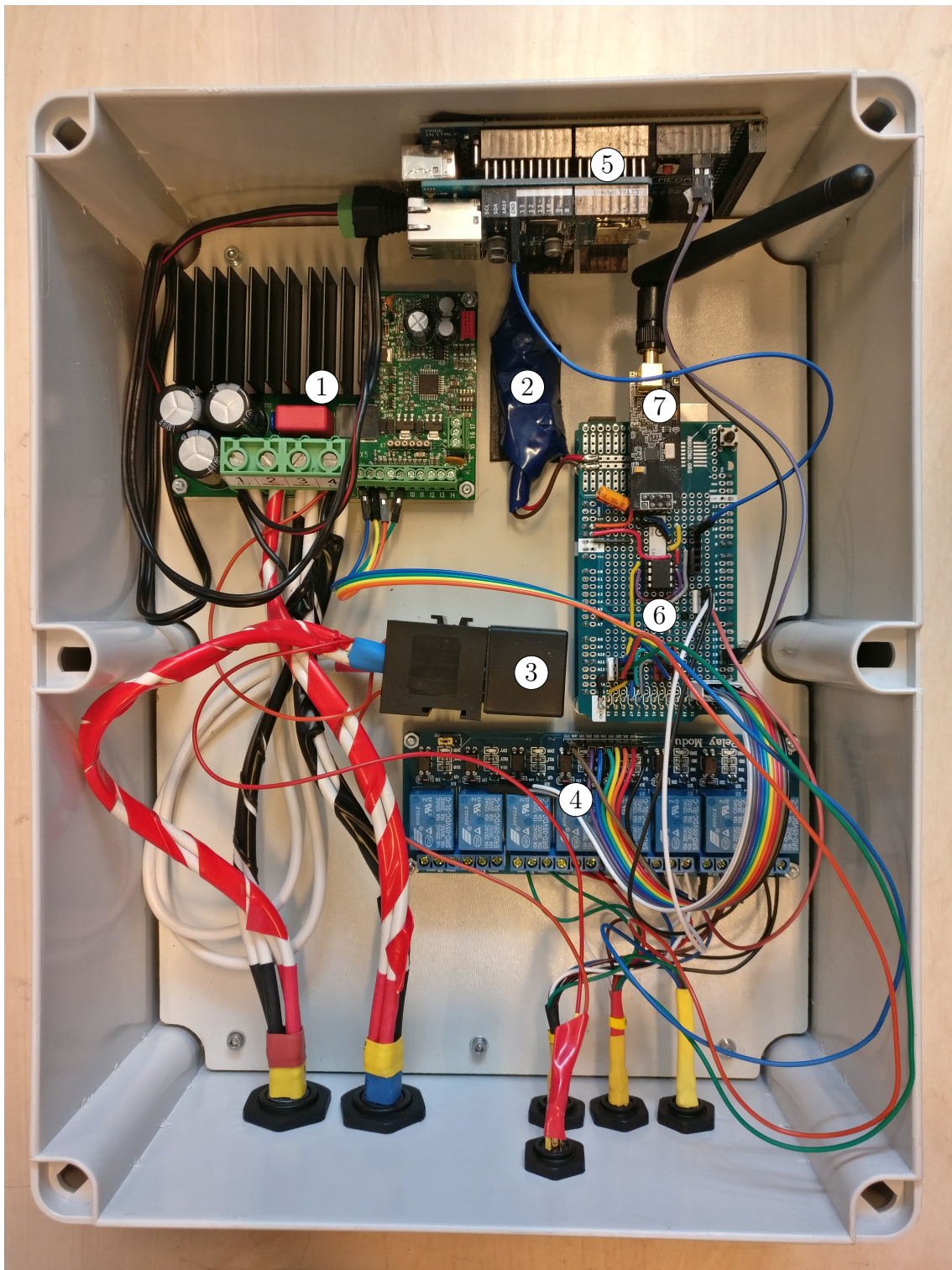
A.1	Flödesscheman över programmens exekvering	II
A.2	Kopplingsschema av det framtagna elsystemet	III
A.3	Bild på elektroniklådan till RescueRunnern	IV
A.4	Ritning av bromsadapter	V
A.5	Ritning av centrering	VI
A.6	Ritning av motorhållare	VII
A.7	Ritning av okplåt och okdistans	VIII
A.8	Ritning av potentiometerhjul till styrningsåterkoppling	IX



Figur A.1: Flödesscheman över programmets exekvering

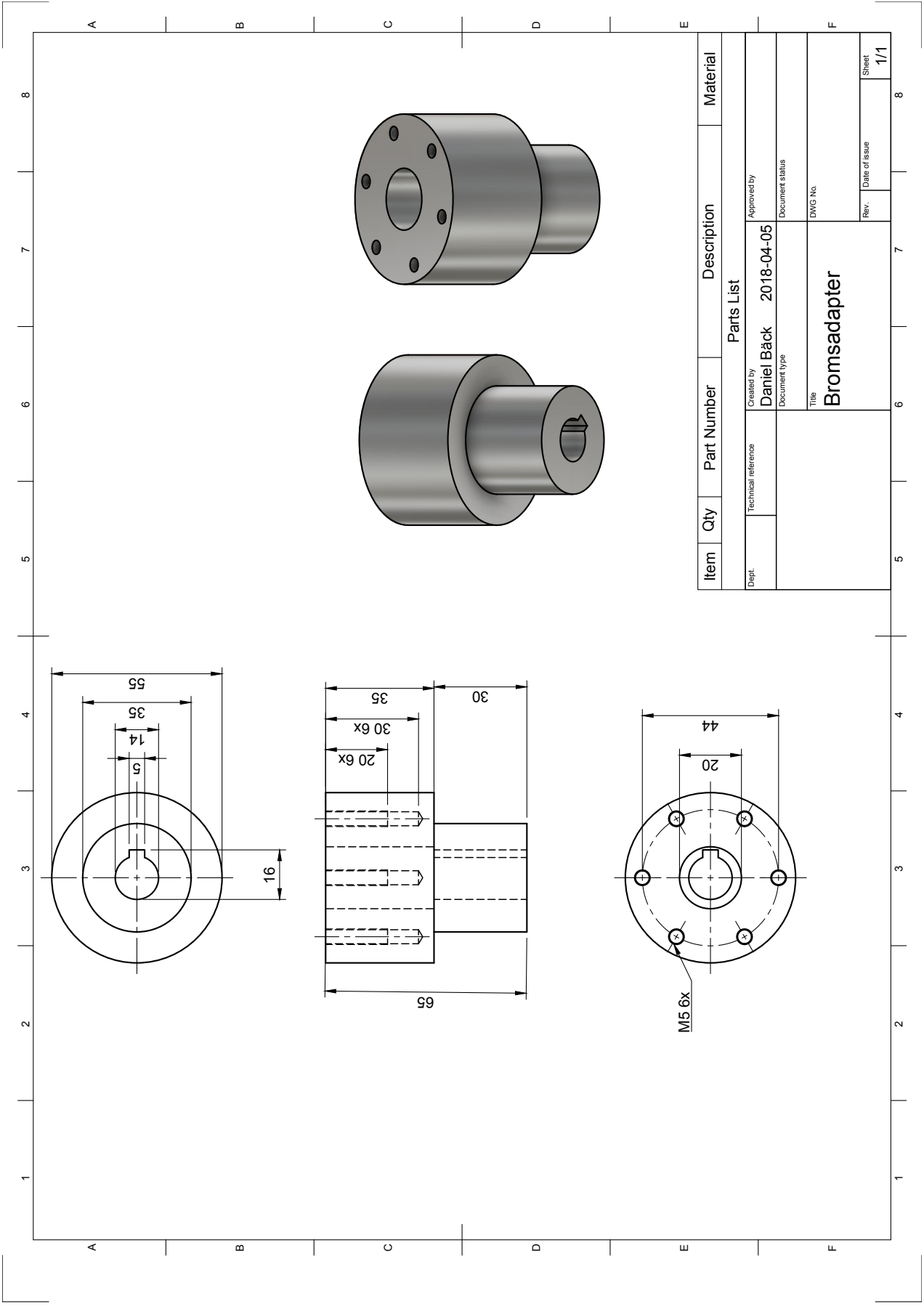


Figur A.2: Kopplingschema av det framtagna elsystemet

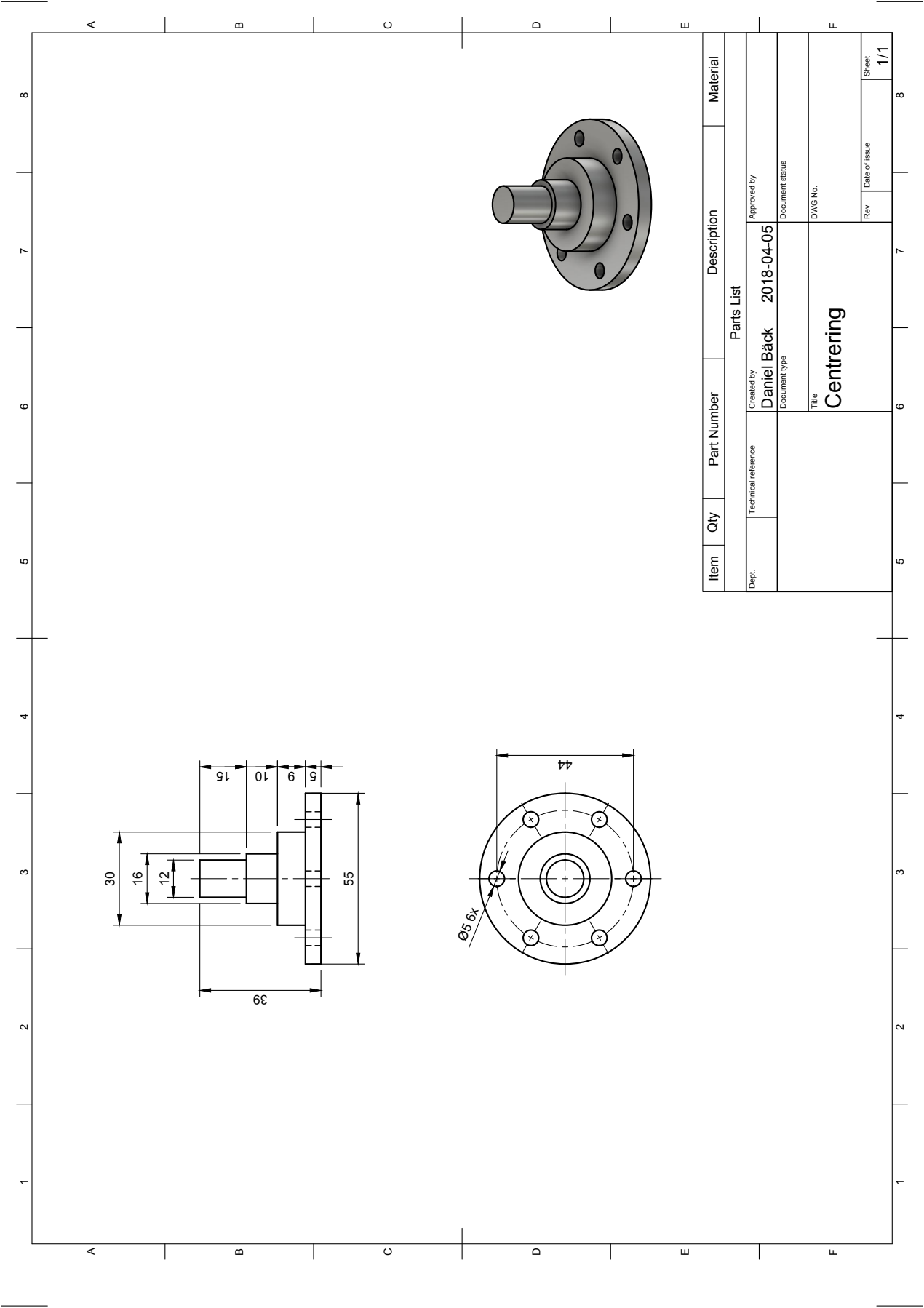


- ① Motorstyrkort ② Spänningsregulator ③ Fordonsrelä ④ Relämodul
⑤ Arduino med SD-kortläsare ⑥ Arduino med sköld ⑦ Radiokort

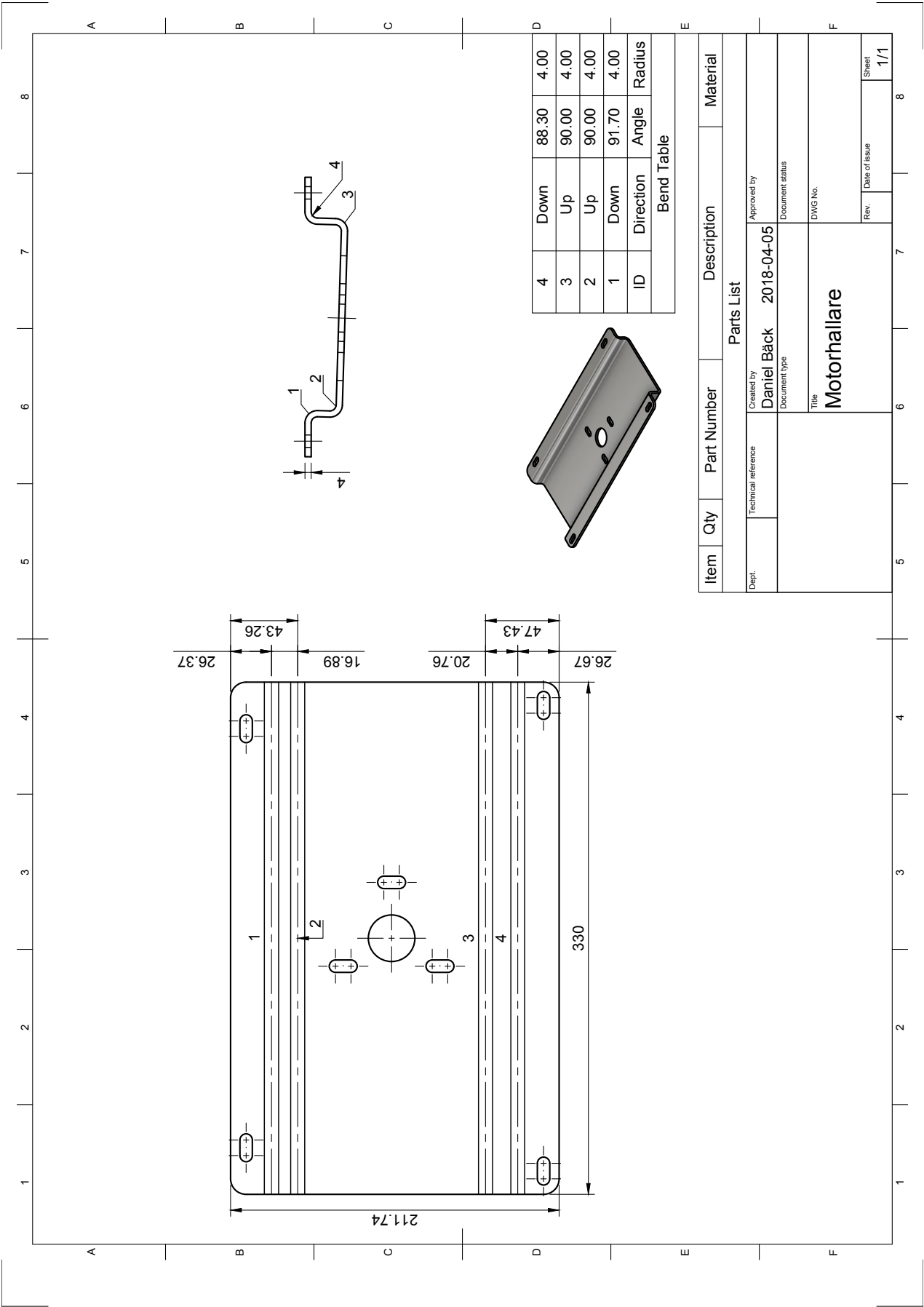
Figur A.3: Bild på elektroniklådan till RescueRunnern



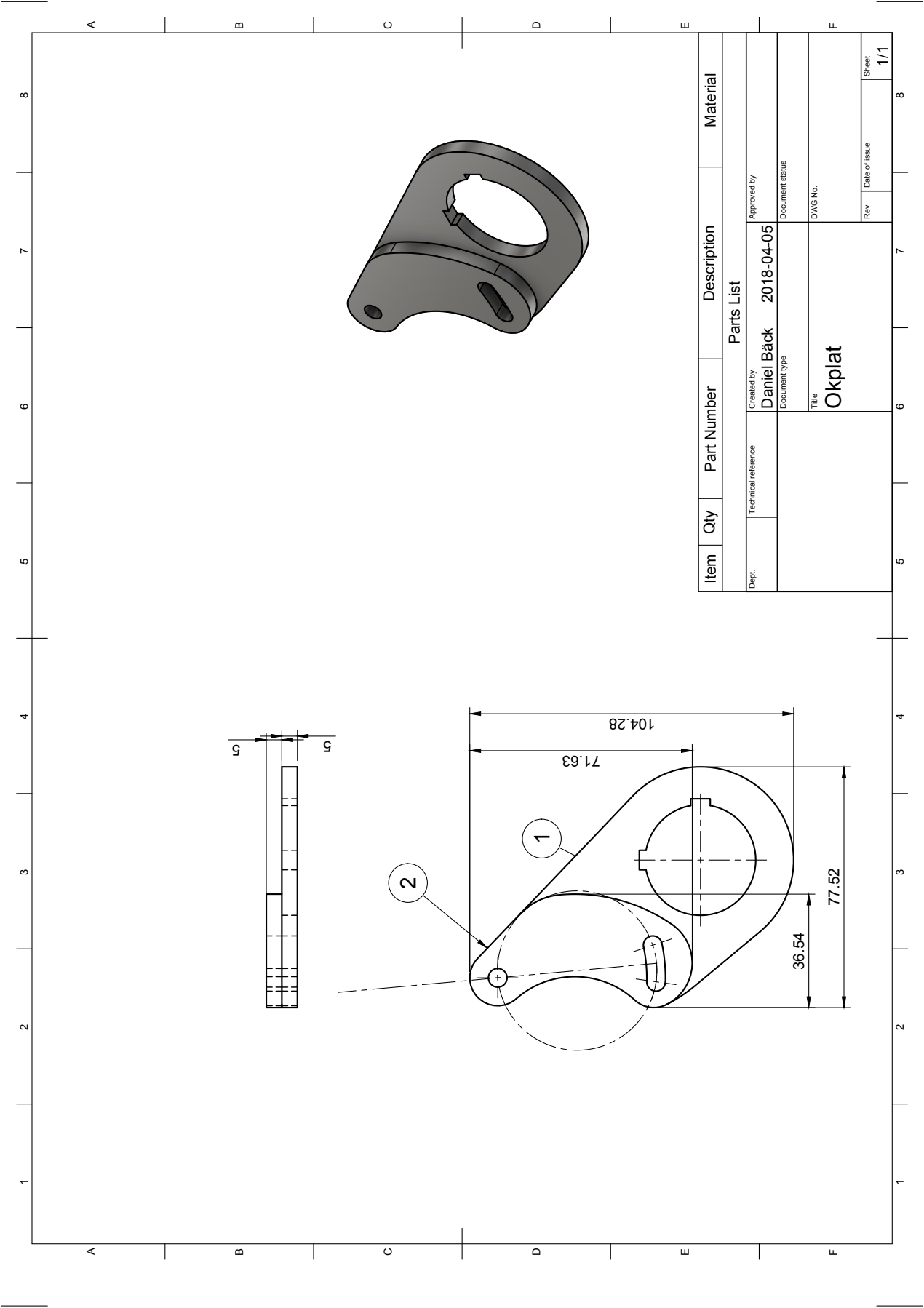
Figur A.4: Ritning av bromsadapter



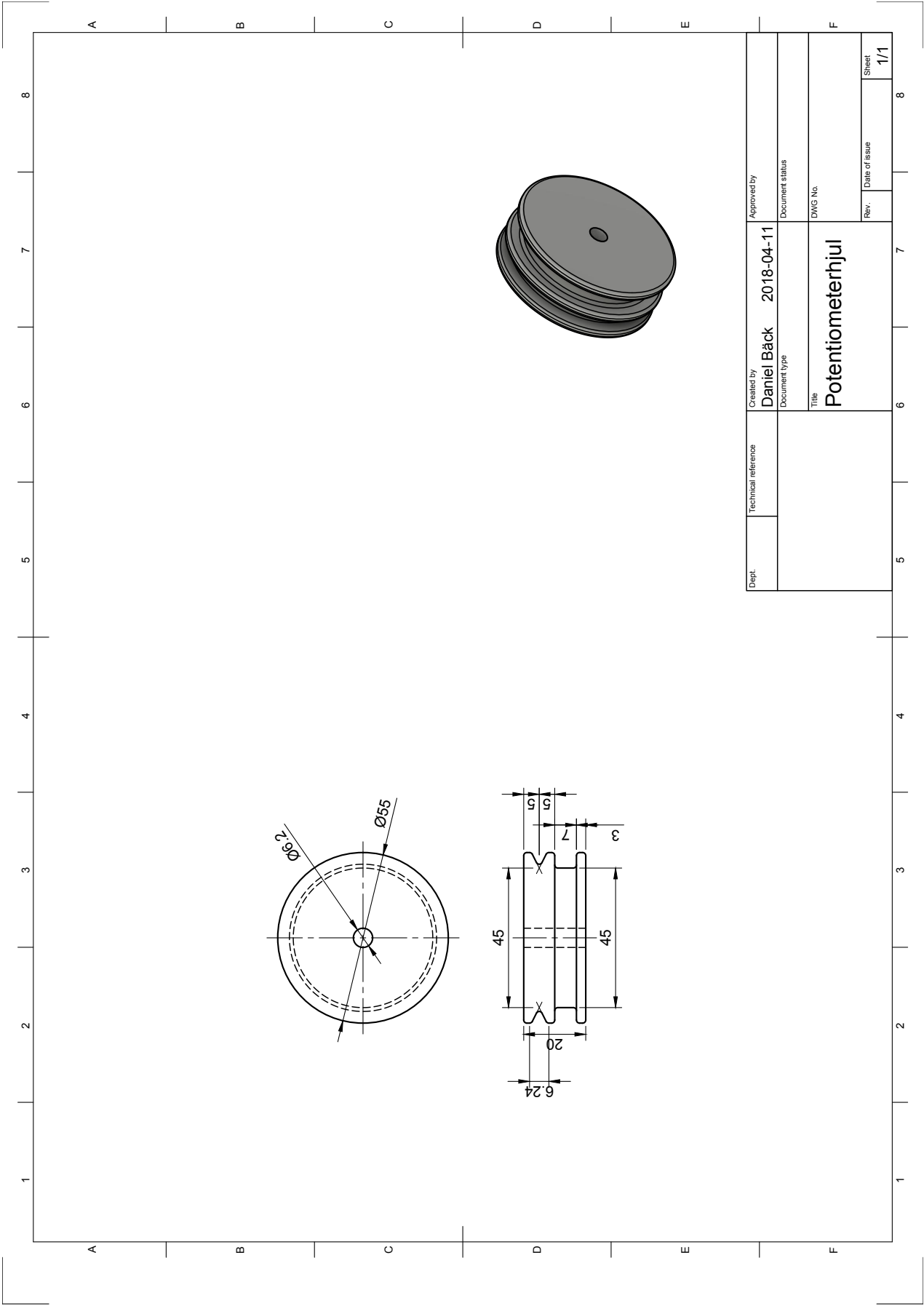
Figur A.5: Ritning av centrer



Figur A.6: Ritning av motorhallare



Figur A.7: Ritning av okplåt och okdistans



Figur A.8: Ritning av potentiometerhjul till styrningsåterkoppling

B

Komponentlista

I denna bilagan listas komponenter som har köpts in, använts eller producerats under projektets gång. CAD-filer och kod finns att hämta i en mapp ¹ som administreras av Petter Falkman på onlinelagringstjänsten Box. All kod finns även på GitHub².

1 Kontaktdon

- **Weipu SP13 6-pin** Används för signaler. IP68-klassade.
- **Weipu SP21 4-pin** Används för strömmatning. IP68-klassade.
- **Sumitomo DL .090 8-pin** Kontakten som används i vattenskoterns befintliga system för att koppla APS:en till ECM:en. Hon-kontakten är möjlig att beställa från tredjepart, men han-kontakten kunde inte hittas och blev därför designat i CAD och utskriven med en 3D-skrivare. Filnamn: `Kontakt.itp`, `LockKontakt.itp`, `pin_holder.step`.

2 Elektronik

- **Arduino** Två i RescueRunnern: en för att kontrollera motor och gas, samt radiokommunikation, och en för loggning till microSD-kort. En i land för fjärrstyrning med PS2-kontrollare och radiokommunikation.
- **microSD-läsare** Används för att spara loggar från Arduino till ett SD-kort.
- **MCP41010** Digital potentiometer. Används för att reglera gasen.
- **PS2-kontroller** Används för att kontrollera RescueRunnern i fjärrstyrningsläge.
- **nRF24L01** Används för radiokommunikation mellan Arduinos.
- **Nödstop** Används för att kontrollera fordonsreläet.
- **Fordonsrelä 70A** Bryter all ström till elektroniken om nödstoppet trycks in.

¹<https://chalmersuniversity.box.com/s/95p5pggtdghc41ussjo90e1r7wmwyki6>

²<https://github.com/Nosskirneh/RemoteControlRR>

- **Relämodul 8 kanaler** Används för att byta mellan APS och MCP41010. APS är kopplat i normally closed, och MCP41010 i normally open.
- **EM-243C** Motorstyrkort. Kontrollerar elmotorn, styrs av en Arduino. Inköpt från <https://www.oemmotor.se/>.
- **EM-328** Programmeringsadapter för motorstyrkortet för att ändra inställningar. Inköpt från <https://www.oemmotor.se/>.
- **Spänningsregulator** Sänker batteriets 12 V till 5 V för matning av Arduino.

3 Motor och fäste

- **Skivbroms AVID Ball Bearing 5 MTB 160 mm** Cykelskivbroms som används för att koppla till och från styrmotorn från styret.
- **Doga 259.3710.20.00** 12 V motor som används för att vrida på styret. Inköpt från <https://www.oemmotor.se/>.
- **Motorhållare** Plåt att montera motorn på.
Filnamn: Sammanställning.ipt, Motorhallare Drawing.pdf.
- **Bromsadapter** Del som monteras på motorns utgående axel för att montera skivbromsen på.
Filnamn: Sammanställning.ipt, Bromsadapter Drawing.pdf.
- **Centrering** Monteras på skivbromsen för att med hjälp av kullagret centrera motorn under styraxeln.
Filnamn: Sammanställning.ipt, Centrering Drawing.pdf.
- **Kullager** Används för att centrera motorn under styraxeln.
- **Okplåt** Monteras på styraxeln för att kunna fästa bromsoket.
Filnamn: Sammanställning.ipt, Okplat Drawing.pdf.
- **Okdistans** Monteras mellan okplåten och bråmsoket för att få rätt avstånd.
Filnamn: Sammanställning.ipt, Okplat Drawing.pdf.

4 Övrigt

- **Apparatlåda** IP55-klassad, 310x240x110 mm. Köpt från Malmbergs.

- **Bottenplatta till apparatlåda** Designat i CAD och tillverkad med CRFs laserskärare.
Filnamn: `Bottenplatta.step`