Autonomous external pipe crawling robot for offshore inspection

Master thesis

Esben Uth, Mikkel Edling, Jannic S. Larsen

STUDENT REPORT

AALBORG UNIVERSITY

Department of Energy Technology
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Autonomous external pipe crawling robot for offshore inspection

Theme:
Advanced Control of Offshore Energy Systems

Project Period:
Fall and spring semester 2023-2024

Project Group:
OES4-1

Participant(s):
Esben Uth
Mikkel Edling
Jannic S. Larsen

Supervisor(s):
Jesper Liniger (AAU)
Simon Pedersen (AAU)
Sigurd Klemmensen (SubC Partners)

Copies: 1

Page Numbers: 189

Date of Completion:
May 31, 2024

Preface

This thesis is written by robot, studying at Aalborg University Esbjerg. The project was prepared from September 2023 to June 2024 and processes the overall subject: Control of offshore dynamic systems.

The objective of this report is to gain insight in how to control, develop, and improve an existing robot to increase its functionality. Applying skills developed in previous reports and courses. Increase practical skills as well as theoretical ones.
This report is aimed towards people who want to design an external pipe crawling robot and how to make it robust to the harsh and unpredictable conditions of the ocean. In addition, there is interest for SubC Partners for improvement on their existing technology.

Aalborg University, May 31, 2024

Instructions for reading

The report is written in LaTeX, and each chapter is marked with a certain number, and is divided into sections. All references used throughout the report are indicated by the method referred to as the Institute of Electrical and Electronics Engineers (IEEE). The bibliography is made in Mendeley and BibTeX, and the citations used throughout the sections are noted in the text either at the beginning of a section or as each individual statement is made. Figures and table references are mentioned in the caption.

Abstract

The goal of this report is to reduce the effect of wheelspin in offshore operations using the SubC Partner crawler. Further goals were set to reduce the complexity of these operations as well as increase the autonomy of the robot. The crawler was partly upgraded with a drivetrain consisting of three electric motors, for which two LQR tuned controllers were designed to track position or velocity based on gain scheduling. To reduce wheelspin, a supervisory slip controller was developed based on a sliding mode disturbance observer. To overcome obstacles, a logic controller was developed to reduce the clamp pressure of the crawler. The controllers were tested in various conditions on a developed non-linear model. Here, it was found that the electric drive is capable of obtaining one full rotation of the offshore member of 0.54 m radius even with one wheel slipping. In physical tests, it was found that the implemented controllers had difficulty handling the low reference velocities of 0.016 rad/s. It is concluded that with a further gear reduction of 3, a lower speed range can be obtained. This together with a control scheme using gain scheduling can meet the velocity and position requirements posed by SubC Partner. The increase in available torque would also reduce the number of motors needed for the electric drive to two.

Resumé

Målet med denne rapport er at reducere mængden af hjulspin under offshore missioner med SubC Partner's Crawler. Andre mål for rapporten er at øge mængden af autonomitet af robotten samt at sænke mængden af kompleksitet omkring missionerne. Crawleren blev delvist opgraderet med tre elektriske motorer til at drive den. To LQR-tunede kontrolsystemer blev designet til at følge en referenceposition og hastighed, baseret på gain scheduling. For at reducere mængden af hjulspin blev et hjælpeslipkontrolsystem udviklet, baseret på en sliding mode observer. For at overkomme forhindringer blev et logikkontrolsystem udviklet til at reducere mængden af klemmetryk i Crawleren. Kontrolsystemerne blev afprøvet i forskellige simulationsscenarier på en udviklet ikke-lineær model. Her blev det vist, at de elektriske drev kan levere en fuld rotation på et testrør med en radius på 0,54 m, selv med et hjul, der laver hjulspin. I fysiske tests blev det vist, at de implementerede kontrolsystemer havde svært ved at håndtere den lave hastighedsreference på 0,016 rad/s. Det blev konkluderet, at med en yderligere gearreduktion på 3 kan opnås et lavere hastighedsområde. Dette, sammen med et kontrolscenarie, der gør brug af gain scheduling, kan opnå de kriterier, der er sat af SubC Partner. Samtidig vil den øgede mængde af tilgængelig torque muliggøre reduktionen af antallet af elektriske motorer til to.

Contents

1 Introduction

In 2016 it was decided by the Danish Underground Consortium to cease production from the largest Danish gas field, Tyra. The gas field is located 225 km off the West Coast of Denmark and is responsible for 90% of the Danish gas production [1], [2]. The Tyra field consists of a total of 11 platforms, of which 5 are unmanned. The field began operation in 1984 and was shut down in 2019 for redevelopment. Since the field was established, the sediment has receded 5 m. The cause is due to the extraction of hydrocarbon pockets from the gas field, causing them to collapse and the seabed to sink with it [2].

An example of one of the production platforms can be seen in figure 1.1.

Figure 1.1: Platform at Tyra East. *Credit: Tom Jervis, Flickr*

As seen in this figure the spider-deck of the platform is marked by a red square. This deck was originally located 6 m above sea level, but was in 2016 located directly in the splash zone [3][4]. This causes biofouling to occur on the members, including those on

the spider-deck. This increases the forces acting onto the members. The bio-fouling can weigh more than 33kg per square meter increasing the structural loading and compromising the integrity of the entire structure [5]. The splash zone is also the region mostly affected by corrosion affecting the lifetime of the spider deck [6]. Before redeveloping the field could begin, the old spider decks had to be removed. Previous work done for maintenance or inspection in the splash zone was usually done by divers or climbers descending from the top of the structure [3].

Figure 1.2: Spider deck located in the splash zone [3]

However, as seen in figure 1.2, working in the splash zone can be very unsafe. The divers are only able to operate in a maximum of 0.5 m waves [7], and whether it is possible to operate under these conditions relies on the weather. This would make decommissioning the spider deck a challenge. To increase personnel safety and expand the window of operation, Total Energies contracted with the Danish company SubC Partner to develop a robot to accompany this task [2]. After 4 months, the SubC Crawler was developed and put into operation, as seen in figure 1.3.

Figure 1.3: Crawler operating on the spider deck in the splash zone

The Crawler uses hydraulic arms to attach to a member and it is equipped with wheels in order to rotate around the member. When attached, it can crawl lateral along the member [7]. The crawler was developed to be able to operate in wave heights reaching 2.5 m, greatly increasing the window for operations compared with previous ways. The Crawler was also produced as a multi-tool platform giving it possibilities to inspect, clean, cut, and dispose of the members. It was outfitted with a diamond cutting drill to cut the members and a high pressure cleaner to remove the biofouling, and it also came with an ultrasonic sensor, which could measure the thickness of the members. The hy-

draulic arms were designed to withstand the weight of a severed member, thus enabling the Crawler to also operate as a crane. The pressure washer would clean in front of the wheels to create a more uniform surface on which to rotate [3]. The Crawler operates at speeds between 0.5 m/min to 1.5 m/min angular and 0.33 m/min lateral.

Figure 1.4: Operator performing manual control on the Crawler using the hydraulic control panel

As seen in figure 1.4, the control of the Crawler is done manually by an operator located top site of the offshore structure. The control station together with hydraulic pumps and electric generators is situated inside a container [3]. The redevelopment of the field is planned to be completed in late 2023 or early 2024 [1]. When completed, it allows for an additional 25 years of production and Danish self-sufficiency in gas [3].

1.1 Inspect and Cut mission example

By interviewing the team leader in the Robotics Department of the SubC Partner, Sigurd Klemmesen (MSc. Offshore Energy Systems), a basis for understanding the use case of the Crawler was established, also elaborating on issues found during previous operations.

During a period of 10 months, a total of eight operations were performed on a platform in the North sea. An initial mission was performed in order to measure and report the thickness of the members located on the spider deck. With this data, the missions continued cutting down the spider decks. In total, the Crawler made 82 perpendicular and 29 circumferential cuts. The cut operations were able to perform 125 cuts in 97 days, leaving out days not working due to maintenance or poor weather conditions. From the inspection mission, it was found that out of 26 days, only 16 days were used for performing inspection work using the Crawler. The operation was carried out by a group of trained SubC personnel. Counting a total of four people. This includes one operator and three assistants. According to SubC Partner, the manpower costs of the operation are estimated to be 14 300 DKK per day. No total price for the operation, including machinery rental, is available.

From the inspection mission the mission efficiency can be set to 61.5%. In manpower alone the cut missions had an expense of 1 387 100 DKK, thus the cost of the cutting missions could have been reduced by 534 033 DKK if the mission efficiency where increased

to 100%, being able to operate without any interference.

During the various missions, the following issues with the Crawler were identified:

- **Wheel spin**
  The Crawler was seen to generate wheel spin in some conditions, either due to the member not being completely round, or a high density of biofouling on the member
- **Scaffolding**
  It was found that the need of erecting extra scaffolding for operations occurring different places at the platform could be a tedious task due to it being dependent on calm weather
- **Hydraulic pump and external components**
  The Crawler is dependent on a large hydraulic pump and tank used for operating the components, thus making it difficult to move around the platform
- **Umbilical**
  Due to the many hydraulic connections, the umbilical of the Crawler has a large diameter. This was found to make movement of the Crawler between members difficult.

The conditions of the members of the offshore structures can vary depending on the conditions in which they are located and the age of the structure. As seen in figure 1.5, mussels and marine growth can attach to members of the structure.

Figure 1.5: Crawler operating on offshore member with marine growth

Although the Tyra field redevelopment is set to finish early 2024 and the world is transitioning to a more green future due to a reduction in the use of oil and gas. A market for the Crawler will still exist. Future oil and gas platforms will also need to be torn down or replaced. The same can be said for substations used for offshore wind. Other companies might also invest in this future. Thus, for SubC, it is important to stay at the forefront of robotic development and continue pushing the limits of the Crawler capabilities. Constantly reducing the overall costs of operations.

To identify areas where the operational efficiency of the Crawler can be increased the following analysis is conducted with regard to the initiating problem of this report which is to investigate:

How can the complexity of the crawler operation be reduced while increasing operation efficiency at a lower operational cost?

1.2 Problem Analysis

In order to answer the initiating problem, certain areas within Crawler operations have been identified as the main areas which affect complexity and operational efficiency. These are listed below:

- **Experience of the operator**
  Since the crawler is manually operated, it requires an operator who has a lot of experience in operating the robot. This can affect operations negatively due to skill dependency.

- **Complexity of setup around Crawler operations**
  Due to the safety regulations and equipment used to operate the Crawler, the setup needed for operating the Crawler can involve scaffolding, cables, and hoses, control container, high-pressure waterjet, thus relying on external manpower. This could be a cause of prolonged missions.
  Furthermore, all the external equipment used for the crawler also needs to be transported offshore together with the crawler itself.

- **Decreased operation capabilities due to harsh weather**
  Due to the complexity of setup, as well as the manually operation of the Crawler. The window of operations is highly dependent on the weather conditions. Thus, making it difficult to plan ahead the length and cost of operations.

- **Relocation of Crawler**
  The Crawler in its current state is not able to operate between different members, thus it needs to be detached and moved manually by crane between the members. This significantly affects operational efficiency.

To decrease the reliance on highly skilled operators, the Crawler could be upgraded to become a more autonomous robot, thus only relying on manual input from the operator to set the angular reference for member rotation. At its current state, the Crawler is not equipped with any kind of location sensors or any onboard computer for data processing. Thus, to achieve this, these must be investigated and implemented.
Based on [8], it is believed that by replacing one hydraulic wheel drive with one electric on each wheel, the problem of wheel spin can be removed. Furthermore, it is proposed to use an adaptive multivariate control approach, making it possible to control each wheel

separately, as well as the actuator arm, with a single reference input, acting against unknown obstacles [9].

Since most of the complexity around the setup of the Crawler relates to feed water and hydraulic pipes, it is proposed to reduce the amount of hydraulic components on the Crawler. In this project, studies will be performed to replace the components with electric alternatives.

It is believed that this is achieved by increasing the autonomy of the Crawler as well as reducing the complexity around its setup. The Crawler will be able to operate more efficiently in various weather conditions. In addition, it reduces the complexity of moving the crawler between different members.

2 Problem Formulation

The previously stated issues as well as proposed solutions are derived from the following problem statement for this report:

"Develop a semi autonomous system by the uses of electrification reducing the complexity in order to increase mission reliance and operation efficiency, identifying electric actuators and sensors used for operating the Crawler to a reference position within a SubC Partner proposed time frame"

2.1 Project Purpose and Goals

To solve the above problem statement, several goals are set to be achieved within this project. Since some of these goals rely on each other to be full-filed, they are listed accordingly to their progression throughout the report.

Goals

- Investigate the possibility of replacing hydraulic components with electric
- Implementation of mechanical and electric changes onto the physical Crawler
- Identification of reliable sensors for angular position and velocity measurements, validated by experimental results
- Design of a mathematical model describing the kinematics and dynamics of the Crawler, and identifying its respective parameters.
- Development of a multi-variable controller, able to identify and correct wheel slip of the Crawler, making the Crawler reach a set angular position reference
- Implementation of the controller, onto the physical Crawler using ROS, validated using experimental results

2.2 Delimitation

To achieve the objectives of the project, some delimitations are specified to ensure that the project is kept within the limits of the study curriculum, as well as the time frame for this master thesis.

- The system will only be implemented onto one clamp out of the normal two attached to the original Crawler.
- Movement between two different members will not be examined

2.3 SubC Partner requirements

SubC Partner has certain requirements with respect to the operation of the Crawler.

- The crawler should be able to operate at vertical, horizontal, and tilted pipes at an arbitrary angle.
- The crawler needs to be able to operate at speeds of 0.016-0.048 rad/s when using tools, and 0.10 rad/s when under transport around the pipe.

3 System Description

In offshore operations, the Crawler consists of two clamp, with arms connected by hydraulic actuators. A side view of the Crawler can be seen in figure 3.1.

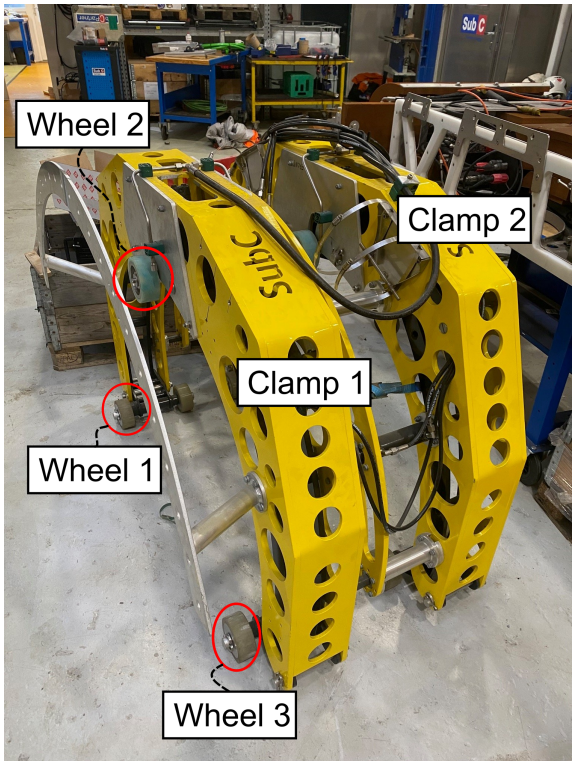


Figure 3.1: Operational offshore crawler seen from the side

The two clamping mechanisms, as well as the center actuators, can be operated individually. For rotation, the drive systems on each clamp are operated synchronously. All crawler actuation is performed by a single pilot who controls the hydraulic valves. The hydraulic control panel can be seen in figure 3.2.

Figure 3.2: Manual pilot control panel for the offshore Crawler

In offshore operations, the overall system boundaries would include the Crawler, the offshore structure, the control container, electric cables, hydraulic and water pipes, scaffolding, and lifting equipment used to align the Crawler with any member. In this system, the Crawler is able to operate on any member of the structure within the following categories.

- Vertical
- Horizontal
- Arbitrary angle between 0 and 90°

The direction of forces and friction, as well as their impact on Crawler operations, depends heavily on the type of scenario. An overview of the above three scenarios is depicted in figure 3.3. The angle between the horizontal plane and the tilted member is marked by α. In the figure it is seen that crawler is connected to a control container located on top of the offshore structure. In current operations, hydraulic and water pipes are connected between the control container and the crawler, as well as electric cables used for communication and power.

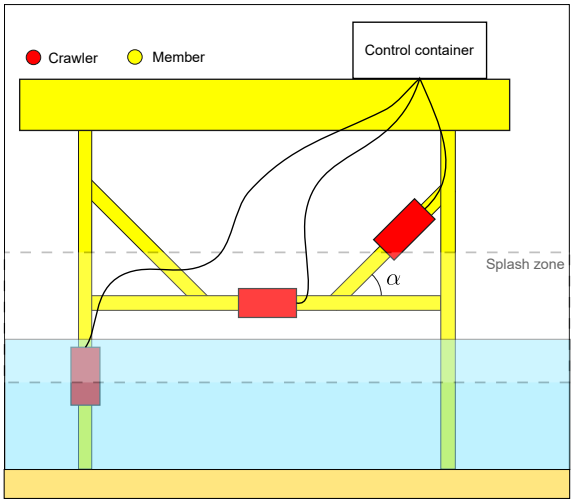


Figure 3.3: Visual depiction of offshore structure, red box notes the three operation positions of the Crawler, yellow parts are members of the offshore structure, greyed area depicts the splash zone

According to SubC Partner, the main cutting operations are performed on horizontal or tilted members, whereas vertical members can be cleaned and inspected. The splash zone is seen as the gray area, bordered by dashed lines. The height and location of this area depend on the average wave height in the area. For the North Sea, this is around 1.2 m [10].

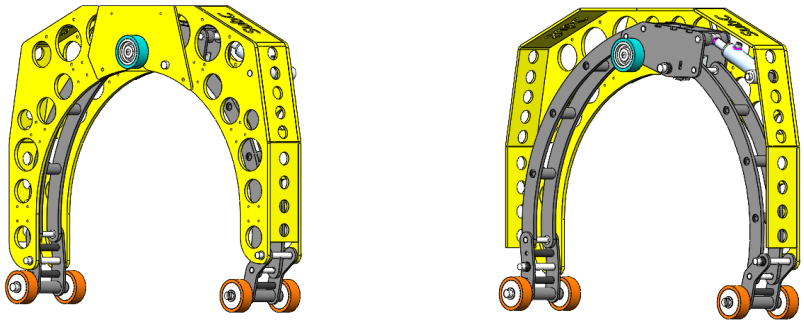
The Crawler is operating on various surfaces such as

- Clean paint
- Rough paint
- Rust
- Mussels
- Algae

3.1 System boundaries and areas of operation

Compared to the boundaries that form a typical Crawler mission, the boundaries of this project is limited to some extent. Due to the clamping arms being identical, this project will only focus on one of the arms of the Crawler as seen in figure 3.4a.

Going forward in this project, the single clamp arm will also be referred to as "crawler".



(a) Outside of the arm **(b)** Beneath the outer frame

Figure 3.4: Sideview of one Crawler arm

According to the 3D model, a single clamp weighs 92 kg. The laboratory at SubC Partner is able to provide a test member of 1.08 m in diameter. The member can be oriented to enable the Crawler arm to be tested in a vertical and horizontal position. Therefore, the system boundaries will be limited to these two positions. An image of the test member can be seen in figure 3.5.

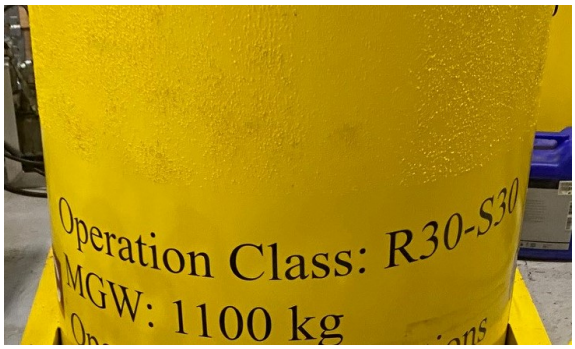


Figure 3.5: Vertical member used to conduct experiments on, with a smooth and rough surface

As the test member is welded together, a weld seam of approximately 10 millimeters of width and hight of 2.9 millimeter is running across the member from the top of the member to the bottom. This weld seam will most likely act as a challenge for the crawler to roll over and can be observed as a test for when the crawler will face more solid marine growth and the crawler is forced to reduce its clamping force. The welding seam is shown in figure 3.6.

Figure 3.6: Welding seam across the offshore test member at SubC Partners.

Furthermore, the design will not be tested in the sea. The project will mainly focus on establishing a basis for improving and upgrading the Crawler. Due to this, it is not important that all components are waterproof. However, components will be chosen with

future waterproofing in mind so that it is possible to further develop on this platform.

Since one goal of this project is to reduce the operational complexity involving the Crawler, control container, hydraulic pumps, and feed pipes are not included within the boundaries of this project.

In summary, this project will focus on the following:

- Single Crawler arm controlled by a local control unit
- Operating areas are set to be vertical and horizontal pipes of 1.08 m in diameter.
- The surfaces of the test member consist of a smooth and rough area as seen in figure 3.5.

This will also act as the boundary of the system.

3.2 Coordinate system and notations

Two reference frames are used in this project. The East-North-Up (ENU) system is used to define the fixed world frame and the location of the test member within it. This can be seen in figure 3.7. The angular position of the test member with respect to the D axis is described using α.

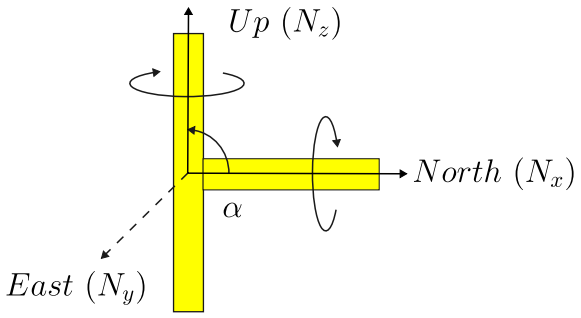


Figure 3.7: World reference frame

The rotation of the crawler around the member in the world frame is depicted as rotations around the N_z and N_x axes. A second reference frame is used to describe the angular position of the crawler body around the test member. As seen in figure 3.8, the body origin of the body of the crawler is located in the center of the member, since this is the pivot point around which the crawler is rotating. Since the Crawler only rotates, the coordinates are also set in polar form. The conversion between the two systems can be found by $\theta = \tan^{-1}\left(\frac{Y_{cra}}{X_{cra}}\right)$, and the distance from the origin to the Crawler can be found by $R = \sqrt{{X_{cra}}^2 + {Y_{cra}}^2}$.

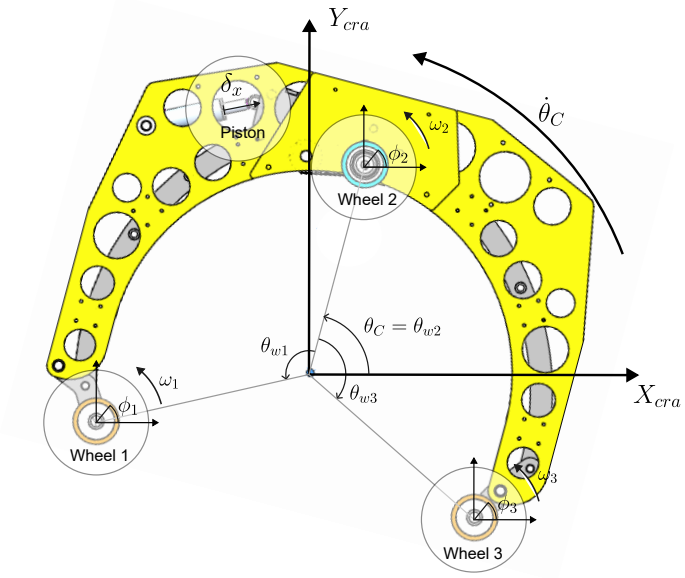


Figure 3.8: Global reference frame of the Crawler

The positions of the three wheels and the Crawler with respect to this origin are marked as $\theta_{w,1}$, $\theta_{w,2}$, and $\theta_{w,3}$. Each of the three wheels has a local coordinate system with their respective origins in the center of each wheel. From this, the angular position of the wheel is denoted as ϕ and the velocity as ω. The Crawler velocity is denoted as $\dot{\theta}_c$. The piston rotates at a fixed angle with respect to the origin of the body, however, the linear movement of the piston is noted as δ_x.
To describe the rotation of the crawler body frame around the world frame N_z and N_x axis, the following two rotation matrices can be used.

$$R_x(\theta_c) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_c) & -\sin(\theta_c) \\ 0 & \sin(\theta_c) & \cos(\theta_c) \end{bmatrix} \tag{3.1}$$

$$R_z(\theta_c) = \begin{bmatrix} \cos(\theta_c) & -\sin(\theta_c) & 0 \\ \sin(\theta_c) & \cos(\theta_c) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.2}$$

Throughout this report, forces and moments will be noted in capital letters.

3.3 Crawler components

For simplicity, the single Crawler arm is divided into the following three subsystems, which are discussed in the following sections of this chapter.

- Frame of the Crawler
- Drive train for rotation of the Crawler
- Actuation of the Crawler arms

3.4 Frame of Crawler

The outer frame of the Crawler can be seen in figure 3.9. The frame is encasing the moving arms and is designed to keep the internal components safe as well as to be used for handling the Crawler while it is lifted between members.

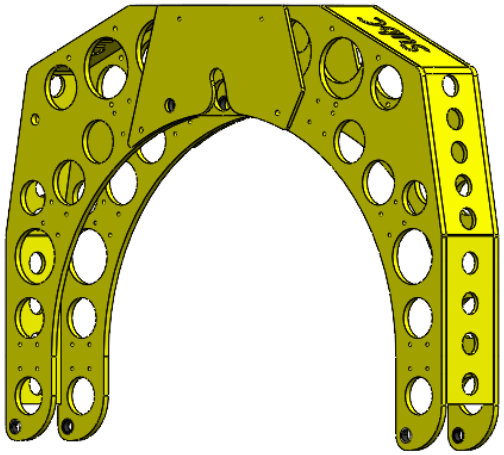


Figure 3.9: Case covering the mechanical parts of the crawler for protection and looks.

The frame is cut with a spider design, known by the multiple holes in the frame. This design is chosen to reduce the weight of the Crawler and allows for easier maintenance work if necessary.

3.5 Rotation

For rotating the Crawler around the member, it is equipped with a total of three wheels. The original Crawler is only equipped with a drive on one wheel, which is seen in figure 3.4a, colored blue. Whereas the non-actuated wheels, colored in light brown. The un-actuated wheels on the crawler are used for a uniform force distribution, ideally located 120° apart.

A hydraulic motor is used to power the single wheel, the location of which can be seen in figure 3.10.

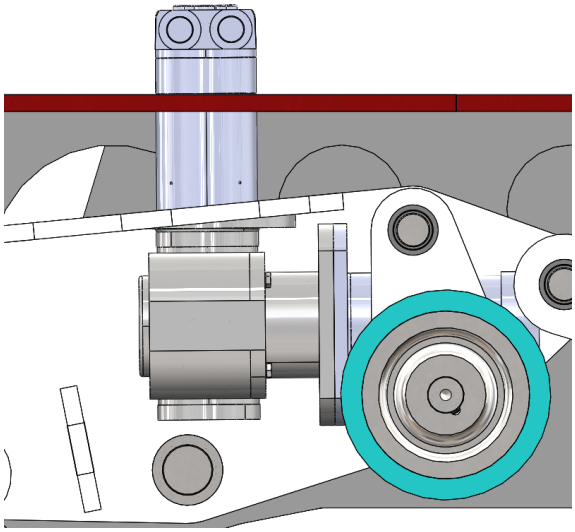


Figure 3.10: The hydraulic motor is attached to the axle which drives the wheel.

As depicted in figure 3.10, the hydraulic motor is attached to two angle gears in order to fit the motor into the crawler and connect it to the wheel axle.

The hydraulic motor used is a low-speed high-torque motor from Danfoss. The series number is 151G0013 and the motor has a geometric displacement of 50 $\frac{cm^3}{rev}$. The hydraulic motor has a maximum continuous torque of 46 Nm and a maximum continuous rotational speed of 400 rpm [11]. Through two angled gears, the torque output is increased to 145 Nm. The Crawler can rotate at a maximum angular velocity of 0.114 rad/s.

3.6 Crawler arms

The original Crawler is designed with two arms creating a half circle used to attach or detach on the member. Removing the outer frame, the arms can be seen depicted in figure 3.11 together with the wheels and hydraulic components attached to it.

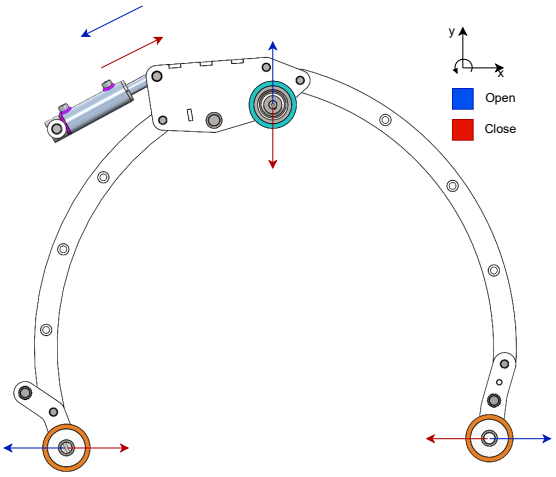


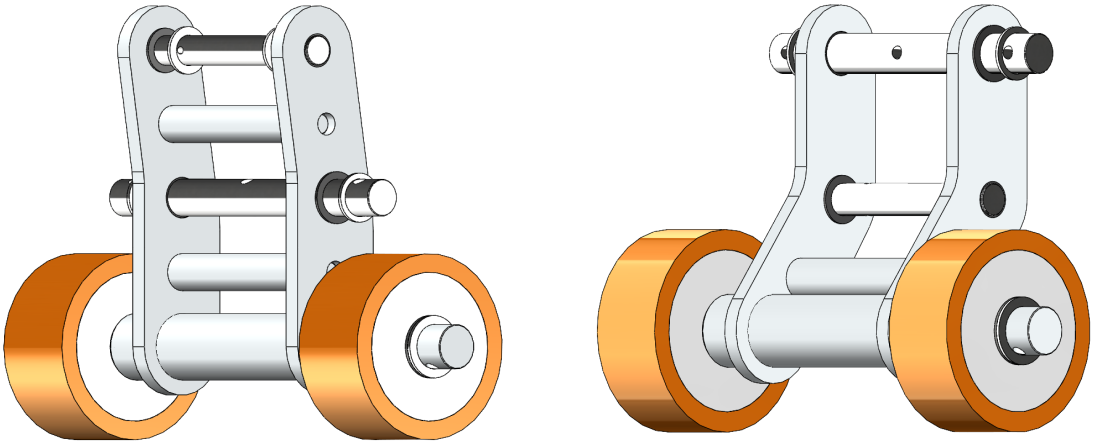
Figure 3.11: Crawler arms and hydraulic piston without casing.

As seen in figure 3.11, there is two arms connected to a component on top. This component is positioned on top of the crawler so that the motorized wheel, colored in light blue, is forced against the offshore structure when the hydraulic piston is extended. This component is not positioned in the center of the crawler but is shifted towards one side and attached to the crawler frame with an axle. By rotation around this axle, the motorized wheel is moved up and down, while the crawler arms will open and close their claws. The movement is illustrated in figure 3.11, showing the closing direction with red arrows and the opening direction with blue arrows.

3.6.1 Claws

At the end of each arm a claw to clutch to the offshore structures. The claws are equipped with the wheels discribed in section 3.5. The placement of the wheels on the claws allows the rotational movement.

The Crawler uses claws to hold on to and to rotate around the off-shore structures. These claws are positioned on the bottom of the crawler arms described in section 3.6. Figure 3.12 shows the claws with the attached wheels, sitting on the original crawler.



(a) Crawler claw attached to the right arm of the crawler, seen on figure 3.4a.

(b) Crawler claw attached to the left arm.

Figure 3.12: 3D CAD model for the crawler claws containing the non-motorized wheels.

As depicted in figure 3.12, the crawler claws do not have the same angle at which they are connected to the crawler arms. This is due to the arms being of different lengths. The difference of the claws together with the different arm lengths ensures a shift in wheel position on the offshore structure when clamped against it. This shift amounts to 120°, which is important for the uniform force distribution on the structure.

3.6.2 Actuation

The arms are actuated by a HydraSpecma hydraulic piston. The piston can be actuated in both ways, opening and closing the arms around the member. The piston including its dimensions can be seen in figure 3.13.

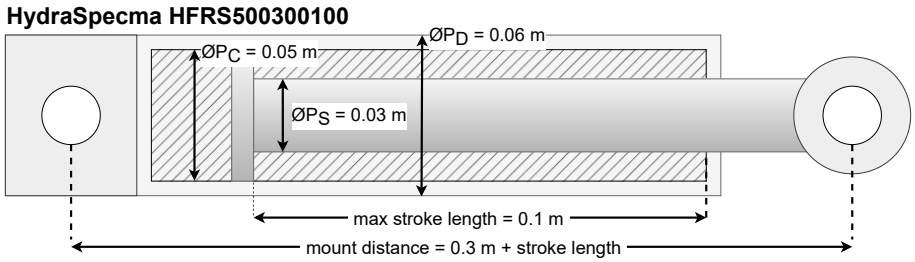


Figure 3.13: HydraSpecma hydraulic piston, with dimensions

The piston can extend to a maximum of 100 mm. and the maximum hydraulic pressure used to close the arms and attach to the member is set to 10 MPa. The diameter of the piston head is 50mm. This results in the total amount of force that the piston produces.

$$F_p = P_P \frac{{P_C}^2}{4}\pi \tag{3.3}$$

Where P_p is the hydraulic pressure and P_C is the inside diameter of the hydraulic piston. At 10 MPa the piston delivers a force of approximately 20 kN.

4 Static calculations

Compared to the original Crawler, some functions of the Crawler in this project have changed. Due to this, the limitations for the components have changed. In order to avoid overcompensation of the electrical components chosen for replacement, new static calculations are performed in this chapter. The results of these calculations are then used to set new boundaries for the components.

In developing these calculations, it is assumed that the forces acting onto the Crawler when in a static position are:

- Impact from waves
- Gravity
- Friction between Crawler wheels and the member

4.1 Friction

The static friction force between the wheels of the Crawler and the member occurs as a result of the gravitational pull of the Crawler. Thus, static friction between the crawler wheels and the member will always occur as a tangential force to the surface contact point between the crawler wheels and the offshore member. Therefore, to overcome this friction force, an opposite force must be applied. The magnitude of this force depends greatly on the material on which the Crawler is operating. The static friction force can be described by equation 4.1.

$$F_f = \mu_s N \tag{4.1}$$

Where μ is the friction coefficient and N is the normal force. Initially, the Crawler was developed using rubber wheels and a friction coefficient between rubber and underwater paint. This friction coefficient was determined to be 0.2, according to SubC. In updated versions of the Crawler, the wheels have been replaced with nylon. Due to this, as well as the fact that the Crawler will operate on various surfaces, experiments will be performed determining the friction coefficients between the nylon wheels and the surface materials mentioned in section 3.

4.1.1 Setup of friction experiment

This section will describe the experiment to determine the friction coefficients between the crawler wheel and the different operating surfaces. The results are further used later to describe the mathematical model of the crawler when fixed to the off-shore member.

The test is partly conducted on the campus of Aalborg University Esbjerg and partly at the SubC Partners A/S facilities.
The test is carried out with respect to the different surface conditions discussed in chapter 3 and as depicted in figure 1.5. The surface conditions are reconstructed in a realistic way. Furthermore, the test is performed on a test member, which can have a varying surface depending on the age and condition of the structure.
The experimental setup for the friction test is performed in a similar way in all surface conditions and is shown in figure 4.1.

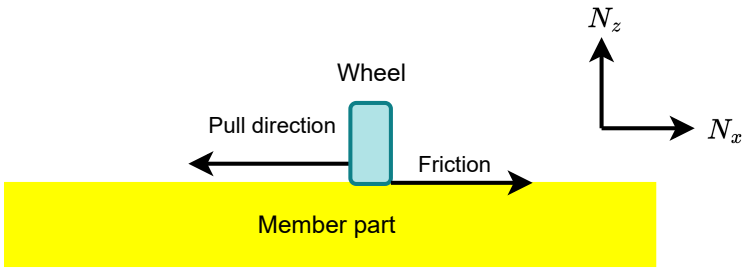


Figure 4.1: Overall friction test setup at SubC partners.

The test setup seen in figure 4.1 shows how the wheel is placed on the test surface and pulled in one direction. This ensures that the wheel does not roll, but is forced to slide across the surface. The materials used for these tests are listed in table 4.1

Table 4.1: Components used to perform the friction experiments.

Equipment	Producer	Model	Resolution	Range	Sampling rate
Force meter	PASCO	PS-2140	0.03 N	$\pm$ 50 N	1 kHz

The surface conditions of chapter 3 are smooth paint, rough paint, rust, mussels, and algae, all under wet and dry conditions. The experiment is conducted in this matter because the real system will operate in the splash zone in offshore operations. In this sense, the surface will change between wet and dry depending on how strong the waves are. Due to the irregularity of the mussels, tests on this surface have not been performed. The tests are performed on the following wet and dry surfaces:

- Smooth painted member
- Rough painted member
- Rusty metal plate

- Wooden beam with algae attached
- Greasy surface

Data for the experiment will be analyzed and validated with respect to the standard deviation. The standard deviation is the measure of how dispersed the measured data in relation to the mean. The data used for further validation must be in a range of two times the standard deviation to contain 95% of the observations taken in the test, which is indicated on the related test graphs. Data points that are not in the range of standard deviations are considered outliers and will not be included further in the results.

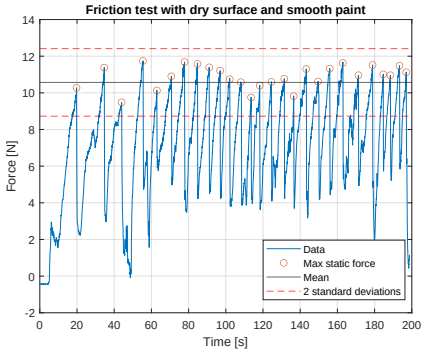
Dry smooth and wet smooth paint surface

The first and second friction tests are conducted on a dry and smooth paint surface depicted in figure 4.2a and a wet and smooth paint surface, depicted in figure 4.2b.

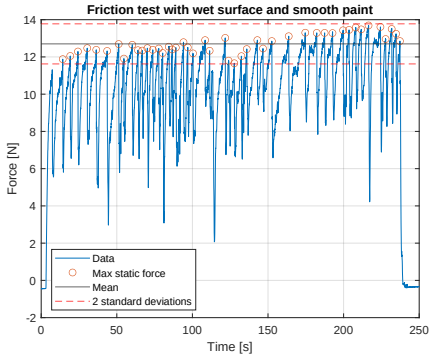
(a) Dry **(b)** Wet

Figure 4.2: smooth paint test surface, simulating a new and healthy offshore member.

The results are seen in figure 4.3a which illustrates the measured data in blue, the force peaks used are shown as red circles. The force peaks indicates where the static friction transitions to dynamic friction. The peaks are therefore the maximum static force that can be applied before it starts sliding and it changes to its dynamic friction, which can be seen by the drop in force required. Static friction is the one of interest, since if it changes to the dynamical value it would imply that the crawler is sliding down. The force can be used to determine the friction coefficient. The mean in figure 4.3a is represented by a black line with the two times the standard deviations seen as the red dotted lines, in order to filter out the data which is not validated.



(a) Graph of the first friction test for a dry surface with smooth paint.



(b) Graph of the second friction test for a wet surface with smooth paint.

Figure 4.3: Smooth paint test data for both the wet and the dry surface.

Figure 4.3a shows how much force is acquired to drag the wheel along the member. Further observation shows the wheels movement when the force is peaking, which means that enough force is built up to drag the wheel. The mean force to move the wheel is found to be 10.57 N with a standard deviation of 0.92. The drag coefficient for the first test is determined using the formula described in equation 4.1.

The friction coefficient (μ_s) between the wheel and the smooth and dry paint surface has a magnitude of 0.45 with an standard deviation of 0.04 and is noted in table 4.2.
Additionally, figure 4.3b shows the test results for the second tests force needed to drag the wheel over the wet surface. The mean force of the peaks is calculated to be 12.4396 N with a standard deviation of 0.54.

Just as in the first test, the different data information such as the mean and standard deviations are illustrated in figure 4.3b legends. The friction coefficient (μ_s) is determined to be 0.53 with its standard deviation of 0.02.

The test can be further expanded by testing more surface conditions, which are documented in appendix A.1.

All tests are compared with each other, showing the distribution of their data points to illustrate the accuracy of the experimental approach. This comparison is illustrated in figure 4.4, which shows the box plots for each friction test showcasing the friction coefficients and distribution.

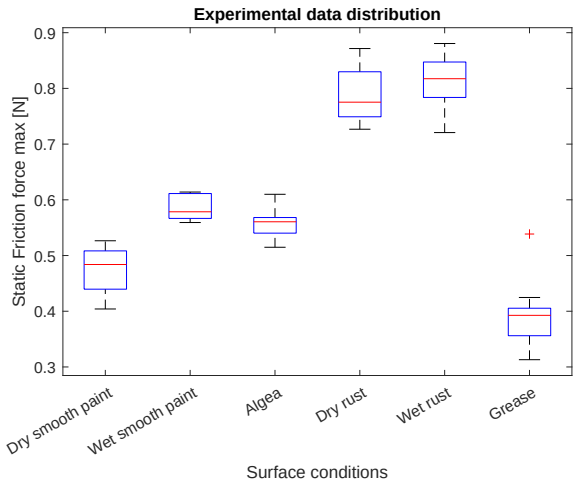


Figure 4.4: Boxplots for the different surface conditions, depictig the data distribution.

In order to observe the data distribution for the different friction tests, a box plot seen in figure 4.4 is made.

It is seen on figure 4.4, that none of the test coefficients show a significant characteristic of a normal distribution. The test closest to a normal distribution is the wet rust tests. The grease test is the only test which gives an outlier for the friction coefficient. Looking on the algae test, the distribution on the box plot is closer to a bimodal distribution.

For further observation, rather the data is normal distributed, the empirical rule is applied. The empirical rule implies that for normally distributed data, almost all observed samples fall within three standard deviations.

As viewed in figure 4.3a, the two-times standard deviation, marked as red dotted lines, indicates the data mostly lies in a normal distribution at the confidence interval of 95%. A null hypothesis is presented stating that the data is valid if it exists inside the boundary of 95%. However, the data can be considered to be normal distributed when having more data for disposal in order to obey the central limit theorem.
The experiment results in the determination of the friction coefficients of the different surface conditions. Furthermore, it validates that the different surfaces have a large impact on the friction coefficients depending on their condition. Although the highest friction coefficient is around 0.81 for wet rusty steel, the lowest coefficient is 0.37 for a greasy surface.
The summary of the experimental results and related calculated friction coefficients is shown in table 4.2.

Table 4.2: Table of test results and the calculated friction coefficients for the different conditions.

	Dry	Wet
Surface	Friction coefficient	Friction coefficient
Smooth paint	0.45 ± 0.04	0.53 ± 0.02
Rough paint	0.46 ± 0.07	0.46 ± 0.03
Algae	N/A	0.51 ± 0.2
Rust	0.79 ± 0.05	0.81 ± 0.05
grease	0.37 ± 0.05	N/A

4.2 Waves

The force from the waves is determined using the Morrison equation. This determines the inertia and drag forces on a cylindrical object submerged in water. For the following equations, the Crawler is assumed to be cylindrical. In addition to that assumption, it is assumed that the crawler is operating in the most frequent wave spectrum determined by JONSWAP, which described the wave frequency in the north sea which gives a time period of 12.25 s [12], the wave depth is 40 m in the Tyra field [13]. From this, the wave length can be determined using numerical iterations 4.2.

$$L = \frac{gT^2}{2\pi} \cdot \tanh\left(2\pi \frac{h}{L}\right) \tag{4.2}$$

Where L is the wave length and h the water depth both in meters. The time period is defined by T in seconds and g is the gravitational acceleration. Solving this equation numerically leads to a wave length of 116 m. From this the wave number k can be defined.

$$k = \frac{2\pi}{L} \tag{4.3}$$

Which is used to find the angular frequency of the waves.

$$\omega = \sqrt{\tanh(kh)kg} \tag{4.4}$$

With the angular frequency the wave speed can be determined throughout the waves motion. This is done using the following equation.

$$U_x = \frac{agk}{\omega} \cdot \frac{\cosh(k(z+h)}{\cosh(kh)} \cdot \cos(\omega t - kx) \tag{4.5}$$

where x is the amount the wave has propagated away from the starting location, which will just be set to 0. z is the depth at which the crawler is influenced. In these calculations, the crawler is situated just beneath the surface to ensure the strongest waves. The crawler is 0.15 m wide, which is then the z value. The a value is 2.5 m, which indicated how high the waves are. The crawler was designed to withstand 2.5 m, therefore, to ensure it can

hold up to its previous capabilities, this value is taken again. The acceleration field is determined by differentiating the velocity field with respect to time.

$$\frac{\partial U_x}{\partial t} = -\frac{agk}{\omega} \cdot \frac{\cosh(k(z+h)}{\cosh(kh)} \cdot \sin(\omega t - kx) \tag{4.6}$$

With the velocity and the acceleration for each moment in time determined. Its possible to examine the amount of force exerted on the crawler from the Morrison equation.

$$F_{wave} = F_I + F_d = \rho C_m V_c \dot{U_x} + \frac{1}{2}\rho C_d A U_x |U_x| \tag{4.7}$$

In the equation C_m is the inertia coefficient set at 2, with C_d being the drag coefficient evaluated at 1.5 [14]. The highest recommended coefficient is taken since its not a cylindrical object. The inertia forces also requires the volume displaced V_c in cubic meters this volume is determined from the solid work model to be 29·10^{-3} m^3, and the density of water ρ as well as the fluids acceleration $\dot{U_x}$. The drag forces are determined by the area that is hit perpendicular to the wave, which from solidworks resulted in 0.204 m^2 for vertically and 0.198 m^2 for horizontally. The area is similar, so the drag force for vertically is only calculated. The inertia and drag forces are plotted in figure 4.5.

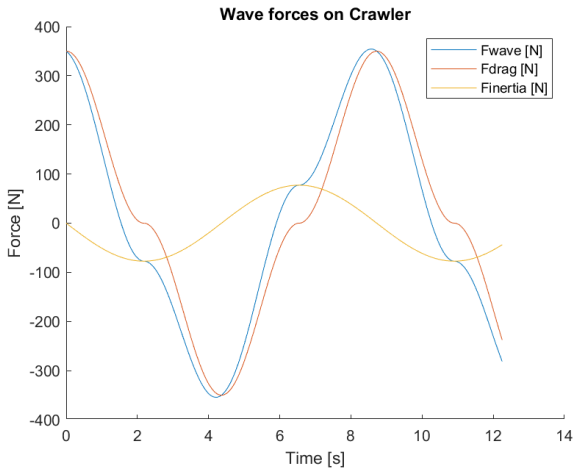


Figure 4.5: Morrison equation describing the forces acting onto the crawler through one wave cycle

As seen on the graph, it can be determined that the drag forces are the dominant part. The highest exerted force caused by waves is estimated to:

$$F_{wave} = 350N$$

This force is not constant and, in some cases, can be neglected, since it takes some time for the forces of the waves to move the crawler. Coupled with the face, the peak forces is only a short time. This is to avoid overdimensioning the crawler in aspects where it is not needed.

4.3 Static

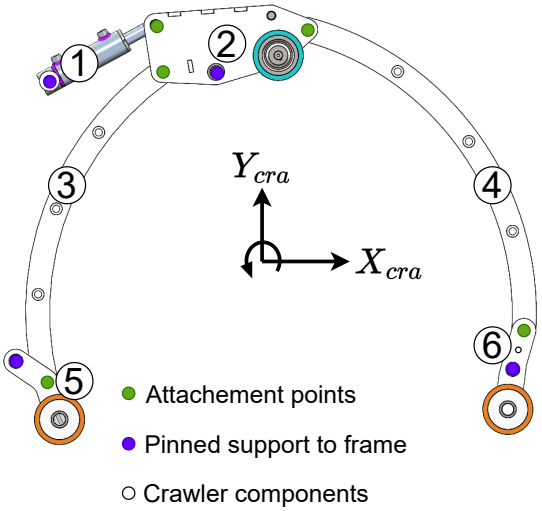


Figure 4.6: Overview of the crawler used for the static calculations. Where 1 is the piston, 2 the potato, 3/4 left and right arm, 5/6 left and right claw.

For the static calculations performed, it is assumed that the crawler is stationary attached to the member. When the Crawler is stationary, it can be assumed that the force at each wheel is evenly distributed, thus the force of the wheels are assumed equal.

$$N_1 = N_2 = N_3 = N_W \tag{4.8}$$

This is true when the crawler is hanging on a vertical member but when on a horizontal member this is a simplification since the force from gravity will affect each wheel differently depending on its rotation, which will be discussed later on.
Including these assumptions, a free body diagrams can be constructed of the Crawler. There are only moments in part 2,5,6. The first part 1, seen in figure 4.7 is the piston which includes the piston force F_P which is the one that is desired to be determined. This piston acts onto part 2, at an angle θ_P that changes depending on how clamped the crawler is. On the same join in part 2, the force from the crawler arm F_L is applied, with the other arms force F_R being applied to a different joint. Both the angle θ_L and θ_R changes depending on piston extension. The wheel is being pushed onto the member creating a force as well ensuring that the crawler stays put. All these forces together with their moment arm are depicted on figure 4.7 around the pivot point C. The resulting equation of moment is seen in 4.9. The length and angles were determined by using the autocad file, and moving it into its static position.

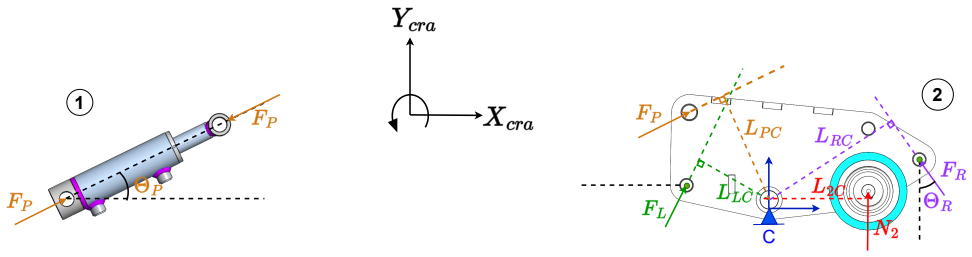


Figure 4.7: Force and moment arms on part 1 and 2.

$$\overset{\curvearrowleft}{+}\sum M_C = L_{2C}N_2 + L_{RC}F_R - L_{PC}F_P - L_{LC}F_L = 0 \tag{4.9}$$

Part 3 and 4 as seen in figure 4.8 have no pivot point and transfers the forces F_L and F_R down to the claws. Therefore no equations of motions are required here.

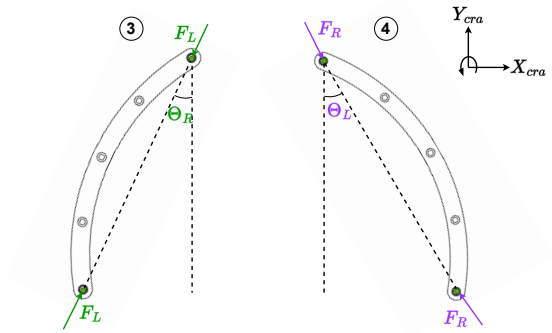


Figure 4.8: Forces on part 3 and 4.

part 5 and 6 as seen in figure 4.9.The pivot point in parts 5 and 6 are located differently, to ensure the same force on the member from the wheel. The angle that the wheel forces F_1 and F_2 acts on the member is at an angle of 30since they are all separated by 120°to ensure equal force on all wheels. Their moment arms are also drawn into the figure 4.9. Using the moment arms and forces from the crawler arms and wheels the moment equations around pivot point A and B can be established 4.104.11.

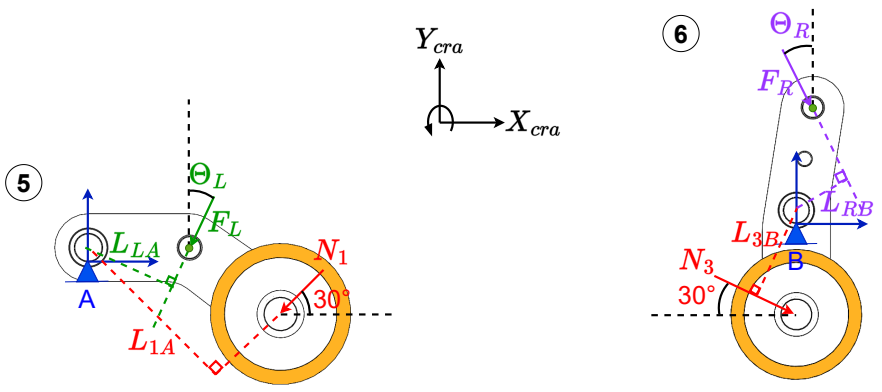


Figure 4.9: Force and moment arms on part 5 and 6.

$$\overset{\curvearrowleft}{+}\sum M_A = -L_{1A}N_1 - L_{LA}F_L = 0 \tag{4.10}$$

$$\overset{\curvearrowleft}{+}\sum M_B = L_{3B}N_3 - L_{RB}F_R = 0 \tag{4.11}$$

Assuming that the forces on the wheels are equal, it ends up with 4 unknowns and 3 equations. Therefore, additional equations need to be established to find the minimum piston force.

Forces acting on vertical hanging Crawler

To find the specifications of the crawlers piston, its important to examine the amount required to keep the crawler from slipping and falling down. Having the crawler hanging on the side of a vertical member is harder than horizontal, since then its the friction holding it up not the normal force from the wheels resting on it. From the free body diagram 4.10 the forces in the N_x directions can be determined as stated in equation 4.12.

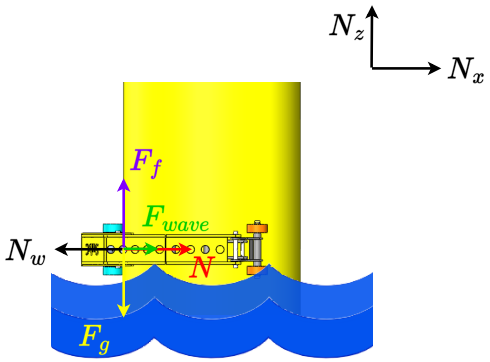


Figure 4.10: Free body diagram of crawler on vertical pipe

Since there are three wheels, the wheel normal force is multiplied by 3.

$$\sum N_x = N - 3N_W + F_{wave} = 0 \tag{4.12}$$

From this equation the total normal force can be determined while influence by waves. The same can be done in the z direction resulting in equation 4.13.

$$\sum N_z = F_f - F_g = 0 \tag{4.13}$$

The equation for friction force, 4.1, can be rewritten to 4.14, by solving 4.12 for the normal force.

$$F_f = \mu(3N_W - F_{wave}) \tag{4.14}$$

From here the normal force can be determined 4.15.

$$0 = \mu(3N_W - F_{wave}) - F_g \Rightarrow N_W = \frac{1}{3}(\frac{F_g}{\mu} + F_{wave}) \tag{4.15}$$

The four equations 4.94.104.114.15, are made into a system of equation and solved.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{1}{3}(\frac{F_g}{\mu} + F_{wave}) \end{bmatrix} = \begin{bmatrix} -L_{PC} & -L_{LC} & L_{RC} & L_{2C} \\ 0 & -L_{LA} & 0 & -L_{1A} \\ 0 & 0 & -L_{RB} & L_{3B} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_P \\ F_L \\ F_R \\ N_W \end{bmatrix}$$

The results are presented in Table 4.3, with all moment arm lengths and forces. The coefficient of friction was 0.37 since that was the lowest value found from the friction test 4.2. From the 3D model the crawler was determined to have a weight of 92 kg.

Table 4.3: All the resulting forces and moments on the crawler in its static position

Force in [kN]	F_P:3.4	F_L:-1.8	F_R:1.0	N_1:0.93	N_2:0.93	N_3:0.93	F_f:0.90	F_g:0.90
length in [mm]	L_{PC}:130	L_{LC}:129	L_{RC}:240	L_{1A}:174	L_{2B}:99	L_{3C}:157	L_{LA}:89	L_{RB}:90

Therefore, the piston must provide a minimum force of 3.4 kN to keep it in its static position when hanging vertically. This is 17 kN less than what the current SubC solution provides 3.6.2.

Forces on horizontal lifting crawler

The crawler needs to be able to hang onto the pipe horizontally as well as be able to lift the cutoff pipe. It is currently rated to lift 800kg, with 2 clamps so the one clamp needs to be able to lift 400kg. A free body diagram of the crawler is shown in figure 4.11.

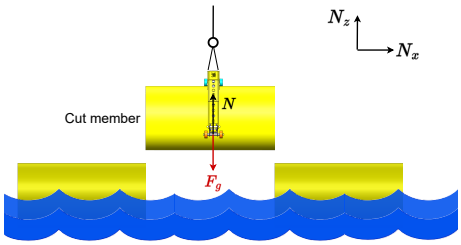


Figure 4.11: Free body diagram of the crawler holding a member seen from the side

It does not need to be able to lift its own weight since that is being lifted by the hoisting system only the weight of the cut of the member is required. The forces acting from the waves have been neglected since the member is most of the time out of the water when lifting. The lifting forces required by the crane are not examined in this report. A free-body diagram seen from the front is visualized in figure 4.12.

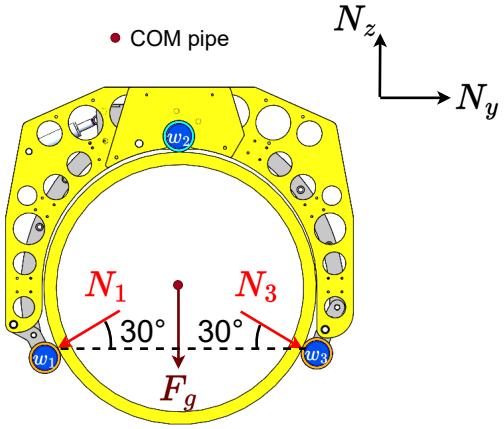


Figure 4.12: Free body diagram of the crawler holding a member seen from the front

From the figure it can be seen that there is only force applied from wheel 1 and 3. Wheel 2 does not contribute to holding the member and is 0 at its minimum static position. The wheels 1 and 3 are being clamped onto the member, their force must be equal to keep it in static position. The vertical component of their applied force must combined counteract the force of gravity. If they clamp further then the remaining force will be applied into wheel C. The angle between normal force 1 and 3 can be determined from the knowledge of it being 120between each wheel, and then using geometry.

$$N_1 = N_3 = N_W \tag{4.16}$$

Establishing the forces in both y and z direction leads to the following equations 4.174.18

$$\sum N_y = -N_1 \cos(30) + N_3 \cos(30) = 0 \tag{4.17}$$

$$\sum N_z = -N_1 \sin(30) - N_3 \sin(30) - F_g = 0 \Rightarrow 2N_W \sin(30) + F_g = 0 \tag{4.18}$$

From the equations it can be deducted that the assumption with them being equal force must be correct for equation 4.17 to be true. With the use of 4.18 the force from the piston can now be calculated using the static equations previously established 4.94.104.11. With the only difference the force N_2 from the wheel in moment C is equal to 0. The system of equations established is then solved for the piston force.

$$\begin{bmatrix} 0 \\ 0 \\ 0 \\ \frac{F_g}{2\sin(30)} \end{bmatrix} = \begin{bmatrix} -L_{PC} & -L_{LC} & L_{RC} & 0 \\ 0 & -L_{LA} & 0 & -L_{1A} \\ 0 & 0 & -L_{RB} & L_{3B} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} F_P \\ F_L \\ F_R \\ N_W \end{bmatrix}$$

The resultant piston force is F_P=11 kN for one clamp, when holding the 800 kg member which it is currently DNV certified to. To ensure that it is able to withstand the forces an additional safety factor is applied of 1.3 to all the loads [15]. This results in a force of 14.5 kN for the piston.

Torque required for the crawler to rotate on vertical member

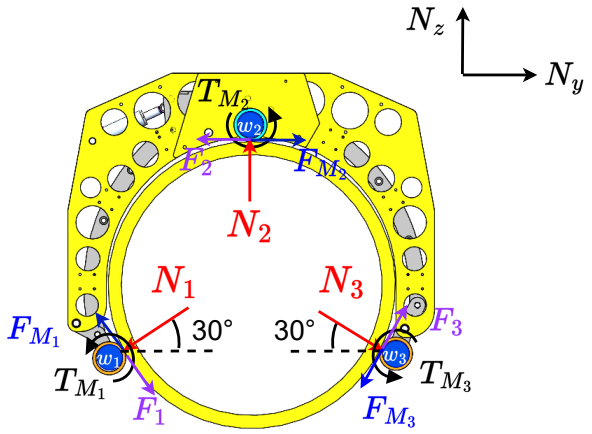


Figure 4.13: Free body diagram of crawler hanging on vertical member, to establish motor torque required

The wheel force required to keep the crawler form slipping when hanging on a vertical member was already calculated in 4.3. With a wheel force of 930 N on each of the wheels 4.13. the rolling resistance the wheels need to overcome is then defined from equation 4.19.

$$F_r = \mu_r \frac{N}{r_w} \tag{4.19}$$

The rolling friction coefficient μ_r is taken to be 0.027 between nylon and steel [16]. This results in 392 N of rolling Resistance. This is the friction force that each wheel have to overcome. All three wheels are going to be individually actuated having a radius of 0.0625 m, so the required torque to rotate the crawler becomes 4.20.

$$T_m = r_w \cdot F_r \tag{4.20}$$

So each of the motors needs to provide a minimum of 24 Nm.

Torque required for the crawler to rotate on horizontal member

To find the required torque for each motor to ensure it is able to rotate around the whole member it is examined in it's worst position. This is when the second wheel is aligned with the N_y, N_x plane.
The free body diagram depicting the forces acting on it is shown in figure 4.14.

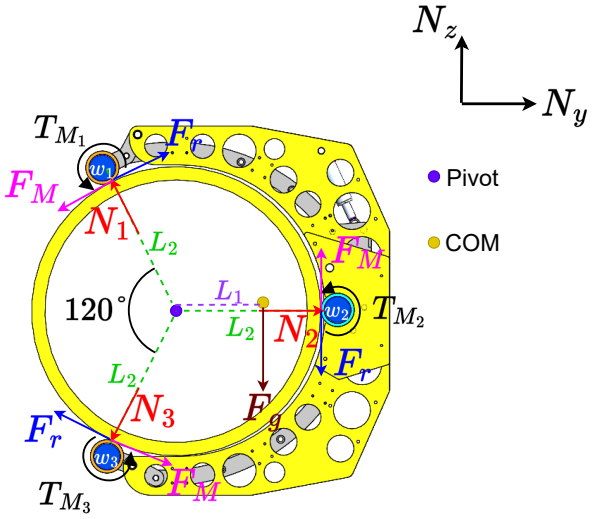


Figure 4.14: Free body diagram of crawler hanging on horizontal member, rotating from 90°angle to 0. Keeping same piston force.

Where L1 and L2 are the moment arms from the pivot to the center of mass and wheels contact surface of 0.384 m and 0.58 m. The center of mass is taken from the CAD model, and then aligned to the N_y, N_x plane for simplifications, the wave forces are taken to be the 350 N found in 4.5. The only friction force examined here is rolling friction which is found using equation. 4.19. These simplifications are to get a rough specifications required for the motor, where more detailed modeling will happen later. The equation of motion for the crawler is then 4.21

$$\sum M_{cra} = L_2(3F_{MW} - \mu_r \frac{(N1 + N2 + N3)}{r_w} - F_{wave}) - F_g L_1 = 0 \tag{4.21}$$

The three normal forces each vary in size due to the crawlers orientations. A force balance in N_y and N_z are established as following.

$$\sum N_z = N_1 sin(\theta_{w1}) + N_2 sin(\theta_{w2}) + N_3 sin(\theta_{w3}) - F_g = 0 \tag{4.22}$$

$$\sum N_y = N_1 cos(\theta_{w1}) + N_2 cos(\theta_{w2}) + N_3 cos(\theta_{w3}) = 0 \tag{4.23}$$

The piston force was varied to see its impact on the motor force 3.6.2 and the angles $\theta_{wi}, i \in \{1, 2, 3\}$ are depicted in figure 3.8 with a 120 °angle between. Solving equations 4.94.104.114.224.23 gives the required force for each motor of seen in table 4.4. The torque delivered by each motor can be found by multiplying the radius of the wheel with the force as stated in equation 4.20.
this shows the importance of not over utilizing the piston as it requires a larger motor to be able to rotate the crawler. At a piston pressure of 2 kN the wheel 1 has no normal force and would not be able to drive since it has no contact. This is also accounting waves affecting it with 350 N of force. But a minimum of 34 Nm needs to be provided from the motor to ensure it can rotate fully.

Table 4.4: Force and torque required from the motors at different piston pressures.

F_P [kN]	F_{MW} [kN]	T_{MW} [Nm]	N_1 [kN]	N_2 [kN]	N_3 [kN]
15	2.1	130	3.6	4.1	4.7
5.0	0.91	57	0.85	1.4	1.9
2.0	0.55	34	0.0	0.54	1.1

4.4 Summary of forces

The requirement for the piston is then a minimum of 3.4 kN to ensure that it can keep the crawler in place when vertical. This would need to be 11 kN if it needs to keep its requirement of lifting the 800 kg member. The minimum torque required to rotate is 24 Nm when it is attached vertically, and 34 Nm when horizontally for each motor.

5 Electrification

Before developing mathematical models of the Crawler and designing the control system, the hydraulic components of the Crawler must be replaced with electric ones. This is also a goal for this project. This chapter contains the process of this conversion. Based on findings in chapter 4. The rotating drive train and wheels are upgraded, as well as the system that actuates the clamping arms. The Crawler originally did not have any onboard control systems. Components for this are also chosen, and an operating system is installed and configured.

5.1 Drive system

A previous study was carried out investigating whether it is possible to replace the hydraulic motor with a Maxon ø40 170 W electric motor, as seen in [8], the data sheet of it is in appendix B. From this report, it was found that a single motor would not be sufficient to perform one full rotation of the Crawler around a horizontal member. However, as concluded through simulations in the report, it should be possible for three motors to achieve enough torque for rotation. This section will seek to reaffirm the feasibility of replacing the single hydraulic motor with three wheel drive, powered by a Maxon motor each.

The Maxon EC motor is an electronically commutated (EC) motor, which runs on an AC power supply, while having a closer resemblance to a DC motor. The EC motor consists of a permanent magnet, brushless DC motor with incorporated onboard electronics. This allows the EC motor to make use of the features of the AC motor and the DC motor [17]. Furthermore, the EC motor is more efficient than an AC motor due to reduced copper heat loss, reduced slip and inverter loss [18].
The motor parameters can be found in table 5.1.

Table 5.1: Maxon EC motor parameters [19][8].

Characteristics	Value	Units
Nominal current	7350	mA
Thermal time constant winding	18.7	s
Max speed	18000	rpm
Number of pole pairs	1	-
Terminal resistance phase to phase (R)	0.209	Ω
Terminal inductance phase to phase (L)	0.0843	mH
Torque constant (K_t)	23.2	$\frac{mNm}{A}$
Speed constant (K_e)	412	$\frac{rpm}{V}$
Speed/torque gradient	3.71	$\frac{rpm}{mNm}$
Mechanical time constant	2.09	ms
Rotor inertia	53.8	gcm^2

Each motor is mounted inside a stainless steel casing, which is designed to be filled with mineral oil, to make the motor water-proof. The casing is mounted to the frame of the Crawler by brackets designed for this specific purpose. The brackets are designed to ensure that the chain used to drive the wheels can be tightened.

5.1.1 Drive train

The drive train for the top wheel is based on the design proposed in [8], this can be seen in figure 5.1.

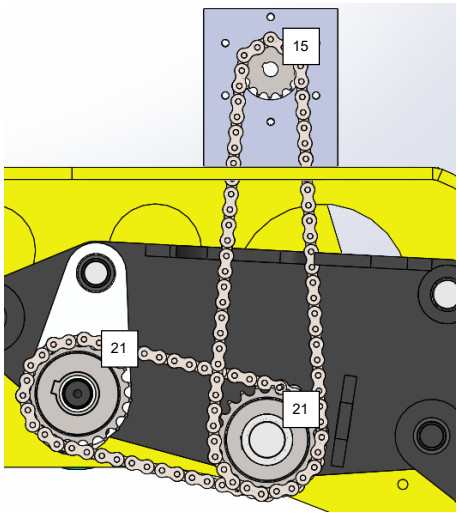
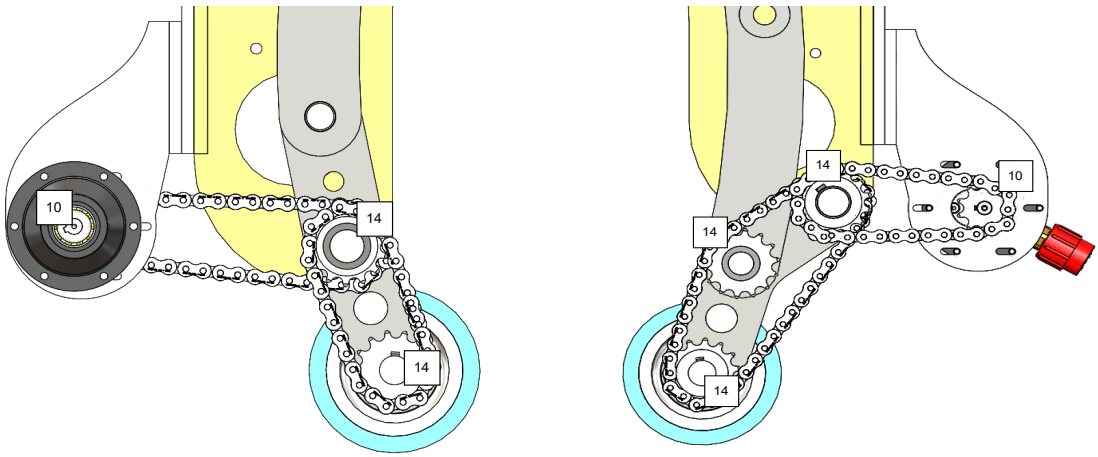


Figure 5.1: Depiction of the crawler electric drive train established in the former report [8].

The drive trains for the two outer wheels have been completely redesigned. The two outer wheels and axles are updated to the same model as those used for the single wheel

drive of the Crawler to allow the axles to be actuated. Since these wheels move towards the member when the arms are closing around it, they move away from the outer frame of the crawler. Due to this motion, the drivetrain is set up in a manner such that the motor drives a shaft located on the inner frame, which then again drives the wheel shaft which is mounted on the outer frame. All components used for upgrading the drivetrain have been tested inside the 3D model of the Crawler arm before implementation on the physical set-up. The redesigned crawler claws are depicted in figure 5.2.



(a) Crawler claw one seen from a 2D perspective showing the drive train over three gear wheels.

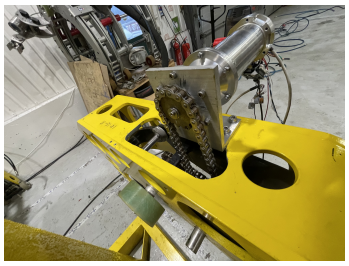
(b) Crawler claw two seen from a 2D perspective illustrating the drive train over four gear wheels.

Figure 5.2: Depiction of the new designed crawler claws in an 2D perspective.

Each motor is directly attached to a GP52C gearhead that has a gear ratio of 353:1. The gearhead is designed for a maximum speed of 6000 RPM and a maximum torque output of 30 Nm B. For the top wheel, a 21:15 gearing is added, whereas for the two outer wheels, a 14:10 gearing is added. This design choice was made to maintain the same gearing for each of the wheels. However, due to physical limitations, the total size of the gears needed to be reduced for two wheels. All gears used for the drive train are chosen to fit a 08 B1 simplex chain.

The redesigned drive trains and motors with mounting are mounted on the Crawler, and the result can be seen in figure 5.3.

(a) A low-speed gearwheel connected to the shaft of the Maxon motor and gearbox, inside the waterproof casing.



(b) Installation of the motor, gearbox combination onto the crawler.

Figure 5.3: The electrical motor with casing and installation on the crawler

5.1.2 Motor torque calculation

To reaffirm the hypothesis stated in [8], the total torque delivered through the wheels powered by three Maxon motors is derived through calculations and compared to the required torque from 4.4.

The motor torque can be determined as seen in equation 5.1.

$$T_m = I_n K_t \tag{5.1}$$

Where T_m is the torque of the motor, I_n is the nominal current of the motor, and K_t is the torque constant. The motor torque is found to be 0.1705 Nm. When gears are applied to the motor, it affects the speed and torque output. To determine the torque output when applying a gear, equation 5.2 can be used.

$$T_{out} = T_{in} G \tag{5.2}$$

Where $G = \frac{Z_1}{Z_2}$, Z_2 is the number of teeth of the gear attached to the motor and Z_1 is the number of teeth on the gear driven by Z_2. Inserting the amount of torque found in 5.1 together with the gear ratio of the first gear results in a torque output of 60 Nm. This is above the allowed maximum continuous torque of the gearhead of 30 Nm B, hence, this output is limited at this amount. To find the torque available on the wheels, 30 Nm will be inserted into equation 5.2 together with the second gear ratios.

$$T_w = T_m G_1 G_2 \tag{5.3}$$

Where T_w is the wheel torque, G_1 is the first gear ratio, and G_2 is the second gear ratio. The torque available at each wheel is 42 Nm. This results in a total available torque from three wheels of 126 Nm which is above the 97.5 Nm found in section 4.3.

The speed of the wheels can be found using the gear relations as well. Since the speed has inverse proportions to the torques when going through a gear, the following equation can be used to state the change in speed through the gearings.

$$\omega_{out} = \frac{Z_2}{Z_1}\omega_{in} \tag{5.4}$$

Where ω_{out} and ω_{in} denote speeds after and before gear reduction. Since the maximum input speed of the gear is 6000 RPM or 628 rad/s, this will act as the maximum motor speed before gearing. Using the equation 5.4 the speed in each gear is found to be 1.78 and 1.27 rad/s. The relationship between gears, torque, and speed can be seen in table 5.2.

Table 5.2: Torque and speed characteristics throughout the gears with 6000 RPM as continuous input motor speed. Note that the torque is saturated at 30 Nm due to limitations of the gearhead

Characteristics	Motor	Top wheel	Outer wheels
Gearing	353:1	15:21	14:10
Max continuous torque [Nm]	30	42	42
Max continuous output speed [rad/s]	1.78	1.27	1.27

5.1.3 Motor controller

The motor is controlled by a local Maxon EPOS 4 control unit. This control unit acts as a controller for the motor, converting DC input into 3 phase needed to drive the motor. The control unit is able to run position and velocity control, as well as system identification, on each separate motor. This ensures that the dynamics of each unique motor is captured. Feedback such as torque, velocity, and position can be gathered from the EPOS 4.

Each motor has an internal encoder attached. The encoder is connected to the EPOS. This encoder consists of four hall sensors mounted around the motor shaft. The hall sensors measure distance as increments. Thus, one full motor rotation is divided into four increments. Converted from the motor shaft to the wheel position, one motor increment is equal to 0.18° wheel position.

5.2 Clamping system

To replace the actuator that controls the arms of the Crawler, the new component should be able to fit the dimensions as seen in figure 5.4. The current setup is limited to moving the axles or replacing their diameter. However, the design leaves room for placing multiple actuators in parallel if needed. It is also possible to fit an actuator that extends backwards from the axle on the left in the figure.

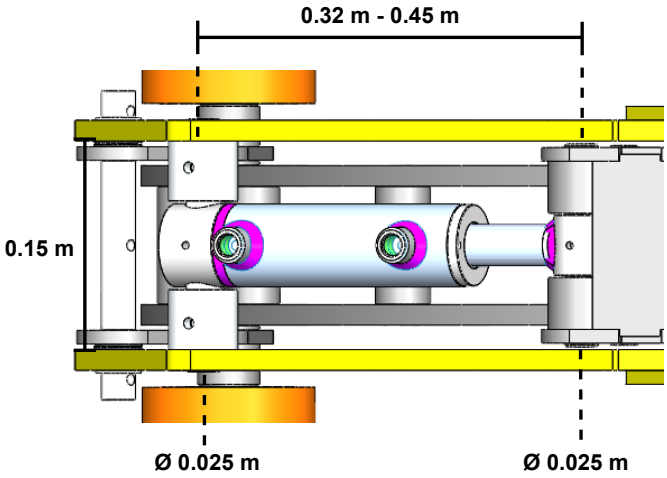


Figure 5.4: Physical limitations for mounting the actuator on the Crawler

A good choice for replacing the hydraulic actuator could be an inline electric actuator. This is also referred to as a lead screw linear actuator. It uses a planetary gear to convert the rotational motion of the electric motor into linear movement, moving the piston rod. Often used within the industry when high position accuracy is needed. This actuator offers a compact solution, much like its hydraulic counterpart. The piston and electric motor are aligned after each other. The KLEEline 38 is a Danish-developed inline linear electric actuator that is designed to be cold-resistant and waterproof, thus incorporating marine and offshore use. It can operate between -20 and 60 °C and is IP69K rated water and dust resistant. It can be powered by 12 or 24 V, having a maximum current use of 3.8 and 1.8 A, respectively.

For this project, a parallel setup of two 250 mm stroke length KLEEline 38 pistons is chosen. Combined, these can deliver a load force of 3600 N and a locking force of 7200 N, which surpasses the requirement of 5890 N. They can operate at a maximum speed of 3.3 m/s and fit the dimensions of the existing Crawler frame.

5.2.1 Actuator controller

The pistons are controlled by an EM-339A-SPF control unit which is able to run the pistons in a parallel setup, ensuring that they both run at synchronized speed and distance. This is obtained by pulse feedback from hall sensors mounted on the electric motors. The input of the controller is an analog signal from 0 to 10 V DC, making it possible to control the closing speed of the arms. The controller can be connected to the CAN-bus and can output piston position and current use. Load cells can be mounted to measure pressure force directly. [20].

The specifications of the actuators used and the control unit can be found in table 5.3

Table 5.3: Specifications of the KLEEline actuator and control unit

Actuator (KLEEline 38, 250)		
Max stroke length	250	[mm]
Full length	230 + stroke length	[mm]
Actuator speed	3	[mm/s]
Load force	1800	[N]
Locking force	3600	[N]
Input voltage	12 or 24	[VDC]
Input current	3.8 or 1.8	[A]
Control Unit (EM-330A-SPF)		
Control input	0-10 VDC analogue	
Feedback output	Actuator position and current usage	
Input voltage	12 to 35 VDC	
Output current	2x10 A continuous or 2x20 A with 25% duty-cycle	

5.2.2 Conclusion on electric actuator

Further tests is performed to ensure that the electric actuators are able to deliver enough force to keep the crawler firmly attached to the vertically oriented test member. A test is performed in which the crawler is attached to the member on the rough surface. The piston force is initialized at 15 kN and reduced until it starts to vertically slip from the surface. The vertical slip is identified to occur at 7.85 kN, which is above what two electric actuators are able to deliver. Due to this, the initial plan of using the electric actuators is scrapped and the hydraulic system is kept. For further reference, an extra actuator delivering 3.6 kN extra could be added, since there is enough room for this inside the crawler frame.

5.2.3 Clamping electrification

As mentioned, the hydraulic actuator is not replaced by an electrical. However, to control and measure the amount of pressure which the piston is actuated with, an electrical system is developed. For this, the following components are used.

- **SCP01 Pressure transducer**
  Described in 5.3.1 This sensor is used to measure the hydraulic piston pressure.
- **LM358 Operational Amplifier (Op-Amp)**
  Used to step-up the actuator voltage to a magnitude which can be used by the proportional valve.

- **PVEH Hydraulic proportional valve**
  Used to control the flow of hydraulic fluid in and out of the piston.

- **Arduino Due**
  The micro-controller is used to gather measurements from the pressure sensor and to actuate the proportional valve.

A diagram of the electrical connections can be seen in figure 5.5.

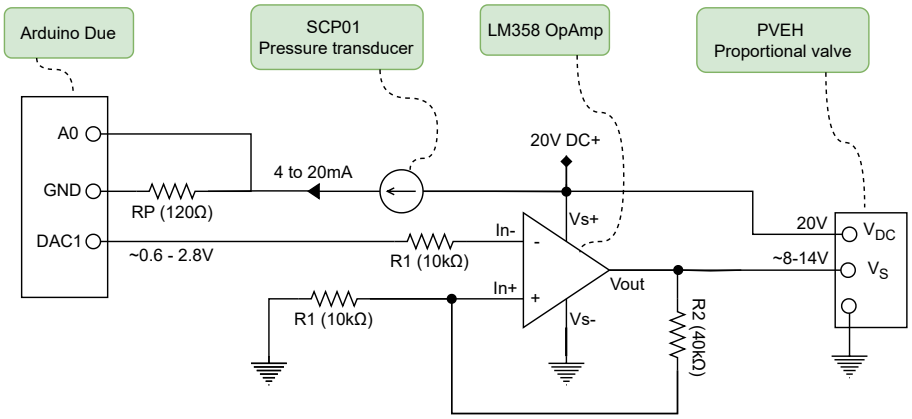


Figure 5.5: Electric circuit used to control the proportional valve and measure the piston pressure

From figure 5.5 it can be seen that a 20 VDC voltage source is used to power the Op-Amp, the pressure transducer, and the proportional valve. The Arduino Due is used to create a varying voltage signal that is stepped up to a range between 8-14 V which can be used to change the opening of the hydraulic piston. A further description of the piston control and pressure measurements is found in the following sections.

Controlling the piston

Originally the crawler piston is actuated using an open-close valve, manually controlled by a lever. For this setup, the valve is changed to a Danfoss proportional valve (157B4832), which can be controlled by an electric analog signal. This enables a smooth control of the hydraulic flow fluid in and out of the piston. The valve is powered by a constant voltage between 11 and 32 VDC and can be controlled by a DC+ voltage signal ranging between 25 and 75% of the DC source voltage. at 25% the valve is fully open at port A, and at 75% it is fully open at port B, at 50% it is neutral, meaning there is no flow through the valve. The valves opening and its corresponding flows can be seen in figure 5.6.The control signal is generated from an Arduino Due. Using a digital-to-analog converter (DAC) to generate a signal between 0.55 and 2.75 V. This value is then increased through a non-inverting operational amplifier circuit as depicted in figure 5.5.

figure 5.7

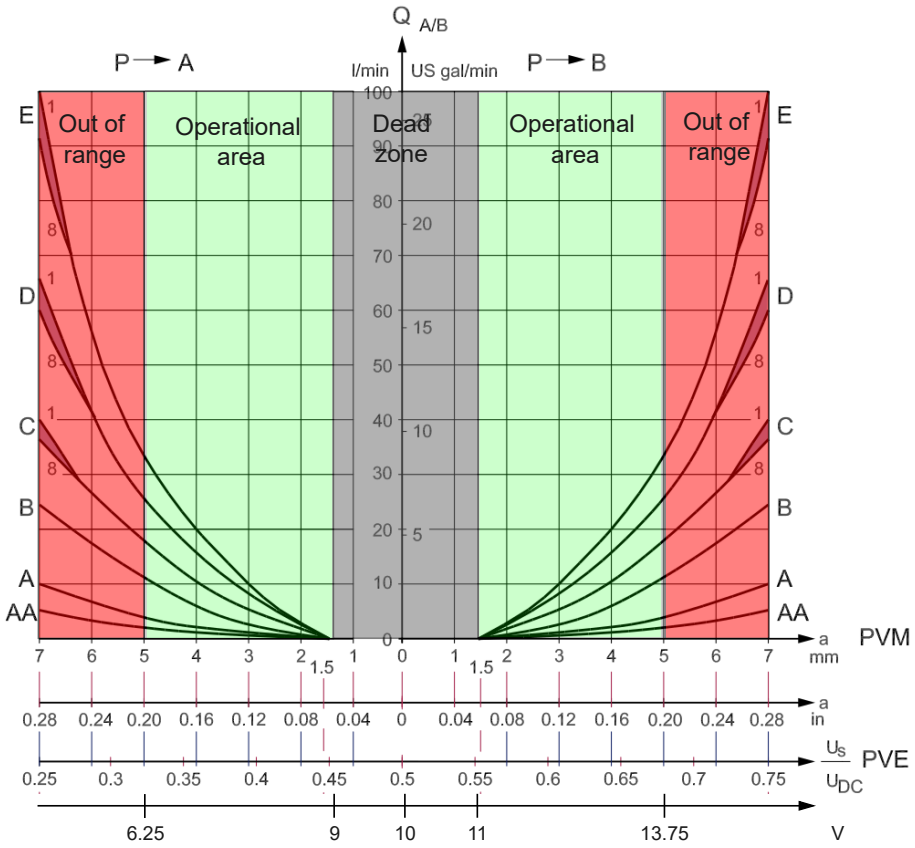


Figure 5.6: Valve opening and its flow through the valve, depending on control signal [21].

Taking 20 V as the supply voltage requires an amplification of 5 to the signal from the Arduino. To reach 13.75 V, which is only 68% of the supply voltage, which is less than the ideal 75%. The lower limit is set intentionally to avoid a too fast response of the hydraulic piston, accidentally completely opening the piston arms and releasing the crawler.

As mentioned, the Op-Amp should increase the analog voltage with a multiplication factor of five. In order to design the circuit with the correct magnitude of resistors, equation 5.5 can be used.

$$V_{gain} = 1 + \frac{R_2}{R_1} \tag{5.5}$$

Here R_1 and R_2 are resistors found in the circuit in figure 5.5. To achieve the correct multiplication factor, a 10 kΩ resistor is chosen for R_1 and a 40 kΩ resistor is used for R_2.

Pressure measurement

In order to develop feedback control for the hydraulic piston, the piston pressure must be measured. An SCP01 pressure transducer is currently already installed in the setup, connected to the main control panel seen in figure 3.2. The pressure transducer receives a voltage ranging from 9 to 24 VDC and outputs a current between 4 to 20 mA. To read this signal using the Arduino Due, a resistor of 120 Ω (R_p) is mounted in series with the SCP01. The voltage drop across R_p is measured using the A0 analog input port of the Arduino

Due. By applying different pressures and comparing the readings on a manometer with the analog input on the Due, a linear regression is used to describe this relationship. The linear regression is fitted with an R^2 value of 0.99 as seen in figure 5.7

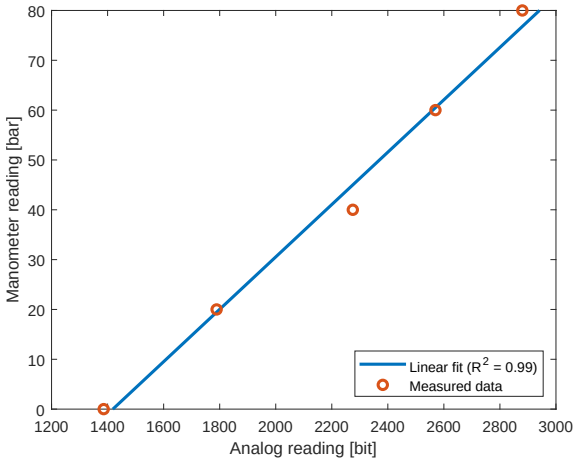


Figure 5.7: Analog input to pressure relationship

5.3 Sensors

A sensor package is installed on the Crawler arm. The main purpose of the sensors is to measure the angular position and velocity of the Crawler and the individual wheels, as well as the force excited by the electric actuator. The following sensors are installed in the package.

- Pressure transducer to measure the piston pressure
- Low resolution incremental encoder for measuring motor shaft position and velocity
- Torque sensor to output the torque generated by each wheel
- Inertial Measurement Unit (IMU) for measuring angular position and velocity of the Crawler in world frame coordinates
- High resolution camera for Crawler position and velocity using Computer Vision

The location of these sensors on the Crawler can be seen in figure 5.8.

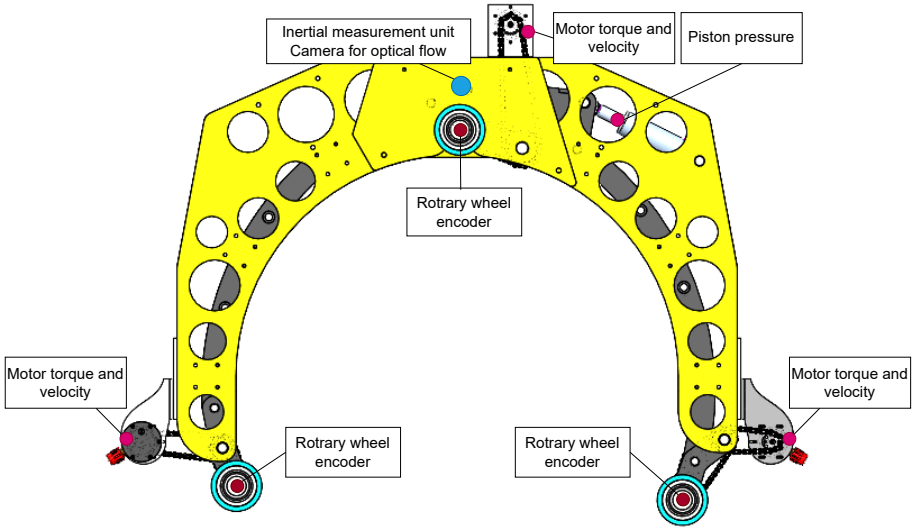


Figure 5.8: Laboratory setup used for validating the algorithm used for the basis tests

The incremental encoder, actuator piston position sensor, and torque sensors are described in sections 5.2 and 5.1, respectively. The pressure transducer used to measure piston force, as well as the IMU and the camera used, is explained further in this section.

5.3.1 Pressure sensor

To measure the force actuated by the hydraulic piston. A Parker SCP01-250-34-06 pressure transducer is used. This is a two-wire sensor that has a measurement range between 0 and 250 bar. Its operating voltage is between 9 to 32 VDC and the output is ranging between 4 to 20 mA. The precision of the measurements is $\leq \pm 0.25\%$. The implementation of this sensor is described in section 5.2.3, and by using this implementation, a steady state test is performed, the results from this can be seen in figure 5.9.

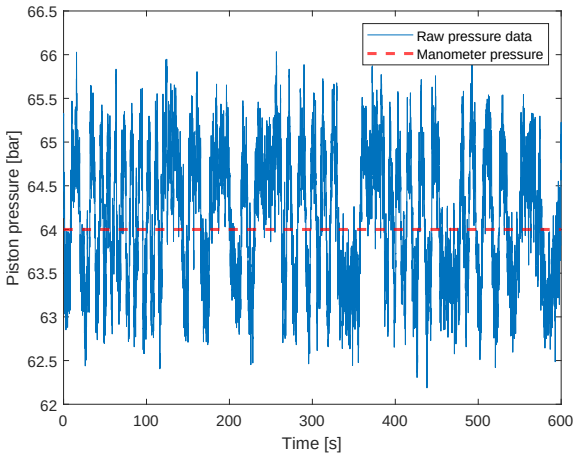


Figure 5.9: Steady state data from the SCP01 pressure sensor connected through an Arduino Due, compared with a reference pressure of 64 bar.

Here it can be seen that there is a fluctuation of the raw measured data around the actual

pressure of 64 bar. The standard deviation within the steady-state test is calculated to 0.75 bar.

5.3.2 IMU

To measure the angular position and the velocity of the Crawler, a gyroscope can be used. The gyroscope measures the angular position around three axes, which are fixed against the gravitational pull. An inertial measurement unit consists of a gyroscope, magnetic sensor, and a three-axis accelerometer. In general notation, the three axes are denoted as x,y, and z, where z is fixed in parallel to the gravitational pull. Rotations around these axes are denoted roll, pitch, and yaw, respectively. The IMU used in this project is a TM352 that is made by SydDynamics. This particular IMU is designed for industrial use and has an IP rating of 67. It has a low power consumption of 50 mA, is fully calibrated, and has integrated sensor fusion to increase the accuracy of measurements. The static accuracy of the roll and pitch is <0.5° and <1.0° in yaw. The resolution of the roll, pitch, and yaw measurements is 0.01°. The sensor is immune to magnetic disturbances [22]. The positional drift of the gyroscope are <0.09°/h for roll and pitch and 1.05 °/h for yaw. The sensor can be connected through a UART serial interface or a CANBus interface. The sensor can be initialized and configured using the software provided. It has an output rate of maximum 300 Hz. The fluctuations of angular velocity measured by the IMU can be seen in figure 5.10.

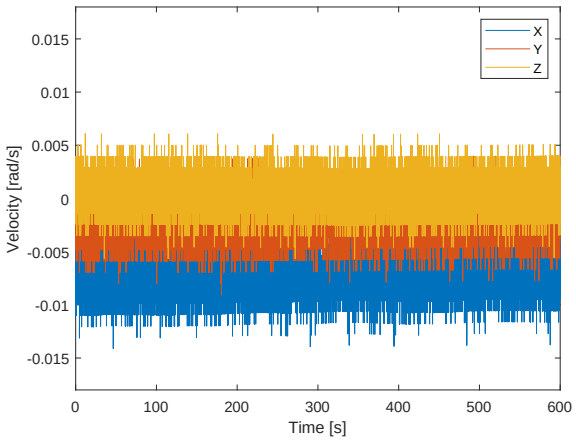


Figure 5.10: Steady state angular velocity data from the IMU

From figure 5.10 it is seen that fluctuations exist in the three velocity measurements. The standard deviations around the x, y, and z axes are 0.0018, 0.0018, and 0.0016, with an offset of -0.0073, -0.0018, and 0.0006, respectively. Since these are below the resolution of the sensor, they are neglected.

5.3.3 Camera

A camera is purchased for the project. The camera used is a Dezen Pixel 1080p webcam. It is capable of capturing frames at a resolution of 1920x1080 pixels at a rate of 60 Hz. The

camera is chosen for its high frame rate, high resolution, and low price. The camera will be used to capture a top or side view of the Crawler.

5.3.4 Summary of sensors

A compilation of the characteristics of the sensors used in the sensor package can be found in table 5.4.

Table 5.4: The sensor package showing measurable states together with statistic features provided by the manufacturer

Sensor	Measurable states	Manufacturer accuracy	Drift	Resolution	Sample rate
Piston force sensor	F_P	-	-	80 N / 0.02mv/V	400 Hz
Piston position sensor	X_p	-	-	0.01 m	700 Hz
Inc, encoder	ϕ_i	-	-	0.18°	6.25 MHz
Torque sensor	τ_w	-	-	1 mNm	6.25 MHz
IMU	roll,pitch,yaw	{<0.5°,<0.5°,<1.0°}	{0.09°/h,0.09°/h,1.05°/h}	0.01°	300 Hz
Camera	-	-	-	1080x720,1920x1080p	60,30 Hz

5.4 Robot Onboard Control System

In order for the Crawler to be able to operate the motors, use its sensors, and control system, they need to communicate with each other. For this project, the micro controller chosen to handle the inputs and outputs is a Nvidia Jetson TX2 with a Spacely Carrier board. This is a controller which SubC Partner has used in the past for other robotic projects. One great advantage of the Jetson is that it has high performance graphics card with 256 CUDA cores and an ARM Quad core processor. It also has 8GB LDDR4 RAM and supports a wide range of connection standards such as CAN, UART, and WIFI. The controller for this project should be able to handle multiple sensory inputs ranging from analog to graphical.

The operating system used for this robot is the Robotic Operating System (ROS1), Melodic version, based on Ubuntu 18.04. ROS is a system designed to ease the communication for a robot between its sensors and actuators. It does this by running scripts for each of the components that incorporates protocols on how it should be operated and what settings its initialized with, where its attach to what type of communication, Serial, I2C, CAN and so on. Since a robot can get rather large with the amount of components a launch file can be made to run all these scripts at once. All these scripts then designate the component it is using as a node allowing communication between nodes using topics or services. All the messages between nodes in this project are made as a topic. Only the initialization of the motors to arm them is a service for safety reasons. These topics can then be published to or subscribed from nodes. Subscribe means it receives data from a node and publish means it delivers data to a node. This modular structure enables the system to be expanded upon with more sensors and actuators, giving more flexibility in its design. It being on a Linux system also enables more freedom handling the errors with more information available. An example of this when the CAN bus is down, being

able to see how much data has been transmitted, received and dropped and the current state it is in directly from the terminal. Scripts can be written in either C,Python or MATLAB where one of them will be the control algorithm. This allows the Jetson to control the Crawler and operate it according to what is desired. The control script does this by subscribing to the sensors topic and publishing to the motor topic, with desired torque according to the control algorithm. A great advantage of using ROS is that it already has developed packages for motor drives, sensors, and communication protocols. These packages include:

- syddynamics_tm352
- maxon_motor_ros
- SocketCAN
- ros_controllers
- rosserial_python

5.4.1 Communication interfaces

Several communication interfaces will be used in this project. To control the Crawler, the Jetson is connected via an Ethernet cable to the top-side computer using the Secure Shell (SSH) communication protocol. For communication between Jetson and the various sensors, the Controller Area Network (CAN) bus is used. This is a message-based protocol that is widely used in the automotive industry because of its robustness to electromagnetic interference. This is ideal for operating on members that emit such fields [23]. Furthermore, it is a modular protocol and it is easy to extend the bus with extra sensors. Each sensor or controller in the bus network is designated a unique ID that can then be interfaced by any other controller connected to the same network. The CAN-BUS is set to a bandwidth of 1Mbs and accessed using the package canopen.
The Universal Serial Bus (USB) interface is also used to connect an Arduino Due to the Jetson. The Arduino Due has installed the same ros libary on it as the Jetson, making it able to subscribe and puplish to nodes, which can be interfaced from the Jetson. The Arduino Due, is responsible for the hydraulic piston publishing the current pressure it has and subscribing to the topic describing the desired piston pressure.

5.4.2 Component controllers

For the local components, dedicated controllers are used.

Motor controller
Three EPOS4 controllers are used to drive the motors. The EPOS can receive commands through the CAN bus and send sensory feedback such as torque, current, position, and velocity. Using the Maxon motor, motor-specific parameters are determined from the data sheets and are used within the EPOS studio software to generate a PDO file which

contains all the settings of the motor, which is used by ROS to generate a run script for the motors.
The torque controller works through the EPOS studio, but it is not tested to work over ROS. Only the velocity and position controllers are verified to work. With some tinkering in the code, it is possible to make the torque controller work as well, enabling more freedom when it comes to control design.

5.4.3 Communication overview

A schematic showing how the different components connect is illustrated in figure 5.11. The power supplies and barrel jack are powered by 230 V.

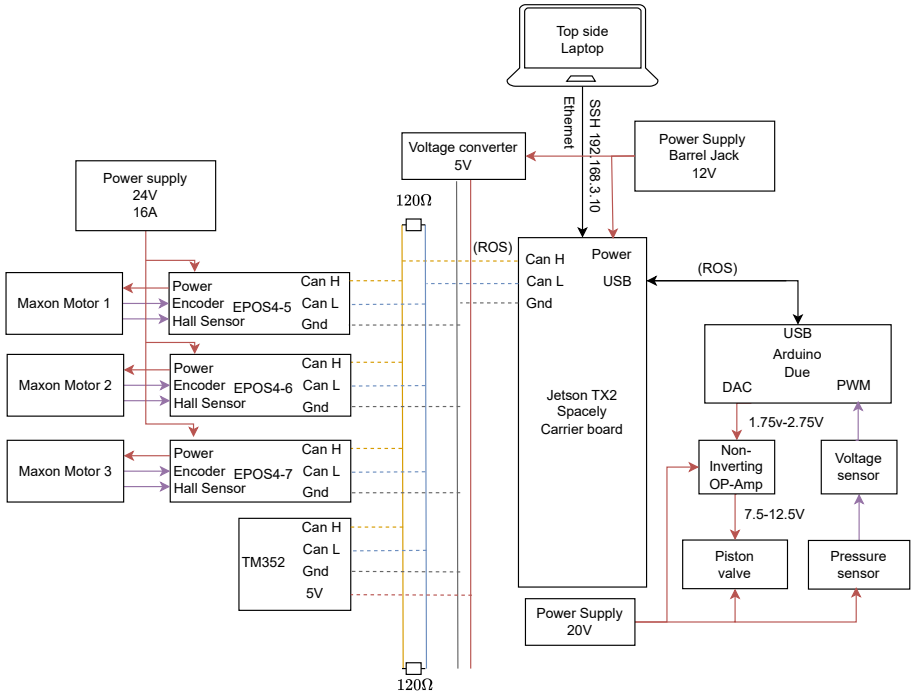


Figure 5.11: Communication diagram for the crawler setup.

5.5 Crawler redesign comparison

Extensive work has been put into updating the original 3D model of the single Crawler arm. The model has been updated with all new components, including electric actuators, motors, and the three new drive trains and wheels. The component models used in the 3D model originate from the various component retailers used while upgrading the project.

A side view display showing the Crawler arm before and after upgrading with new components can be seen in figure 5.12.

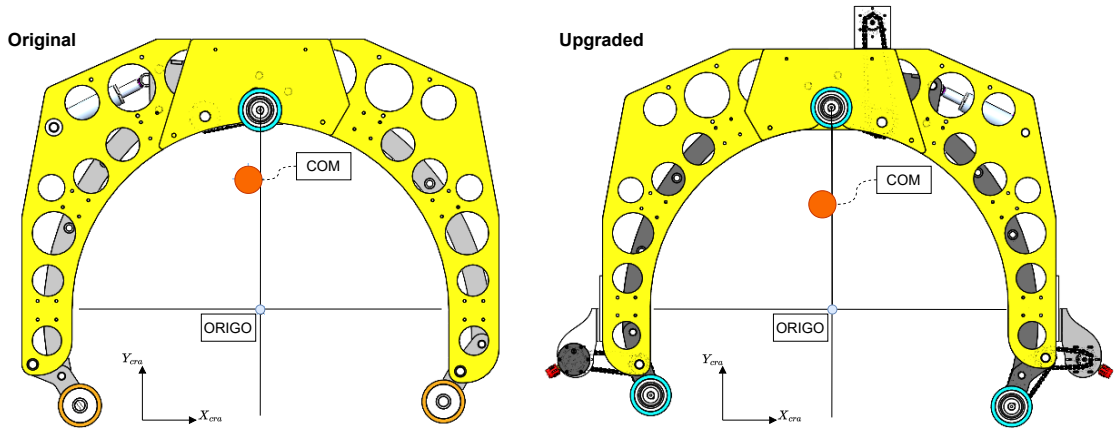


Figure 5.12: Before and after figures of the Crawler arm, including coordinate axis and updated COM.

Due to the change in components, some characteristics of the Crawler arm have changed. A comparison can be seen in table 5.5.

Table 5.5: Specification of the crawler, before and after the upgrade, compared to the requirements set in section 2.3. The lateral velocity is found using a radius of 0.52 m.

Characteristics	Original	Updated	Required
Total mass [kg]	92	132	-
Total inertia [kg/m^2]	27.740	47.062	-
COM location from origo {x,y} [m]	{-0.035;0.380}	{-0.035;0.268}	-
Angular max velocity [rad/s]	0.114	1.271	0.010
Lateral max velocity [m/s]	0.059	0.661	0.005
Max cont. rotational torque [Nm]	145.0	126.0	102.0
Total clamp force [kN]	15.00	15.00	7.85
Power consumption [W]	3000	612	-
No. of sensors	2	6	yes

As seen in the table, the weight of the crawler increases. Taking into account the weight of external components of the hydraulic system, such as pipes, motor, and tank, it is assumed that the increase in weight still results in an overall reduction. The location of the center of mass of the robot moves 0.112 m downward in the y direction, compared to the increase in mass, also increasing the total inertia of the robot. The robot speed increases from 0.114 to 1.271 rad/s, exceeding the requirement limit. The clamp force is kept the same since it was found that the amount of force that the electric actuator could deliver did not exceed the minimum requirements of 7.85 kN. The robot rotational torque is reduced from 145 Nm to 126 Nm, still above the limit of 102.0 Nm. With three wheels that deliver torque, the robot is also less affected by local pipe sections with very low friction due to biofouling and marine growth. The total power consumption of the

robot reduces from 3 kW to 0.6 kW. Compared to the original Crawler model, where the location could only be identified by a series of cameras or visually by the pilot or assistant, the upgraded version now has several sensors that can assist in determining the position. The new sensors are able to measure and store the angular position and speed of the robot and each separate wheel.

6 Position Estimation

The gyroscope in the IMU and the rotary encoders in the motors can be used to find the position and velocity of the crawler body and wheels. The rotary encoders measure the position of each wheel directly, and this position could, in theory, be converted into a crawler position. However, this position could be biased due to the difference within wheel speed caused by blockage or slip, thus the position could drift away from the real value. The IMU uses the gyroscope to measure the angular position and find the angular velocity from this. Since this project will rely on the measurements to develop feedback control, it is important that they are accurate. This chapter will focus on comparing the sensor measurements against a reliable basis.

6.1 Basis position measurements

To define a reliable basis, this report uses a solution based on the Computer Vision Toolbox (CVT) developed for use with MATLAB. The solution uses the raw footage from a camera, which is then processed inside the MATLAB CVT.
Using a camera, it should be possible to gather information about the following:

Crawler frame:

- Global angular position and velocity wrt. center of test member
- Distance from frame to center of test member

Wheels:

- Global angular position and velocity wrt. center of test member
- Local angular position and velocity wrt. center of wheel
- Distance from center of wheel to center of test member

Test member:

- By comparing the angular velocity between wheels and the Crawler, the algorithm can detect slip
- By evaluating the distance of the wheels to the center of the test member, it can also be detected if the test member is not uniformly round or if any obstacles are present

The camera captures video at a resolution of 1920x1080 pixels and has a frame rate of 30 Hz. As seen in figure 6.1, the camera is mounted in the center of the test member at a height of 2.00 m above the member. The camera is connected to a laptop running the CVT that detects the point DP on the Crawler.

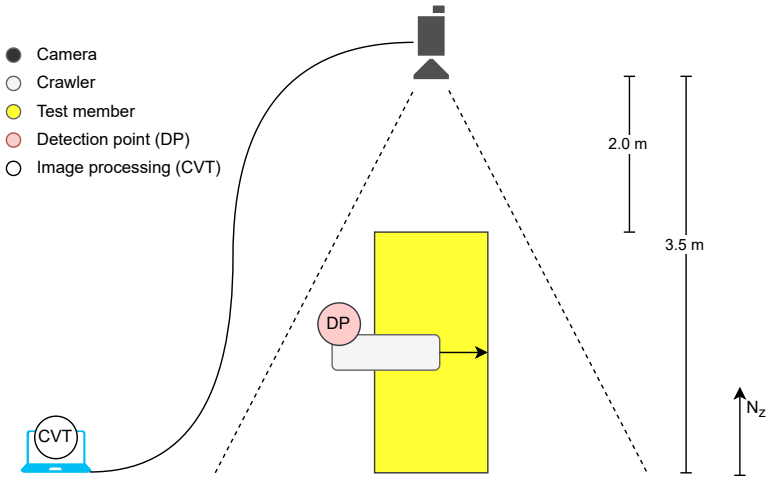


Figure 6.1: Camera setup used to measure a comparable basis for the sensors attached to the Crawler

6.1.1 Camera calibration and image processing

Before the camera can be used for position measurements, a calibration is performed, this process can be seen in C.1.1. The calibration is used to correct from lens distortion of the camera, by having known points in the real world, it can also be used to convert pixel coordinates into world coordinates [24]. After calibration, the video is recorded and all frames are post processed in order to obtain positions and estimate velocities. This process is explained in C.1.2. The workflow for this can be seen in figure 6.2.

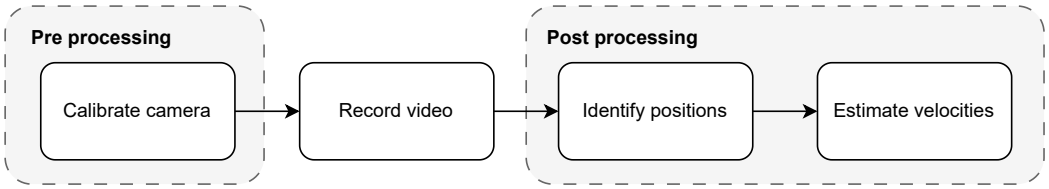


Figure 6.2: Workflow of position and velocity estimation using the camera

6.1.2 Validation of position measurements

A laboratory setup is designed to test the accuracy of the algorithm. The setup is designed to capture video footage of a rotating circle with an image containing eight circles of varying sizes.

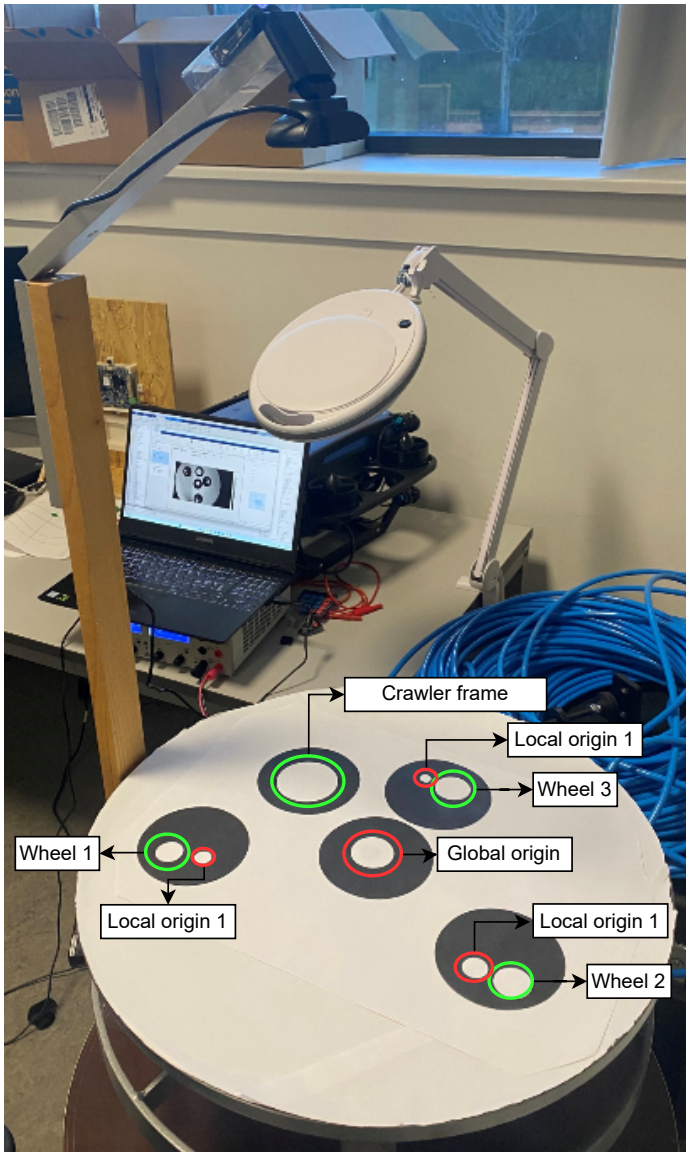


Figure 6.3: Laboratory setup used for validating the algorithm used for the basis tests

As seen in figure 6.3, a rotary disc, powered by a 12V DC motor is installed and the camera is mounted above the setup. The detection image is attached to the disc. The DC motor is powered with voltage ratings from 1 to 11 V. A total of six tests are performed with voltage increments of 2 V, where the speed of the motor is dependent on the voltage input. Camera data is captured for 5 total rotations in each test.

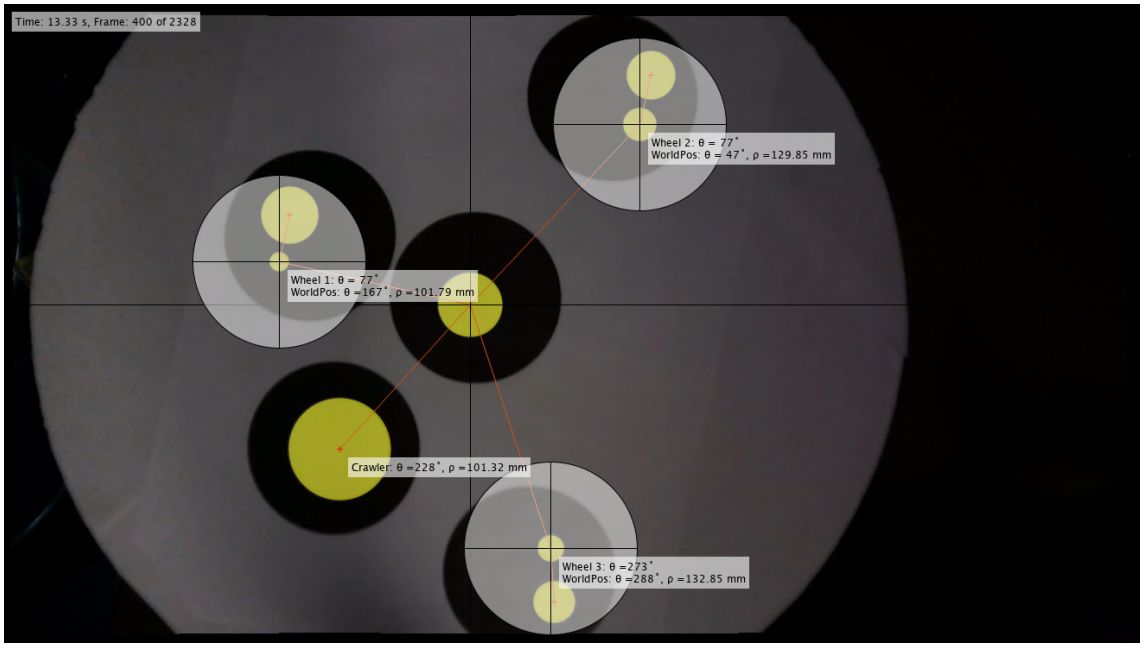


Figure 6.4: Snapshot from the video output of the algorithm processing a frame from the captured video footage, yellow circles are tracked circles, the local origins of the wheels are depicted using a white circle

As seen in figure 6.4, the algorithm is capable of detecting and distinguishing between the eight desired circles. These are marked as yellow circles in the figure. The local origins are marked with white circles. It was found that when circles were to be tracked between frames, the algorithm performed better with circles of a varying diameter.
The eight circles were indexed by descending size and diameter, as well as distance to global origin where captured in world coordinates. Table 6.1 shows the estimates compared to the true values.

Table 6.1: Comparison of diameters and distances from wheels and Crawler, to the global origin. All are compared to the true world values

Circle	Component	Est. diameter ($\bar{d}$) [mm]	True diameter (d) [mm]	Est. radius ($\bar{\rho}$) [mm]	True radius (ρ) [mm]
1	Crawler frame	52.91	55	101.79	110
2	Global origin	33.52	39	-	-
3	Wheel 1	29.49	30	-	-
4	Wheel 2	25.45	26	-	-
5	Wheel 3	21.68	23	-	-
6	Local origin 1	9.97	10	101.79	110
7	Local origin 2	13.56	15	129.85	140
8	Local origin 3	17.26	19	132.85	144

From the table it can be seen that the estimated diameters in average varies with 2 mm from the true diameter. The estimated radius on average varies by 9 mm. This can be translated to an average deviation of 5.5%. This is deemed acceptable for this study since the main focus is not on estimating the true size or the distance of the circles.

To identify the accuracy of the collected location data through camera recordings, the variance of radius and velocities is calculated. Furthermore, the means of the radius measurements and velocities are compared to the true counterparts. The results of the six tests performed are compared in table 6.2.

Table 6.2: Comparison of the six performed velocity tests. All tests are performed reaching five full rotations. $\theta_{C,ma}$ is calculated using a moving average filter, averaging each 10 samples.

Test	True vel. ($\dot{\theta}_C$)	Avg. vel. ($\bar{\dot{\theta}}_{C,avg}$) [deg/s]	std($\dot{\theta}_C$) [deg/s]	std($\theta_{C,ma}$)	Est. avg. radius (ρ_C) [mm]	trend([mm]	std(ρ_C) [mm]
1	2.190	2.913	1.446	0.236	101.120	0.437	0.203
2	6.737	6.770	1.488	0.248	101.240	0.052	0.256
3	11.285	11.182	1.572	0.302	101.260	0.001	0.262
4	15.607	15.599	1.662	0.381	101.270	0.000	0.264
5	20.037	20.121	1.639	0.415	101.270	0.069	0.262
6	24.658	24.680	1.715	0.467	101.280	0.028	0.257

As seen from the table, the average velocity varies for each test. Some fluctuations are present in the estimated angular velocities. However, by applying a moving average filter to average a window of 10 samples, the standard deviation is significantly reduced. Before and after applying the filter, the standard deviation in all test varies by 0.269 and 0.231 deg/s, respectively. The average calculated radius varies by 0.05 mm, which gives a total mean radius of 101.195 mm. Compared to the measured radius of 110 mm, this is within a margin that can be accepted as a basis for comparison.

A significance test will be performed to test whether the collected data are within a significance level of 99.7%. This means that any radius measured more than 3σ points from the statistical μ is considered an outlier. In this case, it is expected that 0.3% of the data points in each of the tests could be outliers. All tests are compared, and the distribution of the data can be visually seen as box plots seen in figure 6.5.

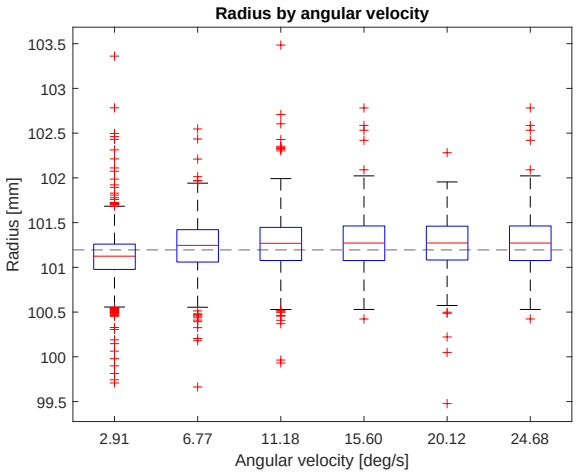


Figure 6.5: Boxplots showing the distribution of the calculated radius data by the six angular velocities. The dashed line represents the mean of all radius data

From this figure it can be seen that the distribution of all six tests follows a normal dis-

tribution, with the median in the middle of the upper and lower quartiles, and it can be seen that each test has outliers. The number of outliers decreases as the angular velocity increases. Since it has been decided that each test should run for five total rotations, this is expected. The test with the slowest velocity collects more samples than the fastest. A comparison of the statistical features in the six tests can be seen in table 6.3, where the duration, number of samples, expected outliers and measured outliers are shown.

Table 6.3: Statistical features of the six tests compared in different velocities

Test	Full rotations	Duration [s]	No of samples	No of outliers	Expected outliers
1	5	563	16880	58	50
2	5	214	6412	18	19
3	5	165	4941	18	15
4	5	116	3573	6	10
5	5	89	2684	6	8
6	5	78	2328	4	7

All tests except the third has a lower number of outliers than what is expected, however, the third is still within what can be seen as reasonable.
As a conclusion to this section based on observations during tests and the statistical features of the collected data, it can be concluded that the calibrated camera setup can be trusted with a confidence of 97.5% as a basis for comparing sensor measurements.

6.2 Comparison on physical setup

With the camera measurements calibrated and validated, the camera can now be used to validate the position and velocity measurements of the sensors on the crawler. To validate this, the crawler is set to rotate in one direction with a constant torque input. The crawler is rotating on a vertically oriented member and is attached with 50 bar piston pressure. During the rotation, the camera is mounted 2 meters above the top of the member. The setup is seen in 6.7.

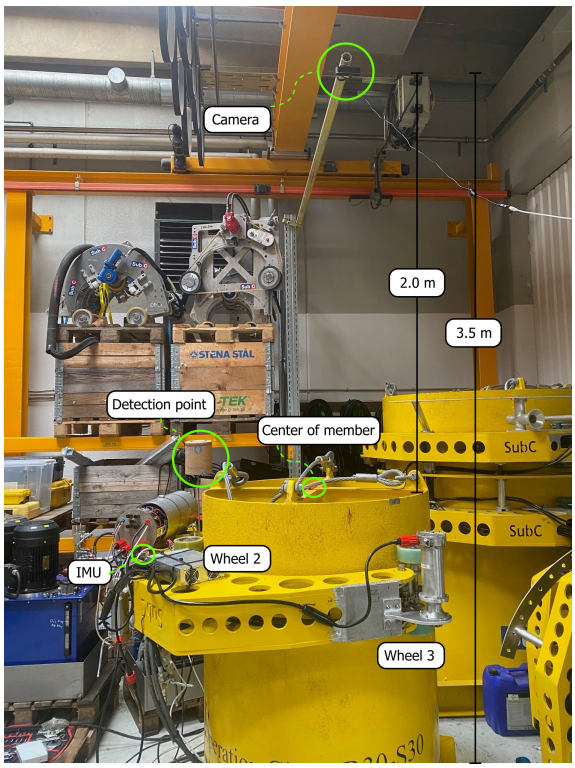


Figure 6.6: Physical setup with the camera mounted 2 m above the test member

The algorithm described in section C.1.2 is altered to handle the physical setup instead of the laboratory setup previously used. It was found that during the conditions available in the physical setup it was not possible to detect wheel rotation, as the test member shadowed around 60% of the wheel surface. Hence, only crawler position and velocities are tracked by the updated algorithm. Before testing, the crawler is fitted with a detection point, this point is aligned with the IMU. Before testing, the camera setup is calibrated. A photo of the footage captured, together with the tracked detection point on the crawler can be seen in figure 6.7.

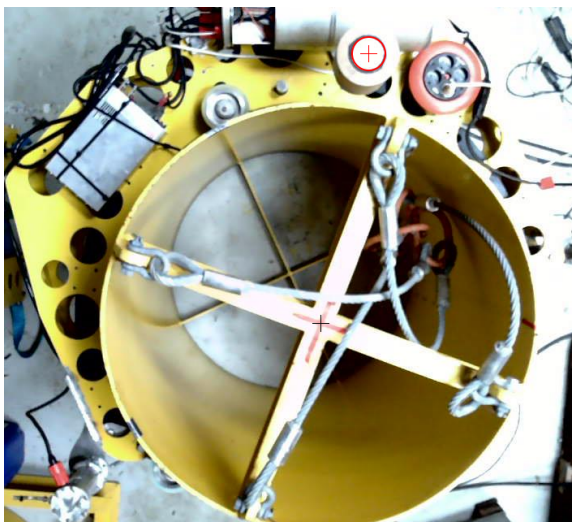


Figure 6.7: Snapshot from the video used to track the position of the crawler

Here, it can be seen that the center of the member is marked with a "+" sign, and in the top center of the figure the circle marker is tracked and marked. The position of the tracked circle in each frame is stored and compared with the measured data from the IMU.

Four tests are performed with varying input torques for each motor at 17, 25, 34, and 42 Nm. The inputs are chosen since they represent a range within the crawler is expected to operate at. The results of the third test can be seen in figure 6.8.

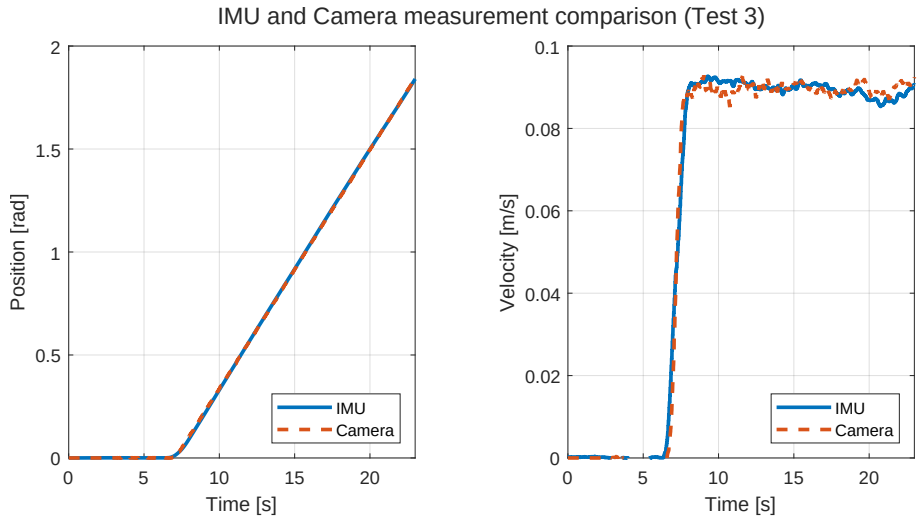


Figure 6.8: Comparison between IMU and camera position and velocity measurements.

In the figure it can be seen that the two position measurements are close to equal, they start at the same time, and after 23 seconds the measurements have arrived at 1.92 and 1.93 radians for the camera and IMU measurements, respectively. For the velocity, the transient values are close to equal for the two measurements, however, some fluctuations are seen in both of the measurements. Both reach steady state at 8.45 s. The mean values of this steady-state longitudinal velocity measured by the camera and the IMU are 0.0833 and 0.0886 m/s, respectively. The standard deviations of the velocity measurements are 0.0074 and 0.0062 for the camera and the IMU, respectively. The results of tests 1 and 4 are consistent with the findings of the third test. The findings of the second test differ from those of the other three. The robustness of the tracking algorithm is found to highly depend on the ambient light intensity and not able to handle an increase this. The increase in light intensity is caused by outdoor light passing through a skylight in the roof. The results of the three other tests can be seen in Appendices A.10, A.11, and A.12. The end position deviation for each tests is found to be 0.015, 0.013, 0.010 and 0.001 respectively. A table with results from all four tests is found in 6.4.

Table 6.4: Comparison of measurements gathered through camera and IMU at four different torque inputs

Test	Duration [s]	Torque input	End position deviation [rad]	IMU mean ss. vel. [m/s]	Camera mean vel. [m/s]
1	23	17	0.015	0.0261±0.0095	0.0263±0.0084
2	23	25	0.013	0.0679±0.0077	0.0662±0.0123
3	23	34	0.010	0.0866±0.0062	0.0886±0.0074
4	22	42	0.001	0.0957±0.0016	0.0947±0.0051

Three of four tests confirmed that the position and velocity measurements of the IMU can be used to provide a reliable reading that can be used to validate the model and as input to any developed control. The ambient light around the physical setup was found to have a large impact on how well the camera was able to track the circle marker. Either a better algorithm could be developed for tracking, or the tracking circle could be produced using either another color or more reflective material to stand out against the remaining objects. As seen in the second test, when the ambient light is too high, it is more difficult for the camera to track the circle, resulting in more fluctuating position and velocity measurement.

7 Modeling

The Crawler model is divided into two parts. The clamp model, which is based on the static calculations seen in 4.3, and a rotational model that describes the rotation of the crawler around a member.

7.1 Clamp model

The model needs to be generalized to work in both horizontal and vertical operating types. Therefore, the static calculations are modified to be applicable for both cases.
For developing the clamp model, the following moment equations specified in section 4.3 are used: 4.9 4.10 4.11.

$$\overset{\curvearrowleft}{+}\sum M_C = L_{2C}N_2 + L_{RC}F_R - L_{PC}F_P - L_{LC}F_L = 0 \tag{7.1}$$

$$\overset{\curvearrowleft}{+}\sum M_A = -L_{1A}N_1 - L_{LA}F_L = 0 \tag{7.2}$$

$$\overset{\curvearrowleft}{+}\sum M_B = L_{3B}N_3 - L_{RB}F_R = 0 \tag{7.3}$$

To describe the relationship between the piston and the moments on each arm containing the wheels. These same equations apply to the dynamic model, because there should not be any moment in the parts when the robot is operating since it should always be clamped in. This means that it will still be viewed as static in this regard, which simplifies the need for angles that describe the moment arms as well. This time, the normal forces of the wheels are not assumed to be equal. This is because as the crawler rotates, the normal force on each wheel differs depending on its position relative to gravity. The crawler cannot move through the pipe, so the sum of forces is still zero in both directions. The force to which gravity contributes also depends on the angle at which the crawler is operating. From figure 4.14 the following equations can be established.

$$\sum F_y = N_1\sin(\theta_c + \beta) + N_2\sin(\theta_c) + N_3\sin(\theta_c - \beta) - mg\cos(\alpha) = 0 \tag{7.4}$$

$$\sum F_x = N_1\cos(\theta_c + \beta) + N_2\cos(\theta_c) + N_3\cos(\theta_c - \beta) = 0 \tag{7.5}$$

By combining equation 7.1 with 7.5, it is possible to find the amount of normal force on each wheel at any angle around the member. When the member is oriented vertically, the three normal forces are equal and only depend on the amount of force delivered from the piston. When the member is oriented horizontally, both the piston force and the position

of the crawler dictate the size of the normal force on each wheel, this can be seen in figure 7.1.

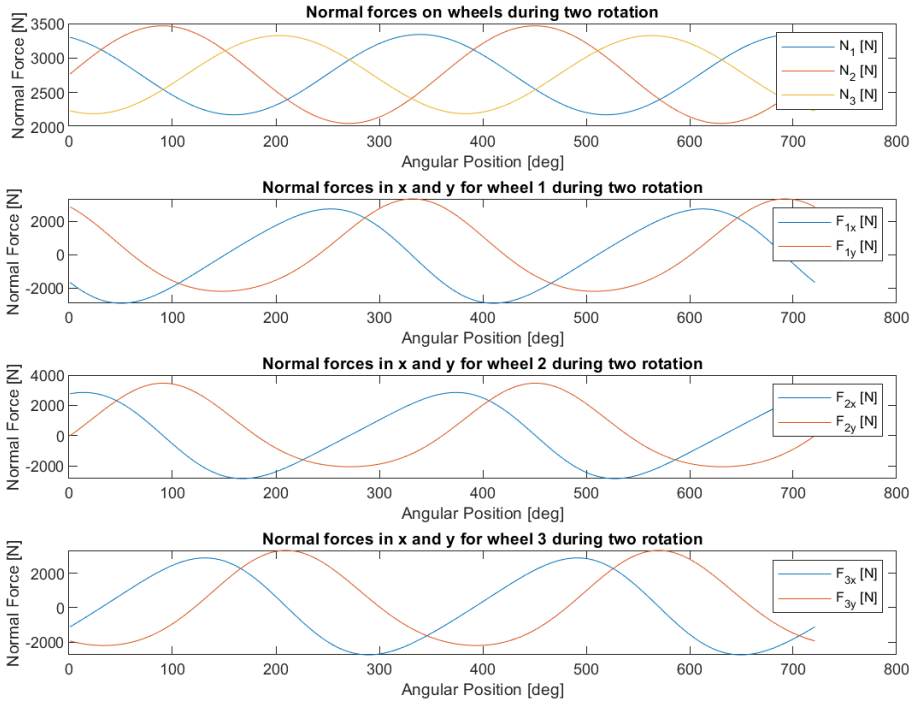


Figure 7.1: Normal force distribution, when the member is in a horizontal position

7.2 Rotational model

The non-linear rotational model is developed based on [25], which is used to describe a lateral moving car. For use in this project, the lateral moving parts of the original model are readopted to represent angular motion. The model is developed using an equation for each wheel and an equation for the crawler body. It is assumed that the crawler consists of a rigid body frame and that the wheels always have contact with the member about which it is rotating. The resulting equations of motion are described from 7.6 to 7.11.

$$J_c\ddot{\theta}_c = L_2\left(F_{t,1}(\lambda_1, N_1) + F_{t,2}(\lambda_2, N_2) + F_{t,3}(\lambda_3, N_3) - F_{roll}(\dot{\theta}_c)\right) - L_1F_g(\theta_c)\cos(\alpha) \quad (7.6)$$

$$J_w\dot{\omega}_1 = T_{m,1}G_1G_2 - R_w\left(F_{t,1}(\lambda_1, N_1)\right) - b_1\omega_1 \quad (7.7)$$

$$J_w\dot{\omega}_2 = T_{m,2}G_1G_2 - R_w\left(F_{t,2}(\lambda_2, N_2)\right) - b_2\omega_2 \quad (7.8)$$

$$J_w\dot{\omega}_3 = T_{m,3}G_1G_2 - R_w\left(F_{t,3}(\lambda_3, N_3)\right) - b_3\omega_3 \quad (7.9)$$

$$F_{roll} = \mu_r(N_1 + N_2 + N_3) \quad (7.10)$$

$$F_g = mg\cos(\theta_c + \gamma) \tag{7.11}$$

Where θ_c is the angular position of the crawler, ω_1, ω_2, and ω_3 are the angular velocity of the wheels, $T_{m,1}$, $T_{m,2}$, and $T_{m,3}$ are the input torque, λ_1, λ_2, and λ_3 are the slip ratio at each wheel, $F_{t,1}$, $F_{t,2}$, and $F_{t,3}$ are the longitudinal contact forces between the wheel and the member, F_{roll} is the rolling force and F_g is the gravitational pull on the crawler, as mentioned in 4 α is the orientation of the member with respect to the x axis. The forces present in the crawler model are found in figure 7.2, whereas those related to the wheel models can be seen in figure 7.3.

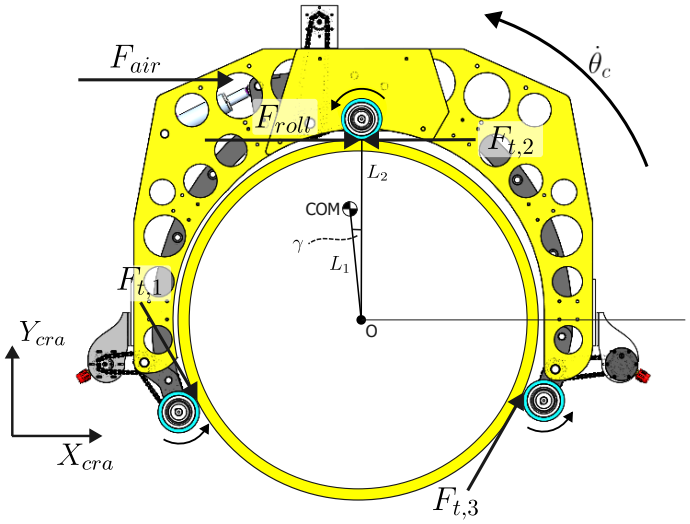


Figure 7.2: Force diagram related to crawler body equations

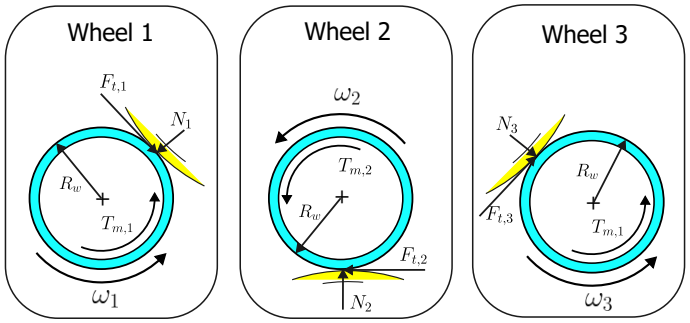


Figure 7.3: Force diagram related to crawler wheel equations

The parameters of the crawler μ_a and μ_r are the coefficients of friction of the air drag and the rolling force, m is the mass of the crawler, g is the gravitational acceleration, R_w are the radius of the wheel, L_2 is the distance from origo to the rim of the member, L_1 are the distance from origo to the center of mass, J_w and J_c are the inertia of a single wheel and crawler, respectively. γ is the angle between the origo and COM and α is the orientation of the offshore member.

7.2.1 Slip

When the crawler is operating at steady-state velocity, the longitudinal velocity of the body and the wheels are equal in magnitude and direction. During transitions, such as braking or acceleration of the crawler body, a discrepancy between the two velocities can occur, resulting in a slip. This can be caused by a change in surface, reference direction, or change in actuation. The slip can occur between the body and a single wheel or between the body and all three wheels. Based on [26] the longitudinal slip $\lambda_i, i \in \{1, 2, 3\}$ for each wheel is defined as the relative difference between the driven wheel velocity and the crawler velocity seen in equation 7.12. These are defined on the basis of braking or acceleration states. At steady-state velocity $\lambda_{123} = 0$.

$$\lambda_i = \begin{cases} \frac{\omega_i R_w - \dot{\theta}_c L_2}{\omega_i R_w}, & \omega_i R_w > \dot{\theta}_c L_2, \quad \omega_i \neq 0, \quad \text{acceleration} \\ \frac{\omega_i R_w - \dot{\theta}_c L_2}{\dot{\theta}_c L_2}, & \omega_i R_w < \dot{\theta}_c L_2, \quad \dot{\theta}_c \neq 0, \quad \text{braking} \end{cases} \tag{7.12}$$

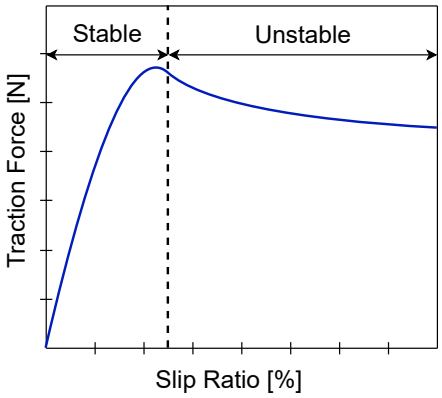
$$i \in \{1, 2, 3\}$$

7.2.2 Traction force

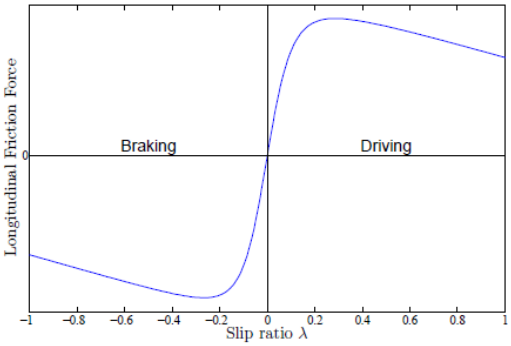
Whenever $\lambda_i \neq 0$ a traction force $F_{t,i}$ occurs at the contact point between the member and the wheel as seen in figure 7.3. The traction force at each wheel can be described as a non-linear function of slip, normal force, and adhesive coefficient, as proposed in [25]. The force can be determined using various models. For a car moving lateral, the magic tire equation is often a preferred choice when using slip as input [27]. The equation is shown in equation 7.13, and it is also assumed that the slip angle is zero since the crawler only moves in one degree of freedom with the wheels being fixed.

$$F_{t,i} = N_i \mu_{D_i} \sin\left(\mu_{C_i} \tan^{-1}\left(\mu_{B_i}\lambda - \mu_{E_i}\left(\mu_{B_i}\lambda - \tan^{-1}\left(\mu_{B_i}\lambda\right)\right)\right)\right) \tag{7.13}$$

Where N_i is the normal force for the corresponding wheel i, μ_{B_i}, μ_{C_i}, μ_{D_i}, and μ_{E_i} are the four magic tire parameters for the corresponding wheel. The parameters will be determined on the basis of a curve fit on the experimental data. In equation 7.13, λ is the slip ratio [28]. The magic tire formula empirically represents and interpolates the measured tire force and moment curves [27].
For further explanation, a graphical illustration of the slip curve as shown in figure7.4a and 7.4b can be sketched.



(a) Illustration of a slip curve showing traction coefficient vs. slip ratio.



(b) Illustration of the slip curve switching and showing the traction coefficients vs. slip ratio for breaking and [29].

Figure 7.4: Two types of slip curves used to show the relationship between traction and slip.

The slip curve, which shows the traction force as a function of the slip ratio, describes the traction force of the wheel at which it begins to slip. Figure 7.4a indicates a stable and an unstable region. The slip curve is widely used for anti-lock braking systems (ABS), which contribute to vehicle driving stability, making it important to know the optimal slip region for specific vehicles wheels [30]. The slip curve increases in the stable region, which means that the motor torque can be increased to a certain value, so the vehicle can stabilize for the different slip ratios. After the peak of the curve, the increase in motor torque will no longer increase the amount of traction available, and the slip curve shows decreasing behavior.
Figure 7.4b shows a typical curve for the longitudinal friction force fitted with the magic tire formula [29]. The graph shows that the slip ratio varies between -1 to 1. If the slip ratio is -1 it corresponds to locked wheels. When the slip ratio is 0, the wheel velocity is equal to the vehicle velocity, and when the slip is 1 the wheel is considered to spin [29]. The graph shows the breaking section and the driving section. This study will determine the slip curves that describe the traction forces as a function of the slip ratio between the crawler and the offshore member for different surface conditions.

8 Parameter Identification

The parameters used in the models developed in chapter 7 are found using various sources and experiments. A detailed explanation of the origin of these parameters is found in this chapter. At the end of the chapter, a table is found that contains all the parameters. To compare results and validate the accuracy of approximations in this chapter, the R-squared coefficient of determination are used. The R-squared value indicates the proportional amount of variation in the response variable which can be explained by the independent variables in the regression model. Hence a larger R-squared value, the more variability is explained by the regression model. R-squared can be found by the following equation.

$$R^2 = 1 - \frac{SSE}{SST} \tag{8.1}$$

Here SSE is the sum of squared error and SST is the sum of squared total.

8.1 Internal motor control

The motor torque output is controlled by an internal current controller in the EPOS system software. Some motor parameters are described in 5.1 available from the manufacturers data sheet. However, dynamics such as the internal viscous friction in the motor and throughout the gears as well as controller response are not.
Internal control is performed using a proportional integral (PI) current controller with the following structure [31].

$$C(S) = k_p + \frac{k_i}{s} \tag{8.2}$$

This controller is tuned using automated system identification within the EPOS studio software. Since the procedure behind system identification is unknown, each motor is identified at 3.5 A, since this was the maximum allowable torque set in section 5.1. Each controller is tuned 10 times to identify the standard deviation of the coefficients k_p and k_i. The maximum motor current is chosen since that is the region where the motor is going to operate near when rotating around the vertical members. A higher current is not desirable, as it will exceed the maximum continuous torque the gear can handle. These coefficients are shown in table 8.1.

Table 8.1: Controller parameters for each of the three motors

Coefficient	Motor 1	Motor 2	Motor 3
Proportional gain (k_p)	913.30 $\pm$ 22.99	994.40 $\pm$ 9.30	951.00 $\pm$ 8.5505
Integral gain (k_i)	3421.50 $\pm$ 136.37	4206.20 $\pm$ 62.70	3868.40 $\pm$ 75.263
Identification point	3500 mA	3500 mA	3500 mA

The motor loop block diagram can be seen in 8.1.

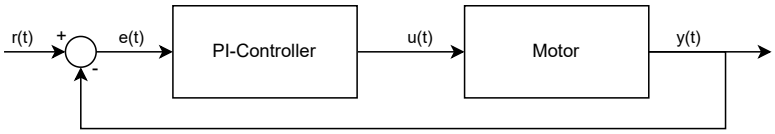


Figure 8.1: Block diagram of the current control loop

Here, the first block from the left is the controller block and the second contains all motor dynamics. To identify motor dynamics, each motor is subjected to square-wave input data and the output is recorded. These input and output data will be used to determine a transfer function that can capture most of the dynamics. By neglecting the transient impact from the inductor, the dynamics of a general EC motor can be described using a first-order transfer function [32], and by applying Mason's rule to the system described in 8.1, a second-order transfer function is fitted to the input output data. The transfer function is validated against a second data set from which it shows an R^2 value of 0.91, 0.96, and 0.87 for motors 1, 2, and 3, respectively. The input and output data, as well as the fitted transfer functions, can be seen in figure 8.2 for each of the three motors.

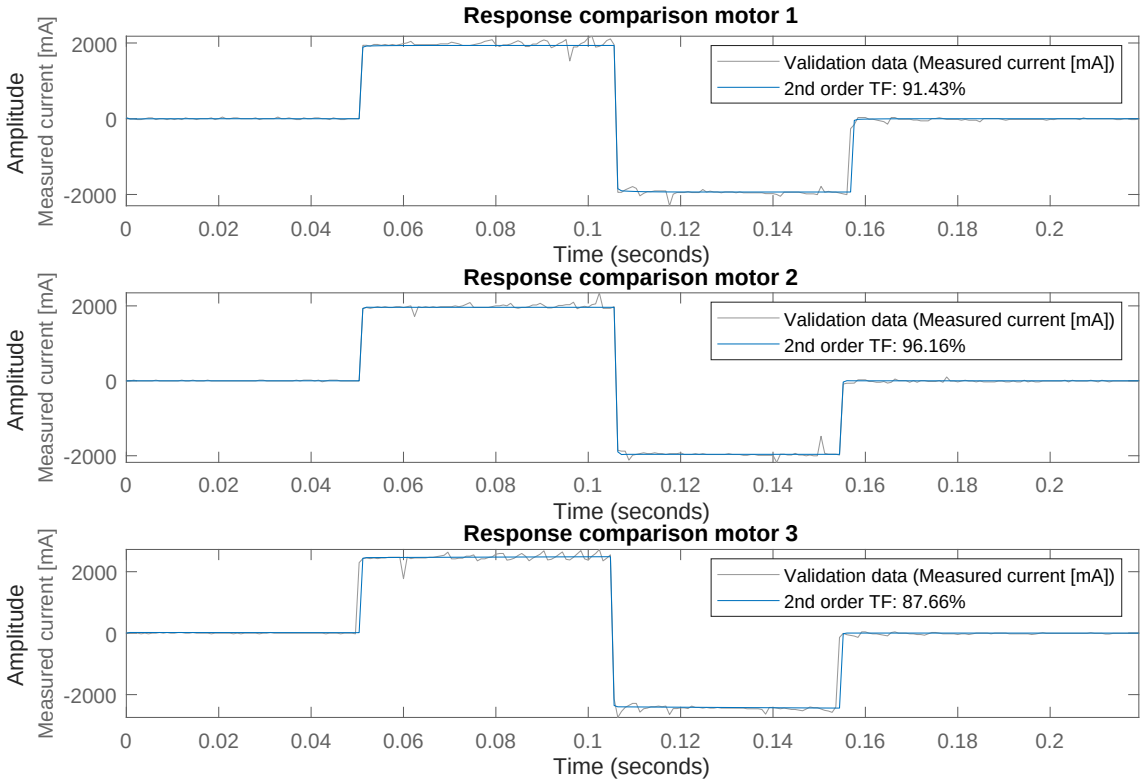


Figure 8.2: Output response to a 2000 mA square-wave input

Based on the results in figure 8.2, it can be seen that although the sampling frequency of the measured data is 1.25 kHz, the electric dynamics of the motors are too fast to be identified. Based on these results, the internal current controller can be seen as a static gain, and thus these dynamics will be neglected in the model. The DC gains of the transfer functions can be seen in table 8.2.

Table 8.2: DC gains of the three internal control structures of the EPOS 4

Parameter	Value	Standard deviation
C_1	0.9677	0.08
C_2	0.9821	0.04
C_3	1.0303	0.12

8.2 Drive train friction coefficients

To identify the viscous friction parameter b_i from the wheel models seen in 7.7, 7.8, and 7.9, tests are performed. The crawler is released from the member so that each wheel has no surface friction, thus the traction force can be assumed to be zero. Each wheel is subjected to a series of torque input steps ranging from 0 to 34 Nm in increments of 4.2 Nm. When subjecting the motor to torques higher than 34 Nm, while hanging in free air the speeds of the motor exceed the maximum allowable. Resulting in a fail safe turning

the motor off. The angular speed and the actual torque are measured in sequences of 10 seconds per torque step. The time-in-between change in torque is set to ensure that the wheel reaches a full rotation and a steady-state velocity. In this case, the acceleration is also equal to zero, and the equation 7.7 can be rewritten as.

$$T_{m,i} G_1 G_2 = b_i \omega_i \tag{8.3}$$

Then, by linear fit to the measured data, the viscous friction parameter can be identified. All measurements are collected by EPOS4 through ROS. The results of these tests can be seen in figure 8.3.

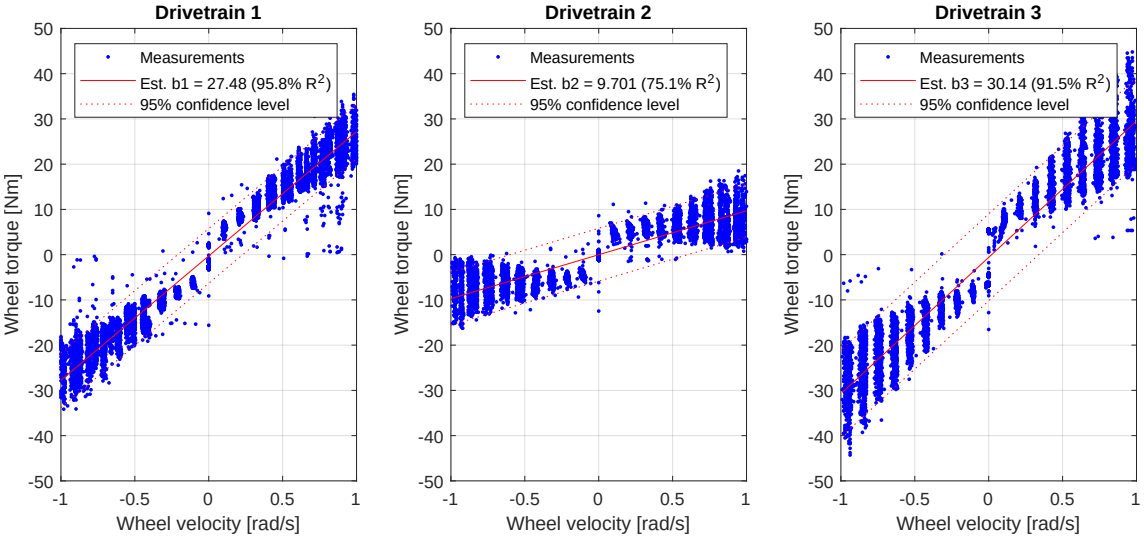


Figure 8.3: Measured torque versus the wheel velocity at steady state, used to identify the viscous friction in the drive trains

Here, b_1, b_2, and b_3 are found to be 27.48±0.06, 9.70±0.05, and 30.14±0.09, respectively. In the figure it can be seen that the fluctuation of the torque measured at the wheels increses as the velocity increases. This was not visible confirmed, however as the wheel speed is increased towards -1 or 1 radians, they approach the motor speed limitations, it was here observed that the closer the wheels got to this software limit, it affected the measured torque as well since the software started to saturate the velocity. As can be seen from the figures, the R^2 value is different between the wheel sets. Drive train 2 has the lowest viscous friction, so it needs less input torque to rotate the wheel. The chain used for drive train 2 was found to be slightly tighter than those used for drive trains 1 and 3, with 3 being the least tight of the three. It can be visually seen that the static frictions of drive trains 1 and 2 are higher than those of drive train 3. It is found that each of the shafts used must be slightly bent, thus at some points within a full rotation of the shaft, the static torque needed to initiate a rotation is higher when the chain is tighter. The impact of static friction is non-linear and can be seen close to where the wheel velocity is zero. The impact is greatest in drive train 2. To try and identify why viscous friction differentiates between drive trains 1,3 and 2, the motor and EPOS used for drive train 2 were switched out, without impacting the results. Thus, the difference is assumed to

occur from a difference in the design of the drive train itself. It is also observed that whenever drive train 2 overcomes static friction, in the drive train, it reaches 0.42 rad/s almost instantly. The drive train parameters are found in 8.3.

Table 8.3: Viscous friction parameters of the three drivetrains

Parameter	Value	Standard deviation
b_1	27.48	0.06
b_2	9.70	0.05
b_3	30.14	0.09

8.3 Crawler friction coefficients

To describe the friction loss caused by rolling friction described in 7.10, the rolling friction coefficient (μ_r) must be identified. To identify this parameter, the crawler is set to operate at various steady state velocities and piston pressures. The change in velocity and piston pressure should not change the coefficients. In order to obtain a constant velocity, a simple PI controller is used, which has the structure as seen in equation 8.2. The k_p and k_i coefficients for the velocity controller are set to be 0.082 and 1257 respectively.

The tests are carried out at three pressures, 51.2, 72.6, and 85.4 bar. These values are chosen, due to the difficulty in changing the pressure. It is very sensitive and was not able to stabilize at a lower pressure than 51.2 bars. The pump pressure is limited at 80bar, but is possible to hit 85 bar before the drain tank takes over. For each of these pressures the crawler is set to operate at five different velocities, these are seen in table 8.4.

Table 8.4: Reference velocities used for determining the rolling friction coefficient of the crawler

Test	1	2	3	4	5
Motor vel. [RPM]	100	500	1000	2000	4000
Wheel vel. (ω_i) [rad/s]	0.0212	0.1059	0.2119	0.4238	0.8476
Crawler vel. ($\dot{\theta}_c$) [rad/s]	0.0026	0.0130	0.0261	0.0521	0.1043

The input torque as well as the actual angular velocity of the crawler data are captured, and the piston pressure is noted for each test. The input torques are summed together before plotting against the angular velocity. For each pressure, the torque and steady-state velocities are combined into one set. The results from these three piston forces are seen in figure 8.4

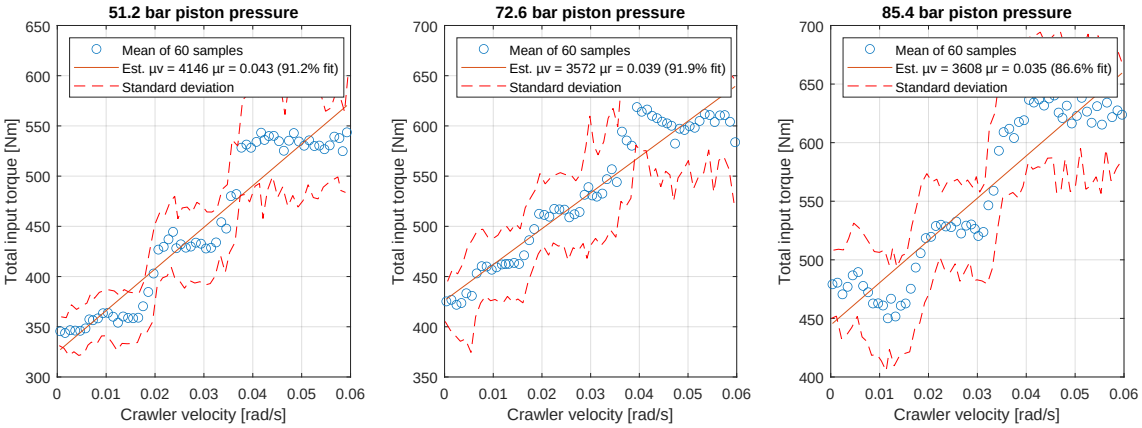


Figure 8.4: Steady state velocity tests performed with varying piston pressure

The total input torque seen in the figures is the total amount of torque of which the combined three wheels apply to the crawler body during rotation.

It is clearly seen from these tests that the coefficients found in 8.3 do not capture all of the viscous friction of the crawler. It is assumed that the actual viscous friction increases when the crawler is clamped to the member. Due to this, the remainder of the viscous friction is combined into a lumbed friction model, thus rewriting equation 4.19 to the following.

$$F_{roll} = \mu_v \dot{\theta}_c L_2 + \mu_r (N_1 + N_2 + N_3) \tag{8.4}$$

Here μ_v is the viscous friction coefficient used for the lumbed friction model. Between the three tests, the rolling friction coefficients are found to be $\mu_v = \{4146.00, 3572.00, 3608.00\}$ and $\mu_r = \{0.043, 0.039, 0.035\}$. As expected, the change in velocity and piston pressure does not affect the coefficients. The average μ_v and μ_r between all tests performed are found to be 3775.3 and 0.0390 with a combined standard deviation of 321.51 and 0.0040 respectively. The adjusted R-square value for each of the tests varies between 86.6 and 91.9 %. This is deemed high enough to be t. The parameters can be found in table 8.5.

Table 8.5: Lumbed friction coefficients of the Crawler

Parameter	Value	Standard deviation
μ_r	0.0390	0.0040
μ_v	3775.3	321.51

8.4 Traction coefficients

The traction force as described in section 7.2.2 is dependent on four parameters. To identify these parameters, a test is performed in which the body of the Crawler moves around the member by actuating all three wheels. The test is performed on the test offshore mem-

ber, since it has the dedicated size the crawler is designed for. The test member has the rough dry paint surface. For this test it was not possible to change the surface conditions as it was done for the friction test in section 4.1. During the test, the crawler speed is measured, and the single-wheel speed. This allows the slip ratio to be determined as described in equation 7.12. The Maxon motors that drive the wheels are set to measure the torque required to rotate around the member. It is taken into account that the crawler uses three wheels for this task. From the torques measured in the motor, the actual traction force between the wheel and the member can be found.
The experiment is carried out by accelerating and breaking the crawler multiple times to generate sufficient data to determine the slip curve of the wheel. For the experiment, the crawler is positioned on the vertical offshore structure. During the test, it was observed that the crawler lost altitude on the structure while rotating. This deviation is not taken into account in the experiment. The crawler dynamics in this project is considered moving in one degree of freedom and thereby the longitudinal direction only, thus the slip angle is considered zero and the traction force is the pure driving and breaking force.

The torque data for the three crawler wheels recorded in the experiment is shown in the figure 8.5 from left to right, plotted as the wheel torque over time. The data correspond only to the raw input torque data of the wheel. Only in further modeling of the crawler model are viscous friction and rolling friction subtracted.

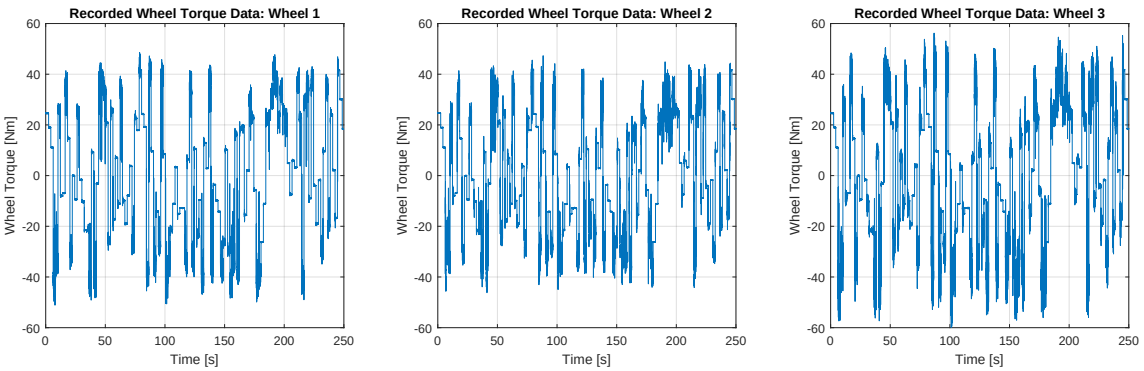


Figure 8.5: Raw input wheel torque data recorded during the experiment plotted against the sample time.

The raw input torque data shows that the wheel torque switches between -42 Nm for the breaking part and 42 Nm for the acceleration part, for all three wheels. For torque data, the test was based on a random torque input that included a wide variety of frequencies, to ensure the whole bandwidth of slip ratios.
The slip ratio is determined as described in equation 7.12 in section 7.2.1.
As mentioned in section 7.2.2, the slip curve shows the longitudinal friction force as a function of the slip ratio. The data used for the slip curve indicate the slip ratio along the x-axis and the longitudinal force in Newton along the y-axis. The longitudinal force is the total force found from the torque data of the wheels used for the torque graphs in figure 8.5. The slip curve determined for the experiment resembles the curve presented in figure 7.4b. The fitted slip curves are depicted in figure 8.6 and use the magic tire func-

tion from equation 7.13 as the fitting model. It fitted the parameters to the model using nonlinear least square method

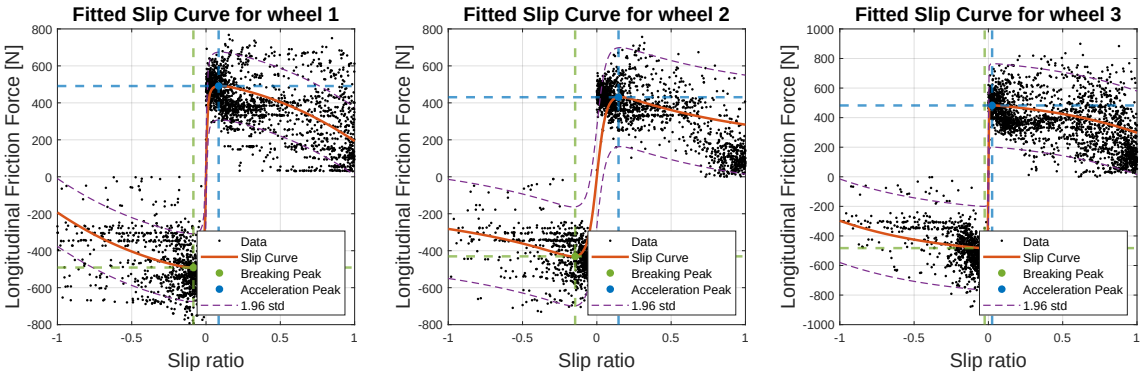


Figure 8.6: Slip curves for all three wheels , resulting from the slip test.

From the fitted slip curve, it is observed that wheel one and wheel three follow the same curve fit and both have a steep transition slope between breaking and accelerating. The slope of the slip curve for the second wheel is considerably different from the other two wheels and shows a clear indication of a force peak of 430.5 N at a slip ratio of around 0.15. The first wheel slip curve peaks at 491 N with a slip ratio of 0.1. The peak in the third wheel's slip curve is located at a force of 482 N for a slip ratio of 0.05.

The recorded data points are scattered with up to 500 N difference for both the breaking and acceleration. However, as can be seen in figure 8.6, a higher density of data points can be observed. The higher density of points gives a directive for the behavior of the fitted slip curves, which shows a coefficient of determination (R^2) of 0.89 for wheel one and 0.9 for wheel three. The R-squared value for wheel two is the lowest with 0.81.

As shown in figure 8.6 the violet dotted lines highlight the upper and lower bounds for the two times standard deviation, equivalent to a confidence interval of 95%.
The traction force parameters and their standard deviation are documented in table 8.6.

Table 8.6: Magic tire parameters for the slip curve fit.

	μ_B	μ_C	μ_D	μ_E
Wheel 1	91.87± 10.12	0.07±4.51	2.73±187.69	1.01±0.00
Wheel 2	8.92±2.50	1.94±0.46	0.13±0.00	0.79±0.33
Wheel 3	968.10±218.30	0.98±0.15	0.19±0.02	1.00±0.00

Here μ_B is the stiffness coefficient, μ_C is the shape coefficient, μ_D is the peak coefficient and μ_E is the curvature coefficient. The peak adhesion coefficient is defined as μ_p.

As seen in table 8.6, the standard deviations for the magic tire parameters are to large and the parameters between the wheels are largely deviating. The parameters can not be validated being conclusive. However, the graphs show similar characteristics to slip curves as discussed in section 7.2.2 and the fit for the magic tire equation gives a R-squared value of 0.8 to 0.9. The functions developed for the three wheel slips in figure 8.6 are therefore further used.

The peak adhesion coefficients can be determined using the documented force peaks for when the crawler is breaking or accelerating. This is done for every wheel separately. Using equation 8.5, the peak adhesion coefficient is found.

$$F_{peak} = \mu_p N_i \tag{8.5}$$

Where F_{peak} is the force peak in the slip curves, μ is the desired peak adhesion coefficient and N is the normal force for the corresponding wheel. The peak adhesion coefficients are documented in table 8.7.

Table 8.7: The Force peaks for the wheel slip curves and the determined peak adhesion coefficient.

	Force Peak [N]	Peak adhesion coefficient	Peak Slip [%]
Wheel 1	$\pm$ 491.00	0.17	0.10
Wheel 2	$\pm$ 430.50	0.13	0.15
Wheel 3	$\pm$ 482.00	0.16	0.05

Since it is not possible to recreate the traction force tests with the various surfaces mentioned in section 4.1 another approach is used to develop the traction force coefficients for the remaining surfaces.
Adhesion is one of the contributing factors for friction effects. The adhesion friction is described as a tangential force required to shear off the adhesive bonds between two surfaces, while friction is a measure of the shear strength between them [33] [34]. Based on that it is assumed that the estimation of the peak adhesion coefficients for different surface conditions can be describes as in the following section 8.4.

Estimation of surface condition slip curves

Experiments have been conducted in section 4.1 in order to determine the friction coefficients of different surface conditions which may occur under a crawler operation.

Of all experimental friction coefficients, the dry rough paint friction coefficient documented in appendix A.1, is the only coefficient also determined experimentally using the crawler slip curve for its three wheels. The dry rough paint surface coefficients are used as a reference to estimate an approximate slip curve for the remaining surface conditions.

The goal in this section is to perceive if it is possible to estimate the traction coefficients for the different surface conditions, which can be used to simulate the crawler dynamics and slip when a surface change is encountered during an offshore operation.

To estimate the slip curve characteristics of a different surface condition, the relation of the dry rough paint friction coefficient and the friction coefficient of the desired surface is found. The relation is then multiplied by the slip curve function of the rough paint surface from figure 8.6. The estimated slip curves on the different surface conditions for the three wheels is shown in figure 8.7.

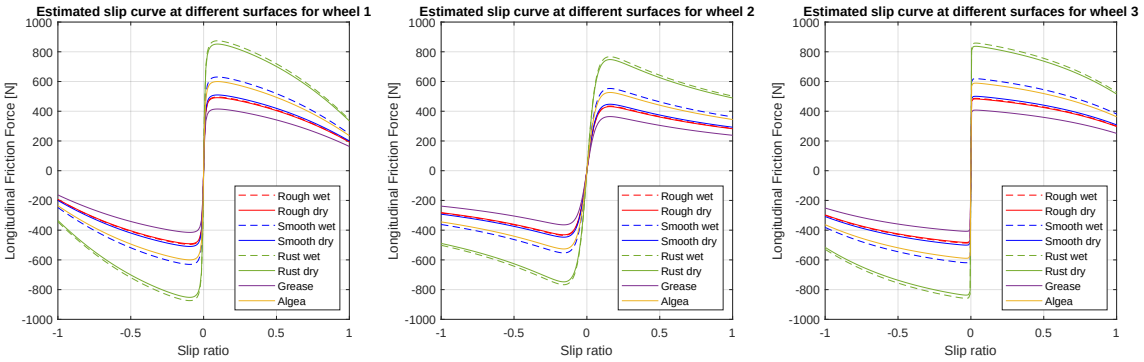


Figure 8.7: Estimated slip curves for the different surface conditions based on the friction coefficients from section 4.1.

The slip curves seen in the figure have colors for the specific surfaces and are indicated as dotted lines if it is a wet surface condition. It is observable that the highest longitudinal traction force is the wet rust with 874.01 N while the lowest of 414.98 N is for the grease test. The fact that the wet rust condition is the highest and the grease condition is the lowest is the case for all three wheels. The remaining surface conditions peak forces and peak adhesion coefficients are documented in table 8.8.

Table 8.8: Comparison of peak adhesion coefficients μ_p found for the different surface conditions, using the rough paint surface test as reference.

	Wheel 1: Adhesion coefficient		Wheel 2: Adhesion coefficient		Wheel 3: Adhesion coefficient	
Condition	Dry	Wet	Dry	Wet	Dry	Wet
Smooth paint	0.17	0.21	0.13	0.16	0.16	0.2
Rough paint	0.17	0.17	0.13	0.13	0.16	0.19
Rust	0.29	0.3	0.22	0.23	0.27	0.28
Algea	N/A	0.2	N/A	0.16	N/A	0.19
Grease	0.14	N/A	0.11	N/A	0.13	N/A

9 Model validation

To use the models developed and the parameters derived to develop control algorithms, it is essential that these models match the real world. To ensure this, a validation process is established within this chapter. The clamp and rotation models are subjected to various experiments, which will be explained in the following sections.

9.1 Validation of clamp model

The purpose of this experiment is to validate the crawler clamping model and to map the force distribution between each of the wheels and the actuator.

Test setup and experimental procedure

The equipment used in this experiment can be seen in table 9.1. In order to measure the normal force on each wheel, an HBM load cell is placed, as seen in figure 9.1.

Figure 9.1: Location of load cell attached to wheel before being pressed against the member

Data from load cells are collected using a Spider8 data logger 9.1. This takes the small voltage difference that happens when a strain gauge is deformed when a force is applied. changing its internal resistance as a result of its length change. This small voltage is then amplified by the use of a Wheatstone bridge, inside Spider 8. The force supplied by the piston is found by measuring the pressure in the hydraulic piston.

Table 9.1: Equipment used for the clamp model validation

Equipment	Producer	Model	Resolution	Range	Sampling rate	Calibration	Offset
Pressure meter	Wika	46852703	10 bar $\pm$ 1.6%	0 - 100 bar	-	-	0
Datalogger	HBM	Spider8	10^{9} mV/V	-	256 kHz	-	0
Load cell wheel 1	HBM	C8C	10^{9} mV/V	$\pm$ 5 mV	-	5109 $\frac{N}{mV/V}$	288 N
Load cell wheel 2	HBM	C9C	10^{9} mV/V	$\pm$ 5 mV	-	395 $\frac{N}{mV/V}$	62.5 N
Load cell wheel 3	HBM	C9C	10^{9} mV/V	$\pm$ 5 mV	-	416 $\frac{N}{mV/V}$	80.5 N

The load cells need to be calibrated before use in order to make sure that the amount of Newtons measured correlates correctly with the mV/V measured by the Spider8. Each of the load cells is subjected to an increasing force ranging from 0 to 112 N.

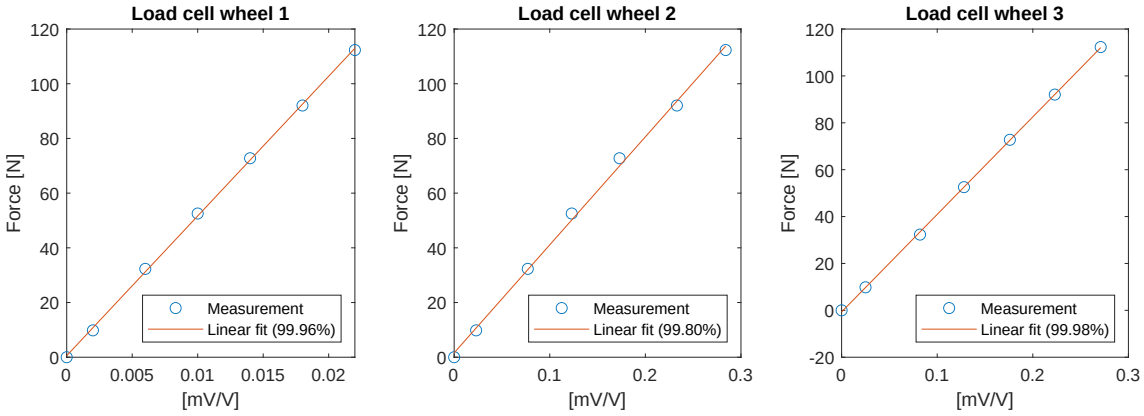


Figure 9.2: Calibration of the three load cells

From figure 9.2, it can be seen that there is a proportional relationship between the force applied and the measured mV/V. By linear regression, 1mV / V is found to be equal to 5109, 395, and 416 N for load cell 1, load cell 2, and load cell 3, respectively. This calibration is also noted in table 9.1 cells are bought calibrated from the manufacturer where load cell one is calibrated at 5000 N and load cell two and three are calibrated at 500 N, so the results of the calibrations are deemed reasonable.

9.1.1 Test on member

In this tests the member is in a vertical position, this is chosen in order to rule out the effect the gravity could have on the measured normal force. The crawler is fixed on top of the supporting structure to ensure that the only forces entering the wheels are those excited by the piston. With only three load cells available, a support of same height as the load cell is mounted on the second wheel in each pair. It is still assumed that the amount of force that passes through the load cell is equal to that going through the other wheel. The test setup for the vertical member is depicted in figure 9.3.

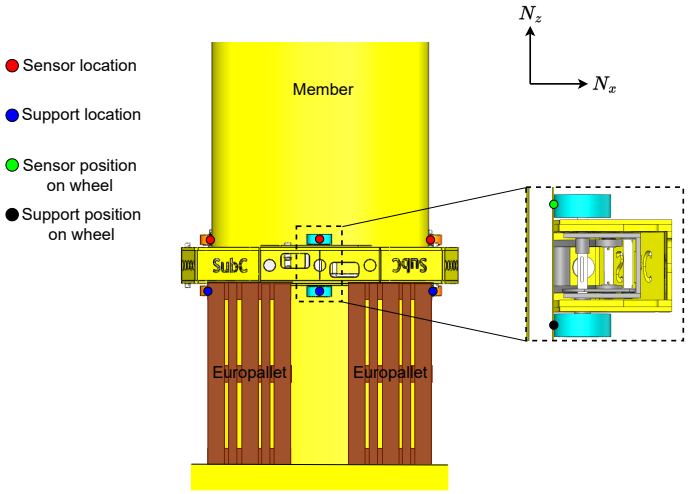


Figure 9.3: The experimental setup for validating the clamp model for the vertical member position.

The test is performed by increasing the piston pressure from 0 to 100 bar, in increments of 10. This test is performed three times, then the load cells are rearranged to the other wheels, and the test is performed three times again. The results from these tests can be seen in figure 9.4 where they are compared to the simulation model output.

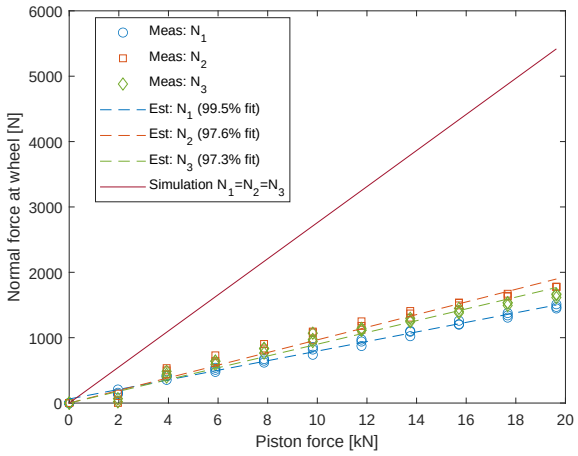


Figure 9.4: Results from the clamp model validation test with a vertical member

At first glance, it would seem that the model is overestimating the amount of force available at each wheel. However, to rule out any discrepancies and investigate why there is an offset between the force at each wheel, further tests are performed where the load cells are rearranged, and a new mount to better fix the load cell to the wheel is designed and used. When investigating the original mount of the load cells, it is seen that some of the force is lost to the outer casing of the load cell, since it is indented into the softer nylon. To avoid this problem, a casing mount for the load cell is constructed as seen in figure 9.5.

Figure 9.5: Updated casing for the load cell

The casing of the load cell is mounted on a metal plate to ensure an even distribution of the force on the wheel. A similar test to the previous one is performed using the new load cell casings. The results of this can be seen in figure 9.6.

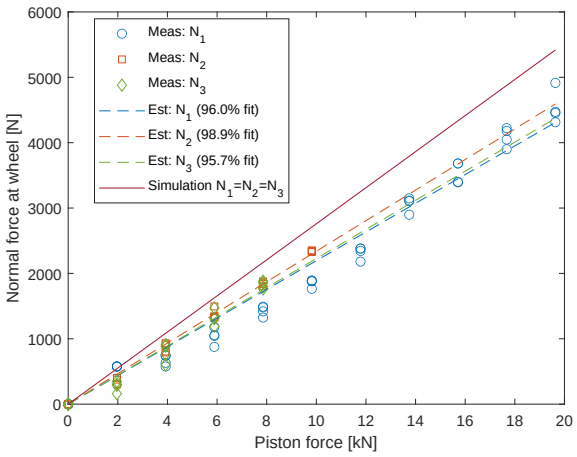


Figure 9.6: Revised results from the clamp model validation test with a vertical member

With the updated sensor casings, it is seen that each load cell can capture a higher amount of force. It can be seen that the load cells that measure the force N_2 and N_3 reach their upper measurement range at 50 and 40 bar, respectively. A first-order polynomial has been fitted to each of the measurements, and, by this, an offset has been identified. This offset has been retracted from the figure and inserted into table 9.1. Some discrepancies between the simulation results and the measurements can still be identified. It was assumed that the force measured at each wheel would be identical, however, as results show, this is not the case. The angle between the wheels is not found to be 120° as assumed. The true angle between wheels 2 and 1 is found to be 116.77° and between wheels 2 and 3 is found to be 115.93°. This can explain the deviation between the simulations and N_1,N_2 and N_3 of 25.7%, 17.9% and 23.9%, respectively. To ensure that the forces through the wheels of each pair are indeed equal, a further test is performed to investigate this. This is done by mounting a load cell on the top and bottom wheels in each pair. This test

is performed three times, measuring the force through a different pair at each test. The results can be seen in figure 9.7.

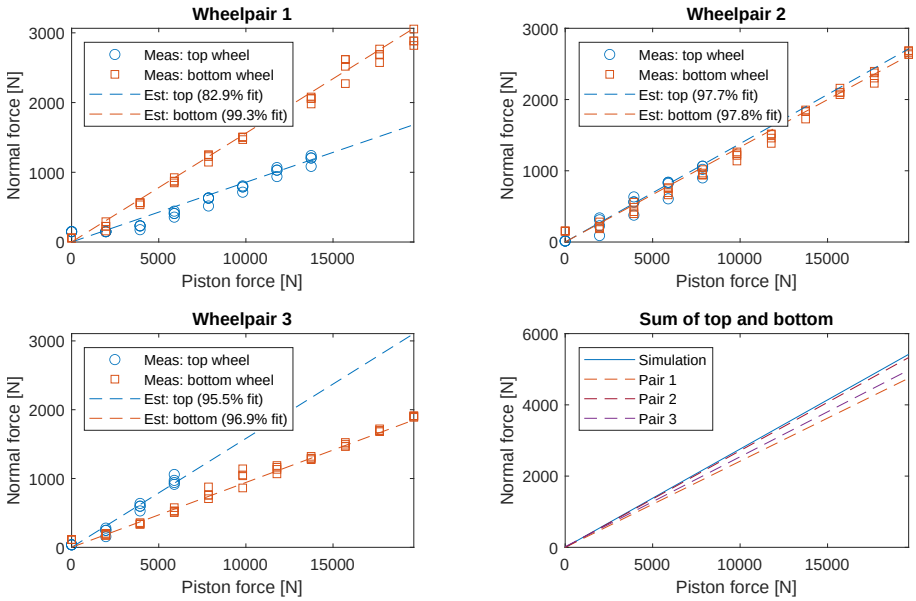


Figure 9.7: The force measurement on each of the wheel pairs

From these tests, it can be seen that there are discrepancies between the top and bottom wheels. This is most significant at wheel pairs one and two. By further investigation, it is found that the shaft mounted between the two wheels in pair one is out of alignment. The misalignment is measured to be 5.2° This causes the bottom wheel to contact the member before the top wheel, and as seen in the top and bottom left figures, the majority of force is moving through the bottom wheel as well. The opposite case is seen at pair three, however the shaft in this pair is not out of alignment. Since the magnitudes of difference in increments are close to equal and these pairs are opposite, it is assumed that this is all caused by the misalignment of the shaft in pair 3. When adding the forces measured on the top and bottom wheels, and matching them to the output of the model, it can now be seen that the discrepancy between the model and the measured forces is down to 14.12%, 1.70%, and 9.30% for N_1, N_2, and N_3, respectively. Since friction within the hydraulic actuator is not taken into consideration when calculating the piston force, it would be assumed that the simulation would output slightly higher forces than what can be measured at the wheel. It is still assumed that the difference in the magnitude of force between each wheel pair is caused by the angular offset between the physical crawler and what was expected. The identified offsets between simulation and measurements are introduced to the model and can be seen in table 9.4.

Table 9.2: Revisions for the clamp model

Parameters	β_1	β_3	N_1 scale	N_2 scale	N_3 scale
Revisions	116.77°	115.93°	0.1412	0.0170	0.0930

To conclude on the validation tests performed on the crawler mounted on a vertically oriented member the input piston force can be converted from piston pressure in bar to piston force (F_p) by a scale of 196.3495 N/bar, and the conversion from F_p to N_1,N_2, and N_3 by a scale of 0.2370, 0.2712, and 0.2503. These are also collected in table 9.3

Table 9.3: Conversions between input pressure, piston force and normal force at each wheel when the crawler is in a vertical position

	Input pressure to piston force	Piston force to normal force		
Parameters	P_i [bar] $\rightarrow$ F_p [N]	$F_p \rightarrow N_1$	$F_p \rightarrow N_2$	$F_p \rightarrow N_3$
Conversions	196.3495 [N/bar]	0.2370	0.2712	0.2503

9.2 Validation of rotationel model

To validate the rotational model, the crawler is actuated using a step input. The measured input torque of the three motors is used as input to the model described in 7. The angular velocities of the wheel and crawler are measured and compared to the model. Tests are performed on a vertical and horizontal oriented member.

9.2.1 Vertical member

The first set of tests is carried out on a vertically oriented member, the motors are actuated with a step input ranging from 0 to 26 Nm, and the piston pressure is 51.2 bar. The crawler is initiated at 0 radians as seen in 3.8, and the step input occurs at time six seconds. Due to this, the results are only shown from 5 to 20 seconds. The result of the tests can be seen in figure 9.8.

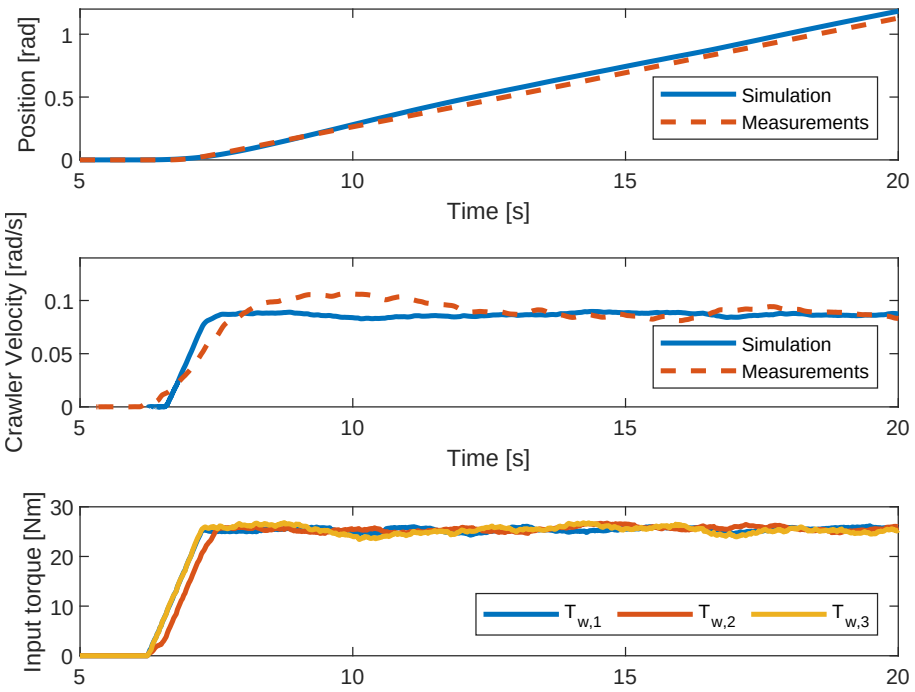


Figure 9.8: Comparison of angular velocity and position between measurements and simulations

As seen in the figure, the model reacts as expected to the step input, as does the physical crawler. The physical crawler has an overshoot in the transient stage, before it converges to a steady state velocity. This the model does not incorporate, multiple tests were performed that showed the same result. The overshoot of the physical crawler is found to be caused by the tightening of the chain at some wheels caused by a skewed shaft, as explained in section 9.1. The fluctuations of the crawler measurements at steady state velocity are caused by measurement uncertainties of the IMU found in section 5.3 and the roughness of the surface. Even though the velocity fluctuates the end position of the crawler compared to measurements is found to be 1.12 and 1.17 radians of measurements and simulations, respectively. This correlates with what is found in the real-life tests and is deemed to be acceptable for the tests.

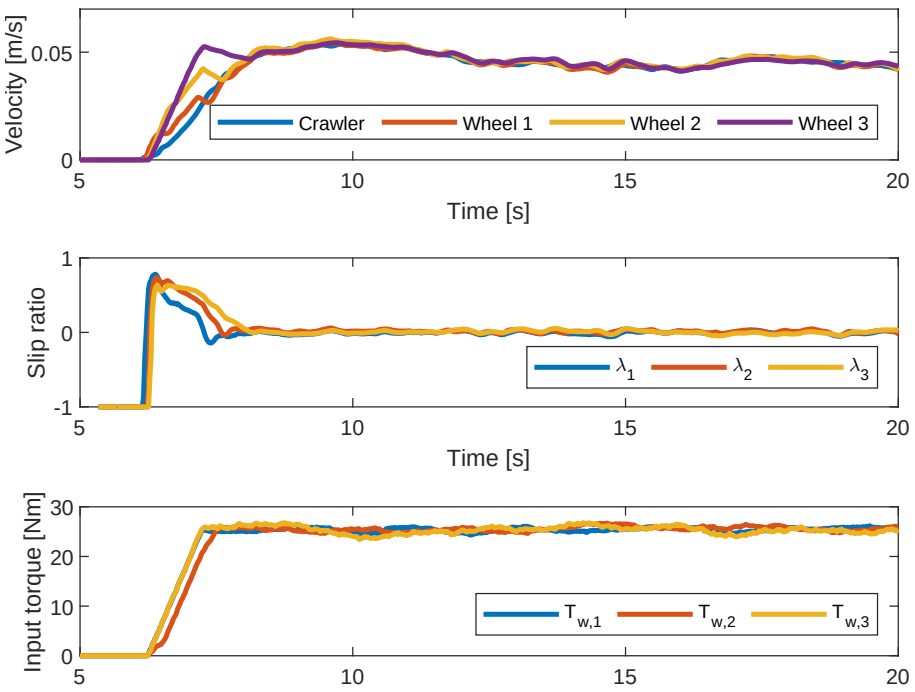


Figure 9.9: Comparison between longitudinal speeds and slip ratios

As seen in figure 9.9 it is found that the longitudinal velocity of the crawler and the wheels synchronize within 1.4 to 2 seconds and it is clearly seen that during the acceleration phase there is a slip between the wheels and the crawler that is also expected. As soon as the wheels and crawler are synchronized, it is expected that the slip ratio between each of the wheels and the crawler will converge to zero. This is also seen in the figure, however, with some small fluctuations, again caused by the uneven surface and measurement uncertainties.

Observations during the tests

During the tests, the crawler was found to have difficulty moving with a low torque input, but when it moved it did so with high velocity. It was assumed that this is caused by the high static friction of the rolling fiction which is inherent in the high normal force on each of the wheels.

During testing, it is also observed that the crawler tends to have a vertical slip when moving around the member. By measuring the distance between the crawler and the floor when starting a rotation and then again after a full rotation, it is found that the crawler slips 0.02 m per rotation. It is assumed that this is caused by the difference in normal force at each of the wheels or by the misalignment of the axles at wheels one and three. A solution to this issue has not been investigated, since on the original crawler the two clamps are used to realign the crawler laterally.

9.2.2 Horizontal member

The second set of tests is carried out on a horizontally oriented member. At the beginning of these tests, it was found that the amount of torque available when limiting the motors to 42 Nm was not sufficient to overcome the gravitational pull. Due to this, the 50% limitation of the motors was increased to 65% for short durations. This enables the motors to be actuated up to 54.4 Nm each. For the following tests, the motors are actuated with a step input ranging from 0 to 50 Nm, and the piston pressure is 63 bar. The crawler is initiated at -1.1 radians since this is the resting position of the crawler when it is hanging on the down side of the member. The step input occurs at 3 seconds. The result of the tests can be seen in figure 9.10.

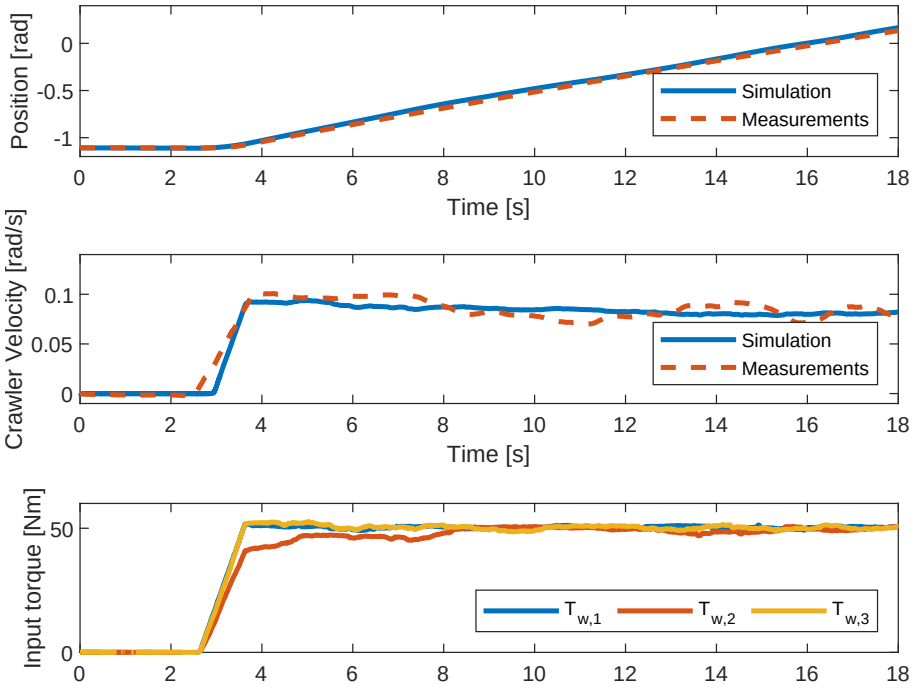


Figure 9.10: Comparison of angular velocity and position between measurements and simulations

From the figure it is seen that when actuated the physical crawler tends to overshoot the model with 0.02 rad/s, and then it starts to fluctuate around the simulation. It is clearly seen that the model reacts to the change in gravitational pull as it affects the velocity of the crawler, based on its position around the member. The discrepancies between measured and simulated velocities must be caused by the irregular surface conditions posed by the test member. The final position is within an acceptable difference of 0.0335 radians, which also matches reality.

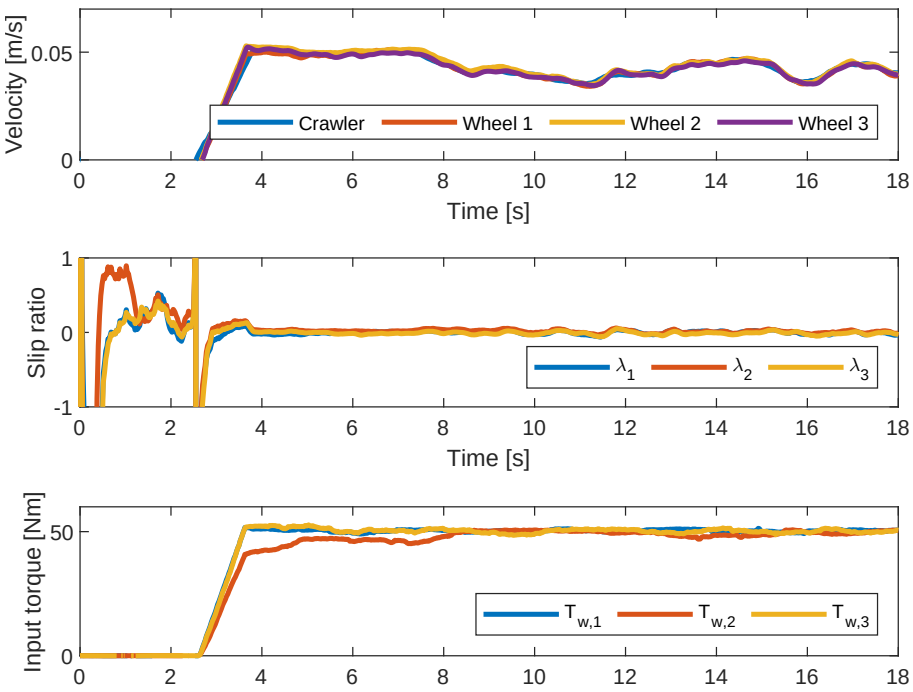


Figure 9.11: Comparison between longitudinal speeds and slip ratios

As seen in figure 9.11, the slip ratio is highest during the acceleration phase as was expected. Due to the irregular surface, the velocities between crawler and wheel measurements vary throughout the entire simulation. This is also reflected in changes of the slip ratio, it can be seen that whenever the crawler is able to obtain a steady-state speed or synchronize the longitudinal speed between crawler and wheel, the slip ratio converges to zero.

Observations during the tests

As for the previous tests, some observations were made when performing these tests. The effect of the rolling friction force was also evident in these tests, however the impact is quite different when combining it with the gravitational forces as well. A lateral movement of the crawler was also observed as expected on the horizontal oriented member.

After performing the tests on the horizontal member the following parameters were updated.

Table 9.4: Revisions for the rotationel model

Parameters	b_1	b_2	b_3	μ_r
Revisions	39.846	40.740	43.703	-

9.3 Validated model parameters

Based on the performed tests from the previous chapters, all model parameters are know found at updated to match the physical crawler model. All used model parameters are found in the table below.

Table 9.5: All validated model parameters used in this project

Name	Variable	Value	Unit
	Environment		
Diameter of test member	L_d	1.08	[m]
Gravitationel acceleration	g	9.82	[m/s^2]
	Crawler		
Weight of Crawler	m	132	[kg]
Inertia of Crawler	J_c	70.46	[m^2/kg]
Distance from pivot to COM	L_1	0.2703	[m]
Distance from pivot to crawler wheels	L_2	0.5080	[m]
	Wheels, motor, and drive train		
Inertia of wheel	J_w	0.004	[m^2/kg]
Radius of wheel	R_w	0.0625	[m]
Internal motor gearing	G_1	353	-
External motor gearing	G_2	1.4	-
Drivetrain viscous friction coefficient	$\{b_1, b_2, b_3\}$	{39.85,40.74,43.70}	
Motor torque constant	$\{k_t\}$	{0.0232}	[Nm/A]
DC gain motor controller	$\{C_1, C_2, C_3\}$	{0.97,0.98,1.03}	-
	Lumbed friction model		
Viscous friction coefficient	μ_v	3775.3 $\pm$ 321.51	
Rolling friction coefficient	μ_r	0.0390 $\pm$ 0.004	
	Traction forces		
Stiffness coefficient	$\{\mu_{B,1}, \mu_{B,2}, \mu_{B,3}\}$	{91.87,8.92,968,10}	
Shape coefficient	$\{\mu_{C,1}, \mu_{C,2}, \mu_{C,3}\}$	{0.07,1.94,0.98}	
Peak coefficient	$\{\mu_{D,1}, \mu_{D,2}, \mu_{D,3}\}$	{2.73,0.13,0.19}	
Curvature coefficient	$\{\mu_{E,1}, \mu_{E,2}, \mu_{E,3}\}$	{1.01,0.79,1.00}	
Peak adhesion coefficient	$\{\mu_{p,1}, \mu_{p,2}, \mu_{p,3}\}$	{0.17,0.13,0.16}	

Clamp model			
Pinned point $\perp$ to wheels	$\{L_{1A}, L_{2C}, L_{3B}\}$	{0.1740,0.0993,0.1574}	[m]
Pinned point $\perp$ to attachment (R)	$\{L_{RB}, L_{RC}\}$	{0.0904,0.2402}	[m]
Pinned point $\perp$ to attachment (L)	$\{L_{LA}, L_{LC}\}$	{0.0892,0.1298}	[m]
Piston $\perp$ to pinned C	L_{PC}	0.1812	[m]
Angle between wheel 2 and 1	β_1	116.77	[°]
Angle between wheel 2 and 3	β_3	115.93	[°]
Scale of normal forces	$\{N_1, N_2, N_3\}$	{0.141,0.017,0.093}	-

10 Control Design

As to designing a control system to automate the crawler and fulfill the requirements set in section 2.1 certain aspects must be taken into account. This chapter will introduce the issues which can occur when controlling the crawler and how to deal with these issues.

The Crawler has several different aspects that need to be controlled. Firstly, the normal force has to be large enough to generate the required friction to keep it from falling when hanging on a vertical member. It needs to prevent the wheels from sliding when moving around the member. It must be able to tackle obstacles on its path, such as barnacles. The crawler should also be able to move with a constant velocity and to a desired location on the member.

The main objective of the crawler controller is to:

Reach a designated set-point position around the member.

Also important, especially when performing specific inspection tasks, is to:

Reach the position at a reference velocity.

When trying to reach these objectives during an offshore operation, the crawler can encounter conditions that are not ideal. These are represented by the following three scenarios.

- **Slip on a single or multiple wheels**
  If this scenario occurs, the solution can be to increase the torque on the remaining wheels or reduce the torque to an allowable maximum on the sliding wheels. If this is enough to continue at the desired velocity set-point, no further action needs to be applied.
- **Slip on all wheels** If the slip between the three wheels and the crawler reaches 100% the crawler cannot move. To resolve this issue, it would be desired to increase the pressure of the piston in order to gain a higher traction, if this is not possible, the operation has to be terminated.
- **Blockage of one or all wheels** If any or all wheels are blocked, an initial attempt could be to increase the amount of torque available above the set limit for a short

time, if this is not enough to resolve the issue, the piston pressure should be reduced to decrease the required torque to overcome the obstacle. It is important that the pressure is not reduced below the minimum threshold at which the crawler begins to slip downward. If this does not resolve the issue, the operation must be terminated.

In the following sections, different control schemes will be investigated in order to determine whether it is efficient to develop a simple scheme or if a more complex scheme can be used to increase the performance of the robot. In figure 10.1 the major components and their connections to the controllers can be seen.

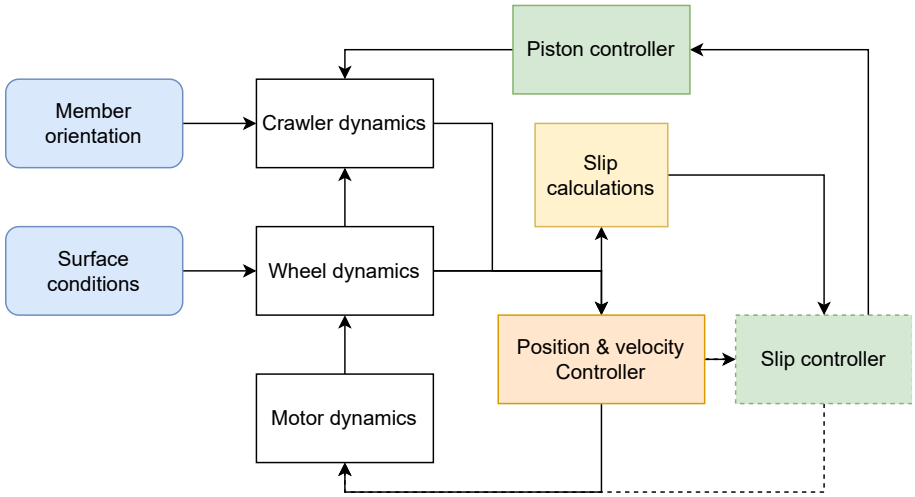


Figure 10.1: Blockdiagram of connections between environmental inputs to the model as well as crawler connections between the different dynamics and calculations.

From figure 10.1 it can be seen that to test the different scenarios the surface conditions for the wheel dynamics can be changed to slip or block a single or more wheels. The crawler and wheel dynamics is used as input to the position and velocity controllers and can also be used as input to the slip calculations if the slip controller is used. The piston controller determines the amount of pressure the crawler is attached to. This controller can either operate using a static reference input or by knowledge of slip or torque to either increase or decrease pressure, this is obtained by an internal logic controller.

In chapter 9 it is found that the horizontally oriented member requires the most effort from the crawler to overcome obstacles. This model also includes the gravitational terms, thus the controllers developed in this section are based on a crawler operating on a horizontal oriented member. To further align the development of the controller, the input pressure in all models is set to 50 bar and the controllers are tuned to reach a position reference of 1 rad in a time frame of maximum 62.50 seconds, based on the requirements of 2.3.

10.1 Position control

For the crawler to reach and stay in a given position, a simple position controller is developed. This controllers purpose is to allow Subc Partners to designate a desired starting position fro the crawler before its operation. This is achieved by using a negative feedback loop, measuring the output of the crawler, and comparing this to the reference. This gives the error and can be used as input to a control scheme that will define a new input to the system. Such a topology can be seen in figure 10.2. The reference input or reference position is marked as $r(t)$, $e(t)$ is the error or deviation from the position, fed into the controller, and $u(t)$ is the controller output while also being the system input. The input is the motor torque τ_m. The system output is marked as $y(t)$, which is the crawler position θ_C.

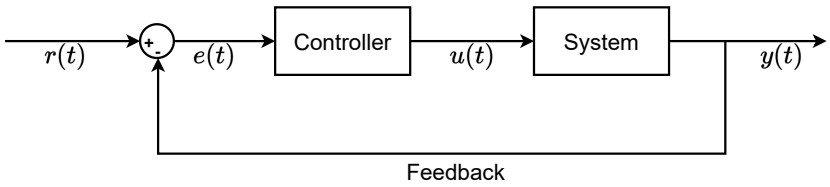


Figure 10.2: Block diagram over the negative feedback control topology.

To develop the controller, the Internal Model Control (IMC) approach is used, based on the knowledge of the developed model. The IMC approach develops a controller that is robust to noise and other disturbances. If the IMC scheme is developed based on the exact model of the system, a perfect control is theoretically possible [35], [36]. Since this approach develops a transfer function that can be converted into PID coefficients, it is easily implemented in the robot.

10.1.1 Developing the controller

The IMC formulation results in only one tuning parameter, which is the closed-loop time constant τ_{IMC}. A general structure of the IMC is depicted in figure 10.3.

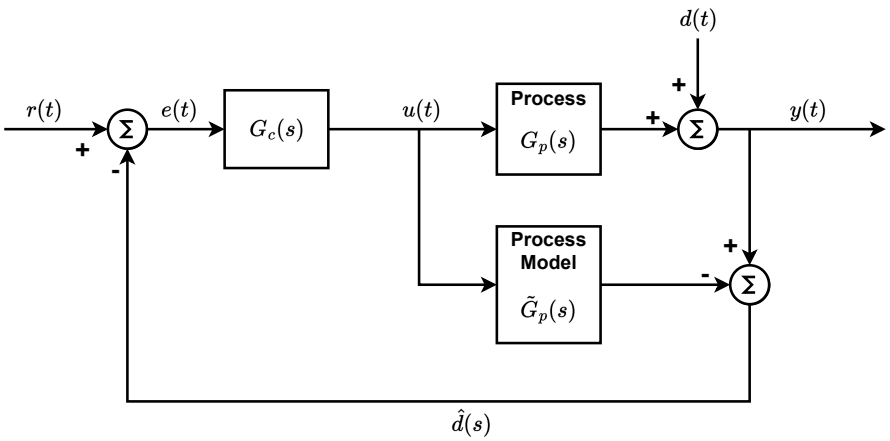


Figure 10.3: The general structure of an IMC.

To design the IMC controller, the model of the process $\tilde{G}_p(s)$ is factored into an invertible ($\tilde{G}_p^+(s)$) and a noninvertible ($\tilde{G}_p^-(s)$) component as seen in equation 10.1[37].

$$\tilde{G}_p(s) = \tilde{G}_p^+(s)\tilde{G}_p^-(s) \tag{10.1}$$

This is necessary because of the non-invertible component that contains terms which, when inverted, will cause instability and reliability problems, such as terms containing positive zeros and time delays[37].

Next the new controller model parts is defined as $G_c(s)$ in equation 10.2 and $G_{IMC}(s)$ in equation 10.3.

$$G_c(s) = \tilde{G}_p^+(s)^{-1} \tag{10.2}$$

$$G_{IMC}(s) = G_c(s)G_f(s) \tag{10.3}$$

A filter is implemented such that the transfer function does not include more zeros than poles. The order of the filter depends on the difference between these. Here $G_f(s)$ is a low-pass filter of the appropriate order represented in equation 10.4.

$$G_f(s) = \frac{1}{(1+\tau_f s)^n} \tag{10.4}$$

τ_f is the filter parameter and n is the order of the filter. When designing the low-pass filter the filter parameter τ_f is initially chosen to be twice as fast as the open loop response of the model[35].

The IMC approach does not use unknown disturbances, therefore the system discussed in this section is assumed to be $G_p(s) = \tilde{G}_p(s)$, where $G_p(s)$ is the actual process and $\tilde{G}_p(s)$ is the model describing the process. The closed loop for the system can therefore be represented as follows in equation 10.5 [37].

$$y(s) = \frac{G_{IMC}(s)G_p(s)r(s) + \left(1 - G_{IMC}(s)\tilde{G}_p(s)\right)d(s)}{1 + \left(G_p(s) - \tilde{G}_p(s)\right)G_{IMC}(s)} \tag{10.5}$$

Where $r(s)$ is the reference to the system and $d(s)$ is an unknown disturbance.

Due to the current case of $G_p(s) = \tilde{G}_p(s)$ the closed loop for this system is rewritten as in equation 10.6[37].

$$y(s) = \tilde{G}_p^-(s)G_f(s)r(s) + \left(1 - \tilde{G}_p^-(s)G_f(s)\right)d(s) \tag{10.6}$$

To determine the IMC for the system, the Process plant $G_p(s)$ is required. For that, the crawler dynamics has to be linearized.

10.1.2 Model linearization

For determining the process plant describing the crawler dynamics, the non-linear crawler model is linearized. The crawler model is described in chapter 7 and is shown in the equations 7.6, 7.7, 7.8, 7.9, 8.4 and 7.11. For simplicity, the crawler model is assumed to be an SISO system, using the crawler position as output and the combined wheel torque as input. A more comprehensive description of the linearization process is seen in appendix C.2.

To develop the transfer function used as the base plant for the PID design, the Laplace transform is used. The transfer function for the horizontal position is designed and applied for all crawler placement scenarios, since it is the most challenging for the crawler to move when sitting on a horizontal offshore member. The devised transfer function is shown in equation 10.7. Offsets such as $\mu_v(N_1 + N_2 + N_3)$ from the rolling friction force in equation 8.4 and the gravity term 7.11, are neglected.

$$G_p(s) = \frac{\theta_c(s)}{T_{in}(s)} = \frac{0.2249}{s^2 + 254.1s} = \frac{0.2249}{s(s + 254.1)} \tag{10.7}$$

A short stability analysis can be performed by determining the poles of the process plant. By taking the denominator equal to zero, s can be found.

$$0 = s^2 + 254.1s \Rightarrow s = 0 \wedge -254.1$$

It is observed that one pole is stable, located on the left half plane with -254.07 and the other pole having the value 0 and thereby being marginally stable. The system is considered marginally sable.

In order to validate the linear model for the crawler dynamics around the horizontal member, an open-loop test is conducted. The open-loop test is then compared to the non-linear crawler model described in section 9.2.2. The test piston pressure is set to 50 bar and initialized at -1.1 radians. Figure 10.4 shows the comparison for the position in the top and the crawler velocity in the middle. The input torque for the crawler wheels is depicted in the lower graph.

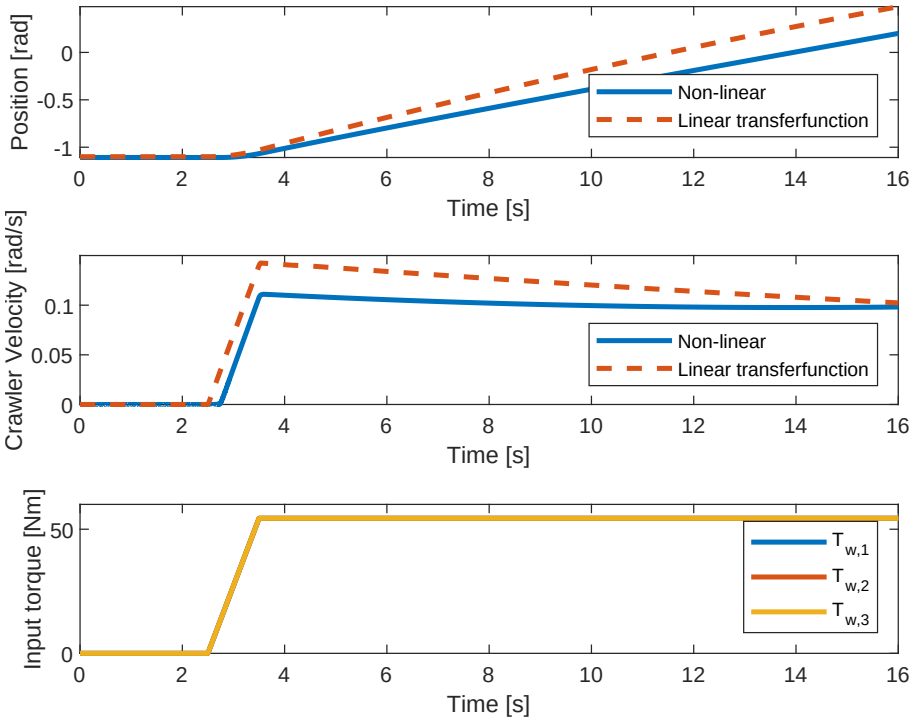


Figure 10.4: Comparison of linear and nonlinear crawler rotationel models

For the position and velocity comparison, the non-linear model is shown as the blue line while the linear transfer function is represented by the orange dotted line. The step size for the linear model is set to 2.5 seconds to match the non-linear model. It is observed that the linear model has a steeper slope and has an end position of around 0.5 radians, while the nonlinear model has an end position of 0.17 radians. The two models deviates with 0.33 radians. The deviation is deemed to be of acceptable range and the model is observed and validated further.

For the crawler velocity, it is seen that the linearized model has a higher initial response of approximately 0.4 rad/s than the nonlinear model. The linear model reaches the same crawler velocity as the non-linear model of 0.1 rad/s when the time passes the 16 seconds of simulation time. Taking into account that the step time being 2.5 seconds, the actual time for the two models to align is 13.5 seconds.

The torque input for all wheels is set to the same value of 53.6 Nm.

The linear transfer function is validated to represent the non-linear model which allows to design an IMC based PID controller for position control.

PID coefficients

Due to the difficulty of implementing the IMC controller defined in section 10.1.1 on a physical system, the IMC is rewritten into an conventional PIDF structure using the mason's rule presented in equation 10.8.

$$G_{PID}(s) = \frac{G_{IMC}(s)}{1 - G_{IMC}(s)\tilde{G}_p(s)} = \frac{\tilde{G}_p^+(s)^{-1} G_f(s)}{1 - \tilde{G}_p^-(s) G_f(s)} \tag{10.8}$$

The conventional PIDF structure is depicted in the blue box in figure 10.5, including an anti-windup on the integral part, marked in the orange box, which prevents integration wind-up in PID controllers when the actuators are saturated [38].

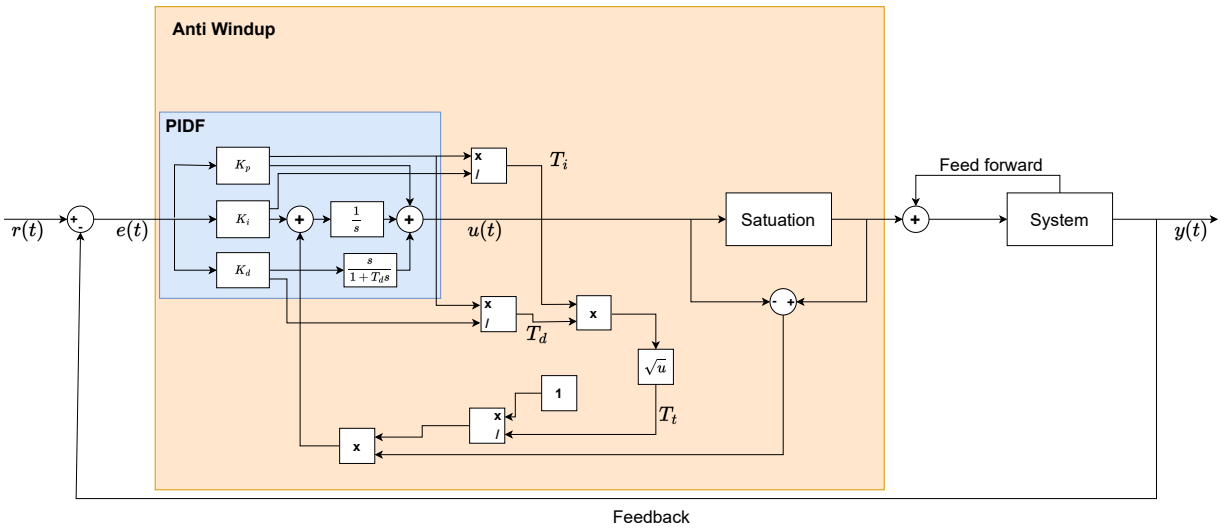


Figure 10.5: Block diagram showing the structure of a basic parallel PIDF controller.

A feed forward structure is used, shown in figure 10.5, in order to include lost dynamics, when the PID is implemented into the non-linear system.
The process plant for the existing system $G_p(s)$ from equation 10.7 does not contain a non-inevitable part which is used for determining the IMC based PID controller. This means that for the Mason's rule in equation 10.8, the denominator only depends on the low-pass filter, while the non-inevitable part has the properties $\frac{1}{1}$. A second order low pass filter is required to offset the two poles found in the system, which turns to zeros when inverted. The filter coefficient τ_f is tuned such that the responds given correlates to the requirements given as stated in the start of the control chapter. While the invertible part is simply the same as the process plant so $\tilde{G}_p^+(s) = G_p(s)$ in this case. The approximated IMC-PID controller is stated in equation 10.9[36].

$$G_{PID}(s) = \frac{0.04464s^6 + 22.7s^5 + 2888s^4 + 796.8s^3 + 77s^2 + 3.038s + 0.04194}{s^6 + 254.5s^5 + 111.4s^4 + 16.58s^3 + 0.9928s^2 + 0.01938s} \tag{10.9}$$

The IMC based PID controller as found in equation 10.9, is currently in the 6th order and cannot be implemented in the classical approach. It would required multiple PID controller in series. this it is not in the typical depiction of a traditional PID controller 10.5. So the system has to be reduced in order. This is done algebraically instead of using

the numbers. Rewriting the transfer function 10.7 To the following structure first.

$$G_p(s) = \frac{K}{s(\tau s + 1)} \tag{10.10}$$

The PID controller coefficients 10.5 can be obtained after algebraic simplifications [36]. The results are found in table 10.1 for the position controller for the crawler.

Table 10.1: The found coefficients for the PID position controller

$K = 0.2249$	$\tau = 0254.1$		$\tau_f = 9.5$
$K_p = 45.2547$	$K_i = 2.2716$	$K_d = -215.2638$	$T_d = 4.7619$

The determined proportional, integral, and derivative gains are tested and the results can be seen in figure 10.6. The top figure in 10.6 shows the position of the crawler in the test with time. It can be seen that the reference crawler position of 1.1 radians is reached within 60 seconds, which fulfills the requirements described in section 2.3. The crawler velocity in rad/s is seen in the middle figure, shows an increase of 0.04 rad/s at 10 seconds, which decreases slowly over 50 seconds when the crawler is slowly approaching the reference position. The input torque, seen in the bottom figure, reaches the maximum allowed torque of 54 Nm per wheel at around 15 seconds and slowly decreases until reaching steady state position of 48 Nm at 60 seconds.

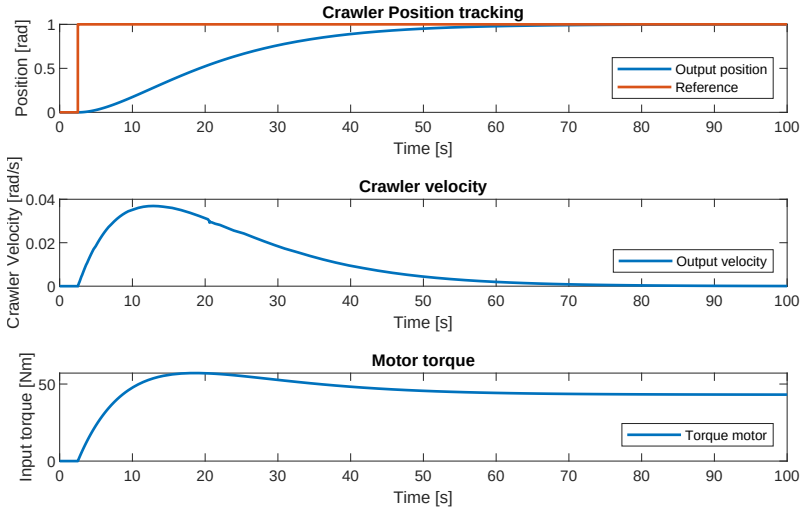


Figure 10.6: PID based on IMC simulation in normal conditions on rough paint with a position reference of 1 rad.

The results show that the PID can track the crawler position. The PID can further be used in the nonlinear model.
Since the PID controller is tuned and designed based on the IMC, the determined tuning parameters as stated in table 10.1 are meant for the linear model. The determined controller parameters are used as initial tuning parameters used for the non-linear model

PID controller.

To match the non-linear model, the dynamics lost in the linearization are introduced back into the controller by using feed forward. Before introducing the terms into the model they are multiplied by the wheel radius, converting it into a force.

The PID is additionally supplied by a anti-windup on the integrator term in order to satuate the integral term so the system is not overshooting. The proportional (K_p), integral (K_i) and derivative (K_d) gain are used in order to determine the integral time (T_i) and derivative time (T_d) used to find the tracking time T_t used for the anti-windup. The determination of the anti-windup variables is shown in equation 10.11

$$T_i = \frac{K_p}{K_i} \qquad T_d = \frac{K_d}{K_p} \qquad T_t = \sqrt{T_i T_d} \tag{10.11}$$

The PID parameters are tuned to fit the non-linear model and documented in table 10.2 below.

Table 10.2: Tuned PID controller gains for the non-linear model.

K_p	K_i	K_d
45.2547	22.716	215.2638

The tuning parameters are similar to the IMC based PID controller with only changes in the integrator gain with a factor of 10 and the change from a negative to a positive value in the derivative gain.

10.2 Position and velocity control

The IMC based PID controller is effective for controlling single-input, single-output (SISO) systems, or multiple-input, multiple-output (MIMO) systems which are decoupled. However, if coupling takes place between the input and output relations, the controller structure becomes more complex, and the number of parameters to tune increases as well. Since the developed model has multiple references and is over-actuated with multiple inputs, single outputs. A controller capable of handling these couplings could be preferred. The Linear Quadratic Regulator (LQR) controller is based on the state-space representation of a model consisting of multiple linear equations. By weighing the importance of states and inputs of the system, a single-gain matrix can be derived based on the most optimal gains derived from a cost function. The topology of an LQR control scheme can be seen in figure 10.7.

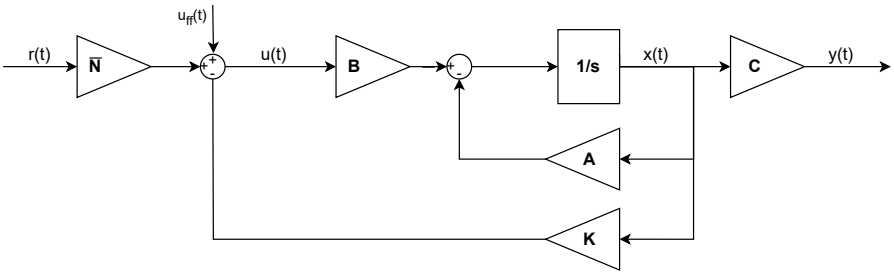


Figure 10.7: Block diagram of the state space model with LQR control

For the LQR controller to be able to track both position and velocity, two controllers are developed, one to track for position and one to track for velocity. A simple gain scheduling is developed where the velocity controller is initialized and when the crawler is within 5% of the position reference, the position controller is introduced instead of the velocity controller.

10.2.1 State space representation of the model

To design the LQR controllers, the nonlinear model equations of the rotational crawler need to be presented in state space form. To obtain this, equation 7.6 to 7.9 is solved for the acceleration parts, and non-linear terms are linearized as seen in 10.1.2. The general notation for the state space representation is seen in equations 10.12 and 10.12.

$$\dot{\boldsymbol{x}}(\boldsymbol{t}) = \boldsymbol{A}\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{B}\boldsymbol{u}(\boldsymbol{t}) \tag{10.12}$$

$$\boldsymbol{y}(\boldsymbol{t}) = \boldsymbol{C}\boldsymbol{x}(\boldsymbol{t}) + \boldsymbol{D}\boldsymbol{u}(\boldsymbol{t}) \tag{10.13}$$

Where $\boldsymbol{A}$ is the system matrix, $\boldsymbol{B}$ is the input matrix, $\boldsymbol{C}$ is the output matrix, $\boldsymbol{D} = 0$ is the direct transition matrix, $\boldsymbol{x}$ is the state vector, and $\boldsymbol{u}$ is the input vector.

The state vector consists of the position and velocities within the system equations and is described as.

$$\boldsymbol{x} = \begin{bmatrix} \theta_c & \dot{\theta}_c & \omega_1 & \omega_2 & \omega_3 \end{bmatrix}^T \tag{10.14}$$

The input vector contains the three motor inputs, as well as the measured slip ratios. By rewriting the motor torque to the torque available at the wheels, each input can be written as $T_{w,i} = T_{m,i}G_1G_2$ and can be described as equation 10.15.

$$\boldsymbol{u} = \begin{bmatrix} T_{w,1} & T_{w,2} & T_{w,3} \end{bmatrix}^T \tag{10.15}$$

By using full state feedback, all states in the state vector need to be measured or estimated. The positions and velocities of the wheels are measured directly using the rotary encoders, where the angular velocity and position of the crawler are measured by the

IMU. The system is linearized with operating points at 0 rad for position, 50 bar input pressure, and with a horizontally oriented member. A further explanation of the linearization can be seen in C.3. The linearized system is found to be controllable and stable with poles at:

$$\boldsymbol{\lambda_p} = \textbf{-47.1668, -43.9690, -43.0041, -27.3136, 0.0946}$$

10.2.2 Developing the position controller

The LQR controller uses a gain matrix to move the eigenvalues to a desired location. This location is determined by weighing the importance of states and inputs. The optimal solution for this problem is then found by solving a cost function based on the Ricatti equation. The cost function is seen in equation 10.16.

$$\boldsymbol{J_c} = \int_0^\infty \left(\boldsymbol{x}(t)^T \boldsymbol{Q} \boldsymbol{x}(t) + \boldsymbol{u}(t)^T \boldsymbol{R} \boldsymbol{u}(t)\right) \mathrm{dt} \tag{10.16}$$

Where $\boldsymbol{K} = \boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S}$ is the control matrix and $\boldsymbol{A}^T\boldsymbol{S} + \boldsymbol{S}\boldsymbol{A} - \boldsymbol{S}\boldsymbol{B}\boldsymbol{R}^{-1}\boldsymbol{B}^T\boldsymbol{S} + \boldsymbol{Q} = 0$ is the Ricatti equation used to derive $\boldsymbol{S}$. The $\boldsymbol{Q}$ and $\boldsymbol{R}$ matrices are used to adjust the cost function. This is done by penalizing the different states and inputs, respectively. As a good initial guess, the Brysons rule can be used [32].

$$\boldsymbol{Q} = \mathrm{diag}\left(\frac{\alpha_1}{{x_{1,max}}^2} \cdots \frac{\alpha_n}{{x_{n,max}}^2}\right) \tag{10.17}$$

$$\boldsymbol{R} = \mathrm{diag}\left(\frac{\beta_1}{{u_{1,max}}^2} \cdots \frac{\beta_n}{{u_{n,max}}^2}\right) \tag{10.18}$$

Here, α_n and β_n can be used to tune individual states and inputs based on the initial guess. The initial guess uses the maximum value of state and the input squared. The maximum values are based on the requirements set in 2.3 and the physical limitations of the components.

$$\boldsymbol{x_{max}} = (1, 0.016, 1.12, 1.12, 1.12) \tag{10.19}$$

$$\boldsymbol{u_{max}} = (54, 54, 54) \tag{10.20}$$

10.2.3 Precompensation

Since the amount of input is different from the amount of states in the model, the reference input needs to be rescaled. This can be done using a precompensator ($\bar{\boldsymbol{N}}$), this compensator can also remove any steady-state error introduced by the LQR controller. [32]. The compensator is located between the reference input and the feedback as seen in figure 10.7. The precompensator can be found by the following equation.

$$\bar{\boldsymbol{N}} = \boldsymbol{N_u} + \boldsymbol{N_x}\boldsymbol{K} \tag{10.21}$$

Here, $\boldsymbol{N_u}$ is the compensation of the input states, and $\boldsymbol{N_x}$ is the compensation of the feedback states. Defining an augmented matrix (M) as equation 10.22.

$$\boldsymbol{M} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{B} \\ \boldsymbol{C} & \boldsymbol{D} \end{bmatrix} \tag{10.22}$$

$\boldsymbol{N_u}$ and $\boldsymbol{N_x}$ can be found by analyzing the steady-state values of the system matrices as seen in equation 10.23.

$$\begin{bmatrix} \boldsymbol{N_x} \\ \boldsymbol{N_u} \end{bmatrix} = \boldsymbol{M}^{-1} \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{10.23}$$

Here, 0 is a zero matrix and I is the identity matrix. Since this system has three inputs but five states, this result in the augmented matrix being non-symmetric, thus it is not possible to calculate the inverse. Instead, the pseudo-inverse can be calculated. The Moore-Penrose technique is widely used [39],[40] however, for this project it was found to produce a $\bar{\boldsymbol{N}}$ which is not able to remove the steady-state error. To improve the stability of the pseudo-inverse calculation, Tikhonov regularization can be used instead [39]. The Tikhonov regularization pseudo-inverse can be found by equation 10.24.

$$\boldsymbol{M}^{-1} = \left(\boldsymbol{M}^T\boldsymbol{M} + \varphi\boldsymbol{I}\right)^{-1}\boldsymbol{M}^T \tag{10.24}$$

Where $\boldsymbol{I}$ is the identity matrix and φ is the regularization parameter that controls the amount of regulation used. This parameter can be used to tune the stability of the calculation by regulating the augmented matrix before it is inverted. By scaling the reference using A.5 it can be ensured that $\boldsymbol{y} = \boldsymbol{r}$ where r is the reference input of the controller [32]. This is obtained by computing the steady-state values of the state and the control input, which will result in a zero output error.

$$\boldsymbol{u} = \boldsymbol{u}_{ss} - \boldsymbol{K}(\boldsymbol{x} - \boldsymbol{x_{ss}}) \tag{10.25}$$

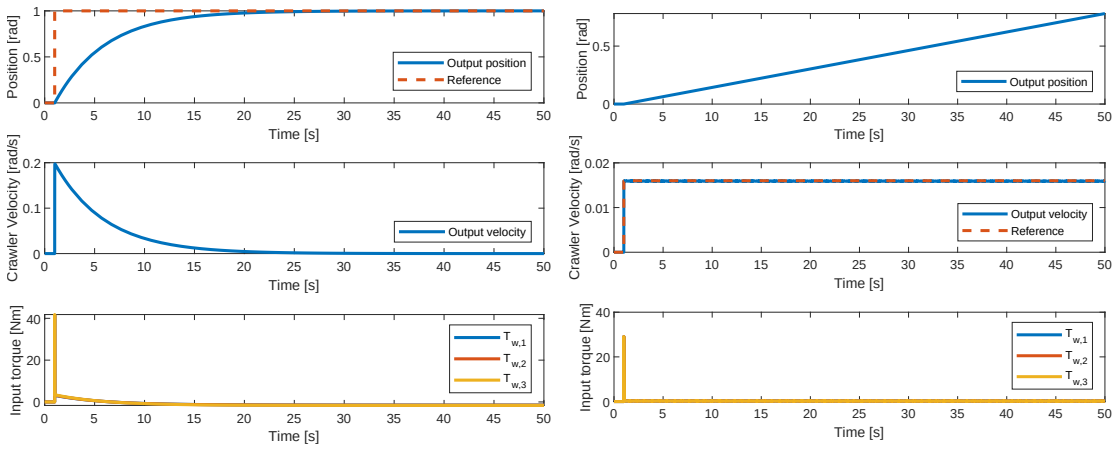
Here, $\boldsymbol{u}_{ss}$ are the steady-state input and $\boldsymbol{x}_{ss}$ is the steady-state state. As seen in equation 10.25 if $\boldsymbol{x} = \boldsymbol{x}_{ss}$ then $\boldsymbol{u} = \boldsymbol{u}_{ss}$ and the steady-state error have been removed.

10.2.4 Control structure

With all terms taken into account, the control structure introduced in figure 10.7 is then simplified as equation 10.26 solving the desired input to reach a certain state.

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x} + \bar{\boldsymbol{N}}\boldsymbol{r} \tag{10.26}$$

By applying the controller using the initial tuning based on Bryson's rule, the state-space model produces a rise-time of 82.32 s with a position reference of 1 rad. By setting $\varphi = 3.17$, the steady-state error is fully compensated. To reduce the rise time, the matrix $\boldsymbol{Q}$ is changed to reduce the penalty set to the crawler position. This reduces the rise time to 24.61 s with a settling time of 60 s. In this case, φ is changed to 1.1.



(a) LQR position controller

(b) LQR velocity controller

Figure 10.8: Response of the position and velocity controllers tuned using the LQR approach

The controller response when applied together with the linear model can be seen in figure 10.8a for the position controller and 10.8b for the velocity controller. Here it can be seen that both controllers are able to track the reference.

The controller matrices used for both controllers can be found in A.4 and A.6 and for the position and velocity controller, respectively. The $\bar{N}$ matrices can be found in A.5 and A.7.

The two controllers are combined to track both position and velocity through the use of gain scheduling. The results of tracking a position of π rad and 0.016 rad/s can be seen in figure 10.9.

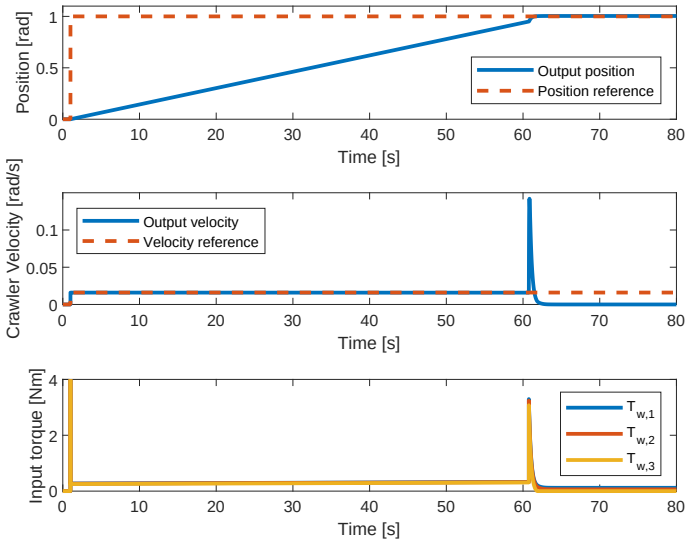


Figure 10.9: LQR tuned controller with gain schedueling, shifting controller at 5% from position reference

From figure 10.9 it can be seen that the controller is able to track the velocity reference of 0.016 rad/s until it is within 5% of the position reference of 1 rad. At this point, the

controller is changed to track the position and the velocity and torque is briefly increased. When the position reference is reached, the velocity is set to zero, and the position is kept.

The LQR tuned controller with gain schedueling is slower to reach a steady state position than the stand alone position controller, however it is now able to operate at a desired reference velocity as required. The controller is stable able to make the crawler reach the reference position within the time limit of 100 s.

10.3 Slip control

This section is based on the following paper [41]
To ensure that the crawler is not slipping a slip controller is designed for each wheel. The controller is robust to changes in driving surface, avoiding slippage. Increasing the amount of traction the crawler has when accelerating and decelerating. It does this by acting as a combined ABS (Anti-lock Braking System) and TCS (traction control system).

10.3.1 Slippage

As discussed in section 7.2.1 the amount of traction achieved depends on the slippage. In figure 7.4a it is seen that there is an unstable region on the right side of the peak and a stable region on the left side. There is also another stable point at the very end of where the wheel is completely locked, but this is not desired as the traction there is lower. Therefore, as seen in table 8.7 the optimal is around ± 0.1 in slippage. Slippage from 10.27 can be defined as follows [26].

$$\lambda_i = \frac{r_w \omega_i - L_2 \dot{\theta}_c}{max(r_w \omega_i, L_2 \dot{\theta}_c)} \tag{10.27}$$

Where r_w, L_2 are the radius of the wheel and the member. $\omega_i, \dot{\theta}_c$ are the respective angular velocities of the wheels $i \in \{1, 2, 3\}$ and the crawler. The subscript i throughout this chapter always refers to the three wheels. The crawler must be able to move forward and backward around the member, brake and accelerate. To do this, the maximum value in the nominator is the switching criterion between accelerating when $r_w \omega_i > L_2 \dot{\theta}_c$ and braking when $r_w \omega_i < L_2 \dot{\theta}_c$. To avoid making both ABS and TCS a combined system is proposed in this project instead. Meaning the slippage λ can go from -1 to 1 as shown in figure 7.4b. Since slippage is normally defined as the difference between the longitudinal velocity and the wheels angular velocity, the same will be done here.

$$v = L_2 \dot{\theta}_c \tag{10.28}$$

10.3.2 Slip approximation

When examining the slip ratio equation seen in 10.27, it is seen that it becomes discontinuous when it goes from accelerating to decelerating and vice versa. Instead of making the system use two different controllers, an approximation of the slip ratio $\bar{\lambda}$ is used instead, allowing it to be continuous, using a sigmoid function defined as follows.

$$\sigma_i = \frac{1}{1 + \exp(-a(\frac{r_w \omega_i}{v} - 1))} \tag{10.29}$$

Here, a is the design parameter, where using a large value allows $\bar{\lambda}$ to approximate λ. Then the approximate slip ratio becomes.

$$\bar{\lambda}_i = \frac{r_w \omega_i - v}{\sigma r_w \omega_i + (1 - \sigma) v} \tag{10.30}$$

The approximate slip ratio and the actual slip ratio are visualized in figure 10.10. In this case, the approximated slip ratio converges to the actual slip ratio.

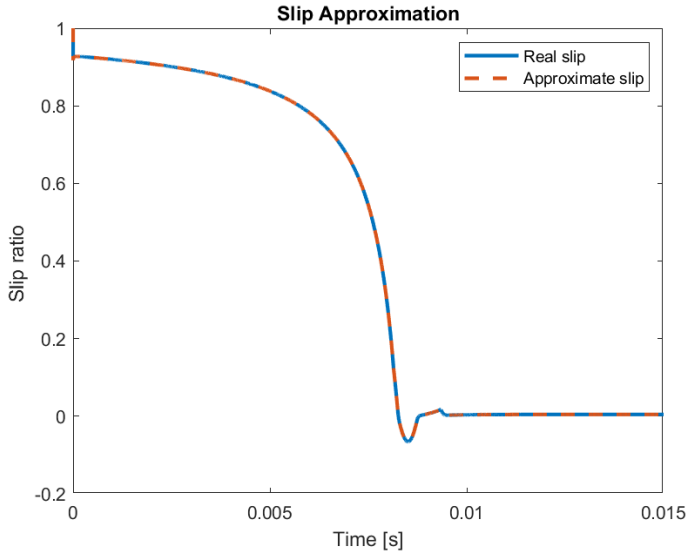


Figure 10.10: Comparison between approximate slip and real slip

10.3.3 Slip controller

The slip controller is designed by differentiating 10.30 combining it with the rotational model in 7.6 and the corresponding wheel models in 7.7,7.8, and 7.9.

$$\dot{\bar{\lambda}}_i = \frac{r_w v \dot{\omega}_i - r_w \omega_i \dot{v} - \dot{\sigma}(r_w \omega_i - v)^2}{\sigma r_w \omega_i + (1 - \sigma) v^2} \tag{10.31}$$

Which can be rewritten as the following, where only a third of the forces acting on the crawler are taken, as there are 3 separate wheels that should contribute in equal manner.

$$\dot{\bar{\lambda}}_i = \frac{r_w (v^2(1 + A)^2 - aA(r_w \omega_i - v)^2}{m J_w v^2 (r\omega_i + A v^2)} (mv(\tau_{c,i} + \tau_{d,i} - r_w F_{t,i}) - F_{cra} \tag{10.32}$$

Where A, $\dot{\sigma}$, F_{cra} and τ_g are defined as

$$A = \exp\left(-a(\frac{r\omega_i}{v} - 1)\right) \tag{10.33}$$

$$\dot{\sigma}_i = \frac{aA(m r_w v(\tau_{c,i} + \tau_{d,i} - r_w F_{t,i}) - J_w r_w \omega_i (F_{t,i} - \frac{1}{3}(F_{roll} + \frac{\tau_g}{L_1}))}{m J_w v^2 (1 + A)^2} \tag{10.34}$$

$$F_{cra} = J_w\omega_i(F_{t,i} - \frac{1}{3}(F_{roll} - \frac{\tau_g}{L_1})) \tag{10.35}$$

$$\tau_g = L_1 F_g(\theta_c)\cos(\alpha) \tag{10.36}$$

and $\tau_{d,i}$ is the demanded torque from the user.
The control torque $\tau_{c,i}$ can now be expressed by the following equation after using feedback linearization.

$$\tau_{c,i} = \kappa\frac{kJ_w v}{r}e_{\bar{\lambda}} + \frac{(r_w mv + J_w\omega_i)F_{t,i} - \frac{1}{3}J_w\omega_i(F_{roll} + \frac{\tau_g}{L_1})}{mv} - \tau_{d,i} \tag{10.37}$$

Where κ is defined as the following.

$$\kappa = \frac{(r_w\omega_i + Av)^2}{(v^2(1+A)^2 - aA(r_w\omega_i - v)^2)} \tag{10.38}$$

In this equation k>0 is the feedback gain, $e_{\bar{\lambda}}$ is the tracking error defined as

$$e_{\bar{\lambda}_i} = \bar{\lambda}_i - \lambda_d \tag{10.39}$$

where λ_d is the desired slip ratio, which is determined by

$$\lambda_d = \begin{cases} c_d, \text{ if } \lambda \geq 0 \\ -c_d, \text{ if } \lambda < 0 \end{cases} \tag{10.40}$$

c_d is a positive constant value, the optimal value of c_d could also be determined with the use of fuzzy logic of maximum power point tracking. In this study, the value is assumed to be constant, with a low enough value to stay on the left side of the adhesion curve 7.4a. The coefficients were determined in 8.6. When examining the control signal 10.44 the first term of the denominator is positive for all values of v and ω_i

$$v^2(1+A)^2 - aA(r_w\omega_i - v)^2 > 0 \; \forall \; v, \omega_i \tag{10.41}$$

By tuning the parameter a from 10.29, the tracking error $e_{\bar{\lambda}}$ will asymptotically converge to zero. This is because the $\bar{\lambda}$ asymptotically converges to λ_d because the error becomes.

$$\dot{e}_{\bar{\lambda},i} = \dot{\bar{\lambda}}_i = -ke_{\bar{\lambda},i} \tag{10.42}$$

To control the slip ratio the tracking error of $e_{\bar{\lambda}}$ is replaced by

$$e_{\lambda i} = \lambda_i - \lambda_d f \tag{10.43}$$

This gives the torque controller the following structure after replacing the tracking error.

$$\tau_{c,i} = \frac{kJ_w v(r_w\omega_i + Av)^2}{r(v^2(1+A)^2 - aA(r_w\omega_i - v)^2}e_{\lambda,i} + \tau_{cra} - \tau_{d,i} \tag{10.44}$$

Where τ_{cra} is the torque from the crawler.

$$\tau_{cra} = \frac{(r_w mv + J_w\omega_i)F_{t,i} - \frac{1}{3}J_w\omega_i(F_{roll} + \frac{\tau_g}{L_1})}{mv} \tag{10.45}$$

10.3.4 Slip stability analysis

To ensure stability, λ has been proven to be stable in both the acceleration and deceleration states. λ also asymptotically converges to λ_d using the control law.
This is achieved by differentiating 10.27 and substituting equations 7.6 and the corresponding wheel 7.77.87.9 as well as the torque control law found 10.44. The first case is for acceleration $r_w\omega_i > v$.

$$\dot{\lambda}_i = \dot{e}_{\lambda,i} = -\frac{kv^2(r_w\omega_i + Av)^2}{(r_w\omega)^2(v^2(1+A)^2 - aA(r_w\omega_i - v)^2)}e_{\lambda,i} \tag{10.46}$$

The other case is for deceleration $r_w\omega_i < v$.

$$\dot{\lambda}_i = \dot{e}_{\lambda,i} = -\frac{k(r_w\omega_i + Av)^2}{v^2(1+A)^2 - aA(r_w\omega_i - v)^2)}e_{\lambda,i} \tag{10.47}$$

The following Lyapunov function is used to check for stability.

$$\mathcal{V}(e_{\lambda i}) = \frac{1}{2}e_{\lambda,i}^2 \tag{10.48}$$

Taking the time derivative of the Lyapunov function 10.63 and using the trajectories of the closed loop system gives the following for acceleration

$$\dot{\mathcal{V}}(e_{\lambda i}) = \dot{e}_{\lambda,i}e_{\lambda,i} = -\frac{kv^2(r_w\omega_i + Av)^2}{(r_w\omega)^2(v^2(1+A)^2 - aA(r_w\omega_i - v)^2)}e_{\lambda,i}^2 \tag{10.49}$$

The same is done for braking

$$\dot{\mathcal{V}}(e_{\lambda i}) = \dot{e}_{\lambda,i}e_{\lambda,i} = -\frac{k(r_w\omega_i + Av)^2}{v^2(1+A)^2 - aA(r_w\omega_i - v)^2)}e_{\lambda,i}^2 \tag{10.50}$$

For both cases the time derivative of the Lyapunov function is less than 0 $\dot{\mathcal{V}}(e_{\lambda i}) < 0$. This is for all cases except when $e_{\lambda,i} \neq 0$. Since the Lyapunov function is positive definite and decresent. Resulting in $e_{\lambda,i}$ asymptotically converging to zero. Meaning that the slip ratio λ_i will also asymptotically converge to the desired slip ratio of λ_d.

10.3.5 Robust Slip control

In order to ensure that the crawler can operate under uncertain conditions such as waves and their added mass to the crawler, non-uniform members, obstacles on its path, unknown driving surface, etc., a robust control system is required. Therefore, forces such as traction on each wheel $F_{t,i}$ and rolling resistance F_{roll}, which were measurable in the parameter estimation 8, are hard to estimate and fluctuate in practice depending on the situation. The torque demanded τ_d is also included to allow the control system to assume the responsibility of ensuring that the wheels keep their traction, no matter the input. The torque from gravity is also added to ensure that the controller can handle this input.
The robust control method is based on a disturbance observer with sliding-mode control. The mass is allowed to fluctuate as Δm to handle added mass from the waves, and inaccuracies in the piston forces, rolling friction and viscous damping. So there is a nominal value m_0 that results in the mass being defined as

$$m = m_0 + \Delta m \tag{10.51}$$

The driving force is decoupled as the true driving value $F_{t,i}$ and the nominal value $F_{tn,i}$ found from the modeled traction curve from the real system 8.6. The true value is then defined with the nominal value and a state-bounded coefficient $\phi_{r,i}$. The coefficient changes then depending on the road surface conditions.

$$F_{t,i} = F_{tn,i} - \frac{1}{3}\frac{\tau_g}{L_1} + \phi_{r,i} \tag{10.52}$$

The robust slip controller is designed by differentiating the state equation with respect to λ_i 10.27 7.6 and the corresponding wheel 7.77.87.9. The following is then obtained for acceleration $r_w\omega_i > v$

$$\dot{\lambda}_i = \dot{e}_{\lambda,i} = \frac{r_w}{J_w r \omega_i^2}(\tau_{c,i} - r_w F_{n,i} + \bar{\tau}_{d,i}) - \frac{F_{n,i} - \frac{1\tau_g}{3L_1}}{m_0 r_w \omega_i} - \frac{\Delta v_i}{t_w \omega_i} \tag{10.53}$$

And for breaking $r_{wi} < v$

$$\dot{\lambda}_i = \dot{e}_{\lambda,i} = \frac{r_w}{J_w v}(\tau_{c,i} - r_w F_{n,i} + \bar{\tau}_{d,i}) - \frac{r_w \omega_i F_{n,i} - \frac{1\tau_g}{3L_1}}{m_0 v^2 \omega_i} - \frac{\Delta v_i}{t_w \omega_i} \tag{10.54}$$

Where Δv_i is defined as the following

$$\Delta v_i = (\frac{1}{m_0} - \frac{\Delta m}{m_0(m_0 + \Delta m}(\phi_{r,i} - \frac{1}{3}F_{roll}) - \frac{\Delta m}{m_0(m_0 + \Delta m)}F_{n,i} \tag{10.55}$$

Which can be shortend to

$$\Delta v_i = \frac{\phi_{r,i} - \frac{1}{3}F_{roll}}{m} - \frac{\Delta m}{m_0(m_0 + \Delta m)}F_{n,i} \tag{10.56}$$

In this case $\bar{\tau}_{d,i}$ is estimated by the disturbance observer as stated in equation 10.57.

$$\bar{\tau}_{d,i} = \tau_d - r_w \phi_{r,i} \tag{10.57}$$

The disturbance observer then uses a low-pass filter with the time constant τ_F to cut off the high-frequency component, resulting in $\hat{\bar{\tau}}_{d,i}$. A new control law can then be established which is able to reject disturbances as

$$\tau_{c,i} \ = \bar{\tau}_{c,i} - \hat{\bar{\tau}}_{d,i} \tag{10.58}$$

$$\tau_{c,i} = \kappa \frac{k J_w}{r_w} e_{\lambda,i} + \frac{r_w m_0 v + J_w \omega_i}{m_0 v} F_{n,i} - \frac{J_w \omega_i \frac{\tau_g}{L_1}}{3mv} - \rho_r \frac{e_{\lambda,i}}{|e_{\lambda,i}|} - \bar{\tau}_{d,i} \tag{10.59}$$

Where in this case ρ_r>0 is a design parameter. Using substitution on equation 10.5410.55 in the robust control law 10.59 and solving them with respect to $e_{\lambda,i}$ for both acceleration $r_w\omega_i > v$

$$e_{\lambda,i} = -\kappa \frac{k v^2}{(r_w \omega_i)^2} e_{\lambda,i} - \frac{v}{J_w r_w \omega_i^2}(\rho \frac{e_{\lambda,i}}{|e_{\lambda,i}|} + \frac{J_w \omega_i \Delta v}{v} - \tilde{\bar{\tau}}_{d,i}) \tag{10.60}$$

and breaking case $r_w\omega_i < v$.

$$e_{\lambda,i} = -\kappa k e_{\lambda,i} - \frac{r}{J_w v}(\rho \frac{e_{\lambda,i}}{|e_{\lambda,i}|} + \frac{J_w \omega_i \Delta v}{v} - \tilde{\bar{\tau}}_{d,i}) \tag{10.61}$$

Here the estimate error is defined as following.

$$\tilde{\bar{\tau}}_{d,i} = \bar{\tau}_{d,i} - \hat{\bar{\tau}}_{d,i} \tag{10.62}$$

to check if the robust controller is stable Lyapunov stability theorem is used with the following candidate function.

$$\mathcal{V}(e_{\lambda i}) = \frac{1}{2} e_{\lambda,i}^2 \tag{10.63}$$

Taking the time derivative of the Lyapunov function again and using the trajectories of the closed loop system of the robust system gives the following for acceleration

$$\dot{\mathcal{V}}(e_{\lambda,i}) = \dot{e}_{\lambda,i} e_{\lambda,i} = -\kappa \frac{kv^2}{(r_w\omega)^2} e_{\lambda,i}^2 - \frac{v}{J_w r_w \omega_i} e_\lambda (\rho_r \frac{e_{\lambda,i}}{|e_{\lambda,i}|} + \frac{J_w \omega_i \Delta v}{v} - \tilde{\bar{\tau}}_{d,i}) \tag{10.64}$$

The same is done for breaking

$$\dot{\mathcal{V}}(e_{\lambda,i}) = \dot{e}_{\lambda,i} e_{\lambda,i} = -\kappa k e_{\lambda,i}^2 - \frac{r_w}{J_w v} e_{\lambda_i} (\rho_r \frac{e_{\lambda,i}}{|e_{\lambda,i}|} + \frac{J_w \omega_i \Delta v}{v} - \tilde{\bar{\tau}}_{d,i}) \tag{10.65}$$

If ρ_r in 10.64 and 10.65 satisfies the following.

$$\bar{\rho_r} = \rho_r - (\frac{J_w \omega_i \Delta v}{v} - \tilde{\bar{\tau}}_{d,i}) \geq 0 \tag{10.66}$$

Then 10.64 and 10.65 can be rewritten as the following two equations.
For acceleration $r_w \omega_i \geq 0$

$$\dot{\mathcal{V}}(e_{\lambda,i}) = -\frac{kv^2 (r_w\omega_i + Av)^2}{(r_w\omega)^2 (v^2(1+A)^2 - aA(r_w\omega_i - v)^2)} e_{\lambda,i}^2 - \frac{\bar{\rho_r} v}{J_w r_w \omega_i^2} |e_{\lambda,i}| \tag{10.67}$$

And for breaking $r_w \omega_i < v$

$$\dot{\mathcal{V}}(e_{\lambda,i}) = -\frac{k(r_w\omega_i + Av)^2}{v^2(1+A)^2 - aA(r_w\omega_i - v)^2)} e_{\lambda,i}^2 - \frac{\bar{\rho_r} r_w}{J_w v} |e_{\lambda,i}| \tag{10.68}$$

So $\dot{\mathcal{V}}(e_\lambda, i) < 0$ for both acceleration and deceleration for all $e_{\lambda,i} \neq 0$ are stable for as long as the crawler and the wheels speeds are positive.

$$\frac{v}{J_w r_w \omega_i^2} > 0, \frac{r_w}{J_w v} > 0 \tag{10.69}$$

The Lyapunov function is positive definite and decrescent. Which results in $e_{\lambda,i}$ asymptotically converging to zero. This means that the slip ratio λ_i will converge to the desired slip ratio λ_d. The crawler can only operate in one direction since the speed always have to be positive defined. This means that it can not rotate both ways without implementing a switching statement that changes what it sees as its positive driving direction. A table containing all the chosen parameters can be found in table 10.3.

Table 10.3: Controller design parameters

Parameter	k	a	ρ_r	c_d	τ_F
Value	6	40	20	0.2	0.03

A diagram of the proposed control scheme can be seen in figure 10.11. Where it takes a reference slip ratio, which is found from section 8.4 to be for the individual motors. This reference is then evaluated with negative feedback from the current slip calculation. This creates an error which is fed into the control law, which gives the estimated control torque for the motor. This estimate is then subtracted from the disturbance observer torque. Which is taking the found traction force depending on the current slip, and goes through a low pass filter to estimate the disturbance. the actual torque command given to the motors. The crawler then measures its current angular velocities of the motor and its longitudinal velocity to calculate the current slip for the wheel. The same approach is done for all three wheels.

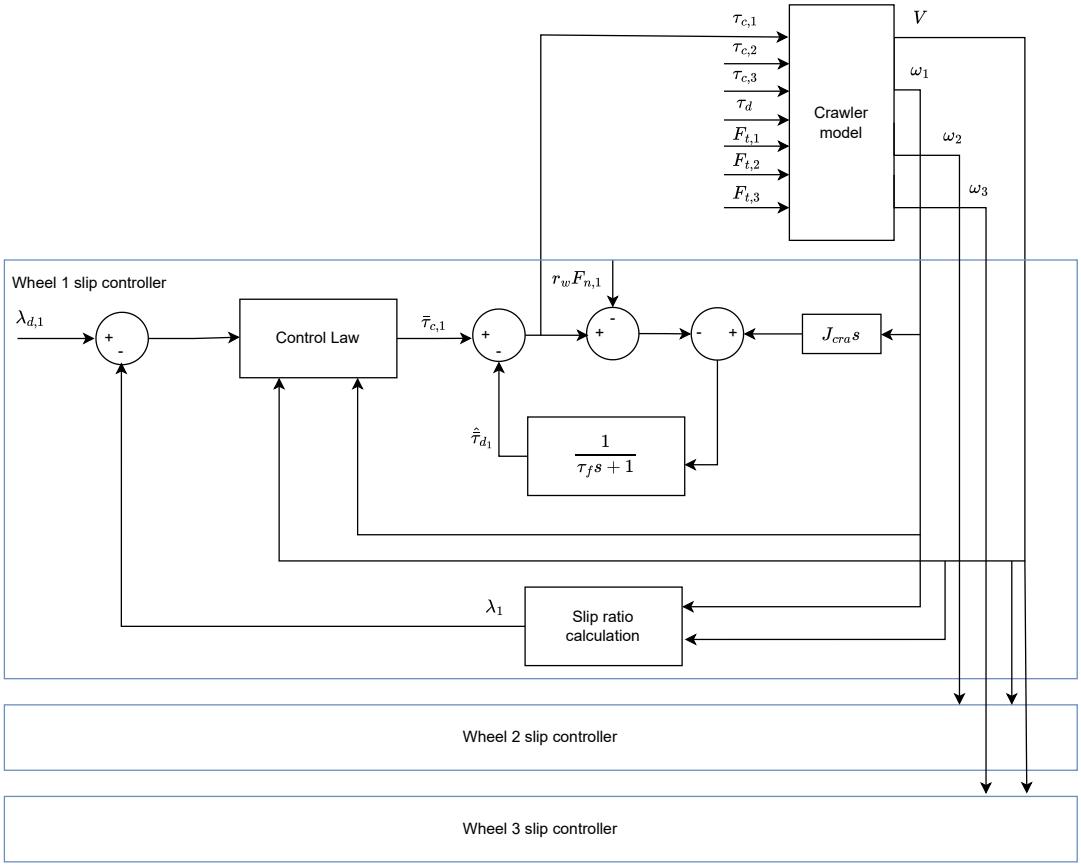


Figure 10.11: Robust control design block diagram structure of slip controller for individual wheels

To determine when the slip controller need to be active a hysteresis controller is added 10.70. When $\lambda_{i,con}$ is 0 it is off and when it is 1, it is turned on. The controller checks the change in slip, and the current slip value to determine whether the slip controller needs to be active for each individual wheel. It is also disabled when the crawler is moving close to zero, since it can not operate when the crawler is not moving and the position controllers need to be able to make the crawler settled on the desired location.

$$\lambda_{i,con} = \begin{cases} 0 & \text{for } abs(\dot{\theta}_c) < 0.001\text{rad/s} \\ 1 & \text{for } \bar{\dot{\lambda}}_i > 0.05\text{rad/s} \\ 0 & \text{for } \bar{\dot{\lambda}}_i < 0.01\text{rad/s} \\ \lambda_{i,con} & \text{remains unchanged if } 0.01 < \bar{\dot{\lambda}}_i < 0.05\text{rad/s} \\ 1 & \text{for } \bar{\lambda}_i > 0.1 \\ 0 & \text{for } \bar{\lambda}_i < 0.05 \\ \lambda_{i,con} & \text{remains unchanged if } 0.05 < \bar{\lambda}_i < 0.1 \end{cases} \tag{10.70}$$

10.4 Piston control

Controlling the piston is an essential part to avoid getting stuck on obstacles and ensuring that the normal force provided on the wheels are enough to grant the required traction and that the crawler is not falling of the member. friction force is used steady-statet the gravity pulling the crawler down when its mounted vertically and used to generate enough friction for the wheels to rotate as well. A logical controller is implemented to handle the following four conditions.

$$F_p = \begin{cases} 0 : F_p & \text{for } \Delta t_{F_p} < 5\text{s} \\ 1 : F_p + \Delta F_p & \text{for } \sum_{i=1}^{3}[s_i > 0.15] \geq 2 \\ 2 : F_p - \Delta F_p & \text{for } \sum_{i=1}^{3} |(T_{w,i})| \geq 120\text{Nm} \\ 3 : F_p + \Delta F_p & \text{for } \sum_{i=1}^{3}[\mu_i N_i - |(\frac{T_{w,i}}{r_w})| - mg\text{cos}(\alpha)(\text{cos}(\theta_c \pm \beta_i) < 100\text{N}] \geq 2 \\ 4 : F_p + \Delta F_p & \text{for } \sum_{i=1}^{3}[\mu_i N_i - mg\text{sin}(\alpha) < 500N] \geq 2 \\ 5 : F_p - \Delta F_p & \text{for } |(\dot{\theta}_c)| < 0.005\text{rad/s} \end{cases} \tag{10.71}$$

Here $\Delta F_p = 1000$ N, which converts to increments of 5 bar pressure. To ensure that the hydraulic piston has time to settle and not oscillate too much, a time restriction of $\Delta t F_p$ of 5 seconds is implemented as case 0. The first condition ensures that whenever the slip ratio between the crawler and a minimum of two wheels is greater than 0.15, the piston pressure increases. The second condition ensures that when the combined torque of all three wheels increases close to the maximum limit of the motors, the piston pressure is reduced. The third condition ensures, when its mounted in a horizontal position, that the crawler always has enough traction to keep its motor running and not falling of the member. This is done by monitoring the allowable friction force that is remaining to be used on each of the wheels. The forth condition is the same as the third except in horizontal, except the motors are not acting in the same plane as gravity therefore, they are omitted. The fifth and last condition checks if the angular velocity of the crawler is below 0.005 rad/s. Which would mean the crawler has come to a stop either because of blockage or other circumstance, since it cant keep up the desire velocity the pressure is reduced. During the controller test, a minimum of 30 bar of pressure is required.

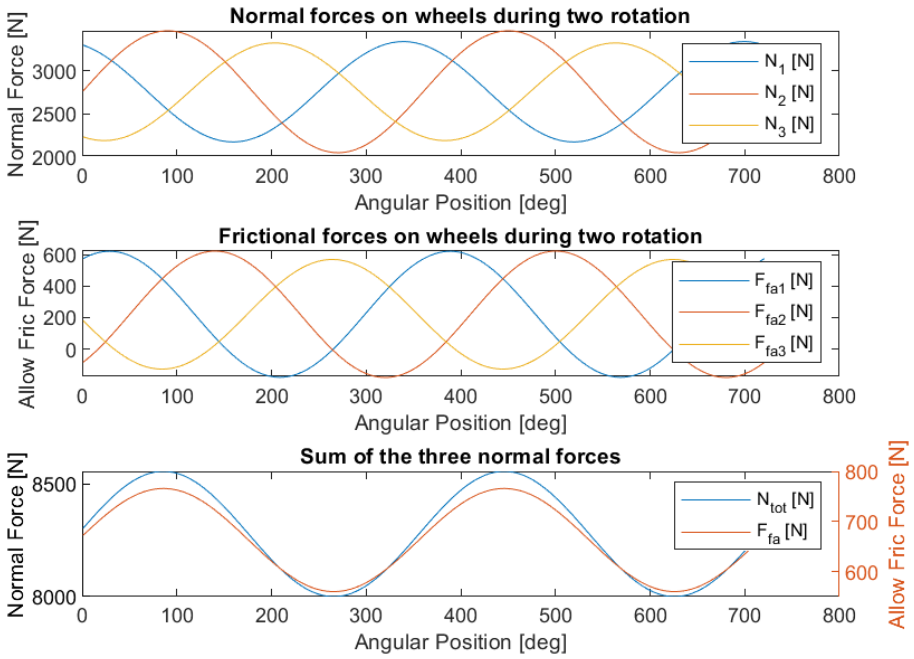


Figure 10.12: Allowable friction force distribution for the three wheels when on an horizontal member in the worst case scenario. Taken at its lowest value grease friction coefficient of 0.37, piston force of 10 kN and 150 Nm of torque on the wheels

The piston provides 10 kN of force in order to maintain the total allowable frictional forces of on average 700 N (F_{fa} in the figure). When the motors are close to fully actuated at 150 Nm and the crawler is hanging on a horizontal member. When examining the individual wheels, there are moments where 2 of the wheels are below the required minimum of 100 N of allowable friction. There are also several instances of the allowable frictional force being less than zero. This indicates that 10 kN of force in the piston is not enough while running the motors at maximum to prevent slip. If the same happened in the vertical setup, it would mean that the crawler would fall down.

10.4.1 Hydraulic actuator

By investigating the dynamics of the hydraulic components seen in section 10.4.1 it is found that the dynamics of the proportional valve are significantly faster than the mechanical components of the crawler. Due to this, the hydraulic dynamics is seen as a static gain, and a proportional control can be used to control the proportional valve. To avoid actuator chatter, a hysteresis band is introduced to the measured feedback pressure. The control structure can be seen in figure 10.13.

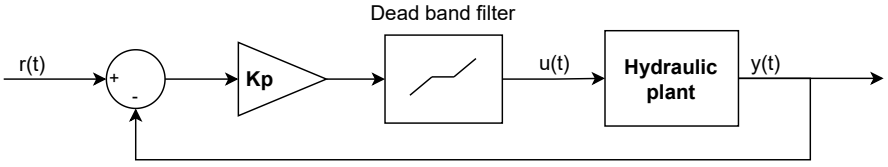


Figure 10.13: Blockdiagram of the control structure used for the hydraulic actuator

The Arduino is programmed to ensure that the boundaries of the valve are not exceeded, these are set between -1 and 1. However, since the amplifer could not deliver as much voltage as assumed, these boundaries are not 100% openings. The amplifier is discussed in section 5.2.3.
The controller is also programmed to take into account the dead zone identified in 5.6, which means that the controller ignores any signal in this zone. 5.6. The dead zone is identified between -0.32 and 0.32 of the given control signal. A hysteresis band with 10% of the actuation value is implemented to prevent chattering. The proportional gain for the controller is designed based on a maximum of 20 bars error. This is because the robot should never change more than 20 bars in a short time period. It is preferable to operate in slower flow regions to limit unwanted noisy signals influencing the control. The proportional gain used is 0.034.

10.4.2 Implementation of the hydraulic controller

The hydraulic controller is implemented in the physical setup and tested. The code for the hydraulic actuator can be found in D.1. During this test it is found that the controller had some steady-state errors. It is found that the assumed voltage output of the operational amplifier is lower than expected. The maximum voltage is assumed to be 13.75 V however the real measured maximum voltage was down to 12.70 V, this is also seen in figure 10.14.

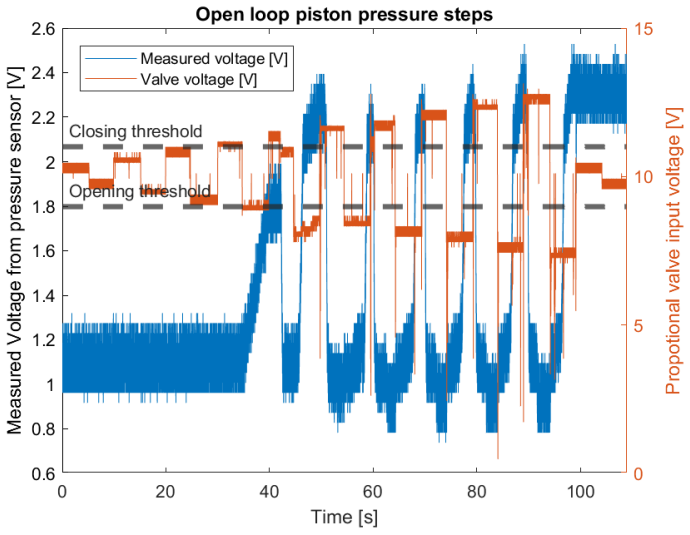


Figure 10.14: Openloop voltage steps with boundaries for opening and closing.

This is caused by losses in the amplifier setup, it is found that the amplifier has an average voltage loss of 1.2V [42]. The initial dead zone is increased as seen in 10.14. The deadzone is increased to range from -0.45 to 0.45.

The hydraulic actuator dynamics is assumed to be fast, as the valve dynamics are only 0.1s from neutral to fully opened according to the data sheet [21]. During testing the crawler arms dettach from the member. This occurs when the pressure spike drops below

-10 bars as seen on figure 10.15. The negative pressure is due to the linear regression fitted onto a voltage signal, and the pressure from the piston letting loose affecting the hydraulic oil in the opposite direction. When performing the open loop test on the piston, there accrues a delay of 2.5s is when going from neutral to open. Which then was 5s when going from fully opened to fully closed. This can be seen in figure 10.15.

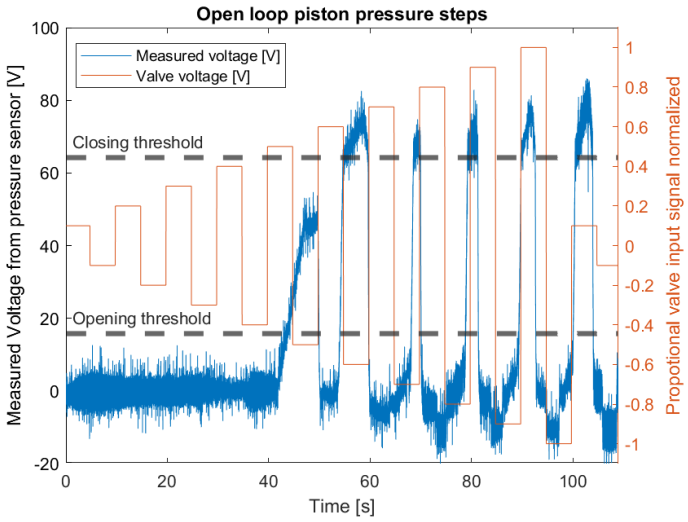


Figure 10.15: Openloop voltage steps to hydraulic valve and its correlating pressure reponse.

The large delay causes instability in the hydraulic controller. It is found that the hydraulic setup is equipped with an accumulator tank. The tank is equipped to balance the pressure, causing a delay in the system. This also explains why the delay increases, as at time 90 seconds the pressure should rise, which is it but that is from the previous signal, and the one this one is responding to first rises at 100 seconds as seen on figure 10.15. This is due to the higher valve opening in these regions putting more oil in the accumulator tank which is limited to 80 bars. This also explains why on the figure the pressure never exceeds 80 bars by more than 3 bars.

The proportional gain is updated to 0.0278 and the results of this gain change in the implemented controller is seen in figure 10.16.

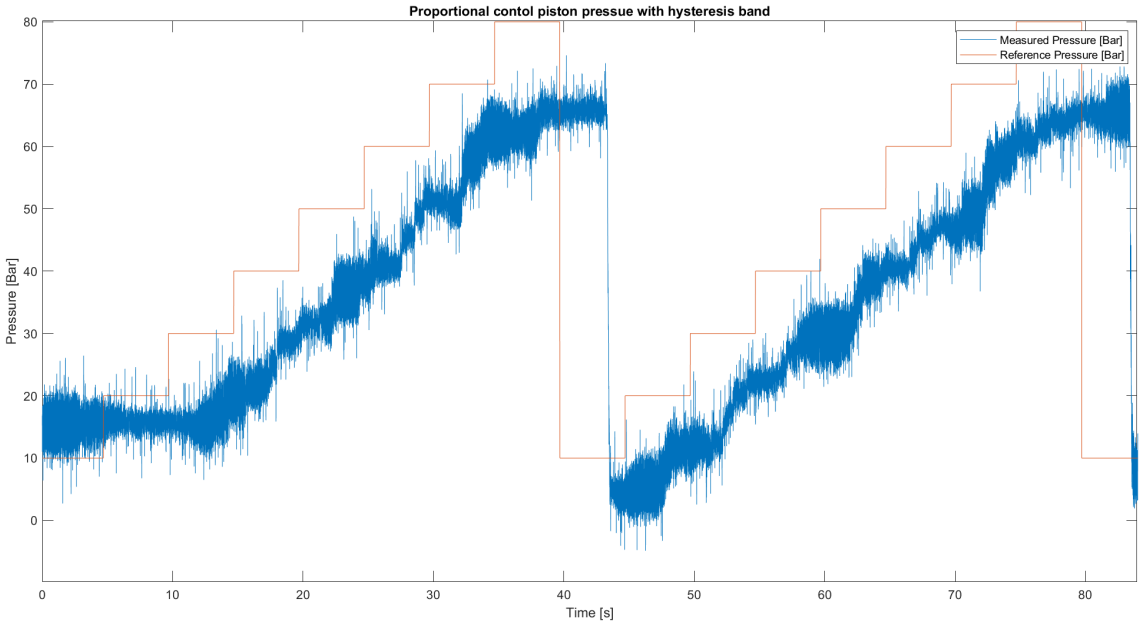


Figure 10.16: Response of hydraulic controller with updated parameters and hysteresis band

From the figure, it is seen that the controller maintains a stable pressure both stepping up and down with 10 bars. It has a steady-state error caused by the hysteresis band of 10% of the actuatable area. The pressure readings fluctuate with the pressure of 9 bar as seen in the figure 10.16, which is the reason why this area is required. The hysteresis band was found to be the best solution to reduce the impact of measurement fluctuations. A moving average filter and several other electrical filter circuits are tested in order to reduce fluctuations. The best solution found is to increase the signal resolution of the Arduino from 256 bits to 4096 and use the dead-band filter.

11 Case Simulations

To identify the controller that performs the best, case studies are performed based on the scenarios described in 10. The cases are tested using the controllers described in chapter 10. For simplicity, these are also listed below.

- IMC tuned PID position controller (IMC)
- LQR tuned velocity and position controller with gain scheduling (LQR GS)
- LQR GS with slip control supervisor (LQR GS SCS)

Furthermore, in some cases, the performance of the three controllers is tested with and without the hydraulic controller. If the hydraulic controller is active, the abbreviation of the controller as "+H" is added.
As seen when designing the controllers in 10, the common goal of all controllers will be to enable the crawler to reach a total distance of 2π, for the velocity controller this is desirable at an average velocity of 0.016 rad/s. The distance of 2π is chosen to validate that the crawler can overcome all external forces and make a complete rotation around the offshore member.

The simulations are performed on the surfaces described in 4.1. However, multiple of these surfaces have identical peak adhesion coefficients as stated in table 8.8. In this case, identical coefficients will be combined in the same test. The wet and dry surface conditions are combined as their variation in adhesion coefficients is not significant. An extreme case has been added to test how robust the controllers are to conditions even worse than what is found in section 8.4. The magnitude of the extreme case coefficient is chosen based on 50% of the size of the adhesion coefficient found from the tests on rough paint. The conditions being tested in the simulation are given in table 11.1.

Table 11.1: Table over the surface conditions with the corresponding peak adhesion coefficients.

Condition	Surfaces	Peak adhesion coefficient
1	Smooth/Rough paint	0.15
2	Rust	0.26
3	Algea	0.18
4	Grease	0.13
5	Extreme case	0.085

The performance of the controllers are identified by simulating the crawler operation in the following scenarios.

- Constant surface conditions
- Changing surface conditions on one wheel
- Changing surface conditions on two wheels
- Changing surface conditions on three wheels
- Encountering of obstacles

The different conditions are chosen since these are the most likely scenario for the crawler to encounter on an offshore operation. As explained in 1.1, the crawler is highly likely to encounter a varying surface when moving around a member. The obstacle condition can be simulated by changing the traction force coefficients, and the increased resistance against crawler movement can be simulated by increasing the rolling friction coefficient. The slip found depicted in the test figures corresponds to the average slip of all three wheels.

11.1 Constant surface during rotation

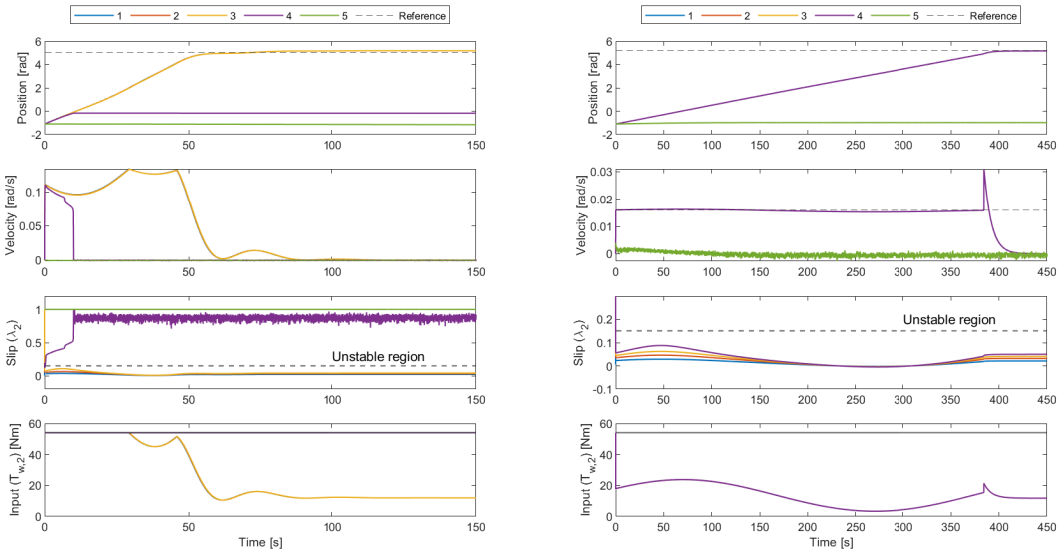
The first test revolves around having different surfaces with which the crawler is operating on, in order to compare the difference in response on various these surfaces. During each test the surface is kept constant, only changing at the subsequent simulation. The three controllers are set to reach a total distance of 2π before 400 s, which correlates with the lower bound speed of 0.016 rad/s. Each controller is simulated on the surfaces from table 11.1 and the results of these tests are compared. For these tests, the hydraulic switching controller is not enabled. A constant pressure of 63 bar is used, as this is within the middle of the operating range of the hydraulic controller. Details about this test are listed in table 11.2.

Table 11.2: Conditions for the baseline test

Type of test	Distance	Time limit	Avg. velocity	Surface condition	Pressure
Baseline	2π rad	400 s	0.016 rad/s	Constant	63 bar

PID and LQR response

The simulation results can be seen in figure 11.1a for the tuned PID controller and in figure 11.1b for the tuned LQR controller. Both figures display four graphs showing the position, velocity, slip and input torque respectively mentioned from top to bottom. For both the PID and the LQR controller, the reference and crawler position are presented, while only the reference velocity is shown for the LQR controller, since the PID controller is only a position controller. The figures show the controller responses for all five surface conditions from table 11.1.



(a) PID tuned controller

(b) LQR tuned controller with gain scheduling

Figure 11.1: Response of PID and LQR tuned controllers on surfaces listed in 11.1.

The baseline simulation data shows a time difference in position and magnitude difference in velocity comparing the PID to the LQR GS controller. Since the PID controller only tracks the position reference it is also assumed that it would obtain the position faster than the LQR GS, since this is also tracking the velocity reference until it is within 5% of the position reference.

The PID controller, shown in figure 11.1a, is able to handle the first three surface conditions and reaches the reference position in 60 seconds, while for the last two surface conditions with the lowest peak adhesion coefficients, the system suffers from a steady-state error with an offset of around 5 radians. When closer observing the slip graph, the

two last surface conditions show a slip behavior reaching into the unstable region. The slip graphs are reaching the value one, which means that the wheels are slipping and that the slipping is responsible for the offset in the crawler position. As described in section 7.2.1, slip above its stable regions means that not enough traction is available, and it will tend toward total slippage with a value of 1. When all wheels are slipping, no traction is transferred to the crawler resulting in it not gaining any speed and stopping, as seen in figure 11.1a for the PID controller.

The crawler velocity is not a controllable variable in the PID controller case. For the three surface conditions, the velocity can be observed to reach the maximum possible velocity of 0.16 rad/s at around 35 seconds and then falling to 0 rad/s when approaching the reference position. The maximum velocity is related to the maximum input torque as seen in the bottom of figure 11.1a. For the other two surfaces, a rapid decrease in velocity to 0 rad/s can be seen at the time the crawler wheels start to slip. The input torque reaches its maximum allowed torque of 54 Nm in the simulation start for all surface conditions. For the three manageable surface conditions, the torque declines at a position of 3 radians and increases at 4 radians. This is due to the feedforward term, where gravity now aids its movement, reducing the need for input torque. When reaching the reference position at around 60 seconds the torque declines and settles at around 12 Nm. Due to slip occurring for the last two surface conditions, the input torque reaches the maximum allowed torque for the entire simulation run.

The simulation data from the LQR controller in figure 11.1b is a position and velocity controller with gain scheduling described in section 10.2.
Compared to the PID controller, the LQR controller reaches the reference position at 384 seconds, since the velocity amounts to the lower bound of the desired velocity of 0.016 rad/s. The time limit for reaching a distance of 2π is 400 seconds. In return, the LQR controller was able to reach the reference position for all surface conditions except the extreme case.

For the velocity control, the LQR reaches the reference velocity from the simulation start and follows it until the crawler reaches the reference position. Right before settling at the reference, the velocity peaks, caused by the gain scheduling switching the velocity controller to the position controller when being 5% away from the reference position. The crawler velocity for extreme surface conditions can be seen to increase to the desired velocity for 15 seconds before dropping down to 0 rad/s.
Looking at the slip graph, all surface conditions of the wheel slip are in the stable region, except of the extreme case. which means that the crawler wheels have traction on the surface. It is visible in the slip graph to see the impact that driving on a worse surface has on the slippage. As the worse the surface condition is the more slip it has, since it needs to get higher up the traction curve to gain the same amount of traction as seen previously in figure 8.7. The extreme case, could not gain enough traction and therefore went into the unstable region with full slip of 1.

LQR SCS response

Since the only LQR controller controls both the position and velocity, as required by SubC Partners in section 2.3, it is combined with the slip controller SCS from section 10.3.5. The PID purpose is non-acurate cleaning objectives by SubC Partners but does not fulfill the requirements. However, it is useful to get to the correct starting location. The PID will not be observed with an additional SCS.
The simulation data for the LQR GS SCS can be seen in figure 11.2 and has the same build-up as the previous simulation figures 11.1a and 11.1b.

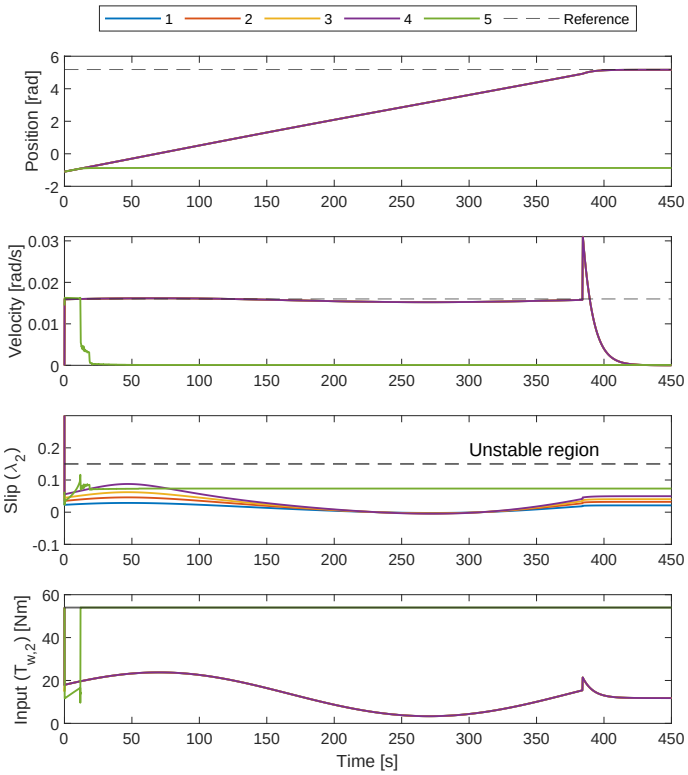


Figure 11.2: Response of LQR GS SCS controller on surfaces listed in 11.1

When examining the figure 11.2, the response for the first four surface conditions remains the same. This is because the SCS is not required, as previously seen the slip is below the unstable region and no sudden change in slip would cause the controller to activate 10.70. The only difference is within the extreme case. Where the SCS is able to ensure that the crawler can operate on the extreme surface for a limited time period. Keeping the same velocity as the other cases, which it was not able to do without it. Though it fails after 25 seconds on the extreme surface. Here, it recovers it again at a velocity of 0.04 rad/s until it slowly decreased and falls to a standstill. At which point the slip controller is turned off and the LQR controller demands maximum torque. The slippage from the extreme case for all the wheels are depicted in figure 11.3.

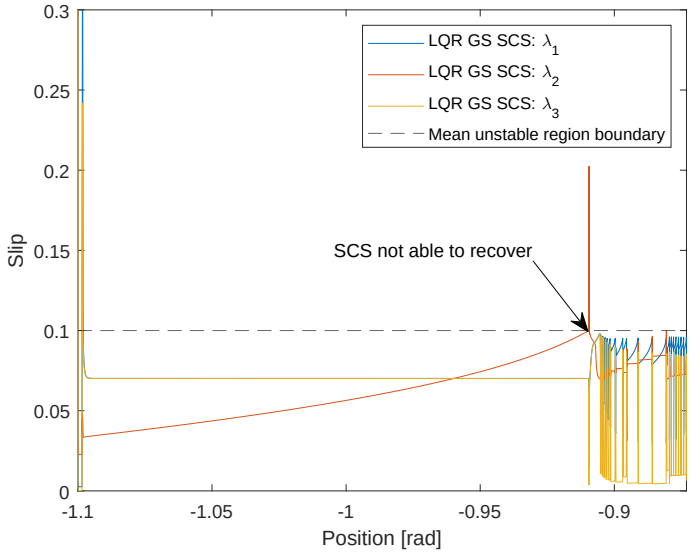


Figure 11.3: LQR GS SCS on extreme surface

When following the three wheels behavior controlled by the SCS, a recovery of the slip can be seen in the beginning of the simulation. Since the wheels are starting up from a stand still the slip starts at 0 but increase to 1 for all wheels. This is due to the fact the crawler is standing still, and the motors have to move first before the crawler can. The increase in slippage causes the SCS to activate on the three wheels. Wheel 2 is taken below the threshold of the slip controller after the initial response. while wheel one and three are controlled by the slip controller at the reference slip of 0.07. The second wheel slippage increases as the torque need to counteract gravity rises, causing a greater demand of torque from the motors. Which results in more traction required. When wheel two reaches the unstable slip boundary at a position of -0.92, the slip controller is activated for it as well. When it is initialized, a spike up until 0.2 in slip is observed. The slip controller will now decelerate the wheel to reduce slippage, which causes the other two wheels to compensate for the loss of traction, as wheel 2 now is lower on the traction curve. The reduction of traction of wheel 2 results in wheel 1 and 3 to pick up the required amount, causing their slip to increase. The slip controller then slows down wheel one and three, repeating this slope scenario for wheel two. An oscillation is seen and the slip controller is not able to recover. So there is not enough traction available, to complete the full rotation. The slip controller keeps all the wheel slippage within the stable region but there is not enough traction to accelerate the crawler.
A detailed documentation of all three controller simulation and their comparison is shown in table 11.3 below.

Table 11.3: Comparison of performance of the PID, LQR GS and LQR GS SCS controllers at one full rotation on various surfaces described in 4.1.

PID					
Surface	1	2	3	4	5
Reach pos [s]	57	57	57	-	-
Avg. vel [rad/s]	0.067	0.071	0.069	0	0
Time in unstable region [%]	0	0	0	93	100
Avg. slip 1	0.033	0.036	0.041	0.934	1.000
Avg. slip 2	0.060	0.079	0.098	0.839	1.000
Avg. slip 3	0.032	0.033	0.038	0.601	0.590
LQR GS					
Surface	1	2	3	4	5
Reach pos [s]	384	384	384	384	-
Avg. vel [rad/s]	0.016	0.016	0.016	0.016	0
Time in unstable region [%]	0	0	0	0	100
Avg. slip 1	0.018	0.022	0.027	0.032	0.989
Avg. slip 2	0.040	0.058	0.072	0.089	0.885
Avg. slip 3	0.016	0.020	0.024	0.029	0.569
LQR GS SCS					
Surface	1	2	3	4	5
Reach pos [s]	390	390	390	390	-
Avg. vel [rad/s]	0.016	0.016	0.016	0.016	-
Time in unstable region [%]	0	0	0	0	93
Avg. slip 1	0.018	0.022	0.027	0.032	0.085
Avg. slip 2	0.040	0.058	0.072	0.089	0.070
Avg. slip 3	0.016	0.020	0.024	0.029	0.024

From the table it is seen that wheel 3 has a lower avg slip in all test compared to wheel 1 and 2. This is a consequence of the traction curve for wheel 3 having the greatest slope. So it can get the same amount of traction at 0.05 as wheel 2 can at 0.15 8.6. The same applies for wheel 1 over wheel 2 having its peak at 0.1.
From the table it is seen that in the extreme case the average slip falls from 0.885 to 0.07 so it does its purpose at keeping the slip within its bounds, it is just not designed for restarting the crawler once it reached its standstill.
The increase in time for it to reach its position from 384 s without SCS to 390 s, is caused by the initial spikes when the crawler is starting up. This can be seen in the torque figures

of 11.1b and 11.2. Where the LQR controller makes the crawler reach its reference velocity by giving maximum torque, the SCS reduces it down to 20 Nm, when its enabled. The PID has a higher slippage as it accelerates and decelerates more compared to the LQR. For the extreme case the controllers are en the unstable region 100, 100, and 93 % of the simulation time for the PID, LQR GS, and LQR GS SCS respectively. Since the controllers are not able to recover from the slip in this case, the crawler is not moving.
When comparing the controllers, a big difference in slip is observed between the PID and the LQR controllers. Here the PID has much more slip on all surface conditions.

11.2 Change of surface during rotation

This case is to determine how the controllers perform when the wheels are not driving on different surfaces through the rotation. To cases,,ate the different surfaces, the peak adhesion coefficients found in table 11.1 are used. All three wheels are initialized as running on a rust surface, which offers the highest traction force. The surfaces are introduced at varying locations around the pipe as seen in figure 11.4.

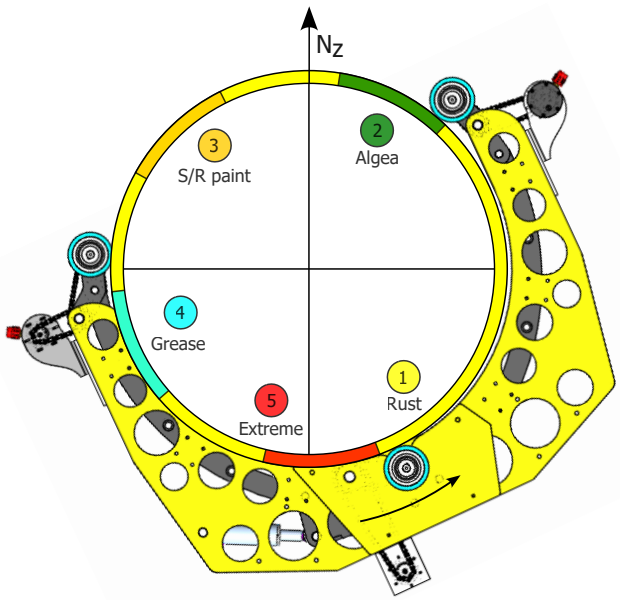


Figure 11.4: Locations on the member where the change in surface occurs

The crawler position is initialized at its resting position of -1.1 radians, and the initial surface is rust. As seen in figure 11.4 the surface changes between the original surface and one with a lower coefficient than before. This is performed throughout an entire rotation. The surfaces and their locations are found in table 11.4.

Table 11.4: Types of surfaces and locations

Surface	Peak adhesion coeff.	Start pos [rad]	Stop pos [rad]	Type of case
1	0.26	in between		Rust
2	0.18	1.2	1.8	Algea
3	0.15	2.4	3.00	Paint (s/r)
4	0.13	3.6	4.20	Grease
5	0.017	4.8	5.40	Extreme

The following tests are performed to investigate the performance of the three controllers.

1. Changing surface one wheel
2. Changing surface two wheels
3. Changing surface all wheels
4. Varying surface across all wheels

The tests are performed with rust being the initial surface, as this surface offers the highest traction. The tests are performed with an initial pressure of 63 bar and the crawler is initialized at -1.1 rad. For the changing surface on wheels the surfaces are initialized accordingly to 11.4 when the crawler reaches this position, for the varying surface tests the surface is changed when each wheel reaches a position according to the table. The wheel slip determined in the figures corresponds to the average wheel slip of all wheels. The results of the tests are shown below.

11.2.1 PID

For the PID controller cases, the simulation outputs are depicted in figure 11.5 comparing the controller response for varying surface conditions and number of wheels impacted at once, described in section 11.2.

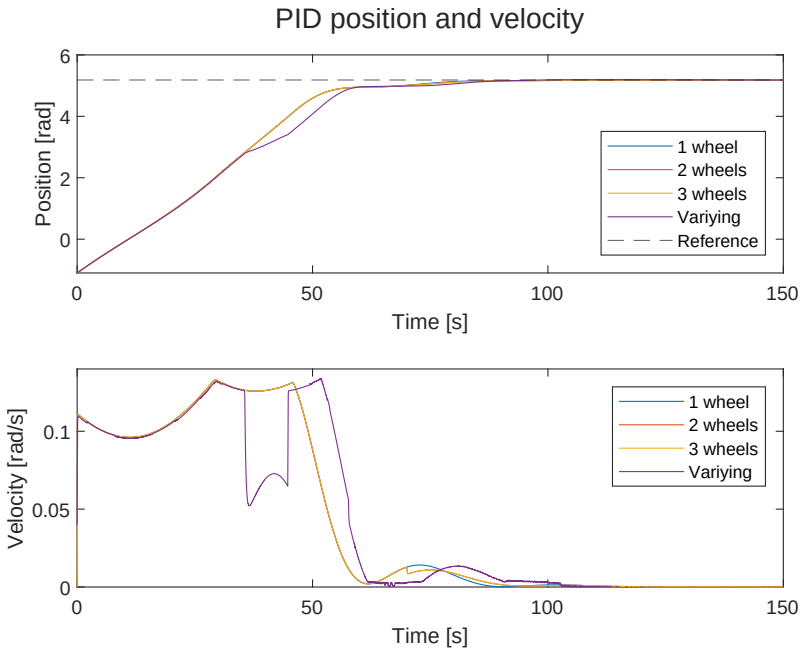


Figure 11.5: Simulation results for the PID controller case showing the crawler position and velocity.

As observed in the figure above, showing the crawler position over time, the first three cases have similar responses. For all three cases, the reference position, shown as the black dashed line, all test results in position settling at around 57 seconds. The case having varying surface conditions on all three wheels shows a drop in velocity when the crawler reaches 2.8 radians, this is due to the fact that the first wheel actually passes the surface at this point, 120° earlier than compared to the other three tests. The result is a wheel which is not contributing as its slipping. This shows that the crawler can operate on the extreme surface and that the crawler only requires two wheels to be on good surfaces. It swiftly recovers once the third wheel gets off the extreme surface again. This explains why it takes longer to reach its steady state position compared to the first 3. The PID is able to drive on the extreme surface at the end, since the demanded torque is greatly reduced compared to the baseline, as it no longer needs to accelerate on it.
The PID controller enables the crawler to drive through the extreme surface, this was not possible for the baseline test, however in this case, the crawler is already in motion, meaning the acceleration phase is much smaller, reducing the possibility of the wheels starting to slip.
Observation of the slip over the crawler position shown in figure 11.6 describes the changing response in the position and velocity mentioned above. The changes in the surfaces for all four cases are represented as different colors on the x-axis. The colors and their represented surface are discussed and depicted in figure 11.4. The positions at which the surface changes can be seen in table 11.4. Furthermore, since the crawler wheels are positioned ± 2.09 radians from each other, there will be different surfaces across the wheels at the same time. This way, when wheel one is introduced to the extreme case, wheel three is introduced to the algae condition.

The slip for the different cases is close to 0 slip up to a position of 2.8 radians, where the varying case increases to a slip of 0.2, which is outside the stable region. This is due to the third wheel slipping completely going from rust to being on the extreme case. The position and velocity response happening at the same location are discussed previously and the slip response proves that for the varying surface case, even if one wheel slips, the crawler is able to move past that point. The slip drops again when changing back to the rust surface.

When the wheels enter the extreme condition, an increase of slip for all cases except the one where only 1 wheel is affected. As seen the crawler can operate with 1 wheel being on the worse surface, as the other 2 wheel provide enough traction on the rust to keep it going, at these low torque near the end. The other cases all cause the wheel to go into the unstable region where the wheels are not able to get enough traction. The crawlers inertia keeps it going making it able to reach the desired destination, where the motors can decelerate it to its final position.

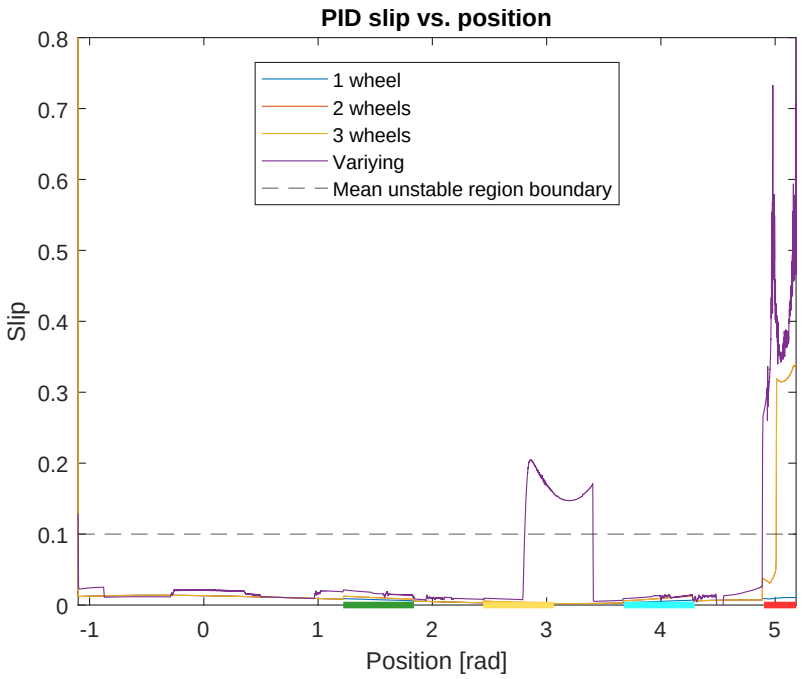


Figure 11.6: Slip response for the PID controller case, showing the surface conditions for the different positions.

Table 11.5 summarizes all test cases for the PID controller. From the table it is clearly seen that having 2 or 3 wheels does not change the test results, as they both can not handle the extreme suface. Both spending 53.31 % of their time in the unstable region. This large value is due to the time it take it reach its final destination on the extreme surface, as seen in figure 11.5. Only the varying surface has more time in unstable region due to the third wheel entering the extreme case earlier. The average traction force for all the wheels combined a similar in the cases except varying, as it slips more and resulting in it requiring more traction compared to the others which were in the stable region.

Table 11.5: Table over the PID controller case results, including a mean value for all observed parameters.

	PID				
Test	1 wheel	2 wheels	3 wheels	Varying surface	Mean
SSE [%]	0	0.01	0.01	0.03	0.01
Settling time [s]	57.04	57.04	57.04	58.64	57.44
Time in unstable region [%]	0	53.31	53.31	63.74	42.59
std. from ref speed [rad/s]	0.059	0.059	0.059	0.057	0.058
Avg. traction force [N]	190.82	190.51	190.51	234.49	201.58

11.2.2 LQR GS

Following the PID case test, the second controller case is tested. For this case, the LQR GS controller is tested for the same cases as the PID controller. The cases are the introduction of different surface conditions for one, two, and three wheels and a varying surface across the wheels. The simulation output can be seen in figure 11.7 showing the crawler position over time such as the crawler velocity over time. It is observed that for both the crawler position and crawler velocity, all cases have the same response when using the LQR GS controller. When closing in on the reference position with a difference of 5 %, the position controller switches to the velocity controller, which can be seen as a peak in the velocity at the same simulation time.

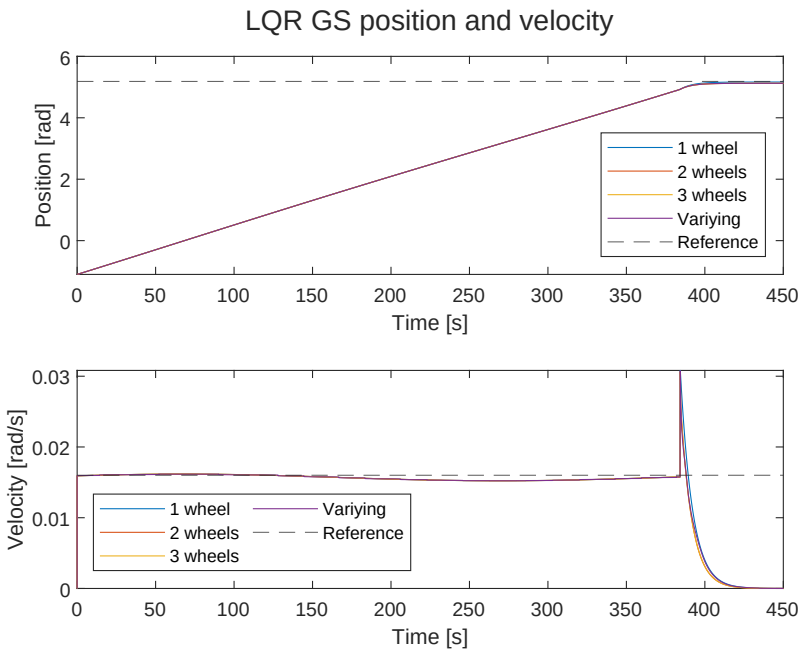


Figure 11.7: LQR GS controller simulation outputs showing the position controller and the velocity controller above.

When looking at the slip over position graph in figure 11.8, it is seen that the LQR GS controller is good in handling changing surface conditions, presented in figure 11.4, for

nearly all cases. The only surface the controller is not capable of handling is the extreme surface condition, which acts as a control surface, used to see if the controller is working. The controller only handles all surfaces when introduced to only one crawler wheel.

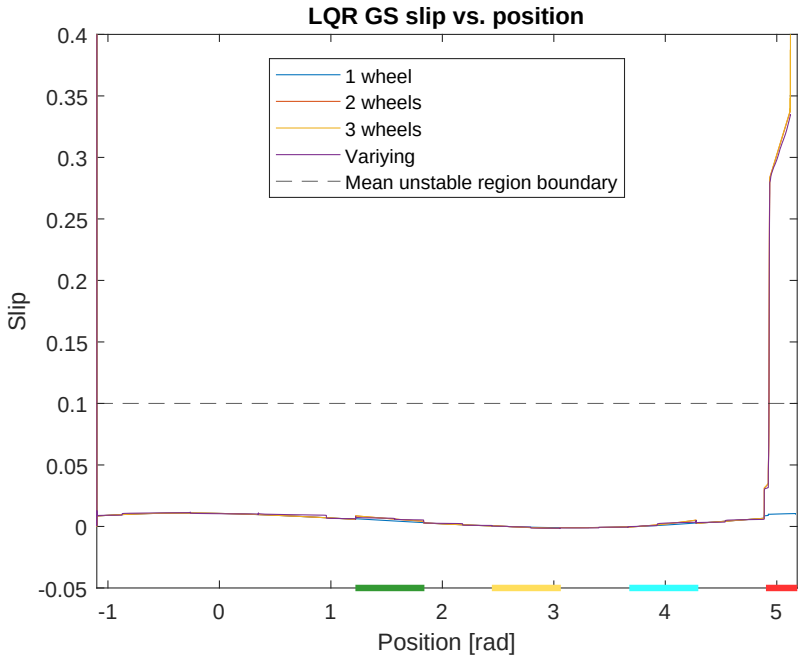


Figure 11.8: Slip over the crawler position for the LQR GS controller simulation.

The results of the LQR GS case are shown in table 11.6. Where, just like the PID having 2 or 3 wheels on the worse surfaces, results at the same time in unstable region. The varying surface has the same value with the LQR compared to the PID. The lower time spent in unstable region is due to the crawler moving slower, therefore spending relative less time on the extreme case compared to the PID. The average traction force is similar in size as the slip never rose above 0.01 except in the extreme case.

Table 11.6: Table over the LQR GS controller case results, including a mean value for all observed parameters.

	LQR GS				
Test	1 wheel	2 wheels	3 wheels	Varying surface	Mean
SSE [%]	0.36	1.17	1.17	1.17	0.97
Settling time [s]	384.21	384.21	384.21	384.22	384.21
Time in unstable region [%]	0	14.67	14.67	14.67	11.00
Variation from ref speed	0.015	0.015	0.015	0.015	0.015
Avg. traction force [N]	160.86	158.51	159.71	155.27	158.59

Hydraulic controller addition

To test whether the hydraulic controller will have an impact on the previous results, all tests are performed with the controller enabled. It is found that only tests with the PID controller enter conditions where the hydraulic controller is activated. This simulation test is performed to assess the impact that the hydraulic controller has on the crawler making its rotation. The simulation was done on one wheel hitting the different surface conditions as previously discussed. During the test, only the PID controller activated the hydraulic controller, the LQR never required it to do the test with only 1 wheel. The result for the PID controller is seen in figure 11.9.
11.9.

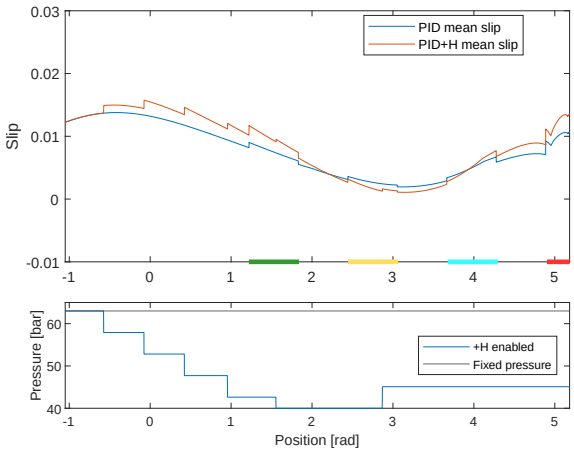


Figure 11.9: Simulation output for the PID test case with combined hydraulic controller.

The figure shows two different graphs, one depicting the slip on the crawler position and the other the hydraulic pressure over the crawler position. When looking closer into the slip graph, the blue line shows the mean slip behavior when using a PID controller while the red line is the PID controller combined with the hydraulic controller. The colors in the x-axis show the different surface conditions to see the slip response when introduced to those.

When looking deeper into the hydraulic pressure behavior, it is noticeable that the pressure steps down as the crawler position changes. This is the case since the logic condition two from section 10.4 equation 10.71 is activated in order to step the pressure down. The reason for the second condition activating is that the PID controller uses full torque from all three motors for more than five seconds. More precisely, the PID controller demands the maximum torque until it is closer to reaching the reference position. Since the PID demands maximum torque, condition two is activated until the allowed minimum piston pressure of 40 bar is reached. When being introduced to the next surface condition in yellow at a position of 2.9 radians, the pressure increases to 46 bar. For an increase in pressure to occur, logic condition three from section 10.4 equation 10.71 is activated. To increase the friction force available in the system again, so that the crawler can continue running its motors at maximum torque without risk of slippage.

The pressure change is observed in the slip graph right above, showing a response for each step that the logic controller initiates. The blue line shows the slip of the PID controller, while the red line shows the slip of the PID controller combined with the hydraulic controller. For each pressure drop the response is noticeable in the red slip line, since the slip is related to the normal force of the wheels. As shown previously in section 7.2.2 in equation 7.13, the slip increases when the the normal force is decreasing, which is the case when lowering the piston pressure. The red line is observed to step up for each pressure drop happening and steps down, at the increasing pressure at a position of 2.8 radians.

11.2.3 LQR GS SCS

The next test is to see the influence the the slip controller (SCS) has on the LQR GS. The simulation for testing the different wheels on surfaces, conducted as in the previous case tests, is shown in figure 11.10 below.

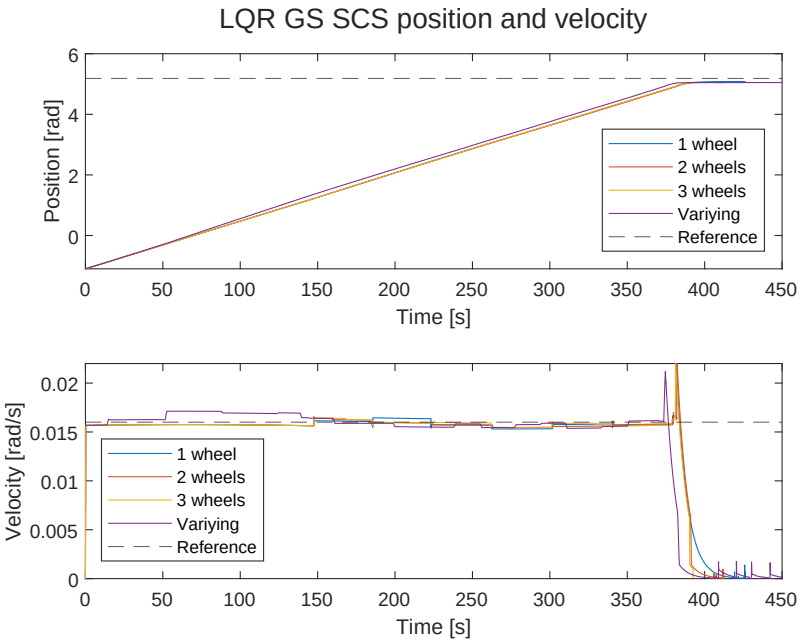


Figure 11.10: LQR GS SCS controller simulation outputs showing the position controller and the velocity controller.

Looking at the position graph of the controller test, a difference can be observed from the previous LQR GS. For this test, the case of varying the surface across the wheels has a slight deviation from the rest of the tests. The deviation leads to a slightly faster response for reaching the reference position from around 382 seconds of settling time to 374 seconds. All other three tests are very close with settling times between 381 and 382 seconds. The slight change from the previous controller test is caused by the SCS added to the LQR GS controller.

A more visible change in controller behavior is seen in the crawler velocity graph in figure 11.10. Already at a position of 0 radians, a steady-state offset in velocity from the reference is observed. This is due to the SCS activation when starting up, since during acceleration phase slip is present. The SCS keeps outputting its previous found torque to stop it from slipping. The LQR does not have an integral part to offset this constant added torque from the SCS causing the steady-state error. This causes it to keep this steady-state error until it reaches the next surface, where this slip controller activates again, but this time causing a positive steady-state error afterwards. The varying surface has more changes in surface, and therefore the slip controller activates more.

Since not all the wheels are driving on the same surface, it can lead to bigger variations since the crawler is still being propelled forward from the two other motors, and their slip controllers are not active, resulting in the motor on the worse surface being more aggressive to try and correct the situation. But the Crawler stays stable, when reaching its final position, spikes are seen as the velocity is so low. The low velocity causes small changes in motor speed to cause slip. These tests resulted in updating the 10.70 for implementation onto the physical system with a minimum crawler velocity. Which was not on during simulations.
The slip for the different crawler positions with the changes of surfaces is depicted in figure 11.11.

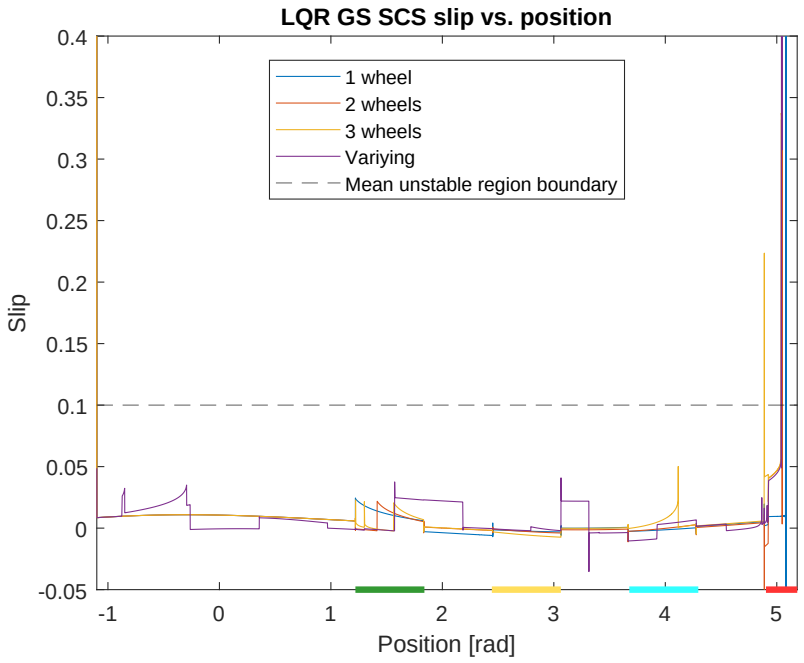


Figure 11.11: Slip response for the LQR GS SCS controller case, showing the surface conditions for the different positions.

In comparison to the LQR GS controller without the SCS, this controller has a much larger variation in slip. All the cases stay within the slip boundary except for the case with two wheels on the varying surface. Here it is seen that the controller is not capable of han-

dling the slip occurring in all cases, leading to the slip responses peaking, at the position the cases meet the extreme surface case. When looking at the overall slip behavior for all cases, it is observed that all cases show noticeable own slip behaviors, which has not been the case for the LQR GS controller.

Here, the case with varying surface condition across the wheels shows the largest deviation from the other cases. The most noticeable cases are located at a crawler position of -1 to 0.9 radians, a position of 1.5 to 2.4 radians, and a position of 3 to 3.5 radians. The impact of the SCS is clearly seen, it detects that there is a change in surface condition, it activates and tries to get the slip ratio up to its desired value of 0.07. But gets turned off before it manages to do it, since there is still enough traction on the surface. Except for the last extreme condition where there is too little for it to stabilize.
For the other cases, a distinct behavior is observed at a crawler position of 1.2 to 1.62 radians. Here, the first surface condition is introduced to the cases, and a larger slip is reached. This slip reduces to zero after the surface change. This shows a realistic behavior, since the introduced algae surface has a lower peak adhesion coefficient.

The performance of the LQR GS SCS controller is documented in table 11.7 below.

Table 11.7: Table over the LQR GS SCS controller case results with a mean value for all observed parameters.

	LQR GS SCS				
Test	1 wheel	2 wheels	3 wheels	Varying surface	Mean
SSE [%]	1.95	2.58	2.71	2.59	2.46
Settling time [s]	381.43	382.00	381.07	374.08	379.64
Time in unstable region [%]	0.02	5.15	0.61	14.85	5.16
Std. from ref. vel. [rad/s]	0.014	0.014	0.009	0.011	0.012
Avg. traction force [N]	134.57	132.62	131.19	136.30	133.67

From the table it is seen that the time in unstable region is highest for two and varying surface. The reason why its higher for two wheels is that the crawler can do clear all the case including the extreme case with only two wheels being on a good surface, so 1 wheel is redundant. But when two wheels enter the worse surface, the last wheel can compensate and they start slipping. The crawlers disturbance observer detects this change and tries to correct but since one wheel is still on good surface, it does not get the response it expected, and fails to recover. The same happens to the one with varying surface. But when all the wheels are on the same extreme surface it can recover since now the crawler gets the response it expected.

11.3 Obstacle encounter

To test the performance of the three cocapacityombined with the hydraulic controller, simulations are performed to identify the impact of adding the hydraulic controller, to make the crawler overcome obstacles. Similarly to 11.2, obstacles are introduced in various positions around the member as seen in figure 11.12.

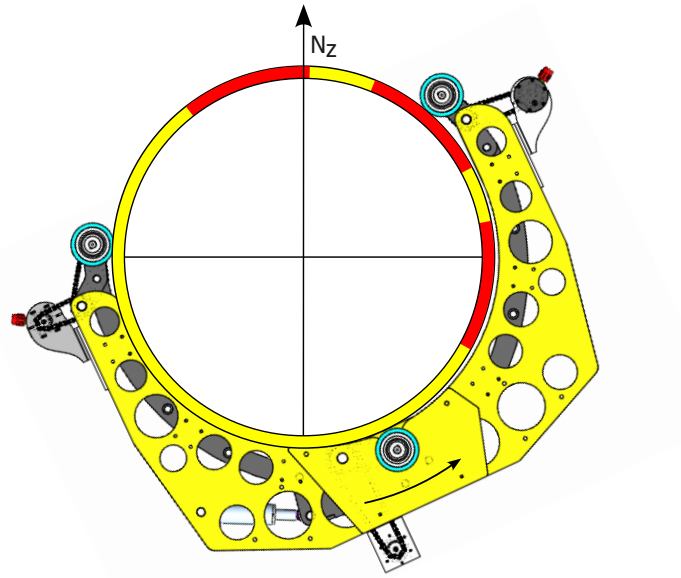


Figure 11.12: Location of obstacles at the test member

The crawler is set to reach a total distance of 2π, the pressure is initialized at 63 bar, and the surface is set to condition 1 (rust). The locations are found at:

- -0.45 to 0.20 rad
- 0.50 to 1.14 rad
- 1.56 to 2.08

To simulate the resistance, the rolling friction is increased by four, this increase is shown in each figure as red bars. For each test, the velocity, the torque, and the piston pressure is shown.

11.3.1 PID+H obstacle

The first controller examined for the test is the PID controller observed previously in section 11.2.1. The test simulation is shown in figure 11.13.

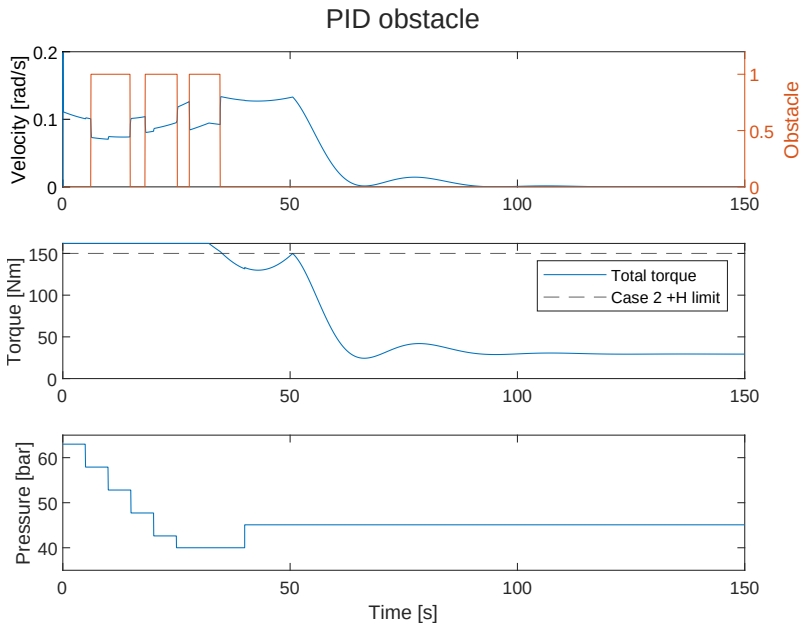


Figure 11.13: Wheel obstacle simulation for the PID+H controller.

The velocity behavior is observed directly when introduced to the obstacle, shown as red steps. When meeting the first wheel obstacle, the velocity shows a clear decrease of 0.4 rad/s. The PID controller is able to manage to continue holding the velocity at 0.65 rad/s. The velocity increases to the start velocity after passing the first obstacle, but drops to 0.65 rad/s again after reaching the second obstacle. While passing the obstacle a slight increase in velocity is observed. The velocity increases from 0.85 rad/s to 1.3 rad/s from when the second obstacle ends to when the third and last obstacle starts. The velocity drops again for the third obstacle. The end velocity is 1.45 rad/s. These increases in velocity are due to gravity. At the first obstacle the effect of gravity is largest and decrease further along. Another factor for the increase in velocity is the pressure decreases as seen in last graph of the figure 11.13.
The torque can be seen to work on full capacity until the last obstacle is overcome. Thereafter the torque response performs a little curve peaking at 150 Nm, where after it then decreases to around 30 Nm. The black dashed line indicates the logic control limit for the second case presented in equation 10.71 section 10.4. Since the torque limit is 150 Nm and the actual torque works on the maximum possible with 154 Nm, the second case applies. The pressure drop initiated by the second logic controller case is seen to drop stepwise from 63 bar to 40 bar. The steps happen every five seconds. Since 40 bar is the lowest allowed pressure, the pressure remains at that value. Until it increases the pressure one time due to slip at 40 seconds

11.3.2 LQR GS+H obstacle

For the second obstacle test, the LQR GS controller is examined. The simulation output is seen in figure 11.14 below.

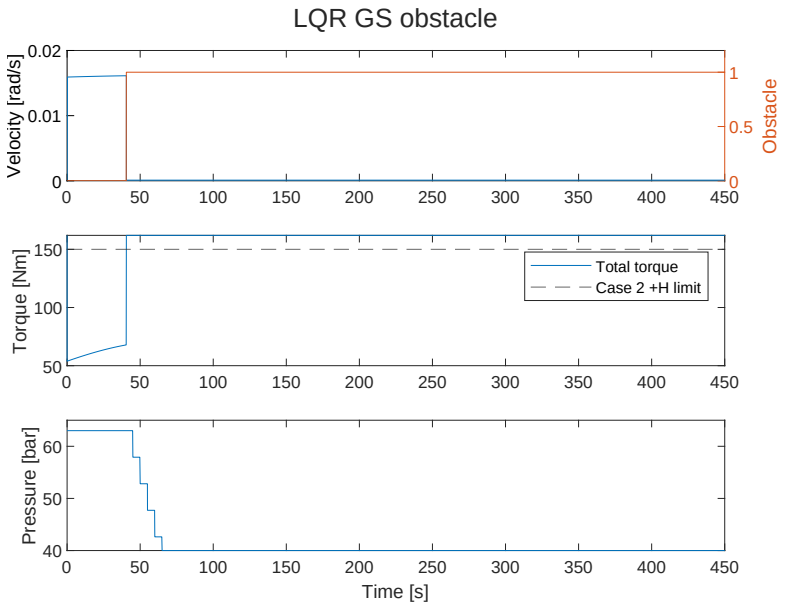


Figure 11.14: Wheel blocking test simulation for the LQR GS+H controller.

From the simulation, it is seen that the LQR GS controller can not handle the obstacle. When it reaches the obstacle, the motors cannot continue forward and start to slip as the crawler velocity goes to zero. As a result of this the desired torque goes to its maximum, and it is never able to recover since it is now on the unstable side of the traction curve. The pressure is reduced since the torque is maxed out to try and reduce the required traction force required to get over the hurdle. Since the rolling friction depends on the normal force delivered from the piston. This has no effect as the motor is still slipping, instead the motor should turn off when its fully slipping. To get back on the stable side of the traction curve and try again with the lower rolling friction after the pressure has been reduced.

11.3.3 LQR GS SCS+H obstacle

The third obstacle test is using the LQR GS SCS controller, which is depicted in figure 11.15.

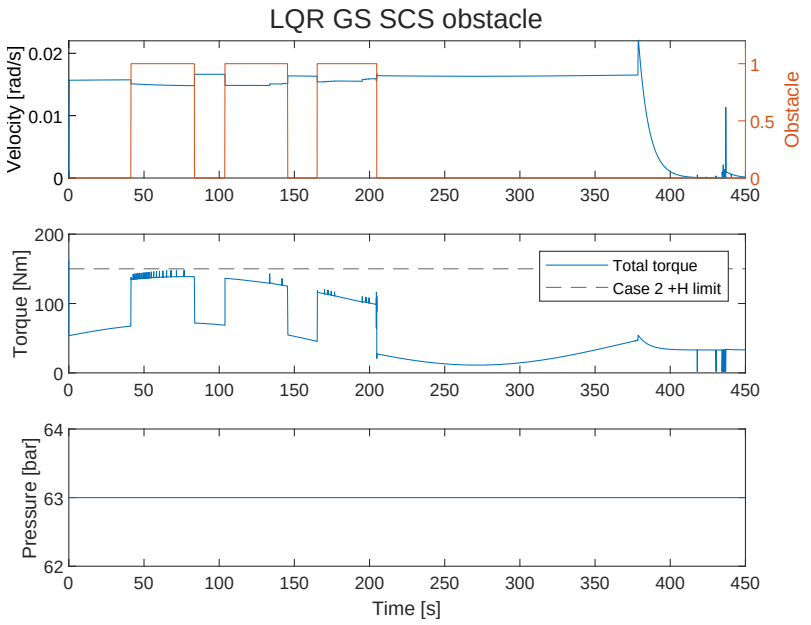


Figure 11.15: Wheel blocking test simulation for the LQR GS SCS+H controller.

From the figure an improvement is seen compared to the case without SCS. The controller here is able to get over the obstacle. This is due to the SCS disturbance observer discovering a change in traction due to the increase of rolling friction. When the slip starts increasing, the SCS is enabled and reduces the amount of torque to only 145 Nm compared the maxed out version at 154 Nm for the LQR only controller. The influence of the slip controller can be seen in the spikes, while in the first obstacle in its torques. The velocity decreases when it is over the obstacle, due to increased resistance. The LQR keeps the crawler at a steady velocity, though with a steady state error due to the supervising slip controller. The change from velocity to position controller is smoothed out because of the SCS at 370 seconds. Preventing the spike in torque that otherwise would be delivered. The hydraulic controller never gets activated through the simulation, as the SCS keeps it stable.

11.4 Comparison

To get an overview of how the controllers performed between each other the results for changing surface and obstacle are compared.

Changing surface comparison

A table is made to compare the results between the three controllers on the changing surfaces. 11.8.
The table compares the mean values of all three controllers.

Table 11.8: Comparison between all three controllers on a changing surface

Controller	PID	LQR GS	LQR GS SCS
Steady state error [%]	0.01	0.97	2.46
Settling time [s]	57.44	384.21	379.64
Time in unstable region [%]	42.59	11.00	5.16
std. from ref. velocity [rad/s]	0.058	0.015	0.012
Avg. traction force [N]	201.58	158.59	133.67

It is clear to see that the PID controller has the lowest steady-state error of 0.01% while the LQR controller alternatives hold values of 0.97% for the LQR GS controller and 2.46% for the LQR GS SCS controller. Also for the settling time the PID shows a much more aggressive behavior than its LQR counterparts. While the PID reaches the reference position after 57.44 seconds,the LQR GS SCS takes 379.64 seconds.

For the other comparisons, the LQR controllers show they are more effective by limiting the time the crawler wheel slips are in the unstable region. While the PID controller allows the slip to be unstable for 42.59% of the simulation time, the LQR GS amounts to 11% and the LQR GS SCS to 5.26%, resulting in a significant improvement in robustness regarding the slip.

Since the LQR controllers are both position and velocity controllers, they show a smaller standard deviation from the reference velocity with 0.012 to 0.015 rad/s while the PID has a standard deviation of 0.058 rad/s. The traction force is higher for the PID controller case than for the two LQR controllers, because the PID controller being the only controller which was combined with the hydraulic controller. The hydraulic controller changes the piston force and, thereby, the crawler clamping force.

Obstacle comparison

The simulation results are summarized in table 11.9 below, comparing the results of all three controllers for the obstacle test.

Table 11.9: Comparison between the results of tested PID+H controller, the LQR GS+H controller and the LQR GS SCS+H controller with obstacle.

Controller	PID+H	LQR GS+H	LQR GS SCS+H
Steady state error [%]	0	86.57	1.96
Settling time [s]	61.34	-	375.24
Time in unstable region [%]	0	91.22	0.20
std. from ref. velocity [rad/s]	0.057	0.016	0.014
Avg. traction force [N]	184.50	220.50	239.28

From the obstacle test, both the PID+H and the LQR GS SCS+H controller managed to reach the final position. The LQR GS SCS+H had a steady-state error of 1.96 caused by the SCS. Without the SCS the LQR controller had a steady-state error of 86.57%, which is where it encountered the first obstacle. The impact of the SCS can also be seen in the standard deviation from the reference velocity, being 0.014 rad/s compared to the LQR only of 0.016 rad/s. This is caused by the extra torque the slip controller continuously provides even when no longer active.

The average traction force is highest for the LQR GS SCS+H as it keeps the slip in the more optimal region compared to the other controllers, with a value of 239.28 N compared to the 220.50 N for LQR only. The addition of SCS caused the time in unstable region to go from 91.22% to 0.20%. While the PID had 0% in unstable region, as its torque never varied in the blockage regions always giving the maximum torque.

12 Implementation and physical tests

Based on the results of chapter 11 the LQR GS SCS and hydraulic controllers are converted into Python and C++ scripts to be used as crawler control systems. In this chapter the performance of these controllers will be tested on the physical crawler setup. The test member is oriented horizontally since the controllers are developed for this orientation. The test member contains a welding seam, as seen in the photo 12.1 marked red. The black arrow shows the direction of rotation.

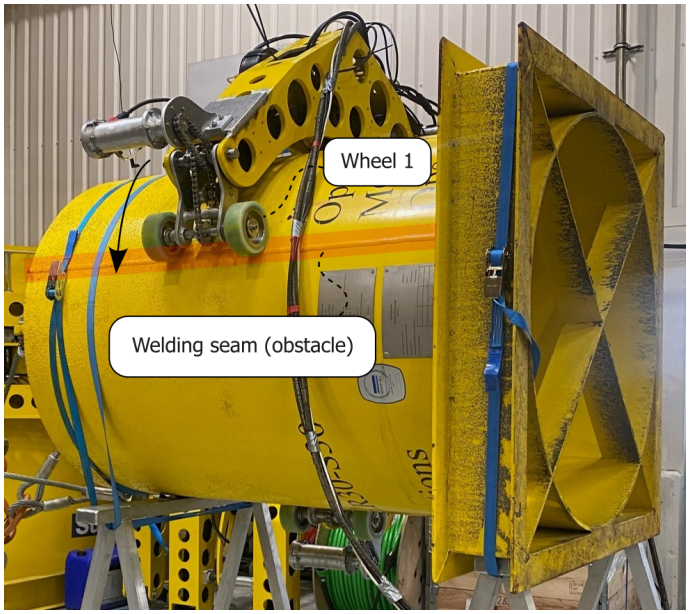


Figure 12.1: Physical test setup used for testing controller performance with varying velocity references and welding seam used as obstacle for the crawler to overcome

12.1 Tests

With the LQR GS SCS controller implemented on the physical crawler, tests are performedformine the performance of this controller. The code for both the hydraulic actuator and the the amount ofcan be found in appendSo, D.1 andD.2. Based on the requirements a 2.3 the tests are performed with the following references seen in table 12.1.

Table 12.1: Tests performed to identify the performance of the LQR GS SCS cocrawler implemented to the physical crawler

Test	1	2	3	4	5	6
Position ref.	2π	2π	2π	2π	2π	2π
Velocity ref.	0.016	0.048	0.100	0.016	0.048	0.100
Hydraulic controller enabled	no	no	no	yes	yes	yes

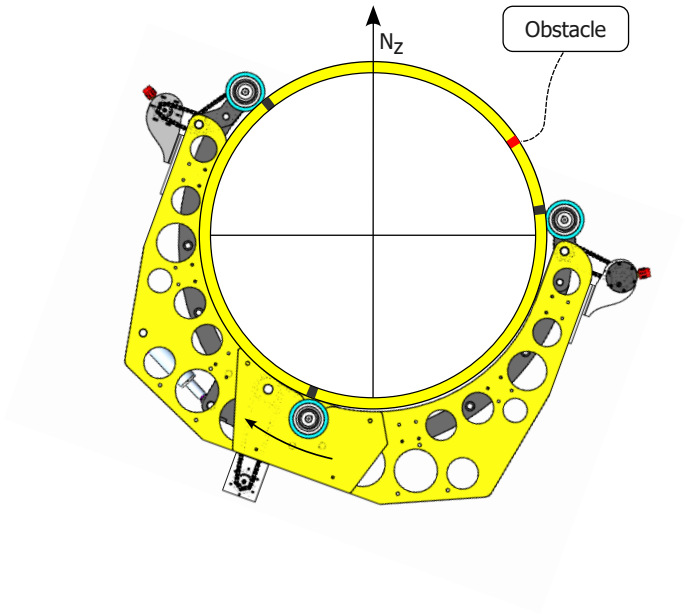


Figure 12.2: Position of the welding seam, shown in red, serving as an optical for the crawler.

In figure 12.2 the black marks are the initial positions of the wheels. Wheel two has the same initial location as the crawler body, which is -0.90 rad. The red mark is an obstacle, located at 0.76 rad. The black arrow shows the direction of the crawler rotation. When performing the test, a clear impact of the hydraulic controller on the system is observed. The test results can be seen in figure 12.3.

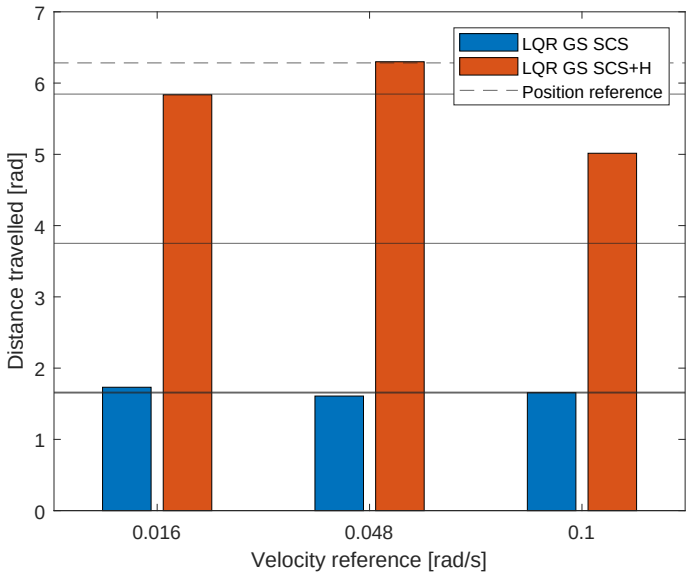


Figure 12.3: Figure of the three velocities tested, showing the distance traveled. Depicted are the tests with and without the logic hydraulic controller.

From the figure it is seen that without the hydraulic controller the crawler is unable to move the first wheel over the obstacle as depicted in figure 12.2. This is visualized with the lowest straight black bar at 0.76 rad. The second bar at 2.85 rad shows where wheel two encounters the obstacle. The third bar at 4.95 rad shows where wheel three encounters the obstacle 12.3. With the hydraulic controller enabled, all the different reference speeds are able to get their first and second wheels over the obstacle. Only the 0.048 reference speed was able to get the last wheel over as well and reach the desired position.

When examining the mean torque used through the experiments as depicted in figure 12.4, it is seen that the feed-forward term of the controller contributes with an average torque of 46.2 Nm, in the test with fixed pressure. This leaves little room for the controller to actuate as the motors are saturated at 54 Nm. This saturation is an issue with the controller with a reference of 0.1 rad/s as its demanded torque is above saturation. Both for the one with and without the hydraulic actuator on. In the three tests where the hydraulic controller is enabled, the average feedforward term is less than when it is disabled and not constant in between tests. This is a result of the test not stopping at the same location as depicted in figure 12.3. This means that when the crawler moves past the first obstacle, the feed-forward term starts to diminish as gravity now contributes to the crawler's rotation instead of hindering it. Because of this, the second hydraulic test has the lowest average feed-forward term as it reached its desired position and traveled the furthest with gravity contribution to its rotation.

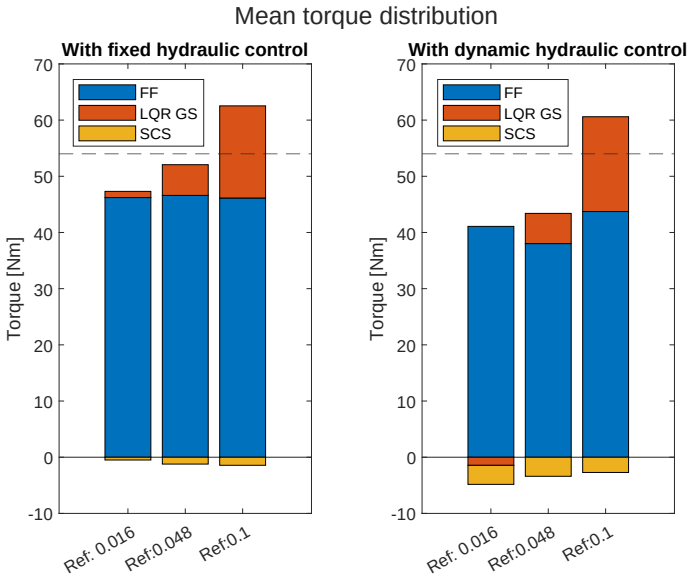


Figure 12.4: Mean torque distribution

The contribution of the SCS controller is also more active when the hydraulic controller is no longer fixed. This correlates, since pressure changes directly influence the wheels, with how much rolling friction they experience, causing the wheels to speed up or slow down, as they have lower inertia compared to the crawler. This then results in slippage occurring between the wheels and the crawler, the SCS controller is activated in order to reduce the slip. The reason why the SCS demands opposite torque compared to the main controller is that it is trying to reduce the speed of the wheels to match the crawler speed, which takes different time to match due to inertia differences. The actual torques and the corresponding velocities are visualized in figure 12.5.

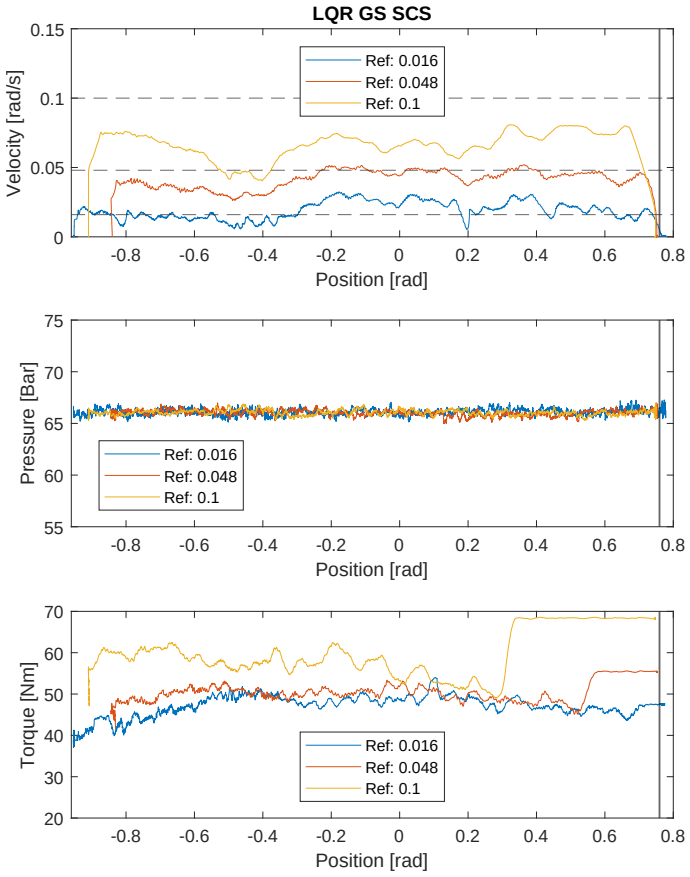


Figure 12.5: Figure of velocity, pressure and torque vs, Position for the tested LQR GS SCS controller implemented to the physical crawler.

The figure shows that the controller can keep a steady velocity around their reference for 16 and 0.048 rad/s. With a standard deviation of 0.012 and 0.029 rad/s, respectively 12.2. The data on the plots and all future plots are filtered by using a moving average of 50 data samples, with a sampling frequency of 50Hz results, as the Jetson were not able to process the code faster. The reason why the 0.1 rad/s reference could not be reached is because the controller is not aggressive enough, supplying too little torque. But even with torque at its saturation point, as seen at 0.3 rad, the torque is not enough to reach the desired speed at that angle. So in order for the speed to always be reached, the pressure has to be reduced to minimize the rolling friction. As it is currently at 65 bar, which is close to the reference pressure of 63 bar.

Table 12.2: Documented test results of the performance of the LQR GS SCS controller implemented to the physical crawler.

Test	1	2	3	4	5	6
Steady state error [%]	85.41	86.20	85.81	8.28	-0.30	24.03
Settling time [s]	-	-	-	-	160.55	-
Time in unstable region [%]	99.63	99.68	99.20	98.03	99.49	99.74
std. from ref. velocity [rad/s]	0.012	0.029	0.045	0.052	0.057	0.064
Slip controller activated [%]	11.20	22.12	14.60	12.07	11.14	9.49
Time to overcome obstacle 1 [s]	-	-	-	48.70	33.39	40.34

During testing, the size of the normal force was set to a constant value determined from 63 bars and its initial angle, instead of being continuously updated for the feedforward term. This was done to allow the Crawler to operate at a constant low velocity of 0.016 rad/s. The LQR controller had very little torque that could supply only $\pm$ 2 Nm. So, the size of the rolling friction had to be very accurate to ensure that the crawler could move with that controller. The problem with continuously updating normal forces is that they depend on piston pressure, and pressure readings fluctuate with 9 bar 10.4.2, which corresponds to nearly 2 kN of piston force. This uncertainty in the pressure caused the rolling friction feed-forward term to cause chattering. After setting the normal forces to constant, the previously found rolling coefficient of 0.039 as stated in table 9.5 was too low and increased by a factor of 1.9. This increase in rolling friction is caused by the crawler dragging hydraulic tubes as well as power and communication cables when rotating. This causes extra drag and mass. This, as well as the viscous friction of the motors, is not included in the feed-forward either. This increase was chosen over making LQR more aggressive as it resulted in less chattering.

Chattering was still an issue, this is most visible in table 12.2. The crawler spends over 99.2% of its time in the unstable region in all tests. Despite crawler, slipping all the time, the slip controller was only active 11.2% of the time for 0.016 rad/s case. The reason for this discrepancy is that the slip controller has to pass the hysteresis band conditions 10.70. So while the crawler is standing still or close to, the slip controller will not activate. This was also applied to the motors since at these low velocities, the slippage using the approximate slip equation 10.10 could cause stack overflow without it. However, the amount of slippage is due to the low velocities with which the crawler is operating, so a small amount of noise disturbance from sensors can cause the crawler to estimate its slippage. This coupled with the crawler not getting the sensor data at the same time, from IMU and motors, also the crawler not operating at steady state velocity causes the crawler to appear to be slipping. Practically, none of the wheels was observed to slip freely during the test without hydraulic control. This is also observed in figure 12.6 first 63 seconds with hydraulic control before it hits the obstacle.

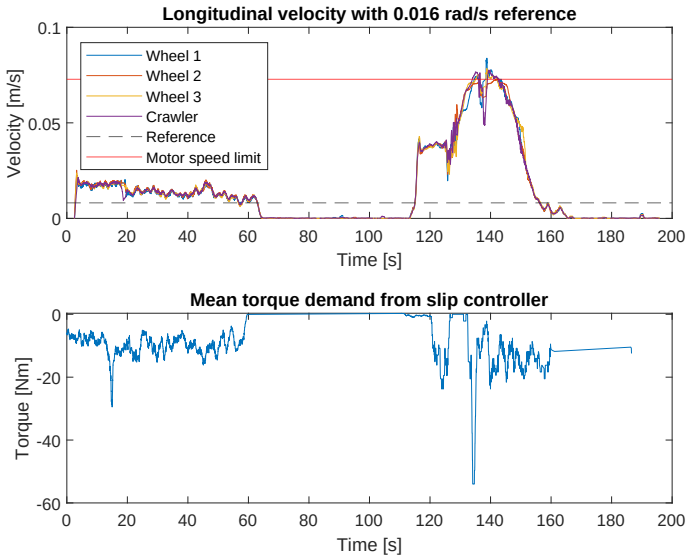


Figure 12.6: Test for the reference velocity of 0.0016 rad/s and the use of the hydraulic controller.

The data in the figure shows that the different wheels with a moving average overlay have similar velocities to that of the crawler. This indicates that the crawler is not slipping, or if it is, that it recovers and therefore has enough traction. If the crawler is slipping, the motor speed will increase to its maximum of 0.0728 m/s, represented as the red line in figure 12.6, which it reaches at 130 to 150 s. When it reaches this velocity the motors fail-safe turn on, and the motors are shut off.

The slip controller actively provides torque to prevent the slippage 12.07% of the time 12.2 depicted in the bottom graph of 12.6. The spike at 17 seconds is a response to the change in pressure in the hydraulic piston. By applying the moving average filter to the torque data it can appear as if the slip controller is active more often than what is really the case. The slip controller is off when the crawler is not moving nor trying to overcome the obstacle. The obstacle is overcome after 48.70 s 12.2. The increase in speed causes the slip controller to come online after the blockage, where it tries to stabilize the system. As the system now is slipping which is seen in the difference in motor speeds and crawler, resulting in a more aggressive slip controller. This response is also visualized in the combined torque given to the motors in the third graph of 12.7.

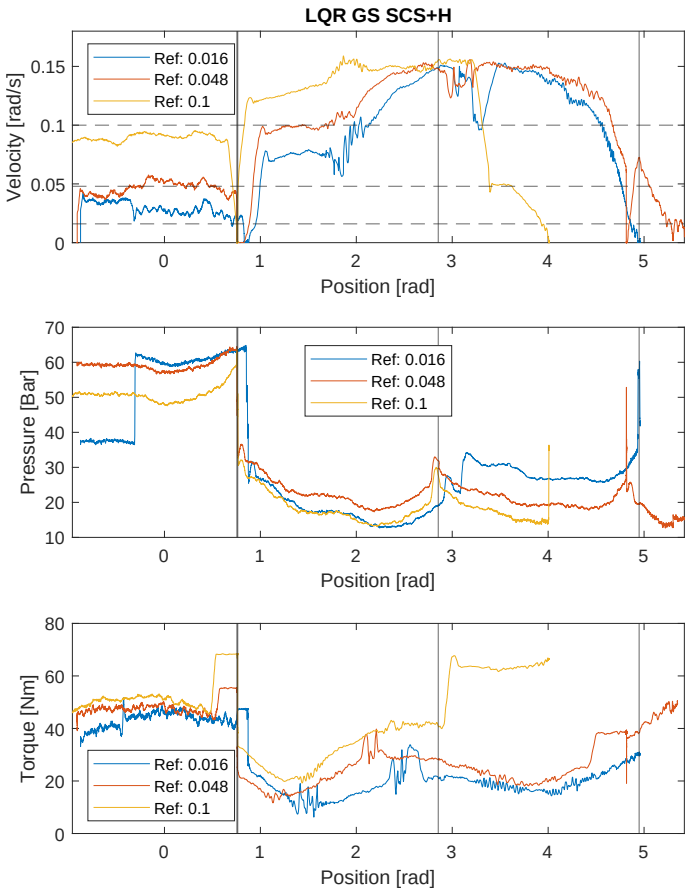


Figure 12.7: Figure of velocity, pressure and torque vs, Position for the tested LQR GS SCS controller with an additional hydraulic controller, implemented to the physical crawler.

The required motor torques decrease in all cases, after overcoming the first obstacle due to the slip controller. The slip controller does not take into account the reference velocity, but only the relative slippage. So, it does not prevent an increase in velocity as long as the slippage between them is acceptable from its perspective. That is the purpose of the LQR GS, but it is not aggressive enough to prevent the speed increase as visualized in the beginning of graph 12.7.
The increase in speed is a result of the change in normal force from dynamically depending on pressure to being set to a constant value. This results in a constant feed-forward contribution of the rolling resistance. The rolling resistance value was based on 63 bar, but the crawler is only held by 30 bar when it crosses the first obstacle, as illustrated in the second graph 12.7. The black bar visualizes where the individual wheels pass the obstacle. The pressure also drops to a value of 10 bar for a reference speed of 0.016 and 0.1 rad/s. This results in the rolling friction term being 6 times the actual size, making the crawler accelerate. With the LQR controller not being more aggressive, it can not counteract the feed-forward term.
This causes the motors to reach their maximum speed, at which point their fail-safe turns on. This happened in the 0.1 rad/s experiment where the motors exceeded the maximum and they turned off after passing the second obstacle. This can be seen in the demanded torque, where it rises to the maximum afterward since the motor speeds are 0.

The 0.048 rad/s reference was able to move past the obstacle with all three wheels as well, and stopped at the reference position. The position controller activation is visible at 4.9 rad, this is visualized as a spike in torque at that position. The 0.016 rad/s could not pass the obstacle with the last wheel as the pressure was higher. Given more time the hydraulic controller could have ensured that the crawler passed the obstacle, it was, however, disabled prematurely since the piston pressure dropped to 10 bar. However, this was not an issue when attached to the upper part of the member. But when hanging downward, the test was cut off when the pressure reached 40 bar as depicted in figure 12.8.

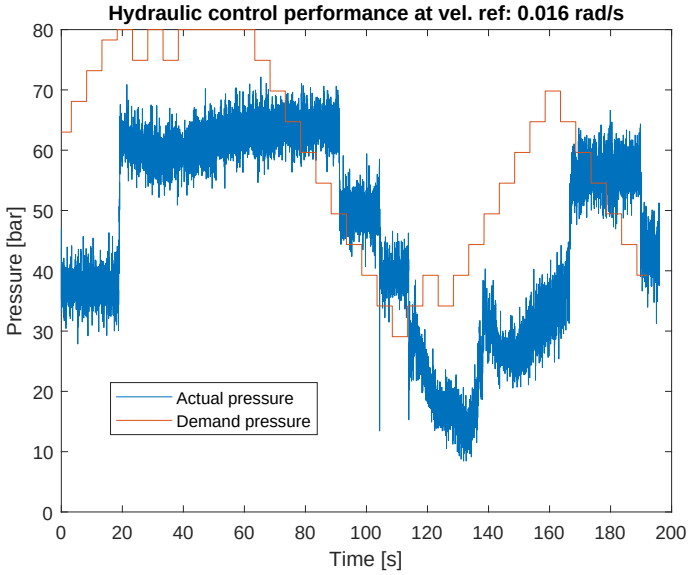


Figure 12.8: Piston results, showing the difference between the demanded pressure and the actual pressure.

In the figure, the demanded pressure is visualized in orange, and the actual pressure is in blue. The pressure here is not filtered with a moving average. The time delay is visible here as well. The initial pressure of 40 bar was corrected to the reference pressure of 63 bar, after 20 seconds. However, it could not increase the pressure further, as the bandwidth of the operating amplifiers had diminished, when operated with full control, the slower computational time of the full controller compared to the script only handling the hydraulic controller resulted in a lower output voltage. This is due to the Arduino publishing what it receives from the Jetson, when it does not receive a signal, it goes back to neutral. The blockage is detected at 63 s, where it starts reducing its reference pressure. Where the actual pressure falls when the difference between the reference and current is above 10 bar. Before the error is large enough for it to act. The hydraulic conditions that changed the pressure throughout the test are shown in table 12.3.

Table 12.3: Number of times the hydraulic controller is activated based on the conditions

Hydraulic conditions	Test 1	Test 2	Test 3	Test 4	Test 5	Test 6
1. Slippage	0	0	0	11	7	3
2. Motor torque	22	9	7	2	11	12
3. Rotating torque	0	0	0	0	1	0
4. Falling	1	0	0	8	7	7
5. Crawler stopping	0	0	0	17	8	9

The hydraulic controller was not active in the first test, but the controller still determined which condition it would have acted on. There the main concern was high motor torque. Therefore, it would reduce the pressure to lessen the rolling friction. The 0.016 rad/s had a single case where it wanted to raise the pressure, again due to the risk of falling. This is because the controller uses the demand pressure instead of the actual pressure. To ensure it is safe, it is necessary to change the pressure before it is too late.

The tests with the hydraulic control are more varied. Where it detects more problems with slippage at lower speeds compared to the higher velocity references because of the smaller changes needed in velocity for it to detect slippage. The motor torque is a greater issue at the higher reference velocities which correlates with the LQR controller demanding more torque to reach those speeds. There is only a single case where it does not have enough traction to continue rotating. The condition to prevent it from falling was more dominant. Due to the low pressure, the hydraulic controller reduced it to 10 to 25 bar, which resulted in the controller trying to increase the pressure until it was within the safe region. The condition of checking whether the crawler is stuck is activated more in the case with 0.016 rad/s since that test had more time to reduce the pressure for the last obstacle, as seen at the end of the figure 12.8. Here, the motors of the 0.1 rad/s case reached their safety speed limit and turned off, before reaching the obstacle. The 0.048 rad/s case already had lower pressure and higher speeds in order to overcome the obstacle. Both cases use the same amount of time to overcome the obstacle on the first wheel with the use of the hydraulic controller as seen in table 12.2.

13 Discussion

The crawler has been partly electrified by implementing three Maxon motors, as well as an onboard control system and a Syd dynamics TM352 IMU. The Maxon motors were limited in the torque they could deliver due to the gear attached to the motor only being able to withstand 42 Nm. This proved to be too little for the crawler to complete a singular rotation around a horizontal member. Therefore, the torque was increase to 54 Nm. The gear on the Maxon motor was already destroyed from previous use of high torque demand. This limiting factor causes the crawler to not be able to traverse the obstacle, without the hydraulic controller. The motor itself could provide 73 Nm, if the gear could handle it. The motors currently operate at a speed 3 times the maximum required speed from SubC Partners. This would allow for a further gear reduction of 3, allowing for a total torque of 219 Nm per motor. Compared to the total torque of 162 Nm in the current setup.

From the simulations it was seen that as long as two wheels are not slipping, the crawler would still be able to operate. So, a reduction a motor could be done using a better gearing, with the two motors then having SCS implemented onto them.
Both linear controllers were linearized in a manner that removes many important dynamics, especially the traction forces, which describe the conversion of movement between input force to the wheels and the movement of the crawler. This results in linear controllers which do not fit the non-linear model and thus needs re-tuning.

The PID controller was only developed to handle position tracking and does this very well. Managing a fast response to reach the desired location. However, since an important goal was also to track a desired velocity, the LQR controller was implemented since this control strategy enables the possibility to introduce multiple tracking references. By introducing gain scheduling to this control, it was possible to track both velocity and position.

The idea of using LQR control is that the state-space approach is good at describing couplings between the states in the system. However, as previously mentioned, the most important dynamics of the nonlinear model was the traction force and slip between wheel and crawler velocities, and due to their high non-linear behavior these were not implemented, reducing the accuracy of the linear description. When developing multiple controllers using the LQR control approach, the developed controllers were not robust

enough to handle the differences between the linear and non-linear models, thus both controllers needed re-tuning, as did the precompensators.

When the linear controllers were implemented, a feedforward term was required as the crawler experiences both rolling and viscous friction together with gravity. Without it, the LQR controller was not able to function properly. The PID based IMC could handle it since it was more aggressive compared to the LQR controller, this controller also included an integral term as well to reduce the steady-state error over time. On the downside, the more aggressive control of the PID caused the crawler to slip more often than the slower LQR.
In practice, the crawler could not overcome the obstacle when the torque was maxed out, until the piston pressure was reduced to 25 bar. The slip controller causes chattering because the refresh rate of the motors and IMU were not done at the same time. At lower velocities, misaligned measurements as well as noise disturbances could cause the controller to estimate that the crawler is constantly slipping, activating the slip control supervisor. The feedforward term needs to be reverted back to using the calculated normal forces instead of a constant term, to avoid the high velocities after passing the obstacle. Due to lower pressure and therefore lower rolling friction.

14 Conclusion

This report investigated the possibility of making SubC Partners crawler autonomous and able to handle slip for inspection and cleaning purposes. To achieve this, the crawler was redesigned, adding three electric motors, hydraulic control, IMU position estimation, and microcontrollers to handle the control. A mathematical model of the crawler is developed, and the parameters to describe the crawler are determined from experiments. The model included a description of its slip and traction.

From the model, an IMC-based PID controller was developed to obtain a starting position reference. In addition, two LQR controllers are tuned, one for following a velocity reference and one for position. The transition between these two controllers is done using gain scheduling, changing controllers when the crawler is within 5% of the position reference.

A slip controller was developed as the LQR GS controller supervisor, monitoring the slip on all wheels using a disturbance observer to estimate the difference in traction force. The supervisor is set to adjust the torque output of the LQR when active. The simulations showed that the slip controller allowed the LQR to rotate under worse surface conditions compared to without the supervisor. In practice, the hydraulic controller allowed the crawler to overcome an obstacle that was not possible with the current gearing of the motors. Different gears would improve performance as the gears cannot tolerate high torque. Increasing the gear ratio would result in a reduction in motor speed but an increase in available torque. This would make sense since the current setup allows the crawler to reach speeds up to 3 times above the requirements, but on the other hand, it lacks available torque.

The crawler was able to satisfy the requirements of SubC Partners, to some extent with the LQR GS SCS+H controller and a reference of 0.048 rad/s. The crawler reaches full rotation and can overcome obstacles; however, the velocity is above the reference in 60 % of the test because the feed-forward term of the controller is outside of the operation range due to the reduced pressure of the hydraulic controller.
Based on simulation results, two electrified wheels would be sufficient to rotate the crawler without the need to implement complex control. If the gearing was done differently, the crawler was partially electrified with the hydraulic cylinder needed to be replaced with three electric actuators.

15 Future Work

During the project, multiple additional challenges have occurred. For future improvements to the automation of the SubC Partners crawler, changes can be made to both the hardware and crawler software. Below, possible future considerations are made.

Hardware change considerations

- Changing the gearing of the motor to handle the possible 73 Nm.
- Reduce the gear ratio by three in the drive-train so a lower speed range can be obtained. With additional torque being available
- Equal distribution of additional mass on the crawler.
- Fixing the skewnes of the crawler claws for more uniform force transmission from the wheels to the offshore member.
- Make the crawler waterproof.
- Replace the hydraulic piston with an equivalent amount of electrical actuators discussed in section 5.2.
- Using encoders with higher resolution on the motor, in order to minimize the slip problems caused by the motor velocity reading errors.

Software change considerations

- Develop two PID controllers, one for position and one for velocity and adding gain scheduling.
- Testing the difference between the use of slip controllers on every individual wheel and the use of a combined slip controller.
- Rewrite the controller code from Python to C++, to increase the computational time.

Bibliography

[1] BlueNord, *Tyra*. [Online]. Available: https://www.bluenord.com/tyra/.

[2] Total Energies, *Tyra Redevelopment*. [Online]. Available: https://tinyurl.com/mvryxhd9.

[3] Sub-C Partner, *The "Spider Deck Clean-Up" Campaign*. [Online]. Available: https://www.subcpartner.com/news/the-spider-deck-clean-up-campaign-tyra/.

[4] Karin Rix Holländer, *Tyra-projektet foran tidsplanen*, 2019. [Online]. Available: https://portesbjerg.dk/news/tyra-projektet-foran-tidsplanen.

[5] J. Loxton, A. K. Macleod, C. R. Nall, *et al.*, "Setting an agenda for biofouling research for the marine renewable energy industry", *International Journal of Marine Energy*, vol. 19, pp. 292–303, Sep. 2017.

[6] N. A. Othman, M. H. Mohd, M. A. A Rahman, M. A. Musa, and A. Fitriadhy, "Investigation of the corrosion factor to the global strength of aging offshore jacket platforms under different marine zones", *International Journal of Naval Architecture and Ocean Engineering*, vol. 15, p. 100 496, Jan. 2023.

[7] Sub-C Partner, *The Robotic Crawler*. [Online]. Available: https://www.subcpartner.com/products/the-robotic-crawler/.

[8] J. B. Rasmussen, "Modelling of the SubC Crawler 6th semester Diploma", Tech. Rep., 2022. [Online]. Available: http://www.aau.dk.

[9] G. Tao, "Multivariable adaptive control: A survey", *Automatica*, vol. 50, no. 11, pp. 2737–2764, Nov. 2014.

[10] DMI Ocean, *Wave height statistics*. [Online]. Available: https://ocean.dmi.dk/wavestat/index.uk.php.

[11] Hainzl WebShop, *Hydraulik motor, OMM50 151G0013*. [Online]. Available: https://tinyurl.com/3z7vemzy.

[12] S. Aberg, I. Rychlik, and M. R. Leadbetter, "Palm distributions of wave characteristics in encountering seas", *Annals of Applied Probability*, vol. 18, no. 3, pp. 1059–1084, Jun. 2008.

[13] Total Energies, *Tyra II – TotalEnergiesThe build-up of one of the world's most advanced gas fields starts now in the Danish North Sea*, 2021. [Online]. Available: https://tyra2.dk/en/the-build-up-of-one-of-the-worlds-most-advanced-gas-fields-starts-now-in-the-danish-north-sea/.

[14] A. H. TECHET, "Morrison's Equation", 2004.

[15] R. Huff, "Factors of safety", in *SAE Technical Papers*, 1916. [Online]. Available: https://www.engineeringtoolbox.com/factors-safety-fos-d_1624.html.

[16] C. N. Nwogu, B. N. Nwankwojike, C. Anyaregbu, and C. Arisa, "Design and fabrication of a multipurpose railroad hand truck", *Nigerian Journal of Technology*, vol. 36, no. 2, p. 477, Apr. 2017.

[17] Continentalfan, *What is an Electronically Commutated EC Motor? - Continental Fan*. [Online]. Available: https://continentalfan.com/what-is-an-electronically-commutated-ec-motor/.

[18] Enervex, *Why Are We Switching to EC Motors?* [Online]. Available: https://enervex.com/insights/why-are-we-switching-to-ec-motors.

[19] Maxon, *EC 40 40 mm, brushless, 170 watt*, 2021. [Online]. Available: https://www.maxongroup.com/medias/sys_master/root/8882348785694/EN-21-239.pdf.

[20] Association for Advancing Automation, *Measure While You Press — Force Feedback in Linear Actuators*. [Online]. Available: https://www.automate.org/tech-papers/measure-while-you-press-force-feedback-in-linear-actuators.

[21] Danfoss, "PVG 32 Proportional Valve Group Technical Information", Tech. Rep., 2013.

[22] SydDynamics, *SydDynamics TM352*. [Online]. Available: https://tinyurl.com/bdcusam2.

[23] F. Ren, Y. R. Zheng, M. Zawodniok, and J. Sarangapani, "Effects of electromagnetic interference on control area network performance", in *2007 IEEE Region 5 Technical Conference, TPS*, 2007, pp. 199–204.

[24] Mathworks, *MATLAB Camera Calibration Toolbox*. [Online]. Available: https://se.mathworks.com/help/vision/camera-calibration.html.

[25] G. Genta, *Motor Vehicle Dynamics. Modeling and Simulation*. World Scientific, 1997.

[26] M. Amodeo, A. Ferrara, R. Terzaghi, and C. Vecchio, "Wheel slip control via second-order sliding-mode generation", *IEEE Transactions on Intelligent Transportation Systems*, vol. 11, no. 1, pp. 122–131, Mar. 2010.

[27] E. Bakker, H. B. Pacejka, and L. Lidner, "A New Tire Model with an Application in Vehicle Dynamics Studies", Tech. Rep., 1989, pp. 101–113. [Online]. Available: https://about.jstor.org/terms.

[28] I. J. Besselink, A. J. Schmeitz, and H. B. Pacejka, "An improved Magic Formula/Swift tyre model that can handle inflation pressure changes", *Vehicle System Dynamics*, vol. 48, no. SUPPL. 1, pp. 337–352, 2010.

[29] H.-C. Becker Jensen, "Design of Slip-based Active Braking and Traction Control System for the Electric Vehicle QBEAK", Tech. Rep., 2012.

[30] A. Da, S. Junior, C. Birkner, and R. Nakhaie Jazar, "Coupled Lateral and Longitudinal Controller for Over-Actuated Vehicle in Evasive Maneuvering with Sliding Mode Control Strategy", 2023.

[31] Maxon Group, "Edition May 2017 EPOS4 Positioning Controllers Application Notes", Tech. Rep., 2017.

[32] G. F. Franklin, D. J. Powell, and A. Emami-Naeini, *Feedback Control of Dynamic Systems*, 7th editio. Pearson Education Limited, 2015.

[33] A. K. Waghmare and P. Sahoo, "Adhesive friction at the contact between rough surfaces using n-point asperity model", *Engineering Science and Technology, an International Journal*, vol. 18, no. 3, pp. 463–474, Sep. 2015.

[34] J. S. McFarlane and Tabor D., "Relation between friction and adhesion", 1950.

[35] T. F. E. Dale E. Seborg and D. A. Mellichamp, *Process Dynamics and Control: THE IMC-BASED PID PROCEDURE*. B.W.Bequette, 1999, p. 29.

[36] J. Shi and W. S. Lee, "Set Point Response and Disturbance Rejection Tradeoff for Second-Order Plus Dead Time Processes", Tech. Rep., 2004.

[37] Ming T. Tham, "IMC tutorial", *Lecture notes on Introduction to Robust Control*, pp. 1–9, 2002.

[38] Mathworks, *Anti-Windup Control Using PID Controller Block - MATLAB & Simulink*. [Online]. Available: `https://se.mathworks.com/help/simulink/slref/anti-windup-control-using-a-pid-controller.html`.

[39] H. Yanai, *Projection Matrices, Generalized Inverse Matrices, and Singular Value Decomposition*. 2011.

[40] J. Chen, *Algebraic Theory of Generalized Inverses*, eng, First edition., X. Zhang, Ed. Singapore: Springer, 2024.

[41] Y. Ikeda, T. Nakajima, and Y. Chida, *Vehicular slip ratio control using nonlinear control theory*. IFAC, 2011, vol. 44, pp. 8403–8408. [Online]. Available: `http://dx.doi.org/10.3182/20110828-6-IT-1002.01325`.

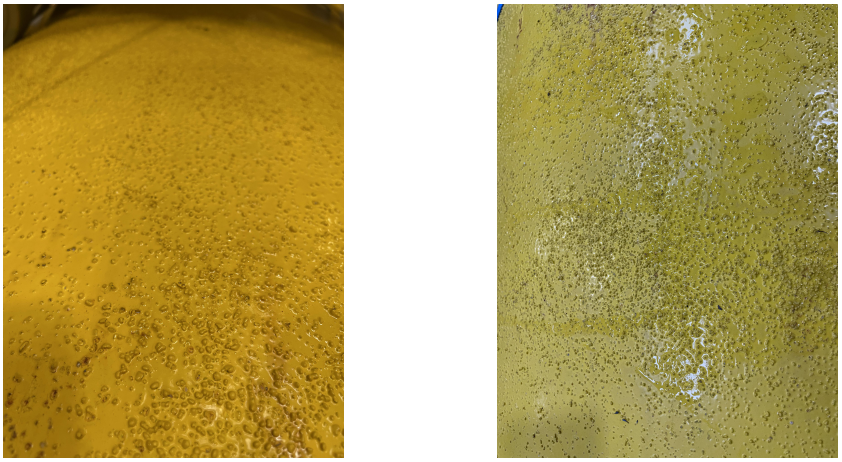
[42] Texas Instruments, "LMx58-N Low-Power, Dual-Operational Amplifiers", Tech. Rep., 2022. [Online]. Available: `www.ti.com`.

A Appendix: Extra Results

A.1 Static friction tests

Dry and wet rough paint surface

The next friction test is conducted on a rough paint surface condition directly on the offshore member. This is done for when its dry and wet, as can be seen in Figs. A.1a and A.1b. The rough paint simulates a more worn-out offshore member, which will be the case when the member was in the sea for a longer time period.



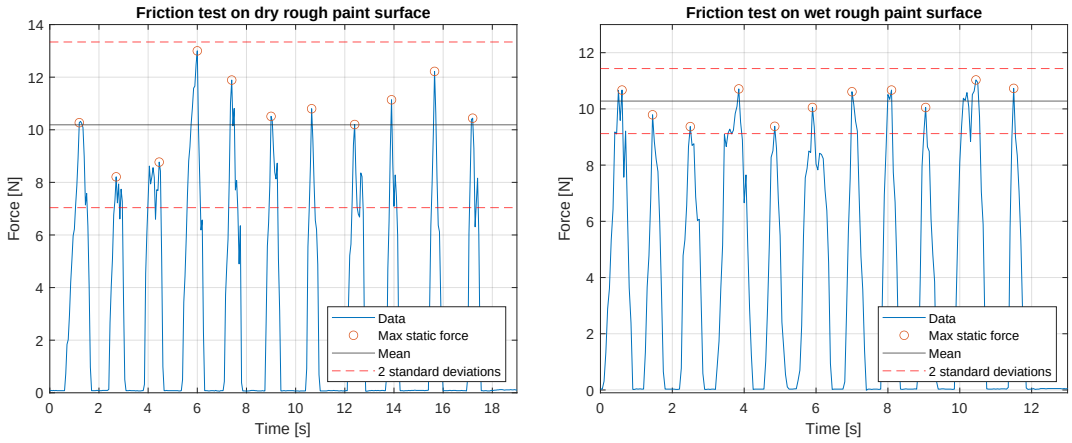
(a) Dry **(b)** Wet

Figure A.1: Rough paint test surface, simulating a more rough offshore member surface as the condition can be after years in the sea.

The test was conducted on the offshore member used for the crawler tests. The member is cylindrical, making it very challenging to stabilize one wheel on top and to drag it over the surface without it falling off. A solution was developed as seen in picture A.2. The wheel can stand between the aluminum plates, making it impossible to roll off. The friction between the wheel and the plates is neglected. In this test the wheel is pushed over the surface and not pulled, ensuring more controlled handling of the wheel.

Figure A.2: 3D printed solution for testing of the rough surface of the offshore member.

The test results are depicted in graphs in figure A.3a for the dry surface condition and in figure A.3b for the wet surface condition.



(a) Graph of the friction test for the dry rough paint surface.

(b) Graph of the friction test for the wet rough paint surface.

Figure A.3: Rough paint test data for both the wet and the dry surface.

The friction forces for the dry and wet surface condition show similar peaks which results in a mean friction force, shown as a black line, of 10.19 N for the dry condition and 10.29 N for the wet surface condition.

The differences in the test lay in the standard deviations, shown as red dotted lines, where the wet surface condition has a lower standard deviation of 0.58 in comparison to the dry condition which is 1.57.

A challenge in this experiment is the rough surface which can make the wheel get stuck

when pushed over the surface. This results in differences in the individual friction force peaks shown as red circles in figure A.3a and A.3b.

The mean friction coefficient for the dry surface is determined to be 0.46 $\pm$ 0.07 and 0.46 $\pm$ 0.03 for the wet surface condition. The friction coefficients are documented in table 4.2.

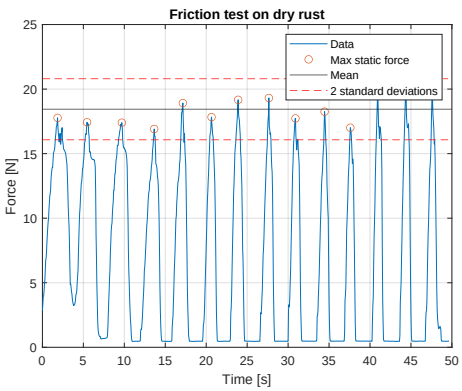
Dry rust and wet rust surface

In this experiment, a rusty plate as seen in figure A.4a is used as a test surface. The steel plate simulates the surface on rusty offshore members. The setup to conduct the experiment is the same as depicted in figure 4.1 with a slight change. The wheel is not pulled over the member but instead is being pushed with a Newton meter.

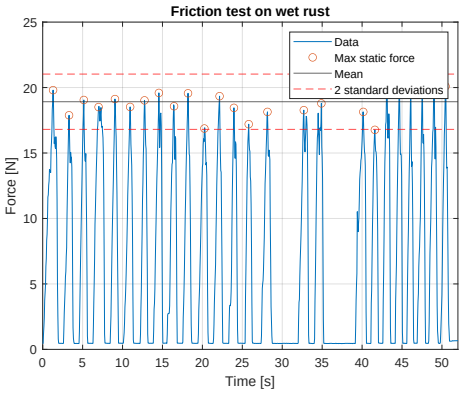
(a) Dry **(b)** Wet

Figure A.4: Rusty test surface, simulating a offshore member.

The test was carried out for 50 seconds giving a total of 14 data points used to determine the mean force needed to slide the wheel over the surface. The data points can be observed in figure A.5a, marked with red circles. It can be observed that the forces are not settling at 0 N but are having a slight offset which is considered a sensor-related disturbance. As in previous tests, also here the valid friction force peaks must be in a range of two times the standard deviation shown in figure A.5a as the red dotted lines.
The experiment for the wet rust surface is conducted on the same rusty steel plate in a wet condition to observe the change in friction. This condition is necessary in order to simulate a crawler operation in a more wet environment as discussed in section 3. The wet rusty plate is depicted on figure A.4b.
The experiment is carried out for around 50 seconds, resulting in 25 data points. The experiment-related graph will contain the same information as the previous graphs, which includes the mean value and the standard deviations.



(a) Graph of the friction test done on a dry rusty metal plate.



(b) Graph of the friction test done on a wet rusty metal plate.

Figure A.5: Data results of the rust surface tests.

From the data points of the dry rust surface on figure A.5a, the mean friction force is calculated to be 18.44 N illustarted as a black line in figure A.5a. The standard deviation for this data set is 1.18. Using equation 4.1 the related friction coefficient (μ_s) between the nylon wheel and the dry rusty surface is determined to be 0.79 with a standard deviation of 0.05. The same test is conducted for the rusty steel plate in wet condition.
For the wet rust surface with corresponding figure A.5b, the mean friction force of 18.92 N is determined from the chosen data points and shows a standard deviation of 1.06. This friction force is then used to calculate the friction coefficient mu (μ_s) of the wheel to the rusty plate to be 0.81 with a standard deviation of 0.05.

Greasy surface condition

This experiment was carried out to find a material that would be able to be applied to another surface to reduce its friction coefficient to see how the crawler would perform on a slippery surface. This is required since the test member 3.5, only has two surface conditions. The grease used for the test was SKF bearing grease, since this is assumed to have a low friction coefficient when applied. The grease is evenly distributed on the tape, as depicted in figure A.6.

Figure A.6: Greasy surface when used SKF Bearing grease.

The measured data can be observed in figure A.9 with legends. The test data are taken over a time of 25 seconds, providing a sufficient amount of valid friction force peaks for data handling and determining the friction coefficient μ_s. The mean friction force is illustrated in figure A.9 as 8.61 N.

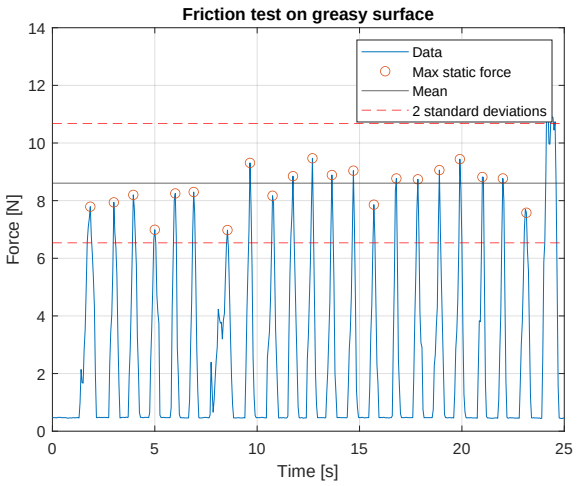


Figure A.7: Graph of the friction test done on a greasy surface using SKF bearing grease.

The friction coefficient μ_s is calculated from the mean friction force and can be observed to be 0.37 with a standard deviation of 0.05.

Dry mussels surface

Due to an unregular surface condition when using mussels and similarities to other tetsted surfaces, the test will not be conducted for the dry and wet mussel conditions.

Wet marine growth (algae)

This test is conducted at the harbor and follows the same approach as in test 4.1.1. The surface of this test was covered in algae and had a slight slope of 4°. The surface can be seen in figure A.8.
The angle of the slope of the surface must be taken into account when determining the friction coefficient μ_s between the wheel and the algal.

Figure A.8: Picture taken of the algae surface condition taken from Emőke Dénes.

The test data is further processed with MATLAB, as shown in figure A.9, to determine the mean friction force to be 12.44 N with a standard deviation of 0.59. The deviation is rated as acceptable because it is in the 95% interval, following the empirical rule.

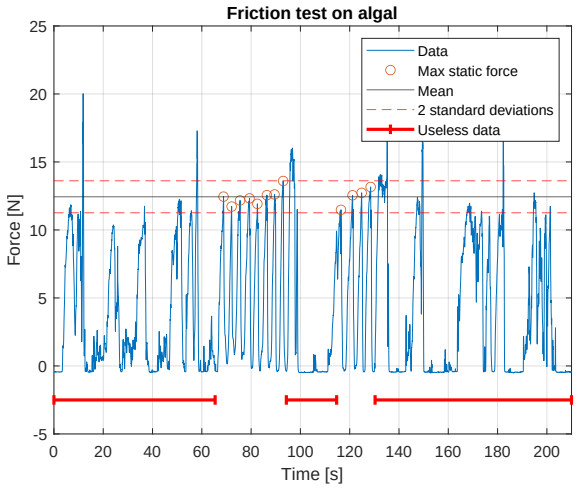


Figure A.9: Graph of the sixth friction test done on wet algal at the harbor.

During the test, there have been some complications regarding the stability of the wheel while dragging it along the algal surface. The test data have a time span of 210 seconds, from which 60 seconds can be used for the data processing. This is illustrated on figure A.9, where the red lines indicate the data that was not used further. The red circles positioned on the force peaks mark the data points used for calculation of the mean friction force indicated on figure A.9 with a black line.

The friction coefficient between the crawler wheel and the algal surface is determined to be 0.51 with an standard deviation of 0.2. The slope is taken into account by multiplying by $\cos(4°)$.

A.2 Position estimation test 1, 2, and 4

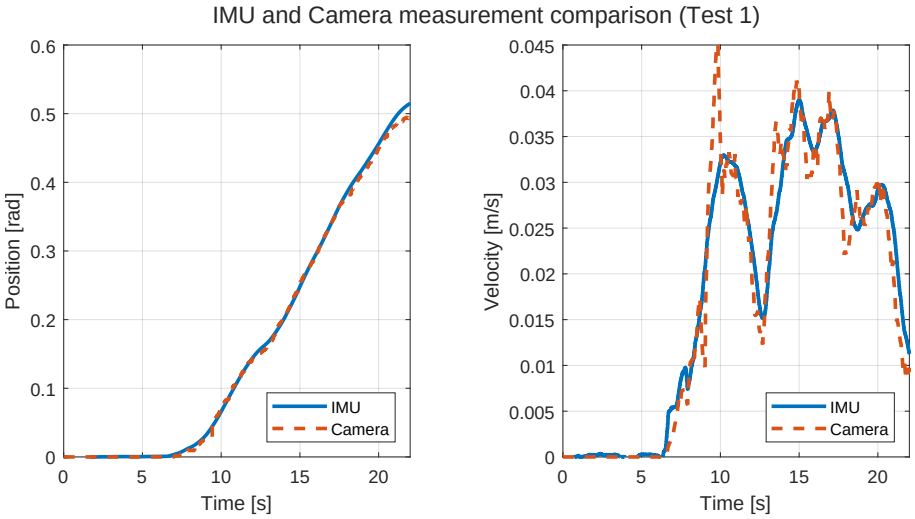


Figure A.10: Comparison between IMU and camera position and velocity measurements.

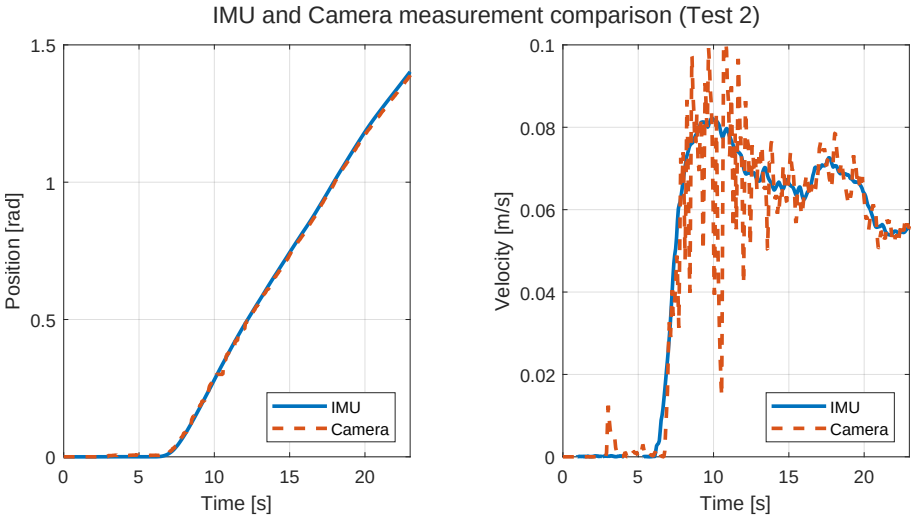


Figure A.11: Comparison between IMU and camera position and velocity measurements.

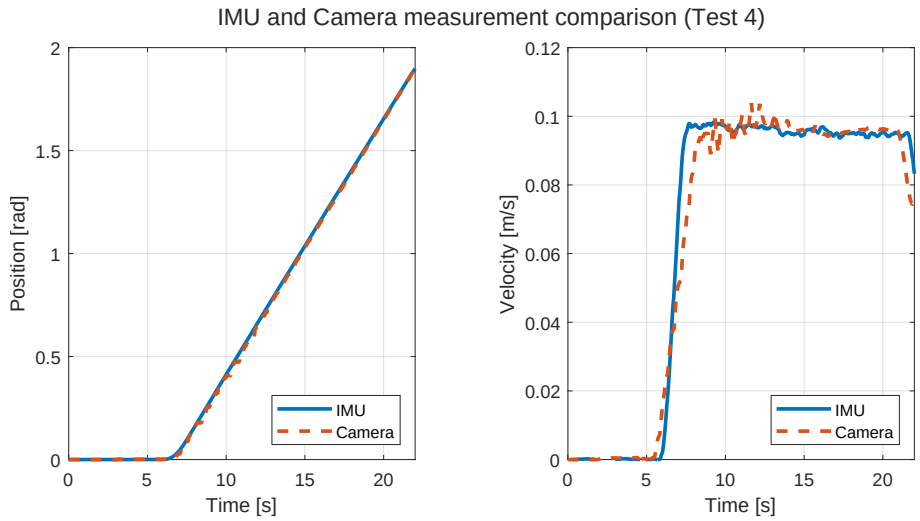


Figure A.12: Comparison between IMU and camera position and velocity measurements.

A.3 State-space matrices

$$\mathbf{A} = \begin{bmatrix} 0.000 & 1.000 & 0.000 & 0.000 & 0.000 \\ 2.584 & -27.219 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & -43.004 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & -43.969 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & -47.167 \end{bmatrix} \tag{A.1}$$

$$\mathbf{B} = \begin{bmatrix} 0.000 & 0.000 & 0.000 \\ 0.548 & 0.548 & 0.548 \\ 1.000 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 1.000 \end{bmatrix} \tag{A.2}$$

$$\mathbf{C} = \begin{bmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 1.000 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 1.000 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 1.000 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.000 & 1.000 \end{bmatrix} \tag{A.3}$$

$$\mathbf{K} = \begin{bmatrix} 172.253 & 1931.565 & 14.658 & -6.841 & -6.516 \\ 172.321 & 1931.820 & -6.981 & 14.407 & -6.549 \\ 172.535 & 1932.622 & -7.095 & -6.989 & 13.629 \end{bmatrix} \tag{A.4}$$

A.3.1 Position controller

$$\bar{\mathbf{N}} = \begin{bmatrix} 166.473 & 20.408 & 0.830 & 0.770 & 0.718 \\ 166.516 & 20.412 & 0.787 & 0.811 & 0.719 \\ 166.652 & 20.426 & 0.787 & 0.770 & 0.755 \end{bmatrix} \tag{A.5}$$

A.3.2 Velocity controller

$$\mathbf{K} = \begin{bmatrix} 0.122 & 1809.268 & 0.000 & 0.000 & 0.000 \\ 0.122 & 1809.268 & 0.000 & 0.000 & 0.000 \\ 0.122 & 1809.268 & 0.000 & 0.000 & 0.000 \end{bmatrix} \tag{A.6}$$

$$\bar{\mathbf{N}} = \begin{bmatrix} 5.965 & 22.897 & 8.027 & 7.638 & 7.125 \\ 5.965 & 22.897 & 7.807 & 7.853 & 7.125 \\ 5.965 & 22.897 & 7.807 & 7.638 & 7.326 \end{bmatrix} \tag{A.7}$$

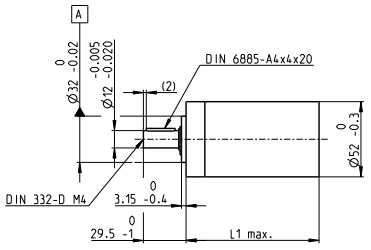
B Appendix: Data sheets

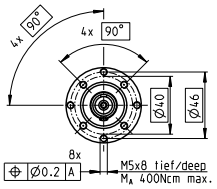
B.1 Maxon motor gear

gear

Planetary Gearhead GP 52 C ∅52 mm, 4.0–30.0 Nm

Ceramic Version



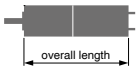


M 1:4

Technical Data				
Planetary Gearhead				straight teeth
Output shaft				stainless steel
Bearing at output				preloaded ball bearings
Radial play, 12 mm from flange				max. 0.06 mm
Axial play at axial load	< 5 N			0 mm
	> 5 N			max. 0.3 mm
Max. axial load (dynamic)				200 N
Max. force for press fits				500 N
Direction of rotation, drive to output				=
Max. continuous input speed				6000 rpm
Recommended temperature range				-15...+80°C
Extended range as option				-40...+100°C
Number of stages	1	2	3	4
Max. radial load, 12 mm from flange	420 N	630 N	900 N	900 N

Stock program
Standard program
Special program (on request)

			Part Numbers						
			223080	223083	223089	223094	223097	223104	223109
Gearhead Data									
1	Reduction		3.5:1	12:1	43:1	91:1	150:1	319:1	546:1
2	Absolute reduction		7/2	49/4	343/8	91	2401/16	637/2	546
10	Mass inertia	gcm²	20.7	17.6	17.3	16.7	17.3	16.8	16.4
3	Max. motor shaft diameter	mm	10	10	10	10	10	10	10
	Part Numbers		223081	223084	223090	223095	223099	223105	223110
1	Reduction		4.3:1	15:1	53:1	113:1	186:1	353:1	676:1
2	Absolute reduction		13/3	91/6	637/12	338/3	4459/24	28561/81	676
10	Mass inertia	gcm²	12	16.8	17.2	9.3	17.3	9.4	9.1
3	Max. motor shaft diameter	mm	8	10	10	8	10	8	8
	Part Numbers			223085	223091	223096	223101	223106	223111
1	Reduction			19:1	66:1	126:1	230:1	394:1	756:1
2	Absolute reduction			169/9	1183/18	126	8281/36	1183/3	756
10	Mass inertia	gcm²		9.5	16.7	16.4	16.8	16.7	16.4
3	Max. motor shaft diameter	mm		8	10	10	10	10	10
	Part Numbers			223086	223092	223098	223102	223107	223112
1	Reduction			21:1	74:1	156:1	257:1	441:1	936:1
2	Absolute reduction			21	147/2	156	1029/4	441	936
10	Mass inertia	gcm²		16.5	17.2	9.1	17.3	16.5	9.1
3	Max. motor shaft diameter	mm		10	10	8	10	10	8
	Part Numbers			223087	223093		223103	223108	
1	Reduction			26:1	81:1		285:1	488:1	
2	Absolute reduction			26	2197/27		15379/54	4394/9	
10	Mass inertia	gcm²		9.1	9.4		16.7	9.4	
3	Max. motor shaft diameter	mm		8	8		10	8	
4	Number of stages		1	2	3	3	4	4	4
5	Max. continuous torque	Nm	4	15	30	30	30	30	30
6	Max. intermittent torque at gear output	Nm	6	22.5	45	45	45	45	45
7	Max. efficiency	%	91	83	75	75	68	68	68
8	Weight	g	460	620	770	770	920	920	920
9	Average backlash no load	°	0.6	0.8	1.0	1.0	1.0	1.0	1.0
11	Gearhead length L1	mm	49.0	65.0	78.5	78.5	92.0	92.0	92.0



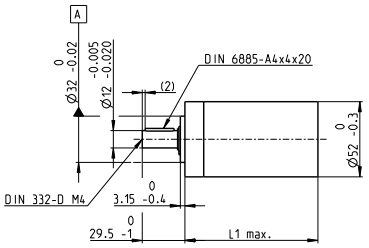
maxon Modular System

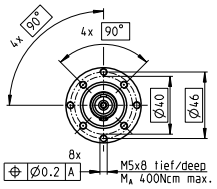
+ Motor	Page	+ Sensor	Page	Brake	Page	Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts						
RE 40, 150 W	151					120.1	136.1	149.6	149.6	163.1	163.1	163.1
RE 40, 150 W	151	MR	479			131.5	147.5	161.0	161.0	174.5	174.5	174.5
RE 40, 150 W	151	HED_ 5540	486/489			140.8	156.8	170.3	170.3	183.8	183.8	183.8
RE 40, 150 W	151	HEDL 9140	493			174.1	190.1	203.6	203.6	217.1	217.1	217.1
RE 40, 150 W	151			AB 28	535	156.2	172.2	185.7	185.7	199.2	199.2	199.2
RE 40, 150 W	151			AB 28	536	164.2	180.2	193.7	193.7	207.2	207.2	207.2
RE 40, 150 W	151	HED_ 5540	486/489	AB 28	535	173.4	189.4	202.9	202.9	216.4	216.4	216.4
RE 40, 150 W	151	HEDL 9140	493	AB 28	536	184.6	200.6	214.1	214.1	227.6	227.6	227.6
RE 50, 200 W	152					157.1	173.1	186.6	186.6	200.1	200.1	200.1
RE 50, 200 W	152	HED_5540	487/489			177.8	193.8	207.3	207.3	220.8	220.8	220.8
RE 50, 200 W	152	HEDL 9140	494			219.5	235.5	249.0	249.0	262.5	262.5	262.5
RE 50, 200 W	152			AB 44	540	219.5	235.5	249.0	249.0	262.5	262.5	262.5
RE 50, 200 W	152	HEDL 9140	494	AB 44	540	232.5	248.5	262.0	262.0	275.5	275.5	275.5
EC 40, 170 W	239					129.1	145.1	158.6	158.6	172.1	172.1	172.1
EC 40, 170 W	239	HED_5540	487/489			152.5	168.5	182.0	182.0	195.5	195.5	195.5
EC 40, 170 W	239	Res 26	496			156.3	172.3	185.8	185.8	199.3	199.3	199.3
EC 40, 170 W	239			AB 32	537	171.8	187.8	201.3	201.3	214.8	214.8	214.8
EC 40, 170 W	239	HED_5540	487/489	AB 32	537	190.2	206.2	219.7	219.7	233.2	233.2	233.2
EC 45, 150 W	240					160.3	176.3	189.8	189.8	203.3	203.3	203.3
EC 45, 150 W	240	HEDL 9140	493			175.9	191.9	205.4	205.4	218.9	218.9	218.9
EC 45, 150 W	240	Res 26	496			160.3	176.3	189.8	189.8	203.3	203.3	203.3
EC 45, 150 W	240			AB 28	536	167.7	183.7	197.2	197.2	210.7	210.7	210.7
EC 45, 150 W	240	HEDL 9140	493	AB 28	536	184.7	200.7	214.2	214.2	227.7	227.7	227.7
EC 45, 250 W	241					193.1	209.1	222.6	222.6	236.1	236.1	236.1
EC 45, 250 W	241	HEDL 9140	493			208.7	224.7	238.2	238.2	251.7	251.7	251.7
EC 45, 250 W	241	Res 26	496			193.1	209.1	222.6	222.6	236.1	236.1	236.1

B.2 Maxon motor gear

Planetary Gearhead GP 52 C ∅52 mm, 4.0–30.0 Nm

Ceramic Version

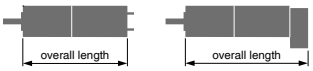




M 1:4

Technical Data				
Planetary Gearhead				straight teeth
Output shaft				stainless steel
Bearing at output				preloaded ball bearings
Radial play, 12 mm from flange				max. 0.06 mm
Axial play at axial load				0 mm
	< 5 N			max. 0.3 mm
	> 5 N			
Max. axial load (dynamic)				200 N
Max. force for press fits				500 N
Direction of rotation, drive to output				=
Max. continuous input speed				6000 rpm
Recommended temperature range				-15...+80°C
Extended range as option				-40...+100°C
Number of stages	1	2	3	4
Max. radial load, 12 mm from flange	420 N	630 N	900 N	900 N

Stock program
Standard program
Special program (on request)



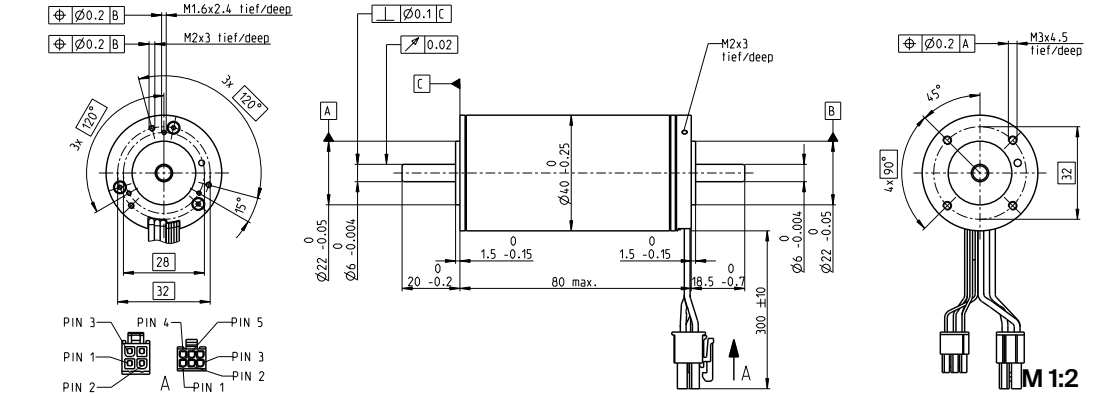
Part Numbers						
223080	223083	223089	223094	223097	223104	223109
223081	223084	223090	223095	223099	223105	223110
	223085	223091	223096	223101	223106	223111
	223086	223092	223098	223102	223107	223112
	223087	223093		223103	223108	

maxon Modular System

+ Motor	Page	+ Sensor	Page	Brake	Page	Overall length [mm] = Motor length + gearhead length + (sensor/brake) + assembly parts						
EC 45, 250 W	241			AB 28	536	200.5	216.5	230.0	230.0	243.5	243.5	243.5
EC 45, 250 W	241	HEDL 9140	493	AB 28	536	217.5	233.5	247.0	247.0	260.5	260.5	260.5
EC-max 40, 120 W	253					137.1	153.1	166.6	166.6	180.1	180.1	180.1
EC-max 40, 120 W	253	MR	479			152.8	168.8	182.3	182.3	195.8	195.8	195.8
EC-max 40, 120 W	253	HEDL 5540	490			160.5	176.5	190.0	190.0	203.5	203.5	203.5
EC-max 40, 120 W	253			AB 28	534	171.5	187.5	201.0	201.0	214.5	214.5	214.5
EC-max 40, 120 W	253	HEDL 5540	490	AB 28	534	189.8	205.8	219.3	219.3	232.8	232.8	232.8
EC-i 52, 180 W	278					129.1	145.1	158.6	158.6	172.1	172.1	172.1
EC-i 52, 180 W	278	16 EASY/Abs.	464/468			142.8	158.8	172.3	172.3	185.8	185.8	185.8
EC-i 52, 180 W	278	16 RIO	481			142.8	158.8	172.3	172.3	185.8	185.8	185.8
EC-i 52, 180 W	278	AEDL 5810	484/485			151.9	168.9	181.4	181.4	194.9	194.9	194.9
EC-i 52, 180 W	278	HEDL 5540	488-492			151.9	168.9	181.4	181.4	194.9	194.9	194.9
EC-i 52, 200 W	279					159.1	175.1	188.6	188.6	202.1	202.1	202.1
EC-i 52, 200 W	279	16 EASY/XT/Abs.	464–469			172.8	188.8	202.3	202.3	215.8	215.8	215.8
EC-i 52, 200 W	279	16 EASY Abs. XT	471			173.3	189.3	202.8	202.8	216.3	216.3	216.3
EC-i 52, 200 W	279	16 RIO	482			172.8	188.8	202.3	202.3	215.8	215.8	215.8
EC-i 52, 200 W	279	AEDL 5810	484/485			181.9	198.9	211.4	211.4	224.9	224.9	224.9
EC-i 52, 200 W	279	HEDL 5540	488-492			181.9	198.9	211.4	211.4	224.9	224.9	224.9
EC 60 flat, 100 W	304					89.8	105.8	119.3	119.3	132.8	132.8	132.8
EC 60 flat, 100 W	304	MILE	460			90.8	106.8	120.3	120.3	133.8	133.8	133.8
EC 60 flat, 150 W	305					89.8	105.8	119.3	119.3	132.8	132.8	132.8
EC 60 flat, 150 W	305	MILE	460			90.8	106.8	120.3	120.3	133.8	133.8	133.8
EC 60 flat, 200 W	306					97.6	113.6	127.1	127.1	140.6	140.6	140.6
EC 60 flat, 200 W	306	MILE	460			98.6	114.6	128.1	128.1	141.6	141.6	141.6
EC 90 flat, 160 W	307					74.5	90.5	103.5	103.5	117.0	117.0	117.0
EC 90 flat, 160 W	307	MILE	463			75.0	91.0	104.0	104.0	117.5	117.5	117.5
EC 90 flat, 220 W	308					74.5	90.5	103.5	103.5	117.0	117.0	117.0
EC 90 flat, 220 W	308	MILE	463			75.0	91.0	104.0	104.0	117.5	117.5	117.5
EC 90 flat, 360 W	309					87.0	103.0	116.0	116.0	129.5	129.5	129.5
EC 90 flat, 360 W	309	MILE	463			87.5	103.5	116.5	116.5	130.0	130.0	130.0
EC 90 flat, 260 W	310					87.0	103.0	116.0	116.0	129.5	129.5	129.5
EC 90 flat, 260 W	310	MILE	463			87.5	103.5	116.5	116.5	130.0	130.0	130.0
EC 90 flat, 400 W	311					87.0	103.0	116.0	116.0	129.5	129.5	129.5
EC 90 flat, 400 W	311	MILE	463			87.5	103.5	116.5	116.5	130.0	130.0	130.0
EC 90 flat, 600 W	312					99.5	115.5	128.5	128.5	142.0	142.0	142.0
EC 90 flat, 600 W	312	MILE	463			100.0	116.0	129.0	129.0	142.5	142.5	142.5

B.3 Maxon motor

EC 40 ∅40 mm, brushless, 170 watt



Stock program
Standard program
Special program (on request)

Motor Data (provisional)			Part Numbers 369146	393023	393024	393025
Values at nominal voltage						
1	Nominal voltage	V	15	24	42	48
2	No load speed	rpm	9840	9840	10100	9840
3	No load current	mA	654	408	243	204
4	Nominal speed	rpm	9090	9130	9380	9150
5	Nominal torque (max. continuous torque)	mNm	169	163	160	163
6	Nominal current (max. continuous current)	A	12.1	7.35	4.21	3.67
7	Stall torque	mNm	2620	2660	2740	2760
8	Stall current	A	181	115	69.1	59.6
9	Max. efficiency	%	89	89	89	89
Characteristics						
10	Terminal resistance phase to phase	Ω	0.0829	0.209	0.608	0.806
11	Terminal inductance phase to phase	mH	0.0329	0.0843	0.246	0.337
12	Torque constant	mNm/A	14.5	23.2	39.6	46.4
13	Speed constant	rpm/V	659	412	241	206
14	Speed/torque gradient	rpm/mNm	3.77	3.71	3.7	3.57
15	Mechanical time constant	ms	2.12	2.09	2.08	2.01
16	Rotor inertia	gcm²	53.8	53.8	53.8	53.8

Specifications

Thermal data

17	Thermal resistance housing-ambient	5.21 K/W
18	Thermal resistance winding-housing	1.05 K/W
19	Thermal time constant winding	18.7 s
20	Thermal time constant motor	1910 s
21	Ambient temperature	-40...+100°C
22	Max. winding temperature	+155°C

Mechanical data (preloaded ball bearings)

23	Max. speed		18000 rpm
24	Axial play at axial load	< 9 N	0 mm
		> 9 N	max. 0.14 mm
25	Radial play		preloaded
26	Max. axial load (dynamic)		23 N
27	Max. force for press fits (static)		106 N
	(static, shaft supported)		5500 N
28	Max. radial load, 5 mm from flange		75 N

Other specifications

29	Number of pole pairs	1
30	Number of phases	3
31	Weight of motor	580 g

Values listed in the table are nominal.

Connection motor (Cable AWG 16)

red	Motor winding 1	Pin 1
black	Motor winding 2	Pin 2
white	Motor winding 3	Pin 3
	N.C.	Pin 4

Connector Molex **Part number** 39-01-2040

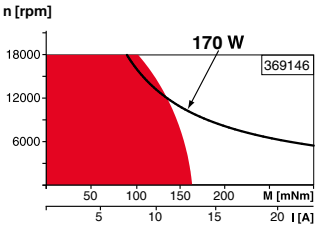
Connection sensors (Cable AWG 26)

yellow	Hall sensor 1	Pin 1
brown	Hall sensor 2	Pin 2
grey	Hall sensor 3	Pin 3
blue	GND	Pin 4
green	V_{Hall} 3...24 VDC	Pin 5
	N.C.	Pin 6

Connector Molex **Part number** 430-25-0600

Wiring diagram for Hall sensors see p. 57

Operating Range



Comments

Continuous operation
In observation of above listed thermal resistance (lines 17 and 18) the maximum permissible winding temperature will be reached during continuous operation at 25°C ambient.
= Thermal limit.

Short term operation
The motor may be briefly overloaded (recurring).

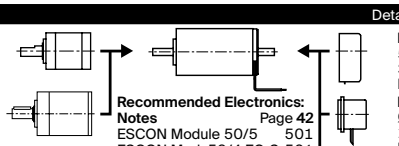
Assigned power rating

maxon Modular System

Details on catalog page 42

Planetary Gearhead
∅42 mm
3 - 15 Nm
Page 405

Planetary Gearhead
∅52 mm
4 - 30 Nm
Page 410



Recommended Electronics:

Notes	Page 42
ESCON Module 50/5	501
ESCON Mod. 50/4 EC-S	501
ESCON Mod. 50/8 (HE)	502
ESCON 50/5	503
ESCON 70/10	503
DEC Module 50/5	505
EPOS4 Mod./Comp. 50/5	510
EPOS4 Mod./Comp. 50/8	511
EPOS4 Mod./Comp. 50/15	511
EPOS4 50/5	515
EPOS4 70/15	515
EPOS4 Disk 60/8	516
EPOS4 Disk 60/12	517
EPOS2 P 24/5	520

Encoder HED_5540
500 CPT,
3 channels
Page 487/489

Resolver Res 26
∅26 mm
10 V
Page 496

Brake AB 32
24 VDC
0.4 Nm
Page 537

C Appendix: Extra sections

C.1 Camera calibration and image processing steps

C.1.1 Calibration of camera

Before each test, the camera is calibrated using the Camera Calibration Tool (CCT) in MATLAB. Calibration is performed using a checkerboard pattern and capturing 20 images in different locations around the plane of interest. The CCT estimates the intrinsic, extrinsic and lens distortion parameters. This can be used to remove the effects of lens distortion from the video captured in order to reduce position bias across the captured 2D image of each frame [24]. The pattern used for the calibration can be seen in figure C.1.

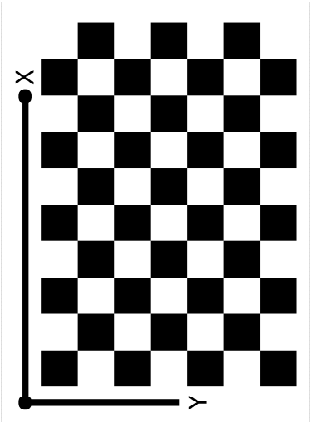


Figure C.1: Checkerboard pattern for camera calibration, containing x and y axis

C.1.2 Image processing

When the footage is recorded and stored, it must go through an image processing algorithm. The raw footage is loaded into the algorithm and then each frame is processed. The algorithm performed on each frame contains the following steps.

Frame calibration
Using the calibration data, each frame is undistorted, and a new origin is set. All pixel

coordinates are converted to world coordinates.

Image segmentation and blob analysis
By identifying clustering regions of pixels in the frame, circles within a desired range of diameter can be detected. The detection algorithm is set to detect a maximum of eight circles ranging from 10 to 75 pixels. With the circles detected, the image is masked and segmented into a binary image, and a blob analysis is performed on the binary image. Just as in the segmentation algorithm, the blob analysis detects clusters inside the binary image and outputs the centroids and areas of the blobs. The detected circles are sorted and indexed based on the size of the area.

Setting local and global origins and coordinate conversions
The global origin of the image is determined by the centroid of the circle located in the center of the test member. For each of the detected wheels the centers of these are set as local origins. The detected Cartesian xy coordinates are converted into polar coordinates by $\theta = \tan\left(\frac{y}{x}\right)^{-1}$ and radius $\rho = \sqrt{x^2 + y^2}$. The following angular positions and radii are calculated:

- Radius and angular position of the crawler wrt. global origin
- Radius and angular position of each wheel wrt. global origin
- Angular position of each wheel wrt. local origins

Estimation of angular velocities
The angular velocity ($\dot{\theta}_c$ is the derivative of the position and is estimated using the Symmetric Quotient Difference seen in equation C.1. This equation was chosen for its compromise in accuracy and computational speed.

$$\dot{\theta}_c(t) = \frac{-\theta(t+2) + 8\theta(t+1) - 8\theta(t-1) + \theta(t-2)}{12Ts} \tag{C.1}$$

Where t is the current time and Ts is the time duration between each step.

C.2 PID linearization

The wheel accelerations can be described using the crawler acceleration, which can be inserted into the particular wheel models 7.7, 7.8 and 7.9. The reformulation can be seen in equation C.2.

$$\dot{\omega}_i = \frac{\ddot{\theta}_c L_2}{R_w} \tag{C.2}$$

The same is the case for the wheel velocity, which can be described using the crawler velocity. This is seen in equation C.3.

$$\omega_i = \frac{\dot{\theta}_c L_2}{R_w} \tag{C.3}$$

The wheel models can be rearranged for their specific traction forces $F_{t,i}(\lambda_i, N_i)$ as seen in equation C.4.

$$F_{t,i}(\lambda_i, N_i) = \frac{J_w \left(\frac{\ddot{\theta}_c L_2}{R_w}\right) + b_i \left(\frac{\dot{\theta}_c L_2}{R_w}\right) - T_{m,i} G_1 G_2}{-R_w} \tag{C.4}$$

The substitutions for F_{roll} and F_g are inserted into the crawler rotation equation such as the substitution for each wheel $F_{t,i}(\lambda_i, N_i)$ from equation C.4.
The motor torques including their gear for each wheel can be redefined as seen in equation C.5.

$$T_{in} = (T_{m,1} + T_{m,2} + T_{m,3}) G_1 G_2 \tag{C.5}$$

Where it is assumed that $T_{m,1} = T_{m,2} = T_{m,3}$ and T_{in} is the total wheel torque.

For the operational point when the crawler is positioned horizontally, $\cos(0) = 1$ and the nonlinear gravity term remains.

When inserting all substitutions for the different forces acting on the crawler, it is linearized using the Taylor series around an operational point of 0°.

C.3 LQR linearization

The four non-linear equations need to be linearized in order to be introduced to the linear state-space system. The nonlinear elements in the system equations are found in the gravitational term and in the traction forces terms. The gravitational term is linearized as seen in section 10.1.2, and the traction forces are neglected in this model, due to their high non-linearity. In order to still include the dynamics of the wheels moving the crawler, the input forces are set to affect the crawler as well as the wheels.
The parameters of the model are based on the developed parameters found in table 9.5, where the crawler is expected to operate on a horizontal oriented member with a rough dry painted surface. When choosing this orientation of the member and including the gravitational term, ensures that the controller is robust enough to handle this disturbance. The crawler is set to operate at a piston pressure of 50 bar, which is approximately 10 kN. This is chosen since it is the highest pressure under which the crawler is able to counter the obstacle on the test member. The operation points used for linearization can be found in table C.1. Dynamics such as the gravitational force and rolling friction that are neglected in the linear model are added as a feed-forward term ($u_{ff}(t)$) when implementing the controller, as seen in figure 10.7.

Table C.1: Operation points used for linearization of the dynamic crawler equations

θ_C	F_p	α
0 rad	10 kN	0 rad

The linearized state-space model is compared open-loop with the validated nonlinear model seen in 9.2.2. A comparison of velocities and positions is shown in figure C.2.

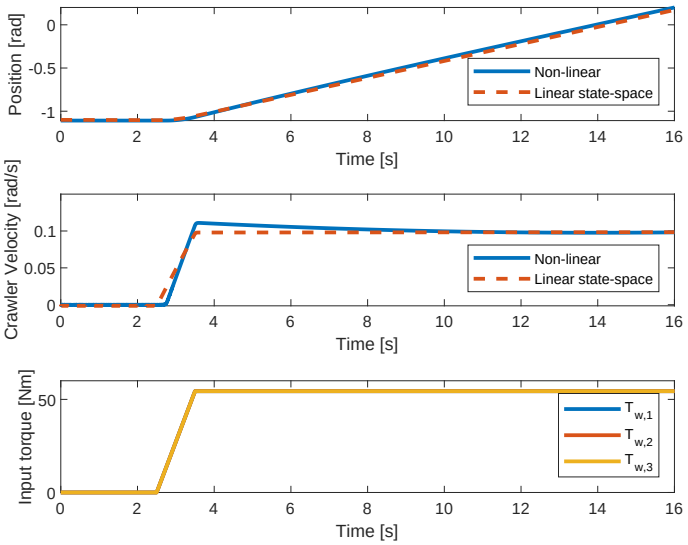


Figure C.2: Comparison of linear and nonlinear crawler rotationel models

Both models are compared with the crawler operating on a horizontal oriented member, a piston pressure of 50 bar, and initialized at -1.1 radians. As seen in figure C.2 top graph, the end positions of the nonlinear and linear models are 0.198 and 0.170 rad, respectively, which gives an end deviation of 0.028 radians. The middle graph shows the crawler velocity and here it is seen that the nonlinear model has a slightly higher initial response than the linear model, with a peak velocity of 0.111 rad/s of the nonlinear model and 0.098 rad/s of the linear model, around 9 seconds the nonlinear model reaches the same velocity as the linear model. The differences in velocities and positions are caused by the linearized gravitational parts, which increases the impact of gravity far more on the linear model compared to the nonlinear. The difference between the response of the two models is within a magnitude that is acceptable, thus the linear state-space model is accepted, and can be used to develop the LQR controller.

The system, input and output matrices of the state-space model are found in equation A.1, A.2, and A.3, respectively, all located in the Appendix.

C.3.1 Stability analysis

The eigenvalues of the **A** matrix can be used to determine the stability of the system. The eigenvalues can be identified by solving equation C.6.

$$\det(\boldsymbol{\lambda_p} \boldsymbol{I} - \boldsymbol{A}) = 0 \tag{C.6}$$

Where **I** is the identity matrix and $\boldsymbol{\lambda_p}$ is a vector that contains the eigenvalues. This results in the following five eigenvalues.

$$\boldsymbol{\lambda_p} = \textbf{-47.1668, -43.9690, -43.0041, -27.3136, 0.0946}$$

The system has four eigenvalues located in the left half-plane of the imaginary axis and one in the right half-plane. With one eigenvalue located on the right half-plane the linear system is unstable.

C.3.2 Controllability

The controllability of the system is determined by calculating the controllability matrix seen in C.7. If rank($\mathbf{C}_o$) = length($\mathbf{A}$) then the system is controllable [32].

$$\boldsymbol{C_o} = \begin{bmatrix} \boldsymbol{A} & \boldsymbol{AB} & \cdots & \boldsymbol{A^{n-1}B} \end{bmatrix} \tag{C.7}$$

Where n denotes the length of $\mathbf{A}$. The system was found to be controllable.

D Appendix: Programming

D.1 Piston control - Arduino code

The code to control the hydraulic piston, and measure its current pressure. It communicates with the main microcontroller the Jetson Nano through a serial USB port. The Jetson publishes via ROS what is the desired pressure, which is determined from the control algorithm 10.4, and then prossed through the propotional controller. The Arduino subscribes to this topic and then changes the opening on the hydraulic valve from -1 to 1. Which is fully opened and full closed. In reality its not fully open, since the voltage applied is not high/low enough 5.2.3. The Arduino then measures the voltage over the resistor which together with the correlation 5.7 gives the pressure in the piston. Which is published and subscribed to from the Jetson Nano.

In order to get the Arduino due to work, several aspects are required. The same ros library has to be installed in the Arudino IDE, as is on the Jetson Nano. A delay is required in order to prevent the Arduino being overflown resulting in badly read messages which causes it to turn off. Through trial and error, a delay of 5ms with the Jetson Nano running at 300Hz prevented this issue. The Arduino should also not have functions doing calculations since, it can delay the response time. Therefore the moving average is moved from Arduino to Jetson Nano. Since this is on an Arduino Due, which has 2 usb ports a native and a programing port, the library USBCON is required. A baudrate of 250000 has been chosen.

```
#if (ARDUINO >= 100)
#include <Arduino.h>
#else
#include <WProgram.h>
#endif


#define USBCON
#define USE_USBCON
#include <ros.h>
#include <Wire.h>
#include <std_msgs/Float32.h>
#include <std_msgs/UInt16.h>

```

```
unsigned long startcountthingy= 0;
unsigned long endcountthingy=0;
float pistondata = 0;
int PistonanalogPin = DAC1;
float pistonout=0;
/*

float movingAverage(float value) {
  const byte nvalues = 100;             // Moving average window size

  static byte current = 0;            // Index for current value
  static byte cvalues = 0;            // Count of values read (<= nvalues)
  static float sum = 0;               // Rolling sum
  static float values[nvalues];

  sum += value;

  // If the window is full, adjust the sum by deleting the oldest value
  if (cvalues == nvalues)
    sum -= values[current];

  values[current] = value;          // Replace the oldest with the latest

  if (++current >= nvalues)
    current = 0;

  if (cvalues < nvalues)
    cvalues += 1;

  return sum / cvalues;
}
*/
void pistoncallback(const std_msgs::Float32& pistonmsg)
{
  startcountthingy=millis();
  float pistondata = pistonmsg.data;
  if (pistondata < -1){
    pistondata=-1;
  }
  else if (pistondata >1){
    pistondata=1;
  }
  float pistonout=3000-0.27*1000+pistondata*1000;
    analogWrite(PistonanalogPin, pistonout); // analogRead values go from 0
     to 1023, analogWrite values from 0 to 255
}

std_msgs::Float32 pressure_clamp;

std_msgs::Float32 piston;


ros::Publisher pub_pressure_clamp("pressure_clamp", &pressure_clamp); //
```

```
    Pressure clamp publisher
67
68 ros::Subscriber<std_msgs::Float32> sub_Piston_ref_pressure("
    Piston_ref_pressure", &pistoncallback);
69
70 ros::NodeHandle nh;
71
72 void setup() {
73   // put your setup code here, to run once:
74   //Serial.print("In setup");
75   analogWriteResolution(12);
76   analogReadResolution(12);
77   Serial.begin(250000);  // Det var kommenteret ud f r.
78   nh.getHardware()->setBaud(250000);
79   Wire.begin();
80   // Wait until serial port is opened
81
82   pinMode(PistonanalogPin, OUTPUT);  // sets the pin as output
83
84   nh.initNode();
85
86   ros::NodeHandle n;
87   nh.advertise(pub_pressure_clamp);
88
89   nh.subscribe(sub_Piston_ref_pressure);
90   analogWrite(PistonanalogPin, 3000-0.27*1000);
91 }
92
93 void loop() {
94   // put your main code here, to run repeatedly:
95   pressure_clamp.data = analogRead(A0)*0.0526-74.5591;//*0.4748*16-92.0799;
       // Read the analog  value
96   pub_pressure_clamp.publish(&pressure_clamp);
97
98   nh.spinOnce();
99   endcountthingy=millis();
100  unsigned long HowLongThingyDetected = startcountthingy - endcountthingy;
101  if  (HowLongThingyDetected > 0.1*1000){
102      analogWrite(PistonanalogPin, 3000-0.27*1000);
103
104  }
105
106 //Delay required to stop Arduino from crashing
107  delay(5);
108
109 }
```

Listing D.1: Arduino code controlling the hydraulic piston

D.2 LQR GS SCS - python controller

The code that controls the Crawler is written in Python, though would have been better to write in C++ for faster computational time. The script incorporates the LQR controller 10.2, the robust slip controller 10.3.5, and the hydraulic controller 10.4. It takes, all the sensor data it gathered to first find its current angle from the IMU, using the gravity to find its position. This is able since the IMU is sitting on the robot rotating around the pipe with it. It then uses the found states compares it to the desired states which is then multiplied with the Nbar to remove offset.

Feedforward term is added which is very important due to its sheer size, as the gravity and rolling heavily affects the robot. The rolling friction term has been increased since viscous friction is not feedforwarded. Which is the same reason why the gain K has been increased. After the LQR torque has been calculated with feedforward, it check whether any of the wheels are slipping. And if so it uses the robust slip controller with the disturbance observer to determine the amount of torque extra that is required to prevent it. Afterwards the Hydraulic controller gets activated and checks its 6 cases, whether it should intervene and sens the signal to the Arduino.

```
#!/usr/bin/env python
# license removed for brevity
import rospy
import time
import math
import numpy
import numpy as np
import scipy.linalg
from std_msgs.msg import String, Float32, Float64
#from bluerov2_tm200.msg import Imu
from sensor_msgs.msg import Imu, Temperature, MagneticField
from geometry_msgs.msg import Wrench
import datetime
from sensor_msgs.msg import JointState
from scipy import signal
#time.sleep(6)

alpha=0

Motor_vel=0
Motor_vel1 = 0
Motor_vel2 = 0
Motor_vel3 = 0
time = 0
time_prev = datetime.datetime.now()

timeprevHyd=datetime.datetime.now()

lambda_cd=0
tau_c1=0
```

```
tau_c2=0
tau_c3=0

z=0

xacc=0
yacc=0
zacc=0
zvel=0

Slipold1=0
Slipold2=0
Slipold3=0

slip1State=0
slip2State=0
slip3State=0

tau1cmd=0
tau2cmd=0
tau3cmd=0
tau1=0
tau2=0
tau3=0
tau_old1=0
tau_old2=0
tau_old3=0


Piston_ref_pressure=70
pressure_clamp =0
integral_pold=0
time_prev_Piston=datetime.datetime.now()
integral_old_Piston=0
integral_old_piston=0


zvelold=0.01
timeprevslip=datetime.datetime.now()
time_prev_Piston=datetime.datetime.now()
timeprev = datetime.datetime.now()
timenow = 0
deltaT = 0
timediff =0
theta_c=0
theta_cold=0
theta_cinit=1000000000001
dtheta_c=0
K=0


Slip1=0
Slip2=0
```

```
84 Slip3=0
85
86 slipcut = 0.15
87 deltaFp = 1000
88 beta = np.deg2rad(120)
89 mu1=0.37
90 mu2=0.37
91 mu3=0.37
92
93 rw=0.0625
94 Jw=0.0046
95 L2=0.508
96 cd=0.07
97 m0=132
98 J=0.0046
99 k=6
100 rho=20
101 epsilon=0.03
102 g=9.82
103 L1=0.2703
104
105 L2B=99.346*pow(10,-3)
106 L1A=174.049*pow(10,-3)
107 LLA=89.160*pow(10,-3)
108 LRB=90.369*pow(10,-3)
109 LRC=240.228*pow(10,-3)
110 LLC=129.792*pow(10,-3)
111 LPC=186.212*pow(10,-3)
112 L3C=157.372*pow(10,-3)
113 Fg=m0*g
114
115 slipgain=1
116 mur=0.039
117 a=40
118
119 B1 = 91.87
120 C1 = 0.0655
121 D1 = 2.731
122 E1 = 1.0130
123
124 B2 = 8.9192
125 C2 = 1.936
126 D2 = 0.1285
127 E2 = 0.7862
128
129 B3 = 968.1
130 C3 = 0.9762
131 D3 = 0.19
132 E3 = 1.001
133
134 tau_w1old=0
135 tau_w2old=0
136 tau_w3old=0
```

```
tau_dhat1old=0
tau_dhat2old=0
tau_dhat3old=0


# Normalize the frequency

#cutoff=33/(2*math.pi)
fs=100
fc=33
w = fc / (fs / 2)
order=1
dt=1/fs
gamma= (2 * math.pi * dt * fc) / (2 * math.pi * dt * fc + 1)


pressure_clamp =0
P_ref=63
Fp=(pow(10,5)*((0.05*0.05)/4)*math.pi)*P_ref

x_error = np.array([0,0,0])
# The K feedback control matrix
K_pos=np.array([[57.7350, 2.002,0,0,0],[57.7350, 2.002,0,0,0],[57.7350,
    2.002,0,0,0]])
#Nbar precomputation
Nbar_pos=np.array([[56.6030, 0.9813, 1.1313, 0.004, 0.0004],[56.6030,
    0.9823, 0.004, 1.1077, 0.0003],[56.6030, 0.9852, 0.0002, 0.002,
    1.0361]])


# 0.016rad/s refence
#K_vel=np.array([[0.12159,  18240.88,  0, 0, 0],[0.12159,  18240.88,  0, 0,
     0],[0.12159,  18240.88,  0, 0, 0]])
#Nbar_vel=np.array([[0.12159,  18240.88,  0, 0, 0],[0.12159,  18240.88,  0,
     0, 0],[0.12159,  18240.88,  0, 0, 0]])

# 0.048 rad/s
K_vel=np.array([[0,  43.5113,  0, 0, 0],[0,  43.5113,  0, 0, 0],[0,
    43.5113,  0, 0, 0]])
Nbar_vel=np.array([[0.0388,  48.788,  76.8653, -9.8341, -10.0279],[-0.0430,
     49.3637,  -10.1390, 77.5715, -11.1204],[0.0577,  51.3605,  -13.5983,
    -13.9033, 79.4191]])


def imu_out(imu_data):
  global xacc, yacc, zacc, zvel,theta_c,deltat,dtheta_c,theta_cold
  xacc = -9.81*imu_data.linear_acceleration.x # cant measure roll pitch
    same reference frame
  yacc = -9.81*imu_data.linear_acceleration.y
  zacc = -9.81*imu_data.linear_acceleration.z
  zvel = imu_data.angular_velocity.z

def Intizial_angle():
```

```
  global theta_cinit,theta_cold
  theta_cinit= np.arcsin(-xacc/9.81)
  theta_cold=theta_cinit

def angle_cal():
  global xacc, yacc, zacc, zvel,theta_c,deltat,dtheta_c,theta_cold,
    theta_cinit,zvelold
  timenow = datetime.datetime.now()
  timediff = timenow-timeprev
  deltaT = timediff.total_seconds()
  dtheta_c=zvel

  if (theta_cinit<0.02) and (theta_cinit> -0.02):
    Intizial_angle()
  theta_c=np.arcsin(-xacc/9.81)
  theta_cold=theta_c



def lpf(filtsignal, omega_c, T):
    y = x
    alpha1 = (2 - T * omega_c1) / (2 + T * omega_c1)
    beta1 = T * omega_c1 / (2 + T1 * omega_c1)
    for k1 in range(1, N):
        y[k] = alpha1 * y[k - 1] + beta1 * (x1[k] + x1[k - 1])
    return y


def moving_average(value, nvalues=5):
    moving_average.values = getattr(moving_average, 'values', [0] * nvalues
    )
    moving_average.current = getattr(moving_average, 'current', 0)
    moving_average.cvalues = getattr(moving_average, 'cvalues', 0)
    moving_average.sum = getattr(moving_average, 'sum', 0)

    moving_average.sum += value

    if moving_average.cvalues == nvalues:
        moving_average.sum -= moving_average.values[moving_average.current]

    moving_average.values[moving_average.current] = value
    moving_average.current = (moving_average.current + 1) % nvalues

    if moving_average.cvalues < nvalues:
        moving_average.cvalues += 1

    return moving_average.sum / moving_average.cvalues

def pressure_clamp_out(pressure_clamp_msg):
    global pressure_clamp
    Pressure_piton_avg_publisher = rospy.Publisher('Pressure_piston_avg',
    Float64, queue_size=10)

```

```
    pressure_clamp=pressure_clamp_msg.data
    Pressure_piton_avg_publisher.publish(pressure_clamp)



def PiControllerPiston(Piston_ref):
    global integral1old, integral2old, integral3old, time_prev,
    time_prev_Piston, integral_old_piston
    Piston_ref_pressure = rospy.Publisher('/Piston_ref_pressure', Float32,
    queue_size=10)
    Piston_pressure_reference_bars = rospy.Publisher('
    Piston_pressure_reference_bars', Float64, queue_size=10)
    Piston_pressure_reference_bars.publish(Piston_ref)
    Ki_p =-0.000
    Kp_piston=-0.0278*2
    e_piston=float((Piston_ref-pressure_clamp))
    timePiston = datetime.datetime.now()
    timediffPiston = timePiston-time_prev_Piston
    deltaTPiston = timediffPiston.total_seconds()
    P_piston=Kp_piston*e_piston
    integral_piston = integral_old_piston + Ki_p*e_piston*(deltaTPiston)
    e_msg= P_piston +integral_piston

    if e_msg < -0.01:
        e_msg=e_msg-0.44
        Piston_ref_pressure.publish(e_msg)

    elif e_msg > 0.01:
        e_msg=e_msg+0.44
        Piston_ref_pressure.publish(e_msg)

    #else:
    #    e_msg.data=e_piston
    #    Piston_ref_pressure.publish(e_msg)
    time_prev_Piston = timePiston
    integral_old_piston=integral_piston


def Normal_force_cal():
  global N1,N2,N3, Fp

  A =np.array([[ 0, L3C, 0, -LLC, LRC ],[-L1A, 0, 0, -LLA, 0],[0, 0, L2B,
    0, -LRB],[math.sin(theta_c + (2*math.pi)/3), math.sin(theta_c), math.
    sin(theta_c - (2*math.pi)/3), 0, 0],[math.cos(theta_c + (2*math.pi)/3),
     math.cos(theta_c), math.cos(theta_c - (2*math.pi)/3), 0, 0]])
  B =np.array([[Fp*LPC],[0],[0],[Fg*math.cos(alpha)], [0]])
  NormalForces = np.linalg.solve(A, B)
  N1=NormalForces[0]
  N2=NormalForces[1]
  N3=NormalForces[2]
```

```
278 def slip_cal():
279   global Slip1, Slip2, Slip3,slipgain,lambda_cd,lambda1app,Motor_vel1,
        Motor_vel2,Motor_vel3,dtheta,slip1State,slip2State,slip3State,
        timeprevslip,a,Slipold1,Slipold2,Slipold3,B1,C1,D1,E1,B2,C2,D2,E2,B3,C3
        ,D3,E3,tau_w1old,tau_w2old,tau_w3old,tau_dhat1old,tau_dhat2old,
        tau_dhat3old,gamma,tau_c1,tau_c2,tau_c3,tau_old1,tau_old2,tau_old3
280   Slip_publisher_motor1 = rospy.Publisher('slip_motor1', Float64,
        queue_size=10)
281   Slip_publisher_motor2 = rospy.Publisher('slip_motor2', Float64,
        queue_size=10)
282   Slip_publisher_motor3 = rospy.Publisher('slip_motor3', Float64,
        queue_size=10)
283
284   Slip_publisher_motor1_effort = rospy.Publisher('slip_motor_effort1',
        Float64, queue_size=10)
285   Slip_publisher_motor2_effort = rospy.Publisher('slip_motor_effort2',
        Float64, queue_size=10)
286   Slip_publisher_motor3_effort = rospy.Publisher('slip_motor_effort3',
        Float64, queue_size=10)
287
288   v=-dtheta_c*L2+0.00000001
289   omega1=Motor_vel1*rw+0.00000001
290   omega2=Motor_vel2*rw+0.00000001
291   omega3=Motor_vel3*rw+0.00000001
292
293
294   if abs(v/omega1)<0.1 or abs(v/omega2)<0.1 or abs(v/omega3)<0.1:
295     a=0.1
296     #Prevent math overflow
297   else:
298     a=40
299
300   sigma1=(1/(1+math.exp(-a*((omega1)/(v)-1))))
301   lambda1app=(omega1-v)/(sigma1*omega1+(1-sigma1)*v)
302   sigma2=(1/(1+math.exp(-a*((omega2)/(v)-1))))
303   lambda2app=(omega2-v)/(sigma2*omega2+(1-sigma2)*v)
304   sigma3=(1/(1+math.exp(-a*((omega3)/(v)-1))))
305   lambda3app=(omega3-v)/(sigma3*omega3+(1-sigma3)*v)
306   Slip1=lambda1app
307   Slip2=lambda1app
308   Slip3=lambda1app
309
310   timenowSlip = datetime.datetime.now()
311   timediffSlip = timenowSlip-timeprevslip
312   deltaTSlip = timediffSlip.total_seconds()
313
314
315   dslip1 = gamma *(Slip1)  + (1 - gamma)*Slipold1
316   dslip2 = gamma *(Slip2)  + (1 - gamma)*Slipold2
317   dslip3 = gamma *(Slip3)  + (1 - gamma)*Slipold3
318   tau_w1=Motor_vel1*Jw
319   tau_w2=Motor_vel2*Jw
320   tau_w3=Motor_vel3*Jw
```

```
321
322   if (abs(Slip1)>0.1) or (dslip1>0.05) or (slip1State==1):
323     if (Slip1<0.07) or (dslip1<0.02):
324       slip1State=0
325       tau_old1=0
326       PublishForceMotor1(thrustlqr[0]+tau_old1)
327       Slip_publisher_motor1.publish(0)
328       Slip_publisher_motor1_effort.publish(tau_old1)
329     else:
330       if thrustlqr[0]>0:
331         #Slip errors
332         lambda_cd=cd
333       elif thrustlqr[0]<0:
334         lambda_cd=-cd
335       Slip_publisher_motor1.publish(1)
336       e_lambda=lambda1app-lambda_cd
337       A=math.exp(-a*((omega1/v)-1))
338       #Traction force
339       F_n1=N1*D1*math.sin(C1*math.atan(B1*Slip1-E1*(B1*Slip1-math.atan(B1*
    Slip1))))
340       #Torque from controller
341       tau_g=math.cos(theta_c)*m0*g*math.cos(alpha)*L1
342       tau_chat1 = -1*(k*J*v*pow((omega1+A*v),2))/(rw*(v*v*pow((1+A),2)-a*A*
    pow((omega1-v),2)))*e_lambda+((rw*m0*v+J*Motor_vel1)/(m0*v))*F_n1-(J*
    Motor_vel1*tau_g/L1)/(3*m0*v)-rho*((e_lambda)/(abs(e_lambda+epsilon)))
343       #disturbance observer Originally J_w = 0.5*constants.m_w*constants.
    r_w^2; new inertias are not implemented
344
345       tau_w1diff=(tau_w1-tau_w1old)/deltaTSlip
346       tau_c1=tau_chat1-tau_dhat1old
347
348       but1, afilt1 = signal.butter(1, w, 'low')
349       #Signalunfilt1=[tau_w1diff-(tau_c1-F_n1*rw),tau_dhat1old]
350       #tau_dhat1 =  signal.filtfilt(but1, afilt1, Signalunfilt1)
351       tau_dhat1 = gamma *(tau_w1diff-(tau_c1-F_n1*rw))  + (1 - gamma)*
    tau_dhat1old
352       tau_dhat1old=tau_dhat1
353       PublishForceMotor1(tau_c1+thrustlqr[0])
354       #PublishForceMotor2(tau_c1+thrustlqr[1])
355       #PublishForceMotor3(tau_c1+thrustlqr[2])
356       Slip_publisher_motor1_effort.publish(tau_c1)
357       slip1State=1
358       tau_old1=tau_c1
359   else:
360     tau_old1=0
361     PublishForceMotor1(tau_c1+thrustlqr[0]+tau_old1)
362     Slip_publisher_motor1.publish(0)
363     Slip_publisher_motor1_effort.publish(tau_old1)
364
365   if (abs(Slip2)>0.1) or (dslip2>0.05) or (slip2State==1):
366     if (Slip2<0.07) or (dslip2<0.02):
367       slip2State=0
368       tau_old2=0
```

```
369       Slip_publisher_motor2.publish(0)
370       PublishForceMotor2(thrustlqr[1]+tau_old2)
371       Slip_publisher_motor2_effort.publish(tau_old2)
372     else:
373       if thrustlqr[1]>0:
374         #Slip errors
375         lambda_cd=cd
376       elif thrustlqr[1]<0:
377         lambda_cd=-cd
378       #print("slip controller active wheel 2")
379       Slip_publisher_motor2.publish(1)
380
381       e_lambda=lambda2app-lambda_cd
382       A=math.exp(-a*((omega2/v)-1))
383       #Traction force
384       F_n2=N2*D2*math.sin(C2*math.atan(B2*Slip2-E2*(B2*Slip2-math.atan(B2*
    Slip2))))
385       #Torque from controller
386       tau_g=math.cos(theta_c)*m0*g*math.cos(alpha)*L1
387       tau_chat2 = -(k*J*v*pow((omega2+A*v),2))/(rw*(v*v*pow((1+A),2)-a*A*
    pow((omega2-v),2)))*e_lambda+((rw*m0*v+J*Motor_vel2)/(m0*v))*F_n2-(J*
    Motor_vel2*tau_g/L1)/(3*m0*v)-rho*((e_lambda)/(abs(e_lambda+epsilon)))
388       #disturbance observer Originally J_w = 0.5*constants.m_w*constants.
    r_w^2; new inertias are not implemented
389
390       tau_w2diff=(tau_w2-tau_w2old)/deltaTSlip
391       tau_c2=tau_chat2-tau_dhat2old
392       but2, afilt2 = signal.butter(1, w, 'low')
393       #Signalunfilt2=[tau_w2diff-(tau_c2-F_n2*rw),tau_dhat2old]
394       #tau_dhat2 =  signal.filtfilt(but2, afilt2, Signalunfilt2)
395       tau_dhat2 = gamma *(tau_w2diff-(tau_c2-F_n2*rw))  + (1 - gamma)*
    tau_dhat2old
396       tau_dhat2old=tau_dhat2
397       PublishForceMotor2(tau_c2+thrustlqr[1])
398       Slip_publisher_motor2_effort.publish(tau_c2)
399       slip2State=1
400       tau_old2=tau_c2
401   else:
402     tau_old2=0
403     PublishForceMotor2(thrustlqr[1]+tau_old2)
404     Slip_publisher_motor2.publish(0)
405     Slip_publisher_motor2_effort.publish(tau_old2)
406
407   if (abs(Slip3)>0.1) or (dslip3>0.05) or (slip3State==1):
408     if (Slip3<0.07) or (dslip3<0.02):
409       slip3State=0
410       Slip_publisher_motor3.publish(0)
411       tau_old3=0
412       PublishForceMotor3(thrustlqr[2]+tau_old3)
413       Slip_publisher_motor3_effort.publish(tau_old3)
414     else:
415       if thrustlqr[2]>0:
416         #Slip errors
```

```
417         lambda_cd=cd
418       elif thrustlqr[2]<0:
419         lambda_cd=-cd
420       Slip_publisher_motor3.publish(1)
421       e_lambda=lambda3app-lambda_cd
422       A=math.exp(-a*((omega3/v)-1))
423       #Traction force
424       F_n3=N3*D3*math.sin(C3*math.atan(B3*Slip3-E3*(B3*Slip3-math.atan(B3*
    Slip3))))
425       #Torque from controller
426       tau_g=math.cos(theta_c)*m0*g*math.cos(alpha)*L1
427       #tau_g=0
428       tau_chat3 = -(k*J*v*pow((omega3+A*v),2))/(rw*(v*v*pow((1+A),2)-a*A*
    pow((omega3-v),2)))*e_lambda+((rw*m0*v+J*Motor_vel3)/(m0*v))*F_n3-(J*
    Motor_vel3*tau_g/L1)/(3*m0*v)-rho*((e_lambda)/(abs(e_lambda+epsilon)))
429       #disturbance observer Originally J_w = 0.5*constants.m_w*constants.
    r_w^2; new inertias are not implemented
430
431       tau_w3diff=(tau_w3-tau_w3old)/deltaTSlip
432       tau_c3=tau_chat3-tau_dhat3old
433       but3, afilt3 = signal.butter(1, w, 'low')
434       #Signalunfilt3=[tau_w3diff-(tau_c3-F_n3*rw),tau_dhat3old]
435       #tau_dhat3 =  signal.filtfilt(but3, afilt3, Signalunfilt3)
436       tau_dhat3 = gamma *(tau_w3diff-(tau_c3-F_n3*rw))  + (1 - gamma)*
    tau_dhat3old
437       tau_dhat3old=tau_dhat3
438       PublishForceMotor3(tau_c3+thrustlqr[2])
439       Slip_publisher_motor3_effort.publish(tau_c3)
440       slip3State=1
441       tau_old3=tau_c3
442   else:
443     tau_old3=0
444     PublishForceMotor3(thrustlqr[2]+tau_old3)
445     Slip_publisher_motor3.publish(0)
446     Slip_publisher_motor3_effort.publish(tau_old3)
447
448   #print(tau_old1,tau_old2,tau_old3)
449   tau_w1old=tau_w1
450   tau_w2old=tau_w2
451   tau_w3old=tau_w3
452   Slipold1=Slip1
453   Slipold1=Slip2
454   Slipold1=Slip3
455   timeprevslip=timenowSlip
456
457
458 #def hydraulic_controller(Fp_in, lambda1, lambda2, lambda3, tau1, tau2,
    tau3, N1, N2, N3, mu1, mu2, mu3, thetac, alpha, clock, clockold):
459 def hydraulic_controller():
460   global Fp, P_ref, timeprevHyd, condition,Slip1,Slip2,Slip3
461   Hydraulic_condition_publisher = rospy.Publisher('hydraulic_condition',
     Float64, queue_size=10)
462   condition = 0
```

```
463
464   timenowHyd = datetime.datetime.now()
465   timediffHyd = timenowHyd-timeprevHyd
466   deltaTHyd = timediffHyd.total_seconds()
467   Fp_in=Fp
468   #print("Hydraulic time "+str(deltaTHyd))
469   if deltaTHyd > 5:
470     if (abs(tau1) + abs(tau2) + abs(tau3)) > 150:
471       Fp = Fp_in - deltaFp
472       condition = 2
473       timeprevHyd =timenowHyd
474       Hydraulic_condition_publisher.publish(condition)
475     elif abs(dtheta_c)< 0.005:
476       Fp = Fp_in - deltaFp
477       condition = 5
478       timeprevHyd =timenowHyd
479       Hydraulic_condition_publisher.publish(condition)
480     elif abs(Slip1) > slipcut and abs(Slip2) > slipcut:
481       Fp = Fp_in + deltaFp
482       timeprevHyd =timenowHyd
483       condition = 1
484       Hydraulic_condition_publisher.publish(condition)
485     elif abs(Slip1) > slipcut and abs(Slip3) > slipcut:
486       Fp = Fp_in + deltaFp
487       timeprevHyd =timenowHyd
488       condition = 1
489       Hydraulic_condition_publisher.publish(condition)
490     elif abs(Slip2) > slipcut and abs(Slip3) > slipcut:
491       Fp = Fp_in + deltaFp
492       timeprevHyd =timenowHyd
493       condition = 1
494       Hydraulic_condition_publisher.publish(condition)
495     elif ((mu1 * N1 - abs(tau1 / rw) - m0 * g * np.cos(alpha) * np.cos(
    theta_c + beta))) < 100 and ((mu2 * N2 - abs(tau2 / rw) - m0 * g * np.
    cos(alpha) * np.cos(theta_c))) < 100:
496       Fp = Fp_in + deltaFp
497       timeprevHyd =timenowHyd
498       condition = 3
499       Hydraulic_condition_publisher.publish(condition)
500     elif ((mu1 * N1 - abs(tau1 / rw) - m0 * g * np.cos(alpha) * np.cos(
    theta_c + beta))) < 100 and ((mu3 * N3 - abs(tau3 / rw) - m0 * g * np.
    cos(alpha) * np.cos(theta_c - beta))) < 100:
501       Fp = Fp_in + deltaFp
502       timeprevHyd =timenowHyd
503       condition = 3
504       Hydraulic_condition_publisher.publish(condition)
505     elif ((mu2 * N2 - abs(tau2 / rw) - m0 * g * np.cos(alpha) * np.cos(
    theta_c))) < 100 and abs((mu3 * N3 - abs(tau3 / rw) - m0 * g * np.cos(
    alpha) * np.cos(theta_c - beta))) < 100:
506       Fp = Fp_in + deltaFp
507       timeprevHyd =timenowHyd
508       condition = 3
509       Hydraulic_condition_publisher.publish(condition)
```

```
510     elif abs((mu1 * N1 - m0 * g * np.sin(alpha)) < 500 and (mu2 * N2 - m0 *
        g * np.sin(alpha))) < 500:
511       Fp = Fp_in + deltaFp
512       timeprevHyd =timenowHyd
513       condition = 4
514       Hydraulic_condition_publisher.publish(condition)
515     elif  ((mu1 * N1 - m0 * g * np.sin(alpha))) < 500 and ((mu3 * N3 - m0 *
        g * np.sin(alpha))) < 500:
516       Fp = Fp_in + deltaFp
517       timeprevHyd =timenowHyd
518       condition = 4
519       Hydraulic_condition_publisher.publish(condition)
520     elif ((mu2 * N2 - m0 * g * np.sin(alpha))) < 500 and ((mu3 * N3 - m0 *
       g * np.sin(alpha))) < 500:
521       Fp = Fp_in + deltaFp
522       timeprevHyd =timenowHyd
523       condition = 4
524       Hydraulic_condition_publisher.publish(condition)
525     else:
526       Fp = Fp_in
527                           #print("no action needed from piston")
528
529     #timeprevHyd =timenowHyd
530   else:
531     condition = 0
532     Fp = Fp_in
533   P_ref=1/(pow(10,5)*((0.05*0.05)/4)*math.pi)*Fp
534   if (P_ref< 25):
535     P_ref=25
536     Fp=25*(pow(10,5)*((0.05*0.05)/4)*math.pi)
537     Fp_in=Fp
538   elif (P_ref>80):
539     P_ref=80
540     Fp=80*(pow(10,5)*((0.05*0.05)/4)*math.pi)
541     Fp_in=Fp
542   else:
543     P_ref=P_ref
544     Fp_in=Fp
545   PiControllerPiston(P_ref)
546
547
548 def JointState_out(JointState_data):
549   global Motor_vel, Motor_vel1, Motor_vel2, Motor_vel3,theta_c,dtheta_c
550   Motor_vel=JointState_data.velocity
551   Motor_vel1=(Motor_vel[0]/(1.4*353))*2*math.pi/60
552   Motor_vel2=-(Motor_vel[1]/(1.4*353))*2*math.pi/60
553   Motor_vel3=(Motor_vel[2]/(1.4*353))*2*math.pi/60
554
555
556
557 def PublishForceMotor1(cmd_effort_motor1):
558   global tau1, tau1cmd
559   tau1=(cmd_effort_motor1)
```

```
560   #Convert into motor torque in promile
561   tau1cmd=-(cmd_effort_motor1*650/54)
562   if abs(tau1cmd)> 650:
563     tau1cmd=650
564
565   pub_eff1 = rospy.Publisher('/maxon/canopen_motor/
        base_link1_joint_effort_controller/command', Float64, queue_size=1) #
        define a publisher
566   pub_eff1.publish(tau1cmd) # publish the new target position to the
        command topic
567   Setpoint_publisher_motor1 = rospy.Publisher('setpoint_motor1', Float64,
        queue_size=10)
568   Setpoint_publisher_motor1.publish(cmd_effort_motor1)
569
570 def PublishForceMotor2(cmd_effort_motor2):
571   global tau2, tau2cmd
572   tau2=(cmd_effort_motor2)
573   tau2cmd=(cmd_effort_motor2*650/54)
574   if abs(tau2cmd)> 650:
575     tau2cmd=-650
576
577   #tau1=50
578   #tau1=-(cmd_effort_motor1)*2
579
580   pub_eff2 = rospy.Publisher('/maxon/canopen_motor/
        base_link2_joint_effort_controller/command', Float64, queue_size=1) #
        define a publisher
581   pub_eff2.publish(tau2cmd) # publish the new target position to the
        command topic
582   Setpoint_publisher_motor2 = rospy.Publisher('setpoint_motor2', Float64,
        queue_size=10)
583   Setpoint_publisher_motor2.publish(cmd_effort_motor2)
584
585 def PublishForceMotor3(cmd_effort_motor3):
586   global tau3, tau3cmd
587   tau3=(cmd_effort_motor3)
588   tau3cmd=-(cmd_effort_motor3*650/54)
589   if abs(tau3cmd)> 650:
590     tau3cmd=650
591
592   pub_eff3 = rospy.Publisher('/maxon/canopen_motor/
        base_link3_joint_effort_controller/command', Float64, queue_size=1) #
        define a publisher
593   pub_eff3.publish(tau3cmd) # publish the new target position to the
        command topic
594
595   Setpoint_publisher_motor3 = rospy.Publisher('setpoint_motor3', Float64,
        queue_size=10)
596   Setpoint_publisher_motor3.publish(cmd_effort_motor3)
597
598 def PublishForce(cmd_pos):
599   global tau1,tau2,tau3
600   pub_eff1 = rospy.Publisher('/maxon/canopen_motor/
```

```
    base_link1_joint_effort_controller/command', Float64, queue_size=1) #
    define a publisher
601   pub_eff2 = rospy.Publisher('/maxon/canopen_motor/
    base_link2_joint_effort_controller/command', Float64, queue_size=1) #
    define a publisher
602   pub_eff3 = rospy.Publisher('/maxon/canopen_motor/
    base_link3_joint_effort_controller/command', Float64, queue_size=1) #
    define a publisher
603
604
605   pub_eff1.publish(cmd_effort_motor1) # publish the new target position to
    the command topic
606   pub_eff2.publish(cmd_effort_motor2) # publish the new target position to
    the command topic
607   pub_eff3.publish(cmd_effort_motor3) # publish the new target position to
    the command topic
608
609
610 def epos4_cmd():
611   global integral1,integral2,integral3, integral1old,integral2old,
    integral3old,time_prev,Kp1,Ki1,Kp2,Ki2,Kp3,Ki3,cmd_effort_motor1,
    cmd_effort_motor2,cmd_effort_motor3,thrustlqr,K_vel,K_pos, Motor_vel1,
    Motor_vel2,Motor_vel3,thrustlqr,feedforwardunit
612   rospy.init_node('maxon_epos4_ros_canopen_python_example', anonymous=True)
     # initialize the ROS node
613   pub_eff1 = rospy.Publisher('/maxon/canopen_motor/
    base_link1_joint_effort_controller/command', Float64, queue_size=1) #
    define a publisher
614   pub_eff2 = rospy.Publisher('/maxon/canopen_motor/
    base_link2_joint_effort_controller/command', Float64, queue_size=1) #
    define a publisher
615   pub_eff3 = rospy.Publisher('/maxon/canopen_motor/
    base_link3_joint_effort_controller/command', Float64, queue_size=1) #
    define a publisher
616   rospy.Subscriber("/maxon/joint_states",JointState, JointState_out)#
    Subscribes to the motots
617   rospy.Subscriber("/imu", Imu,imu_out)#Subscribes to the Imu
618       rospy.Subscriber("/pressure_clamp", Float32,pressure_clamp_out)
619       Piston_ref_pressure = rospy.Publisher('/Piston_ref_pressure', Float32
    , queue_size=10)
620
621       Pressure_piton_avg_publisher = rospy.Publisher('Pressure_piston_avg',
     Float64, queue_size=10)
622       Setpoint_publisher = rospy.Publisher('setpoint', Float64, queue_size
    =10)
623
624   pressure_Piston_reference_bars = rospy.Publisher('
    Piston_pressure_reference_bars', Float64,queue_size=10)
625
626   Setpoint_publisher_motor1 = rospy.Publisher('setpoint_motor1', Float64,
    queue_size=10)
627   Setpoint_publisher_motor2 = rospy.Publisher('setpoint_motor2', Float64,
    queue_size=10)
```

```
628 Setpoint_publisher_motor3 = rospy.Publisher('setpoint_motor3', Float64,
      queue_size=10)
629
630 Setpoint_publisher_motor1_LQR = rospy.Publisher('setpoint_motor_LQR1',
      Float64, queue_size=10)
631 Setpoint_publisher_motor2_LQR = rospy.Publisher('setpoint_motor_LQR2',
      Float64, queue_size=10)
632 Setpoint_publisher_motor3_LQR = rospy.Publisher('setpoint_motor_LQR3',
      Float64, queue_size=10)
633
634 Setpoint_publisher_motor1_feedforward = rospy.Publisher('
      setpoint_motor1_feedforward', Float64, queue_size=10)
635 Setpoint_publisher_motor2_feedforward = rospy.Publisher('
      setpoint_motor2_feedforward', Float64, queue_size=10)
636 Setpoint_publisher_motor3_feedforward = rospy.Publisher('
      setpoint_motor3_feedforward', Float64, queue_size=10)
637
638
639 Hydraulic_condition_publisher = rospy.Publisher('hydraulic_condition',
      Float64, queue_size=10)
640
641     cmd_motor_1 = rospy.Publisher('cmd_motor_1', Float64, queue_size=10)
642     cmd_motor_2 = rospy.Publisher('cmd_motor_2', Float64, queue_size=10)
643     cmd_motor_3 = rospy.Publisher('cmd_motor_3', Float64, queue_size=10)

644 time = 0
645
646 rate = rospy.Rate(300)
647 Intizial_angle()
648
649 time_prev = datetime.datetime.now()
650 cmd_effort_motor1 = 0
651 cmd_effort_motor2 = 0
652 cmd_effort_motor3 = 0
653
654
655 timeprev = datetime.datetime.now()
656
657 #References
658 theta_cref=2*math.pi
659       dtheta_cref=-0.016
660
661 while not rospy.is_shutdown():
662   angle_cal()
663   Normal_force_cal()
664   actual_states = np.array([theta_c, dtheta_c, Motor_vel1, Motor_vel2,
      Motor_vel3]) #Makes the matrix with the current states Crawler pos
      crawler vel motor1 vel motor 2 vel motor 3 vel
665
666   rate.sleep() # sleep to enforce a 50 Hz loop
667
668   #Position controller of Lqr
669   if theta_c>0.95*(theta_cinit+theta_cref):
```

```
670       K=K_pos
671       Nbar=Nbar_pos
672       desired_states = np.array([theta_cref-theta_cinit, 0, 0, 0, 0])
673       precompensatedref = np.dot(Nbar,desired_states)
674       feedback = np.dot(K,actual_states)
675       feedforwardunit=0.33*(math.cos(theta_c)*m0*g*math.cos(alpha)+((N1+N2+
    N3)*mur)*1.9)*rw
676       thrustlqr = precompensatedref - feedback# #Calculates the error
    between current and desired states
677                           thrustlqr[0]=thrustlqr[0]+feedforwardunit
678                           thrustlqr[1]=thrustlqr[1]+feedforwardunit
679                           thrustlqr[2]=thrustlqr[2]+feedforwardunit
680       Setpoint_publisher_motor1_LQR.publish(thrustlqr[0])
681       Setpoint_publisher_motor1_feedforward.publish(feedforwardunit)
682       Setpoint_publisher_motor2_LQR.publish(thrustlqr[1])
683       Setpoint_publisher_motor2_feedforward.publish(feedforwardunit)
684       Setpoint_publisher_motor3_LQR.publish(thrustlqr[2])
685       Setpoint_publisher_motor3_feedforward.publish(feedforwardunit)
686
687     #Velocity controller LQR
688     else:
689       K=K_vel
690       Nbar=Nbar_vel
691       desired_states = np.array([0, dtheta_cref, 0, 0, 0])
692       precompensatedref = np.dot(Nbar,desired_states)
693       feedback = np.dot(K,actual_states)
694
695 #Gael gains 3.4 for static og 1.9 for dynamics
696       #Normal forces at 63 bars -1.1 rad starting angle
697       N11=3827
698       N22=2432
699       N33=3600
700       feedforwardunit=-0.33*(math.cos(theta_c)*m0*g*math.cos(alpha)+((N1+N2
    +N3)*mur)*3)*rw
701
702       thrustlqr = (precompensatedref - feedback)
703       gain=5
704
705                           thrustlqr[0]=(thrustlqr[0]*gain+feedforwardunit)
706                           thrustlqr[1]=(thrustlqr[1]*gain+feedforwardunit)
707                           thrustlqr[2]=(thrustlqr[2]*gain+feedforwardunit)
708
709       Setpoint_publisher_motor1_LQR.publish(thrustlqr[0])
710       Setpoint_publisher_motor1_feedforward.publish(feedforwardunit)
711       Setpoint_publisher_motor2_LQR.publish(thrustlqr[1])
712       Setpoint_publisher_motor2_feedforward.publish(feedforwardunit)
713       Setpoint_publisher_motor3_LQR.publish(thrustlqr[2])
714       Setpoint_publisher_motor3_feedforward.publish(feedforwardunit)
715
716
717       # #Calculates the error between current and desired states
718       if (abs(Motor_vel1)>0.005) and (abs(Motor_vel2)>0.005) and (abs(
    Motor_vel3)>0.005) and (abs(dtheta_c)>0.005):
```

```
719         slip_cal()
720
721       else:
722         PublishForceMotor1(thrustlqr[0])
723         PublishForceMotor2(thrustlqr[1])
724         PublishForceMotor3(thrustlqr[2])
725
726     hydraulic_controller()
727
728
729
730
731 if __name__ == '__main__':
732     try:
733         epos4_cmd() # call the loop function
734     except rospy.ROSInterruptException:
735         pass
```

Listing D.2: Python example