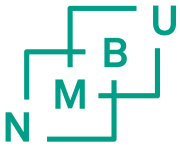
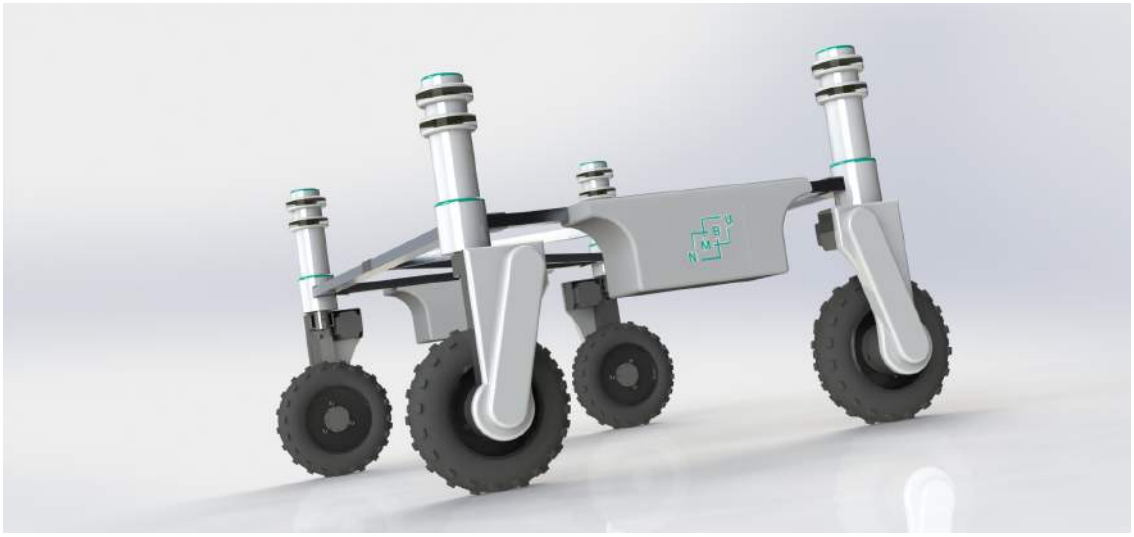

U
B
M
N



Preface

After selecting *robotics* as the subject for my master's thesis, I was a bit worried that I had bitten off more than I could chew. With only one subject related to control and automation, building a 150 kg agricultural robot in just four months seemed like an impossible task. Nevertheless, I was determined to give it a go. So I packed my bags and got on a plane to Brazil together with four other (equally inexperienced) master students.

The three weeks that followed were hectic, and very educational. The guys at *the Federal University of Rio de Janeiro* (UFRJ) provided a quick but thorough crash course in robotics, and taught us all we needed to know to get started designing our robot. The rest of the weeks were spent reading up on motors, transmissions, sensors, automation and control systems, and developing our initial concept.

The work continued after we got back to Norway. Based on the knowledge we acquired in Brazil, the initial concept was refined, and after countless drafts, we agreed upon a final design. The process of finding the right components could then begin. This turned out to be a far more time consuming task than expected. Many hours were spent searching the Internet for suitable parts, and countless e-mails were sent to different manufacturers and suppliers.

After many days (and nights) of testing, troubleshooting, soldering, assembling and reading user manuals, the mobile platform was as good as finished.

I would like to thank the people at the robotic department at UFRJ for taking the time to teach a group of Norwegian students the fundamentals of robotic, and also the CAPES-SIU project for covering travel expenses, enabling us to go to Brazil. I would like to thank the Nordic Association of Agricultural Scientists (NJF) for much needed funding, and Electro Drives AS and ELFA Distrelec for discounted prices on components. I would like to thank Nils Bjugstad, and Geir Terjesen, Tom Ringstad and Petter Heyerdahl for taking their time to discuss ideas and concepts, and the guys at the NMBU workshop for building the robot frame. I would also like to thank my fellow master students: Fredrik Blomberg, Fredrik Meltzer, Marit Svenkerud and Jørgen Torgersen for being such a good team, and of course my adviser Pål J. From for hours of guidance and discussions, and also for letting me build a 150 kg robot! Finally, I would like to thank my friends and family for supporting me through five years of studies, especially my parents who guiding me in the direction of higher education from a very young age.

The NMBU Agricultural Robot project will hopefully live on for many years, recruiting new students and promoting the university. I am proud to have been involved in the start-up phase, of the design choices I have made and of the work I have done, and I am proud to present the following thesis.

Ås
15/5-2014

Lars Grimstad

Powertrain, Steering and Control Components for the NMBU Agricultural Mobile Robotic Platform

Lars Grimstad

Sammendrag

Denne masteroppgaven er delt i to deler. Den første delen av oppgaven tar for seg designet av drivlinjen og svingesystemet til NMBUs mobile landbruksrobot, mens den andre omhandler reguleringen av disse systemene.

Del I

Ulike motorer og transmisjoner presenteres, og krav til fremdrifts og svingesystemer beregnes ut fra et *verst tenkelig* scenario. Med utgangspunkt i beregningene, bestemmes drivlinjedesignet og komponenter velges. Den valgte drivlinjen baserer seg på navreduksjon. Hvert hjul monteres på en *hjulmodul*. Over hjulet monteres en børsteløs DC-motor. Kraftoverføring skjer via en tannreim til et planetgir. Hjulet er montert direkte på utgangen til dette giret. For svingesystemet benyttes børsteløse servomotorer med integrert motorkontroller og planetgir. Robotens kraftdistribusjon diskuteres, og to ulike alternativer presenteres. Det ene alternativet skal benyttes i første fase av utviklingen, det andre er et forslag til fremtidig løsning.

Alle komponenter er i hovedsak valgt på bakgrunn av tilgjengelighet, kvalitet og pris. Det er en viss uoverensstemmelse mellom de ulike drivlinjekomponentenes merketurtall, men ikke mer enn at konseptet kan prøves ut uten problemer. Foreløpige tester viser gode resultater.

Del II

Komponenter og programvare relater til nettverk og regulering av roboten presenteres, og utformingen av reguleringssystemer for fremdrift og svingesystem bestemmes. Systemene utformes på en måte som tilrettelegger for enkel implementering av fremtidige systemer. *ROS* (Robotic Operating System) velges som rammeverk, og CANopen protokoll velges for kommunikasjon mellom enhetene. Prosessen ved å tilpasse motorstyring og fremdriftsmotorer, og prosessen ved å konfigurere servomotorene beskrives. Forsøk gjøres på å benytte *Hall sensorer* som tilbakekobling ved regulering av fremdriftsmotorenes hastighet i lukket sløyfe. Dette gir dårlige resultater, og alternativ tilbakekobling basert på sensorer med høyere oppløsning foreslås.

Abstract

The first part of this thesis addresses the design of the NMBU Mobile Agricultural Robot's drivetrain and steering assembly, while the second part discusses aspects related to controlling these systems.

Part I

First, power requirements are calculated based on the expected *worst case* scenario. Second, the principle design of the systems is determined and parts are selected. For the drivetrain, a hub reduction system is used. The system comprises of four separate *wheel modules*, each with one brushless DC motor installed over the module's wheel. Power is transferred to an in-wheel planetary gearbox via a timing belt. For the steering assembly, brushless integrated servomotors with planetary gearboxes are used. The power distribution system is also discussed, and two designs are proposed, one for future and one for immediate implementation.

All parts are selected based on availability, quality and cost. As a result, the components for the drivetrain are not perfectly matched. However, they allow for testing of the basic principle. Preliminary tests of the powertrain and steering assembly show promising results.

Part II

Different components related to communication and control of the robot are presented, and the principle design of the motor control system is determined with interoperability between current and future systems in mind. *ROS* (Robotic Operating System) is selected to be used as the robotic framework, and the main computer and the various motor controllers are to communicate using the *CANopen* protocol. The process of configuring the propulsion motor controllers to match the propulsion motors is described, as is the process of setting up the servomotors. Problems were faced using the propulsion motors built in *Hall sensors* for feedback in *closed loop speed mode*, as the sensor assembly's resolution proved to be too coarse. Alternative feedback sensors are then proposed.

List of Tables

List of Figures

Abbreviations

4WD Four-Wheel Drive

4WS Four-Wheel Steering

BDC Brushed Direct Current

BFL Bayesian Filtering Library

BLDC Brushless Direct Current

CAN Controller Area Network

CiA CAN in Automation

COM Communication (Port)

CPU Central Processing Unit

CWDC Compound-wound Direct Current

D-sub D-subminiature

DC Direct Current

HTD High Torque Design

ISO International Organization for Standardization

KDL Kinematics and Dynamics Library

MRDS Microsoft Robotic Development Studio

NJF Nordic Association of Agricultural Scientists

NMBU Norwegian University of Life Sciences

OCL Orocos Component Library

OD Object Dictionary

Orocos Open Robot Control Software

PMDC Permanent Magnet Direct Current

ROS Robotic Operating System

RPM Revolutions Per Minute

RTK-GPS Real Time Kinematic Global Positioning System

RTT Real-time Toolkit

SAE Society of Automotive Engineers

SHWDC Shunt-Wound Direct Current

SWDC Series-Wound Direct Current

UFRJ Federal University of Rio de Janeiro

URDF Unified Robot Description Format

Contents

Chapter 1

Introduction

1.1 Motivation

The basic principles of farming have not changed much since man began using horses in the fields. Modern tractors still pull equipment in the same way horses used to. The equipment has simply grown bigger to cover more land in less time. As the machines grow bigger, they also gain weight. This development is unfortunate. Large, heavy fossil fuel powered machines not only waste energy and pollute the environment. They also require large areas for maneuvering, and cause compaction of the soil. The damaged they inflict on the fields leads to lower crop yields and the need for expensive, energy guzzling soil repair processes. As we all know, the world's population is growing like never before. The challenges we face related to the increasing demand for food and energy are no less than enormous. Productivity and efficiency of the world's crops must increase. This is why the current development in agricultural machine design cannot continue! Cultivable land is not in abundance, neither is energy. It is time to take a step back and completely rethink the way we grow our food.

(a) Horse pulling plow, ca. 1939.

(b) Tractor with plow, 2006.

Figure 1.1: The evolution of farm equipment.

Compaction is when water and air is forced out of the soil, making it more dens. The absence of pores restricts root movement, infiltration, drainage and air circulation. This means that roots are not able to take up sufficient amount of nutrition resulting in less plant growth and lower yield. It is estimated that 90% of

the energy consumption in the farm fields is related to repairing the damages caused by heavy machines (according to Bennedsen [5]).

In theory, doubling the weight and doubling the tire width of a tractor or farm machine should leave the ground pressure unchanged. In practice the pressure is rarely spread evenly over the contact surface, and the peak pressure caused by a heavy machine will still cause damage [41]. Some manufacturers equip their machines with one or more extra axles to reduce ground pressure, but this is expensive and comes at the cost of reduced mobility. Another popular solution is to fit tracks, but even though tracks will increase the contact area between the tractor and the ground, they are more expensive and complex, and less efficient than wheels. As 75% of soil damage is caused by the first pass of the machine [41], it is obvious that effort should be put in reducing farm machine weight.

Many of the tasks that have to be performed in a modern farm field are tedious, and require little more from the farmer than driving a tractor back and forth. Such processes could easily be automated, freeing up time for the farmer. This would increase overall productivity, something that is sorely needed in this subsidy dependent part of the Norwegian prime sector.

Another problem with modern farming is the fact that the equipment is still being pulled behind the tractor. The difficult and time consuming turning process of such an assembly requires large headlands. The turning process also makes the headlands subject to an even greater level of soil compaction then the rest of the field.

Although some farm machines have built in equipment, such as combined harvesters, they can only serve one purpose. Many of the machines are also massive. An example is beat harvesters, which are fitted with enormous bunkers to minimize the time spent unloading.

The farm machine of the future must be completely redesigned compared to the technology of today. It should be modular with exchangeable equipment integrated in the machine itself. It should be autonomous and light weight, so that several small vehicles can work in the field at the same time, working continuously 24 hours a day, automatically changing equipment and batteries/refuel when needed.

Our goal is to create such a robot.

Figure 1.2: A small and lightweight tractor, ca. 1992.

1.2 Existing Consepts

There are already a few existing agricultural robot concepts. Some are made to do only one task, while other are modular with tools for multiple applications. For

(a) Mobile robot for weeding [23].

(b) BoniRob [2].

(c) Mobile agricultural robot [37].

(d) Kongskilde Robotti [19].

Figure 1.3: Existing agricultural robots.

reference, a small sample of project will be presented in brief.

Project varies in size from master projects like "Mobile Robot for Weeding" [23] by the Danish students T. E. Madsen and H. L. Jakobsen, to research projects like "Mobile Agricultural Robot" [37] at EESC - USP, Brazil, to commercial projects like Robotti [19] and BoniRob [2].

Robotti is a robot developed by Kongskilde, a Danish manufacturer of agricultural equipment. The robot uses tracks driven by two powerful electric motors and have a maximum operating speed of 10 km/h. It can be equipped with implements for precision seeding, ridging discs and mechanical row crop cleaning units, and is currently used as a base in research projects. Tracks provide good traction, but are inefficient to drive. At 500 kg it is much lighter than a tractor, and thus causes far less damage to the soil.

BoniRob is a four wheel steering, four wheel drive robot developed by German manufacturer of agricultural machinery, Amazonen-Werke. The robot is modular with implements for different applications, and have a top speed of 13 km/h. At approximately 800 kg it too is lighter than a tractor, but still quite heavy compared to similar concepts.

Basic specifications for the four aforementioned robots are listed in Table 1.1. Figure 1.3 show the robots.

1.3 The NMBU Robot Concept

We want to build a small but strong, mobile, autonomous robot capable of performing a wide range of different tasks in the field. The robot will be modular with a wide array of intelligent tools for high efficiency precision farming. The robot

Table 1.1: Specifications of four existing robots.

	Mobile Robot for Weeding [23]	BoniRob [2]	Mobile Agricultural Robot [37]	Kongskilde Robotti [19]
Year	2001	2014	2011	2013
Country	Denmark	Germany	Brazil	Denmark
Application	Weeding	Modular	Data collection	Modular
Project type	Research	Commercial	Research	Commercial
Speed	6.3 km/h	13 km/h	UNK	10 km/h
Weight	312 kg	800-1000 kg	UNK	<500 kg
Length x width [m]	1.00 x 1.00	1.50 x 1.50	2.00 x 1.20	UNK
Ground clerance	50 cm	80 cm	UNK	UNK
Frame material	Aluminium	Steel	Steel	UNK
Drive	4WD	4WD	4WD	Tracks
Propulsion motor	4 x 150 W	UNK	4 x 750 W	2 x 5000 W
Propulsion gear ratio	9.3	UNK	75	UNK
Steering system	4WS Ackerman	4WS Ackerman	2WS Ackerman	Skid steering
Steering motor	UNK	UNK	4 x 150 W	n/a
Steering gear ratio	UNK	UNK	230	n/a
Suspension	Passive	Passive	Passive	Passive
Battery technology	Gel	Lead	UNK	Lead
Battery energy	1.68 kWh	UNK	7.8 kWh	5.28 kWh
Battery weight	60 kg	UNK	UNK	UNK

frame will contain most of the sensors needed for autonomous operation, reducing the need for tool-mounted sensors to a minimum. Linear actuators installed on the robot frame will adjust the height of the tool during operation, and any active tools will draw power from the robot's energy storage.

We want the robot to be completely autonomous, automatically changing equipment and topping up on energy when needed. It should be capable of working 24 hours a day, 7 days a week with minimal supervision. The robot should also be inexpensive to acquire, making it profitable for the farmer to swap his or hers tractors for robots.

Our robot will have four-wheel drive ensuring traction in any conditions. To maximize mobility and minimize the need for headlands, it will also have four wheel steering. The robot shall be capable of working speeds up to 3.5 km/h ($\sim$ 1 m/s) which will enable each robot to cover large areas every day.

This is a project that will take many years to complete. We are only at the very

beginning, and in this first phase of the project we will concentrate on building a mobile platform that easily can be automated, and is ready for implementation of a modular equipment system and navigation systems.

The target specifications of the mobile platform are summarized in Table 1.2.

Table 1.2: Mobile platform specifications.

Drive	4WD
Steering	4WS
Top speed	3.5 km/h
Max acceleration time, 0-3.5 m/s	2 s
Width	1.7 m
Length	1.1 m
Mass, platform	150 kg
Mass, full payload	300 kg

1.4 Scope of this Thesis

The purpose of this thesis is to determine the design and select the components of the robots powetrain and steering assembly, and to determine the design of a safe and reliable power distribution system. It is also to select the best suited software, sensors and communication protocols to enable robust, precise and efficient control of these systems.

The process of assembling and setting up the various systems will involve hours of testing, building and programming, all of which will be described in this thesis.

It is important that the choices made at this early stage do not complicate the further development of the robot. Components, control systems and software must be selected in such a way that it can be used with future sensors, navigation systems, intelligent farm equipment, etc. This is especially important when selecting the software framework, as this must be compatible with all parts of the robot.

The mobile platform is to be designed and built in just four months. Time is therefore a factor that must be considered when choosing the components. Finding solutions that allow fast installation and easy control is essential if the platform is to be completed in time.

To minimize time required for repair and maintenance work, the design should allow easy access to all components. It is also important to build the robot in such a way that modifications can be done without having to completely rebuild larger parts of the robot.

Although effort will be put in keeping the weight at a minimum, this cannot come at the cost of low quality components. We are building a field robot, and it is important that it is capable of tackling the rough and harsh conditions associated with field operation.

Part I

Powertrain and Steering System

Chapter 2

Motors and Transmissions

According to Wong [43] the two limiting factors to a vehicles performance is the traction between the wheel and the surface, and the maximum torque provided by the on-board power plant and transmission. The smaller of these factors will determine the potential of the vehicle. This thesis will concentrate on finding the best suited powertrain for the NMBU Robot concept. For discussions regarding traction see Meltzers [24].

An ideal power plant will generate large amounts of torque at low speeds where acceleration or grade climbing capability is needed, and then maintain a constant power output over a wide speed range as shown Figure 2.1. Different power plants will be further discussed in the following sections of this chapter.

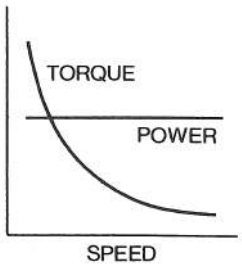


Figure 2.1: Ideal vehicle performance characteristics. Figure courtesy of Wong [43].

Merriam-Websters on-line dictionary defines a motor as "a machine that produces motion or power for doing work" and a transmission as "an assembly of parts including the speed-changing gears and the propeller shaft by which the power is transmitted from an engine to a live axle; also: the speed-changing gears in such an assembly" [40]. From such a definition, it can seem like the word "transmissions" only should be used for mechanical gearboxes, but it should also cover other principles of power transferring devices, like the hydrostatic transmission.

Transmissions are used to exchange speed for torque, or torque for speed. Combustion engines and electric motors generally operate at a higher speed than what is desired for most application. A transmission is then used to transmit the power from the high speed, low torque motor shaft at the input, to a low speed high torque output shaft.

In this thesis the gear ratio is defined as the ratio of the angular velocity of the input gear to the angular velocity of the output gear. A combustion engine typically

needs a gearbox with multiple available gear ratios to get acceptable efficiency, while a DC motor generally only needs one gear ratio.

2.1 Internal Combustion Engine

The internal combustion engine emerged in the 19th century as a substitute for the steam engine. Different engines running on different fuels were developed, and internal combustion gradually replaced steam engines in many industrial and transport applications.

In an internal combustion engine, fuel is burned inside the engine and the energy of the fuel is converted into mechanical energy. The combustion can be intermittent (like in a piston engine), or continuous (like in a jet engine). For land based, agricultural field machines, piston engines running on diesel are most common, both for small and large machines. Figure 2.3 show a compact, diesel powered tractor.

Piston engines vary in size from a fraction of a kilowatt in model vehicles to 50 000 - 60 000 kilowatts in ship engines, and run on energy-dense liquid fuels. The high energy density of the fuel is an advantage in mobile applications in that long operating time between refueling combined with low weight and volume of the energy storage unit can be achieved. In addition, the engines have high power to weight ratios which make them ideal for flying robots or large robots [27].

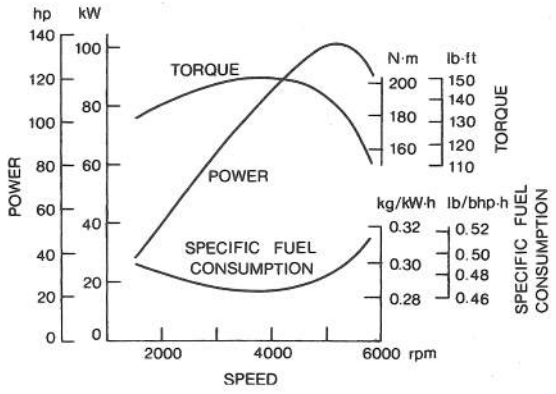
In robotics, the internal combustion engine can be used directly for propulsion, or indirectly by driving an electric generator like a dynamo or alternator, or by driving a hydraulic pump.

As Figure 2.2 show, the performance curves of piston engines are far from the ideal presented by Figure 2.1. An engine of this type requires a multi-step gearbox or a continuously variable transmission to "chase" the best efficiency point on the curve if the engine is to be used directly for propulsion. For this reason, an in-wheel solution is virtually impossible.

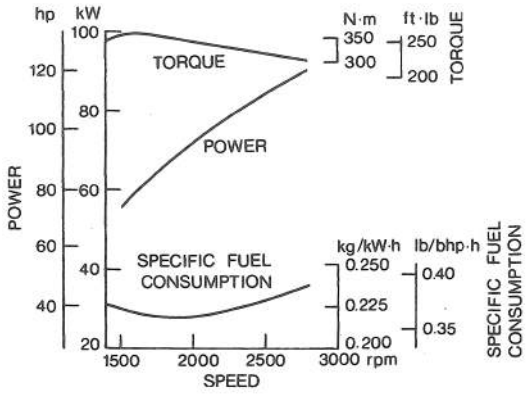
There are many other disadvantages with combustion engines. The efficiency of a traditional piston engine as used in modern cars is very poor. The excess heat generated calls for the need of a cooling system, further increasing the mass and complexity of the system. The combustion process also produces toxic gases and particles, and common fuels like gasoline are highly flammable and may presents an explosion hazard. In addition, the majority of common fuels are made from non-renewable sources.

The concern for future oil supply and the growing awareness for global climate changes has led to an increasing interest for alternative fuels for the combustion engine. Biofuels such as ethanol, methanol and plant oil produced from biomass are getting more and more attention. Although diesel engines can be made to run on pure plant oil, and spark ignition engines can be made to run on pure ethanol, biofuels are often blended with traditional fossil fuels so it can be used in unaltered fossil fuel engines or to improve fuel properties.

For more on combustion, see the following references: [21] [7] [27] [43].



(a) Gasoline engine performance curve.



(b) Diesel engine performance curve.

Figure 2.2: Performance curves of piston engines. Figures courtesy of Wong [43].

2.2 Electric Motor

Electric motors convert electric energy into mechanical energy, and are used in a vast multitude of applications. Motor ratings range from less than one watt (see *Maxon Motor* [25]) to more than 100 megawatts (see *Siemens* [36]), and the motors can be found in watches, toys, cars, trains, ships, pumped storage plant and more. Electric motors are also the preferred and most common form of actuators in robotics [27].

Many different principles exist for rotary and linear motion with alternating current or direct current. For their ability to run on batteries, electric DC motors are often used in mobile applications.

Electric motors take advantage of the fact that like magnetic poles repel each other whilst different magnetic poles attract. Current running through a coil of wire will generate an electromagnetic field aligned with the center of the coil. If the current is reversed, the field is reversed.

A rotary DC motor consists of a stator (the stationary part of the motor) and a rotor (the rotating part of the motor). At least one of the two parts contains electromagnets; the other may have electromagnets or permanent magnets.

A basic DC motor is shown in Figure 2.4. When electric current is passed through the electromagnet (rotor and/or stator depending on motor type), an electric field is created. This generates a magnetic force between the rotor and the stator, and the rotor rotates so that the south pole of the rotor aligns with the north pole

Figure 2.3: Compact tractor with diesel engine. Picture courtesy of Shibaura [35].

of the stator. As this is about to happen, the current in the electromagnet is reversed, flipping its poles. The poles that were attracting each other are now repelling each other, and the rotor continues to rotate (if both rotor and stator contain electromagnets, the current is reversed in only one of the two).

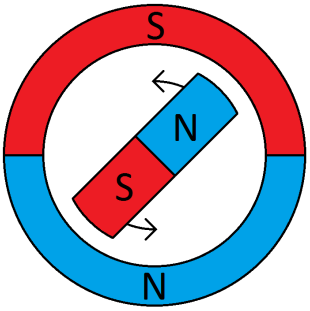


Figure 2.4: Basic principle of a DC motor.

To flip the field of the electromagnet at the right time, commutation is needed. That is, we need to know which windings to energize at any given time. In traditional DC motors, commutation is done mechanically. With increasing availability of inexpensive power semiconductors, electronic switching of the current in the motor windings is getting more common.

2.2.1 Brushed DC Motors (BDC)

The brushed DC motor, or simply the "DC motor", has one or more electromagnetic windings in the rotor and permanent magnets or windings in the stator. Current is supplied to the rotor windings through spring loaded brushes which slides on a commutator ring fixed to the motor shaft. The brushes are usually made from carbon. The commutator ring has separated conducting segments and rotates with the shaft. Each electromagnet in the rotor is connected to two conductive segments on the commutator ring. As the shaft rotates, the commutator ring rotates, energizing the rotor windings in the appropriate sequence.

Brushed DC motors are inexpensive to make, come in a variety of sizes and shapes, and most are easy to control. The direction of rotation is reversed by changing the polarity of the two motor wires.

During operation, the brushes are worn, and the electrical connection between the brushes and the commutator deteriorate. The brushed DC motor therefore

requires periodic maintenance. The mechanical connection between the brushes and commutator generates noise and sparks, and also limits the maximum speed of the motor.

Because of the relatively low torque generated, the brushed DC motor is often used in combination with a transmission. This combination is generally referred to as "geared DC motor".

The different types of BDC motors are categorized by how the static magnetic field is generated. The different types are briefly presented in the following subsections. For more information on the different type of DC motors, see Condit [8]. For more information on the DC motor in general, see the following references: [8], [20], [27]

Permanent Magnet (PMDC)

Permanent magnets are generally more cost efficient for fractional horsepower applications. The torque generated is inversely proportional to the speed and the speed-torque characteristics can be approximated by a straight line from stall torque (speed is zero) to no load speed (torque is zero). They also respond quickly to change in voltage due to the constant magnetic field. A problem with the PMDC motors is that the magnets lose their magnetic properties over time. To avoid demagnetization, peak starting torque is commonly limited to 150% of rated [4]. Another solution is to have some windings built into the stator. Figure 2.5 show the PMDC motor schematically and also the motor performance curve.

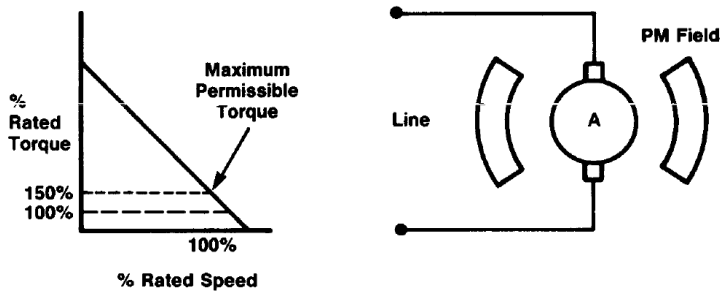


Figure 2.5: Sharacteristics and schematics of a DC motor with permanent magnets. Figure courtesy of *Rockwell Automation* [4].

Shunt-Wound (SHWDC)

Shunt-wound motors have stator windings separately controlled form the rotor windings (Figure 2.6). The fact that the current in the stator and rotor windings are independent, gives the SHWDC motor excellent speed control. Shunt-motors are typically used in applications where five or more horsepower are required [**BDC˙fundamentals**].

Series-Wound (SWDC)

Series-wound motors have stator windings in series with the rotor windings (Figure 2.7). This makes them ideal for high torque applications, in that the current in both stator and rotor windings increase under load. For the same reason, speed control is

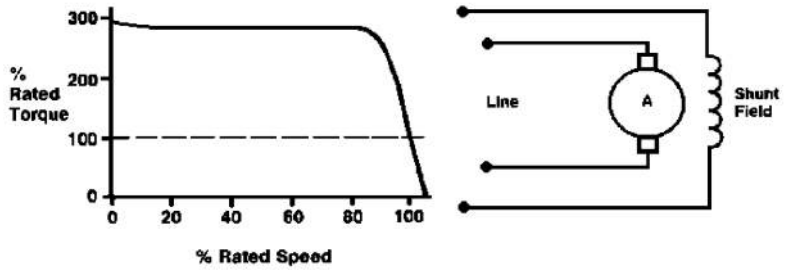


Figure 2.6: Shunt-wound DC motor with characteristics. Figure courtesy of *Rockwell Automation* [4].

more difficult than with PMDC or SHWDC motors. The efficiency is also relatively low, but the motor is still commonly used in electric industrial trucks because of the simple, low cost design [43].

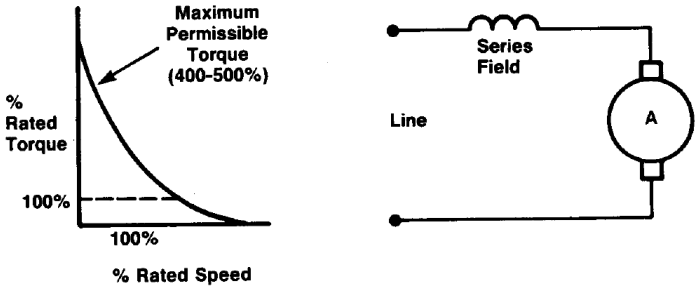


Figure 2.7: Series-wound DC motor with characteristics. Figure courtesy of *Rockwell Automation* [4].

Compound-Wound (CWDC)

Compound-wound motors have a combination between series and parallel stator and rotor windings (Figure 2.8). They offer higher torque then the SHWDC motor and better speed control than the SWDC motor.

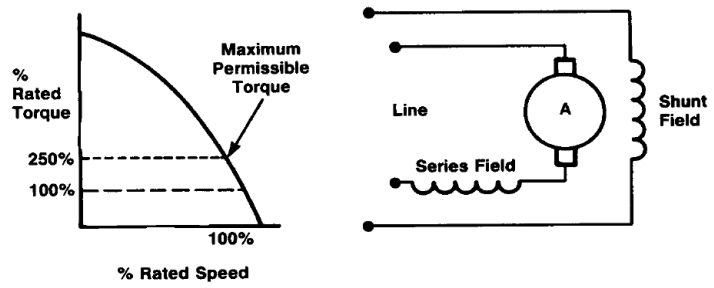


Figure 2.8: Compound-wound DC motor with characteristics. Figure courtesy of *Rockwell Automation* [4].

2.2.2 Brushless DC Motors (BLDC)

Brushless DC motors are gaining popularity and are displacing the BDC motor in a range of applications. They are used in anything from electric cars to medical equipment to automation.

The BLDC motor is a type of synchronous motor. This means that the stator field and rotor field rotate at the same frequency. As the name imply, these motors do not have brushes. Instead they are electrically commutated. Windings in the stator are continuously turned off and on by transistors to keep the permanent magnet fitted rotor rotating. Single-phase, 2-phase and 3-phase motors exist. Single-phase motors are typpicaly used for low power applications, while 2-phase motors are widely used in cooling fans for computer components, such as CPUs or power supplies. These two types of BLDC motors are not well suited for drive or steering applications. This thesis will therefor only address the 3-phase motor which is far better suited for such purposes.

To make the stator, slotted steel laminations are stacked, and coils are wound in the axial slots. The three phases are connected in star fashion, and each of the phases are formed by an even number of interconnected coils. Figure 2.10 shows a BLDC stator.

The rotor field is generated by permanent magnets, and the rotor generally has two to eight poles. Traditionally ferrite magnets are used, but as technology progress, rare earth alloy magnets are gaining popularity. Ferrite magnets are less expensive, but the rare earth alloy magnets has higher magnetic density per volume, giving a motor using rare earth alloy magnets higher torque capacity than a same size motor using ferrite magnets.

For the motor shaft to rotate, the stator coils have to be energized in the right sequence. To know which winding to energize, the position of the rotor must be known. This is done by Hall effect sensors embedded in the stator. Whenever a magnetic pole passes the Hall sensor, it gives a high or low signal indicating N or S poles. By using three Hall sensors at 60° or 120° phase shift to each other, the exact sequence of commutation is determined.

Ensuring correct alignment of the Hall sensors is difficult. To simplify the process of installing the sensors, some motors have dedicated Hall effect sensor magnets on the rotor in addition to the main rotor magnets.

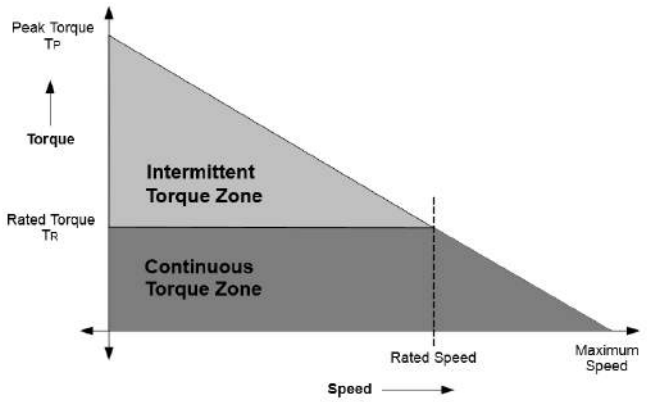


Figure 2.9: Torque-speed characteristics. Figure courtesy of *Microchip* [44].

The BLDC motor can generate its rated torque across a wide speed range, but

as Figure 2.9 show, it can generate far more than this for brief periods of time at low speeds. The linear characteristics makes speed control predictable, but because of the electric commutation, the BLDC motor requires a more complex controller than the BDC motor. The permanent magnets in the rotor make the BLDC motors more expensive to manufacture than BDC motors. The BLDC motor also needs more wiring between the motor and controller to be able to work. While the low resolution feedback from the Hall sensors may be sufficient for speed control in many applications, the BDC motor will always require an encoder for feedback, making the wiring (almost) equally complex for the two motors.

Although it is more expensive to manufacture and more complicated to control, the brushless DC motor has many advantages compared to its brushed counterpart. The absence of brushes reduces friction loss, makes the motor run quieter, reduces electric noise, and makes higher speeds possible. The only parts prone to mechanical wear are the bearings. For this reason the motor require very little maintenance, making it ideal for limited access applications. The BLDC motor delivers more torque than a same size BDC motor, the motor inertia is lower, and the overall efficiency better. The rated torque can also be delivered over a larger speed range than for a similar BDC motor. Because the windings are in the stator, the generated heat is more easily dissipated, reducing the need for cooling.

For more information on BLDC motors see the following references: [44] [27] [34].

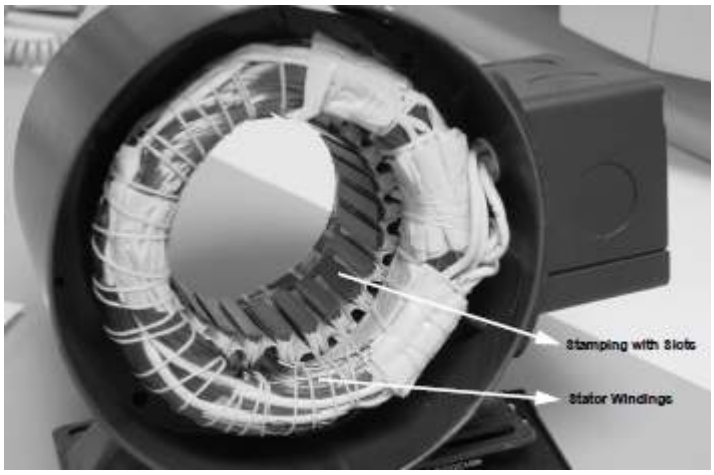


Figure 2.10: BLDC stator. Figure courtesy of *Microchip* [44].

2.3 Gears

Gears are the most common form of power transmission [33]. They are used to transmit power in form of rotational motion from one shaft to another, change rotational direction, convert between angular and linear motion or increase torque at the cost of rotational speed (or vice versa). Power can be transmitted between parallel, intersecting or skew shafts. High efficiency and quiet operation can be achieved. This requires high precision in the shape of the teeth and the distance between mating gears.

Any two gears with equal tooth size can be used together. However, a large ratio between a single pair of mating gear causes problems with tooth wear. To obtain

large ratios, multiple stages are therefor used. The gears must be shielded from dirt, and should be enclosed in a sealed case with the required lubricating oil or grease.

Many different types of gears are available. The ones relevant for the NMBU Robot project are explained in more detail in the following subsections.

For more information on gears, see the following references: [33], [39], [13].

Spur Gear

Spur gears are cylindrical gears that transmit power between parallel shafts (Figure 2.11). They have straight teeth parallel to the axis of rotation. No more than two sets of teeth are in mesh at one time. The simple design makes them inexpensive to manufacture, but they will generate some noise at higher speeds.

Figure 2.11: Spur gear. Figure courtesy of *Martin Sprocket* [39].

Helical Gear

Helical gears (Figure 2.12) are similar to spur gear, but their teeth are set at an angle, making the teeth longer for gears of the same width. This results in smoother and quieter operation at the cost of slightly less efficiency. Helical gears are well suited for high speeds and loads, making them ideal for use in automotive gearboxes. However, the angled teeth will generate axial thrust forces on the shaft.

Figure 2.12: Helical gear. Figure courtesy of *Martin Sprocket* [39].

Herringbone Gear

Herringbone gears, also called double helical gears (Figure 2.13), are gears with opposite helical teeth. The opposing teeth eliminate the axial thrust forces on the shaft that are present in single helical gears. Herringbone gears are suited for heavy loads and medium to high speeds. Due to difficulties of manufacturing, herringbone gears are more expensive then single helical gears. This limits their applications to high capacity reduction gears like that of cement mills or crushers [13].

Figure 2.13: Herringbone gear. Figure courtesy of *Martin Sprocket* [39].

Worm Gear Pair

A worm gear pair (Figure 2.14) consists of a screw-like worm at the input and a helical gear called the worm gear at the output. The worm and the worm gear are mounted at non-intersecting shafts at 90°. The continuous sliding action between the worm and worm gear result in smooth and quiet operation, but also causes considerable friction heat. The efficiency is therefore low, generally $40\% - 90\%$ [13], and good lubrication is important for sufficient heat dissipation and acceptable efficiency.

Because of the compact design, the worm gear is used for high transmission ratios, typically 5:1 to 100s:1 [33]. It is widely used in anything from material handling to automotive applications [13]. The large ratio gears are self-locking. This means that the output cannot be rotated if the input power is turned off.

Figure 2.14: Worm gear. Figure courtesy of *Martin Sprocket* [39].

Straight Teeth Bevel Gear

Straight teeth bevel gears (Figure 2.15) transmit power between intersecting shafts. They provide moderate torque and are capable of medium loads.

Figure 2.15: Straight teeth bevel gear. Figure courtesy of *Martin Sprocket* [39].

Spiral Bevel Gear

The difference between spiral bevel gears (Figure 2.16) and straight teeth bevel gear is, as the name imply, the shape of the teeth. Spiral bevel gears have curved oblique teeth that engage gradually, resulting in smoother operation. At least two sets of teeth are in contact at the same time. They have high load capability and can turn up to eight times faster than the straight teeth equivalent [33].

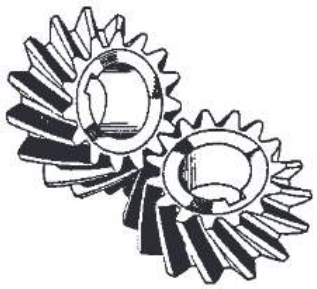


Figure 2.16: Spiral bevel gear. Figure courtesy of *Martin Sprocket* [39].

Planetary Gear

As shown in Figure 2.17, planetary gears consist of three coaxial elements. The first element is the sun gear which is in the center. Surrounding (and in mesh with) the sun gear are the planet or idler gears. The planet gears are mounted on a planet carrier, also called a spider. This ensemble is the second element. The third element is the ring gear, which is a gear with internal teeth surrounding (and in mesh with) the planet gears.

The input shaft is connected to one of the three elements and the output to another. If the last element is free to rotate, the output shaft will not rotate. If two of the elements are fixed together, the result is direct drive and the gear assembly will not work as a transmission.

To work as a transmission, the third element must be fixed so it cannot rotate. This gives the gear six different possible configurations. Two for reducing speed and

two for increasing speed with forward direction of rotation, and one for reducing speed and one for increasing speed with reversed direction of rotation. In real life one, or possibly two, of these configurations can be used for a given gearbox. For some gearboxes neutral and direct drive are also included.

The planetary gearbox is one of the most efficient and compact gearbox designs, but they are also expensive compared to other gearboxes [33].

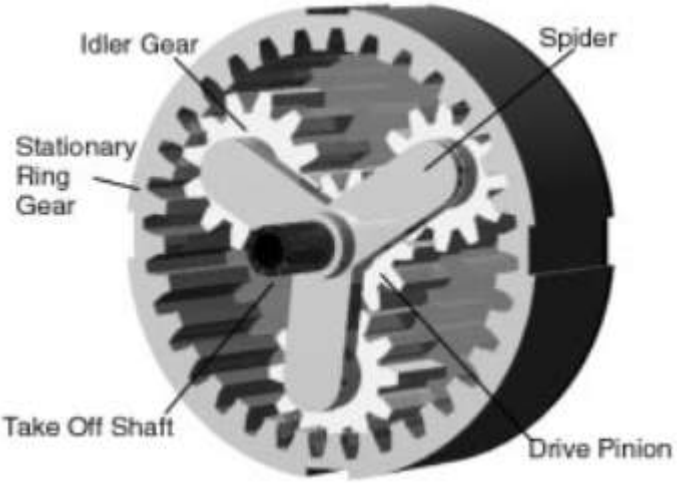


Figure 2.17: Planetary gear. Figure courtesy of *Martin Sprocket* [39].

2.4 Hydrostatic Transmission

Hydraulic motors convert hydraulic energy into mechanical energy and are used in a wide range of application from garden mowers to industrial sized agricultural machines.

The hydraulic motor is a part of a hydrostatic power transmission, and cannot function as an isolated unit like an electric motor. The main purpose of a hydraulic system is to transfer energy from a prime mover, like a combustion engine or an electric motor, to an actuator producing useful work.

A basic hydrostatic transmission is a complete hydraulic system, and incorporates a pump, a hydraulic motor, oil filters, valves etc. To make the transmission versatile, either or both the pump and motor is made variable displacement, resulting in a continuously variable speed drive. The pump is mechanically powered by the prime mover and is generating flow. The system responds by setting up a high pressure between the outlet side of the pump and the motor. As the fluid passes through the motor, hydraulic energy is converted to mechanical energy, and the fluid exits the pump at a lower pressure. The fluid is then filtered and reused.

Like any DC motor in theory can work as a generator, any hydraulic pump may in theory be used as a motor, and vice versa. This is useful if the hydraulic transmission is used in the propulsion system of a vehicle. When the vehicle, say an agricultural farm machine, is going uphill, the pump and motor will work as described in the previous paragraph. When the vehicle is going downhill, the vehicle will try to accelerate due to gravity. The energy of the descending vehicle will be forced into the motor, and the motor will start working as a pump. The motor (now acting as

a pump) will try to over-speed the prime mover connected to the pump (now acting as a motor). This causes a breaking effect. If the prime mover is an electric motor, absorbed energy can be fed to the battery.

Hydraulic motors are small compared to their power output, and the hydraulic fluid which supplies the motor with energy, enters and exits the motor in flexible hoses. This makes the motor suited for in-wheel applications as shown in 2.18. The hydrostatic transmission is fast responding and can maintain constant speed under varying loads. It also offers infinitely speed control from zero to maximum in both forward and backward direction.

The hydraulic transmission also has its drawbacks. While the efficiency of sliding gear transmissions is about 95%, the efficiency of hydrostatic transmissions is about 75%-80% [10]. The hydrostatic transmission is also more expensive than mechanical alternatives.

For more information on hydrostatic transmissions, see the following references: [17] [10].

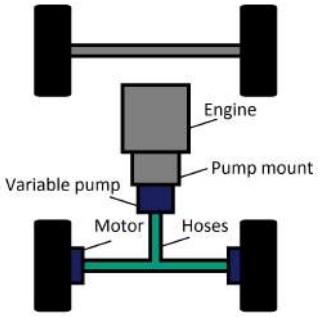


Figure 2.18: Example of hydrostatic transmission in a two wheel drive vehicle.

2.5 Belt

Belts are available in four major variations, namely flat belts, O-ring belts, V-belts and timing belts. Of these, only timing belts are synchronous, meaning there is no slip between the sprockets and the belt.

Flat belts are an old design with moderate efficiency, and have limited use today. O-ring belts are very cheap, but these too suffer from moderate efficiency, which limits their application to low power devices and toys. Flat, O-ring and V-belts are shown in Figure 2.19.

For more information on belts, see the following references: [33].

2.5.1 V-Belt

The cross section of a V-belt looks like a V with a flat bottom (hence the name). The belt jams in a matching groove in the pulley. Like flat and O-ring belts, V-belts rely on friction, but for the same belt tension the friction force is greater for the V-belt. The reaction forces on the belt from the angled groove surfaces are larger than those from a flat pulley. This is because only the radial components of the angled reaction forces counteract the belt tension. As friction is proportional to the force normal to the mating surface, the friction force must be greater for V-belts.

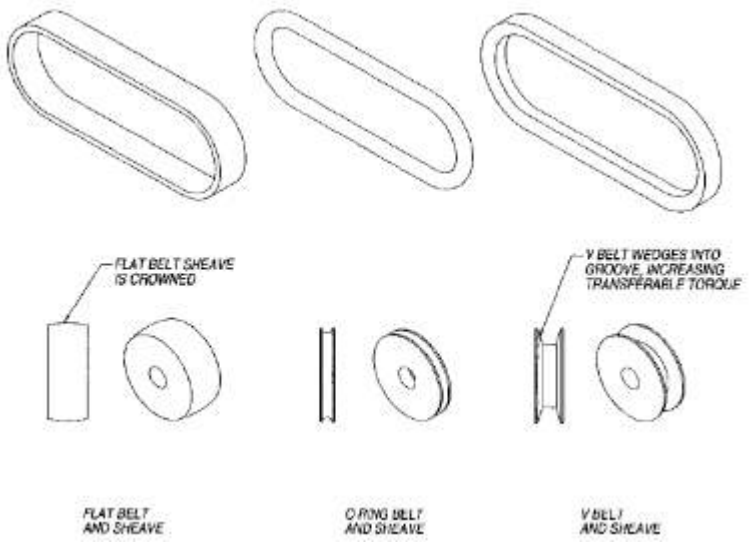


Figure 2.19: Flat, O-belt and V-belt profiles and pulleys. Figure courtesy of Paul E. Sandin [33].

Thus for the same belt tension, V-belt are able to transmit more torque than flat or O-ring belts.

V-belts are quiet, efficient, and they allow some misalignment of the pulleys. They are well suited for power levels from fractional to tens of horsepower [33]. The main drawback of such belts is the tendency for the belt to slip over time. If precise control of the orientation of the output shaft is required, a feedback device must be installed on the driven pulley.

Variable speed drive is possible by variable diameter sheaves. However, in a mobile robot, this may cause control problems in some cases. The reason for this is that the computer does not have direct control over speed of the output shaft [33].

2.5.2 Timing Belt

The timing belts are toothed to eliminate slip, making them synchronous (or positive) drive. Flexible teeth ensure that the load is spread out to all the teeth in contact with the pulley. There are different types of belts depending on the shape of the teeth. The most common is the trapezoidal shaped tooth [33] as shown in Figure 2.20, but this can have some issues with tooth deformation increasing wear. For this reason many other shapes have been developed. The best being the curved tooth profile [33] also known by the trade name HTD (High Torque Design).

Timing belts can transmit up to 250 horsepower [33], with low speed and high torque. They are efficient and can be used in wet conditions, but are slightly more expensive than chains.

2.6 Roller Chain

Roller chains consist of steel rollers which turns on pins held together by links as shown in Figure 2.21. They are strong, and transfer power efficiently. The roller chain is also very robust, in that it can handle some misalignment between

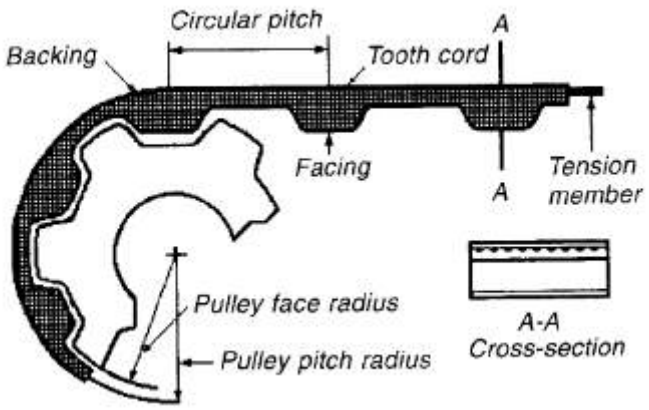


Figure 2.20: Trapezoidal tooth timing belt. Figure courtesy of Paul E. Sandin [33].

driver and driven sprockets, and in many applications do not require precise pretensioning. Reduction of up to 6:1 is possible [33], making such chains well suited for simple but efficient high reduction systems. The most common size chain is #40 (Distance between rollers is .4") [33]. At 300 rpm, this size chain can transfer up to 2 horsepower without special lubrication. The smaller #25 can transfer up to 5 horsepower at 3000 rpm with adequate forced lubrication and sufficiently large pulleys [33].

There are also some drawbacks with the roller chain. It has low toleration for sand or abrasive environments, and it can be quite noisy. While toothed belts fit tightly on the sprocket, the rollers on a roller chain allow some relative motion between the chain and the sprocket. For this reason the roller chain is usually not referred to as synchronous device [33].

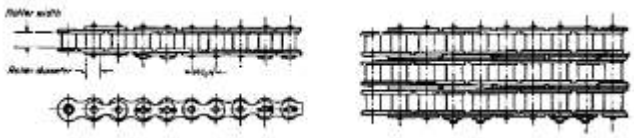


Figure 2.21: Standard roller chain. Figure courtesy of Paul E. Sandin [33].

Chapter 3

Power Requirements

3.1 Propulsion Power Requirements

3.1.1 Gradient Resistance

When driving on a slope, the weight of the vehicle will have a component parallel to the surface (Figure 3.1). This force will try to pull the vehicle down the slope. The magnitude of the force is sine of the slope angle multiplied with the weight of the vehicle as described by Equation 3.1.

$$F_g = mg \sin \theta \tag{3.1}$$

Where:

F_g = gradiant resistance.
m = mass of the vehicle
g = gravitational acceleration
θ = slope angle

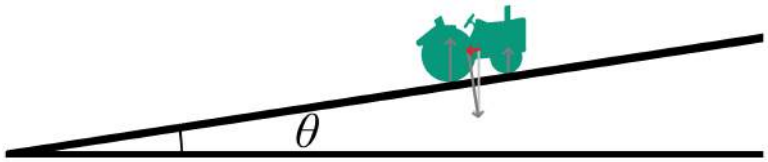


Figure 3.1: Tractor on slope.

3.1.2 Rolling Resistance

The tire and the ground is deformed when a vehicle (say an agricultural robot with four wheels) move through the terrain. This deformation consumes energy, and not all the energy is recovered when the pressure is released. The energy consumption results in a force resisting the movement. This force is called rolling resistance, and it is expressed as the normal force multiplied with a coefficient. This coefficient is dependent on the type of tire and the ground surface in question.

$$F_r = C_{rr} N \tag{3.2}$$

Where:

F_r = rolling resistance force.
C_{rr} = rolling resistance coefficient.
N = normal force. The force acting on the wheel perpendicular to the ground.

When moving on a slope, the normal force is reduced by cosine of the slope angle. The gradient also causes additional shear stress in the tire, increasing the rolling resistance. This roughly cancels out the effect of reduced normal force. The reduction in the normal force should therefor be ignored.

3.1.3 Drag

When air flows over the body of a vehicle, friction forces are generated. This is called air resistance or drag. The magnitude of the forces increases with the square of the relative velocity between the air and the vehicle. Therefore, for a vehicle moving at high speeds, say a car, this friction can have a great negative impact on the overall efficiency.

In addition to vehicle speed, any strong winds should also be taken into account, especially for slow moving vehicles where wind speeds may be many times greater than the speed of the vehicle relative to the surface.

The shape of the vehicle will of course affect the amount of drag it experiences. The shape of a sports car is much better suited for cutting through the air than a big box shaped motorhome. The ability for a geometric shape to cut through the air can be expressed by a drag coefficient. To determine the drag coefficient of a geometric shape, like the shape of a specific car model, it can be tested in a wind tunnel.

The size of the vehicle is naturally also of importance. The orthographic projection of the vehicle seen from the front, that is the parallel projection of the vehicle on a plane orthogonal to the projections lines, is used in drag calculations.

The final factor that must be accounted for in drag calculations is the density of the air (or whichever other fluid) the vehicle is moving through.

For a vehicle moving against the wind, the drag force is calculated by Equation 3.3.

$$F_D = \frac{1}{2} C_D A \rho (v + v_0)^2 \tag{3.3}$$

Where:

F_D = drag force.
C_D = drag coefficient.
A = orthographic projection of the vehicle
ρ = density of the fluid
v = speed of the vehicle relative to the surface on which it travels
v_0 = wind speed relative to the surface on which the vehicle travels

3.1.4 Acceleration

The torque needed to accelerate a motor is equal to the desired angular acceleration multiplied with the mass moment of inertia of the motor. If the motor is connected to

a powertrain in a vehicle, the equivalent moment of inertia of each of the components in the powertrain must be calculated with respect to the motor shaft.

The mass of the vehicle is off course also of importance, and must be accounted for. This can be done by first dividing the mass of the vehicle on the number of drive wheels, then multiplying it with the square of the wheel radius. The result is a mass moment of inertia representing the mass of the vehicle with respect to the wheel shaft. The equivalent moment of inertia is then calculated with respect to the motor shaft and added together with those of the other components.

Equation 3.4 describe how to calculate acceleration torque for the example drivetrain in Figure 3.2.

This is a simplification, but the result should yield sufficiently accurate results.

$$M_A = \alpha_A \left(I_A + \frac{I_B}{(i_{A\to B})^2 \eta} + \frac{I_C + \frac{m}{n_w} r_w^2}{(i_{A\to C})^2 \eta^2} \right) \tag{3.4}$$

Where:

M_A = shaft A acceleration torque.
α_A = shaft A angular acceleration.
I = moment of inertia of each shaft.
i = gear ratio between shafts
η = efficiency of each power transmission stage
m = mass of the vehicle
r_w = radius of the drive wheels
n_w = number of drive wheels.

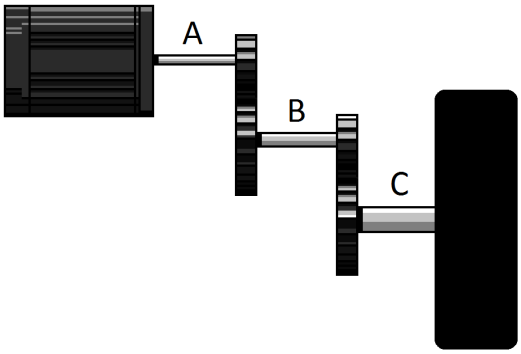


Figure 3.2: Example of powertrain.

3.1.5 NMBU Robot Power Requirements

It is difficult to determine the exact power requirements of our agricultural robot. The terrain and the properties of the soil varies from filed to field, while the requirements for pulling capability and maximum payload depends on the tool and the task the robot is set to do.

The torque exerted on the wheel by the motor must be sufficient to overcome rolling, gradient and air resistance, whilst accelerating the robot from standstill to working speed in an acceptable short amount of time.

Minimum Power Requirements

Norwegian farm fields can be quite uneven and often contains steep slopes. A field robot should be able to move around everywhere in the field at any time, and thus have to be strong enough to climb steep slopes even when it is carrying maximum payload. Our robot should be able to handle a 10° incline. For a 300 kg robot, Equation 3.1 yields 511 N of gradient resistance.

According to Terjesen [38], the rolling resistance coefficient can be set to 0.05 for rubber tires on dirt. For a 300 kg robot, Equation 3.2 yields 147 N of rolling resistance.

The working speed of our robot is slow, and on calm day drag is negligible. However, the robot should be able to operate on windy days with wind speeds of up to 15 m/s.

It is difficult to estimate the drag coefficient and the size of various orthographic projection of our robot before the robot has been designed. For these calculations the drag coefficient is set to 0.50 which is the same as for an off-road vehicle [38]. The orthographic projection is set to 0.80 m^2. This is based on the rough estimated shape shown in Figure 3.3. The density of air is set to 1.293 kg/m^3 (273 K) [38].

If the robot is moving at 3.5 km/h against the wind, Equation 3.3 yields 66 N of air resistance.

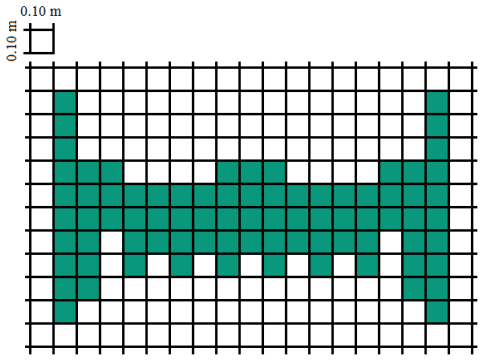


Figure 3.3: Estimated orthographic projection, front view.

If the robot is moving on dirt at constant speed (3.5 km/h) with no slip and 15 m/s headwind up a 10° slope, the combined resistance to movement is $511\,N + 147\,N + 66\,N = 724\,N$. As the robot is moving at 3.5 km/h, the power needed to overcome this resistance is $724\,N \cdot \frac{3.5\,km/h}{3.6\,(km/h)/(m/s)} = 704\,W$.

The terrain in which the robot will be operating can be slippery and rough, and it is not unlikely for the robot to lose traction on at least one of the wheels during normal operation. If the robot only have traction on three of the four drive wheels, the power transmitted by each wheel will have to be $\frac{704\,W}{3} = 235\,W$ at a speed of 3.5 km/h.

The torque exerted by the motor(s) on the wheels must be sufficient to overcome the torque generated by the friction forces whilst accelerating the robot at a sufficient rate. It is important that motors, transmissions and wheel sizes are selected with this in mind. Calculations should be done based on the specific drivetrain.

3.2 Steering Power Requirements

Calculating required steering torque for various conditions is no easy task. There are many factors that influence steering power requirements (See Figure 3.4). Liljedahl et al. [22] lists the following parameters:

- "Tire loading
- Road surface and soil conditions
- Tire inflation pressure
- Tire sizes and tread patterns
- Kingpin inclination
- Caster angle
- Camber angle
- Kingpin offset or scrub radius
- Toe-in and toe-out
- Tread setting
- Travel speeds
- Steering rates
- System efficiency
- Front-end type (tricycle, single wheel, standard)
- Tractive and braking forces
- Chassis type"

For off-road vehicles the steering motors must not only generate sufficient torque to overcome friction between the tire and the surface. If a wheel digs into the surface, the motor also have to displace dirt and mud as the steering module turns.

Although there are many factors to consider when calculating steering torque Liljedahl et al. [22] suggest that the heaviest steering loads for tractors with Ackerman-type steering occur when the tractor is stationary on dry, clean concrete, and that this should be used when calculating maximum power requirements. The kingpin torque needed to turn the wheel module is calculated by Equation 3.5 [22]. Although we are building a 4WS robot, not a 2WS tractor, this equation will be used in our calculations.

$$T = W f \sqrt{\frac{I_0}{A_0} + e^2} \tag{3.5}$$

Where:

T = kingpin steering torque
W = wheel load
f = effective friction coefficient
I_0 = polar moment of inertia of tire print
A = tire print area
e = kingpin offset

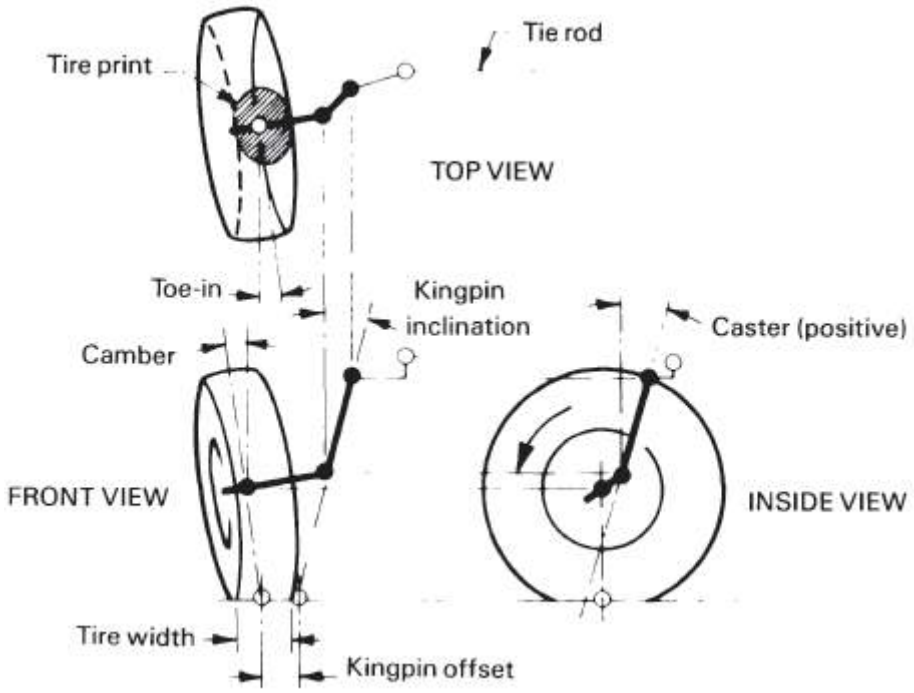


Figure 3.4: Tractor steering geometry. Figures courtesy of Wittren [42].

If the tire print is assumed to be circular (as suggested by Liljedahl et al. [22]), then:

$$\frac{I_0}{A} = \frac{\frac{\pi D^4}{32}}{\frac{\pi D^2}{4}} = \frac{D^2}{8} \tag{3.6}$$

The steering torque equation can then be written as:

$$T = W f \sqrt{\frac{D^2}{8} + e^2} \tag{3.7}$$

As described, our robot will have four wheel steering. Each wheel will be mounted on a wheel module, and each wheel module will be connected to the robot frame via a kingpin directly above the tire center-line (no kingpin offset). According to Figure 3.5, the effective friction coefficient is set to 0.7.

The wheels for the robot has not been defined, but will have an approximate width of 0.1 m. The tire print is assumed to be circular with a diameter equal to the tire width. The weight of the robot is assumed to be equally distributed on the four wheels.

For a 300 kg robot, the maximum steering torque is calculated using Equation 3.7.

$$T = \left(\frac{300\,kg \cdot 9.81\,N/kg}{4}\right) 0.7 \sqrt{\frac{(0.10\,m)^2}{8} + 0^2} = 18\,Nm \tag{3.8}$$

To be on the safe side, the minimum torque our steering motor must generate at kingpin is set to 120% of this value.

$$18\,Nm \cdot 1.2 = 22\,Nm$$

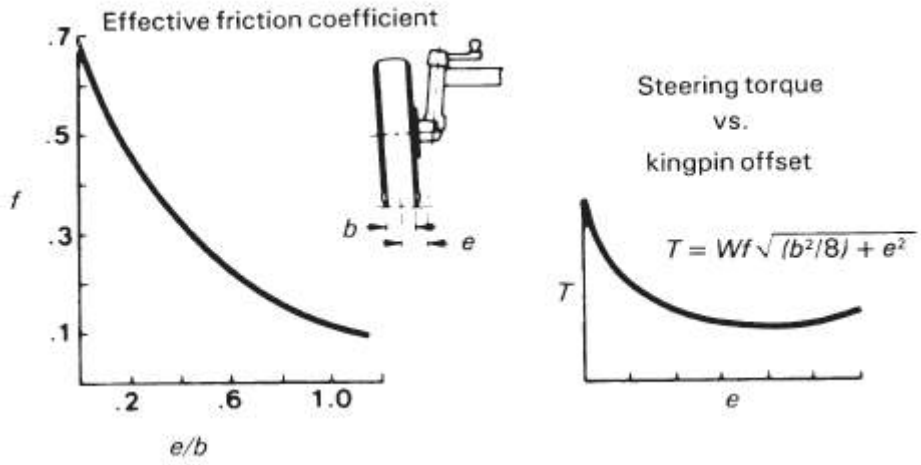


Figure 3.5: Typical curves based on rubber-tired vehicles on dry concrete. Figures courtesy of Wittren [42].

The steering system should have a wheel module turning rate in the range of 30-80 RPM. This is sufficiently fast for the robot to move around smoothly without having to wait for the wheel modules to turn, but not so fast that it represent any hazard to safety.

Chapter 4

Component Selection

4.1 Propulsion Components

The first thing to specify is what kind of motors to use on the robot. Given the short deadline and the arguments stated in the previous chapter, it is fairly obvious that electric motors with batteries will be our best shot at finishing the project in time. A combustion engine used directly for propulsion requires a multi-step gearbox and cooling systems. It will be too time consuming to install and control. The engine could of course be used to drive a generator, eliminating the need for a complex gearbox. This would however still be far more complicated and time consuming then using batteries for energy storage.

Electric motors generally do not generate enough torque to be connected directly to the wheel. Some kind of transmission may be required. This leaves us with two main options. Either to use one powerful motor connected to a hydraulic pump and installing hydraulic motors in each wheel, or using four smaller motors with mechanical transmissions, each installed close to its respective wheel.

Although hydraulic transmissions allow in-wheel motors with variable speed drive, the efficiency is to low and the process of automating is too time consuming to be used in this project. Electric motors allow efficient variable speed drive and easy control

As discussed in the chapter "Motors and transmissions", the brushless DC motor has many advantages and only two drawbacks compared to the brushed alternative. The fact that they are smaller, more efficient, and require less maintenance than brushed motors clearly weighs up for the slightly higher price and the need for more sophisticated controllers. This is why brushless motors will be used on the robot.

4.1.1 Drivetrain Solutions

Hub Motors

Brushless motors for wheel hub installation exist, and are used in anything from electric bikes (Figure 4.1(a)) to wheelchairs. The motors are made to work without a gearbox and thus have limited torque capability. Some high torque hub motors can be found, like the Chinese wheelbarrow motor shown in Figure 4.1(b). Motors like this one are meant for hobby use, and lack the documentation needed for a project like the NMBU Robot.

(a) E-bike with brushless motor. Image courtesy of *Heinzmann*.

(b) Wheelbarrow motor. Image courtesy of *alibaba.com*

Figure 4.1: Hub motors.

Motor with Worm Gear

A worm gear pair is compact and mounts to the motor at a right angle (Figure 4.2(a)). The gear is smooth in operation and offers high reduction in one single stage. The gear could be connected directly to the wheel with the motor inside the wheel module, making a sleek, compact design possible. However, due to the poor efficiency of these gears, this option is discarded.

Motor with Planetary Gear

Large trucks have hub reduction systems that reduce the speed at the wheel. This allows smaller diameter drive shafts, spinning faster with lower torque. Because of their compact design and high efficiency, planetary gears are used for such systems.

This principle could also be used on a robot. Having a planetary gearbox concealed inside the wheel allows for a compact and efficient design.

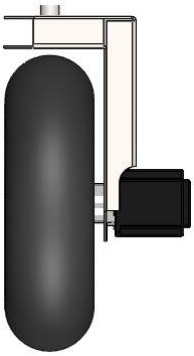
The large reduction ratio required for high speed low torque brushless motors cannot be obtained in a single stage planetary gearbox. A two stage gearbox is therefore required. This adds in axial size, but the gearbox will still fit inside a wheel. Connecting the motor directly to the gearbox increases the total length even further. As shown in Figure 4.2(b) the result is a lumpy design with the motors exposed to near-ground hazards like rocks or mud.

Obviously the motor will have to be moved away from the ground. A solution is to flip the motor and install it over the wheel as shown in 4.2(c). Bevel gears and different V-belts and roller chains are possible means of power transmission between the motor and the gearbox.

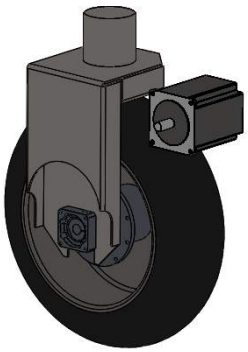
Bevel gears are more complicated and time consuming to install than the alternatives and V-belts will require speed measuring feedback devices at the wheels to account for slippage. Roller chains or timing belts provide easy installation, high efficiency and do not require any speed measuring sensors to account for slippage or even backlash (as the error will be divided by the gear ratio of the gearbox). Timing belts require more precise installation, but are quieter and generally seen as more sophisticated than roller chains. Timing belts is therefore the better option.

Most of the agricultural robot concepts we have seen uses some kind of reduction gear between the motor and the wheel, but we have not yet seen any that implement hub reduction. Gearboxes are heavy, so moving them closer to the ground will noticeably alter the overall center of gravity of the robot. A lower center of gravity

(a) Worm gears are used on the ZEUS platform. Image courtesy of the SAVAGE project [29].



(b) Coaxial motor, planet gear and wheel.



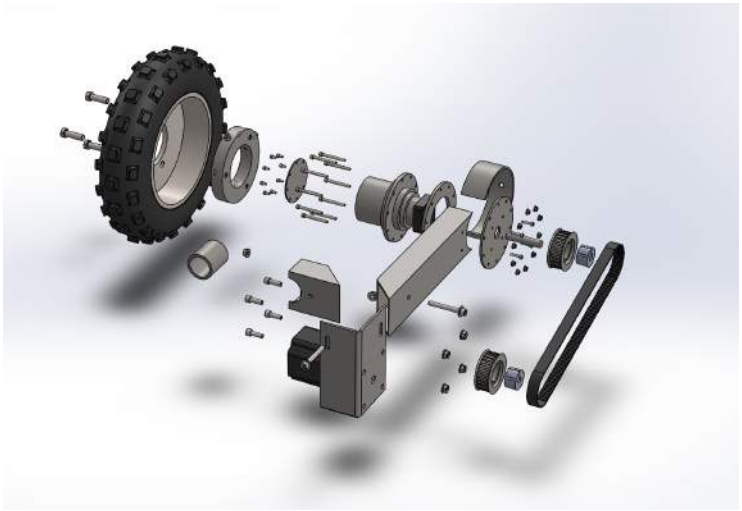
(c) Motor placed over the wheel.

Figure 4.2

will of course increase the robots stability, reducing the risks of rollover accidents. Another advantage of mounting the wheel directly on the gearbox, is that the other parts of the power transmission can be made lighter because they operate at very low torque levels.

Mounting the motor over the wheel makes it possible to use larger (more powerful) motors compared to in-wheel solutions, and still maintain a sleek, safe and compact design. It is also fairly easy to install components, and to do maintenance work.

Electro Drives AS, a local motor dealer, can provide both brushless motors and planetary gearboxes for our design, at reduced prices. The extra safety of dealing with local suppliers is seen as a huge advantage. Electro Drives cannot supply motor controllers to be used on a network, but recommend buying Roboteq controllers form roboteq.com. All in all this is the best drivetrain for our robot. Figure 4.3(b) show the assembly.



(a) Exploded view.

(b) Wheel module assembly.

Figure 4.3: The NMBU Robot's wheel module with motor and power transmissions. Figures courtesy of Fredrik Blomberg [6].

4.1.2 Propulsion Components

After settling on the principal design, motors and gearboxes with gear ratios of 60 were ordered from Electro Drives, motor controllers were ordered from Roboteq and timing belts with gear ratio 1 were ordered from *Tess*.

Motor - 3Men BL823-A02

The specifications for the BL823-A02 brushless motor from Taiwanese motor manufacturer 3Men are listed in Table 4.1. These inexpensive motors are compact compared to their power output, but have a somewhat short data sheet. However, Electro Drives have long experience with 3Men, and vouch for the quality of the motors.

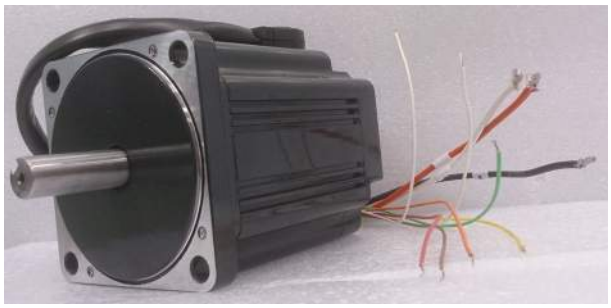


Figure 4.4: 3Men BL823-A02 BLDC motor.

The motors are delivered with Hall sensors and a temperature sensor, but no encoder (an encoder is a type of sensor for measuring shaft rotation, and are presented in the second part of this thesis). The shaft is accessible from the backside of the motor, and an encoder may be glued to the shaft. The need for such sensors is discussed in the second part of this thesis.

Table 4.1: 3Men BL823-A02 motor specifications.

Rated output	600 W
Rated torque	1.32 N m
Rated speed	4400 RPM
Rated current	14.5 A
Length	112.5 mm
Weight	3.5 kg

Motor controller - Roboteq HBL2360

The Roboteq HBL2360 motor controller (Figure 4.5) is a generic controller designed to be able to run a wide range of brushless DC motors. It has two independent channels, each capable of a continuous current of 50 A, a maximum current of 75 A, at a maximum voltage of 60 V. The controller supports CANopen, with bus connection through a 9-pin D-sub male connector. The controller can also quickly be configured via USB, and feature a microcomputer that can be programmed through

Figure 4.5: Roboteq HBL2360 BLDC motor controller.

the included software. The Hall sensors connects to the controller via a 10-pin Molex Microfit 3.0 connector, and a transition board with cable and Molex connector was acquired for easy soldering. The controller is also equipped with a 25-pin connector that may be used for digital/analog inputs/outputs, RC radio, analog joystick or encoder input.

For more information, see the manufacturers product catalogue [30].

Gearbox - Apex Dynamics AL110

Key specifications for the Apex Dynamics AL110 gearbox are listed in Table 4.2. Figure 4.6 show the gearbox.

The output of this efficient two stage planetary gearbox is intended to be used with a timing belt pulley. Instead we will mount it directly to the wheel, using the gearbox as a hub reduction unit. Each gearbox is fitted with a flange that will be bolted to the robot frame, which means that the gearboxes have to support the entire weight of the robot.

Figure 4.6: Apex Dynamics AL110 planetary gear.

The gearbox is designed to withstand 6500 N of radial load. Assuming static conditions and a 300 kg robot with weight equally distributed on the four wheels yields a safety factor of:

$$\frac{6500\,N}{9.81\,N/kg \cdot \frac{300\,kg}{4}} = 8.83$$

With such high safety factor, the gearbox should also be more than capable of withstanding any shocks that may occur during field operation (dynamic conditions).

The Apex Dynamics AL110 gearbox is maintenance free and sealed to IP65 standard protection (dust tight, protected against water jets), with helical gears ensuring smooth and quiet operation.

For more information, see the manufacturers product catalogue [12].

Table 4.2: Apex Dynamics AL110 planetary gearbox specifications.

Number of stages	2
Nominal output torque	140 N m
Emergency stop torque	420 N m
Max. Acceleration torque	252 N m
Continuous input speed	3000 RPM
Max. Input speed	8000 RPM
Backlash	≤ 7 arcmin
Max. radial load	6500 N
Max. axial load	3250 N
Efficiency	$\geq 94\%$
Mass moment of inertia	0.13 kgcm^2
Weight	4.1 kg

Timing Belts

A 1 : 1 timing belt will be used to transfer power between the motor and gearbox in each wheel module. The belt will spin with high speed and low torque. If needed, the gear ratio can easily be modified by replacing one of the pulleys. This will of course alter the overall gear ratio.

Wheels

The wheel supplier Røwde was contacted early in the planning process. Røwde has a wide selection of wheels for different farm machines, also for small sized ones. Different tire sizes and treads were considered. In the end 40 cm diameter snow blower wheels with steel rims were selected. For discussions regarding traction, See Meltzeter [24].

The AL110 planetary gears are not intended to be used as hub reduction units, and did not fit directly on the wheel. A flange had to be machined and fitted to the gearbox output. To make each flange, holes were drilled in a ring. The ring was slip-fit over the gear and fastened with a pair of set screws 90° apart. The wheel bore was extended to fit over the gear, and holes corresponding to those of the flange were drilled. The wheel could then easily be fitted to the gear. The set screw solution causes minimal harm to the gearbox, and makes it possible to shift the wheel so it can be perfectly aligned under the steering king pin. Figure 4.7 show the different parts of the assembly.

4.1.3 Verification of Drivetrain Torque Capacity

Each wheel module can be represented with the assembly in Figure 4.8. The wheel size, gear ratios and efficiencies are listed in Table 4.3. The efficiency of the timing

(a) Original wheel. (b) Modified rim, and gearbox with flange and hub-cap.

Figure 4.7: Adapting wheel and gearbox.

belt power transmission is set to 94%, which is equal to that of the Apex gearbox. This is somewhat conservative. According to Beardmore [32] a correctly design and installed timing belt should have an efficiency of 98%.

The required torque per motor (constant speed) can be estimated by multiplying the combined friction force calculated in the chapter "Power Requirements" with the wheel radius and dividing by the number of wheels in contact with the ground and each power transmissions efficiency and gear ratio:

$$M_A = \frac{724\,N \cdot 0.20\,m}{3 \cdot 0.94^2(1 \cdot 60)} = 0.91\,Nm$$

The power required from each motor (at the 3.5 km/h robot speed) is calculated by dividing the required wheel power by the overall efficiency of the power transmissions:

$$P_A = \frac{235\,W}{0.94^2} = 266\,W$$

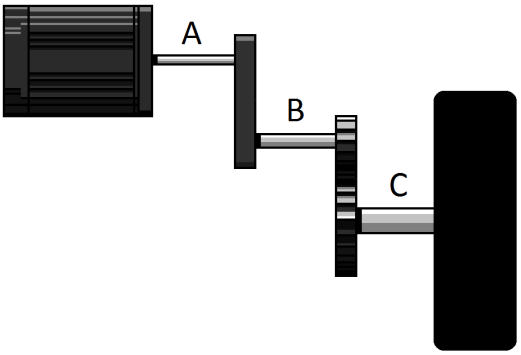


Figure 4.8: Principle drawing of the powertrain.

To calculate the additional torque needed to accelerate the robot, the moment of inertia of shaft A, B and C must be calculated.

The moment of inertia of the 3Men motors is not listed in the data sheet of the motor. For these calculations the moment of inertia of the motor is set to $3.8 \times 10^{-4}\,\mathrm{kgm}^2$, which is the same as for a Heinzmann PMS080 600 W motor [15].

According to a SolidWorks 3D model of the timing belt pulleys, each have a moment of inertia of $4.8 \times 10^{-4}\,\mathrm{kgm}^2$. The AL110 gearbox has a moment of inertia of $1.3 \times 10^{-5}\,\mathrm{kgm}^2$ [12], and the wheel is approximated as a thin disc with radius of 0.20 m and mass of 4 kg. Using Equation 4.1, which is the equation for the moment of inertia of a thin disc, the moment of inertia of each wheel is calculated to be $0.080\,\mathrm{kgm}^2$.

$$I_{thindisc} = \frac{m_{thindisc}(r_{thindisc})^2}{2} \tag{4.1}$$

Where:

$I_{thindisc}$ = moment of inertia of a thin disc.
$m_{thindisc}$ = disc mass
$r_{thindisc}$ = disc radius

The total moment of inertia of each of the shafts is then:

$I_A = 3.8 \cdot 10^{-4}\,kgm^2 + 4.8 \cdot 10^{-4}\,kgm^2 = 8.6 \cdot 10^{-4}\,kgm^2$,
$I_B = 4.8 \cdot 10^{-4}\,kgm^2 + 1.3 \cdot 10^{-5} = 4.9 \cdot 10^{-4}\,kgm^2$,
$I_C = 0.080\,kgm^2$.

We want our robot to be able to accelerate from standstill to 3.5 km/h in two seconds, that is an acceleration of $0.49\,\mathrm{m/s}^2$. As the wheel has a radius of 0.20 m, the angular acceleration at the motor shaft is:

$$\alpha_A = \frac{0.49\,m/s^2 \cdot 1 \cdot 60}{0.20\,m} = 146\,s^{-2}$$

Assuming three wheel traction and a total mass of 300 kg, Equation 3.4 yields 0.39 N m of acceleration torque per motor.

Table 4.3: Powertrain data.

I_A	moment of inertia, shaft A	$8.6 \times 10^{-4}\,\mathrm{kgm}^2$
I_B	moment of inertia, shaft B	$4.9 \times 10^{-4}\,\mathrm{kgm}^2$
I_C	moment of inertia, shaft C	$0.080\,\mathrm{kgm}^2$
$i_{A\to B}$	gear ratio, shaft $A \to B$	1
$i_{A\to C}$	gear ratio, shaft $A \to C$	60
η	efficiency of each transmission stage	0.94

Each of our motors have a rated torque of 1.32 N m. According to these calculations, this should be sufficient to meet the stated requirements.

600 W is somewhat more motor power than needed according to the calculations. However, as the gearbox continuous speed is 3000 RPM, and the timing belt gear ratio is 1, the maximum motor speed is limited to 3000 RPM, meaning we are not able to make use of the full motor potential (assuming rated torque). To utilize more of the motor potential, the timing belt gear ratio must be altered. This will enable the robot to generate more torque at the wheel with the same maximum speed of:

$$\frac{3000\,RPM}{60} \cdot \frac{\pi\,(rad/s)}{30\,RPM} \cdot 0.2\,m \cdot 3.6\frac{km/h}{m/s} = 3.8\,km/h$$

If higher speed is required, larger diameter wheels can be fitted.

4.2 Steering Components

Given the arguments in the "Propulsion" section of this chapter, also the steering motors should be electric with mechanical transmissions. They should run of the same voltage and be powered by the same battery pack as the propulsion motors. Electric motors allow 360° rotation of each wheel module with easy control.

The gearbox should have low backlash to ensure precise positioning, especially if the positioning feedback device is in the motor and not at the output shaft of the gearbox. The gearbox should also be able to handle vibrations and shocks that may occur while driving through a farm field.

The steering motors are expected to consume much less power than the propulsion motors. The efficiency of the steering motors and gearboxes will therefor make a smaller impact on the overall efficiency of the robot than those of the propulsion system. For this reason not only planetary gearboxes, but also inexpensive, compact, quiet and smooth operating worm gears could be used. For the same reason not only brushless, but also brushed motors should be considered.

4.2.1 Finding Components

The first motors we considered were high precision brushed DC motors from Swizz motor manufacturer Maxon Motor [25]. These could be delivered with encoders and motor controllers, but not with gearboxes of required gear ratio and load capability. Suitable worm gears were found in an on-line product catalogue from Lönne Drive Technology. This option would require us to manufacture an adapter to connect the motor to the gearbox, and was therefore discarded.

Motors from Kählig Antriebstechnik GmbH (KAG) were also considered. These were 94.2 W brushed servomotors that could be delivered with encoders and planetary gears with a gear ratio of 169. They could however not supply motor controllers. This option would limit the steering speed to 18 RPM, which is somewhat on the slow side.

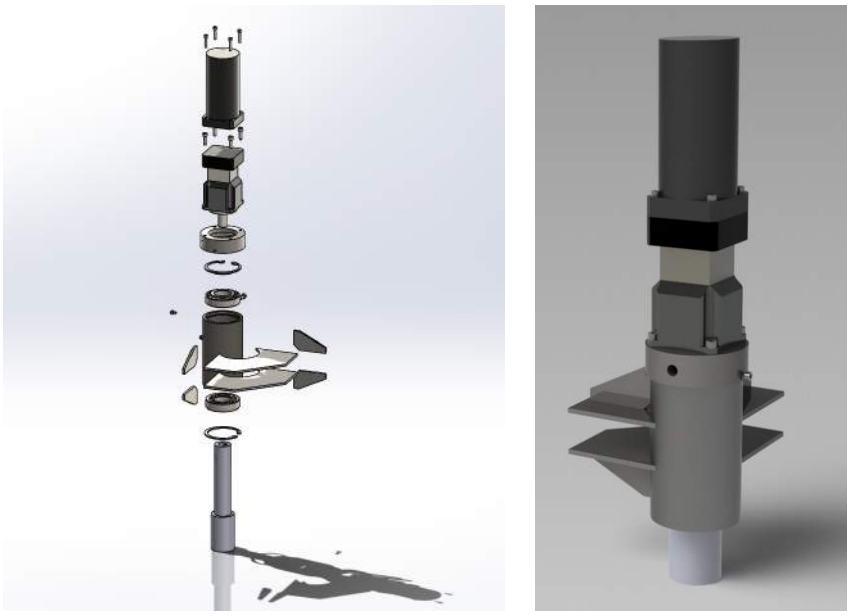
Electro Drives AS could provide sophisticated integrated servomotors from Danish manufacturer JVL with adapted planetary gearboxes from Apex Dynamics. Integrated motors have built in controllers, which saves time in the assembly and setup process. These particulate motors were of the brushless type with built in encoders. The motors can be controlled through the same network system as the propulsion motor controllers, and also run of the same voltage. The comfort of buying many components from one local dealer combined with excellent customer service and quality components at discount prices made us go for this option.

4.2.2 Steering System Components

Four JVL Mac141 servomotors and four Apex dynamics AB060 planetary gears was ordered from Electro Drives.

Motor - JVL Mac141

The Mac141 is an integrated servomotor that can be run of 12-48 VDC, with rated power of 134 W (at 48 VDC). The motor compact and is capable of generating 300%



(a) Exploded view. (b) Steering assembly.

Figure 4.9: The NMBU Robot's steering components from motor to kingpin. Figures courtesy of Fredrik Blomberg [6].

of rated torque for short periods of time. It also has a built-in optical incremental encoder. The servomotors key specifications are listed in 4.4.

An integrated motor is a motor with built-in controller This saves space and eliminates the time consuming process of setting up the motor with a generic controller from another manufacturer.

The JVL brushless motors can be delivered with a wide range of exchangeable expansion modules for different types of communication, like wi-fi, DeviceNet, Bluetooth, CANopen, etc. We ordered the CANopen module. Communication protocols are discussed in the second part of this thesis.

The expansion module features one 5-pin M12 power connection, two interconnected 5-pin M12 bus connectors and an 8-pin M12 I/O connector. The I/O can for example be used for end of travel or zero search feedback. It can also be used to connect the controller to a computer. The controller can then be configured through the JVL MacTalk software.

Table 4.4: JVL MAC141 servomotor specifications.

Rated output	134 W
Rated torque	0.48 N m
Peak torque	1.59 N m
Speed range	0-2700 RPM
Mass moment of inertia	0.227 kgcm^2
Length	172 mm
Weight	1.33 kg

(a)

(b)

(c)

Figure 4.10: JVL MAC141 BLDC servomotor.

Gearbox - Apex Dynamics AB060

Key specifications for the Apex Dynamics AB060 gearbox are listed in Table 4.5. This efficient two stage gearbox is lighter than those used for the propulsion motors, and is made for smaller loads. It is made to fit directly to the JVL servomotor, and has low backlash to allow precise steering. As the AL110, the AB060 also utilizes helical gears, it is maintenance free and sealed to IP65 standard protection. For more information, see the manufacturers product catalogue [11].

Table 4.5: Apex Dynamics AB060 planetary gearbox specifications.

Number of stages	2
Nominal output torque	55 N m
Emergency stop torque	165 N m
Max. Acceleration torque	99 N m
Nominal input speed	5000 RPM
Max. Input speed	10000 RPM
Backlash	$\leq$ 7 arcmin
Max. radial load	1530 N
Max. axial load	765 N
Efficiency	$\geq$ 94%
Mass moment of inertia	0.03 kgcm^2
Weight	1.5 kg

4.2.3 Verification of Steering System Torque Capacity

Given the servomotors rated torque, and the gearbox gear ratio and efficiency listed in Table 4.4 and Table 4.5, the torque generated at the steering kingpin can easily

Figure 4.11: Apex Dynamics AB060 planetary gear.

be calculated:

$$0.48\,Nm \cdot 60 \cdot 0.94 = 27\,Nm$$

According to the calculations done in the previous chapter, this should be sufficient to turn the wheel module even in the most difficult conditions. However, the motor is capable of generating three times this amount of torque, which surpasses the nominal output torque of the gearbox. Although the emergency stop torque of the gearbox is higher than what the motor can produce at the gearbox output, a software limit triggering an emergency stop should be set at 0.98 N m motor torque. This corresponds to 55 N m of gearbox output torque (nominal gearbox torque). It is highly unlikely that the steering system will encounter loads as great as this, and if it does, it is probably due to an error situation. In such case the system should be shut down for safety reasons.

Axial and radial load on the gearbox will be negligible as the kingpin bearings will support the weight of the robot.

At recommended maximum motors speed of 2700 RPM [18], the wheel module will turn at rate of:

$$\frac{2700\,RPM}{60} = 45\,RPM$$

This is within the desired range stated in the previous chapter.

Figure 4.12 show the robot during the assembly process.

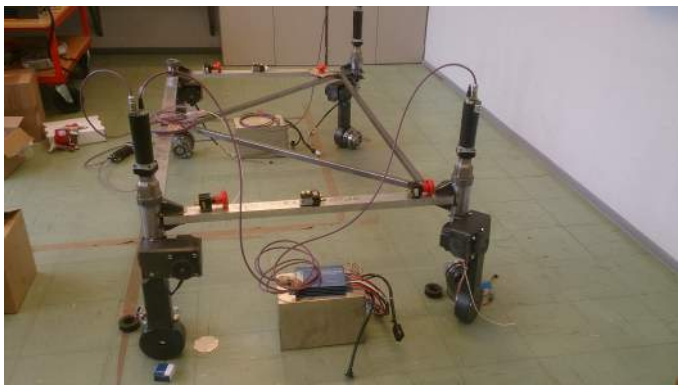


Figure 4.12: Assembling the robot.

Chapter 5

Power

5.1 Batteries

The battery selection process is not a part of this thesis, but the power distribution circuitry given these specific batteries is. Two 30 A h, 48 V $LiFePO_4$ batteries were ordered from Golden Motor in China.

Ideally the battery cells should all be connected to form one single battery pack managed by one smart BMS (battery management system). Instead we have two separate batteries managed by two separate BMSs. The batteries do not feature any interface for communication, so the two BMSs cannot communicate. Connecting the two batteries without a common BMS may shorten battery life compared to using one single pack.

The first battery was delivered one day before the other. When the battery was connected to the including charger via a (3-pin) XLR connector, it short circuited. This caused one of the pins of the male connector to break. When we checked the battery with a multimeter, we discovered that there was a $53\,V$ difference in potential between two battery slots corresponding to two interconnected pins in the charger. The battery had not been wired in the correct manner by the manufacturer. According to powerstream.com, pin assignment should be as shown in Figure 5.1. On our battery, slot 3 of the connector had been tied to the positive, not the negative pole of the battery. The second battery was correctly wired, but both batteries had a voltage of $53\,V$, not $48\,V$ across the terminals.

The cell potential for a $LiFePo_4$ varies from a minimum discharge voltage of 2.8 V to 3.6 V when fully charged. Our batteries have 16 cells in series each, consequently the voltage of the batteries can be anything from 44.8 V to 57.6 V depending on the state of charge. 57.6 V is considerably higher than 48 V, which the NMBU Robot has been design for.

The Roboteq motor controller is able to handle up to 60 V, and according to Electro Drives, the extra voltage should not be harmful to the propulsion motors either. The high voltage is however a problem for the steering motors. According to the JVL customer support, the absolute maximum voltage the motors can handle is 50 V. Higher voltage will cause the motor to brake after few hours of operation.

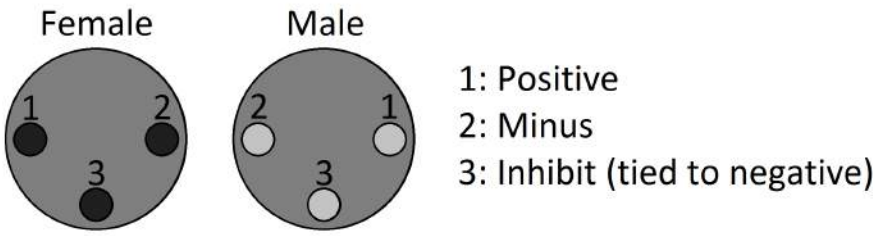


Figure 5.1: Standard pin assignment according to powerstream.com.

5.2 Power Circuit

Figure 5.3(a) shows schematics for the NMBU Robot's power distribution. The various parts will be explained in the following sections.

5.2.1 Power Supply

It is possible to remove some cells from each battery to reduce the voltage, but then our *smart battery charger* can no longer be used to charge them. The charger is made to provide constant current, then constant voltage at 57 V. As a preliminary solution, DC-DC converters will be used with the servomotors. Two 350 W Mean Well DC-DC converters has been acquired form ELFA Distrelec to ensure stable 48 V power supply for the servomotors. Each converter will power two motors. The converters take variable high voltage in the range 36-72 V at the input and output a constant voltage, but will only work with an output current of 7.3 A or less. This is more than sufficient to allow the servomotors to run at rated power.

The robot steering system requires four inductive sensors for calibration (see *Part II*). These sensors run on 12-24 VDC. Two converters from Recom 12 V, 7.0 W will be used. Each will supply two sensors with power.

5.2.2 Safety

It is important to distribute the robots power in a safe manner. Each of the batteries will have its own main fuse (*S1*, *S5*), and a contactor relay (*RY1*, *RY2*) that breaks the circuit if one of the four emergency stop switches (*ESS*) is activated. All components are also to be protected by fuses.

JVL recommends using a 10 A fuse for each motor. As the DC/DC converter will stop transmitting power at 7.3 A output current, a 7.5 A fuse will be used in front of each converter. The Roboteq controllers can handle a current of 70 A, but as the rated current of each propulsion motors is only 14.5 A, a 40 A fuse will be used in front of each Roboteq. A 20 A current limit will also be set in the controller for each motor.

It is important that at least one emergency stop button can be reached at any time by the operator or others in the immediate vicinity of the robot. Four emergency stop buttons will therefore be distributed across the robot, one in each corner. Both contactor coils are connected to the same fuse. The ESSs are connected between this fuse and the coils. This is to ensure that both batteries are disconnected

Figure 5.2: Contacts damaged by high current flow.

if the fuse breaks or one of the EESs are pushed. For more information on the contactors, see the datasheet [1].

The robot will also feature an ignition key to prevent unauthorized people to start-up the robot by accident, and to ensure that the batteries are disconnected when the robot is not in use.

5.2.3 Regeneration

Regeneration occur when an outside force try to run the motor at a higher speed than the rotating stator field. An example of this is when the robot is driving down a slope. The controller needs to get rid of this extra energy. For low regenerative power, the energy can be dissipated as heat in the controller, but for large regenerative power, energy must be fed to the batteries. Therefore, if one of the propulsion motor controller fuses blows or one of the emergency buttons are activated, a diode (*D1* and *D2*) ensures a passage to the battery for any regenerative current (as suggested in the controller data sheet [30]).

5.2.4 Start-Up

To prevent reduction in the servo motors peak performance, JVL recommend connecting 4700 µF/50 V capacitors across the servo motors less than 3 meters from the motor, presumably to smooth out the ripples in the current from the battery caused by the conversion from DC to AC in the motor controller. The capacitors are represented by *C1*, *C2*, *C3* and *C4* in the schematics.

In addition to *C1*, *C2*, *C3* and *C4*, the Roboteq controllers contain large capacitors. Engaging the main relays without pre-charging these will result in large current flows. This is harmful to the relays in that it may cause arcing damage. Arcing damage leads to increased resistance of the relay contacts, and the voltage drop across the relay will therefor increase with the number of operating cycles. Figure 5.2 shows the battery contact after a few cycles of connecting/disconnecting it to one of the Roboteq controllers without any current limiting device. The contacts are clearly damaged.

To avoid damage caused by arcing, the capacitors must be pre-charged whilst the current is limited by a resistor. The battery voltage can be as high as 57 V. By using a 120 Ω resistor the initial power dissipated in the resistor is limited to

$$\frac{(57\,V)^2}{120\,\Omega} = 27\,W$$

.

The Roboteq must be turned off for its capacitor to be pre-charged. That is, the power control (*Cv*) must be connected to ground during the start-up sequence. Also it is not possible to use the Roboteq's built in microcomputer while the capacitors are pre-charging.

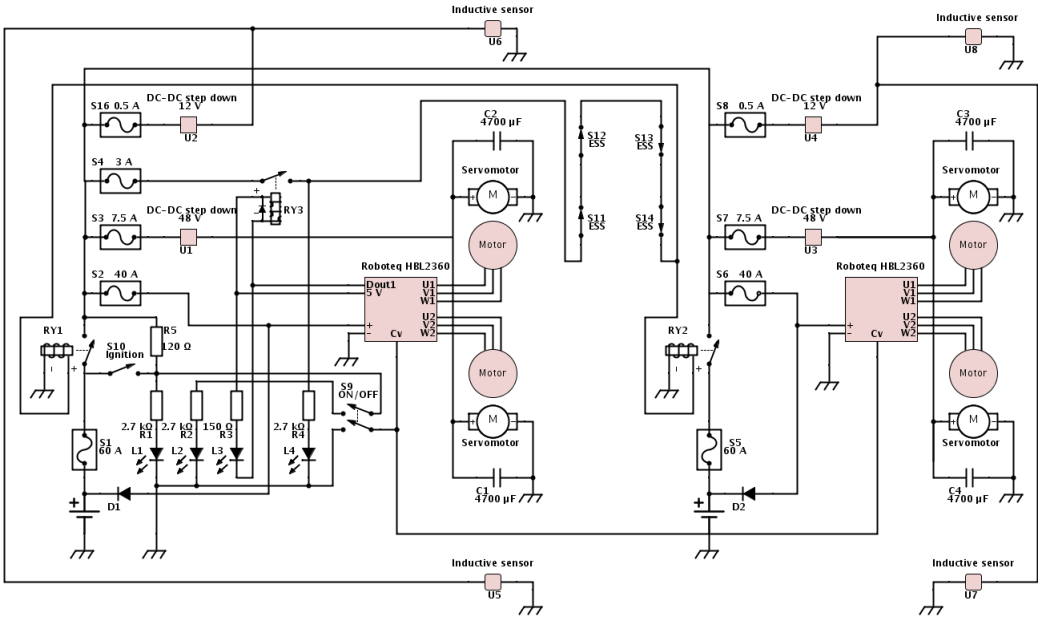
As Figure 5.3(a) shows, a resistor is connected across the main contactor relays. By turning the ignition key (*S10*), current will start to flow through the resistor (*R5*) to the various capacitors. The operator should then wait a few seconds for the capacitors to charge before setting the ON/OFF switch (*S9*) to the ON position. When ON, the Roboteq controllers power control is disconnected from ground, thus turning the controllers on. Drawing current from the capacitors, a digital output on the left side controller will then immediately close a relay (*RY3*) on the power supply of the main contactor coils. The main contactors will then engage, bypassing the current limiting resistor and supplying all components with power. The start-up sequence is then completed.

By setting the ON/OFF switch to the OFF position, the Roboteq controllers are turned off, and the relay on the main contacor coil circuit will disengage, disconnecting the batteries.

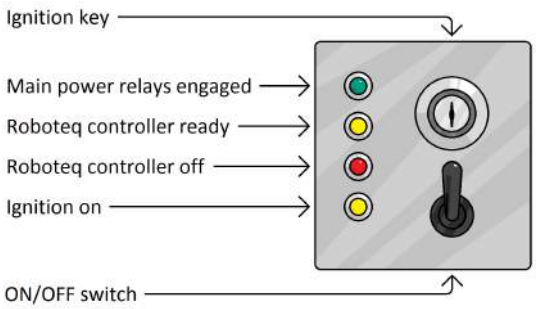
Future Start-Up

In the future the start-up sequence can be done by using a microcontroller or a I/O device connected to a computer. An example is shown in Figure 5.4.

To start the robot, one first have to turn on the ignition key (*S15*), then press the start button (*S9*). Pressing the start button will activate an input initiating the start-up program. First an output closes a relay connecting the Roboteq controllers power control wires to ground (*RY5*), thus enabling charging of the capacitors. A relay (*RY4*) then engages so current can flow to the capacitors through a resistor limiting the current (*R5*). This relay is connected across the left side main contactor relay. After a time delay, a digital output engages a relay (*RY3*) on the power supply of the main contactor relays, and the main relays engages. The Roboteqs are turned on by disengaging *RY5*. *RY4* also disengages as it is no longer needed. A green LEDs tells the operator that the start-up sequence is finished and that the main relay coil have power. The stop button is also connected to one of the controller's inputs. Pushing this will make the controller disengage *RY3*, thus cutting the power supply for the main contactor coils which disconnects the batteries.

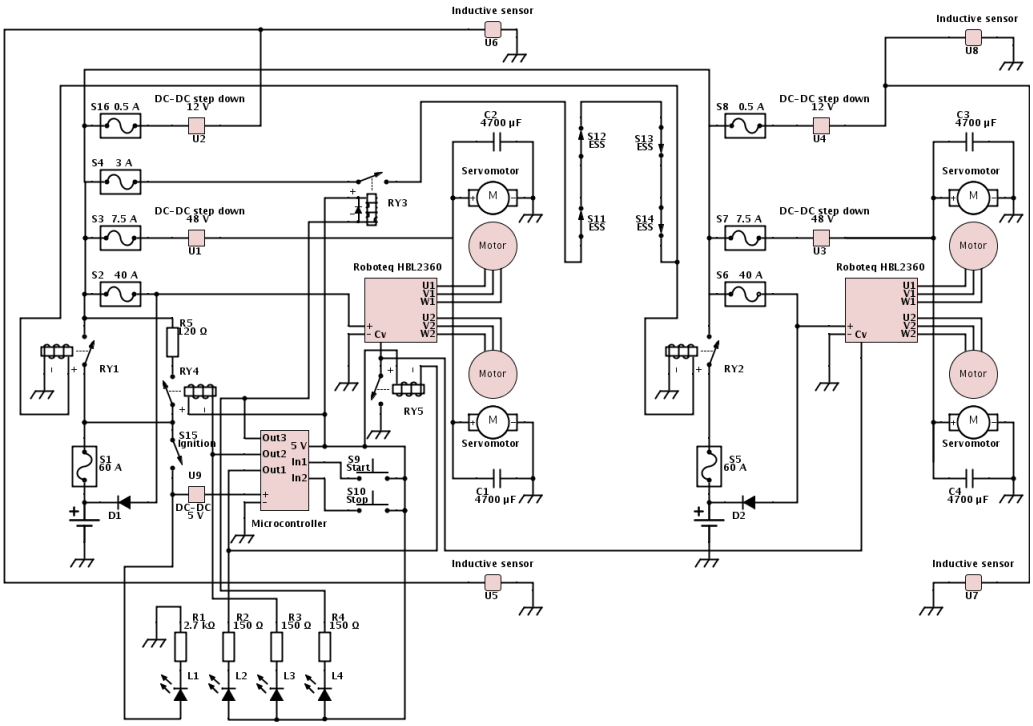


(a) The NMBU Robot's powercircuit.

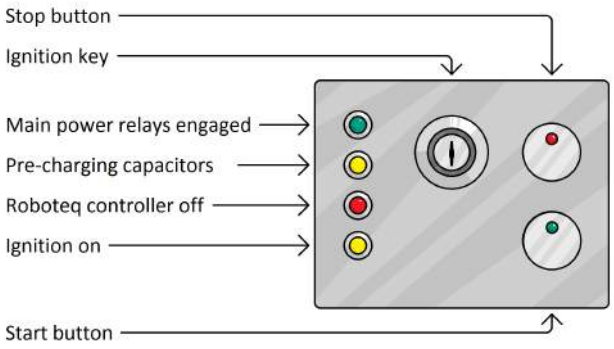


(b) Diodes and switches explained (LED from bottom to top: *L1*, *L2*, *L3*, *L4*.

Figure 5.3: NMBU Robot start-up and power.



(a) Possible future power circuit.



(b) Diodes and switches explained (LED from bottom to top: *L1*, *L2*, *L3*, *L4*.

Figure 5.4: Example of power circuit with microcontroller controlling the start-up sequence.

Chapter 6

Conclusion, Part I

Availability and short delivery time was of priority when selecting parts for the powertrain and steering system, and looking at the data sheets, we may not have perfectly matched components. For the propulsion, the continuous gearbox speed rating (3000 RPM) is less than the rated motor speed (4400 RPM). As the gearbox limits the motor speed to 68% of rated, the current drivetrain configuration do not make use of the full motor potential of 600 W per (assuming rated torque). Although not perfectly matched, the components are more than suited for testing the hub reduction based drivetrain. If higher speed is required in the future, larger diameter wheels can be fitted. In that case, the timing belt ratio should also be altered to increase the amount of torque that can be generated at the wheel.

The steering could probably be made with less expensive parts, and still have acceptable performance and efficiency. Cheap worm gears could have been used instead of the more costly planetary gears, and brushed DC motors could have replaced the expensive brushless motors. On the other hand, it is unlikely that the acquired components will fail. Both the gearboxes and the integrated motors are compact, precise and seem to be well-made. They also give the robot a more modern, *state of the art* feel. The steering assembly is sturdy with low backlash, and will probably not need any major modifications for the robots lifetime.

In the future, all subsystems must be started from the main computer. With the *future power distribution schematics* shown in Figure 5.4(a) it will be possible to start-up and shut down the propulsion and steering systems using ROS (with the main computer giving inputs to the microcontroller instead of buttons).

The current configuration of the acquired components is sufficient for the initial testing and development of the NMBU Robot, but as the robot is tested in various conditions, modifications may have to be made. Both the steering and propulsion system are highly modifiable, and their characteristics are noticeably altered by changing few components.

Part II

Control

Chapter 7

Controlling the Robot

Making a robot move and perform tasks require interaction between many different components. The position and orientation of the robot and its actuators, and information of the robot's surroundings is measured by sensors. From the sensor data, processors calculate the required actuation according to algorithms in the robot's different programs, and commands are sent to the actuators.

Real time control requires efficient and reliable communication and robust programs. Choosing the correct components and methods are therefore critical for safe and predictable robot operation.

This chapter covers the basics of control, communication protocols and different sensors that are relevant for building a mobile platform.

7.1 Motor Control

When driving a car the driver must constantly adjust the throttle to maintain constant speed at varying conditions. With a quick look at the speedometer the difference between desired speed and actual speed is calculated and the throttle is adjusted accordingly.

In automation and robotics constant adjustment like this is done by a controller. If the controller receives feedback from sensors monitoring the process, it is called closed loop control. Closed loop control enables the controller to compensate for disturbances in the process. An open loop controller receives no feedback from the process it controls, and does not know anything about the process output.

Feedback is critical for accurate motor control, both for speed and position. Varying loads and disturbances cause the motor to react differently to the same

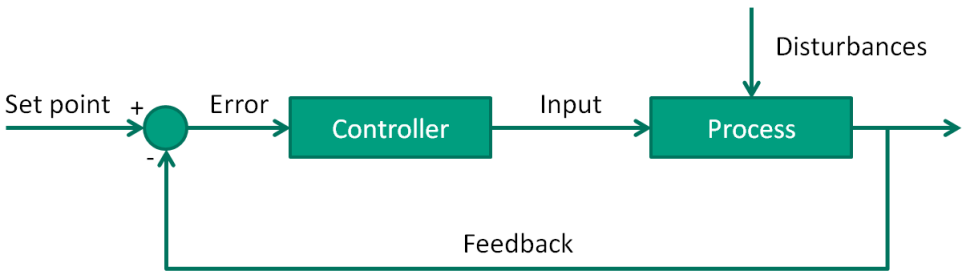


Figure 7.1: Closed loop process control.

input under different conditions, and it is therefore necessary to adjust the motor input based on information from sensors monitoring the output shaft. Different sensors can be used to measure shaft rotation, like encoders, Hall sensors, resolvers etc. The basic principles of some of these sensors are explained later in this chapter.

The most common type of controller is the PID controller which will be presented in brief in the following subsection. Other controllers try to predict the future based on a dynamic model of the system. These are called MPC (Model Predicative Controllers) and they account for future timeslots by optimizing a cost function over a finite time horizon. Only the current timeslot is implemented for each calculation. For more in-depth descriptions, see From [14].

7.1.1 PID Controller

A proportional-integral-derivative controller calculates the difference between the measured process variable and the desired setpoint, and controls the input to minimize this error. This is done with three different constant parameters. In essence, the proportional part of the controller deals with the present by multiplying the error with a gain value, the integral part deals with the past by accumulating past errors, while the derivative part tries to predict future error based on the current rate of change. The weighted sum of these is used to control the process input.

Although it is the most common type of controller, using a PID controller does not guarantee an optimal controller or stability. Most motor controllers use some form of PID.

7.2 Sensors

7.2.1 Rotary Encoders

Rotary encoders are electromechanical transducers used for measuring shaft rotation. They sense a number of discrete positions per revolution and output pulses that can be counted, or data describing the exact position of the shaft. Rotary encoders are used to measure shaft speed or shaft angle in servo feedback loops. They can be mounted on the motor shaft to measure position indirectly, or they can be mounted on the rotating machine part of interest to measure position directly.

Rotary encoders come in shaft or hollow shaft mounting configuration, with resolution ranging from 50 to 2 304 000 count per revolution [33]. Magnetic encoders and direct contact or brush-type encoders exist and are used to some extent, but the by far most popular rotary encoders are incremental optical shaft-angle encoders and absolute optical shaft-angle encoders [33].

Incremental Encoder

Incremental encoders consist of a glass or plastic code disk rotating between a light source and a photodetector assembly. The light source is usually a light emitting diode (LED).

A pattern of equally spaced transparent and opaque segments circles a *code disc*, and causes the light from the LED to be "chopped" as the shaft rotates, as seen in Figure 7.2. Depending on whether it can "See" the light from the LED or

not, a photodetector output high or low signals. The rotational speed of the shaft can then be calculated by dividing the number of counts by time and the known number of counts per revolution. By adding an additional photodetector 90° out of phase to the first, the direction of rotation can be determined. Each photodetector outputs square signals to its own *output channel*, channel *A* and *B*. Looking at which channel is leading (outputting high signals 90° before the other) yields the direction of rotation. This type of encoder is called quadrature encoder, and is the most common type of incremental encoders [33].

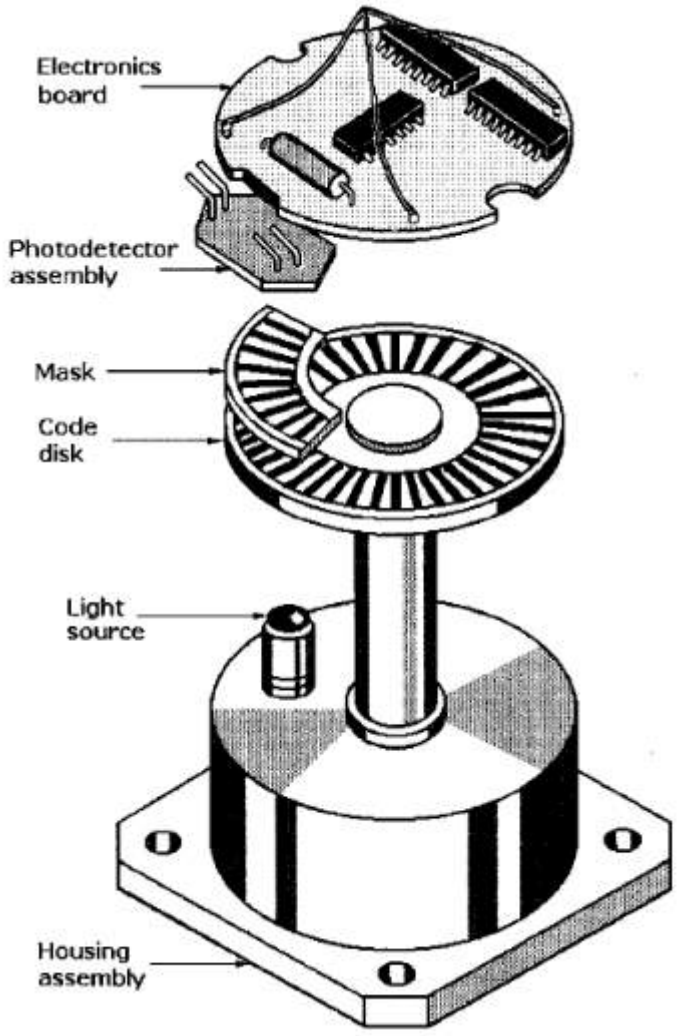


Figure 7.2: Incremental encoder. Figure courtesy of Sandin [33].

A third output channel, Z, is included in many incremental encoders. This gives of a high signal for a certain angle once per revolution, making it possible to determine the rotor position relative to this angle.

As an incremental encoder provides information about the motion of the shaft only, and not the absolute position of the shaft, a servomotor using an incremental encoder for feedback have to perform a homing sequence at every power-up to find some known reference.

Absolute Encoder

An absolute encoder provides a unique binary output for each shaft angle. The encoder is similar to an incremental encoder, but the code disk can have up to 20 tracks of segments [33]. The photodetectors are radially arranged and read a unique natural binary or Gray code for every new step within one revolution. Figure 7.3 example of a absolute encoder code disk.

Multi-turn encoders exist. These consist of multiple single-turn encoders and reduction gears. The first encoder yields the orientation of the shaft within the current revolution; the next counts the number of revolutions etc.

If the encoder loses power, an absolute encoder will still output the correct angle at start-up, and there is no need for a homing sequence.

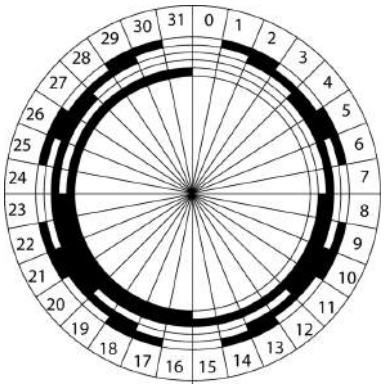


Figure 7.3: Absolute encoder with grey code. Figure courtesy of Rockwell Automation.

7.2.2 Proximity Sensor

In many situations it is necessary to sense the robot's environment, or the position of actuators without making physical contact. A proximity sensor is used to sense if two objects are close to each other before contact is made. The sensor finds many uses, from navigation to measuring the speed of a rotor. It can also be used for calibrating a servo motor e.g. by sensing when the servo motor is in zero-position.

Different types of proximity sensors exist, such as magnetic, eddy current and Hall-effect, optical, ultrasonic, inductive and capacitive. Of these, only proximity sensors will be covered by this thesis. For description of other types of proximity sensors, see Paul E. Sandin [33].

Inductive Proximity Sensor

An inductive proximity sensor is a coil with a ferrite core, an oscillator/detector and a solid state switch. The sensor sense metal surfaces. When metal is close to the sensor, the amplitude of the oscillation diminish. This is sensed by the detector which turns the solid state switch off. When the metal is removed, the switch is turned back on.

7.3 Communication Protocol

7.3.1 Controller Area Network

CAN bus was developed by Robert Bosh GmbH in the mid-1980s as a method for serial communication in the automotive industry. The aim was to reduce the complex wiring harness in cars, which would save weight and increase reliability, fuel efficiency and safety.

Since it was introduced, CAN bus has gained popularity, not only in the automotive industry, but in other industries where network solutions are beneficial. Today, CAN bus is found in medical equipment, automated equipment, robots etc.

CAN is an ISO defined bus with high immunity to electrical interference and the ability to self-diagnose and repair data errors. While traditional networks, such as USB or Ethernet, send large blocks of data from point to point under supervision of a central bus master, CAN broadcast short messages to the entire network. These short messages can be temperature, RPM, etc. Some messages are destined for just one node (motor controllers, sensors etc.), others for multiple nodes, but all nodes receive all messages on the network, and all nodes will acknowledge that the message was properly received. It is then up to each node to decide if it should process or discard the message based on the message ID. An advantage with this system is that new nodes can be added to the network without having to reprogram all other nodes.

Open System Interconnection (OSI) is a reference framework defined by ISO (ISO/IEC 7498-1). This is a seven-layered reference model for internal functions of a communication system (Figure 7.4). Each layer serves the layer above it, and is served by the layer below it. The CAN protocol implements most of the lower two OSI layers (the Data Link Layer and the Physical Layer). Boch initially left out the communication medium portion of the model. This was to enable system designers to select the best communication medium (single wire, twisted pair, optical, etc) for each individual application. To ensure interoperability, ISO and SAE (Society of Automotive Engineers) have defined some protocols based on CAN that defines the communication medium, specifying all of the lower two OSI levels.

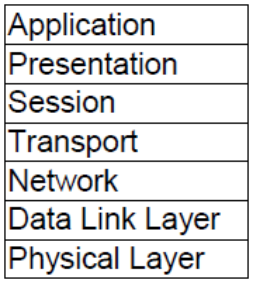


Figure 7.4: OSI model.

Four different types of messages (frames) are specified in CAN. Data frames are used when a node transmit information to any or all other nodes. The Remote frame is used when a node request data to be sent to it. Error frames are generated by any node that detect one of the protocol errors specified in CAN, and Overload errors are sent by nodes that require more time to process former messages before receiving new ones.

CAN is a CSMA/CD communication protocol. CSMA stands for Carrier Sense Multiple Access, meaning that any node on the network must wait a period of no bus activity before trying to transmit a message. After this period, all nodes have an equal opportunity to transmit messages. CD stands for Collision Detection. In CAN protocol, if two nodes try to transmit at the same time, the message with the highest priority will be sent without delay while the other node wait. CAN defines the logic bit 0 as dominant and 1 as recessive. A dominant bit state will always win arbitration. The nodes are monitoring the bus to make sure the bits they try to send appear on the bus. If two nodes try to transmit at the same time, the message with the lowest identifier will at some point have a dominant bit where the message with the higher identifier has a recessive. The node transmitting the message with the high value identifier will detect that its recessive bit did not appear on the bus and stop transmitting. Thus the lower the identifier value, the higher the priority of the message.

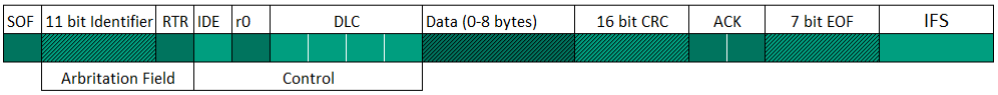


Figure 7.5: Can Data Frame.

The data frame consists of different fields. The standard frame as shown in Figure 7.5 has an 11-bit identifier. In CAN version 2.0B, there is also an extended frame with a 29-bit identifier.

- SOF: One single dominant start of frame (SOF) bit marks the start of a message.
- 11-bit Identifier: A unique identifier for the data (e.g. front left wheel RPM). The lower the binary value of the message identifier, the higher its priority.
- RTR: A single Remote Transmission Request bit. This is dominant (0) for data frames and recessive (1) for remote frames.
- IDE: A single identifier extension bit. Dominant bit indicates standard message without extension.
- r0: Reserved bit
- DLC: 4 bit indicating the number of bytes of data being transmitted (or number of bytes requested for remote frame).
- Data: The application data. Maximum size is 8 bytes.
- CRC: Cyclic Redundancy Check contains the number of transmitted bits of the application data for error detection.
- ACK: 2 bit. Transmitter sends recessive acknowledgment bit. Every node receiving an accurate message overwrites with dominant. The second bit is a recessive delimiter.
- EOF: 7 recessive bit marks End of Frame (during normal operation, 5 bits of same logic level will be followed by a *stuff* bit of the opposite logic level).
- IFS: Minimum number of bits separating consecutive messages.

CAN protocol version 2.0B is capable of communication rates up to 1 Mbit/s. This is sufficient for time critical parameters to be serially transmitted without latency. CAN nodes are able to detect faults, and transition to different modes accordingly. Based on the severity of the problem, they can go to complete shut-down (bus-off). This is called Fault Confinements, and prevents faulty nodes from monopolizing the entire bandwidth.

To terminate the bus, a 120 Ω resistor is connected between the high and can low wires on either end of the network (See Figure 7.6.

For more information on CAN, see the following references: [26] [9].

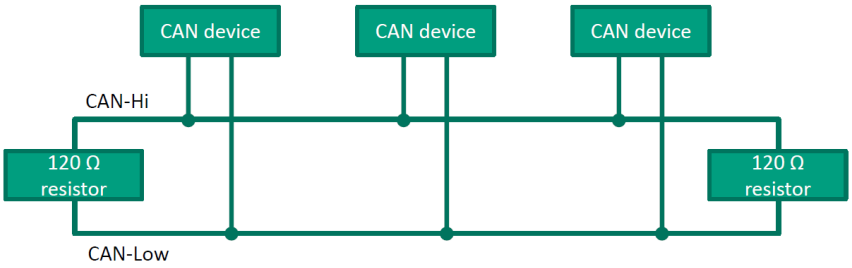


Figure 7.6: Example of CAN network.

7.3.2 CANopen

CANopen is a communication protocol and device profile specification developed by CiA (CAN in Automation) to be used in distributed industrial automation systems based on CAN. While CAN covers the two lowest layers of the OSI model, CANopen covers all the layers above.

All the specifications of CANopen are developed by CiA member companies. The basic specifications are defined in the document CiA 301 *CANopen Application Layer and Communication Protocol*, while CiA 302 defines *Framework for Programmable Devices* in particular PLCs, HMIs and CANopen tools.

An important property of CANopen, is the Object Dictionary (OD). This is a standardized device description all CANopen devices must have for communication and configuration. The entries in the OD are data, functions and parameters of the device. All devices must have a server that can handle read write requests to the OD. The part of the OD describing the device and communication parameters is common for all types of devices.

Messages are based on CAN frame format. The 11 bit identifier (29 for extended CAN) is split into two parts, a 4 bit function code and a 7 bit node ID. Together with the start of frame bit and the remote transmission request bit, these form the COB-ID (communication Object Identifier). The 7 bit node ID limits the number of devices on a CANopen network to 127 nodes.

There are two ways of accessing the ODs, SDOs and PDOs. SDOs (Service Data Object) provide direct access to a device, and are used for configuring a device, or for up- and downloading of large data blocks based on a client server communication. PDOs (Process Data Objects) transmit process data according to the producer-consumer model. Short messages containing between one and eight bytes of data, are efficiently transferred between nodes. The PDOs do not have a fixed format.

Both the transmitter and the receiver must therefore know how the content of a PDO is to be interpreted.

For more information regarding CANopen, see the CAN in Automation webpage [3]. IXXAT, supplier of data communication technology, also provides a good introduction to the CANopen communication protocol [16].

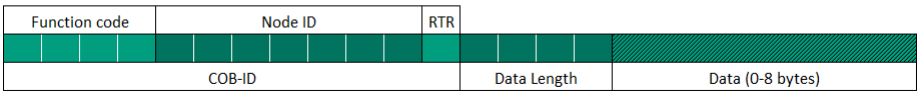


Figure 7.7: CANopen message.

7.3.3 ISOBUS

Modern farm machinery is typically set up and controlled from a monitor in the tractor cabin. In 2001, to prevent having to fill the cabin with monitors controlling different equipment, manufacturers of farm machinery agreed to implement a common standard for communication interfaces on tractors, implements and farm management systems. The ISO 11783 standard (commonly referred to as ISOBUS) was developed. This CAN based network system allows the farmer to control all machinery from one monitor, and ensures interoperability between equipment from different manufacturers. Equipment connects through a standard cable for plug-and-play operation. All data collected by the equipment and the tractor is centralized, and can be transferred to a computer for analyzing.

Europe has been the main arena for the development of ISOBUS, but North American manufacturers are starting to see the potential of the standard, and are currently making rapid advancements.

7.4 Framework

A problem for many robot projects has been the lack of a common framework that can be adapted by different platforms, and which implements support for frequently needed systems such as for example navigation and vision systems. When building a robot from scratch, time is often wasted on getting these systems to work, and researchers who are meant to be developing, often have to spend a lot of time reinventing the wheel by writing code that have been written by others many times before.

A software framework is a platform that consists of software providing generic functionality. This software can be modified by user-written code to best suit a given application. A robotic framework provides the infrastructure to build applications, and must be able to cover a lot of different aspects like communication, computing, hardware abstractions and so on. There is an increasing demand for common frameworks to speed up the development process of new robots. This has led to the creation of several such frameworks over the last decade. Some are commercial and have been developed by large software companies, such as MRDS (Microsoft Robotic Development Studio), other are based on open-source mentality, such as ROS (Robotic Operating System) and Orocos (Open RObot COntrol Software).

The main advantage of the open-source frameworks is the huge communities of users who write and share code. Algorithms for navigation, path planning, vision systems, drivers and wrappers for different devices, etc. can be found in ever growing on-line repositories. This availability of ready-made open-source software saves valuable time in the development of new robots, and is a feature that attracts more and more users.

The following subsections will discuss some of the most important frameworks.

7.4.1 ROS

ROS has its roots in various framework prototypes developed at Stanford University in the mid-2000s. In 2007 Willow garage, a robotics research lab and technology incubator, started their "Personal Robotics project" with ROS as a formal entity. From the start ROS has been developed under a permissive open-source license, and many have made contributions to its progress.

ROS is a framework consisting of tools libraries and conventions for writing robot software in Python or C++, and can be used with many different robots. It is fully supported on Linux Ubuntu, while experimental versions exist for Windows and other platforms.

The goal of ROS is to support code reuse and to provide a general framework that can be used with a wide range of robotic software. ROS is a distributed framework of processes called nodes. A control system generally consists of many nodes. One node may control a motor, another node path planning and so on. This allows individual design of executables that can be loosely coupled at runtime.

Nodes can communicate by subscribing to or publishing *messages* via different *topics*. Each topic transmits messages of a certain kind of data, for example messages concerning the control of a motor. Each topic may have several publishing/subscribing nodes, and each node may subscribe and/or publish to multiple topics. This decouples the production and consumption of information. The nodes are therefore unaware of each other's existence.

The ROS topics publish/subscribe model is not suited for request/reply interactions. For one-to-one communication a *service* is used. A service is defined by one request message and one reply message, and allows nodes to send a request and receive a response.

ROS uses a XML format called URDF (Unified Robot Description Format) to represent the robot model. The URDF file is used to define the geometry of the robot, and the kinematic and dynamic properties of the links and joints. It can also describe the robot's sensors, and may include simulation properties such as friction and dampening. Collision zones can be defined in a volume surrounding fragile equipment to ensure that no other part of the robot will crash into it.

The ROS framework includes powerful visualization and simulation tools. This saves costs and increases safety, because it enables developers to test their programs on computers before trying it out in the real world. It is also possible to get a visual representation of what the robots different sensors *see*, thus making it easier for a human to analyze sensor data.

It is important to note that ROS is not a real-time framework. It is however possible to integrate ROS with real-time code. Systems exist for transporting ROS messages in and out of real-time processes. ROS also offers "seamless integration

with the Orocos Real-time Toolkit" [31]. The Orocos project is described in the next section.

Anyone can start their own ROS code repository on their own server. No *permission* is needed. They will not be required to share their code, but may do so if they wish. Many have made their repositories publicly available, and an ever increasing amount of ROS code can be downloaded and used by others. ROS code is organized in packages and stacks, and are available under different licenses, however ROS states that they "prefer permissive Open Source licenses that facilitate commercial use of the code" [31].

Today, ROS is used in anything from hobby projects to industrial automation systems. There are already tens of thousands of users with more joining every year.

For more information on ROS, see the ros.org [31].

Figure 7.8: ROS logo [31].

7.4.2 Orocos

The Orocos project's aim is to "develop a general-purpose, free software, and modular framework for robot and machine control" [28]. The project supports four different libraries:

- The Real-time Toolkit (RTT) provides infrastructure to build robotic applications in C++.
- The Orocos Component Library (OCL) provides some control components.
- The Kinematics and Dynamics Library (KDL) allow calculations of kinematic chains in real-time.
- the Bayesian Filtering Library (BFL) provides an application independent framework for recursive information processing based on Bayes' rule.

An Orocos *application* is comprised of different software *components* forming an application-specific network. The OROCOS community has contributed a range of predefined components, but users may also build their own components using RTT.

The tools provided by Orocos make hard real-time control possible. This makes it well suited for robotic and automation applications where hard real-time is required.

7.4.3 Robot Raconteur

Designed around the idea of object oriented *service definitions*, Robot Raconteur is an open-source distributed control architecture designed for engineers wanting to control components from a high level language. The service definitions are capable of fully defining a software interface. These are provided to a client at runtime, and a high level language like Python will not require any prior knowledge of the device before connecting and interacting.

Figure 7.9: Orocos logo [28].

Robot Raconteur supports many languages like C++, Python and MATLAB, to name a few. The architecture has support for Windows, Linux and Mac OSX.

Chapter 8

Architecture

Control of a robot happens at different levels. The overall desired behavior is determined at a high level. Low level processes are then responsible for the robot's various subsystems to behave in a manner that satisfy the overall plan.

This chapter addresses the robot's control architecture, defining the software framework, control methods and the communication protocols to be used. The control and power flow of our robot is shown in Figure 8.1. The following sections will discuss the various elements of this chart.

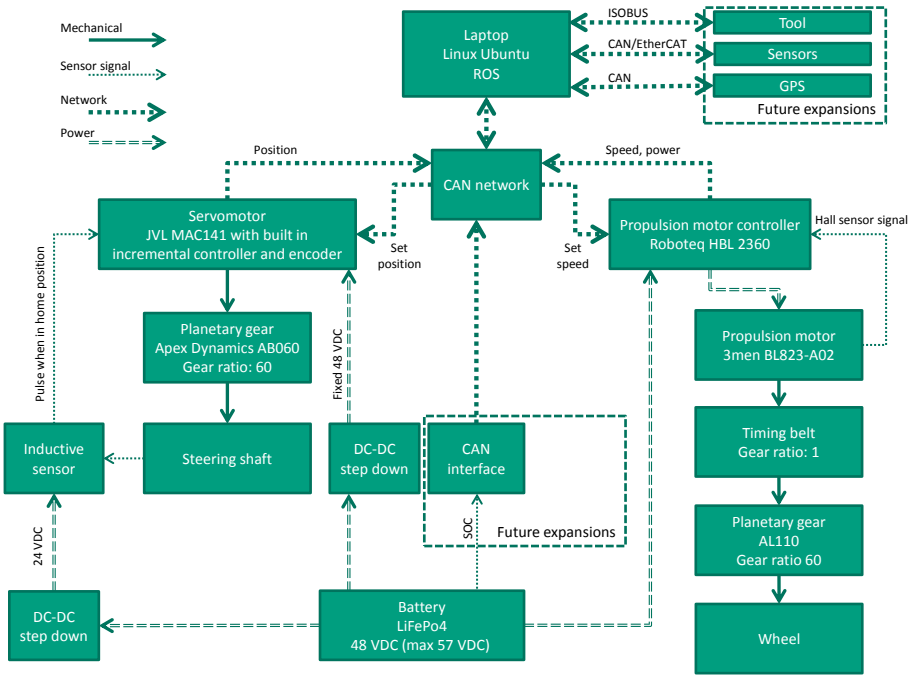


Figure 8.1: Control and power flow chart.

8.1 Robotic Framework

ROS will be used as the framework for the NMBU Robot. The main reasons for this are ROS' versatility and the huge community of users. The community has already developed a large number of packages, and new packages are being uploaded

Figure 8.2: ROS Hydro Medusa [31].

to on-line repositories every month. Packages cover anything from navigation to motor control to vision systems, and most are available under permissive open-source licenses. Using existing packages (as opposed to making new ones) saves time.

ROS has a comprehensive beginner tutorial, and packages often come with their own tutorial which should simplify the process of getting things working.

The current version of ROS, ROS Hydro Medusa, will be installed on a laptop running Linux Ubuntu 12.04 LTS, and connect to the motor control network via USB.

In the future it will be necessary integrate ROS with Rocos Real-time Toolkit to enable real-time control of various subsystems. Depending on maximum latency that can be tolerated, the kernel of the operating system may need to be modified.

8.2 Robot Communication Protocol

Controlling the propulsion and steering motors does not require transmission of large amounts of data. The same is true for a wide range of sensors that are relevant for the future development of the NMBU Robot. The important thing is that communication is robust and fast. It should also be compatible with a wide range of devices, simplifying the process of finding components when expanding the network in the future. For these reasons, CANopen was selected to be used as communication protocol for the NMBU Robot's motor control.

CANopen is used in a wide range of applications, from cars to industrial automation to robotics. There is a large selection of CANopen devices, and wrappers for USB/CAN adapters are available in a public ROS repository.

8.3 Robot Speed Control

The required speed for each motor is calculated by a node in ROS and published to a ROS topic. Another node subscribing to the same topic will put the message on the CAN bus, and the message is passed on to the two Roboteq motor controllers. Each motor controller must then ensure that the two motors it controls actually run at the requested speed. To do this it will need feedback from sensors measuring rotational speed.

The Hall sensor may be used as a feedback device, but offers rather poor resolution compared to an encoder. However, we are using a transmission with a gear ratio of 60 to connect the motor to the wheel, making the feedback resolution 60 times higher for the wheel than for the motor shaft. The 3Men propulsion motors have four rotor pole pairs (see the next chapter), which means that the motor controller receives $3 \cdot 8 = 24$ counts per revolution. The wheels have a diameter of 400 mm, thus the distance the robot travels between counts is $\frac{400\,mm \cdot \pi}{24 \cdot 60} = 0.87\,mm$. This should be sufficient for closed loop speed control.

As a backup, the wheel module has been designed for easy encoder implementation. Encoders can be fitted to the motor shaft or to the hub reduction gear input shaft. The latter alternative will require a hollow bore encoder to be fitted inside the steel frame of each wheel module between the belt pulley and the gearbox. By keeping sensor wires inside the rectangular steel tubing, they will be protected from getting caught in the rotating sprockets or the belt. Having the encoder closer to the component of interest (the wheel) can be an advantage. Alternatively, an encoder could be fitted to the the motor. The motor shaft is accessible form the rear, and an encoder can be glued in place. If this is done, it will be difficult to remove the encoder without breaking it.

We assume the mechanical connection between the motor and the wheel to be synchronous, meaning that the speed of the wheel always will equal the speed of the motor divided by the transmission ratio. If the resolution of the Hall sensors turns out to be to coarse, encoders can be installed either on the motor or on the gearbox input shaft.

Based on the sensor feedback, the motor controller will adjust motor speed, and ROS can be informed of actual motor speed. Figure 8.3 show the steering control schematically.

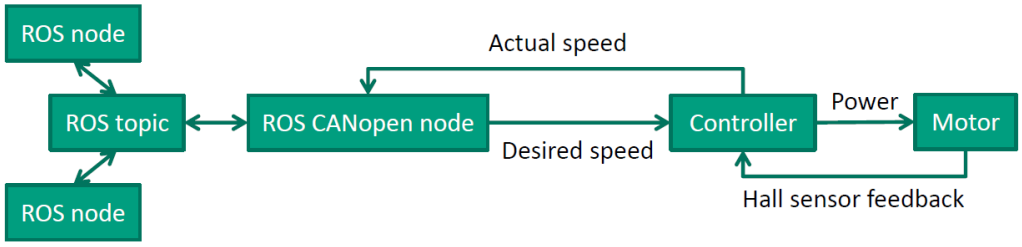


Figure 8.3: Propulsion control.

8.4 Steering Control

A node in ROS will calculate the required steering angel of each wheel module and publish these to a topic. The CAN node will translate the messages into CANopen messages, and get them on the bus. Each of the servo motor controllers will receive individual messages and adjust the steering angle accordingly. ROS can then be informed of each wheel module's actual steering angle.

Each controller can get current steering angle from its motor's built in incremental encoder. If we are to use these encoders for feedback, the motors have to home

at each power-up. Alternatively we could use an absolute single-turn encoder at the gearbox output of each wheel module. An encoder of this type cannot be connected directly to the motor controller, but must be a node on the CAN network. This means that the controller would have to receive feedback via the bus.

As absolute encoders are expensive, the built in incremental encoders will be used. This option will also save time in the process of building and setting up the robot.

We will be using inductive sensors for homing, one sensor per wheel module. The reason for this is the low price, and ease of installation. The sensor connects directly to the I/O of the motor controller, and the homing sequence is set up via the JVL MacTalk software.

Due to the gearbox, each revolution of the wheel module requires the motor to complete 60 revolutions. This must of course be accounted for by the ROS node calculating steering motor position. The motor will be requested to home at system start-up via the bus. Figure 8.4 show the steering control schematically.

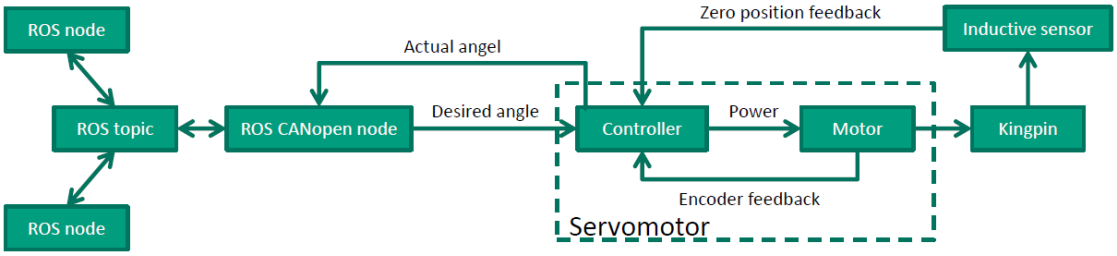


Figure 8.4: Steering control.

8.5 Future Expansions

After the robot frame is built and all devices are communicating, the work on the navigation systems can start. A system for implementing tools and a smart battery pack will also be required in the future. The design of these systems is not a part of this thesis, and will only be discussed in brief.

8.5.1 Navigation

Accurate navigation relies on information from several of the robots subsystems. While RTK-GPS (Real Time Kinematic Global Positioning System) may determine the position of the robot with centimeter precision, sensors like sonars and lasers are required to spot objects and hazards in the immediate vicinity of the robot. Most of these sensors should easily connect as nodes on the CAN network. For devices that require transmission of large amounts of data, such as a vision system with cameras, it is likely that EtherCAT (an open real-time Ethernet network) will be used.

There is a great number of relevant ROS packages available on-line. We will use existing packages in the design of the navigation system whenever this is possible.

8.5.2 Tool Communication

It is important to simplify the process of changing tools. Tool communication and power supply should therefore be provided by one single connector. The ISOBUS standard uses connectors of this type, and is also widely used in modern farm equipment and machines. It has proven to be a suitable network solution for the industry, and should also be considered for the NMBU Robot.

8.5.3 Smart Battery System

It is not possible to communicate with the robots's current batteries. The energy level of the batteries at any given time can therefor only be roughly estimated based on the battery voltage. In the future it will be necessary to implement a smart battery pack with a battery management system that can communicate with ROS. It is then possible to get the battery packs SOC and the SOH (State of Health) of each cell at any given time. The robot will know when to change energy packs and will also be able detect malfunctioning cells.

Chapter 9

Control Setup

Setting up the motor controllers, configuring ROS and making everthing communicate was a process filled with challenges, as descried in this chapter.

9.1 Matching Propulsion Motor and Controller

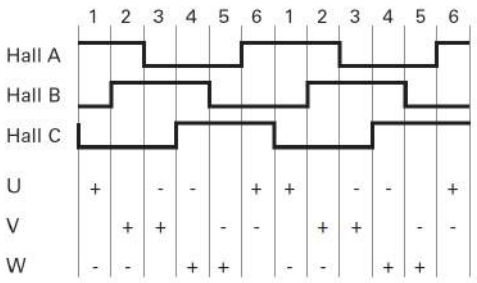
The 3men BL823 propulsion motors are among the less expensive brushless DC motors available. The data sheet supplied by the manufacturer only contains the most essential specifications like rated power and rated speed, and does not include information about the number of rotor pole pairs or the Hall sensor sequence. The process of matching the motor and controller thus involved some trial and error.

9.1.1 Controller Setup

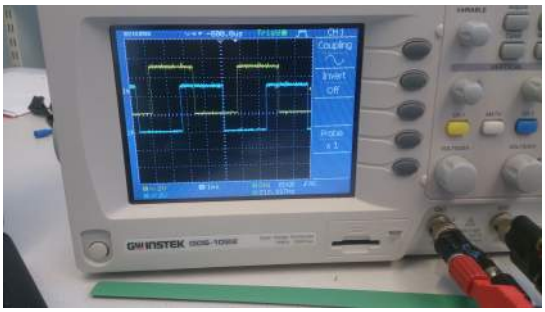
The Hall sensor power and ground wires of one of the motors were connected to one of the Roboteq controllers. Two 24 V DC power supplies in series were used to power the controller. Using a multimeter, the potential difference between one of the sensor wires and ground was monitored. For one revolution, four high and four low signals were observed, i.e. the number of pole pairs is four. The Hall sensor sequence was determined by measuring the potential difference between each signal wire and ground for one motor revolution. The power and Hall sensor wires could then be connected as indicated by the motor controller data sheets.

Roborun+, the software for setting up the controller, was downloaded from the Roboteq website and installed to a laptop running Windows 8. The controller was then connected to the laptop using USB. After some persuasion, Windows managed to find the correct driver for the controller, and Roborun+ could load the controller settings.

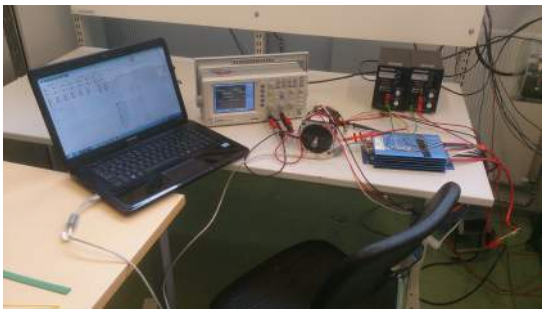
Through the Roborun+ interface, the number of rotor pole pairs was stored to the controller. To protect the motor, a 20 A limit was set for the phase current. Maximum speed and acceleration was set to 4400 RPM and 1700 RPM/s. The motor was now running smoothly in open loop mode, but the speed measured by the Hall sensor was in the opposite direction of requested. This would unable the motor to run in closed loop. Setting the number of pole pairs to minus four in the controller solved the problem.



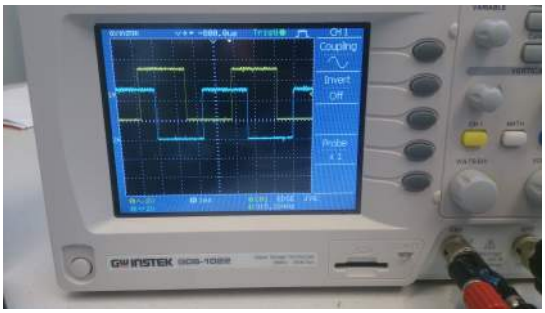
(a) Hall sensor sequence according to the Roboteq manual[30]



(b) Blue:A, Yellow:B.



(c) Test setup.



(d) Blue:A, Yellow:C

Figure 9.1: Studying Hall sensor signals. The oscilloscope show the mirror image of the sequence depicted in the controller manual due to the direction of rotation.

9.1.2 Propulsion Control Loop

In open loop mode, the supplied power is set manually. If more load is applied to the motor shaft, the motor slows down to increase torque. The Hall sensor signals are only used to ensure that the motor windings are energizing in the correct sequence, and not for regulating the motor speed.

The attempt of running the motor in closed loop speed mode using the Roboteq default PID parameters and the Hall sensor ensemble as feedback device, resulted in a jerky shaft motion before the motor stalled. Via some trial and error, new PID parameters were set, and the motor was able to run smoothly in the positive direction in closed loop speed mode

The motor did not run smoothly in the negative direction. It would be OK for a while, but then the RPM would make a sudden jump for a fraction of a second. As both the motor phase wires and the Hall sensor wires ran through the same cable, we suspected electrical noise, but separating the wires did not solve the problem. The attempt of shielding the sensor wires with an aluminum shield connected to ground did not work either. As Figure 9.1 show, the Hall sensor signals were studied using an oscilloscope. Well defined square waved signal were observed in both closed and open loop for all three sensors. The Roboteq customer support was contacted. To analyze the problem they required test data from the motor starting from zero and accelerating to different speeds in both directions, open loop. Tests were run and *Motor Command* (set point motor power), *Motor Power* (actual motor power) and *Hall Speed* (motor RPM) were recorded. Roboteq could not find the problem, but advised us to try different controller firmware versions. A range of different firmware were tested, but the problem remained unsolved.

There were also some problems related to stopping the motor in closed loop speed mode. The resolution of the Hall sensors was too coarse to accurately determine the speed of the motor shaft when running at very low speeds. For every signal it received from the sensors, the controller seemed to overcompensate by setting up a large breaking torque, forcing the motor to change the direction of rotation. When asked to stop, the shaft would slow down, and then start changing direction at very high frequency (many times per second), oscillating about one of the shaft positions where a Hall sensors would change its output. This caused the motor to vibrate.

The problem could not be solved by adjusting the PID parameters, but by changing the *Loop Error Detection* to *250ms @ 10% Error*, the motor would come to rest in a controlled manner. What loop error detection does, is to cut the power to the motor if a sufficiently large error (set point speed - measured speed) is present for given amount of time (in our case a 10% error during a time interval of 250 ms). As soon as the motor receives a new command, the power supply is restored.

This generated a new problem. It was now possible to rotate the motor shaft by hand with speed set to zero. The motor controller would occasionally try to counteract the applied torque with small bursts of breaking torque, but not enough to stop the rotation. This is of course a problem, in that the robot main computer may *think* that the robot has stopped when in fact it may be moving.

9.1.3 Programming the Roboteq Controllers

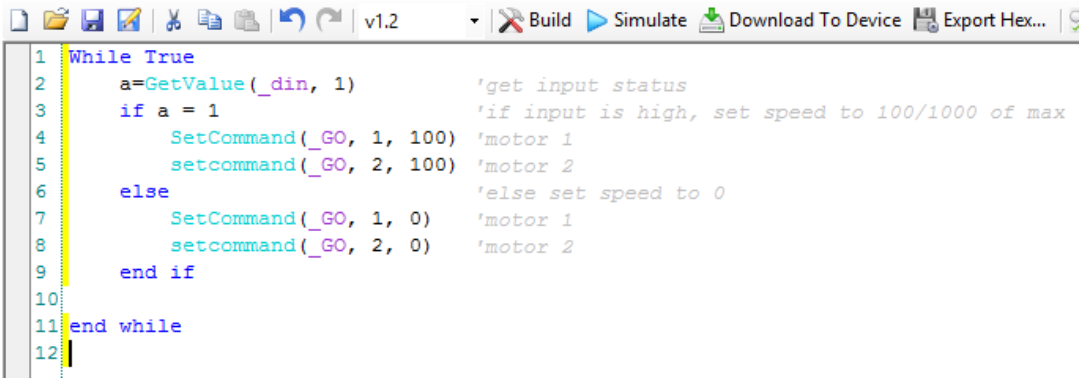


Figure 9.2: A simple MicroBasic script.

The motors and a Roboteq controller was installed to the robot (Figure 9.3). A simple program was written in "MicroBasic", the programming language used by Roborun+ (Figure 9.2). The program was first simulated on a computer, then stored to the controller via USB. The controller was configured to run the script at start-up, before it was disconnected from the computer. The controller should then drive both its motors at 100/1000 of the preset maximum speed in the forward direction whenever *digital input 1* went high. This worked perfectly. The current flow to the controller was monitored using the power supply ammeter as load was applied.

Other scripts were also tested. Among others, a script turning an output on immediately after start-up, then turning it off by giving an input high. Test were also run to ensure that the outputs responded immediately when the controller was

turned on. This is important as we will be using an output to close the relay (*RY3*) on the main contactor coil circuit in Figure 5.3(a) as described in *Part I*.

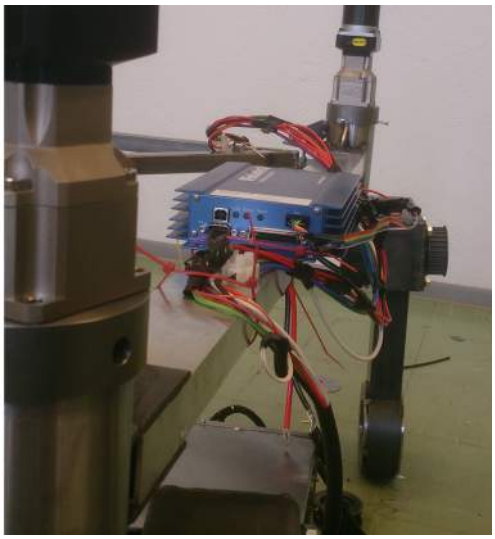


Figure 9.3: Roboteq controller installed on robot frame.

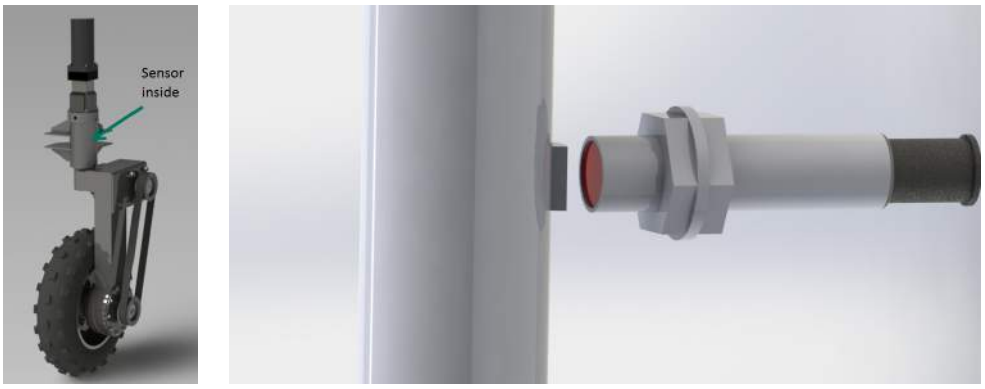
9.2 Setting up the Steering Motor Controllers

The servomotors for the steering have an integrated controller, and are overall much more sophisticated than the propulsion motors. The fact that the controllers were *custom fitted* to match the motors by the manufacturer, made it easy to get the motors operating smoothly.

9.2.1 Setup of Steering Controller and Control Loop

Each motor was, one by one, connected to a computer running Windows 7 via the computers COM port. Operating mode, homing sequence, speed and acceleration could easily be configured through the JVL MacTalk software. Using this software, the controller was set up to run in position mode, using the built in encoder for positioning. The inductive sensors were connected, and the homing sequence set to *Type 2 sensor*.

With type 2 sensor zeroing, the motor rotates until the sensor finds the reference (metal extremity) on the shaft. Then go back until the sensor goes low. This point is then set as zero position. The wheel module then rotates a specified number of counts before it stops in the forward position. For safety, the wheel module will encounter a mechanical barrier just before reaching the backward position. As the sensor goes high before, and stays high until this barrier is reached, collisions should not occur (if the sensor is high at power-up, it will rotate away from the barrier until it goes low, and set zero position) Figure 9.4 show the sensor and sensor placement.



(a) Sensor placement.

(b) Inductive sensor and steering kingpin (exaggerated distance between kingpin and sensor).

Figure 9.4: Inductive sensor for zero search.

9.3 Network Setup

Both the Roboteq and JVL controllers are made to work with CANopen. While the Roboteq connects to the bus via a 9-pin D-sub, the JVL controller features 5-pin M12 connectors. To get all controllers on one network, cables had to be custom made. Electro Drives supplied cable and M12 connectors, while D-sub T-connectors were ordered on-line from RS Components. All connectors featured screw contacts which simplified the cable assembly process. Figure 9.5(a) show the network schematically while Figure 9.5(b) and 9.5(c) show the assembly of the connectors.

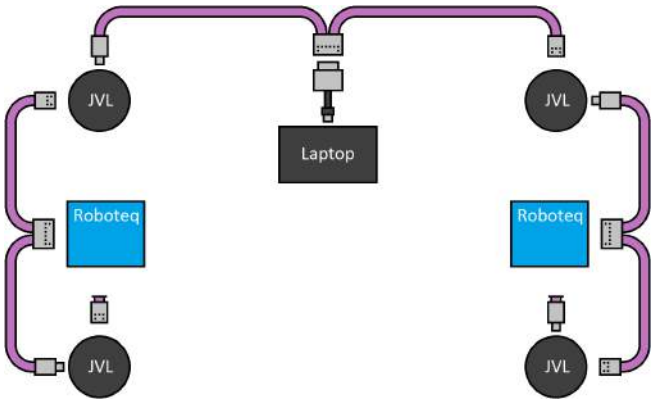
Electro Drives supplied one ready-made male M12 120 Ω connector for terminating the bus. One terminating female connector had to be assembled by hand.

The computer was connected to the network via a USB/CAN (9-pin D-sub) adapter from PEAK system. This device is supported by ROS.

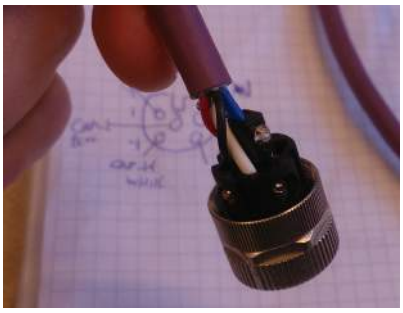
9.4 Configuring ROS

A laptop had been ordered for the robot, with delivery time estimated to a couple of weeks. Before this laptop arrived, ROS was installed on a private laptop running Linux Ubuntu. The beginner tutorial provided a quick introduction to ROS packages, topics, services and nodes. Work on setting up the robot's framework could then begin.

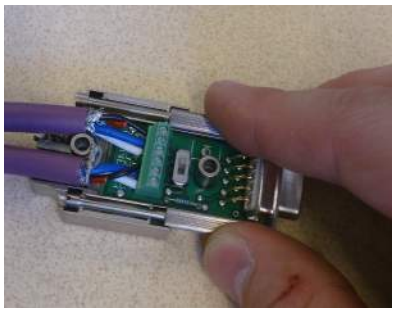
A preliminary URDF file was written to describe the robot. In the file, three links representing the three rectangular aluminum tubes of the robot frame were connected with fixed joints (joints with no degrees of freedom). Cylinders representing the wheel modules where connected to the frame with revolute joints, with upper and lower limits constraining the rotational motion about the steering axis to 180° in each direction. The wheels were represented by cylinders and connected to the wheel module with revolute joints. Figure 9.6 show parts of the URDF file and a visualization of the resulting robot using Rviz, a 3D visualization tool for ROS.



(a) CAN network.



(b) M12 connector.



(c) D-sub connector.

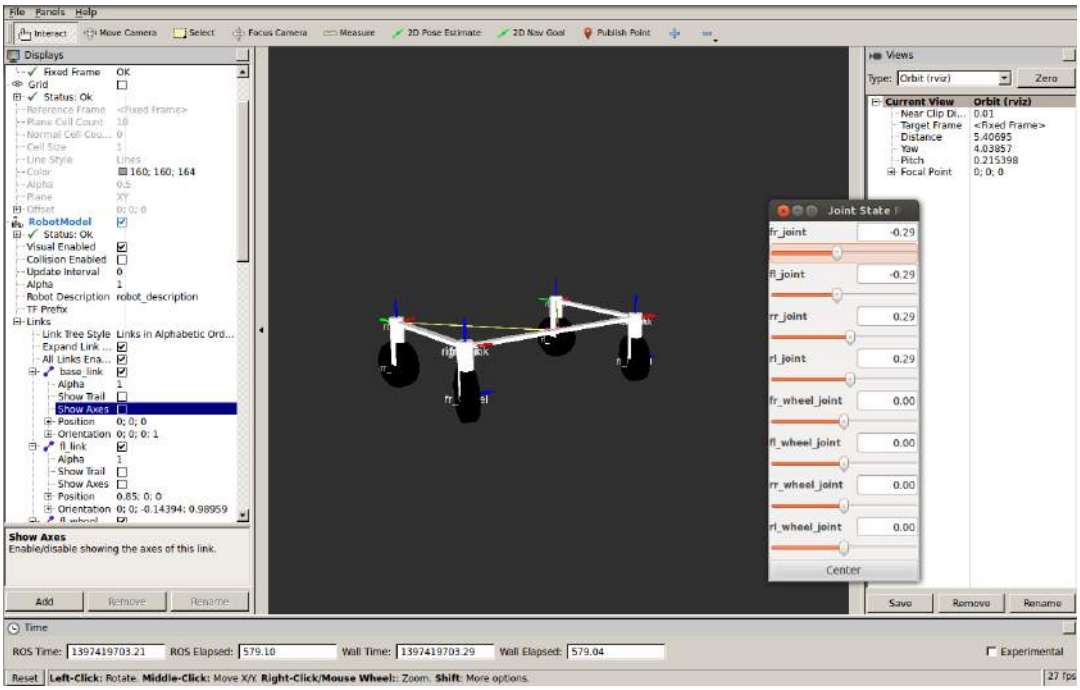
Figure 9.5: Assembling the CAN network.

```
  <link name="rl_link">
    <visual>
      <geometry>
        <cylinder length="0.4" radius="0.06"/>
      </geometry>
      <origin rpy="0 0 0" xyz="0 0 -0.15913"/>
      <material name="white">
        <color rgba="1 1 1 1"/>
      </material>
    </visual>
  </link>

  <joint name="rl_joint" type="revolute">
    <axis xyz="0 0 1"/>
    <limit effort="1000.0" lower="-3.14" upper="3.14"
velocity="0.5"/>
    <origin rpy="0 0 0" xyz="0.85 1.1  0"/>
    <parent link="base_link"/>
    <child link="rl_link"/>
  </joint>

  <link name="fr_wheel">
    <visual>
      <geometry>
        <cylinder length=".1" radius="0.2"/>
      </geometry>
      <material name="black">
        <color rgba="0 0 0 1"/>
      </material>
    </visual>
  </link>
```

(a) A small part of the URDF file.



(b) The robot visualised in Rviz.

Figure 9.6: Describing the robot with a URDF file in ROS.

Chapter 10

Conclusion, Part II

The attempt of using only Hall sensors for propulsion feedback seems to have failed. Activating *loop error detection* is required to make the propulsion motors come to rest when speed is set to zero, but then it is possible to move the robot without the controllers detecting the movement. This is not acceptable.

The motors also refuse to run smoothly in the reversed direction, closed loop speed mode. Examining the feedback sensor output using an oscilloscope did not reveal any faulty sensors or wiring, and in open loop the motor windings are energized in the correct sequence without any mishaps. As there are no issues when running the motors in open loop, the problem is likely to lay within the feedback loop, or the controller firmware. It is peculiar that the problem occurs in only one direction. The only difference between running the motor forward and backwards is that the Hall sensor sequence is reversed (see Figure 9.1).

Although a bit disappointing, the Hall sensors *failure* did not come as a big surprise. Hall sensors have only a fraction of the resolution compared to the optical incremental encoders used in similar closed loop applications. Using such course feedback as the Hall sensors provide was seen as experimental from the start, and the wheel modules were therefor designed with possible future encoder implementation in mind.

An encoder has been ordered from ELFA Distrelec for testing. If this sensor solve the problems regarding speed control, three more will be purchased. The encoders yields 500 pulses per revolution. As one pulse referrers to one period of the feedback signal, the roboteq will receive $4 \cdot 500 = 2000$ counts (transitions between high and low) per revolution from the two output channels, A and B. That is 1976 more counts per revolution than the Hall sensors. This should be sufficient to solve the problems at low and zero speed. Hopefully it will also allow the motor to run smoothly in the reversed direction.

Programming the Roboteq controller and making use of the controllers inputs and outputs has been successful. The controllers built-in microcomputer is a powerful feature that can be useful in the future.

The next phase of the project will involve setting up ROS to communicate through CANopen, and making simple programs to get the robot moving. When this is done, sensors can be fitted so that the robot can start interacting with its environment. From there, it is all about getting the robot to preform tasks in the field.

Bibliography

[1] Albright. *SW8O Series of D.C. Contactors*. 2014. URL: http://www.thetoolboxshop.com/user/pdf-documents/sw80-catalogue.pdf.

[2] Amazonen-Werke. *BoniRob field robot establishes the basis for future agricultural technology*. 2014. URL: http://info.amazone.de/DisplayInfo.aspx?id=14033.

[3] CAN in Automation. *CANopen*. 2014. URL: http://www.can-cia.org/.

[4] Rockwell Automation. *Drive and Motor Basics*. 2014. URL: http://www.ab.com/support/abdrives/documentation/techpapers/DriveMotorBasics01.pdf.

[5] Bent S. Bennedsen. "Selvkørende robotter gødsker og sprøjter". In: (2009).

[6] Fredrik Blomberg. MA thesis. Norwegian University of Life Sciences, 2014.

[7] Encyclopædia Britannica. *On-line Dictionary*. 2014. URL: http://corporate.britannica.com/.

[8] Reston Condit. *Brushed DC Motor Fundamentals*. 2010.

[9] Steve Corrigan. *Introduction to the Controller Area Network (CAN)*. 2008.

[10] Lubrication Engineers Technical Department. *Hydrostatic transmissions*. 1996. URL: http://www.le-international.com/uploads/documents/075_Hydrostatic%20Transmissions.pdf.

[11] Apex Dynamics. *AB/ABR Series High Precision Planetary Gearboxes*. 2014. URL: http://www.apexdyna.com/download/catalog/ABABR-Eng.pdf.

[12] Apex Dynamics. *AL/ALR Series High Precision Planetary Gearboxes With Adapting Timing Belt Pulley*. 2014. URL: http://www.apexdyna.com/download/catalog/AL-ALR-Eng.pdf.

[13] Gopinath and Mayuram. *Machine Design II, Module 2 - GEARS Lecture 1 - INTRODUCTION*. 2014. URL: http://nptel.ac.in/courses/IIT-MADRAS/Machine_Design_II/pdf/2_1.pdf.

[14] Tommy Gravdahl and Pål Johan From. *Innføring i Dynamikk og Reguleringsteknikk*. 2013.

[15] Heinzmann. *PMS series product catalogue*. 2014. URL: http://www.heinzmann.com/jdownloads/electric-and-hybrid-drives/catalog-parts/CAT_ED_part_Synchronous-Motors.pdf.

[16] IXXAT. *CANopen basics*. 2014. URL: http://www.canopensolutions.com/english/about_canopen/about_canopen.shtml.

[17] Jack Johnson. *Understanding hydrostatic transmissions*. 2010. URL: http://hydraulicspneumatics.com/200/TechZone/HydraulicPumpsM/Article/False/86140/TechZone-HydraulicPumpsM?page=2.

[18] JVL. *The MAC motor. AC-servo motor with Integrated driver MAC50, 95, 140 and 141*. 2014. URL: http://www.jvl.dk/files/pdf-1/datasheets/ld0043gb.pdf.

[19] Kongskilde. *Concept machine - new autonomous agricultural platform*. 2013. URL: http://www.kongskilde.com/in/da/News/Year%202013/09-09-2013%20-%20New%20automated%20agricultural%20platform%20-%20Kongskilde%20Vibro%20Crop%20Robotti.

[20] Giorgos Lazaridis. *How DC Motors are made and how they work*. 2010. URL: http://www.pcbheaven.com/wikipages/How_DC_Motors_Work/.

[21] Store Norske Leksikon. *On-line Dictionary*. 2014. URL: http://snl.no/.

[22] John B. Liljedahl et al. *Tractors and their power units*. 1989.

[23] Tommy Ertbølle Madsen and Hans Lavdal Jakobsen. "Mobile Robot for Weeding". MA thesis. 2013.

[24] Fredrik Meltzer. MA thesis. Norwegian University of Life Sciences, 2014.

[25] Maxon Motor. *On-line Product catalogue*. 2014. URL: http://www.maxonmotor.com/maxon/view/catalog/.

[26] Keith Pazul. *Controller Area Network (CAN) Basics*. 1999.

[27] Robot Platform. *Knowledge, Learning and Reasoning*. 2014. URL: http://www.robotplatform.com/knowledge.html.

[28] The Orocos Project. *Official Website*. 2014. URL: http://www.orocos.org/.

[29] The SAVAGE Project. *ZEUS*. 2014. URL: http://www.savage.gr/.

[30] Roboteq. *HBL23xx Data Sheet*. 2014. URL: http://www.roboteq.com/index.php/docman/motor-controllers-documents-and-files/documentation/datasheets/hbl23xx/59-hbl23xx-datasheet/file.

[31] ROS. *Official Website*. 2014. URL: http://www.ros.org/.

[32] RoyMech. *Timing belts*. 2014. URL: http://www.roymech.co.uk/Useful_Tables/Drive/Timing_belts.html.

[33] Paul E. Sandin. *Robot Mechanisms*. 2003.

[34] ON Semiconductor. *DC Motor Driver Fundamentals*. 2014. URL: http://www.onsemi.com/pub_link/Collateral/TND6041-D.PDF.

[35] Shibaura. *On-line catalogue*. 2014. URL: http://www.shibaura.com/index.php.

[36] Siemens. *On-line Motor page*. 2014. URL: http://www.industry.siemens.com/drives/global/en/motor/pages/default.aspx.

[37] Rubens A. Tabile et al. "Design and development of the architecture of an agricultural mobile robot". In: (2011).

[38] Geir Terjesen. *Grunnlag drivkraftteori TMP270*. 2014.

[39] Dave Walton. *Martin Sprocket & Gear, Inc. - Gear Manual*. 2014. URL: http://www.martinsprocket.com/docs/default-source/brochures---gears/martin-gear-manual.pdf?sfvrsn=12.

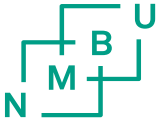
[40] Merriam Webster. *On-line Dictionary*. 2014. URL: http://www.merriam-webster.com/.

[41] Farmers Weekly. *Is farm machinery getting too heavy?* 2014. URL: http://www.fwi.co.uk/articles/06/02/2014/143174/is-farm-machinery-getting-too-heavy.htm.

[42] R. A. Wittren. "Power Steering for Agricultural Tractors". In: (1975).

[43] J. Y. Wong. *Theory of ground vehicles*. 2008.

[44] Padmaraja Yedamale. *Brushless DC (BLDC) Motor Fundamentals*. 2003.



Norwegian University
of Life Sciences

Postboks 5003
NO-1432 Ås, Norway
+47 67 23 00 00
www.nmbu.no