

Mark II

Design and Control of an Autonomous Mobile Agricultural Robot

Master Thesis Project



Kristoffer Amtoft
Søren Bæk Jensen



MARK II

Theme: Autonomous robot navigation through a corn field

Synopsis:

Type:	Semester Report
Semester:	10 th VT/EMSD
ECTS:	30
Project period:	3/2/2011-31/5/2011
Project group:	33D
Supervisors:	Ewa Kolakowska Morten Kristiansen
Prints in circulation:	5
Pages excl. Appendix:	95
Pages incl. Appendix:	120
Enclosed:	1 DVD

This project documents the development of an autonomous mobile robot, designed to compete in the Field Robot Event. The robot is to drive through a corn field without knocking down plants. A platform and components required for the robot are chosen, and the robot system is assembled. A dynamic vehicle model is developed on the background of research papers. An output, similar to data measurements obtained using a laser range scanner, is made using the dynamic vehicle model. A motor model is developed and verified. A navigation algorithm is developed enabling the robot to find its way through a corn field passage. A controller is developed for the autonomous robot. This hybrid system controller is formed in cascade with a PID motor controller. The motor controller is designed using the models developed for the system. From knowledge of the system, a fuzzy logic controller is developed that utilizes the navigation algorithm. The fuzzy logic controller is made so it could compensate aggressively when a large angular error occurs and vice versa. Two other controllers are described that are to be used by the hybrid controller as well. When the controllers are installed the robot will be ready for the Field Robot Event.

Kristoffer Amtoft

Søren Bæk Jensen

By signing this document, each member of the project group certifies that all have contributed equally in the writing of this project and thereby jointly being liable for the contents of this report.

Preface

This is a report by Kristoffer Amtoft and Søren Bæk Jensen, project group 33D. The project period concludes the 10th semester *EMSD* and *VT* respectively from February 3rd 2011 to May 31st 2011. The project theme is *Autonomous robot navigation through a corn field*.

Guide

This report uses the Harvard system of referencing. In context this appears as the authors surname and year of publishing in square brackets, e.g. [Phillips, 2000]. If the year of publishing is uncertain or unknown this will be followed by an asterisk, e.g. [Phillips, 2000*]. The references cited are found in detail in the *Bibliography* sorted alphabetically by authors surname. In the beginning of each chapter a table of content can be found, if it is applicable to the chapter.

Appendices are generally referred to as e.g. *appendix A.2*. A *PDF* version of the report and enclosures are found on the enclosed DVD. Models, data sheets and can be found on the enclosed *DVD*.

Contents

1	Introduction	1
1.1	The Field Robot Event	2
1.2	Previous Work	4
2	Thesis Statement	5
3	Hardware	7
3.1	Survey of Competitors	7
3.2	The Platform	8
3.3	Sensors	14
3.4	Printed Circuit Boards	16
4	System Introduction	19
5	Dynamic Vehicle Model	23
5.1	Observations	23
5.2	Kinetic Model Of A Skid Steered Robot	25
5.3	Skid Steering in 4-Wheel-Drive Electric Vehicle	33
5.4	Verification of Model	37
6	Motor Model	49
6.1	Armature Resistance	50
6.2	Motor Constant	51
6.3	Armature Inductance	51
6.4	Mass Moment of Inertia	53
6.5	Motor Model Block Diagram	56

6.6	Motor Model Verification	58
7	Navigation	61
7.1	Basic Navigation Principles	61
7.2	Obstacle Avoidance	63
7.3	Vector Field Histogram	64
8	Controller Design	69
8.1	Motor Controller	70
8.2	Navigation Controller	79
9	Conclusion	89
10	Discussion	91
	Bibliography	93
I	Appendix	97
A	DVM Equations	99
A.1	DVM Equations B3	99
A.2	DVM Equations B2	99
A.3	DVM Equations B1	100
B	Motor Model	101
B.1	Unit Comparison of Machine Constant	101
B.2	<i>RL</i> Circuit Analysis	101
B.3	Motor Model Step Inputs	102
C	Experimental Verification	105
D	<i>mark I</i> navigation principle	109
E	Hybrid Controller	111
E.1	Step Responses using PID	111

F Wiring Schematic	113
List of Figures	115
I Abstract	119

Introduction¹

1.1 The Field Robot Event	2
1.2 Previous Work	4

Ever since the 70's there has been an increasing focus on the environment. We realized that the way we treat this planet will impact future generations. This has led to many environmental improvements, but there is still a long way to a fully sustainable earth. Whenever a new technology emerges, that has the potential to reduce the environmental impact of an activity, it often attracts political and investor interest. This is because it is often expensive or has dire consequences to modern society, to reduce an environmental impact with conventional technology.

A present environmental issue is the overuse of pesticides and organic nitrogen in agriculture. The pesticides are sprayed onto the entire field, but only the unwanted vegetation is affected while the crop remain unharmed. While the crop is unharmed by the surplus pesticides, it is still polluted and require treatment before further processing. The surplus pesticides may also leach into the ground water or streams. The problem is similar with surplus nitrogen if the soil is unable to absorb it all. Different political initiatives has been undertaken to reduce the use of pesticides and nitrogen. In 2009 a proposal called *Grøn Vækst* (Green Growth) was adopted in Denmark. Among other things this proposal seeks to reduce the harmful effects from pesticides and nitrogen in agriculture, by increasing the pesticide tax and regulating the use of organic nitrogen so more is used for biomass [Miljøministeriet, 2010]. This has met increasing resistance because this results in a reduction of the crop yield thus reducing the competitiveness of the farmer.

The loss of competitiveness is a major issue in Denmark because the agriculture, like all industries, is threatened by the high wage level. Denmark also has a more restrictive policy toward the kinds of pesticides allowed [Landbrugsavisen, 2006]. In agriculture, the primary way of increasing efficiency is to make bigger and more powerful machines, that can handle a larger area [Ingeniøren, 2008]. But the machines have now reached a size where the soil is compacted by the weight of the machines and therefore require further treatment to prevent loss of fertility. The threat of loss of competitiveness due to restrictions, high wage level and soil compaction paves the way for new technologies.

Automation is already an integral part of agriculture. Milking robots, cleaning, sorting and packaging of vegetables, feeding of animals are examples of automated processes. In the field the combine is used to harvest and sort different corn, but it still require an operator. The operator in the combine is mainly there to compensate for all unknown influences in an outdoor environment. The unknown influences in an outdoor environment is the main issue in automating the field work in agriculture.

Precision farming is a term used for farming where the use of seeds, fertilizer and pesticides is optimized by the use of positioning technology. The technology used includes GPS, RTK (Real Time Kinematics) and different trademarked technologies. The idea is to minimize the waste of the aforementioned products, and optimize the output of the crop by removing overlap and skip. This precision can be achieved by using the position feedback from the RTK system. Manufactures of agriculture machinery have already released commercial products for precision farming. John Deere has developed a system for tractors called the iTEC Pro-system. This system is capable of automatic adjustment of the implements and can be programmed by the farmer. This can be combined with AutoTrac for automatic guidance of the vehicle [Ingeniøren, 2008]. New Holland and Case IH have similar commercial products. However while these systems have the potential to reduce the cost of farming, soil compaction and the excess pesticide and nitrogen are still unsolved problems.

Different solutions have been proposed to reduce the usage of pesticides. Most proposals revolve around a more focused targeting of the weeds. This includes using vision to locate weeds and then use pesticides in this area or target the growth center with a laser [Henrik Skov Midtiby, 2011]. Another project uses a helicopter to locate the weeds and then send a signal to a mobile ground robot that treat the weeds [Emil Fibiger Petersen, 2010].

To sum up to problems and solutions:

High Wage Level More automation.

Surplus Pesticides Focused targeting of weeds.

Soil Compaction Lighter machinery.

A possible overall solution is to make a lightweight autonomous mobile robot that can navigate in the field, locate weeds and treat it. This reduces the expenses to wages, soil compaction is no longer an issue and the focused targeting of weeds reduces the amount of excess pesticides. To create interest and increase development of autonomous outdoor agriculture robots, an annual competition called Field Robot Event has taken place since 2003, and this year it will be held in Herning, Denmark.

1.1 The Field Robot Event

The Field Robot Event is a competition created with the intend of improving off-road robot building skills, and contribute to the creation of a knowledge base for intelligent navigation, sensing and acting of autonomous and off-road vehicles [FRE2011, 2011]. The Field Robot Event 2011 is an annual agricultural competition where autonomous mobile robots compete in driving through a corn field. The competition has a set of rules and limitations that participating robot are to follow in order to achieve a high ranking [FRE2011, 2011]. The competition has 4 different categories:

Basic Follow curved corn rows and turn at the end of a row.

Advanced Follow straight corn rows with missing plants and turn at the end of a row. There will be obstacles in the form of a deep tractor track, loose soil and a large rock.

Weed-it Straight rows with artificial plants. The artificial plants needs to be detected and marked physically.

Freestyle Expression of ideas and concepts with no limitations.

A schematic approximating the field that is used at the advanced category is shown in figure 1.1 [FRE2011, 2011]. In the schematic are both the *Start* and the *Finish* point marked with red dots. The red dashed line denotes the route that the autonomous mobile is to follow through the corn field. In the schematic are the aforementioned obstacles marked with gray dots. The distance from the end of a corn row to the headland is given as $2.0[m]$ and the distance between corn rows is given as $0.75[m]$ [FRE2011, 2011].

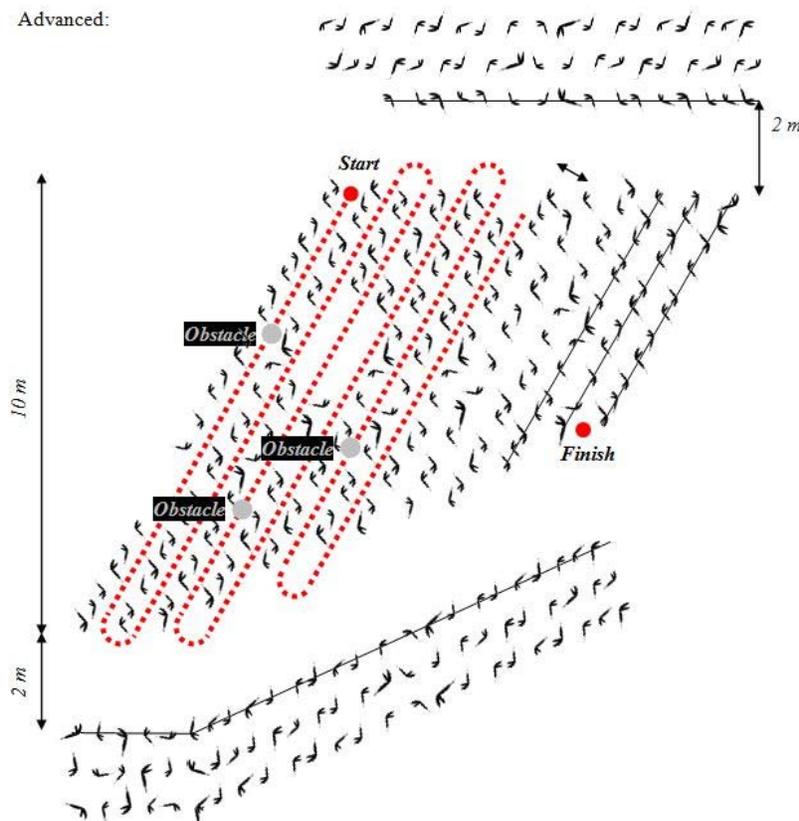


Figure 1.1: Schematic of the track used for the advanced category [FRE2011, 2011].

When participating in the advanced category, the only prohibited sensor systems are global navigation satellite systems. When participating in the competition robots are assessed from the points in the following list [FRE2011, 2011].

- Mainly rankings are given by traveled length within a given time frame.
- Advanced hardware and software solutions is awarded by a higher score.
- Using simple components instead of high end equipment will be given bonus points.
- Robots are not to be altered during the competition and 10[*min*] is given to the team to prepare the robot between challenges.
- When given permission to start, robots have to commence the task within one minute.
- Large robots, where a concern for damaging the plants is present, will start last. Damage on plants will result in penalty points.
- Human intervention during a task will also result in penalty points.

The ultimate objective of this project is to build a mobile agriculture robot to participate in the Field Robot Event 2011.

1.2 Previous Work

In the fall of 2010, a project group at Aalborg University built a robot, with the goal of participating in the Field Robot Event 2011 [Fritsen et al., 2010]. This robot was called *mark I* and was build in such a way that it is not compatible with outdoor off-road use, due to little damping in the chassis, stiff plastic wheels and no weatherproofing. Experiences from *mark I* project will be used throughout this project.

Thesis Statement

There are several issues with the way field work is undertaken in agriculture today. Problems like soil compaction, low competitiveness and overuse of pesticides. In chapter 1 it is suggested, that the solution may be to create small autonomous robots to operate in the field. The Field Robot Event is a competition that seeks to increase the interest in autonomous agriculture robots, by encouraging students and private parties to participate. With this competition as a foundation of the project, the problem can be formulated in the following questions:

- *How is an autonomous field robot, capable of navigating in a field, build?*
- *What precautions are necessary, when building a robot capable of operating in a rough, outdoor environment?*

The robot in this project is named *mark II* after its predecessor *mark I*.

When looking to participate in the Field Robot Event an agricultural autonomous robot consist of three major parts.

- 1 Building a platform that can drive and navigate in an off-road environment.
- 2 Mounting a vision camera that can distinguish between corn and weeds.
- 3 Mounting a device that can physically mark the found weeds.

Only the first major part will be in the scope of this project, though it is important that the platform, for future use, can carry and utilize both camera and weed-device.

Hardware 3

3.1 Survey of Competitors	7
3.2 The Platform	8
3.2.1 Available Platforms	9
3.2.2 Selection of Platform	11
3.3 Sensors	14
3.3.1 Encoders	14
3.3.2 Laser Range Scanner	15
3.4 Printed Circuit Boards	16
3.4.1 Computer	16
3.4.2 Micro Controller	17
3.4.3 H-Bridge	17
3.4.4 Linear Regulator	17

In this chapter, the different hardware used in the final robot design, is selected. The chapter is divided into sections regarding the platform, sensors and printed circuit boards. To get an overview of what it takes to create a successful robot, the best performing competitors from the FRE2010 are reviewed in section 3.1 *Survey of Competitors*.

3.1 Survey of Competitors

An important part of engineering is the ability to learn from other peoples experiences. The Field Robot Event 2011 in Herning is the 9th annual competition, which means that data from earlier events is available [FRE2011, 2011]. Table 3.1 shows an overview of the best performing robots of the FRE10.

Robot	Rank	Wheels	Wheel Configuration	Software/Control	Camera	Other Sensors
Ceres	1	4	4 steered 4 driven	LabVIEW	Vertical IEEE 1394 camera with parabolic mirror	
Eduro	2	3	Differential drive	Phyton running on Linux with Xenomai real-time extension.	IP camera with a fisheye lens	Sonar, GPS, compass, laser range finder
Turtle Beetle	3	4	4 steered 4 driven	C#, LabVIEW and Simulink	Logitech QuickCam	Laserscanner, ultrasonic ranger, infrared ranger, compass module, gyroscope/accelerometer

Table 3.1: Review of the best performing robots at the Field Robot Event 2010.

It is clear that the graphical programming toolbox LabVIEW is a popular choice of software. The wheel configuration with 4 steered 4 driven wheels is the most used. Because of little slipping of the wheels, this configuration is less complex to model than the skid-steering used in *mark I*. A differentially steered vehicle is less durable as wheel joints are exposed to the outdoor environment. Both fisheye lens and a parabolic mirror are ways to enhance the viewing angle of the camera. Popular sensors choices include laser range scanners and compasses. This overview is a useful tool in the decision making process when designing the *mark II* [Frank van Gennip et al., 2010],[Martin Dlouhý, 2010],[Aalto University, 2010].

3.2 The Platform

This section deals with the choice of platform for the robot. The platform consists of chassis, wheels and motors. There are different approaches to obtaining a platform:

- Continue work on the *mark I* platform.
- Build a new platform from scratch.
- Buy a new platform.

As mentioned in section 1.2, the *mark I* robot has some fundamental design flaws that makes it unfit for outdoor off-road use. It is assessed that a a modification of this platform is too extensive. Building a new platform from scratch will be time consuming, and since this is a two person project, it is decided to purchase a new platform.

3.2.1 Available Platforms

In designing the *mark II* it has been decided to acquire a complete platform and modify it, instead of building the chassis ourselves and choosing motors and encoders. To be able to utilize the work of the previous group it was decided to acquire a skid steered platform. By searching different robot vendors four research platforms have been found that are roughly in the right size and price range to function as a platform for the *mark II* robot. Outdoor robots in the research class that is necessary for FRE2011 are not easily available as most robots that can be purchased, are either toys that are too brittle and weak, or made for police and military use, and thereby completely out of the price range available for the *mark II*. In the following are four platforms, that have been deemed possible *mark II* platforms, presented, before embarking a selection process.

3.2.1.1 FORBOT Platform

Firstly the FORBOT platform, which is six-wheeled all terrain vehicle, is a candidate [Roboterwerk, 2011]. The robot has two motors, each actuating three wheels through a drive train. The chassis is made from aluminum and is manufactured to be weatherproof. The FORBOT can be seen in figure 3.1. At a glance does the FORBOT look to be both rugged and weatherproof.



Figure 3.1: Picture of the FORBOT platform [Roboterwerk, 2011].

3.2.1.2 Dr. Robot Jaguar

Secondly is the *Dr. Robot Jaguar*, which is a four-wheeled all terrain vehicle [Dr. Robot, 2011]. The robot has four motors, each actuating a wheel. The chassis is made from aluminum and plastic and is manufactured to be weatherproof and shock resistant. The *Dr. Robot Jaguar* can be seen in figure 3.2.



Figure 3.2: Picture of the *Dr. Robot Jaguar* platform [Dr. Robot, 2011].

3.2.1.3 *Gears SMP*

Thirdly is the *Gears SMP (Surface Mobility Platform)*, which is a four-wheeled all terrain vehicle [Gears Educational Systems, 2011b]. The robot has four motors, each actuating a wheel. The chassis is made from aluminum and a center crossbar made from a titanium alloy. The chassis is build in such a manner that the right and left wheel-set can rotate independently from the crossbar which fixates the main house, that contains the electronics driving the platform. The *Gears SMP* can be seen in figure 3.3.



Figure 3.3: Picture of the *Gears SMP* platform [Gears Educational Systems, 2011b].

3.2.1.4 *H-MAC*

Fourthly is the *H-MAC (Heavy Metal Articulated Chassis)*, which is a four-wheeled all terrain vehicle [Gears Educational Systems, 2011a]. The robot has two motors, each actuating a wheel-set through a drive train. The chassis is made from two connected beams and, as with the *Gears SMP*, the wheel-sets can move independently of each other, though the house is mounted on one of the moving beams and not the crossbar. The *H-MAC* can be seen in figure 3.4.



Figure 3.4: Picture of the *H-MAC* platform [Gears Educational Systems, 2011a].

3.2.2 Selection of Platform

There are different methods for concept selection. The decision can be based on tests, voting, intuition, pros and cons and decision matrices [Ulrich and Eppinger, 2004]. Voting are suited for big teams where two or more concepts are tied or the decision is not based on performance. Tests are a good base for decisions, but obviously require access to test subjects, which is not the case for the platforms deemed suitable. A list of pros and cons and decision matrices are essentially the same. However, using the list of pros and cons, only the strengths and weaknesses of each concept are considered, where the decision matrices forces the user to consider performance of all selected criteria.

3.2.2.1 Concept Screening Matrix

To select the most suitable platform a structured two-stage concept selection methodology from [Ulrich and Eppinger, 2004] is used. This method is used because the selection is based on consideration of all criteria and in this case it is not possible to test the platforms before buying them. This method uses a concept screening matrix to quickly narrow down the number of concepts and focus on the best

concepts. Stage two is a concept scoring matrix where the criteria is weighted for a more detailed review. The criteria are selected by brainstorming important features of an outdoor autonomous robot. Table 3.2 shows the concept screening matrix for the different platforms. The different criteria of the platforms are awarded +, 0 or – depending on their performance. The score is then calculated and the concepts are ranked. Hereby the selection is narrowed down to two platforms.

Selection Criteria	Concepts			
	Roboterwerk FORBOT	Dr. Robot Jaguar 4x4	GEARS SMP	GEARS H-MAC
Weather resistance	+	+	0	-
Ground clearance	-	+	+	0
Passing obstacles	0	0	+	+
Sensor stabilization	0	0	+	-
Payload	+	+	0	-
Customizable	0	0	+	0
Robustness	+	+	0	0
Price	-	-	0	+
Sum +'s	3	4	4	2
Sum 0's	3	3	4	3
Sum -'s	2	1	0	3
Net Score	1	3	4	-1
Rank	3	2	1	4
Continue?	No	Yes	Yes	No

Table 3.2: The concept screening matrix for the different platforms. The price of the FORBOT is unknown but assumed to be high due to specifications.

From the scores given in table 3.2 it is the *Gears SMP* and the *Dr. Robot Jaguar* that have been awarded the most points and are therefore subjected to further investigation.

3.2.2.2 Requirements

Before continuing with the two stage approach, the requirements of the platform are investigated. The requirements are shown in table 3.4. From evaluating the track shown in figure 1.1 and knowing that three minutes is given for each task the required average speed of the vehicle is roughly calculated to be:

$$v_{avg} = \frac{100[m]}{180[s]} = 0.55[m/s] \approx 2[km/hr] \quad (3.1)$$

This value is an estimate and has been approximated without regarding the bends when turning to the next row. Therefore is a closer estimate $3 - 4[km/hr]$ to make up for time lost in bends or if any navigation errors has occurred. Another important requirement is derived from the fact, that agricultural soil is full of small rocks and bulges therefore is ground clearance important. A ground clearance of $4 - 6[cm]$ is deemed necessary to pass small rocks and bulges without hindering the vehicle. From figure 1.1 it is also seen that there is a distance of $2[m]$ from the corn rows to the headland which requires to have a

maximum length of approximately $1[m]$. A critical parameter is the width of the vehicle, as the distance between the corn rows is $0.75[m]$, which means that a narrow vehicle is preferable as it leaves more room for navigating through a corn row passage. In the following table is a weight estimation for the equipment that is to be used on the robot. Which also is an important parameter when choosing the platform for the robot.

	Dr. Robot Jaguar	GEARS SMP
Controller	1.5[kg]	1.5[kg]
Ass. PCB and Adapters	1.5[kg]	1.5[kg]
Batteries	7.5[kg]	3.5[kg]
Extra Chassis	1.0[kg]	2.0[kg]
Camera + Mount	2.5[kg]	2.5[kg]
Sum	14.5[kg]	12.0[kg]

Table 3.3: Weight estimation of additional equipment for the *Dr. Robot Jaguar* and *Gears SMP* .

In the weight estimation it is important to emphasize that the numerical values are made from conservative intuition and not obtained through any sources. The controller board has not been decided yet and it can take the form of a standard laptop or a series of printed circuit boards. The weight of the controller depends greatly on the type that is to be chosen. Assorted PCBs and adapters are even more variable as this covers all electronics other than the controller and the sensors. It cannot be determined, at this time, how many adapters and extra circuits boards that are needed or the weight of the individual components. The weight of the batteries has been chosen by browsing catalogs of $12[V]$ batteries, as the *Dr. Robot Jaguar* motors can function with two of them in series. The extra chassis entry has been estimated by looking at the vehicle illustrations, and estimate how much extra room is required and assuming it is to be made from sheet metal. There is no numerical data for the estimation of the extra chassis. The camera that is to be used is a smart cam, that weighs very little ($< 0.3[kg]$), though the power supply and mount might be of significant weight. It is difficult to conclude upon an estimation as the one given in table 3.3, but as the weight has the possibility of being even higher than the estimated and the given payload is exceeded. On this background the *Dr. Robot Jaguar* is the better choice for an off-road mobile robot which can be seen in table 3.4.

	Dr. Robot Jaguar	GEARS SMP	Min. Requirements	Ideal Requirements
Motor				
Power	$4 \times 80[W]$	$4 \times 22.1[W]$	-	-
Voltage	24[V]	12[V]	-	-
Encoder	Yes	Yes	Yes	Yes
Speed	6.5[km/h]	2.2[km/h]	2.5[km/h]	4[km/h]
Chassis				
Weight	14.5[kg]	8.2[kg]	-	-
Payload	34[kg]	8.2[kg]	$> 12[kg]$	$> 15[kg]$
Width	0.53[m]	0.60[m]	$< 0.60[m]$	$\approx 0.50[m]$
Ground Clearance	0.088[m]	0.110[m]	0.04 – 0.06[m]	$> 0.06[m]$
Length	0.57[m]	0.60[m]	$< 1[m]$	$\approx 0.6[m]$
Wheel diameter	0.255[m]	0.178[m]	-	-
Price*	18,166[DKK]	12,208[DKK]	-	-

Table 3.4: Motor and chassis specifications for *Dr. Robot Jaguar* and *Gears SMP*.

*With the exchange rate at the 10/02/2011.

When looking at the requirements set from the track it is only *Dr. Robot Jaguar* that fulfills the minimum requirements. It is when looking at the speed of the vehicles, width and load capacity the *Gears SMP* fails, even though it has good suspension and ground clearance. Because the *Gears SMP* does not meet the requirements the concept scoring matrix becomes redundant. The final choice of platform becomes the *Dr. Robot Jaguar*.

3.3 Sensors

In this section the sensor requirements are discussed and the needed sensors are chosen and described. Sensors are important for the *Dr. Robot Jaguar* because it is to drive as an autonomous robot, and the navigation and movement will rely completely on the sensors mounted on the robot.

3.3.1 Encoders

Encoders are important to the robot system, as they are a direct source of information regarding the motors. It is important to have information about the motors, as they are the only actuators driving the robot. Each of the four motors in the *Dr. Robot Jaguar* are fitted with an encoder, by the manufacturer, and each encoder has two square wave outputs channels, *A* and *B* with a high signal value at 5[V] and a low signal value at 0[V] where *B* has a 90° phase shift relative to *A*. Figure 3.5 shows the square wave output. If the encoder is rotating clockwise *B* is lacking *A* by 90° . If the encoder is rotating counterclockwise *B* is leading *A* by 90° . Therefore only one wave is required to measure the speed, but the second wave is required to know the velocity. The encoders have five poles per revolution and the motor has a gearing of 1:19. This yields 95 waves per revolution of the wheel. To get a more accurate

reading, a measurement is taken every time there is a change in A . This yields two changes per wave or 190 *ticks* per revolution.

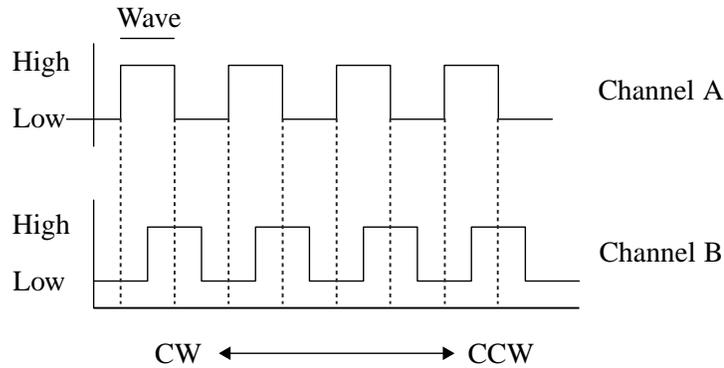


Figure 3.5: Illustration of the two square waves.

3.3.2 Laser Range Scanner

A sensor which is capable of detecting objects in front of the robot is necessary for the robot to navigate through a field with obstacles. Such a sensor was available from the project start, which was the Laser Range Scanner used in the *mark I* [Fritsen et al., 2010]. The Laser Range Scanner is manufactured by Hokuyo Automatic and is capable of scanning a 240° semicircle with maximum radius of $4000[mm]$ [Hokuyo Automatic, 2009]. Figure 3.6 shows the Laser Range Scanner. The scanner works by calculating the phase difference which ensures a minimum interference from the scanned objects color and reflectance. A complete scan takes $100[ms]$.



Figure 3.6: Picture of the Hokuyo Laser Range Scanner [Hokuyo Automatic, 2010].

Other options are available as a navigation sensor. A *SICK* Laser Range Scanner more features and is built for mobile outdoor use, but the price is much greater than for the Hokuyo Laser range scanner. Other options could be in the form of a camera with vision control or ultra sonic sensors. To implement a camera for a navigation purpose may require heavy computational power and might be difficult to

implement into a navigation control algorithm. The ultra sonic sensor option is cheaper than a Laser Range Scanner though it has a reputation of being unreliable, having poor range and influenced by ambient disturbances. Without further investigation is the Laser Range Scanner deemed a sufficient choice of navigation sensor.

3.4 Printed Circuit Boards

In this section the different printed circuit boards, required to control the robot, are described.

3.4.1 Computer

The computer is required in order to run an operating system, which can run the controller designed for the robot. Furthermore, since it is to be mounted in an off-road mobile robot it is required to be:

Sturdy A mobile robot operating in a field is required to cope with vibration and small impacts, this includes the computer controlling it.

Small The available space in the chassis is limited.

Powerful The robot may require much computing power to handle the data processing and controller.

Low power consumption Only the batteries mounted on the robot are available, so the power consumption must be low.

An inquiry, regarding computing power in a mobile robot, was sent to Kjeld Jensen [Kjeld Jensen, 2011]. Two different solutions were suggested by Kjeld Jensen [Kjeld Jensen, 2011].

PC/104 A small ($90 \times 96[mm]$) and sturdy PC with low power consumption.

Mini-ITX Board Larger ($170 \times 170[mm]$) and more powerful than the PC/104. This also has a higher power consumption.

The PC/104 standard is chosen because of its small form factor, low power consumption and sufficient available computing power. From a Danish vendor four different PC/104 platforms are available:

	PV-6070	PV-6154	PV-6151	PV-6354
Clock Frequency [MHz]	166	300	300	800
Memory [MB]	128	128	128	256
Current [mA]	1400	640	620	740
Weight [g]		76	90	90

Table 3.5: Different hardware configurations of the PC/104.

The PV-6354 is chosen because of the superior computing power, which still is a requirement. A data sheet of the chosen PC/104 can be found on the enclosed *DVD*.

3.4.2 Micro Controller

For micro controller a Arduino Mega board is used. This board is based on the *ATmega1280* processor and was available from the beginning of the project and has 54 I/O ports whereof six are interrupt and 14 can be used to generate PWM signals for the motors. This board has sufficient ports to control motors and measure speed from encoders, and will therefore function as a breakout board. A complete description of the Arduino board can be found in the data sheet located on the enclosed *DVD*.

3.4.3 H-Bridge

To control the voltage between the batteries and motors are H-bridges from Robot Power used [Robot Power, 2008]. It uses a PWM signal, generated by the Arduino break-out board, to open and close the circuit to the motor at a high frequency so the motor experiences a steady voltage, that is depending on the applied duty cycle. By applying I/O signals the H-bridge can change the polarity of the voltage sent to the motors yielding another way of motor rotation. The H-bridges chosen were available at project start. A data sheet for the H-bridges can be found on the enclosed *DVD*.

3.4.4 Linear Regulator

A printed circuit board with a linear voltage regulator has been built. The data sheet for the voltage regulator can be found on the enclosed *DVD*. When looking at the data sheet for the PC/104 board a stable 5[V] DC supply is necessary in order to make it work properly. To obtain such voltage a *LT1084CP-5* component was chosen. The component transform a supply voltage above 7[V], maximum 20[V], to yield a supply of 5[V]. To obtain a stable supply an application note in the data sheet is used. To follow the application note two capacitors are added to the circuit and a heat sink is mounted to dissipate developed heat. The heat sink is experienced to become very hot during usage and an even bigger heat sink is required if the robot is used for longer period of time. The circuit board is connected to one of the batteries, yielding a 12[V] DC supply. Also the Arduino board and encoders needs 5[V] supplies. The Arduino board is powered by the PC/104 and the encoders are powered by the Arduino board.

System Introduction 4

Having chosen the platform for *Dr. Robot Jaguar* and belonging hardware, the layout of the autonomous robot system is given in the following. Firstly is an image of the assembled system, with the lid off, presented, viewing the internal components.



Figure 4.1: Open lid image of the *mark II* .

In the center of the vehicle is a hard drive, which has a white sticker on it. Right behind the hard drive is the PC/104 computer. Adjacent to each wheel can a motor be seen, as well as the H-Bridges that can be seen as the green PCBs. In the bottom of the image can an aluminum sheet be seen, which has the function of housing the Laser Range Scanner. Right above the aluminum plate can a blue printed circuit board be seen, this is the Arduino board that is connected to both the PC/104 and the four H-Bridges. In *appendix F Wiring Schematic* the components shown in figure 4.1 are shown in a schematic form, with descriptions denoting the connections. In order to obtain a better overview is a simplified block diagram showing the elements of the robot system shown in figure 4.2. The elements in the figure are described in the following and will function as a contents list for the following chapters.

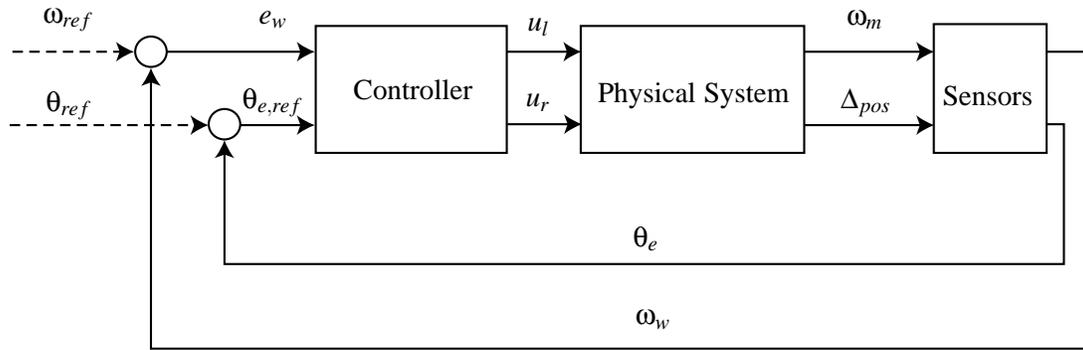


Figure 4.2: Block diagram depicting the *mark II* system.

where:

Physical System: The Physical System describes the dynamics of the system comprising the chassis, wheels, motors etc. This block both represent a dynamic vehicle model and a motor model of the robot system and the actual robot. A final model is made in order to have an environment where a controller for the robot can be developed. Before designing a controller for the system it is necessary to verify data from a final model with data measurements from test runs of the real robot. The dynamic vehicle models describing the chassis, with wheels, and is developed in chapter 5 *Dynamic Vehicle Model*. The motor model used in the final model is developed in chapter 6 *Motor Model*.

Controller: The controller of the system is software installed on the PC/104 and the Arduino board, which both are contained in the Controller block. The Arduino will control the motors according to a reference velocity generated in the controller installed on the PC/104. The Arduino controls the motors from a velocity feedback from the mounted encoders. The velocity error e_w is shown outside the controller block, but will be conducted inside the motor controller. The PC/104 will have the main controller, which generates a trajectory the robot is to follow. The trajectory is found from the data fed back from the Laser Range Scanner. The controller on the PC/104 will generate a wheel velocity reference for each motor set, which are depending on the trajectory. As the robot will encounter different situations along its path, when driving in a corn field, different controllers will be needed to handle each situation. Such a controller can be obtained through the use of a hybrid system controller. The controller for the *mark II* is developed in chapter 8 *Controller Design*.

ω_w and ω_m : ω_w denotes the angular velocity of wheels, ω_m denotes motor velocity, which is obtained through the encoders mounted on the motors. The wheel velocity is fed back to the controller block to be used in the motor controller on the Arduino board. This feedback will be a part of the controller designed in chapter 8 *Controller Design*.

ω_{ref} : ω_{ref} denotes the wheel velocity references for the motor controller. The references are based on the angular position error. If the angular position error is large the difference between the motor

velocity references will be large and vice versa. This reference will be a part of the controller designed in chapter 8 *Controller Design*.

θ_e : The angular error between the heading of the robot and the best found trajectory, is extracted from the *LRS* data using a path finding algorithm. The *LRS* will be mounted on the robot with its zero angle position reference in a frontal central position on the robot, marked with a dotted line. In figure 4.3 the angular error is shown between the yaw of the vehicle and a vector to a point on the path, the vehicle is to follow. This feedback will be a part of the controller designed in chapter 8 *Controller Design* and will be explained in further detail in chapter 7 *Navigation*.

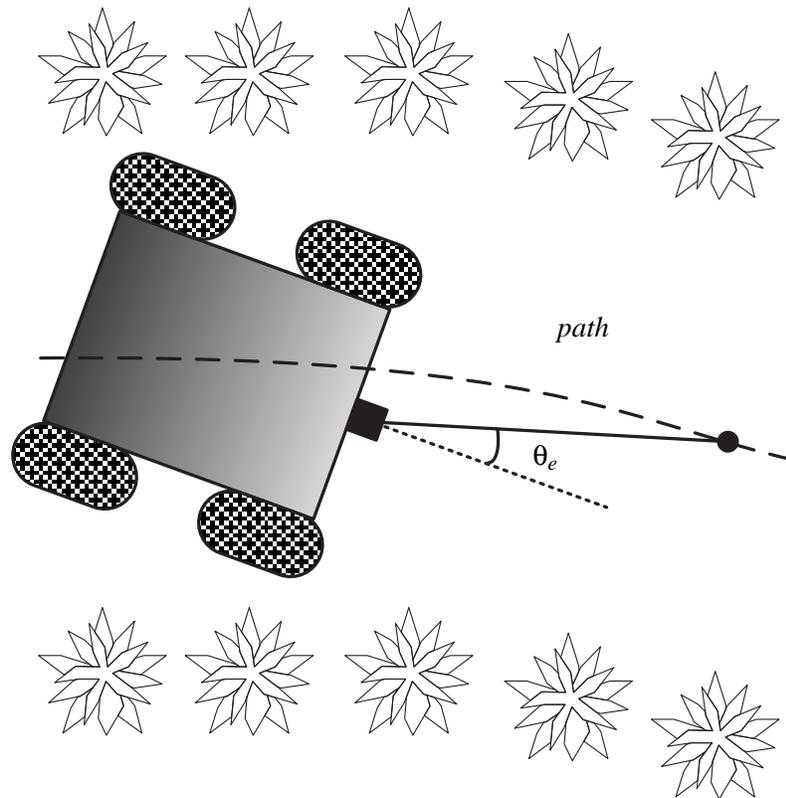


Figure 4.3: Illustration of the angle θ_e .

θ_{ref} : θ_{ref} is the angular position reference between the vehicle and the path found by the *LRS*. Set to zero in order to follow the path as close as possible. This reference will be used in the controller described in chapter 8 *Controller Design*

u_l and u_r : u_l and u_r are the voltage outputs from the motor controller to the H-bridges connected to the left and right motors according to index.

e_w : e_w denotes the wheel velocity errors when comparing velocity references and feedback velocities. Is to be utilized in motor controller.

$\theta_{e,ref}$: $\theta_{e,ref}$ denotes the error between the angular error θ_e from the *LRS* and the angular position reference θ_{ref} . For navigation the angular reference θ_{ref} will be set to zero, and the angular error will be passed directly to the navigation controller.

Δ_{pos} : Δ_{pos} is the change of position of the vehicle. This position is indirectly utilized by the *LRS* to determine θ_e .

As mentioned earlier a dynamic vehicle model of the system is to be developed. The dynamic vehicle model is verified and used as a background for controller design. The developed dynamic vehicle model is given in chapter 5 *Dynamic Vehicle Model*.

Dynamic Vehicle Model 5

5.1	Observations	23
5.2	Kinetic Model Of A Skid Steered Robot	25
5.2.1	DVM [Solc and Sembrera, 2008]	26
5.2.2	Assessment of the Built DVM	29
5.3	Skid Steering in 4-Wheel-Drive Electric Vehicle	33
5.4	Verification of Model	37
5.4.1	Design of experiment	37
5.4.2	Experiment setup	43
5.4.3	Results	45

In this chapter a Dynamic Vehicle Model (DVM) for the robot is designed, to gather information regarding the systems to create a controller. As shown in figure 5.1, the final model comprises a motor model and a dynamic vehicle model. The input for this DVM is the rotational velocity for the left and right wheel set of the robot, and the output is motion in a global coordinate system. First some general observations on skid steered vehicles are made. Hereafter two models are created based on the approached in two different articles. The best suited model is chosen and is verified experimentally.

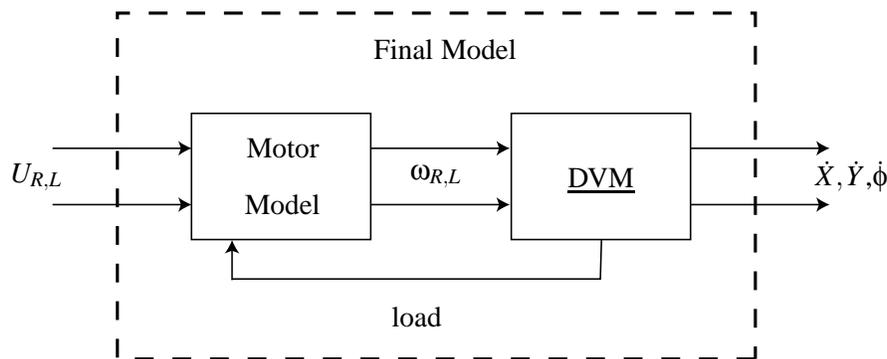


Figure 5.1: Final model with sub models.

5.1 Observations

Before looking into dynamic simulation models of the robot, the following observations have been made regarding the system at hand. The *Dr. Robot Jaguar* is a four wheel skid-steered vehicle with a motor driving each wheel, which is dictating when modeling the system. Skid-steered vehicles have three or more wheels that only have one rotational degree of freedom usually driven by a motor or a drive train.

Skid-steering is obtained by controlling the velocity of the wheels on each side of the vehicle individually, making the vehicle turn in the direction of the wheel-set with the lowest velocity. In the following a few points are highlighted regarding skid-steered vehicles.

Durability Skid-steered vehicles are often very durable and reliable with only few moving parts, which are desirable parameters when building an off-road vehicle. Skid-steered vehicles are easier to weather proof with only the wheels-axles leaving the chassis, which leaves out wheel-joints for local wheel yawing, that can be damaged by the off-road environment [Kozlowski and Pazderski, 2004].

Maneuverability A great advantage of skid-steered vehicles is that they can be highly maneuverable, i.e. by choosing equal but opposite wheel velocities, on the left and right wheel sets, the vehicle can, ideally, turn around its own axis[Shuang and Cheung, 2007].

Lateral velocity There is a downside when choosing skid-steer though, which is the phenomenon of lateral skidding of the wheels when turning the vehicle[Shuang and Cheung, 2007]. Lateral skidding is necessary when turning the vehicle due to the fixed wheels, as they have to slide sideways in order to obtain angular displacement of the vehicle. If no lateral skidding is present the robot would only be able to drive in a straight line. The lateral skidding also yields a higher power consumption as the energy used here is avoided if turning wheels were used.

The problem with lateral skidding arises when wanting to predict the angular velocity of the vehicle on an arbitrary surface, as the interaction between wheels and ground can be difficult to estimate. Estimation of parameters describing the interaction are often iteratively determined in a model, as a part of verifying the model against the real system. The vehicle will have higher or lower angular velocity than anticipated if the model is not verified against the real system, resulting in the vehicle over or under compensating, in an unpredictable manner, when correcting its path between corn rows.

In figure 5.2 can the longitudinal and lateral directions be seen relative to a robot wheel. These terms will be used throughout this chapter. This also applies for the chassis, where the longitudinal direction is the direction of the robot when driving in a straight line.

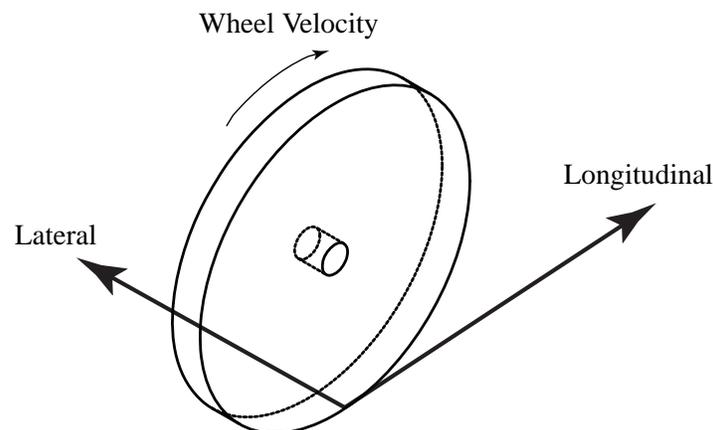


Figure 5.2: Local wheel velocities.

When the vehicle is driving in a straight line the longitudinal velocity of the vehicle can be obtained from encoders when a no slipping condition is assumed for the wheels. When the vehicle is turning, at any angular velocity, all wheels will experience both lateral and longitudinal velocities, which neither are equal to the encoder velocities.

In order to obtain an approximation of the overall movement of the vehicle, a dynamic model including the wheel to ground interaction is necessary, or at least a compensation thereof. The model is built in order to create an environment in which a controller can be designed, that can compensate for the lack of information from the wheel to ground interaction and control the vehicles onto a given path. In the past years a range of dynamic models for skid-steered vehicles have been presented in research papers. Three of these of these papers have been examined in order to determine which dynamic model is to be used to simulate the behavior of *mark II*. The first paper is [Solc and Sembrera, 2008] which is the approach, to describe the dynamics of a skid-steered system, that was used in [Fritsen et al., 2010]. The second paper is [Shuang and Cheung, 2007] which is chosen because it comprises many factors that [Solc and Sembrera, 2008] does not, and has a different approach to modeling the forces acting on the wheels. The third paper is made by [Kozlowski and Pazderski, 2004] which has a third approach, though alike [Shuang and Cheung, 2007], it ends up with a compact vectorial equation describing the dynamics of the robot. The [Kozlowski and Pazderski, 2004] paper has been used to create a simulation model, though the built model did not work and therefore has this paper been omitted from the project report. At a glance the [Kozlowski and Pazderski, 2004] model seemed very promising, and is worth another try, if another model is sought. The two papers, [Solc and Sembrera, 2008] and [Shuang and Cheung, 2007], containing the dynamic models that have been examined, can be found on the enclosed *DVD*. Other than the three previously mentioned papers, has other papers been read on the subject, but the two chosen to be examined, are chosen as they seemed useful and the approaches used seemed different at a glance. From other papers it is noticed that most skid-steered vehicle papers has an empirical approach, which have been omitted from this project since it requires many laboratory hours.

5.2 Kinetic Model Of A Skid Steered Robot

The first paper that has been examined is *Kinetic Model Of A Skid Steered Robot* by [Solc and Sembrera, 2008]. The vehicle utilized in the paper is a mobile robot much alike *Dr. Robot Jaguar*, though it has only two electrical motors, each actuating a drive train connected to two wheels. This is different from *Dr. Robot Jaguar* though effectively it will be equal, as both magnitude and direction of the motor signals will be equal for motors on the same side when actuating the *Dr. Robot Jaguar*. A DVM utilizing forces and masses is formed, which is necessary to estimate the lateral slip, [Solc and Sembrera, 2008].

The model created by [Solc and Sembrera, 2008] differs from the other papers concerning skid-steered vehicles, by obtaining forces acting in the lateral and longitudinal direction from wheel velocities, from mounted encoders, and not by using the torques created by the electric motors. It is these forces that are used along with the mass and mass moment of inertia to obtain lateral, longitudinal and angular accelerations of the vehicle. The modeling done to find these forces is what makes the [Solc and Sembrera,

2008] model differ, and which is why it is explained in the following. On the background of the dynamic model to be described a model has been built by the project group, completely in in thread with [Solc and Sembrera, 2008], which is to be considered as the model used in the final model.

5.2.1 DVM [Solc and Sembrera, 2008]

In figure 5.3 is the [Solc and Sembrera, 2008] DVM shown on block diagram form. The DVM created by the project group has the same form as presented in figure 5.3, and the parameter naming and nomenclature used is taken from the [Solc and Sembrera, 2008] paper. The notations used in the figure are described in the following.

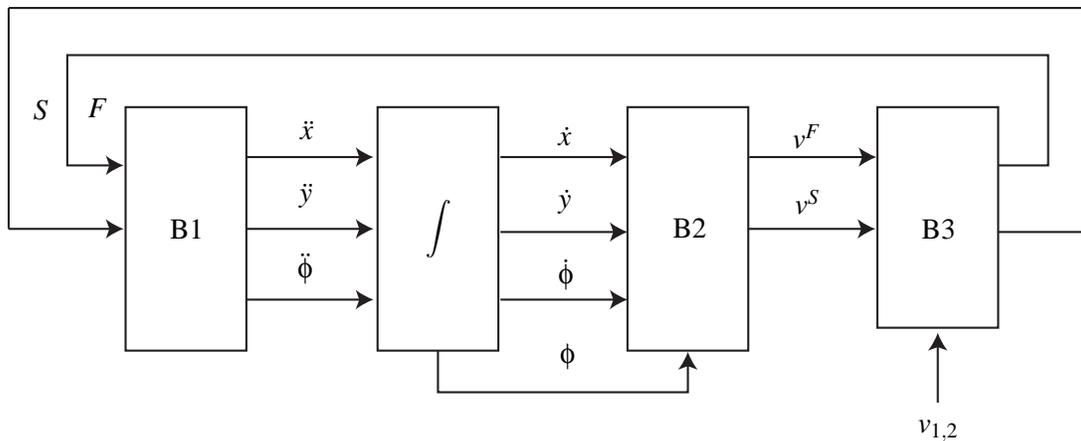


Figure 5.3: Block diagram of dynamic vehicle model [Solc and Sembrera, 2008].

B3 Block three in figure 5.3 is the most comprehensive of the DVM blocks and is therefore explained first. Block three receives longitudinal (v^F) and lateral (v^S) velocities of the vehicle, a set from each wheel, calculated in the previous block¹. Block three also receives wheel velocities $v_{1,2}$, which are the velocities measured by the encoders. From these velocities are the total velocities of slipping for each wheel (v_f) estimated, which is found from slipping in the longitudinal direction (v_s) and velocities in the lateral direction (v^F). v_f can be thought of as a parameter for each wheel describing the velocity components not contributing to longitudinal velocity of the vehicle. The total velocities of slipping are then used in a *Coulomb model of friction* [Solc and Sembrera, 2008], which is shown in equation 5.1. The equations used in the [Solc and Sembrera, 2008] DVM block three are listed in *appendix A.1 DVM Equations B3*.

¹Notice the difference in sub and super scripts at velocity notation.

$$P = C \cdot v_f \quad \text{for } v_f \leq \frac{\mu_0 \cdot W}{C} \quad (5.1a)$$

$$P = \mu_0 \cdot W \quad \text{for } v_f > \frac{\mu_0 \cdot W}{C} \quad (5.1b)$$

Where C is stiffness of the tires, μ_0 is a coefficient of friction between tire and ground, W is the weight of the vehicle and v_f are the aforementioned total velocities of slipping [Solc and Sembrera, 2008]. P is a parameter used to calculate the magnitude of the forces of adhesion in the lateral and longitudinal directions, see equation 5.2. In equation 5.1a it is seen that P depends on the stiffness of the tires and the total velocity of slipping. This equation is only valid when the total velocity of slipping is less than or equal to the right hand term, which is a term comprised of a relation between the friction from normal force and the stiffness of the tires. When the total slipping velocity is low, the vehicle is driving in a large circle experiencing little lateral velocity, and the friction term is dependent on the tire stiffness and the total slipping velocity. Equation 5.1b is only valid when the total velocity of slipping is greater than the right hand term opposite to equation 5.1a. When the total slipping velocity is high, the vehicle is driving in a smaller circle experiencing greater lateral velocity, and the friction term is given by a constant term from the normal force and the friction coefficient.

With the P variable described longitudinal and lateral forces of adhesion are found. In equation 5.2 the longitudinal and lateral forces are given. F are the longitudinal forces of adhesion and are given by P and the ratio between the velocity of longitudinal slipping v_s and the total velocity of slipping v_f . S are the lateral forces of adhesion and are given in a similar manner, though the signage of P is negative and the ratio is given by lateral wheel velocity v^S and the total velocity of slipping v_f [Solc and Sembrera, 2008].

$$F = P \cdot \frac{v_s}{v_f} \quad (5.2a)$$

$$S = -P \cdot \frac{v^S}{v_f} \quad (5.2b)$$

As mentioned earlier, a skid-steered vehicle will drive in a large circle if the difference between the wheel sets are low and vice versa. When the vehicle is driving in a larger circle, the wheels experience little lateral velocity and the total velocity of slipping v_f is low, yielding a greater forces of adhesion in the longitudinal direction, equation 5.2a. The lateral adhesion forces become small as both the lateral wheel velocity v^S and the total velocity of slipping v_f are low, especially v^S becomes low when driving in a larger circle.

When driving in the smaller circle, the lateral adhesion forces become greater as the lateral wheel velocities v^S become greater, and the longitudinal forces of adhesion are smaller as the total velocities of slipping becomes greater.

B2 Block two in figure 5.3 calculates the longitudinal and lateral velocities used in block three. These are calculated from the input values seen in figure 5.3 which are the longitudinal velocity of the chassis (\dot{x}), lateral velocity (\dot{y}), angular velocity ($\dot{\phi}$) and the angular position (ϕ). To calculate the longitudinal and lateral velocities for each of the four wheels calls for eight equations. By assuming that the longitudinal velocities are equal for the left side wheels and equal for the right side wheels, can the four equations be reduced to two. The same can be done for the lateral velocities by assuming that the frontal wheel lateral velocities are equal, and the same for the rear wheels. The four equations calculating the longitudinal and lateral velocities can be seen in *appendix A.2 DVM Equations B2*. To keep track of the index numbering of variables is figure 5.4 given, describing nomenclature of the vehicle.

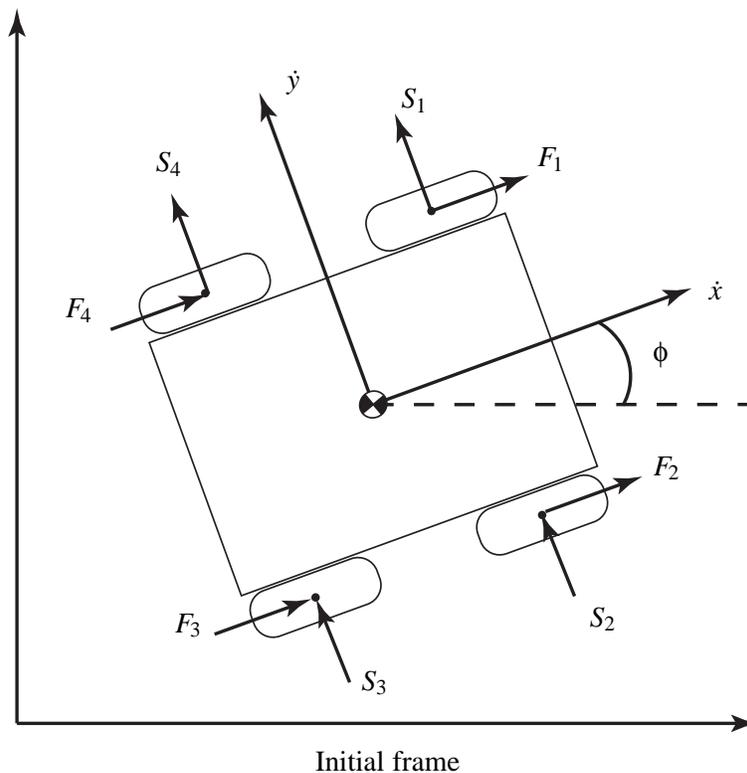


Figure 5.4: Sketch of vehicle with index numbering of the wheels.

B1 Block one receives the longitudinal and lateral adhesion forces mentioned earlier as a feedback in the model. The block utilizes these forces in force and torque equilibriums. The equilibriums, which are given in *appendix A.3 DVM Equations B1*, yield the longitudinal, lateral and angular accelerations of the chassis on the 2D surface it is assumed to be driving on. The accelerations are passed to the integrator block where the velocities are calculated and the angular acceleration is integrated twice to obtain the angular position.

The only input of the model described are the peripheral velocities of the wheel sets. The DVM built by the project group can be found on the enclosed *DVD*. In the following an assessment of the built model

is relative to the results given in the [Solc and Sembrera, 2008] paper.

5.2.2 Assessment of the Built DVM

In [Solc and Sembrera, 2008] the simulation model is verified against the robot of which it is a model of. Two test runs were made where the movement of the vehicle was measured over time for steady state wheel velocities. In the first test run the angular velocity of the vehicle was 3.2% greater than simulated and in the second test run it was 6.6% greater.

An attempt to replicate the results of the paper has been made with the built model, in order to know if the built model is built correctly. The motor model is not given in [Solc and Sembrera, 2008], therefore has the approach used in [Kozlowski and Pazderski, 2004] been used in the [Solc and Sembrera, 2008] model recreation. Parameters of the motor model have estimated values, as the motor model is only used, in this context, to maintain the steady state wheel velocities, in order to avoid step inputs that can result in unwanted model behavior in the start up period of the simulations, i.e. oscillations in vehicle movement.

The input voltage to the motor model has been fitted to give the ± 0.12 [V] utilized in the [Solc and Sembrera, 2008] 1st simulation run. The vehicle specific model parameter values utilized in the following recreation are identical to the ones given in the [Solc and Sembrera, 2008] paper. Applying the velocities used in [Solc and Sembrera, 2008] does not recreate the results of angular velocity $\dot{\phi}$ of 0.3 [rad/s] nor the turning radius R of 0.025 [m], which leads to the suspicion that the built model does contain modeling errors. The results given by the built model are 0.233 [rad/s] for the angular velocity and 0.020 [m] for the turning radius. The recreated model has been looked over many times for errors and none have been found after the simulations have been run. For a better overview have the different settings for alterations and results been given in table 5.1.

Settings	p -value	\pm Wheel Velocities v_i [m/s]	Angular Velocity ϕ [rad/s]	Turning radius R [m]
Original [Solc and Sembrera, 2008] settings	0.54	0.12	-0.30	0.025
Recreated [Solc and Sembrera, 2008] settings	0.54	0.12	-0.233	0.020
Recreated [Solc and Sembrera, 2008] Altered p	0.55	0.12	-0.233	0.025
Recreated [Solc and Sembrera, 2008] Altered v_i	0.54	0.153	-0.30	0.020
Recreated [Solc and Sembrera, 2008] Altered v_i & p	0.55	0.153	-0.30	0.025

Table 5.1: Results of recreated simulations compared to the [Solc and Sembrera, 2008] model for the 1st simulation run.

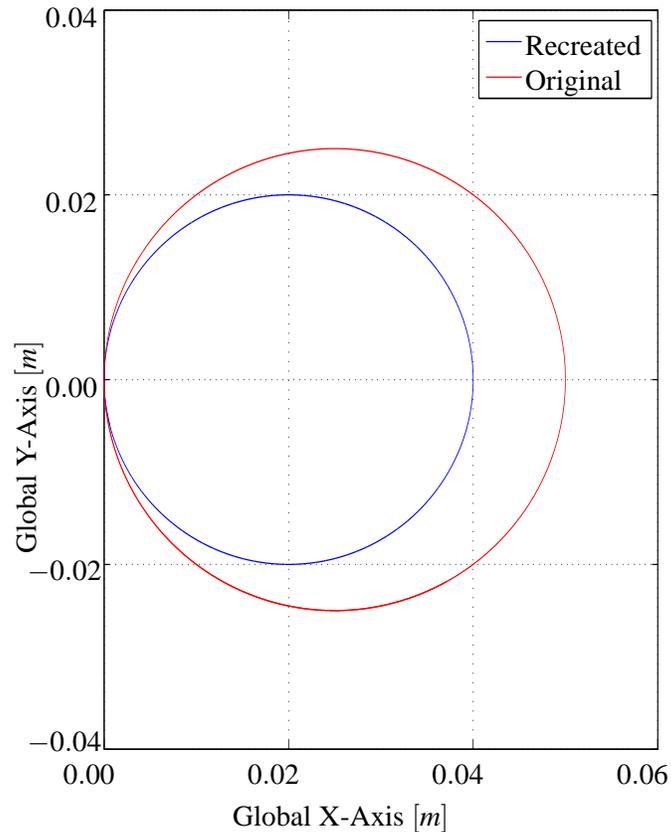


Figure 5.5: Plot of original and recreated circles of 1st simulation run.

If the geometrical ratio $[p]$, that describes the location of center of mass and the geometrical center, is altered to 0.55 instead of 0.54, it is possible to recreate the turning radius of $0.025[m]$ but the angular velocity remains at $0.233[rad/s]$. By altering the wheel velocities from $\pm 0.12[m/s]$ to $\pm 0.153[m/s]$ the angular velocity of $0.3[rad/s]$ can be recreated. The turning radius on the other hand has dropped back to $0.02[m]$, which is caused by the original p value. By running a recreated simulation using a p value of 0.55 and the greater wheel velocities of $\pm 0.153 [m/s]$, can the given [Solc and Sembrera, 2008] simulation results be replicated for the 1st simulation run. In figure 5.5 is a plot of the circles created by the [Solc and Sembrera, 2008] model and the recreated model, where both are given the same original parameter settings. It is clearly seen from the figure that the circles are not coincident, which indicates that there is an error in the simulation model. Though when altering the v_i and p values, accordingly with the table, the circles become coincident.

This implies that when using the [Solc and Sembrera, 2008] model for verification against the *mark II* system, parameters are to be tweaked in order to verify the model. When tweaking the model it is important that the tweaked parameters remain constant for a range of verification runs, to ensure that the model works with other velocities, as the model should represent all velocity scenarios for the vehicle. In [Solc and Sembrera, 2008] there is given a second, and final, verification simulation run. The second simulation run utilizes wheel velocities of $0.0[m/s]$ and $0.12[m/s]$ on the left and right wheel respectively. The results for 2nd simulation run are given as $-0.14[rad/s]$ for the angular velocity $\dot{\phi}$ and a turning radius R of $0.4[m]$. The approach for recreating the first simulation run is used once again, with the same values for the tweaked parameters, with the results given in table 5.2.

Settings	p -value	Right Wheel Velocity $v_R [m/s]$	Angular Velocity $\dot{\phi} [rad/s]$	Turning radius $R [m]$
Original [Solc and Sembrera, 2008] settings	0.54	0.12	-0.14	0.4
Recreated [Solc and Sembrera, 2008] settings	0.54	0.12	-0.117	0.51
Recreated [Solc and Sembrera, 2008] Altered p	0.55	0.12	-0.117	0.51
Recreated [Solc and Sembrera, 2008] Altered v_R	0.54	0.143	-0.14	0.51
Recreated [Solc and Sembrera, 2008] Altered v_R & p	0.55	0.143	-0.14	0.51

Table 5.2: Results of recreated simulations and compared to the [Solc and Sembrera, 2008] model for the 2nd simulation run.

As shown in table 5.2 adjustments of p have no effect on the turning radius, opposite to the results of table 5.1. Greater and smaller values of p have also been used, though it made little difference. In both

table 5.1 and table 5.2 it was necessary in the recreated model to increase the wheel velocities to obtain the angular velocity of [Solc and Sembrera, 2008]. This suggests that an error in the recreated simulation model is likely to be found in the vicinity of the model part that calculates velocities.

In figure 5.6 is a plot of the recreated model with [Solc and Sembrera, 2008] settings from the 2nd simulation run given along with two circles, made from the radii values listed in [Solc and Sembrera, 2008] for the 2nd simulation run. The red circle has the radius of the simulated result and the green circle has the radius of the data measurement from the 2nd simulation run of the [Solc and Sembrera, 2008] paper. The green circle denoting data measurements is not plotted in figure 5.5 as the radius in the 1st simulation run was measured to 0[m]

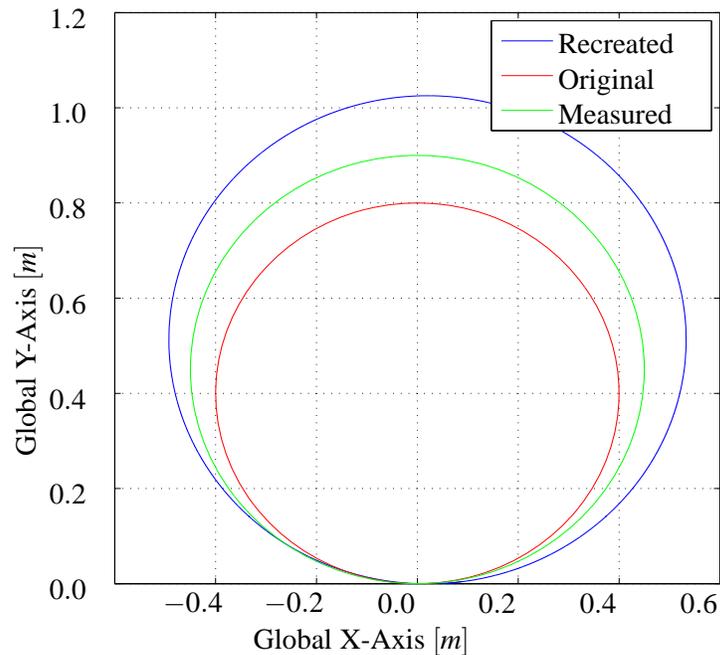


Figure 5.6: Plot of original, recreated and measured circles of 2nd simulation run.

As the model is different from both simulation results listed in [Solc and Sembrera, 2008], it is concluded that the recreated model is faulty, though numerous attempts have been made to locate errors in the recreated model. The alterations used in the tables have been found iteratively, and it has only been the p and v_i parameters that have been iteratively adjusted. The results from the recreated model does deviate from the [Solc and Sembrera, 2008] results, though if tweaked it is believed to replicate the measurement data for *Dr. Robot Jaguar* to a satisfying extend. In the results of the 2nd simulation run illustrated in figure 5.6 it is seen that the differences between the measured data and both simulated results are 5[cm]. This can be interpreted as being just as much off as the [Solc and Sembrera, 2008] model, which can be considered acceptable. Though in fact it is the two simulation models that are wanted to coincide, which is not the case for the recreated model and the [Solc and Sembrera, 2008] model.

Before being able to utilize the [Solc and Sembrera, 2008] model robot specific parameters are to be interchanged, such as weight and wheel radius. Also parameters such as tire-stiffness and the friction factor are important and are to be iteratively determined when fitting the model to measurement data.

This will be key when verifying data from *Dr. Robot Jaguar* with the recreated [Solc and Sembrera, 2008] model. It should also be noted that the [Solc and Sembrera, 2008] DVM is made for a vehicle driving inside on a vinyl floor, which is why some deviance is expected.

5.3 Skid Steering in 4-Wheel-Drive Electric Vehicle

The second paper that has been examined is *Skid Steering in 4-Wheel-Drive Electric Vehicle* by [Shuang and Cheung, 2007]. The vehicle utilized in the paper is a mobile skid-steered robot alike *Dr. Robot Jaguar*, though after obtaining parameter values for the paper it has become clear that the vehicle used in the paper is much larger, i.e. the weight of the vehicle is 500[kg]. [Shuang and Cheung, 2007] has a different approach to creating a dynamic model, as the model comprises other factors describing the robot compared to the model given in [Solc and Sembrera, 2008]. As for the model in [Solc and Sembrera, 2008] the [Shuang and Cheung, 2007] model has also been modeled by the project group. The model has been modeled in *Matlab Simulink* and can be found on the enclosed *DVD*. The model created on the background on the information given in the paper has not been functional, as results retrieved from simulations have not been close to what was expected from the model. Singularities and algebraic loops have been a recurring problem when working with the model, and a lot of time has been spent in solving simulation problems and double checking equations. Therefore the model is described instead, and will not be used as the simulation model for the designing of the controller for the *Dr. Robot Jaguar*.

The [Shuang and Cheung, 2007] model is divided into a tire model and a vehicle model. In the tire model the equation describing friction coefficients is given in equation 5.3.

$$\mu = \mu_0 \cdot e^{-\frac{(V_x - \omega \cdot R)^2 + V_y^2}{V_0^2}} \quad (5.3)$$

The model of friction has a very different approach than the one proposed in [Solc and Sembrera, 2008]. The formula describing the friction coefficients is made in such a fashion that it is dependent on longitudinal (V_x) and lateral (V_y) velocities of the wheels as well as the peripheral velocity of the wheels ($\omega \cdot R$). Which is understood as the friction being calculated by the magnitude and direction of the wheel velocities. The friction becomes greater as the lateral velocity becomes greater and the longitudinal becomes smaller, with a steady peripheral wheel velocity.

Slipping of the wheels are found relative to the peripheral wheel velocity for both longitudinal and lateral velocities. From the found slips a parameter, infinitude slip, is found describing slip relative to the friction term. Three infinitude relative slips are found in the model, though little physical meaning can be given to them, other than they describe the relationship between the slipping of the wheels and the calculated friction coefficient. This is done for both the lateral and longitudinal direction for the wheels, and the two found infinitude relative slips to calculate the infinitude total slip. These relative slip ratios are used along with the friction term and normal forces on the wheels to obtain the longitudinal and lateral forces acting on the wheels. [Shuang and Cheung, 2007] mentions a source, *Bakker et. al*, where

the tire model has been obtained, though it has not been possible to obtain this source and retrieve a further explanation of the components used to calculate the lateral and longitudinal forces. The wheel forces are used in dynamic equations as done in section 5.2 *Kinetic Model Of A Skid Steered Robot*, to obtain longitudinal, lateral and angular accelerations.

Longitudinal acceleration The equation describing longitudinal acceleration in [Shuang and Cheung, 2007] differs from the approach used in [Solc and Sembrera, 2008], though in both models are the longitudinal forces used to obtain the longitudinal acceleration.

Firstly it differs by subtracting velocity contributions of the other motion directions, lateral and angular, from the acceleration. [Shuang and Cheung, 2007] has not labeled a source for the equations, where the velocity contributions are included, and literature has not been found, when browsing fundamental physics literature to supports the equation constellation. It was assumed correct and has been included in the simulation model.

Secondly it differs by subtracting roll friction, that is given from the weight of the vehicle and longitudinal velocity, which is analogous with viscous friction due to its dependency of the longitudinal velocity.

Thirdly it differs by subtracting a force generated by wind resistance, which is thought of being added because of the [Shuang and Cheung, 2007] being rather larger compared to the vehicles of the other papers concerning skid-steered vehicles. When modeling the [Shuang and Cheung, 2007] dynamic model the wind resistance term has been neglected, because it was thought of as insignificant when evaluating the model to function as a simulation model for the *Dr. Robot Jaguar*.

Lateral acceleration The equation describing lateral acceleration in [Shuang and Cheung, 2007] only differs from the equation in [Solc and Sembrera, 2008] by adding a velocity contribution similar to the one subtracted in the longitudinal acceleration equation. The contribution added is a product of longitudinal velocity and angular velocity.

Angular acceleration [Shuang and Cheung, 2007] writes that the lateral tire forces are thought of as causing resistance to turning, which is overcome by the differential longitudinal forces between the two sides of the vehicle, in order to turn the vehicle. The forces are used in a torque equilibrium and balances to obtain the yaw acceleration. When comparing the equation for angular acceleration with the approach used in [Solc and Sembrera, 2008] it is very similar, almost identical.

Other than the acceleration equations, the [Shuang and Cheung, 2007] vehicle model also differ by obtaining the wheel accelerations from torque equilibriums. Such a torque equilibrium is given for wheel (1,1) in equation 5.4, where M_f is the roll resistance moment, J_w the inertia of a wheel and T_d is the torque applied by the motor.

$$J_w \cdot \dot{\omega}(1,1) = T_d(1,1) - F_x(1,1) \cdot R - M_f(1,1) \quad (5.4)$$

By utilizing equilibriums for the wheels the dynamics of the wheels are comprised in the model unlike in [Solc and Sembrera, 2008], which can only be a model improvement, when wanting to describe the dynamics of the system.

The model also differs by having rather complex equations describing both normal forces on the wheels, which are dependent on the equations of motion, and equations that describe the rolling resistance moment which depends on normal forces and longitudinal velocity of the vehicle.

The entire model is not described here, nor are the equations given here, due to the lack of functionality of the simulation model. Though as an overall comment of the model, it is thought that the [Shuang and Cheung, 2007] model likely is to be more precise, as model components are used in the model that are not used in [Solc and Sembrera, 2008], such as torque equilibriums for the wheels, a complex function describing the friction (μ) and variable normal forces, though it is unknown if the components contribute to a more precise model regarding simulating the behavior of the *Dr. Robot Jaguar*. Not all the components are necessary in describing a model for *Dr. Robot Jaguar*, e.g. wind resistance, which would be unnecessary as the robot will not move fast enough, or have large enough surface area to experience significant wind drag, as a result of the longitudinal velocity. The model components might have been necessary in the robot build by [Shuang and Cheung, 2007] as it much larger than *Dr. Robot Jaguar*. If the simulation model used for *Dr. Robot Jaguar*, [Solc and Sembrera, 2008], is insufficient and another is sought, the paper presented by [Shuang and Cheung, 2007] have components that can be added to obtain a more precise model.

The simulation model has been tested with different settings, of which one is equal but opposite wheel velocities as done in [Solc and Sembrera, 2008]. Furthermore the constant parameters used in the [Shuang and Cheung, 2007] model have been copied from the [Solc and Sembrera, 2008] model. From these settings it is assumed that the correct XY-plot output should be a circle, as shown in figure 5.5 on page 30. From the [Shuang and Cheung, 2007] paper are semi-circular patterns given in the simulation results, these appear to be created by applying equal but opposite torque on the wheel sets. The recreated model can make semi-circular paths, as shown in the paper, but when trying to recreate a full circle, as done in [Solc and Sembrera, 2008], the recreated model fails. Figure 5.7 shows an XY-plot of the recreated model attempting to make a full circle. The model creates what appears to be a circular path, but when it returns to the origin of the plot, an anomaly occurs, which makes the model go straight instead of continuing to drive in a circle. This is also reflected in the motor torque which is plotted in figure 5.8. The sudden drop in motor torque occurs at the same time as when the circular pattern is broken in figure 5.7. This leads to the suspicion that the fault is coming from either the roll resistance M_f or the normal forces F_z . As the [Shuang and Cheung, 2007] paper does not conclude on a full circle path, but only on semi-circles, it is not clear whether the anomaly is caused by equations of the [Shuang and Cheung, 2007] model or if it is caused by a modeling error in the recreated model. Different from the [Solc and Sembrera, 2008] model and intuition, does the [Shuang and Cheung, 2007] model not make a small circle similar to the 1st simulation run of [Solc and Sembrera, 2008]. The circle created in

[Solc and Sembrera, 2008], with equal and opposite velocities has a turn radius in the vicinity of $0.02[m]$, but when using the [Shuang and Cheung, 2007] model the turn radius is found to be in the vicinity of $2[m]$. Like in the recreated [Solc and Sembrera, 2008] model has the recreated [Shuang and Cheung, 2007] model been gone through many times in order to locate the error, though it has not been found.

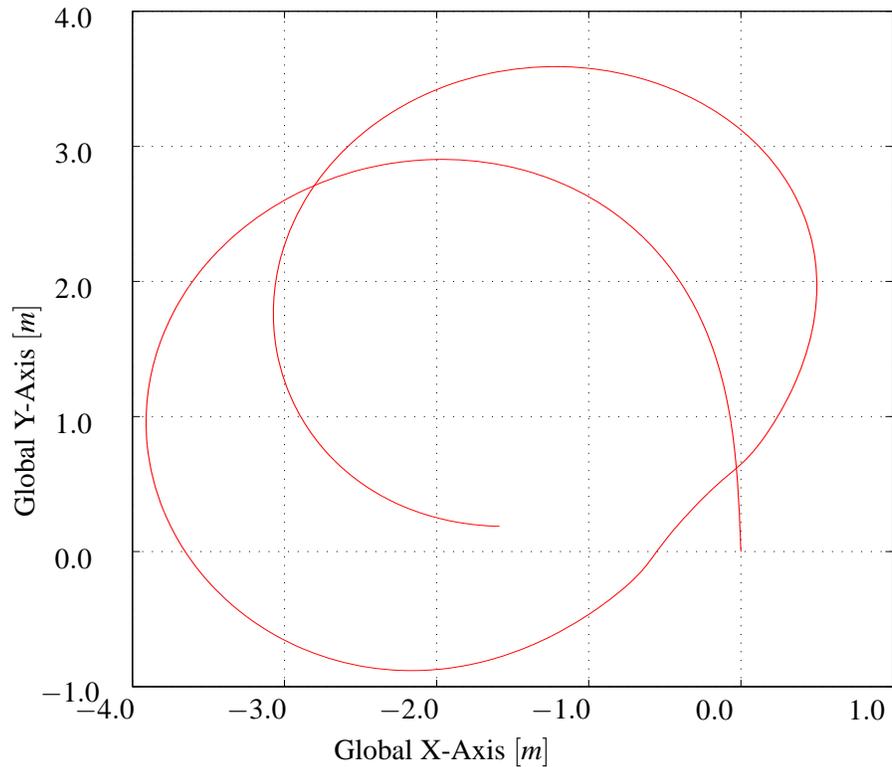


Figure 5.7: XY-Plot of equal but opposite wheel velocities.

In order to locate the error in the recreated model, it is likely that all parameters are to be monitored in order to find the origin of the error. As the error occurs when the vehicle return to its starting position, a full circle, it is possible that the error is caused by a modeling error regarding angular position that is not reset when reaching the vicinity of $2\pi[rad]$.

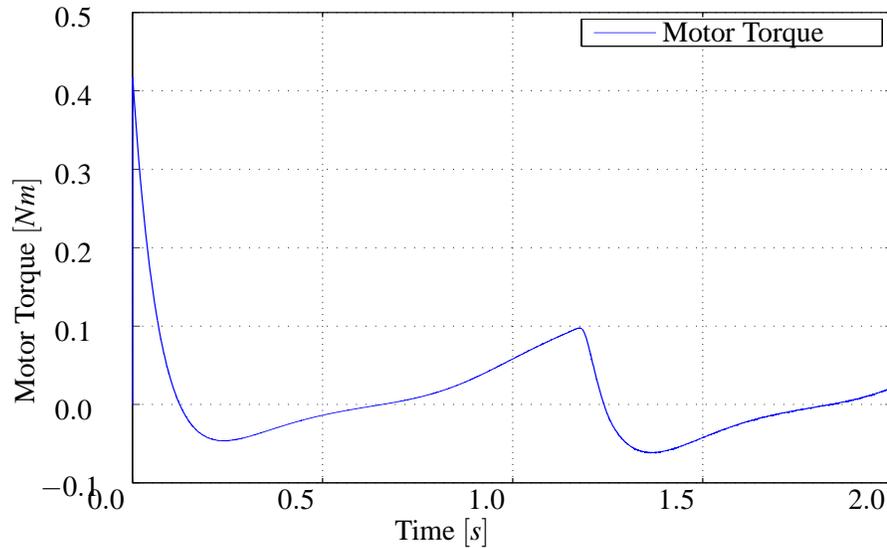


Figure 5.8: Plot of the torque generated by a left motor during the XY-plot.

Different from the [Solc and Sembrera, 2008] model the [Shuang and Cheung, 2007] model does contain many unknown parameters that needs to be estimated for the *Dr. Robot Jaguar* in order to make the model valid to verify it against its measurement data from the *Dr. Robot Jaguar*. These parameters are to be iteratively determined when verifying against data measurements, which is a great time consuming task, because at least seven independent constant parameters are to be determined. The [Shuang and Cheung, 2007] model is almost functional, though several attempts have been made to make it work. Since the [Solc and Sembrera, 2008] model is found to work, it is decided to use the [Solc and Sembrera, 2008] model.

5.4 Verification of Model

In this section the experimental verification of the Dynamic Vehicle Model (DVM) is described. This includes design, setup and results of the experiment. The DVM used for the verification is the one presented in the [Solc and Sembrera, 2008] paper.

5.4.1 Design of experiment

The purpose of the model is to be able to predict the motion of the robot and thereby be able to design a controller for the robot. Given the same input, the DVM and the real system must behave in a similar manner. The DVM's input is the wheel rotational velocity, and the output is motion in a global coordinate system. A way to determine the motion of the physical robot is required in order to verify, that the output of the physical system corresponds to the output of the model given the same input. The measurements are divided into internal and external measurements:

Internal measurements These are measurements conducted by the robot. The wheel rotational velocity,

that is required as input, is measured by the encoders on the motors and is logged on the robots hard drive.

External measurements Measurements conducted by an external sensor. This is required to compare the output of the DVM to the robots motion in the real frame of reference.

There is no feasible alternative to the encoders for measuring rotational velocity of the wheels. However there are different ways to externally determine the robots motion in the global frame. In literature different sensors are used to determine the motion of a mobile robot:

Differential GPS This enhanced GPS system is used to record the motion of a mobile robot in [Mandow et al., 2007].

Laser Range Scanner The laser range scanner is suggested in [Mandow et al., 2007].

Camera A camera is the most commonly used sensor to record motion. This method is used in [Solc and Sembrera, 2008].

Since the purpose of the robot is to drive outside in the field the verification must take place on a similar surface.

Using differential GPS requires both the necessary electronics and *know how*, which neither are available at this point in the project, which is why this method is rejected. Recording the motion of the robot in an outside environment makes it hard to use a camera to track the movement. This is because it must be placed above the robot, requiring both scaffolding and a mount as well being remotely accessible, and the changing ambient lighting can influence the picture, especially if the *long shutter time* approach is used. The LRS and software for this is available and is easy to use outside, which is why this solution is chosen for recording the motion of the robot.

5.4.1.1 Internal Measurements

The internal measurements is done by using the encoders, an Arduino Mega board and a LabVIEW program running on a PC/104 computer, all mounted inside the robot chassis. The Arduino code is based on an exemplary encoder code found at [PSU Robotics, 2011]. The schematic is shown in figure 5.9

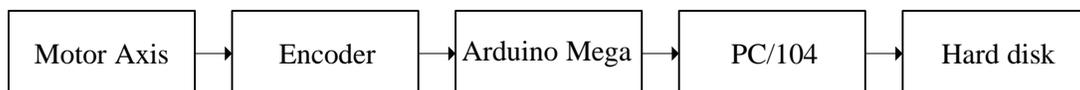


Figure 5.9: Schematic of the internal measurements.

As discussed in Section 3.3 each encoder has two square wave outputs channels, *A* and *B* with a high signal value at 5[V] and a low signal value at 0[V] where *B* has a 90° phase shift to *A*. The illustration of the waves is shown again in figure 5.10.

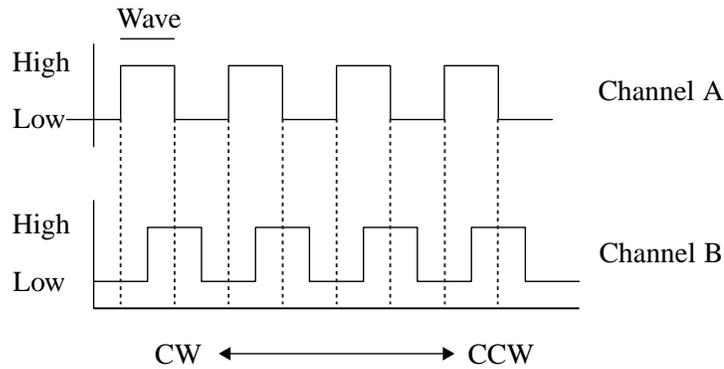


Figure 5.10: Illustration of the two square waves.

The Arduino Mega board is used to measure the square waves. The Arduino board has 54 I/O ports, whereof six are interrupt ports. These ports are suited for measuring events while another part of the code runs, by simply interrupting the code with an interrupt service routine and simply return to the code when the routine has been run. Channel A of each encoder is connected to an interrupt port. This means that an event is triggered every time there is a change on port A². Listing 5.1 shows the code for the interrupt service routine for the encoder mounted on the right side frontal motor. If A is high and B is low the motor is turning counterclockwise (CCW) and the position counter variable counts one up, but if both A and B is high it is turning clockwise (CW) and the position counter variable is counting one down. This is called gray logic and can be seen in table 5.3.

```

1  void RightFrontEncoderEvent()
2  {
3      if (digitalRead(RightFrontEncoderPinA) == HIGH)
4      {
5          if (digitalRead(RightFrontEncoderPinB) == LOW)
6              RightFrontEncoderPos++;
7          else
8              RightFrontEncoderPos--;
9      }
10     else
11     {
12         if (digitalRead(RightFrontEncoderPinB) == LOW)
13             RightFrontEncoderPos--;
14         else
15             RightFrontEncoderPos++;
16     }
17 }

```

Listing 5.1: Position counter loop for right front encoder.

	A Low	A High
B Low	-	+
B High	+	-

Table 5.3: Gray logic for the encoders - is CW and + is CCW.

To clarify, examples of CCW and CW rotation are given in figure 5.11 and 5.12. Figure 5.11 shows the two possible interrupt events when the wheel is rotating CCW. The red dot shows A going Low when B

²When A is changing from high to low or from low to high.

is High. The green dot shows A going High when B is Low.

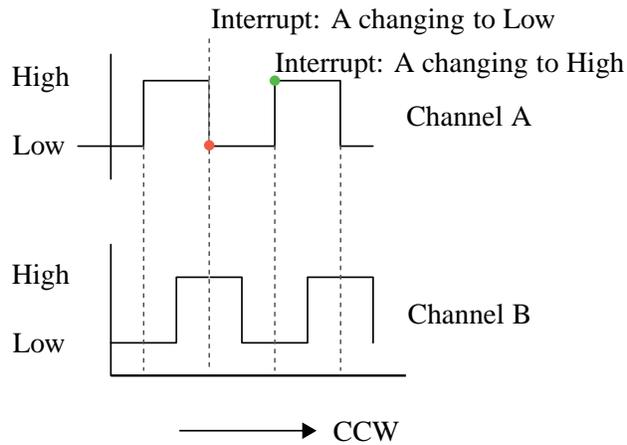


Figure 5.11: Example of counter clockwise rotation.

Figure 5.12 shows the two possible interrupt events when the wheel is rotating CW. Note that the red and green dot is placed opposite compared to figure 5.11. This is because the rotation is opposite and the event that was a falling before is a rising now and vice versa. The green dot shows A going Low when B is Low. The red dot shows A going High when B is High.

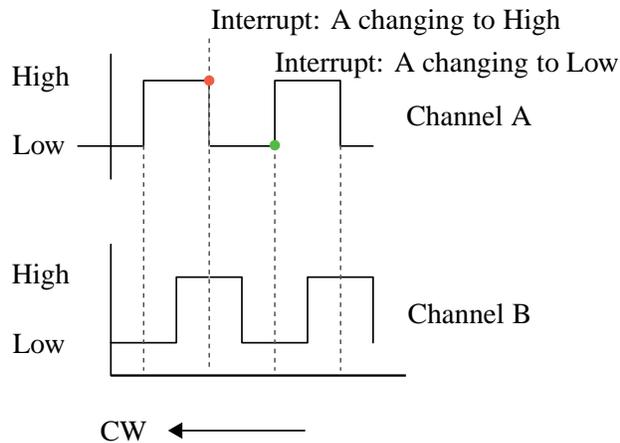


Figure 5.12: Example of clockwise rotation.

To measure the speed, the program samples ticks over a preset sample time and finds the difference in wheel position, by utilizing the position count before and after the sample time. The position difference divided by the time difference yields the speed in ticks per second. Dividing this with the 190 ticks per revolution yields the speed in revolutions per second. Choosing the sample time is a tradeoff between precision and sampling frequency. If the sample time is low, the measurements are not very precise but a high sampling frequency is achieved and vice versa. A sampling time between 10[ms] and 200[ms] is used depending on the purpose of the measurements. When sampling data to view transients a high sampling frequency is desirable and when measuring steady state values is the lower sampling frequency

preferable. If the encoders mounted on the motors had a better resolution, more poles per revolution, would a fast sampling frequency be available while still having good precision in data measurements. The LabVIEW program, running on the PC104 provides a GUI and is used to log the speed of each wheel and set the voltage of the motors prior to an experiment.

5.4.1.2 External Measurements

In this section the external measurements are described. As mentioned earlier the external measurements are obtained utilizing the LRS. All specifications for the LRS can be found in [Hokuyo Automatic, 2009]. The idea is to make a program that can use the LRS to track the movement of the robot. There are four challenges in doing this.

- There cannot be any other objects in the measurement area than the one that is tracked.
- The scanner requires a well defined target to hit.
- The scanner returns several points when it detects the target. Only one point is needed.
- The tracked point needs to be the same regardless of the heading of the vehicle.

The target is chosen to be a cylinder placed in the geometric center of the robot so it has the same shadow surface area in all directions. The rated range of the LRS is 4[m] it can however measure a little further. It is assessed that tracking the robot in a 4[m] × 4[m] area is sufficient for this experiment. As shown in figure 5.13 the LRS monitors the square area from the middle of the side. Therefore the maximum distance to the LRS is 4.47[m] as shown in equation 5.5. With a sensor resolution of 0.36° the maximum distance between measurement points become 28[mm] as shown in equation 5.6. Thus the target must be larger than 28[mm] to make sure that the sensor will capture it. A 58[mm] diameter tube is chosen for the target.

$$\sqrt{2^2 + 4^2} = 4.47[m] \quad (5.5)$$

$$\tan(0.36) \cdot 4.47 = 0.028[m] \quad (5.6)$$

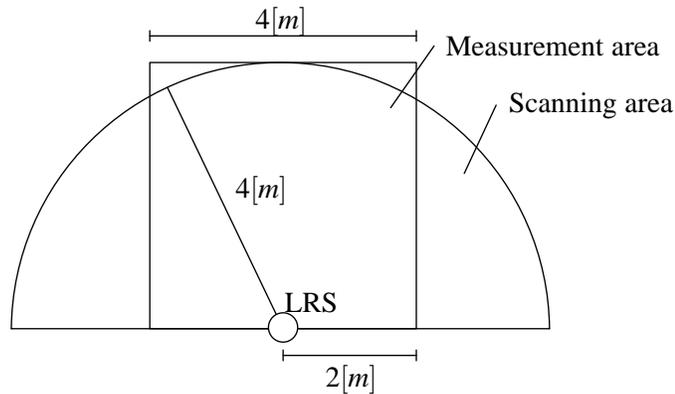


Figure 5.13: The scanning area and the measurement area.

The measured points are averaged to get only one point to log, which is why it is important that only the target is situated inside the measurement area. Figure 5.14 shows the approximate position of the average point, A, given the shown measured points. The problem with this approach is, that A is not the same when measured from different directions. Assuming that A is very close to the surface of the target, the center, C, can be found. This is done by calculating the unit vector between the LRS and A, multiplying this with the radius of the target and add this to A.

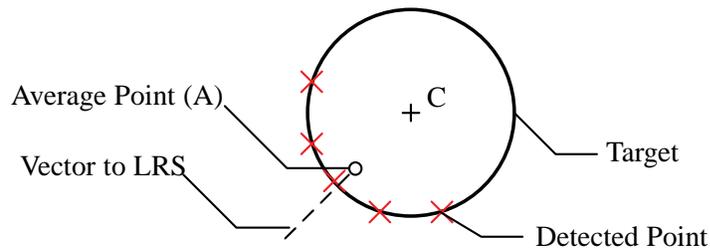


Figure 5.14: Principle behind the tracking of the cylinder.

This is not completely accurate because of the assumption, that A is on the surface of the target. Figure 5.15 shows a test of the software where the target, now with a diameter of 110[mm], is placed in (0, 600). The measured points are almost a straight line, which makes the assumption good. The measured center is located at (-29.94, 608.59). The distance error is only about 9[mm] or 1.5%, but the measured center is deflected almost 30[mm] to the left. Further study showed that the LRS measures about 5° to the left. This is a problem that needs to be corrected when the sensor is used for navigation. However, when measuring motion this problem is less significant, as the curvature and trajectory remains unchanged, other than the fact that the measurements are deflected by 5°. An inaccuracy of 1.5% is deemed acceptable. With an equipment measurement accuracy of ±3% of measurement distance the maximum error becomes ±4.5%.

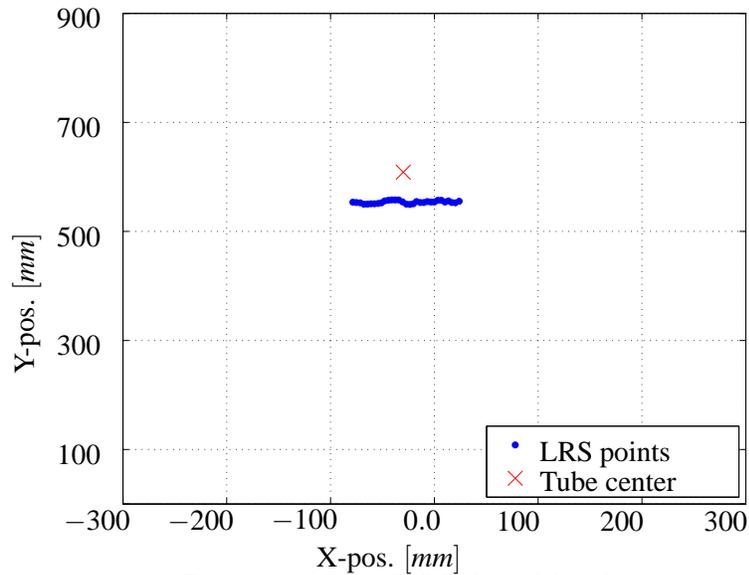


Figure 5.15: Plot of LRS view of the tube.

5.4.2 Experiment setup

In this section the setup of the experiment is described. Figure 5.16 shows the setup. To control the robot, the PC104 is connected to the laptop with a cross cable. When the test parameters are set and the program is executed the cable is unplugged, and the robot is ready to start after a preset time delay. The experiment must be conducted on a surface similar to that in a field. This is important in order to get the correct slip and friction. However rough terrain is undesirable because clods can knock the robot out of its course and introduce an error that the model is unable to account for. Another advantage of conducting the experiment outside is that all the equipment are tested in an outdoor environment. The final experiment is undertaken on a grass area. This area is not completely planar, so the error compared to the model may accumulate quickly. This is deemed acceptable because the error will accumulate on every surface, but more rapidly on a rougher surface.

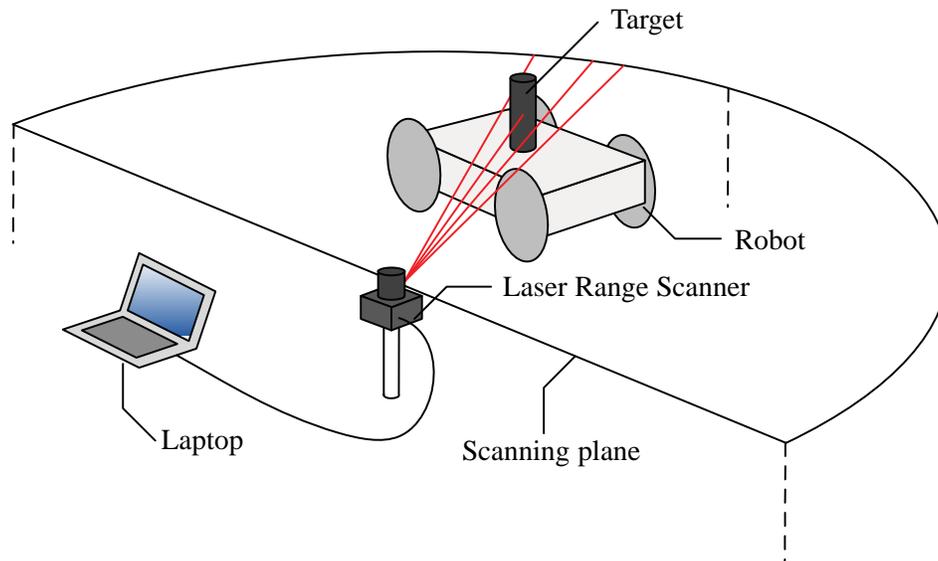


Figure 5.16: Experimental setup.

During the experiment different observations were made.

- Heat-shrinkable tubing used for target is not visible to the LRS. The surface is most likely too smooth which leads to a specular reflection. This means that the entire beam may be reflected in one direction and never return to the LRS. A rough surface will cause a diffuse reflection thus reflecting the beam in many different directions. Different solutions were tested and the target was redesigned with a bigger diameter and a rougher surface, realized with a common cloth. The increased diameter of the target will decrease the accuracy of the measurement.
- The LRS sometimes has problems with initializing and running outdoors. Different error messages were observed, some suggesting a laser malfunction. Some ideas were tested and the problem has seemingly nothing to do with insufficient power draw from a battery powered laptop or the ambient lightning. The problem encountered was never solved, though with patience the LRS works outdoors.
- When the motors on either side of the robot is given small equal opposite voltages, the robot sometimes remains stationary during startup, and two of the wheels will spin and burrow themselves. This is most likely because the wheels on a four wheeled vehicle have different grip, especially when the vehicle has no suspension. In some cases, the robot will be balancing on only two wheels. During a startup with low voltage, the two wheels without grip will spin, and the other wheels will remain stationary because of insufficient power. The vibration of the rotating wheels will eventually cause them to touch the ground and burrow themselves.

5.4.3 Results

In this section the results from the experiment are used to verify the model. This is done by comparing the measured movement of the vehicle and the curvature of the trajectory to the simulated movement and curvature. The curvature is calculated from three points (x_i, y_i) using equation 5.7 through 5.13. Equation 5.7, 5.8 and 5.9 is the length of the sides of the triangle the three points create. Equation 5.10 is the semiperimeter, equation 5.11 is Heron's formula, equation 5.12 is the formula for a circle enclosing a triangle and equation 5.13 is the curvature.

$$a = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \quad (5.7)$$

$$b = \sqrt{(x_2 - x_3)^2 + (y_2 - y_3)^2} \quad (5.8)$$

$$c = \sqrt{(x_3 - x_1)^2 + (y_3 - y_1)^2} \quad (5.9)$$

$$s = \frac{a + b + c}{2} \quad (5.10)$$

$$A = \sqrt{s \cdot (s - a) \cdot (s - b) \cdot (s - c)} \quad (5.11)$$

$$R = \frac{a \cdot b \cdot c}{4 \cdot A} \quad (5.12)$$

$$\kappa = \frac{1}{R} \quad (5.13)$$

The validation is based on five sets of data. Three data sets where the robot is driving with 18[V] on the right side motors and 2[V] on the left side motors, and two data sets where the setup is 22[V] on right side and 3[V] on left side. The data is collected over 10 seconds. The model is adjusted so the starting point and angle corresponds to the data. Figure 5.17 and 5.19 shows two comparisons of movement and figure 5.18 and 5.18 shows comparison of the curvature. The rest is shown in Appendix C.

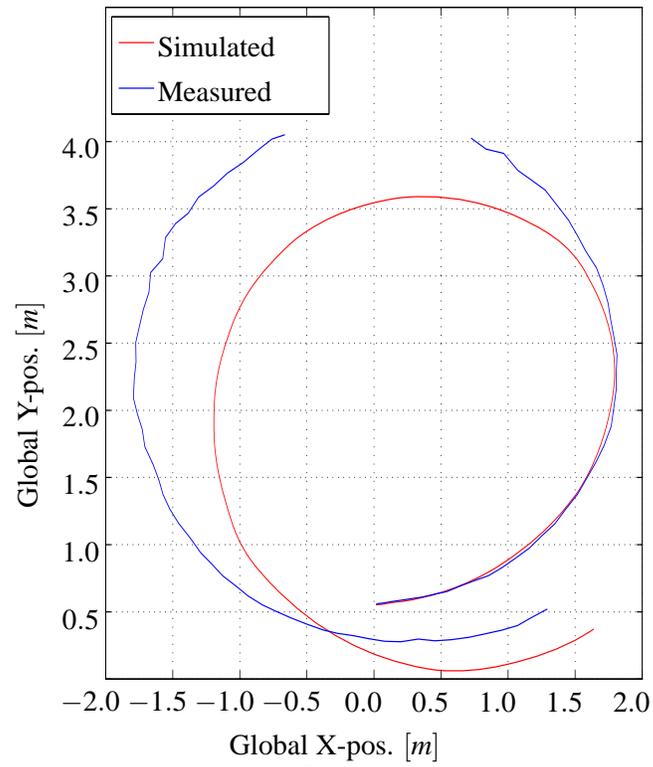


Figure 5.17: Position comparison for data set 2 when the robot is running with 18[V] on right and 2[V] on left.

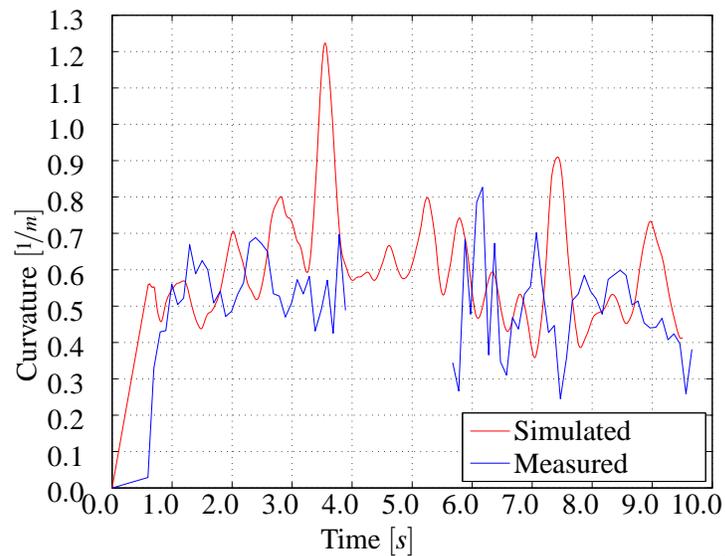


Figure 5.18: Curvature comparison for data set 2 when the robot is running with 18[V] on right and 2[V] on left.

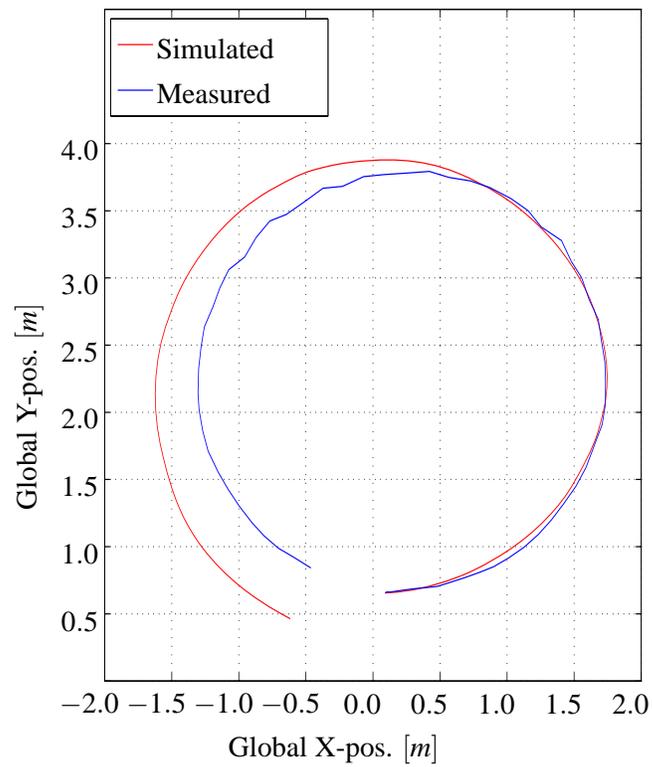


Figure 5.19: Position comparison for data set 6 when the robot is running with 22[V] on right and 3[V] on left.

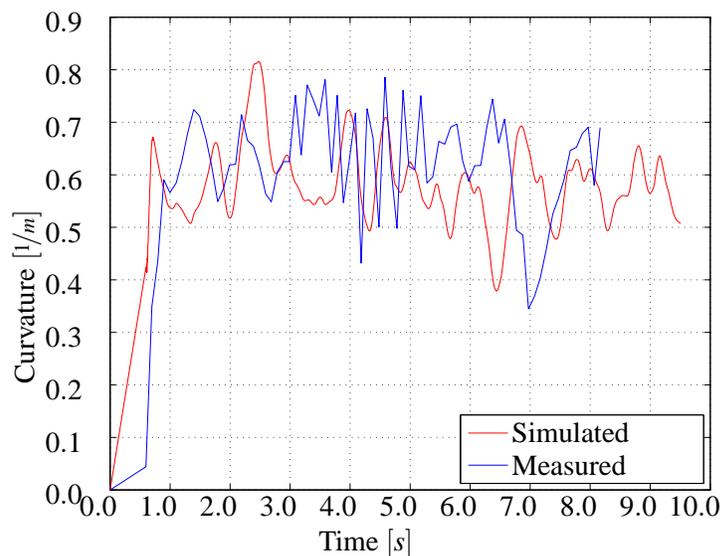


Figure 5.20: Curvature comparison for data set 6 when the robot is running with 22[V] on right and 3[V] on left.

As expected the simulation of the movement shows good correspondence in the beginning but the error accumulates as the simulation progresses. This is because the roughness of the surface is impossible to account for. When comparing the curvature it is apparent that the simulated and measured is roughly on

the same level. This suggests that the model and the real system has roughly the same turning rate but the error grows because of influences from the surface. In the curvature plots it can be seen that at time instances the difference in curvature is great, e.g. at 6.5[s] in figure 5.20 the curvature is different. The measured data has a higher curvature than the simulated data, which means that the measured data will generate a tighter circle at this point. The sudden differences that can be seen in the plots is thought to be because of imperfections in the 2D surface area of the terrain. The accuracy is however assessed to be sufficient for the purpose of closed-loop controller design, as the error is continuously corrected.

Motor Model 6

6.1 Armature Resistance	50
6.2 Motor Constant	51
6.3 Armature Inductance	51
6.4 Mass Moment of Inertia	53
6.4.1 Load Characteristic	53
6.4.2 Inertia Calculation	55
6.5 Motor Model Block Diagram	56
6.6 Motor Model Verification	58

In addition to the DVM developed in chapter 5 *Dynamic Vehicle Model* a motor model is necessary, in order to have a dynamic model of the system at hand. The system comprising both motor model and the vehicle model will be denoted as the final model henceforth. In figure 6.1 the final model is denoted.

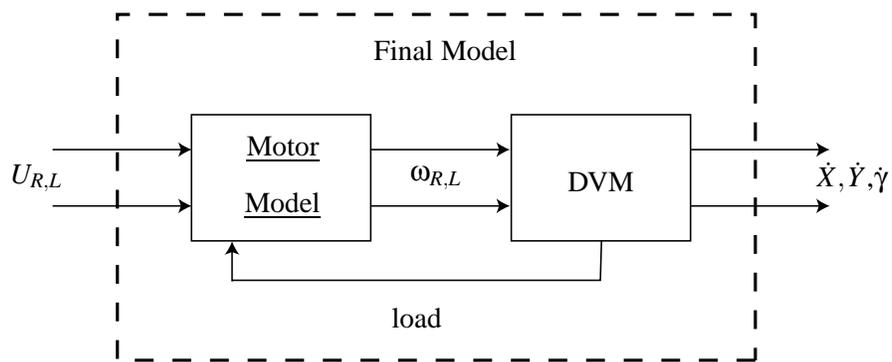


Figure 6.1: Final model with sub models.

The motor model is to be verified against the motors mounted on *Dr. Robot Jaguar* in order to obtain a higher level of similarity between model and reality. A high level of similarity is important for both motor model and vehicle model in order to obtain a final model with high level of similarity. The high level of similarity for the final model is important, because the final model is the background on which the robot controller is to be evaluated.

Firstly the motor model is to be determined. In order to create a motor model of the motors mounted on the *Dr. Robot Jaguar*, it is necessary to obtain a range of motor specific parameters. The motor model developed can be found on the enclosed *DVD*. In the following are such parameters determined, and a motor model is to be built on the background of these.

Motor parameters are often given in the enclosed data sheet or given by the manufacturer through other means. Obtaining motor parameters from the data sheet or from the manufacturer has not been possible,

even though several attempts have been made to contact the manufacturer. Therefore it has been determined to obtain motor parameters, that cannot be obtained from the data sheet, through experiments. The motors in question are mounted onto the *Dr. Robot Jaguar* and are 24[V] carbon brushed *DC* motors, with a stator comprised of permanent magnets. For a PMDC (*Permanent Magnet Direct Current*) motor are the following equations in 6.1 given from [Centinkunt, 2007]. These equations describe the electrical circuit of the motor.

$$u_a = R_a \cdot i_a + L_a \cdot \frac{di_a}{dt} + e_a \quad (6.1a)$$

$$e_a = K_\phi \cdot \omega_a \quad (6.1b)$$

$$\tau_m = K_\phi \cdot i_a \quad (6.1c)$$

Where:

u_a is the *armature voltage* applied across the motor terminals.

R_a is the *armature resistance* in the motor i.e. the armature windings and carbon brushes.

i_a is the *armature current*.

L_a is the *armature inductance* of the copper windings.

e_a is the *back-emf* in the armature generated by movement of the magnetic fields relative to each other.

ω_a is the *angular velocity* of the armature, and the motor shaft through a gearing.

τ_m is the *motor torque* developed by the motor.

K_ϕ is the *motor constant* describing a linear relationship between *back-emf* and *angular velocity*. K_ϕ also function as a *motor constant* describing the linear relationship between *motor torque* and *armature current*, which can be assumed to be equal in numerical value and is therefore described by the same parameter. The assumption of equal value of the *motor constant* is made from observations made from [Rasmussen, 2010], [Andersen and Pedersen, 2010] and [Mohan et al., 2003], but is an approximation non the less.

To create a motor model, it is necessary to determine constant motor parameters R_a , K_ϕ and L_a . The first value to be determined is the armature resistance in the following section 6.1 *Armature Resistance*.

6.1 Armature Resistance

From the data sheet it is given that the rated output power P_m is 48.3[W] and the rated current I is 2.75[A], from which the armature resistance is calculated [Serway and Jewett, 2004].

$$P_m = I^2 \cdot R \quad \Rightarrow R_a = \frac{P_m}{I^2} \quad \Rightarrow R_a = \frac{48.3[W]}{(2.75[A])^2} = 6.4[\Omega] \quad (6.2)$$

By applying a multimeter to the motor terminals, to measure the resistance, a value of $6.8[\Omega]$ was measured. As motor parameters can alter from identical seeming motors, the measured value is assumed to be a more correct value of armature resistance. To obtain the resistance through an experiment, the voltage drop applied to the terminals and the armature current could be recorded for different steady motor velocities and be determined from the slope of the plot of the recorded data. The armature resistance is a volatile parameter as it is dependent of the ambient temperature, which is the cause of why the resistance can change from measurement to measurement. As the motor is in continuous drive it will give off heat and thereby raising the armature resistance. The temperature dependency is assumed to be negligible and is not in the scope of the motor model in this project.

6.2 Motor Constant

The *motor constant* can be calculated in a similar manner to the resistance, by utilizing the rated current and the rated torque from the data sheet. As the definition of the *motor constant* is the linear relationship between current and torque. This is done in equation 6.3 utilizing equation 6.1c.

$$K_{\phi} = \frac{0.06374[Nm]}{2.75[A]} = 0.175 \left[\frac{Nm}{A} \right] \quad (6.3)$$

Though the motor constant is used in two different equations, the units are still valid, which is sought explained in *appendix B.1 Unit Comparison of Machine Constant*. The motor constant used in equation 6.1b is more difficult to find as the only the armature velocity is known. If the second motor constant is to be found experimentally, the motor terminals should be disconnected from the voltage source and connected to voltmeter in order to measure the *back-emf* generated. By spinning the armature at a constant velocity, the motor constant will be the only unknown in the equation and determined in a similar manner to the firstly determined motor constant. Due to the assumption mentioned earlier has this not been done.

6.3 Armature Inductance

The third motor parameter to be estimated is the armature inductance L_a of the copper windings, though there is not given an equation from which the inductance can be estimated through data sheet, therefore is it necessary to obtain this value through an experiment.

In figure 6.2 there is shown an equivalent circuit diagram of the PMDC motor, illustrated with the armature kept stationary by blocking the motor shaft. The motor shaft has been blocked to render the back-emf e_a term zero, and thereby enabling the motor circuit to be perceived as an *RL*-circuit. This diagram is utilized in the method used to obtain the armature inductance [Dr. Guillermo Rico, Fall 1996].

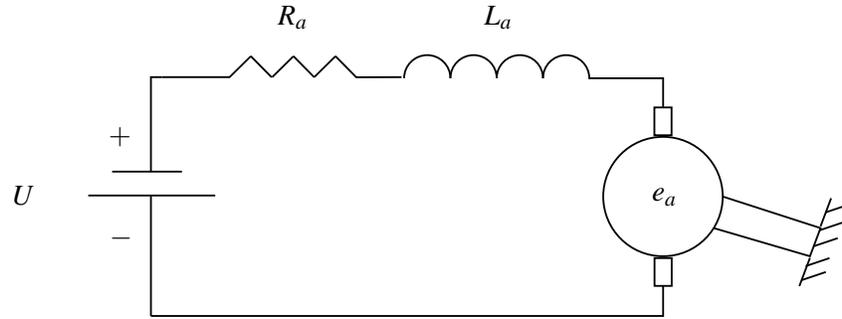


Figure 6.2: Equivalent diagram of the PMDC motor circuit.

An experiment, to obtain the inductance, has been proposed by [Dr. Guillermo Rico, Fall 1996] and the method is described in the following. By adding a known resistor R_K , in the range of $100[\Omega]$ to $1000[\Omega]$, and keeping the armature stationary, can the following method lead to the inductance of the circuit [Dr. Guillermo Rico, Fall 1996]. The obtained resistor, of $333[\Omega]$, is set in series with the motor as shown in figure 6.3 [Dr. Guillermo Rico, Fall 1996]. The supply voltage for the circuit u_{AC} is a $1.00[\text{V}]$ RMS sinusoidal voltage with frequency in the region of $2 - 10[\text{kHz}]$ (ω_{AC}) [Dr. Guillermo Rico, Fall 1996]. The supply voltage is to be measured with a voltmeter between point 1 and 3. The voltmeter is then moved to measure the voltage across resistor R_K , which is denoted V_K .

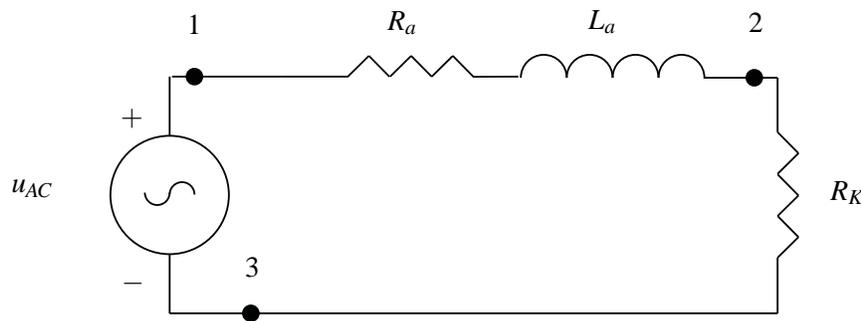


Figure 6.3: Equivalent diagram of armature inductance experiment with a resistor in series.

From the measurements and the specific imposed inputs the armature inductance can be calculated as shown in equation 6.4, according to [Dr. Guillermo Rico, Fall 1996]. A brief circuit analysis is presented in *appendix B.2 RL Circuit Analysis*.

$$L_a = \frac{1}{\omega_{AC}} \cdot \sqrt{\left(\frac{R_K}{V_K}\right)^2 - (R_a + R_K)^2} \quad (6.4)$$

The measurement data and the calculated inductance is shown in table 6.1.

R_K [Ω]	V_K [V]	R_a [Ω]	ω_{AC} [rad/s]	L_a [mH]
333	0.005	6.8	31415 (5kHz)	0.2i

Table 6.1: Data measurements of inductance experiment.

From the experiment a mean value of the armature inductance has been calculated to be $0.2i[mH]$. The complex operator on the inductance was unexpected, though it is clear when looking at equation 6.4, the complex operator heirs from the square root where the armature resistance is a fairly large value compared to the other values used. It should also be mentioned that the voltmeter used, was at its minimum limit of voltage detection, and the voltage across the known resistor could have been smaller, enabling the possibility of the a real inductance value. On this background it is determined that the inductance is prone to altering when verifying the motor model.

6.4 Mass Moment of Inertia

When determining the mass moment of inertia of the motor, it is necessary to include the inertia of the attached wheel as well. It is necessary because the wheel is firmly mounted to the motor shaft, and demounting it does not appear to be an option, as it might damage the setup. Determining the inertia with the wheel mounted, will result in a combined inertia for both the armature, the gearing and the wheel. This combined inertia is desirable, as neither armature or wheel inertias are given the data sheets supplied. The inertia will be obtained through an experiment with a no-load deceleration of the wheel. In equation 6.5 is a torque equilibrium as given in [Andersen and Pedersen, 2010].

$$\tau_m = K_\phi \cdot i_a = J_c \cdot \frac{d\omega_m}{dt} + B \cdot \omega_m + \tau_C \quad (6.5)$$

Where:

τ_m is the torque generated by the motor.

J_c is the combined inertia of the armature and wheel.

ω_m is the angular velocity of the motor.

B is the velocity dependent viscous friction term.

τ_C is a constant torque contribution from Coulomb friction.

Before utilizing equation 6.5 any further it is necessary to experimentally determine a load characteristic for the motor-wheel setup.

6.4.1 Load Characteristic

This is done by applying a range of different *DC* voltages to the motor terminals and record both the armature current and the motor velocity, for each applied voltage at steady velocity [Andersen and Pedersen, 2010]. By recording steady velocity values the inertia term can be excluded, as it is dependent on changes in wheel velocity. The equation for obtaining the load characteristic then takes the following form.

$$\tau_m = \tau_L = B \cdot \omega_m + \tau_C \quad (6.6)$$

The data is presented in table 6.2, and by utilizing equation 6.6 it is possible to create plot of the load characteristic.

u_a [V]	i_a [A]	ω_a [rad/s]	τ_m [Nm]
4.0	0.33	94.3	0.076
6.0	0.36	148.0	0.083
8.0	0.41	200.5	0.094
10.0	0.46	254.2	0.106
12.0	0.47	335.4	0.108
14.0	0.50	364.1	0.115
16.0	0.51	421.4	0.117
18.0	0.53	473.9	0.122
20.0	0.54	531.2	0.124
22.0	0.55	587.3	0.126
24.0	0.56	637.4	0.129

Table 6.2: Data measurements for load characteristics.

The load characteristic is found by plotting the torque against the motor velocity from table 6.2. The data is shown in figure 6.4 where the data points have been fitted with a linear approximation [Andersen and Pedersen, 2010].

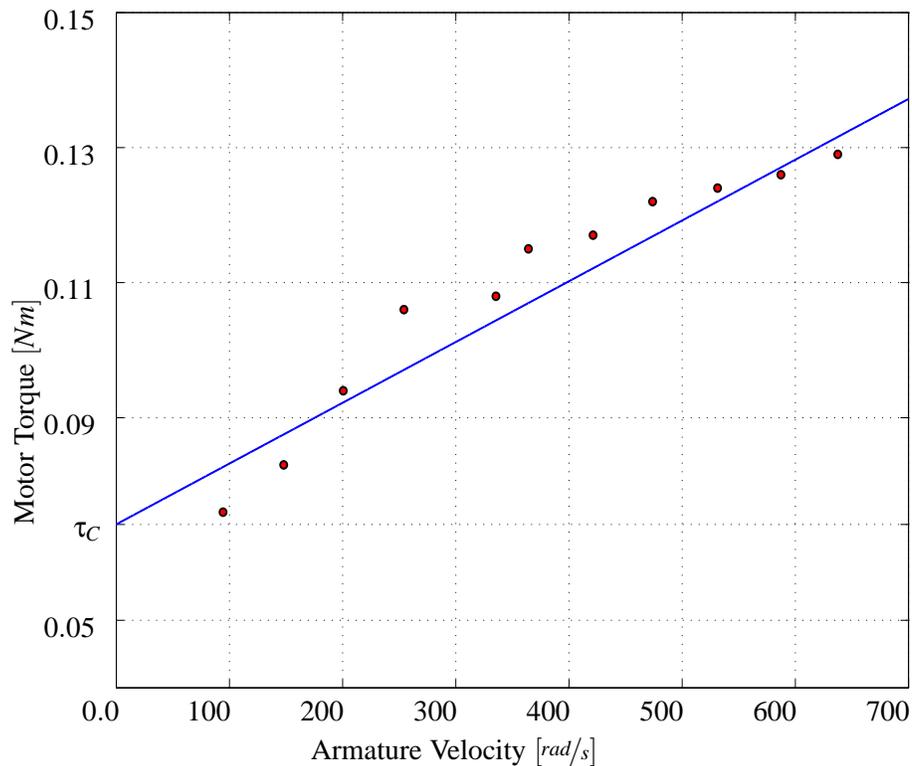


Figure 6.4: Plot of load characteristic.

In figure 6.4 it is possible to determine τ_c , as it is represented at the zero velocity crossing, and is read to have a value of $(0.0742)[Nm]$. Furthermore the viscous friction term B can be decided by determining the slope of the plot. The value of B is determined as $(9.0 \cdot 10^{-5})[Nm \cdot s]$.

6.4.2 Inertia Calculation

With the friction terms decided the inertia can be determined through rearranging of equation 6.5 to yield equation 6.7.

$$\tau_m = J_c \cdot \frac{d\omega_m}{dt} + \tau_L(\omega_m) \quad (6.7a)$$

$$\Rightarrow J_c = \frac{\tau_m - \tau_L(\omega_m)}{\frac{d\omega_m}{dt}} \quad (6.7b)$$

When measuring the inertia it is done through a deceleration test, where the motor at full speed, will be disconnected from the supply, leading to a deceleration of the motor until reaching a standstill [Andersen and Pedersen, 2010]. By disconnecting the supply to the motor there will not flow a current and therefore the τ_m torque can be set to zero, yielding the following equation.

$$J_c = \frac{-\tau_L(\omega_m)}{d\omega_m/dt} \quad (6.8)$$

During the deceleration the motor velocity and the time are recorded, to calculate the acceleration and the load torque. A plot of the motor velocity can be seen in figure 6.5.

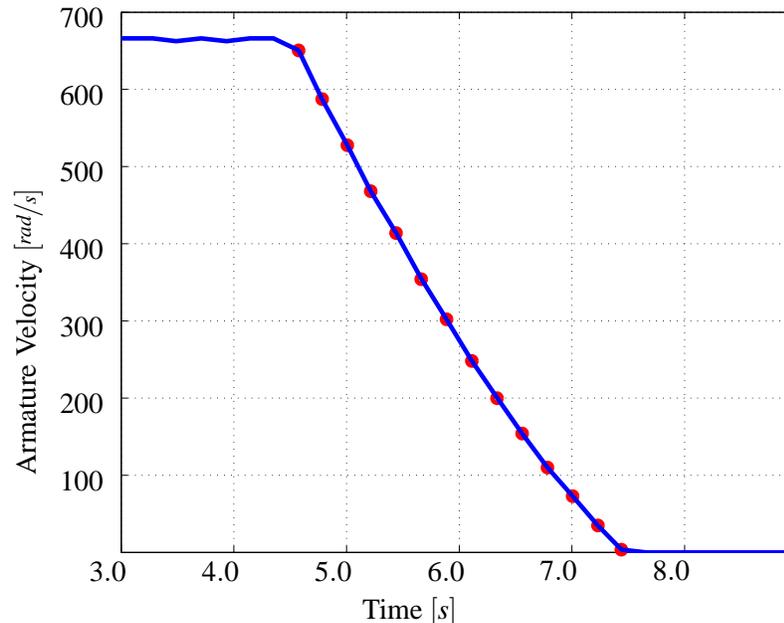


Figure 6.5: Plot of deceleration test velocity over time.

The deceleration is calculated for two coordinates in the plot, at a time, and τ_L is then calculated from

the velocity of the mean time in the deceleration coordinates. The coordinates used for the calculation of deceleration are shown in figure 6.5 with red dots. The inertia is then calculated for each coordinate set and a mean value is then calculated, that is to be used in the motor model [Andersen and Pedersen, 2010]. The mean value that is used in the motor model is calculated to be $(0.00015)[Kg \cdot m^2]$. It is noticed that all calculation done in this chapter are done on the armature side, which means that the calculated inertia only is valid for the armature side of the gearing, and the inertia does not represent the inertia of the wheel.

The motor parameters measured in this section are not exact values, as measurement precision of the equipment used is not of the best quality. Non the less does it represent good estimation of the parameters, as they will be used when verifying the motor model against measured data in a step response. Before comparing the step responses is a motor model formed for the parameters found. The parameters found are given again in table 6.3 as a conclusion to the parameter estimation of the *PMDC* motor.

R_a [Ω]	K_ϕ [Nm/A]	L_a [mH]	J_c [$kg \cdot m^2$]	τ_c [Nm]	B [$Nm \cdot s$]
4.8	0.175	0.2	$15 \cdot 10^{-5}$	$7.42 \cdot 10^{-2}$	$9.0 \cdot 10^{-5}$

Table 6.3: Data measurements of inductance experiment.

6.5 Motor Model Block Diagram

Having found the necessary motor model parameters in the previous sections, a block diagram of the motor can be formed. By utilizing equation 6.1 on page 50 and equation 6.5 to create the model, the equations have been *Laplace* transformed, to have a block diagram that is easily implemented into *MATLAB Simulink*.

$$\mathcal{L} : u_a = R_a \cdot i_a + L_a \cdot \frac{di_a}{dt} + e_a \Rightarrow u_a = R_a \cdot i_a + s \cdot L_a \cdot i_a + e_a \quad (6.9a)$$

$$\mathcal{L} : e_a = K_\phi \cdot \omega_a \Rightarrow e_a = K_\phi \cdot \omega_a \quad (6.9b)$$

$$\mathcal{L} : \tau_m = K_\phi \cdot i_a \Rightarrow \tau_m = K_\phi \cdot i_a \quad (6.9c)$$

$$\mathcal{L} : \tau_m = J_c \cdot \frac{d\omega_m}{dt} + B \cdot \omega_m + \tau_C \Rightarrow \tau_m = s \cdot J_c \cdot \omega_m + B \cdot \omega_m + \tau_C \quad (6.9d)$$

By isolating the velocity in equation 6.9d the following equation can be formed:

$$(\tau_m - B \cdot \omega_m - \tau_C) \cdot \frac{1}{s \cdot J_c} = \omega_m \quad (6.10)$$

From this rearrangement a block diagram can be formed as shown in figure 6.6.

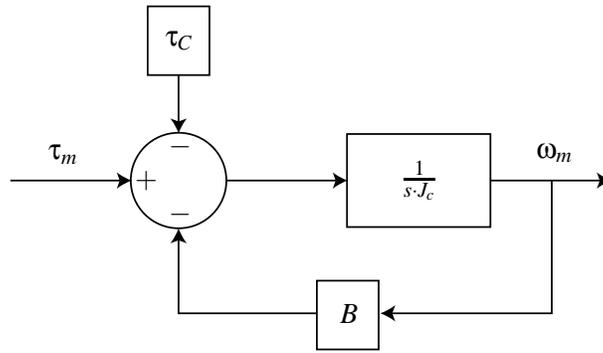


Figure 6.6: Block diagram of equation 6.10.

The setup in figure 6.6 calls for the motor torque, which can be obtained from equation 6.9c. By utilizing equation 6.9c the armature current is required which is obtained from equation 6.9a, as shown in equation 6.11.

$$(u_a - e_a) \cdot \frac{1}{s \cdot L_a + R_a} = i_a \tag{6.11}$$

And by substituting equation 6.9b into equation 6.11 the block diagram in figure 6.7 can be formed.

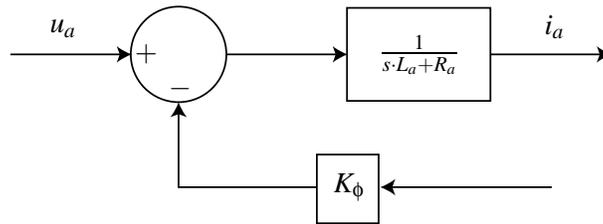


Figure 6.7: Block diagram of equation 6.11 with substituted *back-emf*.

From the previous block diagrams a block diagram describing the motor can be formed, as shown in figure 6.8.

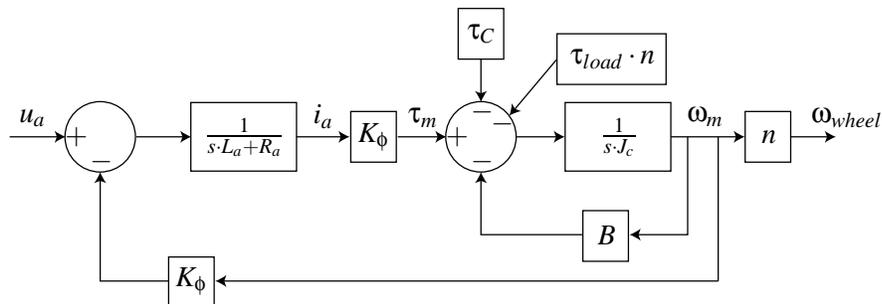


Figure 6.8: Block diagram motor model.

The final block diagram shown in figure 6.8 has two additions compared to the four equations from equation 6.9. The two additions consist of n and τ_{load} . n is the gearing of the motor which is given

as $1/19$, i.e. the wheel spins 19 times slower than the armature. The gearing is added as the model is to be connected with the simulation model of the vehicle. At all times it is assumed that no friction is present in the gearing, other than the contribution contained in the τ_L term. τ_{load} is added to the torque equilibrium and represents the counter acting torque applied to the motor by the system it is actuating. For motor model verification the block diagram takes the shape as shown in figure 6.9.

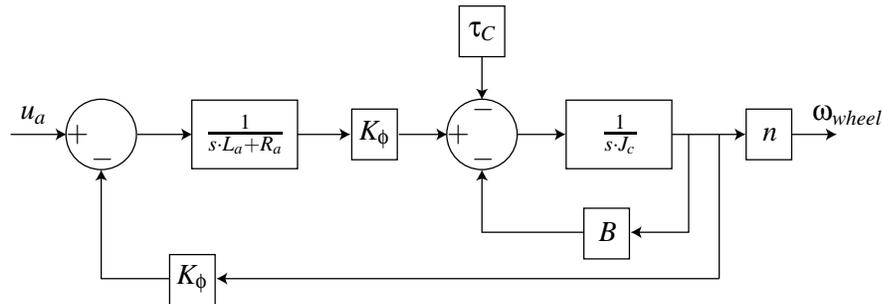


Figure 6.9: Block diagram motor model.

Where u_a is the applied voltage and ω_{wheel} is the wheel velocity.

6.6 Motor Model Verification

To verify the motor model the block diagram given in figure 6.9 is utilized. The block diagram is build in *MATLAB Simulink*, where the time and armature velocity are recorded. The recorded simulation data is then compared to the time and armature velocity measured on the real system. For both data sets the voltages are given as step inputs in order to see if the model can simulate the transient behavior of the real system. If it is not the case the motor parameters are adjusted in order to obtain a similar transient response. It is important that the motor model emulates the transient system well, because the final model comprising both motor model and DVM should represent the real system as close as possible. The transient behavior, of the motor model, is very important when the final model is to simulate driving in a trajectory where several turns are made, and thereby several transient situations for the motors. If the transient behavior of the motor model is very slow compared to real system, a controller made from the final model will be too fast and the vehicle will overcompensate when turning, and vice versa for a very fast transient motor model behavior.

When measuring the step response for the real system there will be no load applied to the wheel as it is not represented in the motor model used. In figure 6.10 can the unaltered step response of the motor model and the measured step response of the real system, where the motor model has been given the values found in table 6.3.

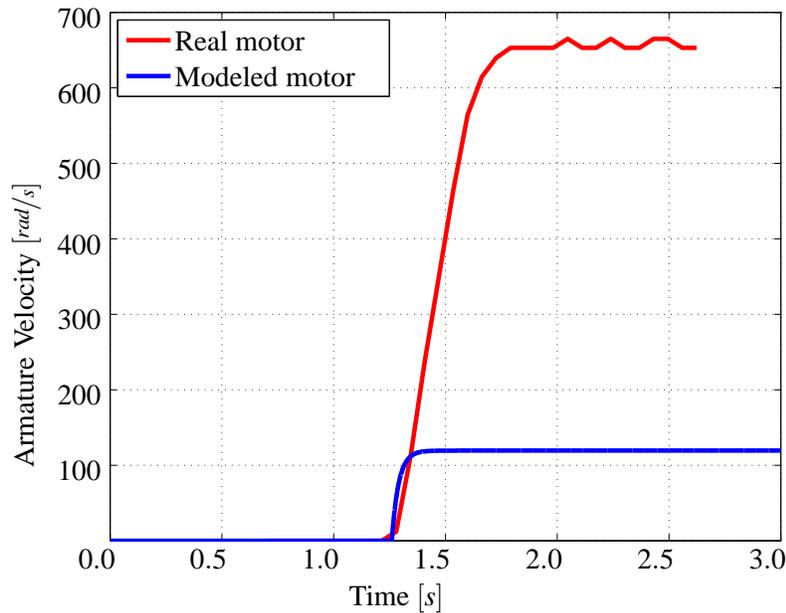


Figure 6.10: Plot of step responses for model with original settings and the real system.

In figure 6.10 it is seen that the steady state velocity is off by more than a factor of 6, which is not acceptable when striving for adequate similarity between model and reality. Furthermore the simulated response is much faster than the real system. These differences are to be minimized through altering motor model parameters. Each parameter has a function in shaping the step response, therefore all the parameters are to be altered in order to obtain a high level of similarity. When altering parameters it is important to keep the parameter values in the same vicinity as the measured and calculated values, in order to maintain a realistic motor model. By iteratively altering the parameters have the step response shown in figure 6.11 been obtained.

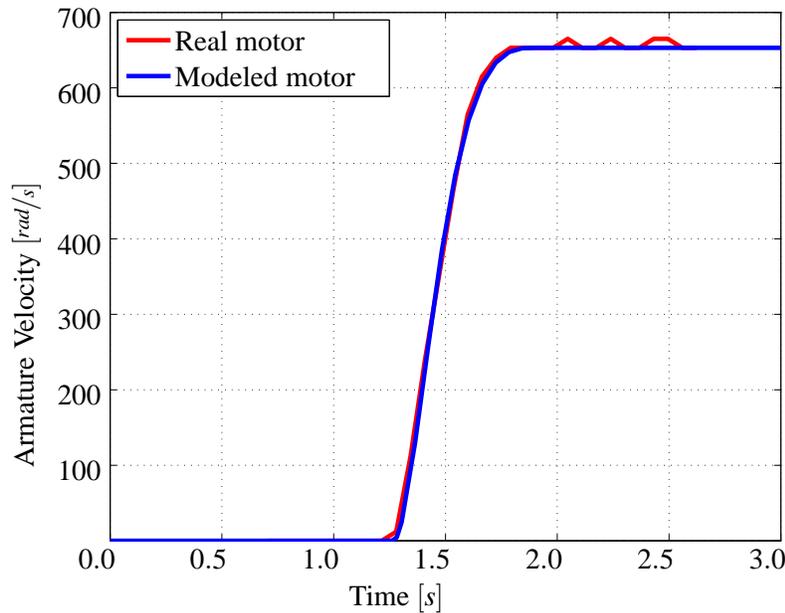


Figure 6.11: Plot of step responses for model with altered settings and the real system.

A much clearer similarity can be seen with the altered parameters. The motor model parameters used in figure 6.11 are given in table 6.4.

R_a [Ω]	K_ϕ [Nm/A]	L_a [H]	J_c [$kg \cdot m^2$]	τ_c [Nm]	B [$Nm \cdot s$]
1.7	0.03	0.15	$11 \cdot 10^{-5}$	$29 \cdot 10^{-3}$	$9.0 \cdot 10^{-5}$

Table 6.4: Altered motor model parameters.

The altered parameters are very different than the original obtained values. Only the viscous friction has the original value. The difference in the parameters most likely are to be credited to measuring mistakes and fluctuating readings from measuring devices. In order to see if the motor model responds correctly to other inputs other than the 24[V] step input, two other situations have been given. Step inputs with 12[V] and 18[V] are shown in *appendix B.3 Motor Model Step Inputs*. The step inputs for 12[V] and 18[V] does not fit as well as for the 24[V] step input, but for modeling purposes it is deemed sufficient. Having a model that corresponds satisfactory to the three different step situations, it is concluded that the model with altered parameters is to be used for the final model.

Navigation 7

7.1 Basic Navigation Principles	61
7.2 Obstacle Avoidance	63
7.3 Vector Field Histogram	64
7.3.1 Data Reduction 1	65
7.3.2 Data Reduction 2	66

In this section a system capable of navigating a corn field is developed. This system will work as the feedback to the controller. The system uses a laser range scanner to create view of the surroundings to find the best trajectory. The goal is to drive in the middle of the rows and turn at the end of a row. The chapter is based on the book *Introduction to Autonomous Mobile Robots* [Siegwart and Nourbakhsh, 2004] and different articles.

7.1 Basic Navigation Principles

Navigation consists of two basic competences [Siegwart and Nourbakhsh, 2004, p.257]:

Path planning Using a map to plan a trajectory that will cause the robot to reach it's goal.

Obstacle avoidance Using real-time sensor readings to react and alter the trajectory in order to avoid a collision.

Both of these competences are important for for a successful navigation. Like with most other applications, the quality of the output depends on the quality of the input. This means that the final path is greatly dependant on the quality of the map and the sensor readings. If the ability to react or plan fails the robot is not likely to reach its goal.

The goal of the *mark II* is however not to reach a destination, but rather patrolling a field driving in a weaving pattern. Considering the map of the advanced corn field shown in figure 1.1 on page 3, the shortest path to the goal, also called the optimal path, is not the intended path for the robot to travel. Furthermore since the Field Robot Event takes place outside the real track may not look exactly like the map.

Representation of navigation in done in an environment called *Configuration space* (C) [Siegwart and Nourbakhsh, 2004, p.259]. This environment consists of *Obstacle space* (O) and *Free space* (F) Hence:

$$C = O + F \quad (7.1)$$

When the robot is moving in a field, the rows provide an Obstacle space on either side of the robot and thus a Free space in between as shown in figure 7.1. With Obstacle space on both sides the robot only has one correct path from A to B. This simplifies the navigation because a simple obstacle avoidance algorithm can be sufficient to get a robot from A to B. However in the real field there will be holes in the rows and a less well defined obstacle space, so a robust obstacle avoidance algorithm is required. This algorithm cannot be used during turning from point B to D.



Figure 7.1: Configuration space of three rows in a corn field.

To get a smooth path following, a principle is called *wagon tongue* is used. Using this principle will yield a motion similar to a child pulling a wagon, hence the name. The Figure 7.2 shows the principle where the point ahead of the robot is referred to as the *rabbit* and the distance to the point is constant and is referred to as the *lead*. The rabbit is always on the path and the lead vector is always pointing at the rabbit. The robot utilizes an error angle feedback, θ_e , when navigating. This angle is the angle between the current heading of the robot and the lead vector. As the robot moves, the rabbit moves along the path thus changing the error if there is a change in the path. This principle will yield a smooth motion for the robot [Holland, 2004, p.129].

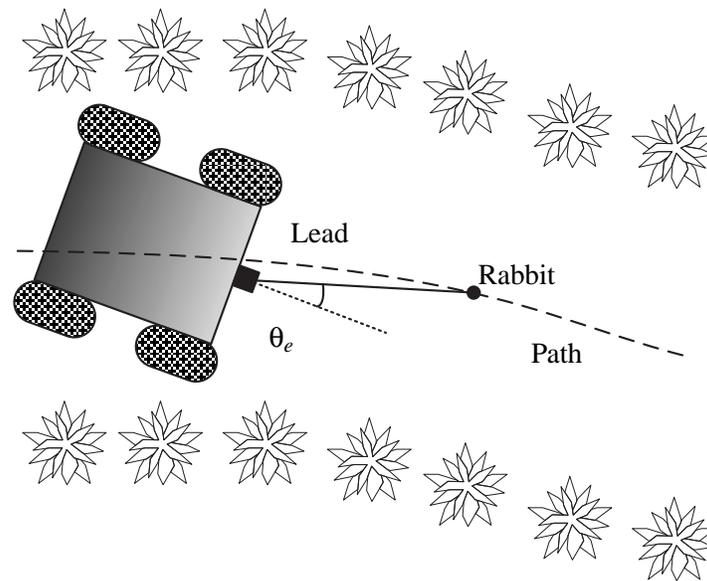


Figure 7.2: The wagon tongue principle.

7.2 Obstacle Avoidance

There are many different obstacle avoidance algorithms, and some are quite similar, or have been developed further and given a new name. To narrow down the number of choices some system requirements set up.

No prior knowledge about environment Some obstacle avoidance algorithms are in the grey zone between obstacle avoidance and path planning and rely on having a map. Since the map of the track is deemed unfit for navigation these algorithms are discarded.

Suited to use with LRS The laser range scanner returns an array where each entry consists of an angle and the distance to the obstacle in that angle. The same data can be obtained with sonar, but obstacle avoidance algorithms that require a camera cannot be substituted with a LRS.

Most of the algorithms are found in [Siegwart and Nourbakhsh, 2004] and some are described in the article [Borenstein and Koren, 1991]. After the initial sorting, seven algorithms are left. They are briefly described here.

Bug algorithm This algorithm is one of the simplest imaginable. It circumnavigate each obstacle in the robot's path by following the contour of the obstacle.

Vector Field Histogram This method uses a two stage data reduction process to reduce range data to the find the best trajectory for travel.

Curvature velocity This approach takes dynamic constraints into account to create a velocity space consisting of a rotational and translational velocity. The algorithm assumes that the robot only travels along arcs, and finds legal trajectories from different constraints.

Dynamic Window This approach is very similar to curvature velocity. It take the kinematics into account when searching the velocity space for all sets of rotational and translational velocity that is possible to reach within the next sample period. These sets are called the dynamic window. The dynamic window is hereafter reduced, so only the safe trajectories are kept in what is called the admissible velocities. The motion direction is chosen by an objective function that prefers fast forward motion.

ASL approach This method combines the Dynamic Window approach and an approach called Elastic Band to create an algorithm that is well suited to move through dense crowds.

Potential Field Method Potential Field Method utilizes imaginary attractive and repulsive forces from goal and obstacles to find the resultant force vector. The resultant force vector is the suggested travel vector [Borenstein and Koren, 1991].

Edge detection This algorithm steer the robot around obstacles by locating the vertical edges of the obstacle [Borenstein and Koren, 1991].

The Vector Field Histogram (VFH) method is chosen because it is a lightweight and robust algorithm, than can be implemented within a reasonable time frame, because it is already available in LabVIEW.

7.3 Vector Field Histogram

Vector Field Histogram is a motion planning algorithm proposed in 1991 by Johann Borenstein and Yoram Koren [Borenstein and Koren, 1991]. This section uses the nomenclature from this article. This is a robust and light weight obstacle avoidance method, that has proved efficient when traversing densely populated obstacle courses.

As mentioned earlier it uses a two stage data reduction process to find the best trajectory. This is shown in 7.3.

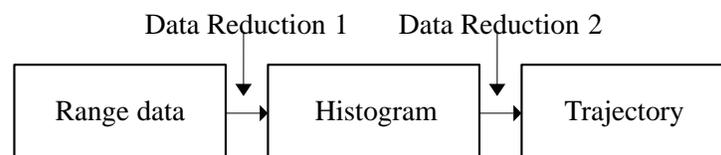


Figure 7.3: The two stage data reduction process.

7.3.1 Data Reduction 1

The first part of the data reduction reduces the range data to a one dimensional, polar histogram. Each scanning sweep consists of an array with n sets of polar coordinates (r_i, φ_i) . The histogram is created by comparing the magnitude of each point in a scan to a predefined distance threshold (DT) and assigning the i^{th} angle 1 if there is an obstacle and 0 if the angle is free. The magnitude threshold can be thought of as how far the robot looks ahead. This is a simplification of the original algorithm in [Borenstein and Koren, 1991], where several sensor readings were used to create a grid with certainty values (CV) representing how certain the algorithm is that there is an obstacle in that cell. Hereafter each angle (φ_i) is assigned a Polar Obstacle Density (POD), representing how certain the algorithm is that there is an obstacle in that direction. The robot used in the original paper was equipped with ultrasonic sensors, which are more inaccurate than a LRS. Therefore. Since the LRS is quite accurate the algorithm can be simplified. Figure 7.4 shows an example of a configuration space, and figure 7.5 is the histogram for the given example. The angles without obstacles are called candidate valleys.

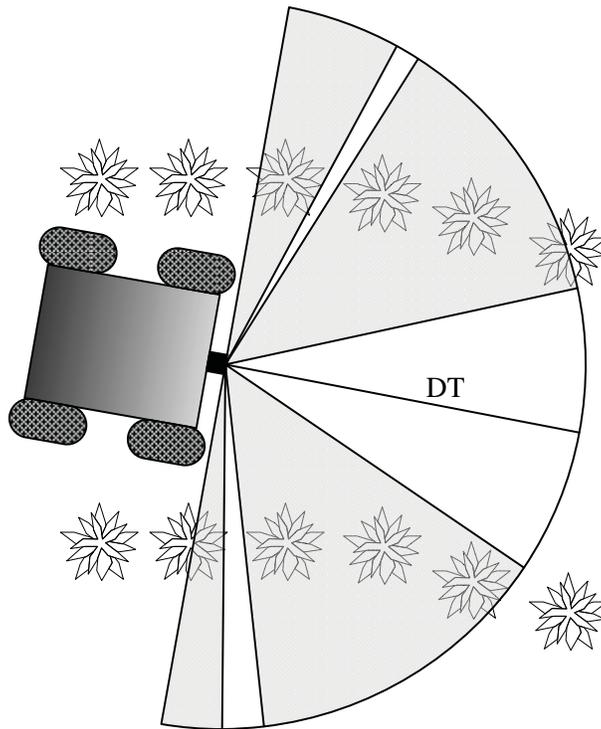


Figure 7.4: Example of a polar configuration space with the given DT. Gray is obstacle space and white is free space.

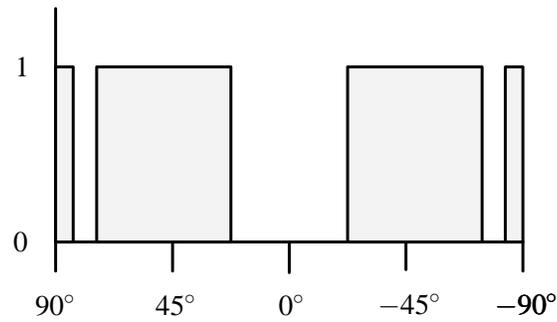


Figure 7.5: Histogram for the polar configuration space shown in figure 7.4.

7.3.2 Data Reduction 2

The second part of the data reduction computes the required steering direction (θ_e). The candidate valley closest to the current heading of the robot is selected for travel. The near and far border of the same valley are denoted k_n and k_f . θ_e is now defined as:

$$\theta_e = \frac{k_f + k_n}{2} \quad (7.2)$$

This is also a simplification of the original algorithm that distinguishes between wide and narrow valleys, and utilizes a target direction when selecting a candidate valley for travel.

Figure 7.6 shows the algorithm at work. The top shows the Cartesian grid with the obstacles and the chosen heading plotted, and the bottom shows the polar histogram with the candidate valley. Note that algorithm yields a reasonable trajectory with the given obstacle space.

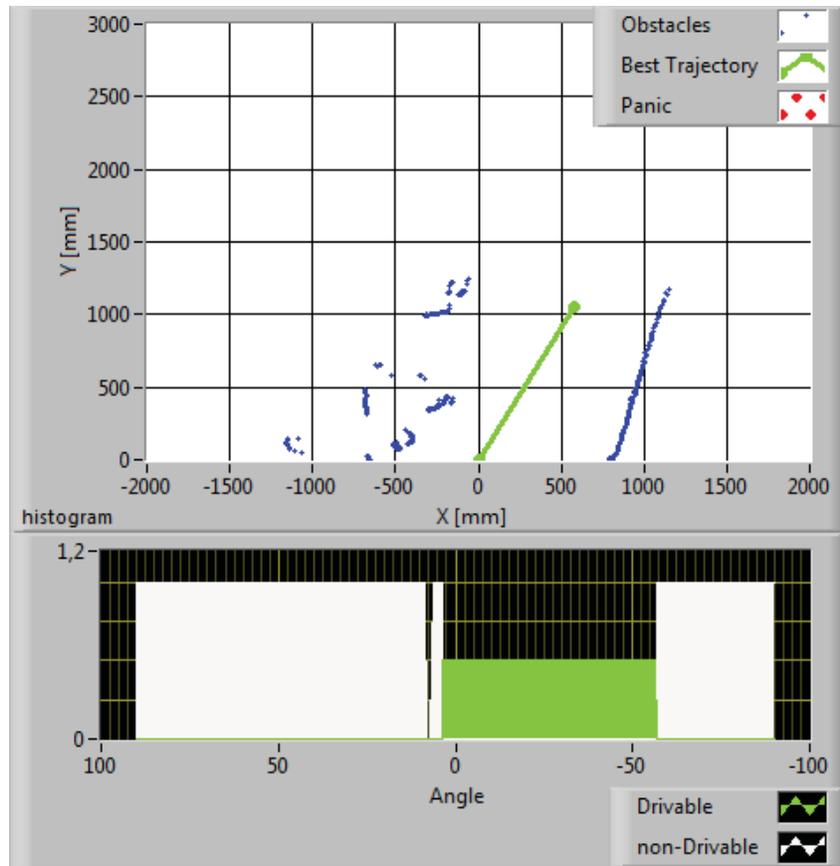


Figure 7.6: Vector field histogram.

There is one major issue with this algorithm. When approaching two equally good candidate valleys, the algorithm's trajectory will oscillate. This is because the white noise in the measurements can make the algorithm shift between the two candidate valleys. As mentioned earlier, the robot will operate in a situation, where there will be obstacles on either side of the robot most of the time. Therefore the robot will, for most of the time, have only one possible candidate valley. There are two exceptions to this. When there are missing plants in the corn rows, or when the robot is at the end of a row. When at the end of a row, the robot will need to turn or stop and the algorithm is no longer required. The case with missing plants could however prove a problem. The magnitude of the problem depends on how large the hole is. If the hole is bigger than the DT, the robot will choose a wrong trajectory. The DT is set around $1[m]$ but can be changed if necessary. It is possible that the algorithm will yield an oscillating or wrong trajectory, but it requires a hole in the row that is larger than $1[m]$ or that the robot's heading is closer to the hole than the desired path. It is hard to predict how the algorithm will perform when operating in the real system, but it is assessed that it will perform well in the current application.

Controller Design 8

8.1	Motor Controller	70
8.2	Navigation Controller	79
8.2.1	Path Following Fuzzy Logic Controller	82
8.2.2	Open Loop Sequential Turning Controller	85
8.2.3	Obstacle Handling Controller	86

In this chapter the controller for the *mark II* is designed. The controller can be divided into a navigation controller and a motor controller placed in cascade. The controller for the motors is designed to keep the motor velocities close to reference velocities from the navigation controller. The motor controller must be able to operate on its own, only receiving references from the navigation controller and velocity feedback from the encoders.

The navigation controller keeps the robot from colliding with obstacles and steer it through the corn field, using the feedback from the *LRS*.

As mentioned, the two controllers will be cascaded, with the closed loop motor controller situated inside the closed loop navigation controller as shown in figure 8.1. The navigation controller is a name given to the controller that utilizes the *LRS* feedback to control the movement of the robot. The navigation controller will be explained in section 8.2 *Navigation Controller*.

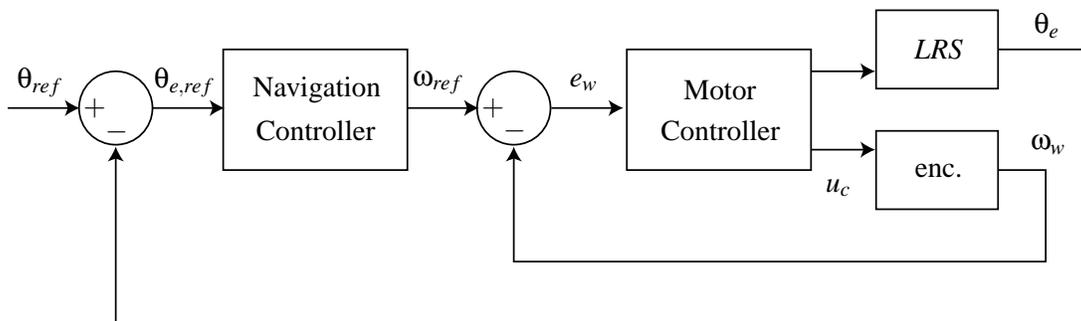


Figure 8.1: Block diagram of the controllers.

First the controller for a motor is designed.

8.1 Motor Controller

In this section the motor controller is designed. In section 5.4.1.1 the rotational wheel velocity is measured using the encoders. This wheel velocity can be used as feedback for the motor controller. It is also known that the input to the controller is velocity references, ω_{ref} , from the navigation controller. The only job for the motor controller is to reach and maintain a velocity for a motor, in close vicinity of to a given reference velocity, and correct if disturbances occur.

When creating a controller for a electrically driven motor, a PID controller is a common choice. This is due to its simple structure and its performance. The system, in which the motor controller is to be designed, is shown in figure 8.2. A motor controller will be implemented on all four motors. The motor controllers will be identical as the motors are assumed to be identical. The only difference between the controllers are the motors they control, and the reference inputs they receive, as they will be sent to left and right side motors respectively.

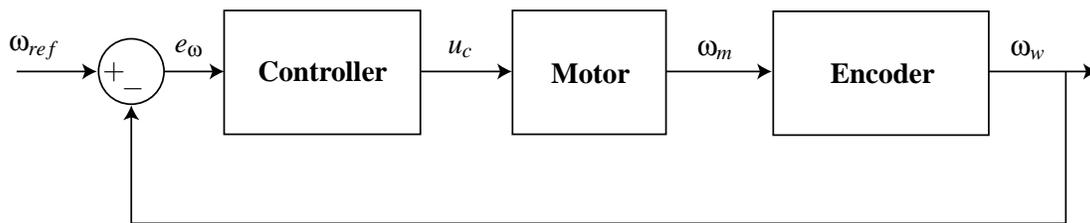


Figure 8.2: Block diagram of the motor system with controller.

Where in figure 8.2.

Controller denotes the motor controller that is to be designed.

Motor denotes a *PMDC* motor and the attached H-bridge¹.

Encoder denotes the mounted encoder.

ω_{ref} denotes the reference generated by the navigation controller designed in section 8.2 *Navigation Controller*.

u_c is the voltage signal from the controller, passed as a preset PWM signal to the H-bridge, and then to the motor.

ω_m is the angular velocity of the motor.

ω_w is the angular velocity of the wheel.

Before looking into the controller design the system, that is to be controlled, is reviewed. The block diagram of the system can be seen in figure 6.8 on page 57. In this block diagram it can be seen, that the system is a second order system, comprised of two first order systems in cascade. From this observation

¹The H-bridges are not included in the controller design.

it can be concluded, that the second order system, describing motor and wheel, is a second order system with two real distinct poles, meaning that the system has a damping coefficient greater than one. With a damping coefficient greater than one the system will have no overshoot [Phillips and Harbor, 2000]. As the system is a second order system, though it behaves much like a first order system, parameters such as rise time, overshoot and settling time will be design criteria parameters when designing the controller.

The PID motor controller will take the place named *Controller* in figure 8.2 and will have input ω_{ref} and ω_w and output u_c . The controller is chosen to be a PID controller as it is a common choice for *PMDC* motors. The PID controller is shown in figure 8.3 [Phillips and Harbor, 2000].

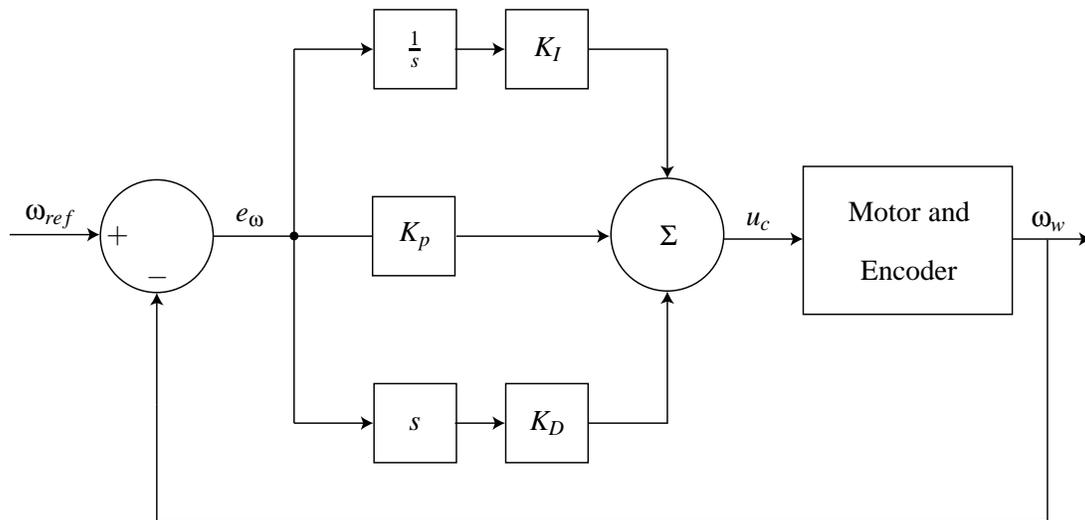


Figure 8.3: Block diagram of PID controller.

Where.

K_I is the integral gain.

K_p is the proportional gain.

K_D is the derivative gain.

u_c is the voltage passed from the PID controller to the motor.

e_ω is the error signal between the reference and the feedback signal.

In figure 8.3 it can be seen that the controller consists of three parts, from which the sum is passed on as the voltage signal u_c .

Integral element The upper part of the PID controller is the integral element, which is proportional to the integral of the error up to the time t [Araki, *]. The integral element can be interpreted as the accumulation of the past error [Araki, *].

Proportional element The middle part of the PID controller is the proportional element, which is the error at the time t , which is the present error [Araki, *].

Derivative element The lower part of the PID controller is the derivative element, which is proportional to the derivative of the error at the instant t , which can be interpreted as the prediction of the future error [Araki, *].

To implement the controller into the robot system, the Arduino board is used. Before implementing the controller into the robot system, the elements of PID controller, K_I , K_p and K_D , are to be determined. This is done in the final model, and gains have been fitted by reviewing the step response. When a desired response is obtained and the PID controller is implemented into the Arduino board, the parameters must be adjusted again, as the final model is not a precise replication of the real system. This is because motor model created differs from the real system and the encoder signal is also thought to be of poorer resolution than described earlier due to the 5 poles. When adjusting the elements of the PID controller, each element has its own effect on the system. When a motor in the robot is activated, the input signal is given as a step input.

The proportional gain K_p , that utilizes the present error, will, if adjusted correctly and implemented alone, lowers the rise time of the motor. The rise time is the time it takes the motor to go from 10% to 90% of the steady state velocity [Phillips and Harbor, 2000]. The rise time is only used to get an estimation of how fast the system reacts to a given reference. The proportional gain can be set to any relative high value to obtain a fast system. By setting a proportional gain high, two particular drawbacks occurs. The first drawback is that the controller reacts aggressively to minor errors, and with a sampled feedback signal, this can result in oscillations near the set reference. The higher the proportional gain the greater the oscillations. The second drawback is that a sufficiently high proportional gain will yield a overshoot when reaching its steady state velocity, due to the sampling intervals from the encoders. When choosing the numeric value of a proportional gain, it will be a tradeoff between the magnitude of the unwanted high overshoot, if present, and the desired low rise time[Wescott, *]. When applying this to the system at hand, a small overshoot can be accepted ranging around, 10% – 20%, as the controller is situated inside another closed loop controller described in chapter 8.2 *Navigation Controller*. The outer controller will compensate for velocity errors with the outer closed loop, by altering the motor references if necessary. An overshoot will, in worst case, yield a smaller turn radius of the vehicle, as the difference between the wheel set velocities will briefly become too high. A low rise time is often a desired parameter, though when keeping the system in mind, it can be a undesired parameters for the robot. Undesired because a motor, with a low rise time, can accelerate the wheel very fast, which will lead to slipping between the wheels and the ground. Slipping of the wheels has the effect, that the outer controller, navigation controller, will expect a fast response of the wheels, when in fact the wheel slipping will hardly propel the vehicle at all. To illustrate the effect of a proportional gain, a plot with a gain of 10, stepped to $18[\text{rad/s}]$ and sampling frequency $4[\text{Hz}]$ has been given in figure 8.4. The effect of sampling will be described later. The sampling frequency of $4[\text{Hz}]$ was initially chosen as it yielded a smooth steady state

velocity signal.

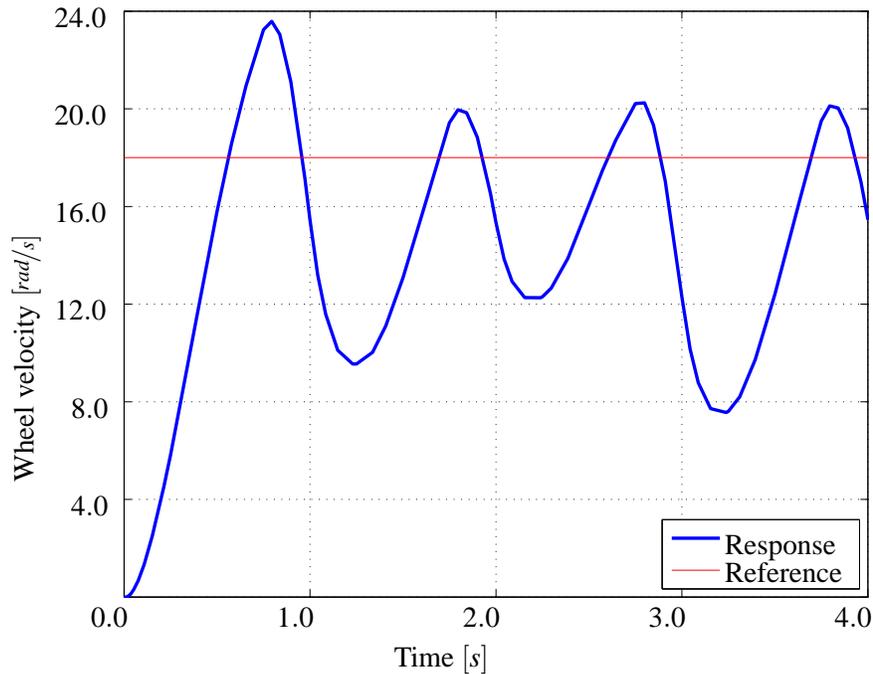


Figure 8.4: Step response to $18[\text{rad/s}]$ with proportional gain 10 and sampling frequency $4[\text{Hz}]$.

The integral gain, that utilizes the past error, is an element that can remember the previous errors through integration, which allows the controller to cancel out any long term errors. The integral term alone will not function as a controller, because the memory of the integrator will keep accelerating the motor, much alike a great overshoot. This effect will also be present if the value of the integral value is set too high in the PID controller. The integrator element is to be designed so it will compensate for disturbances, that will be accessible through the accumulated error. Combined with the proportional element can the response be both fast and have only a little overshoot, though a longer settling time might occur. To illustrate the effect of a integral gain, a plot with a proportional gain of 1 and integral gain of 3, stepped to $18[\text{rad/s}]$ and sampling frequency $4[\text{Hz}]$ has been given in figure 8.5. From the figure it is seen that the integral gain can have a devastating effect on the settling time and overshoot. With a higher integral gain the oscillation will resemble a marginal stable system.

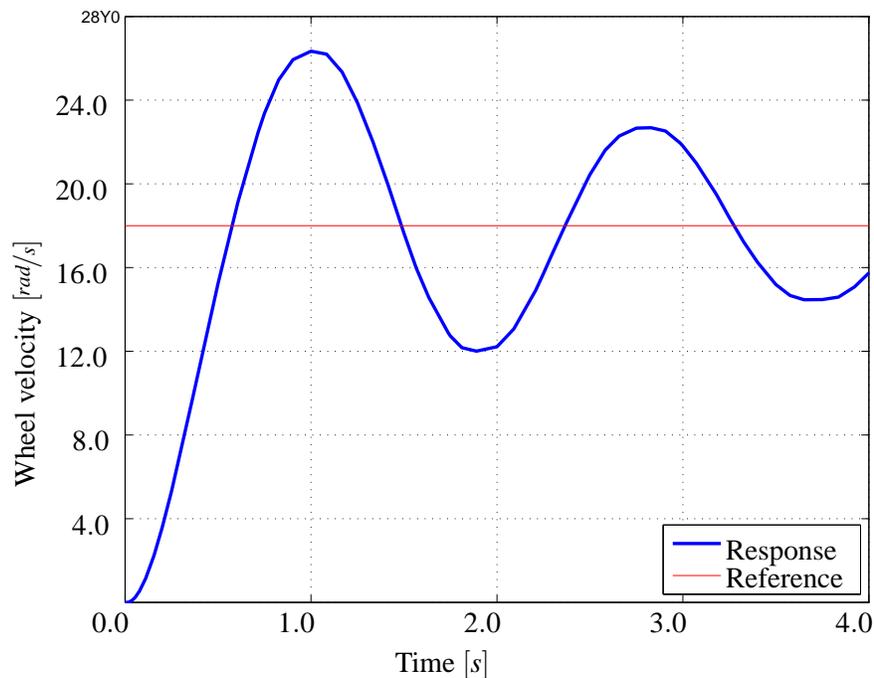


Figure 8.5: Step response to $18[\text{rad/s}]$ with proportional gain 1, integral gain 3 and sampling frequency $4[\text{Hz}]$.

The derivative gain is an element that tries to predict the future output and compensates accordingly. The derivative element can be problematic as it, in this case, will amplify the acceleration of the motor. This can reduce the overshoot greatly, but the tradeoff is, that it also amplifies the noise in the system. The noise can be handled by introducing a low-pass filter on the feedback signal, though this will reduce the effectiveness of the derivative element. A low-pass filter will not be in scope of this project. To illustrate the effect of a derivative gain, has a plot with a proportional gain of 1 and derivative gain of 3, stepped to $18[\text{rad/s}]$ and sampling frequency $4[\text{Hz}]$ been given in figure 8.6. From the figure it is seen that the derivative gain can yield a great steady state error, which is highly undesirable.

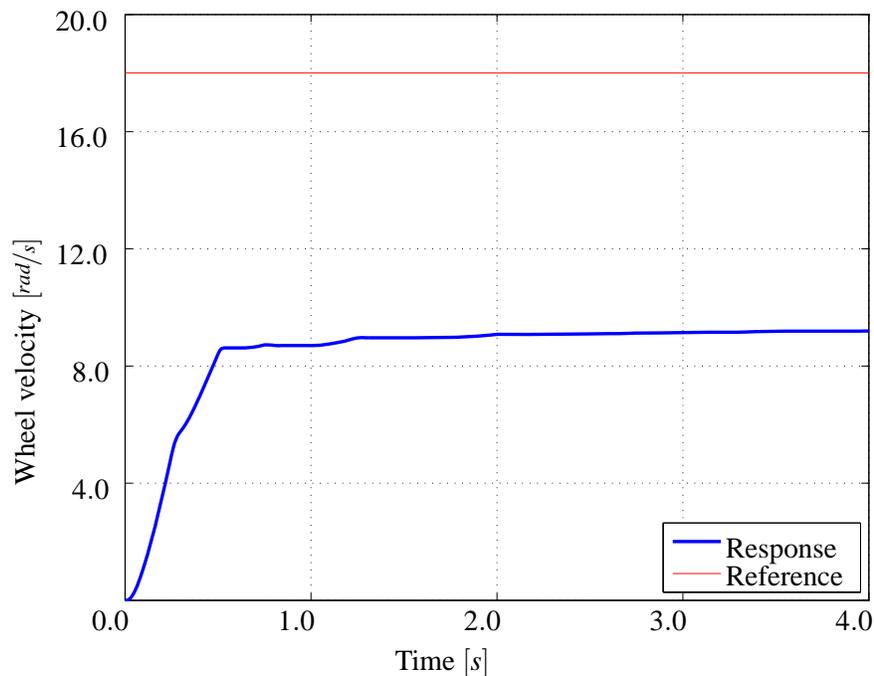


Figure 8.6: Step response to $18[\text{rad/s}]$ with proportional gain 1, derivative gain 3 and sampling frequency $4[\text{Hz}]$.

The numeric values for the three PID elements are to be determined. This can be done in several ways. For instance can the Ziegler-Nichols Tuning Rules be used. There are two methods, where the first method is preferable for the system as it specifically applicable to a system yielding a S-shape step response as shown in figure 6.11. Instead of using tuning rules or manual tuning a transfer function can be developed for the plant and a controller can be developed analytically [Ogata, 2002]. Due to the project time perspective, it is chosen to set the gains manually in the PID controller implemented in the motor model in the final model. This is done to ensure the motors experiences a load.

The criteria for the controller is to have rise time of a minimum of $0.5[\text{s}]$ and maximum of $1.0[\text{s}]$ when stepping from rest to full speed at $\approx 27[\text{rad/s}]$, which is the maximum recorded velocity of the wheels during experiments. This is $\approx 14[\text{km/hr}]$, which is well faster than the $3 - 4[\text{km/hr}]$ specified earlier. A faster rise time is undesirable, as this could lead to spinning of the wheels which does not contribute to movement of the robot, and is unwanted. The boundary values given for the rise time are merely estimated from observation. This means that when the controller is implemented, and the wheels spin when the robot is tested in an outdoor environment, the PID gains needs to be adjusted to eliminate the spinning.

Also a criteria of a maximum overshoot of 20% is chosen. The overshoot parameter is not of great concern, though a great overshoot of 50% is undesirable, as great unwanted variations in motor speed will not be accounted for in the navigation controller. It should be possible to avoid overshoot due to poor sampling by adjusting the controller. A third design parameter is the settling time. The maximum settling time, where the reference is within 5% of the steady state velocity, is set to $2[\text{s}]$, but a shorter settling time is preferred [Phillips and Harbor, 2000]. The settling time is often used to get an idea of when

an oscillating system settles, though here it will serve as a time for when a given reference is nearly met. The boundaries set for the step response parameters are made from intuition, keeping the final system, with navigation control, in mind. As the boundaries are made from intuition, these are to be reevaluated when the complete robot system is running.

Before adjusting the PID gains an addition to the block diagram in figure 8.3 is made. The addition consist of a zero order hold block, in which the sampling time can be set. The zero order hold block is placed on the velocity feedback line. This is done to emulate the feedback from the encoders as their signals are not continuous velocity signals. In figure 6.10 on page 59 shows a velocity data sampling marked with red. It can easily be seen that he plot is comprised of data points that are connected by linear segments. The poor resolution of the encoders is due to a slow sampling rate. In the plot the sampling time is $50[ms]$. The high sampling time is necessary as the encoders only have five poles per revolution of the armature. Setting the sampling frequency faster will result in a flickering of the velocity signal at steady state velocity. Figure 8.7 shows a step input for a motor, from the final model, shown with a continuous feedback signal and a sampled feedback signal. The step inputs, shown in figure 8.7, have been given a step input reference of $18[rad/s]$ and PID controller gains are set to zero, apart from the proportional gain that has been set to 4, to illustrate the effect of the sampling time. For the shown step responses, all four motors have been given the same input and PID gains are equal, meaning that the vehicle is driving in a straight line. The sampled feedback signal time, in the plot, has been set to $250[ms]$ to clearly illustrate the influence of the slow sampling.

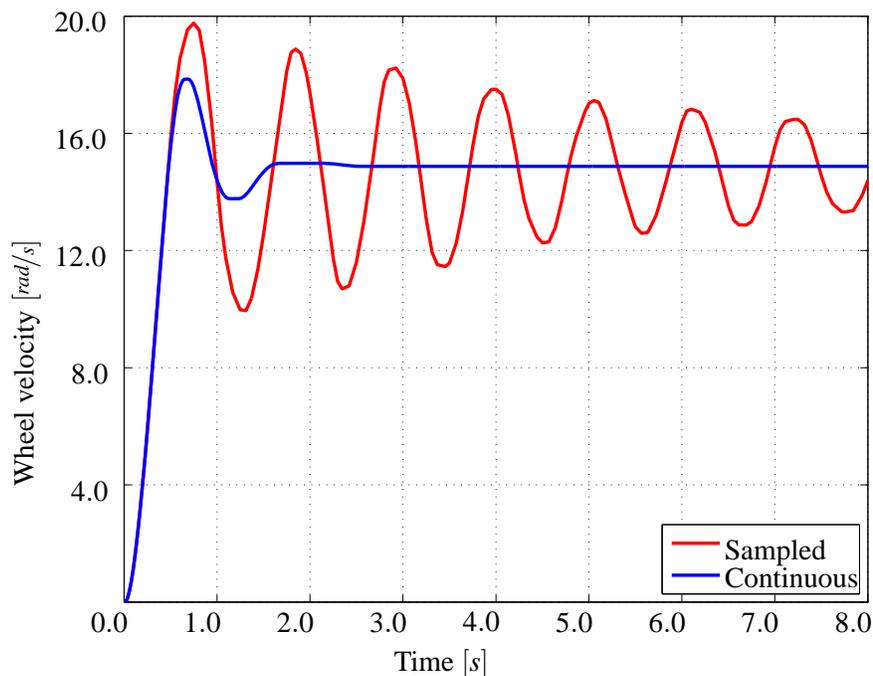


Figure 8.7: Simulated step inputs with and without zero order hold feedback.

It is seen in figure 8.7 that the sampling time has great influence on the wheel velocity when using a

controller. The longer the sampling time is, the longer the controller will correct for the same error measurement. This is also the reason why the overshoot is much greater. It is also seen, that the response with long sampling time has a very long settling time, well beyond the $2[s]$. By adjusting the integral and derivative gains, the controller can compensate by utilizing the past and future error. As stated earlier, the PID gains will be adjusted manually in the final model, in order to obtain an acceptable response. Instead of tuning the parameters manually, the Ziegler-Nichols method could be applied. The tuned parameters are given in table 8.1. The parameters are by no means final, as further iteratively tuning would yield better step responses. Tuning is also necessary when implementing the controller into the real system, as mentioned earlier the model differs from the real system.

Proportional Gain K_p	Integral Gain K_I	Derivative Gain K_D
2.1	1.9	0.6

Table 8.1: PID final model gains.

A step input using the PID gains is shown in figure 8.8. The only alteration from figure 8.7 to figure 8.8, other than the PID gains, is that the sampling time for the encoders are set to $50ms$, as this is the sampling time vicinity that has been found to have the best tradeoff between precision and frequency of error passing.

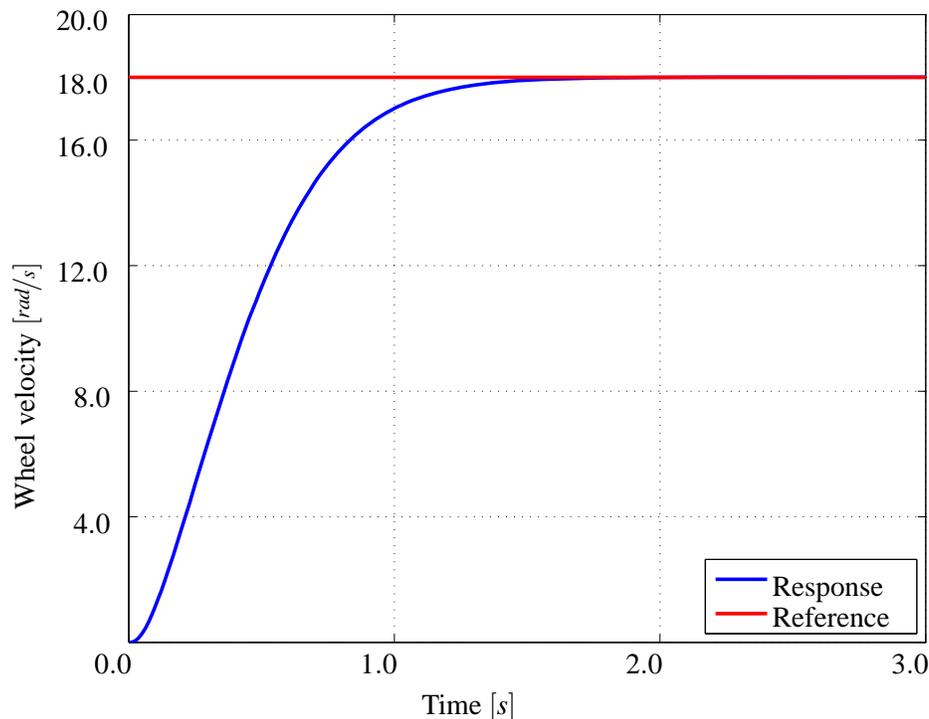


Figure 8.8: Step response of PID controlled motor.

With the implemented and tuned PID controller, in figure 8.8, it is seen that the motor follows the reference well and is relatively fast. A settling time of $1.02[s]$ has been measured, which is within the boundaries set earlier. If the settling time criteria of 2% is used, which is the other settling time criteria

often used, the settling time is $1.2[s]$. It can also be seen that the overshoot is no longer present with the PID controller, which makes the boundary redundant. In figure 8.9 is the same plot given, though the time scale is shorter to illustrate the transient portion of the step response.

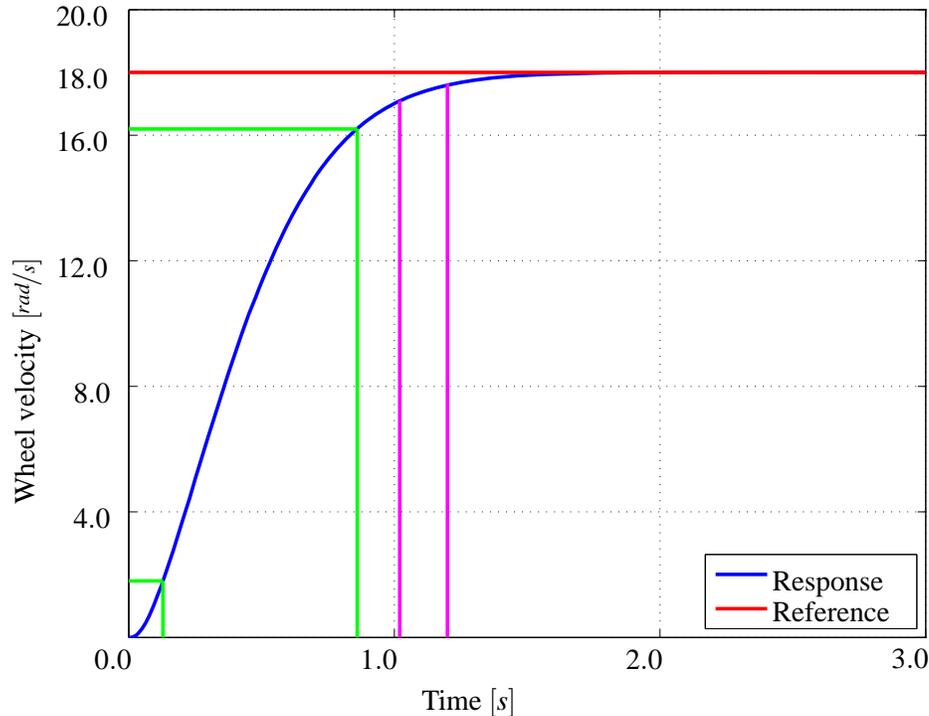


Figure 8.9: Step response of PID controlled motor with shorter time axis.

In figure 8.9 the rise time is marked with green lines, and is measured to be $0.73[s]$, which is within the boundaries set earlier. The first green line set denotes the 10% mark, and the second set marks the 90% mark. The 2% and 5% settling times mentioned earlier are marked with magenta lines in figure 8.9. The PID controller is also tested for step input with $10[rad/s]$ and the maximum velocity of $27[rad/s]$, which is illustrated in *appendix E.1 Step Responses using PID*.

From all the step responses shown it is concluded that the PID controller, with the gains shown in table 8.1, is an acceptable motor controller. In all the responses, with the controller implemented, the criteria parameters has been determined to be within the set parameters. The controller has only been tested in the final model, and is yet to be implemented into the real system. When the controller is implemented, the gains are adjusted again in order to yield an acceptable output. A clear difference can be seen when setting the sample frequency of the encoder feedback. In the model this is a continuous feedback that is sampled afterward. Though this is not the case in the real system. In the real system the encoders have a resolution of 190 ticks per revolution, which compared with the continuous model, is a further deterioration of the feedback signal. This deterioration might have a noticeable effect when using the PID controller in the real system. It is also due to the deterioration that the gains will require readjustment. The step response criteria and the values of the three responses are shown in table 8.2.

Criteria	Boundary	10 rad/s	18 rad/s	27 rad/s
Rise time	0.5 – 1.0[s]	0.8[s]	0.73[s]	0.99[s]
Settling time	< 2[s]	1.27[s]	1.02[s]	1.36[s]
Overshoot	10 – 20%	–	–	–

Table 8.2: Step response criteria and values.

As mentioned earlier, the motor controller will function as inner controllers in a cascaded system, where the outer controller is described in section 8.2 *Navigation Controller*. When using controllers in cascade, it is required that the inner controller is faster than the outer. If not, the system behavior can be unpredictable making it oscillate. The sampling frequency of the encoders are very important in this matter. As a higher sampling frequency will lead to a faster motor controller. The feedback to the navigation controller is also important as the controller will be setup to yield motor controller references for each data set fed back from the laser range scanner. To slow down the navigation controller a certain amount of feedback data can be discarded, lowering the motor controller reference frequency. Lowering the frequency of the navigation controllers output is unwanted, as it will have an effect on the top velocity of the vehicle. If the navigation controller is slow then the robot needs to travel at a slower pace in order for the controller to keep up. Setting the PID gains to obtain a very fast controller will also yield an unwanted effect in the form of wheel slipping. As mentioned earlier this is unwanted. Therefore it is necessary to find a tradeoff in controller speeds, when the controllers are implemented.

8.2 Navigation Controller

The navigation controller is, as mentioned earlier, the outer closed loop controller. The navigation controller will utilize the angular feedback signal $\theta_{e,ref}$ and information about candidate valley width, to generate motor velocity references in order to reduce the error in relation to a reference. The navigation controller will be formed as a hybrid system controller in order to cope with the different situations the robot may find itself in during the Field Robot Event .

A hybrid system controller can be formed as shown in figure 8.10.

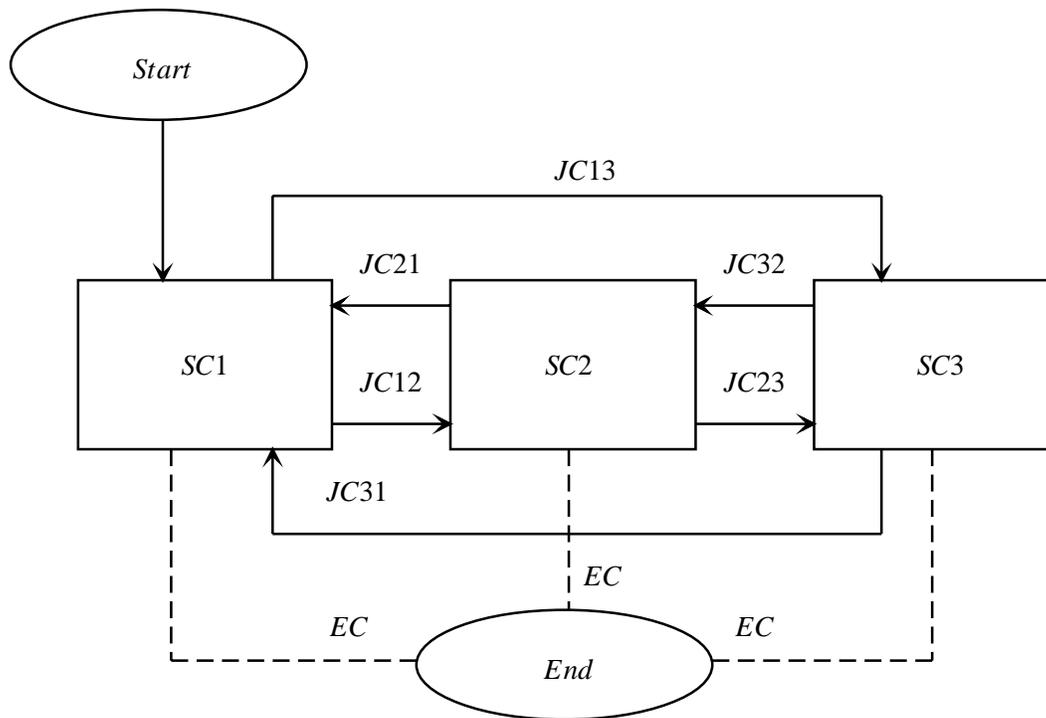


Figure 8.10: Diagram of a hybrid system controller algorithm.

In the figure the oval named *Start*, illustrates the situation the controller is in, when the start button is pushed. The controller then jumps to control algorithm *SC1*, which denotes *Switch Controller 1*. Each of the *Switch Controllers* have arrows pointing to the others. These arrows, denoted *JC*, symbolize *Jump Conditions*. *Jump Conditions* are special cases that are enabled by data detected by the *LRS*, which determines whether another *SC* is to be used. *JCs* will be activated if the *LRS* data exceeds preset threshold values. Furthermore does the *SCs* have dashed lines leading to the *End* condition. These dashed lines denotes a condition inside each *SC* that will end the navigation controller, outputting zero references to the motor controller, and brings the robot to a halt. The *ECs* could be based on a situation where robot has met a dead end or it could be time based. The hybrid system is to function as an autonomous controller where the only human input is the *Start* command. In figure 8.10 can three *SCs* be seen. This is just an estimate, the hybrid system controller can have a minimum of two and a maximum that is only restricted by the hardware and software used in the robot. The *mark II* is thought to have three different *SCs* initially, which are described in the following. More *SCs* can be added if deemed necessary. Figure 8.11 shows an overview of the system with the different *JCs*.

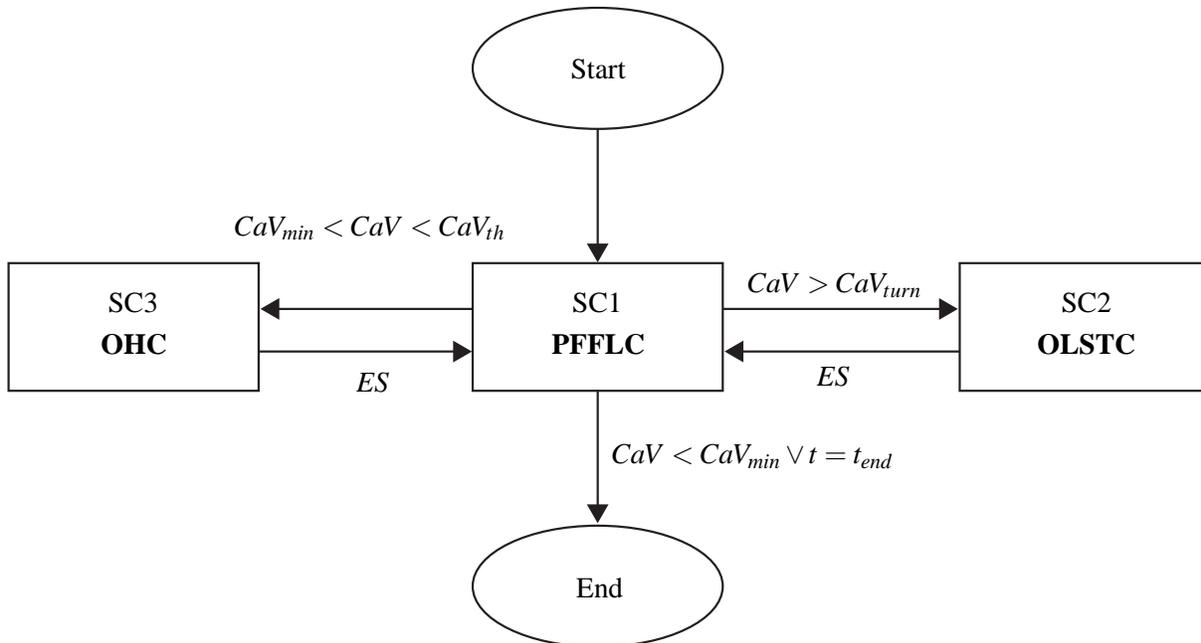


Figure 8.11: Hybrid controller.

PFFLC Path Following Fuzzy Logic Controller.

OLSTC Open Loop Sequential Turning Controller.

OHC Obstacle Handling Controller.

CaV The width of the current candidate valley (CaV) [deg].

CaV_{min} The minimum allowed width of the CaV [deg].

CaV_{turn} The CaVs threshold width for turning [deg].

CaV_{th} The minimum CaV for the robot to be able to pass through without collision [deg].

ES End of Sequence.

t Time.

t_{end} End time.

The control algorithm assumes that the robot has been placed in the corn field, with corn rows to both sides. The first control algorithm is named SC1. This controller is described in section 8.2.1 *Path Following Fuzzy Logic Controller*.

8.2.1 Path Following Fuzzy Logic Controller

Switch Controller 1, named *Path Following Fuzzy Logic Controller*, will control the robot when driving down the field between two rows, where the distance between rows allows for the robot to pass without collision. In this controller the feedback will be the error angle described in section 7 *Navigation*. The decision to choose a fuzzy logic controller (FLC) instead of a classic controller, for the navigation, is based on a few aspects. First of all is a classic controller more time consuming to make than a FLC. Furthermore is the classic controller based on a transfer function of the system which is created from a model of the system. The DVM created in chapter 5 is assessed to be accurate enough to be used for this. It is however assessed, given the intuitive correlation between input and output, that a FLC is better controller for the present application. The controller will yield two motor velocity references that will be passed to the left and right side motor controllers respectively. The larger the error the greater is the difference between the motor controller references. *SC1* will be valid as long as the *LRS* can find a candidate valley that is wider than the *mark II* and is narrower than CaV_{turn} . If the candidate valley is wider than CaV_{turn} , *JC12* will be activated and if the candidate valley is smaller than CaV_{th} , *JC13* will be activated. The controller will have preset motor velocity references, that are equal, for both motor sets. When the controller needs to steer the vehicle, the controller will lower one of the motor controller velocity references, making the vehicle turn in the direction of the lower reference wheel set. The higher the angular error the higher is the difference between the motor controller references. If the angular error is larger than a preset value the controller will give two equal but opposite motor controller references to make the robot turn around its own axis. The *SC1* will be the primary controller in the hybrid system controller, as the robot is thought to spend most time in the corn row passages. The *Path Following Fuzzy Logic Controller* will be developed using fuzzy logic.

8.2.1.1 Fuzzy Logic Controller

Fuzzy logic is a logic based control theory, where the input is evaluated using a set of rules to find the output. Instead of designing a controller on the basis of a model of the system, like traditional control theory, fuzzy logics are derived from experience and knowledge of the present system. Thus Fuzzy logics is a reasonable approach when an exact model is difficult to obtain, but a good understanding of the system has been acquired by experience. Opposed to traditional control theory, fuzzy logic provides a conceptual framework for representation of commonsense knowledge. Fuzzy logic is truth to a certain degree. For instance, when representing a glass of water using classic logic, the glass can either be full or empty. With fuzzy logic the glass can be represented as 0.7 full and 0.3 empty, thus choosing a value between true and false. The amount of the rule, that is applied in a certain situation, is called degree of membership. The downside when using fuzzy logic is the lack of precision. However for many applications, where precision is less relevant, this approach provides a reasonable result [Alavala, 2008]. Figure 8.12 shows the fuzzy logic approach.

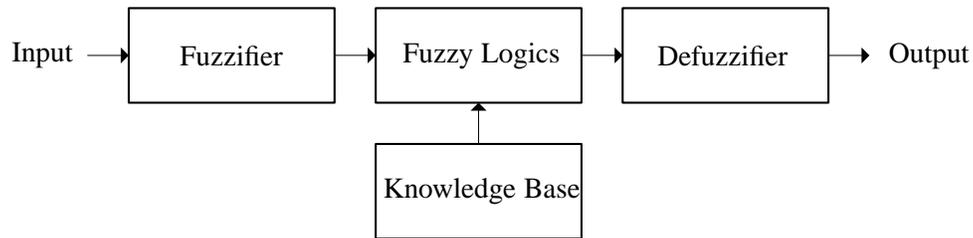


Figure 8.12: Fuzzy logic controller.

The knowledge base consists of a set of *expert rules* for the present system. They need to be established by a person that knows the systems behavior well. For the present system the rules are shown below:

- If *Angle error* ($\theta_{e,ref}$) is *High Left* then *Reverse Left* and *Forward Right*
- If *Angle error* ($\theta_{e,ref}$) is *Left* then *Forward Left* and *Full Forward Right*
- If *Angle error* ($\theta_{e,ref}$) is *Middle* then *Full Forward Left* and *Full Forward Right*
- If *Angle error* ($\theta_{e,ref}$) is *Right* then *Full Forward Left* and *Forward Right*
- If *Angle error* ($\theta_{e,ref}$) is *High Right* then *Forward Left* and *Reverse Right*

Next step is to create the fuzzifier for the input and two defuzzifiers for the outputs. The fuzzifier and defuzzifier shown in figure 8.13 and 8.14 are representations of the rules degree of membership for a given input. For instance with an input angle error of -10° the membership is 0.5 *Left* and 0.5 *Middle*. The outputs are defined to be a value between -1 and 1 where -1 is *Full Reverse* and 1 is *Full Forward*. The model has one output for the left and one for the right motor set. Defuzzification is done using the *center of gravity* method. This is a way to compute the output from the different memberships. This method will not be described further.

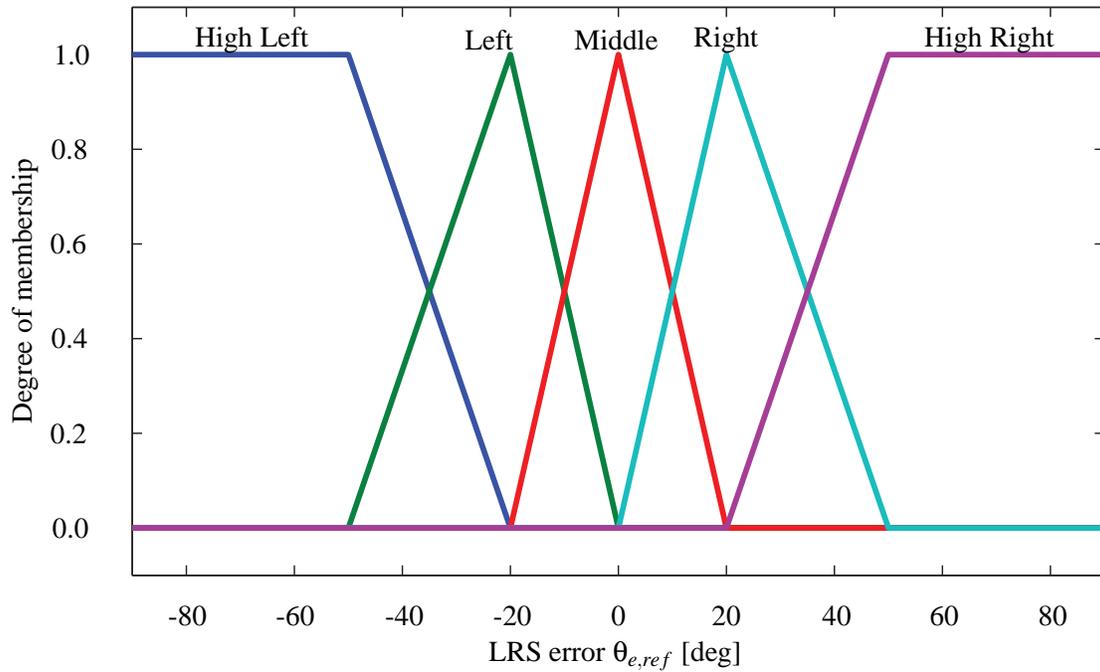


Figure 8.13: Fuzzy set used for fuzzyfication of the angle error θ_e .

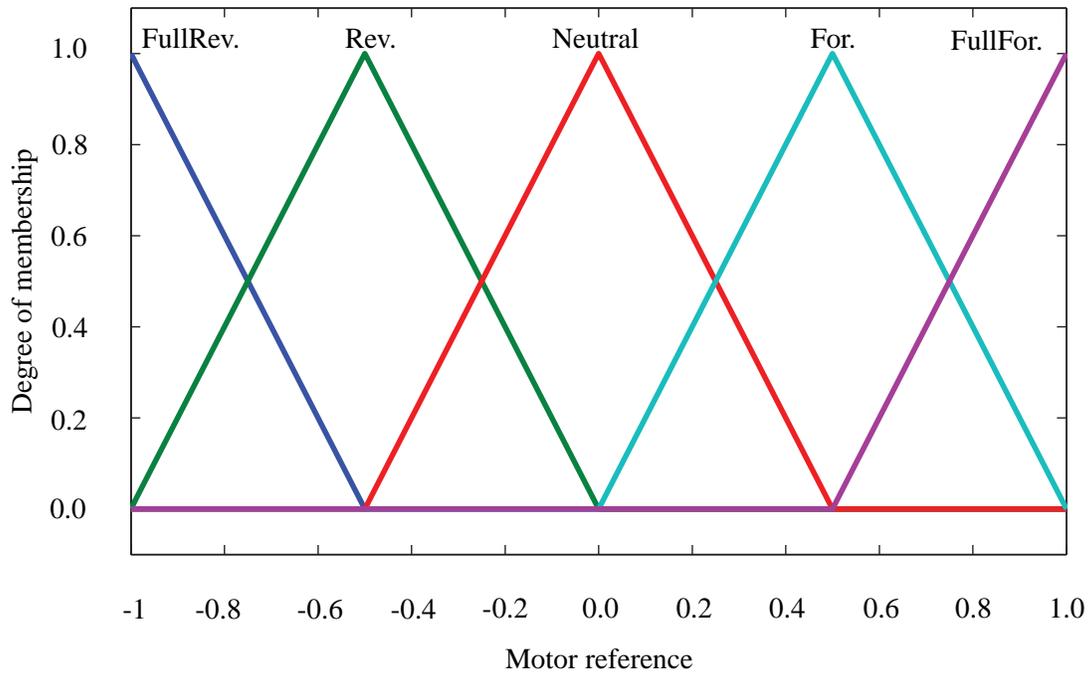


Figure 8.14: Fuzzy set used for defuzzyfication to get motor references.

Figure 8.15 shows a representation of the controllers input and output. Here it can be seen that the controller reaction increases, when the error grows and when the error is larger than $\pm 50^\circ$, the robot will rotate around its own axis. When implementing the controller it is easy to change the output if the

robots reaction is too small or too large.

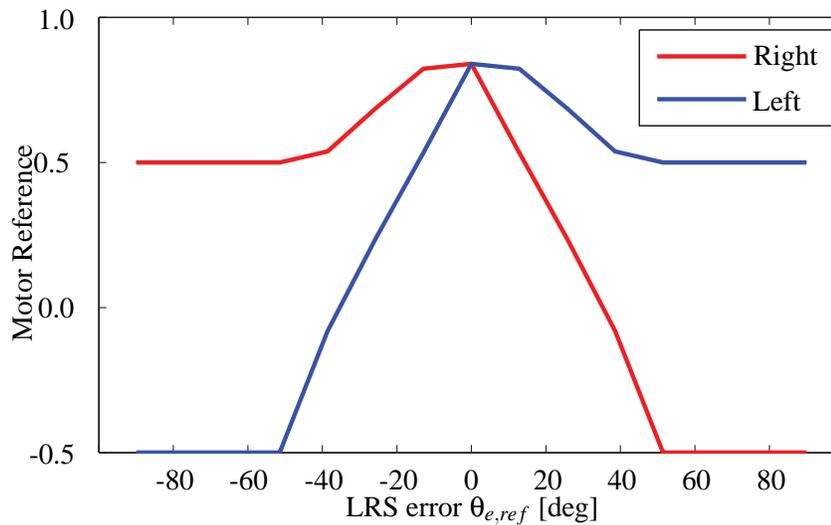


Figure 8.15: The input and the two outputs of the fuzzy logic controller.

8.2.2 Open Loop Sequential Turning Controller

Switch Controller 2, named *Open Loop Sequential Turning Controller*, will control the robot when it comes to an end of a corn row passage. The *SC2* will be activated when the candidate valley is larger than the preset threshold angle CaV_{turn} . This controller will steer the vehicle into the next corn row passage. *SC2* can function in two ways. The first, that will be used in this project, is an open loop controller governed by time instead of the angular feedback used in *SC1*. The open loop path is sketched in figure 8.16.

An overview of the sequence is given in following.

1. Drive straight with motor controller reference ω_1 for time period T_1 .
2. Turn with equal and opposite reference ω_2 for time period T_2 , utilizing Boolean variable B_2 .
3. Drive straight with motor controller reference ω_1 for time period T_3 .
4. Turn using variables from step 2.
5. Drive straight as in T_1 .

When step five is completed, the robot should be in front of next path and the system will jump back to *Path Following Fuzzy Logic Controller* after the preset time. It is assessed that the robot will be able to find a path between the corn rows and follow it.

When *JC21* is activated a Boolean operator will change value. This value will determine whether *SC2* is to make a right or left turn at the end of a passage. This is done assuming that the robot is to make a

weaving pattern through the corn field as illustrated in figure 1.1 on page 3. A figure of the first strategy can be seen in figure 8.16. In the figure the three horizontal lines denote corn rows forming two paths. For simplicity, the slopes of the rows has been neglected in the illustration. The lower path is the path that the robot is leaving and the upper is the path that the robot is driving to. The vehicle is drawn as a square with an arrowhead depicting the forward direction where the *LRS* is mounted.

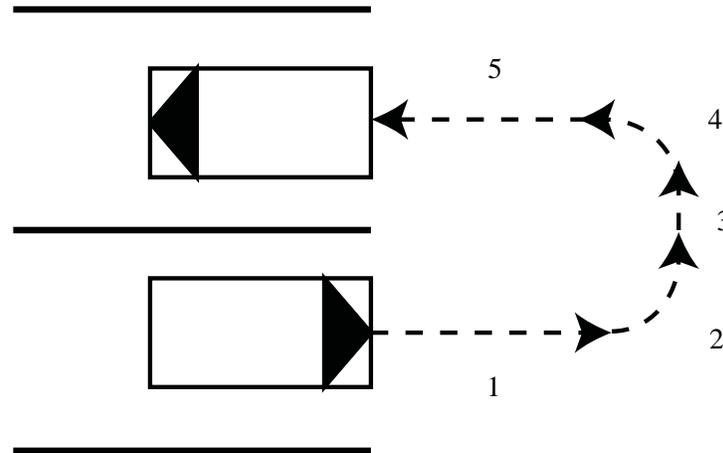


Figure 8.16: Open loop controller *SC2* strategy 1 sketch.

The second strategy for the *SC2* has the advantage that it is a closed loop controller, which can yield a significantly smaller margin of error. The strategy will use the *LRS* to get into the next row, though still use the Boolean variable to decide turn direction. The strategy will choose the corn row to the side it is turning to. The last measured point in the chosen row will function as the center of the semi-circle that will lead to the next corn row. The strategy requires an algorithm that can identify the end of a row, and compute the error between the distance to than point and the desired turning radius. *SC2* will give references to the motor controller, based on the error computed by the algorithm.

8.2.3 Obstacle Handling Controller

Switch Controller 3, named *Obstacle Handling Controller*, is a controller almost similar to *SC1*. This controller will be activated if the candidate valley is too narrow for the robot to pass through, by *JC13*. A candidate valley that is too narrow could be due of the narrowing of the corn row itself, or because an obstacle is in the way e.g. a rock. When the *SC3* is activated, the robot will drive at a lower speed. The *LRS* will know if the passage is to the left or right of its own orientation. The narrow passage location will be recorded into a variable. The variable can assume the values -1 , 0 and 1 . Where:

-1 denotes that the narrow passage location is below a set threshold value.

1 denotes that the narrow passage location is above a set threshold value.

0 denotes that the narrow passage location is between the two threshold values. In this situation will the robot pass through the candidate valley at low velocity, but will not turn, as with the other values

of the variable. The two threshold values will be met when the center of the angle to the *NCV* is greater than a preset angle at a given distance.

By knowing this, the robot will attempt to drive through the narrow candidate valley and steer back toward the path it was on. This is done because when the robot is driving through the narrow passage the *LRS* data will be discarded and the robot will drive open loop as in *section 8.2.2*. Though this will be done with motor controller references predetermined to make the vehicle turn slightly. The turn direction of the predefined motor controller references will be determined by the variable mentioned earlier.

With all the mentioned conditions, controllers and values the hybrid system controller are formed as shown in figure 8.11. When the controller is implemented are values and thresholds to be evaluated. Test runs of the robot will also make apparent if any *SCs* are missing to handle certain situations the robot may encounter. For instance has there not been developed a controller that can handle the situation of the robot driving into a ditch. Also the situation of plants missing is uncertain. It might not trigger any of the jump conditions and continue in *SC1* and it might see it as an end of its current passage and try to perform the five step closed loop algorithm. A *SC* for this situation is still to be made. If the controller, described in this section, works as assumed, the robot will come a long way through the course given in figure 1.1.

Conclusion

The objective of this project has been to develop an autonomous mobile robot capable of navigating through a corn field. To do this a robot platform that fulfilled the setup specifications was acquired. Through a selection process it was concluded, that the *Dr. Robot Jaguar* was the best choice. The *Dr. Robot Jaguar* is a robust platform with large off-road wheels. The chassis is a closed box, which makes it weather resistant. The *Dr. Robot Jaguar* has four wheels with the axles fitted directly into the chassis. This configuration makes the robot a skid-steered robot. Through selection processes and inspection of available equipment the remaining parts of the *Dr. Robot Jaguar* were chosen. By choosing the *Dr. Robot Jaguar* and by selecting components with no internal moving parts it was concluded that necessary precautions were taken, in order to make the robot able to operate in a rough outdoor environment.

Knowing that the robot is a skid-steered vehicle, several research papers were found on the subject. This was done to create a knowledge base for the design of a dynamic vehicle model, that could describe the dynamics of the vehicle. This proved to be a difficult task. Three different models were created based on three research papers. One of the dynamic vehicle models worked and was chosen for further use in the project. The dynamic vehicle model was verified against recorded data of the real system. This was done by placing a target on the robot, and use a tracking program developed for the laser range scanner to track the target. Model and reality showed some similarity, though over time the position error increased. This was thought to be because the parameters in the dynamic vehicle model were not fitted sufficiently, and because the dynamic vehicle model did not account for off-road terrain. The dynamic vehicle model was originally developed for a skid-steered vehicle driving on a smooth 2D indoor surface. When driving in an off-road terrain bulges, rocks and small holes would alter the position of the robot. This was thought to be the main contributor to the position error, as such errors would accumulate over time.

The motors mounted on *Dr. Robot Jaguar* have also been modeled. Parameters in the motor model have been found through a series of experiments. The motor model was verified against recorded data of the motors. The response of the motor model was not similar to the recorded data, and the experimentally found parameters were altered to obtain similarity. The necessity of altering the parameters was most likely due to inaccurate experiments. The experiment done, to find a value of the inductance of the armature windings, yielded a wrong value. This calls for other experiments to obtain it. Through usage of the system, it became clear that the resolution of the encoders was very poor. This was due to the construction of the encoders, and there was little to be done for better resolution. By setting the sampling frequency high, the resolution became very poor, and when setting it low the resolution improved but the interval between samples became too high, which was undesirable for controller design purposes. The best apparent tradeoff between sampling frequency and precision was estimated and used.

Methods of navigation were investigated and the algorithm Vector Field Histogram was used. The method was found to be the best option, and it was compatible with the software used for laser range

scanner. By using the Vector Field Histogram it was possible to find the best available trajectory through an error angle, which is a simple feedback signal to use in a controller compared to a long sequence of polar coordinates.

On the background of the information gathered of the robot system, a controller was developed for the robot. The first controller developed was a motor controller. It was cascaded as a closed loop controller inside the navigation controller. The motor controller was chosen to be a PID controller and it was inserted into a model, consisting of both the motor model and the dynamic vehicle model. The model environment was chosen because it would provide a more realistic load on the motors, when designing the motor controller. Each of the three gains was investigated individually, and it was concluded that all three parameters were necessary. It was discovered that the encoder resolution and frequency had a noticeable impact on the wheel velocities, when the motor controller was implemented. A PID controller was designed that met the criteria.

The navigation controller was designed as the outer controller in the cascade. The navigation controller utilized the angular error feedback and candidate valley width, found with the Vector Field Histogram method. The navigation controller took the form of a hybrid system controller. This was done because it was necessary to handle situations arising in the field differently. The situations in the field were divided into three. The first and most important was the controller that allows the robot to navigate through a corn field passage. The controller utilized the angular feedback from the Vector Field Histogram. The feedback was used in a fuzzy logic controller, that generated motor controller references. The controller was designed in such a way, that it could handle large error angles and could turn around its own axis if the error angle was too large. The fuzzy logic approach to create a controller did compensate for the lack of a transfer function describing the dynamics of the vehicle. If a continuous controller was wanted for future use of the robot it is necessary to develop a transfer function of the plant, which was not in the scope of this project report.

Furthermore two controllers were described for future use in the hybrid system controller. One to handle the situation where the robot has come to an end of a corn row passage and needs to turn into the next. This controller was described as an open loop controller. The other controller was described in such a manner that it partly utilized the fuzzy logic controller and partly functioned as an open loop controller. By using the controller setup described previously, it is concluded that the robot system built is capable of navigating through a field autonomously, until meeting an end condition.

The controllers designed in the project report are still to be implemented into the assembled robot system in order to make the *mark II* ready to compete in the *Field Robot Event 2011*.

Discussion 10

In this chapter, experiences gathered through the project are discussed, and recommendations for future participants of the *mark II* project are given.

Encoders with higher resolution The encoders measurements is a tradeoff between precision and sampling frequency. During the measurements of the motors step response and deceleration, it was realized that the encoders resolution was too low. This is also an issue when using the encoders as feedback in the closed loop motor controllers in the cascade controller. This is because the inner loop requires a higher sampling frequency, than the outer loop. Therefore will a poor sampling frequency yield an overall, slower system. Encoders with higher resolution or another measuring scheme is required in order to obtain a better system performance.

Controller based on transfer functions The fuzzy logic controller designed in this project is based on intuition and experience. This yields a reasonable controller that solves the problem. To obtain a high performance controller, the controller must be designed from a transfer function, created from an accurate model of the system. A better dynamic model is required in order to get an accurate transfer function.

Wireless control When setting test parameters on the robot, a cross cable is used. A more convenient solution would be to have wireless access to the robot. That way the robot could also be shut down remotely.

Better LRS The Laser Range Scanner sometimes malfunction during conduction of experiments outside. The manufacturer deprecate the use of the LRS outside, so therefore another LRS is recommended.

IMU Implementation of an Inertial Measurement Unit could yield some improvements to the system. The Yaw sensor could be used during turning, and the tilt and roll could provide information about obstacles, inclines and ditches.

Separate power Supply The linear regulator used for the 5[V] supply wastes a lot of power and therefore gets very hot. A bigger heat sink or a transformer is recommended. A transformer would waste less power and require less cooling.

Bibliography

- Aalto University [2010], '*Turtle Beetle*', [Online]. [Cited 10. Feb 2011]
<http://autsys.tkk.fi/en/FieldRobot2010>.
- Alavala, C. R. [2008], *Fuzzy Logic and Neural Networks*, New Age International Publishers. ISBN: 8122421822.
- Andersen, T. O. and Pedersen, H. C. [2010], *Laboratory Exercise - Electrical DC servo motor*, Technical report, Aalborg University.
- Araki, M. [*], 'PID control', *CONTROL SYSTEMS, ROBOTICS, AND AUTOMATIONS Vol. II*. Kyoto University, Japan.
- Borenstein, J. and Koren, Y. [1991], '*The Vector Field Histogram-Fast Obstacle Avoidance for Mobile Robots*', [Online]. [Cited 6. Apr 2011]
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=88137>.
- Centinkunt, S. [2007], *Mechatronics*, 1. edn, John Wiley & Sons, 111 River Street, Hoboken, NJ. ISBN: 978-0-471-47987-1.
- Dr. Guillermo Rico [Fall 1996], 'A Simple Procedure for Measuring Inductance', [Online]. [Cited 3. May 2011]
<http://engr.nmsu.edu/~etti/fall96/electronics/induct/induct.html>.
- Dr. Robot [2011], '*Jaguar-4x4-wheel Specification*', [Online]. [Cited 9. Feb 2011]
http://jaguar.drrobot.com/specification_4x4w.asp.
- Emil Fibiger Petersen [2010], 'Helikoptere mod ukrudt', [Online]. [Cited 17. Mar 2011]
<http://www.nordjyske.dk/aalborg/forside.aspx?ctrl=10&data=28%2c3543670%2c2875%2c4>.
- Frank van Gennip et al. [2010], '*The Challenges of Rapidly Developing an Autonomous Robotic System*', [Online]. [Cited 10. Feb 2011]
<http://www.ecnmag.com/blog/2010/09/research/Autonomous-Robotic-System.aspx>.
- FRE2011 [2011], '*FieldRobotEvent2011*', [Online]. [Cited 8. Feb 2011]
<http://fre2011.dk/Field%20Robot%20Event%202011.pdf>.

- Fritsen, P. G., Justesen, B. D., Vangsgaard, N. A., Rice, K. S., Hansen, L. J. R. and Nissen, C. F. [2010], *Mark I*, Technical report, Aalborg University.
- Gears Educational Systems [2011a], '*Heavy Metal Articulating Chassis*', [Online]. [Cited 9. Feb 2011]
<http://www.gearseds.com/hmac.html>.
- Gears Educational Systems [2011b], '*The Surface Mobility Platform*', [Online]. [Cited 9. Feb 2011]
http://www.gearseds.com/surface_mobility_platform.html.
- Henrik Skov Midtiby [2011], '*Robotter i landbruget og villahaven*', [Online]. [Cited 11. Mar 2011]
http://www.dr.dk/DR2/Danskernes+akademi/IT_teknik/Robotter_i_landbruget_og_villahaven_1.htm.
- Hokuyo Automatic [2009], '*Scanning Laser Range Finder*', [Online]. [Cited 17. Mar 2011]
http://www.hokuyo-aut.jp/02sensor/07scanner/download/data/URG-04LX_UG01_spec.pdf.
- Hokuyo Automatic [2010], '*Scanning range finder (SOKUIKI sensor)*', [Online]. [Cited 17. Mar 2011]
http://www.hokuyo-aut.jp/02sensor/07scanner/urg_04lx_ug01.html.
- Holland, J. [2004], *Designing Autonomous Mobile Robots*, Elsevier. ISBN: 0-7506-7683-3.
- Ingeniøren [2008], '*Robotter bliver dansk landbrugs nye Ferguson*', [Online]. [Cited 11. Mar 2011]
<http://ing.dk/artikel/93771-robotter-bliver-dansk-landbrugs-nye-ferguson>.
- Kjeld Jensen [2011], '*Autonomous Agriculture Robot - MARK*', [Mail]. Copy of mail is enclosed on DVD in Bibliography/Jensen.Kjeld.2011.pdf.
- Kozłowski, K. and Pazderski, D. [2004], 'Modeling and Control of a 4-Wheel Skid-Steering Mobile Robot', *J. Appl. Math. Comput. Sci.* **14**(04), pp. 477–496.
- Landbrugsavisen [2006], '*Målet er at reducere pesticid-forbruget*', [Online]. [Cited 11. Mar 2011]
<http://www.landbrugsavisen.dk/LandbrugsAvisen/2006/1/19/Maalet%20er%20at%20reducere%20pesticid-forbruget.htm>.
- Mandow, A., Martínez, J. L., Morales, J., Blanco, J. L., García-Cerezo, A. and González, J. [2007], 'Experimental kinematics for wheeled skid-steer mobile robots'. [Cited 20. Apr 2011]
<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4399139>.
- Martin Dlouhý [2010], '*Field Robot 2010*', [Online]. [Cited 10. Feb 2011]
<http://robotika.cz/competitions/fieldrobot/2010/en>.
- Michael Tse [1998], 'Voltage and Current Division Formulae', [Online]. [Cited 6. May 2011]
http://www.eie.polyu.edu.hk/~cktse/linear_circuits/main/node15.html.

- Miljøministeriet [2010], 'Aftale mellem Regeringen og Dansk Folkeparti om Grøn Vækst 2.0', [Online]. [Cited 16. Mar 2011]
http://www.mim.dk/NR/rdonlyres/89DD7552-E925-4CA0-8E5B-C5072D40D18D/0/GroenVaekst2_0.pdf.
- Mohan, N., Undeland, T. M. and Robbins, W. P. [2003], *Power Electronics - Converters, Applications and Design*, 3. edn, John Wiley & Sons, New York. ISBN: 978-0-471-22693-2.
- Ogata, K. [2002], *Modern Control Engineering*, Prentice Hall. ISBN: 0-13-043245-8.
- Phillips, C. L. and Harbor, R. D. [2000], *Feedback Control Systems*, 4. edn, Prentice Hall, Upper Saddle River, New Jersey. ISBN: 0-13-949090-6.
- PSU Robotics [2011], 'Wheel Encoders', [Online]. [Cited 12. May 2011]
http://psurobotics.org/wiki/index.php?title=Wheel_Encoders.
- Rasmussen, P. O. [2010], 'Slides,EA MM3'. slides.
- Robot Power [2008], '*Simple-H User Manual*'. Copy of datasheet is enclosed on DVD in Bibliography/Simple-H-user-manual.pdf.
- Roboterwerk [2011], '*FORBOT*', [Online]. [Cited 9. Feb 2011]
<http://www.roboterwerk.de/en/forbot.html>.
- Serway, R. A. and Jewett, J. W. [2004], *Physics - for Scientists and Engineers - with Modern Physics*, 6. edn, Thomson Learning, London. ISBN: 0-534-40949-0.
- Shuang, G. and Cheung, N. [2007], 'Skid Steering in 4-Wheel-Drive Electric Vehicle', *PEDS* pp. pp. 1548–1553.
- Siegwart, R. and Nourbakhsh, I. R. [2004], *Introduction to Autonomous Mobile Robots*, A Bradford book. ISBN: 0-262-19502-X.
- Solc, F. and Sembrera, J. [2008], 'Kinetic Model Of A Skid Steered Robot', *ISPRA 08* pp. pp. 61–65.
- Ulrich, K. T. and Eppinger, S. D. [2004], *Product Design and Development*, McGraw-Hill. ISBN: 0-07-247146-8.
- Wescott, T. [*], 'PID control', *EE Times India* . FLIR, India.

Part I

Appendix

DVM Equations

In this chapter the remaining equations, that are not given in section 5.2 *Kinetic Model Of A Skid Steered Robot*, are used in the [Solc and Sembrera, 2008] DVM given.

A.1 DVM Equations B3

The total velocity of slipping v_f are found from equation A.1.

$$v_f = \sqrt{v_s^2 + (v^S)^2} \quad (\text{A.1})$$

Where v_s is the velocity of longitudinal slipping and v^S are the lateral velocities. v_s is given in equation A.2

$$v_s = v_{1,2} - v^F \quad (\text{A.2})$$

Where $v_{1,2}$ is the peripheral wheel velocities of either the right or the left side, and v^F is the velocity in the longitudinal direction. The longitudinal and lateral velocities mentioned are calculated in block B2.

A.2 DVM Equations B2

In this section the four equations describing the longitudinal and lateral velocities for the wheels are given. In equation A.3a and equation A.3b the longitudinal velocity equations for the left and right side wheels are respectively.

$$v_1^F = v_4^F = \dot{x} \cdot \cos(\phi) + \dot{y} \cdot \sin(\phi) - \frac{b}{2} \cdot \dot{\phi} \quad (\text{A.3a})$$

$$v_2^F = v_3^F = \dot{x} \cdot \cos(\phi) + \dot{y} \cdot \sin(\phi) + \frac{b}{2} \cdot \dot{\phi} \quad (\text{A.3b})$$

In equation A.4a and equation A.4b the lateral velocity equations for the frontal and rear wheels are respectively.

$$v_1^S = v_2^S = \dot{y} \cdot \cos(\phi) - \dot{x} \cdot \sin(\phi) + p \cdot a \cdot \dot{\phi} \quad (\text{A.4a})$$

$$v_3^S = v_4^S = \dot{y} \cdot \cos(\phi) - \dot{x} \cdot \sin(\phi) - (1 - p) \cdot a \cdot \dot{\phi} \quad (\text{A.4b})$$

In order to keep track of the index numbering figure 5.4 on page 28 can be useful.

A.3 DVM Equations B1

In equation A.5 the three equations describing the dynamics of the chassis can be seen. The longitudinal and lateral adhesion forces are utilized along with chassis constants, wheel track (a), wheel base (b) and parameter (p) denoting the position of the center of gravity relative to the geometrical center. For the *Dr. Robot Jaguar* the center of mass is located nearer the front of the vehicle relative to the geometrical center.

$$M \cdot \ddot{x} = \sum_{i=1}^4 F_i \cdot \cos(\phi) - \sum_{i=1}^4 F_i \cdot \sin(\phi) \quad (\text{A.5a})$$

$$M \cdot \ddot{y} = \sum_{i=1}^4 S_i \cdot \sin(\phi) - \sum_{i=1}^4 S_i \cdot \cos(\phi) \quad (\text{A.5b})$$

$$J \cdot \ddot{\phi} = (F_2 + F_3 - F_1 - F_4) \cdot \frac{b}{2} + (S_1 + S_2) \cdot pa - (S_3 + S_4) \cdot (1 - P) \cdot a \quad (\text{A.5c})$$

Motor Model B

B.1 Unit Comparison of Machine Constant

As K_ϕ is to function as motor constant for both equation 6.1b and equation 6.1c it is necessary to perform a unit check for the equations. From equation 6.1b it is seen that the unit of K_ϕ can be given with the following units:

$$K_\phi = \frac{e_a}{\omega_a} = \frac{[V]}{[rad/s]} = [V] \cdot [s] \quad (B.1)$$

And from equation 6.3 it is seen that the unit of K_ϕ is:

$$K_\phi = \left[\frac{Nm}{A} \right] = \frac{[kg] \cdot \frac{[m^2]}{[s^2]}}{[A]} \quad (B.2)$$

By setting the units describing the two cases K_ϕ equal to each and balancing them out, it should become apparent that they can substitute each other:

$$[kg] \cdot \frac{[m^2]}{[s^2]} / [A] = [V] \cdot [s] \quad (B.3a)$$

$$\Rightarrow [kg] \cdot \frac{[m^2]}{[s^2]} = [W] \cdot [s] \quad (B.3b)$$

$$[kg] \cdot \frac{[m^2]}{[s^2]} = [J] \quad (B.3c)$$

$$\Rightarrow [J] = [W] \cdot [s] \Rightarrow [J] = [J] \quad (B.3d)$$

By the unit comparison,, and from the given sources, it can be concluded that the motor constant can function in both linear coherences in chapter 6 *Motor Model*.

B.2 RL Circuit Analysis

By blocking the motor shaft the motor and setting a known resistor in series, the motor can be perceived as the *RL*-circuit shown in figure B.1.

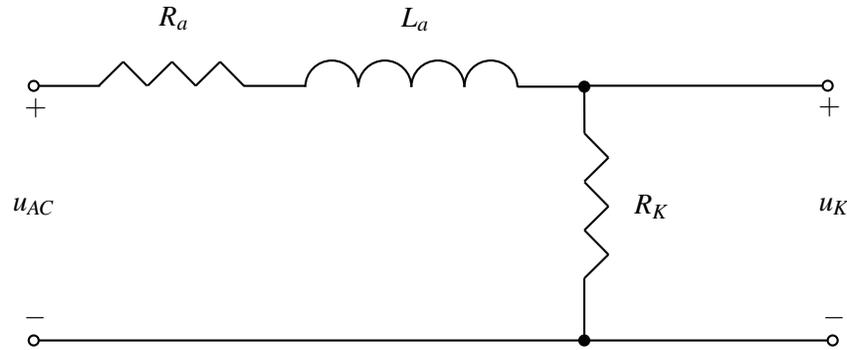


Figure B.1: Circuit diagram of armature inductance experiment.

By using voltage division an expression for the measured voltage u_K can be determined. The equation is found by calculating a the voltage drop V_{drop} over a known resistor R_{known} in series with a system with known system resistance R_{system} and known supply voltage V_{supply} [Michael Tse, 1998]. That is:

$$V_{drop} = \frac{V_{supply} \cdot R_{known}}{R_{system}} \Rightarrow u_K = \frac{u_{AC} \cdot R_K}{R_a + R_K + jX_a} \quad (B.4)$$

Where jX_a is the imaginary part of the armature resistance, the reactance. From equation B.4 a voltage ratio can be formed, denoted u_{rat} . Notice that an the absolute value of the denominator is found in order to loose the imaginary operator j .

$$u_{rat} = \frac{u_K}{u_{ac}} = \frac{R_K}{\sqrt{(R_a + R_K)^2 + (X_a)^2}} \quad (B.5)$$

When the supply voltage is set to 1.00[V] the voltage ratio is equal in magnitude to the measured voltage across R_K . Utilizing this and solving for the reactance yields:

$$X_a^2 = \left(\frac{R_K}{u_{rat}} \right)^2 - (R_a + R_K)^2 \quad (B.6)$$

Utilizing that the relationship between reactance and inductance is given as $X_L = \omega_{ac} \cdot L$, the inductance can be found from equation B.6.

$$L_a = \frac{1}{\omega_{AC}} \cdot \sqrt{\left(\frac{R_K}{u_{rat}} \right)^2 - (R_a + R_K)^2} \quad (B.7)$$

B.3 Motor Model Step Inputs

In the following the step inputs with 12[V] and 18[V] are shown.

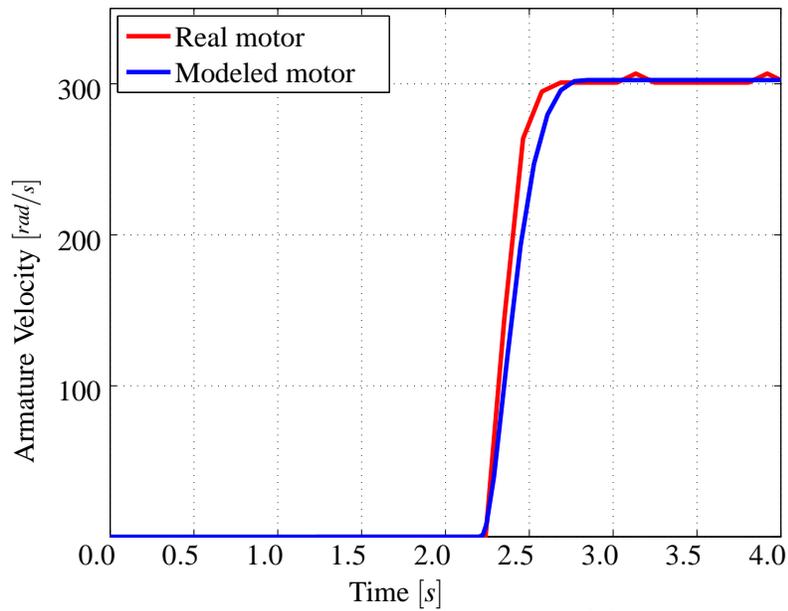


Figure B.2: Plot of step responses for the motor model a with 12[V] step input and the real system.

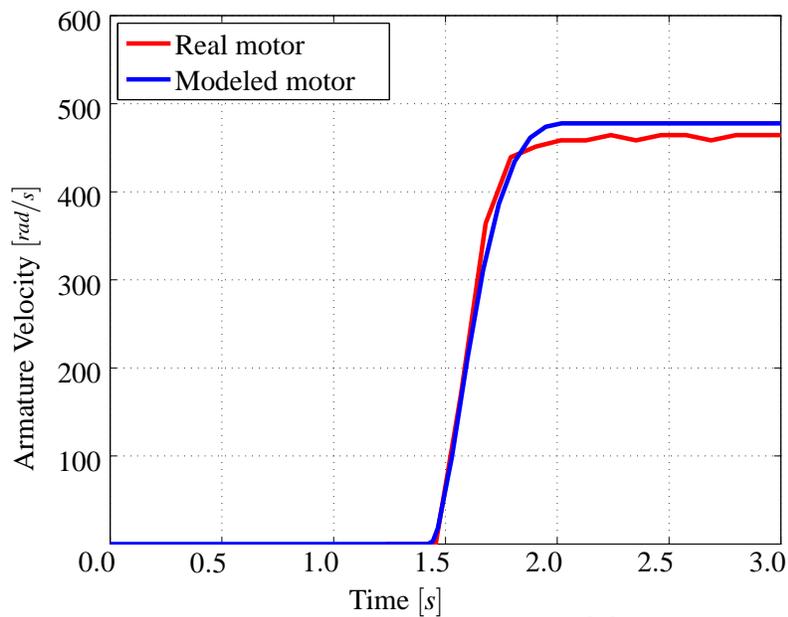


Figure B.3: Plot of step responses for the motor model a with 18[V] step input and the real system.

Experimental Verification

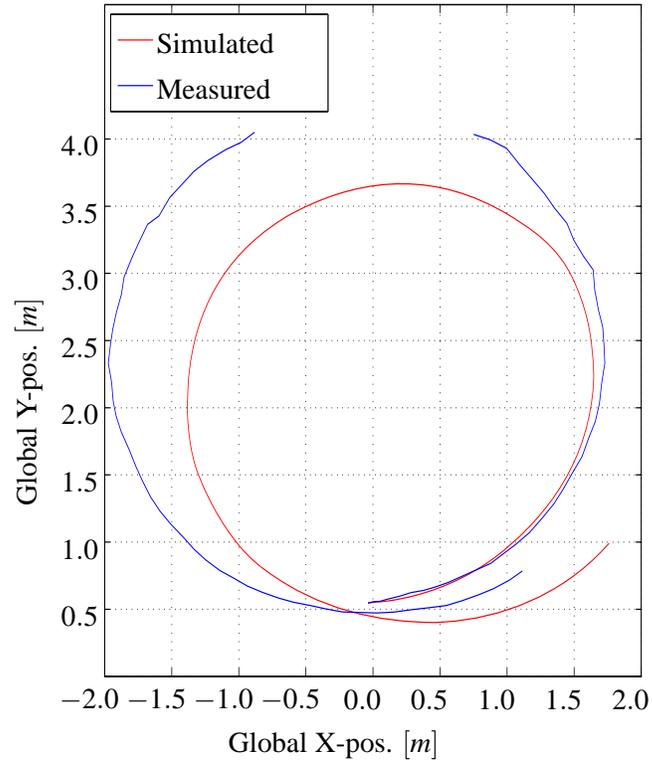


Figure C.1: Position comparison for data set 1 when the robot is running with 18[V] on right and 2[V] on left.

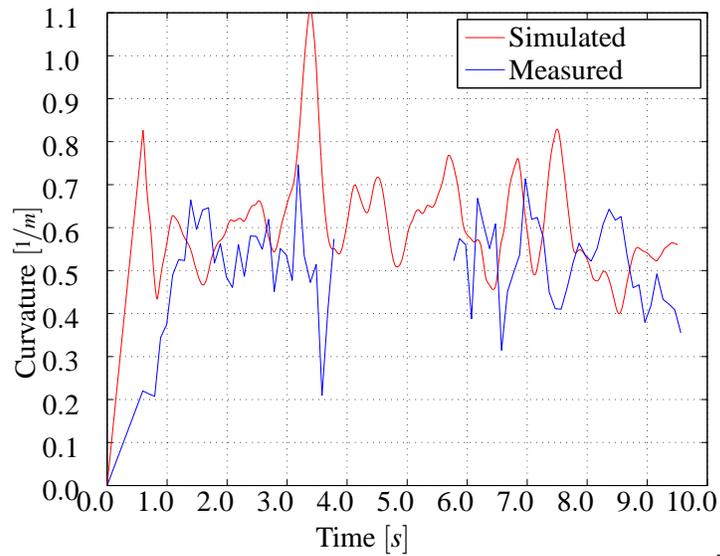


Figure C.2: Curvature comparison for data set 1 when the robot is running with 18[V] on right and 2[V] on left.

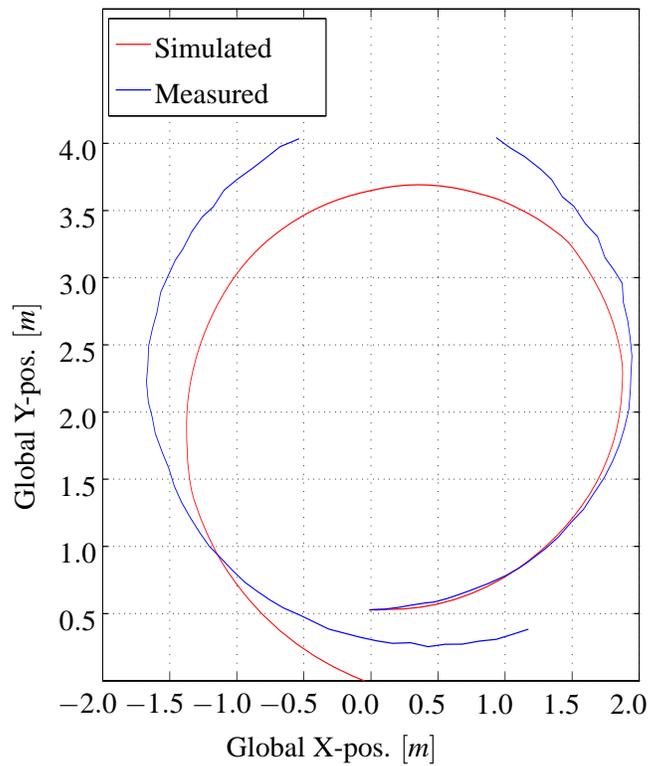


Figure C.3: Position comparison for data set 3 when the robot is running with 18[V] on right and 2[V] on left.

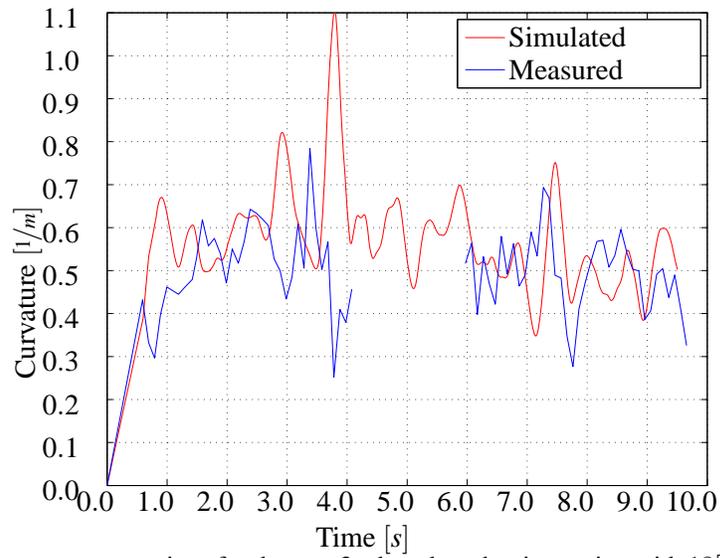


Figure C.4: Curvature comparison for data set 3 when the robot is running with 18[V] on right and 2[V] on left.

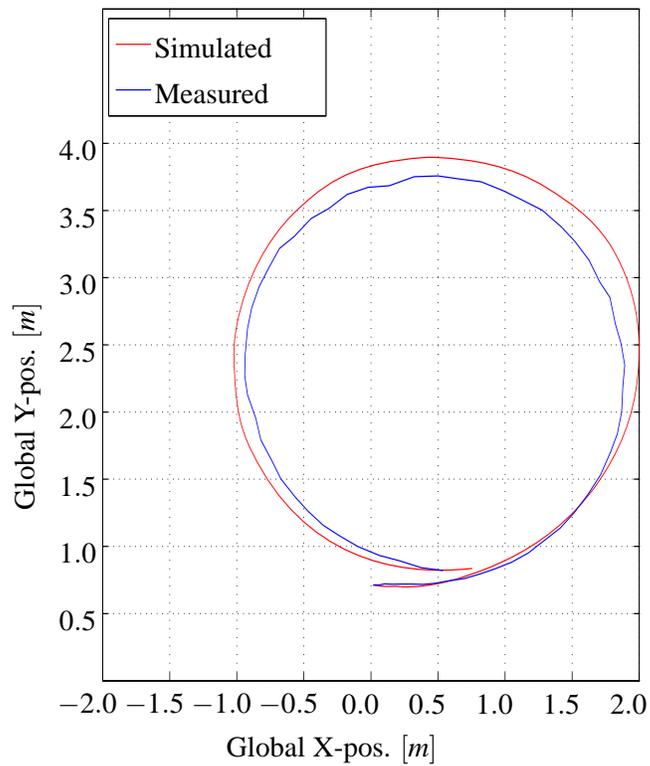


Figure C.5: Position comparison for data set 4 when the robot is running with 22[V] on right and 3[V] on left.

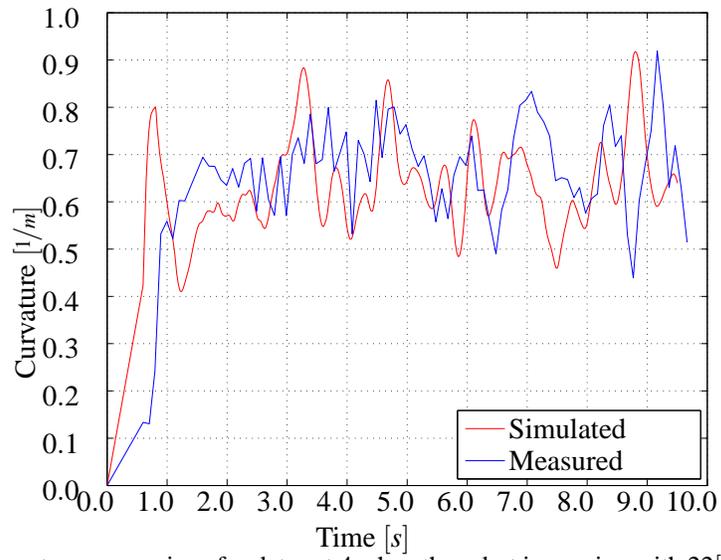


Figure C.6: Curvature comparison for data set 4 when the robot is running with 22[V] on right and 3[V] on left.

mark I navigation principle **D**

The method used by the previous group to work on the project, was to do a linear regression of the points on either the left or right side of the robot. This was used to identify the distance to, and angle of the row of corn or wall. The problem with this strategy is that it uses a predefined row width and keeps the distance to one row and disregards the other. This strategy could cause problems like undetected obstacles on the side where the robot is not looking or a bad trajectory caused by detecting an obstacle that is not part of the corn row. Figure D.1 shows this strategy.

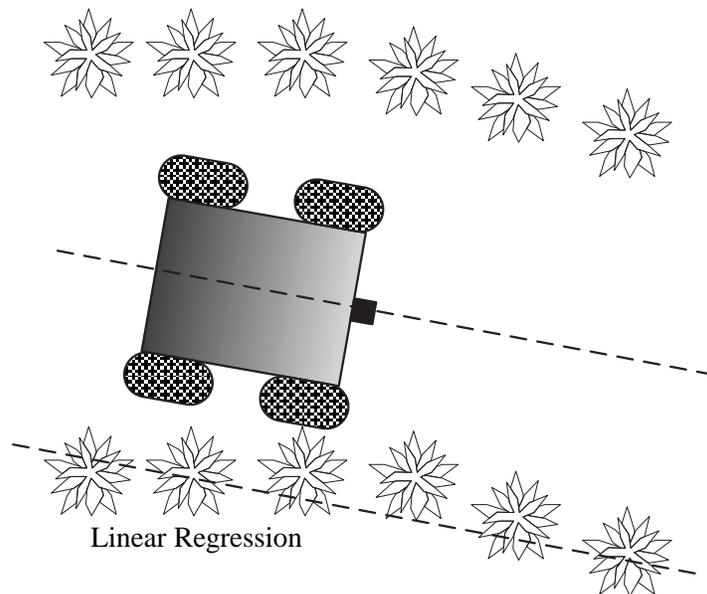


Figure D.1: Trajectory planned by identifying angle and distance to wall.

Hybrid Controller E

E.1 Step Responses using PID

In this appendix are the step responses for the implemented PID controller given. The PID gain values are given in table 8.1 on page 77. Firstly is the $27[\text{rad}/\text{s}]$ reference step response shown in figure E.1.

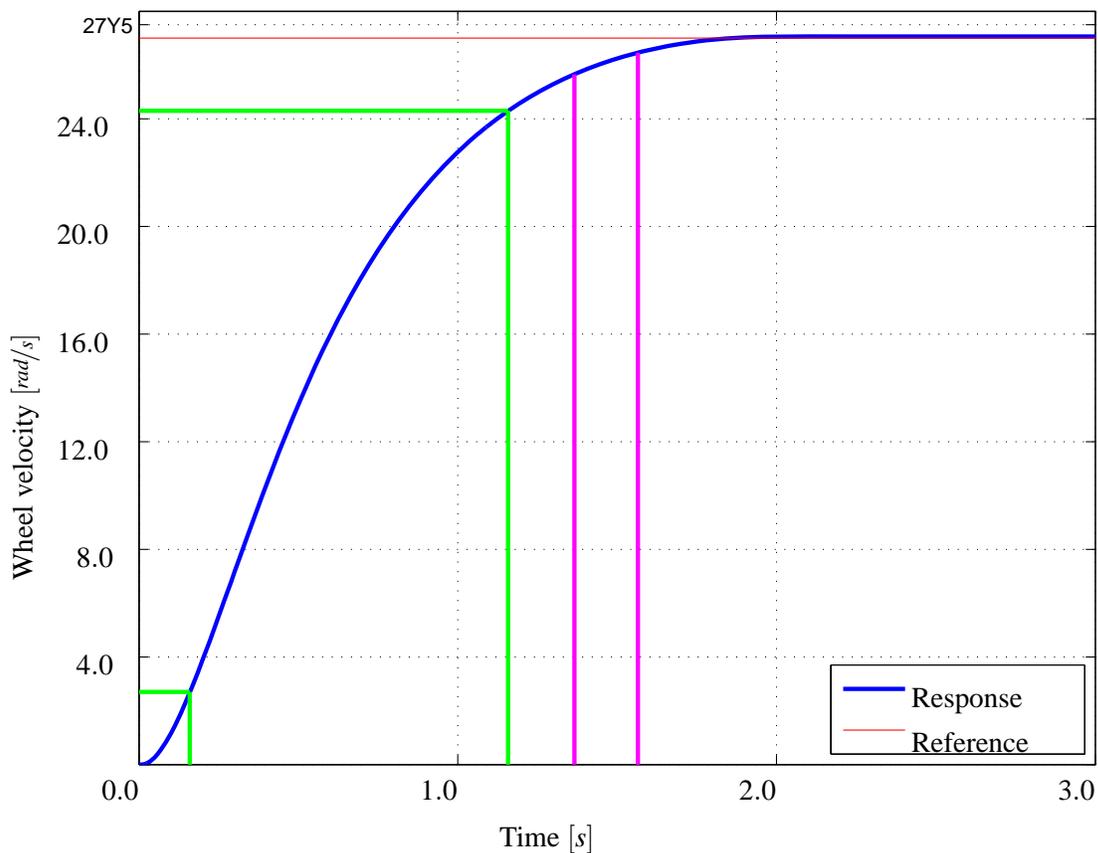


Figure E.1: Step response of PID controlled motor with a 27 reference (max).

For the step response in figure E.1 is the rise time found to be $0.99[\text{s}]$, which is within the boundaries. The settling time for 2% and 5% are found to be $1.56[\text{s}]$ and $1.36[\text{s}]$ respectively. These are both within the set boundaries. There is no overshoot when stepping to the maximum velocity, which indicates that the sampling frequency is set sufficiently high.

Secondly is the step response for a $10\text{rad}/\text{s}$ reference shown. This reference step response is shown in order to see what happens at lower velocity step responses. The step response is shown in figure E.2.

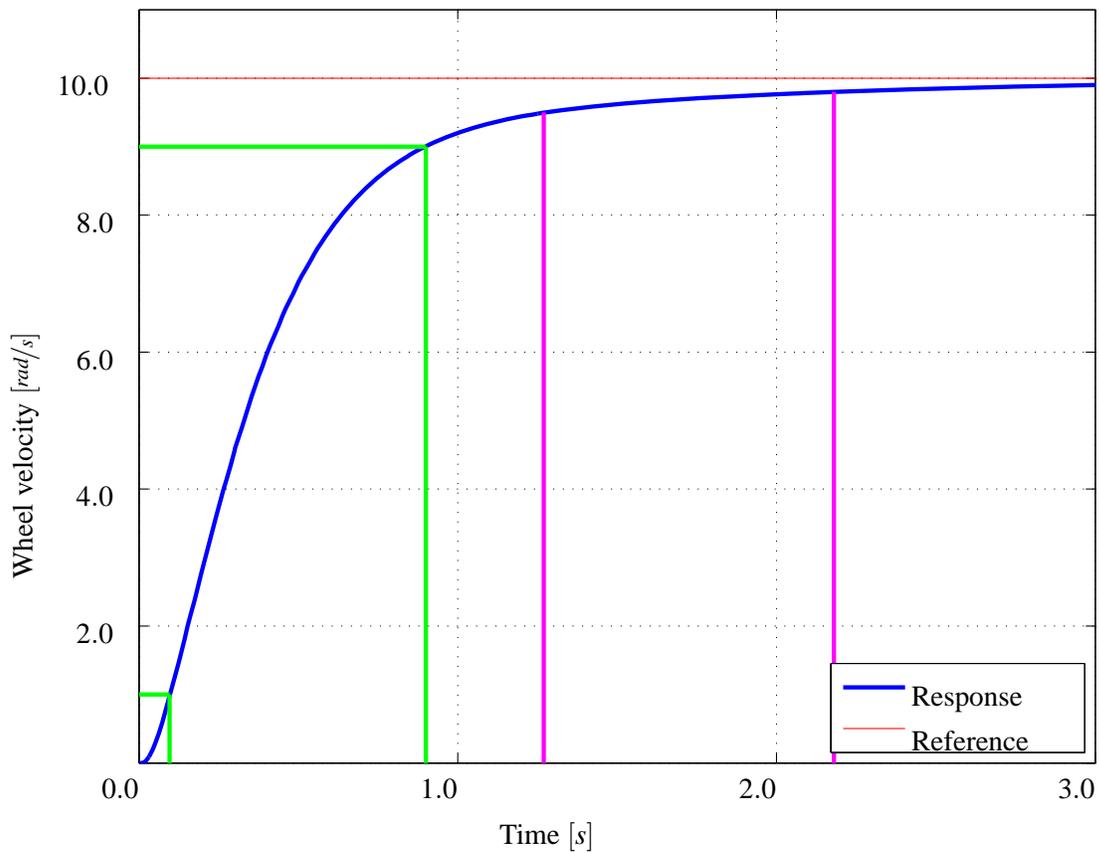


Figure E.2: Step response of PID controlled motor with a 10.0 reference.

For the step response in figure E.2 is the rise time found to be 0.8[s], which is within the boundaries. The settling time for 2% and 5% are found be 2.18[s] and 1.27[s] respectively. These are both within the set boundaries. There is no overshoot when stepping to the maximum velocity, which indicates that the sampling frequency is set sufficiently high.

Wiring Schematic

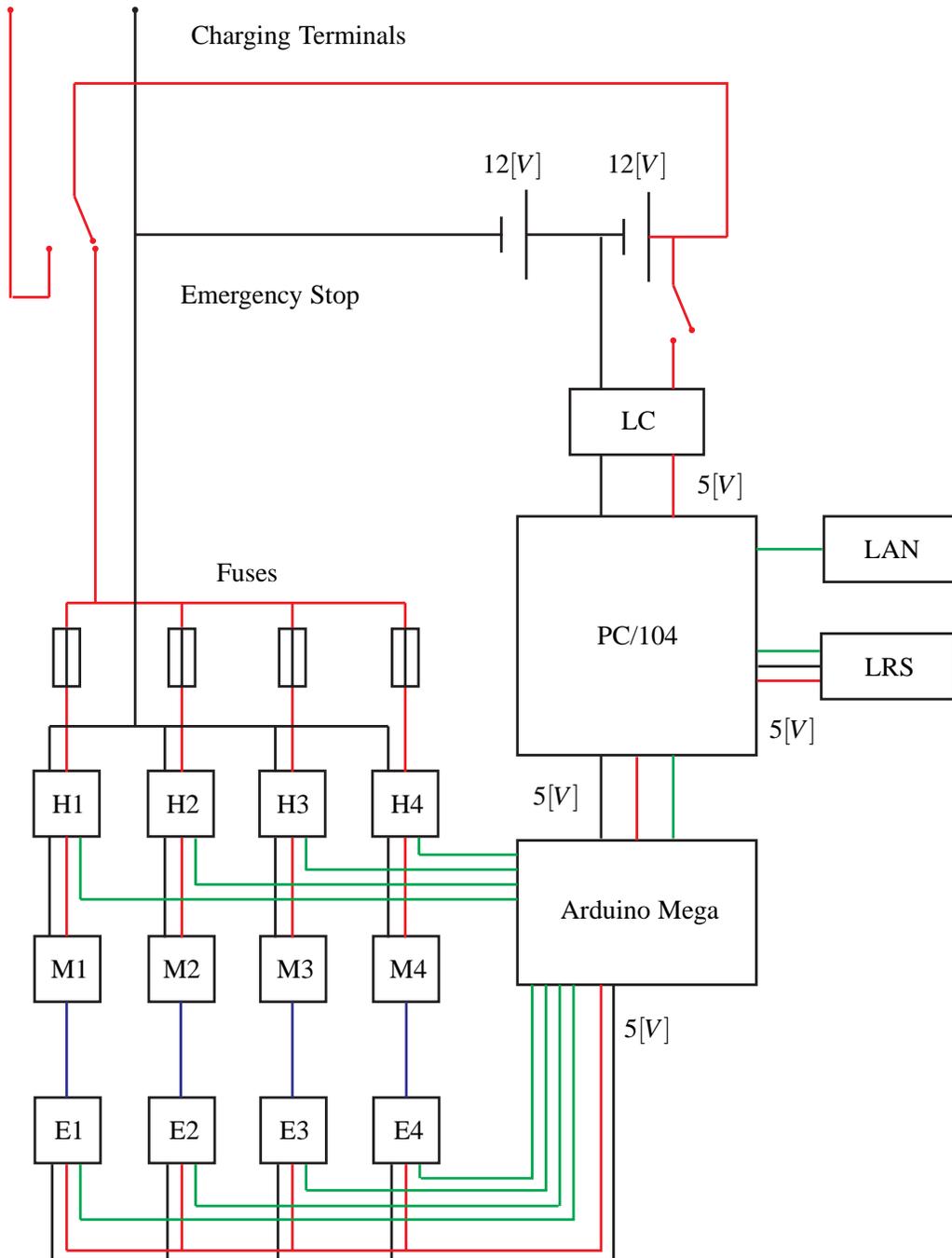


Figure F.1: Wiring schematic. Green lines are signals, Red is positive terminals, Black is Ground, and blue is mechanical connections.

LC The Linear Controller creates a 5[V] stable supply for the PC/104 board.

Batteries Batteries, denoted with 12[V], are placed in series to yield 24[V]. 12[V] goes to the LC.

PC/104 The PC/104 runs the controller and sends signals to the Arduino Mega board. It also handles LAN and LRS communications.

LAN The LAN port is used to communicate with the host computer.

LRS The Laser Range Scanner sends signals used in the controller.

Arduino Mega The Arduino Mega handles PWM and enable signals to the H-bridges and power to and signals from the encoders. The signals from the encodes are used in a PID controller that controls the motor PWM signals.

H-Bridges The H-bridges, denoted $H\#$, utilizes a PWM signal from the Arduino Mega to emulate a DC voltage.

Motors The motors, denoted $M\#$, uses the DC voltage from the H-bridges to create motion. The motor armature is connected to the encoders.

Encoders The encoders, denoted $E\#$, are connected to the armature of the motors sends signals to the Arduino Mega.

Fuses 10[A] fuses safeguard the H-bridges and motors.

Emergency Stop The emergency stop cuts power to the H-bridges and activates the charging terminals.

Charging Terminals The charging terminals are used to charge the batteries and are activated by the emergency stop.

List of Figures

1.1 Schematic of the track used for the advanced category [FRE2011, 2011].	3
3.1 Picture of the FORBOT platform [Roboterwerk, 2011].	9
3.2 Picture of the <i>Dr. Robot Jaguar</i> platform [Dr. Robot, 2011].	10
3.3 Picture of the <i>Gears SMP</i> platform [Gears Educational Systems, 2011 <i>b</i>].	10
3.4 Picture of the <i>H-MAC</i> platform [Gears Educational Systems, 2011 <i>a</i>].	11
3.5 Illustration of the two square waves.	15
3.6 Picture of the Hokuyo Laser Range Scanner [Hokuyo Automatic, 2010].	15
4.1 Open lid image of the <i>mark II</i>	19
4.2 Block diagram depicting the <i>mark II</i> system.	20
4.3 Illustration of the angle θ_e	21
5.1 Final model with sub models.	23
5.2 Local wheel velocities.	24
5.3 Block diagram of dynamic vehicle model [Solc and Sembrera, 2008].	26
5.4 Sketch of vehicle with index numbering of the wheels.	28
5.5 Plot of original and recreated circles of 1 st simulation run.	30
5.6 Plot of original, recreated and measured circles of 2 nd simulation run.	32
5.7 XY-Plot of equal but opposite wheel velocities.	36
5.8 Plot of the torque generated by a left motor during the XY-plot.	37
5.9 Schematic of the internal measurements.	38
5.10 Illustration of the two square waves.	39
5.11 Example of counter clockwise rotation.	40
5.12 Example of clockwise rotation.	40

5.13	The scanning area and the measurement area.	42
5.14	Principle behind the tracking of the cylinder.	42
5.15	Plot of LRS view of the tube.	43
5.16	Experimental setup.	44
5.17	Position comparison for data set 2 when the robot is running with 18[V] on right and 2[V] on left. 46	
5.18	Curvature comparison for data set 2 when the robot is running with 18[V] on right and 2[V] on left. 46	
5.19	Position comparison for data set 6 when the robot is running with 22[V] on right and 3[V] on left. 47	
5.20	Curvature comparison for data set 6 when the robot is running with 22[V] on right and 3[V] on left. 47	
6.1	Final model with sub models.	49
6.2	Equivalent diagram of the PMDC motor circuit.	52
6.3	Equivalent diagram of armature inductance experiment with a resistor in series.	52
6.4	Plot of load characteristic.	54
6.5	Plot of deceleration test velocity over time.	55
6.6	Block diagram of equation 6.10.	57
6.7	Block diagram of equation 6.11 with substituted <i>back-emf</i>	57
6.8	Block diagram motor model.	57
6.9	Block diagram motor model.	58
6.10	Plot of step responses for model with original settings and the real system.	59
6.11	Plot of step responses for model with altered settings and the real system.	60
7.1	Configuration space of three rows in a corn field.	62
7.2	The wagon tongue principle.	63
7.3	The two stage data reduction process.	64
7.4	Example of a polar configuration space with the given DT. Gray is obstacle space and white is free space. 65	
7.5	Histogram for the polar configuration space shown in figure 7.4.	66
7.6	Vector field histogram.	67
8.1	Block diagram of the controllers.	69
8.2	Block diagram of the motor system with controller.	70
8.3	Block diagram of PID controller.	71
8.4	Step response to 18[rad/s] with proportional gain 10 and sampling frequency 4[Hz].	73

8.5	Step response to $18[\text{rad/s}]$ with proportional gain 1, integral gain 3 and sampling frequency $4[\text{Hz}]$.	74
8.6	Step response to $18[\text{rad/s}]$ with proportional gain 1, derivative gain 3 and sampling frequency $4[\text{Hz}]$.	75
8.7	Simulated step inputs with and without zero order hold feedback.	76
8.8	Step response of PID controlled motor.	77
8.9	Step response of PID controlled motor with shorter time axis.	78
8.10	Diagram of a hybrid system controller algorithm.	80
8.11	Hybrid controller.	81
8.12	Fuzzy logic controller.	83
8.13	Fuzzy set used for fuzzyfication of the angle error θ_e .	84
8.14	Fuzzy set used for defuzzyfication to get motor references.	84
8.15	The input and the two outputs of the fuzzy logic controller.	85
8.16	Open loop controller <i>SC2</i> strategy 1 sketch.	86
B.1	Circuit diagram of armature inductance experiment.	102
B.2	Plot of step responses for the motor model a with $12[\text{V}]$ step input and the real system.	103
B.3	Plot of step responses for the motor model a with $18[\text{V}]$ step input and the real system.	103
C.1	Position comparison for data set 1 when the robot is running with $18[\text{V}]$ on right and $2[\text{V}]$ on left.	105
C.2	Curvature comparison for data set 1 when the robot is running with $18[\text{V}]$ on right and $2[\text{V}]$ on left.	106
C.3	Position comparison for data set 3 when the robot is running with $18[\text{V}]$ on right and $2[\text{V}]$ on left.	106
C.4	Curvature comparison for data set 3 when the robot is running with $18[\text{V}]$ on right and $2[\text{V}]$ on left.	107
C.5	Position comparison for data set 4 when the robot is running with $22[\text{V}]$ on right and $3[\text{V}]$ on left.	107
C.6	Curvature comparison for data set 4 when the robot is running with $22[\text{V}]$ on right and $3[\text{V}]$ on left.	108
D.1	Trajectory planned by identifying angle and distance to wall.	109
E.1	Step response of PID controlled motor with a 27 reference (max).	111
E.2	Step response of PID controlled motor with a 10.0 reference.	112
F.1	Wiring schematic. Green lines are signals, Red is positive terminals, Black is Ground, and blue is mechanical co	

Abstract

The project *mark II* with the theme *Autonomous robot navigation through a corn field* is concerned with an autonomous skid-steered robot. The background for the project is The Field Robot Event 2011, where robots compete in driving through a corn field with obstacles. There are clear rules regarding the competition, where only autonomous robots can compete. The goal of the competition is to drive through a corn field without knocking down any plants, being able to turn the robot at the end of a passage and detect and mark weeds scattered across the field. Detecting and marking weeds has not been part of this project.

Through a selection process of available platforms, the platform, that fulfilled set up criteria the best, was chosen. The chosen platform comprises a chassis, four wheels, four *PMDC* motors and four encoders. The components that are needed to complete the robot system have been chosen. Some components through a selection process and some because they have been available from start of the project. The components chosen are a PC/104 board, Arduino board, 4 H-bridges, a laser range scanner and a linear voltage controller.

A dynamic vehicle model has been made from reviewing a range of papers concerning skid-steering vehicles. Two papers were chosen and the dynamic models described in the papers were built. The first dynamic vehicle model showed similar outputs to the outputs given in the paper. The second dynamic vehicle model yielded many problems and the final version still had a problem. The first dynamic vehicle model, that showed the best results, was chosen as the simulation for further use in the project. The laser range scanner was used in an experiment to record the path traveled by the assembled robot system. The recorded data was compared against the chosen dynamic vehicle model. This was done to verify that the model could simulate the behavior of the robot system.

A motor model describing the dynamics of the motors was built on background of a series of experiments. The transient behavior of the motor model was verified against recorded data for the motors. Step inputs were used for both model and motors.

As the robot was to be an autonomous robot, it was necessary to make a navigation algorithm using the laser range scanner. A selection process yielded the Vector Field Histogram method, that could find the best route for robot by using the laser range scanner data.

Controllers were developed for the robot system on background of the information obtained through working with the robot system. The controller was made as a hybrid system controller that allows the systems to change between controllers. A fuzzy logic controller was made to determine motor velocity references when it was driving down a corn field passage. The references differed in magnitude, as the difference determined the rate of angular change for the robot system. If the robot was oriented correctly on the path the motor velocity references were set to be equal, which made the robot drive in a straight line. The fuzzy logic controller generated references by evaluating feedback data from the laser range

scanner. Other controllers were made for the hybrid system controller, though these controllers were open loop and based on simple logic. The simple logic controllers were made to handle situations like obstacles and steering the robot into the next corn field passage.

Cascaded inside the hybrid system controller are four PID controllers, one for each motor. The PID controllers were given references from the hybrid system controller and utilized encoder feedbacks to control the motors. The PID controllers were designed in the final model, that comprised both the dynamic vehicle model and the motor model.

The hybrid system controller is still to be installed onto the PC/104 board and the PID controllers are still to be installed onto the Arduino board. When the controllers are installed the next step is to adjust the controllers when driving through a corn field passage. When the controllers are implemented and adjusted the robot will be ready for The Field Robot Event 2011.