

QUALITY CONTROL

This report has been reviewed and approved in accordance with the policies of IHC MTI B.V.

	Name	Date	Signature
WRITTEN BY	Jelle Spijker	June 10, 2020	
REVIEWD BY	ir. Frits Hofstra	June 10, 2020	
REVIEWD BY	J. B. van Elburg MSc. BEng.	June 10, 2020	
APPROVED BY	A. J. P. M. Koevoets MBA BSc.	June 10, 2020	

DISCLAIMER

This document has been prepared by IHC MTI B.V. for Royal IHC. The opinions and information in this report are entirely those of IHC MTI B.V., based on data and assumptions as reported throughout the text and upon information and data obtained from sources which IHC MTI B.V. believes to be reliable. While IHC MTI B.V. has taken all reasonable care to ensure that the facts and opinions expressed in this document are accurate, it makes no guarantee to any person, organization or company, representation or warranty, express or implied as to fairness, accuracy and liability for any loss howsoever, arising directly or indirectly from its use or contents.

This document is intended for use by professionals or institutions. This document remains the property of IHC MTI B.V.. All rights reserved. This document or any part thereof may not be made public or disclosed, copied or otherwise reproduced or used in any form or by any means, without prior permission in writing from IHC MTI B.V..

This document is dated June 10, 2020

CHAPTER CONTENTS

Contents

1 INTRODUCTION	1
1.1 USE CASES	1
1.1.1 ARBITRARY SHAPED SPACE	1
1.1.2 MARINA AQUA DELTA	1
1.1.3 THREE GORGES DAM	1
1.2 ARCHIMEDES DRIVEN CRAWLER	1
2 DREDGING PRINCIPLES AND APPLICATIONS	2
2.1 BASIC DREDGING APPLICATION	2
2.2 COMMONLY USED VESSELS AND EQUIPMENT	2
2.2.1 MECHANICAL DREDGERS	2
2.2.2 HYDRAULIC DREDGERS	2
2.3 HYDRAULIC DREDGING PRINCIPALS	3
2.3.1 DREDGE PUMP	3
2.3.2 AUGER DREDGE HEAD	4
3 RESEARCH	5
3.1 UNDERWATER COMMUNICATION	5
3.1.1 WIRED COMMUNICATION	5
3.1.2 WIRELESS COMMUNICATION	7
3.2 SENSORS	11
3.2.1 STATE SENSING	11
3.2.2 EXTERNAL SENSOR	17
3.3 LOCATION UNDER UNCERTAINTY	17
3.3.1 LOCALIZATION REFINEMENT USING KALMAN FILTERS	18
3.3.2 BASIC KALMAN FILTERING	18
3.4 COVERAGE PATH PLANNING	22
3.4.1 MORSE-BASED CELLULAR DECOMPOSITION	23
3.4.2 LANDMARK-BASED TOPOLOGICAL COVERAGE	28
3.4.3 GRID-BASED METHODS	32
4 CONTROLLER DESIGN	39
4.1 STRATEGY DECISION	39
4.2 PERIPHERALS	40
4.3 KALMAN FILTER DESIGN	40
4.3.1 STATE REPRESENTATION	41
4.3.2 MOTION MODEL	42
4.3.3 SOIL DYNAMIC MODEL	47
4.3.4 DREDGE MODEL	52
4.3.5 STEERING MODEL	55
4.4 CONTROLLER FRAMEWORK	56
4.4.1 DESIGN PRINCIPLES	56
4.4.2 NAMING SCHEME	57
4.4.3 THE CAPTAIN	57
4.4.4 A FIRST MATE	59
4.4.5 A NAVIGATOR	59
4.4.6 A BOATSWAIN	60
4.4.7 A WORLD	61
4.4.8 A VESSEL	61



5 DESIGN VALIDATION	62
5.1 SIMULATION	62
5.1.1 SENSOR SIMULATION	63
5.1.2 SIMULATION MODEL	64
5.2 RESULTS	65
NOMENCLATURE	68
6 NOMENCLATURE	68
GLOSSARY	77
7 GLOSSARY	77
ACRONYMS	81
8 ACRONYMS	81
9 BIBLIOGRAPHY	84
APPENDICES	90
A CRAWLER PARTLIST	91
B APPLIED RESEARCH METHODS	93
C DATASHEET PUMP TT15-55	96



CHAPTER 1 INTRODUCTION

This chapter will first specify three use-cases, specified in the project assignment, in which an crawler must operate. It then describes basic principles, applications and tools relevant for these use cases.

1.1 USE CASES

The use case below are determined by ir. F. Hofstra, these cases are expected to be valid and realistic. Keeping in mind their marketability. These cases will determine the needed functionality for an crawler and stand at the basis for the controller design.

1.1.1 ARBITRARY SHAPED SPACE

An crawler is placed in a predefined arbitrary shaped space, not too complex, with an area of 3500m². The shape of this space is set, but the movement pattern is unrestricted. The crawler has to remove a layer with a depth of 5cm. The controller has to determine an optimal path with the least amount of time or the shortest path. This can be coupled with learning capabilities and an analyze capacity. At a later time additional constrains can be added which keep in mind the deployment location of a flexible dredgeline and an umbilical.

1.1.2 MARINA AQUA DELTA

The crawler operates in a predefined space with obstacles, not every obstacles is known. The actual location is marina Aqua Delta located in Bruinisse, the Netherlands. The shape of this location is set but the movement pattern is unrestricted. An crawler has to remove a layer with a depth of 5cm. The controller has to determine an optimal path with the least amount of time or the shortest path. This can be coupled with learning capabilities and an analyze capacity. The marina has enough depth for the crawler to move underneath the scaffolding. No consideration has to be made for a flexible dredgeline and a umbilical. These conditions are introduced at a later stage.

1.1.3 THREE GORGES DAM

An crawler operates in a predefined space with obstacles, not every location of those obstacles is known. The predefined space is located at the foot of three Gorges dam. Silt is deposited at the foot of this dam, due to natural occurring erosion and sedimentation. The accumulation of silt can be controlled by dredging localized pits. Which in turn create locations with a lower density. This induces a gravity driven density current towards those locations. The crawler has to maintain an average nominal depth with a certain silt deposit rate.

1.2 ARCHIMEDES DRIVEN CRAWLER

2 CHAPTER **DREDGING PRINCIPLES AND APPLICATIONS**

This chapter describes the dredging task in some detail. Readers familiar with dredging and commonly used terminology can skip this chapter, since no new information will be provided. It first describes basic principles, applications and tools applicable by the used machinery for the use-cases.

2.1 BASIC DREDGING APPLICATION

Training Institute for Dredging [43] defines dredging as the underwater removal of soil and its transport from one place to another for the purpose of deepening or making profitable use of the removed soil. They make a distinction between nine types of operations: dredging for prosperity, dredging in ports and channels, exploitation of agricultural resources, mineral dredging, coastal protection, land reclamation, infrastructural projects, improvement of the environment and trenches for cables and pipelines.

All three described use-cases are of the maintenance type. Schriek [73] states that, in order to maintain existing waterways and harbours, the depth of the bed must be preserved by regularly removing silt. In canals and ports basins, where currents are low, the sediment is mostly fine-grained silt and sludge. Where currents are stronger, as in access channels in tidal zones, or rivers, the sediment is sand. He further describes that a characteristic of this kind of work is the weak cohesion of the soil to be removed, since it consists of recently deposited sediment and no significant consolidation has taken place yet.

Sanitation dredging is a distinct form of maintenance dredging and is a process that has been specifically designed for contaminated sediment. Just in the way sediment settles in rivers, harbours and deltas so does heavy metal, inorganic and aromatic compounds, especially downstream of industrial areas. When these contaminated sediments become a risk towards public health and environment, they need to be removed with care and precision.

2.2 COMMONLY USED VESSELS AND EQUIPMENT

Common dredge tools used during maintenance work are listed below. Out of this list, backhoes and suction dredgers are mostly used during port maintenance. Vlasblom [84] states that dredgers can be divided into two categories: mechanical dredgers and hydraulic dredgers. The difference lies in the way the soil is excavated; either mechanically or hydraulically.

2.2.1 MECHANICAL DREDGERS

They work by removing soil and sediment from the submerged soil bed by mechanically excavating it and transporting it to a storage location, such as a hopper. The various types of mechanical dredgers won't be described in this section, since the crawler used in our use-cases will be of a hydraulic type.

2.2.2 HYDRAULIC DREDGERS

These types of dredgers work by removing and transporting soil from the seabed. They use a hydraulic system, where the necessary work needed for mass transportation is delivered by a pump. The soil is transported as a slurry which is a mixture that consists of both solid and fluid phases, and this is usually stored in a dedicated place such as a hopper.

PLAIN SUCTION DREDGERS

Vlasblom [84] describes a plain suction dredger as a stationary dredger, consisting of a pontoon anchored by one or more wires and with at least one sand pump that is connected to a suction pipe.

The discharge of the dredged material can take place via a pipeline or via a barge-loading installation. During sand dredging, the dredger is moved slowly forwards by a set of winches.

TRAILING SUCTION HOPPER DREDGERS

The TSHD is a seagoing ship equipped with one or two suction tubes, a pump installation and a hopper with multiple bottom doors and one or more overflows. A draghead is attached to each suction tube and is trailed across the sea bed to loosen the soil before it's pumped up [73]. This soil is stored in a hopper which is periodically discharged, at a designated location, through dumping or pumping out.

AUGER SUCTION DREDGERS

According to VBKO Vereniging van waterbouwers in bagger-, kust- en oeverswerken [12] an ASD consists of a double symmetrical Archimedes screw, also called an auger, surrounded with a steel protective cover and a flexible rubber curtain. This auger is lowered, on a rigid arm, and positioned on the soil bed. Here, it cuts the material and actively transports it into the centre where it's sucked away by a dredge pump. Because the complete dredging process takes place behind a flexible rubber curtain and the auger guides all material towards the suction mouth, this type of dredger is well suited for sanitation maintenance.

CUTTER SUCTION DREDGERS

According to Vlasblom [84] a CSD is a stationary dredger equipped with a cutter device (cutter head) which excavates the soil before it's sucked up by the flow dredge-pump. During this operation, the dredger moves around a spud pole by pulling and slackening on the two fore sideline wires. This type of dredger is accurate and can cut almost all types of sediment.

2.3 HYDRAULIC DREDGING PRINCIPALS

According to Van Den Berg [68] hydraulic systems are the de-facto industry of transportation for dredged sedimented or slurry; hydraulic systems consist of pipes, either flexible or rigid, combined with centrifugal pumps, a suction mouth and a discharge unit. The pump adds energy to a slurry, such that a required flowrate can be achieved, this energy is needed to overcome a system specific pressure drop. Which is the result of energy losses due to potential height differences, kinematic behaviour of the fluid and friction, both from shearing of a fluid along a wall and internal shearing of the fluid itself.

The section below briefly describes the workings of two main components in this hydraulic system, namely a dredge-pump and a draghead.

NOTE 2.1: OUT-OFF SCOPE

Two of the use-cases mention that additional constraints such as a flexible dredge line to shore, can be added to the assignment. It was however opted, to not apply these additional constraints, due to a time constraint on the assignment as a whole.

2.3.1 DREDGE PUMP

In order to transport slurry with a particular density and velocity through a pipeline, a pressure, equal to the sum of all the resistances and geodetic head must be generated. A pump supplies this pressure [68]. Assuming a steady flow, the pump basically increases the Bernoulli head of the flow between point 1, the eye, and point 2, the exit [57].

2.3.2 AUGER DREDGE HEAD

An auger dredge head excavate soil by employing a Archimedes screw transportation principle. This method ensures an extremely quiet cutting and mixing process with little spillage and turbidity in the surroundings. The large working width of the auger makes it extremely suited to dredge thin, possibly polluted, layers at a relatively high production rate [73].

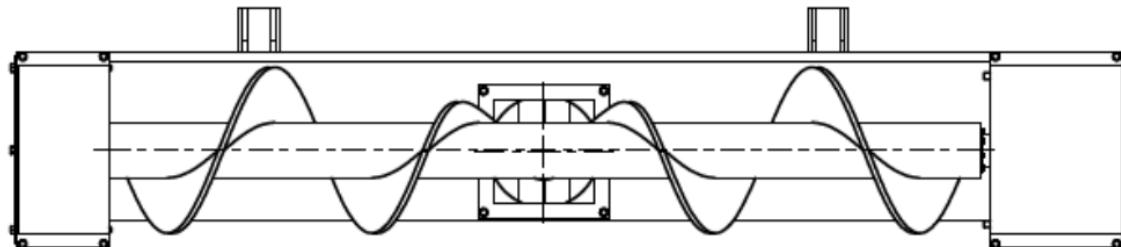


FIGURE 2.1: SCHEMATIC DRAWING OF AN AUGER DREDGE HEAD [92]

The auger is in effect a screw conveyor which guides the material towards the suction head. Green and Perry [36] states that the screw conveyor is one of the oldest and most versatile conveyor types there is. It consists of a helicoid flight mounted on a pipe which turns in a trough. Screw conveyors are well standardized, using International Standard ISO [4] empirical gathered factor values for filling rates and progress resistance.

NOTE 2.2: ASSUMPTION

The assumption is made that the hydraulic system, consisting of flexible pipes and a pump, is the limiting factor in the mass flow, and that the auger simply delivers what is needed.

CHAPTER RESEARCH 3

A crawler performs its tasks in an underwater environment. Its task consists of moving, mapping and dredging a certain basin or area. In order to fulfill tasks its own accord, it has to be able to sense its surrounding environment and execute its task using a strategy. Which ensures performance according to specification.

In the next sections the key philosophies and processes are investigated; All of these are needed to fulfill its objective. Firstly, in Section 3.1, different ways of underwater communication are reviewed. This is after all the interface between man and machine. A second review regarding useful sensors made in Section 3.2, their workings and possible applications are described.

Once the low-level tools, such as communication devices and sensors are discussed. A careful study is made into possible implementation and fusion of these sensors. Such that they can be used to estimate a location of a crawler. Which needs to operate in a GPS deprived environment.

Section 3.3 describes the use of cooperative localization techniques and Kalman filter.

Lastly a survey is made for useful strategy at a higher abstraction level. Section 3.4, describes how a crawler could best perform its main task: covering and dredging a large basin, uniformly. These so called CPP algorithms, describe and propose different strategies that allow a crawler to perform its task in an unknown and changing environment.

3.1 UNDERWATER COMMUNICATION

This section describes various principles of underwater communication. It identifies two basic methods of transmitting data, namely: wired communication or wireless communication. Wired communication will be in a form of an umbilical. Using regular and industry standard communication protocols. While wireless communication can be performed through four basic principles. These are: electromagnetic, electric current, acoustic or optical signals. Of these principles only electromagnetic and acoustic are explored, since an electrical current doesn't work in a fresh water reservoir and optical signals get sub-optimal performance in a dredging environment. Due the diffraction and scattering of light by floating sand particles.

The environment presented in the uses cases, described in Section 1.1, state that the crawler will operate in fresh water basins. It's also likely that it will be connected to the water surface with a floating dredgeline. The choice for wired communication is therefore easily made. There may however still be a need for wireless communication with external sensors. Such as the principles presented in Section 3.3 illustrate. Where an option to minimize a localization error using multiple bots, is presented.

3.1.1 WIRED COMMUNICATION

With wired communication, data signals are transmitted over a wire. Which acts as a pathway where the information is transmitted as a digital bitstream. Transmission of information through this wire is limited by a certain bandwidth in Hz. Where the limiting factors are material properties such as: conductivity, permittivity and permeability. As well as processing of the signals at the end and start node. Communication wires are made of a carrier medium, such as copper or glass fibre. This carrier medium facilitate transmission of electromagnetic waves or currents. Where electromagnetic waves, such as light, are transmitted through fiber optic cables, where a modulated pulse of light propagates through a glass tube through the principle of TIR. Where electromagnetic communication makes use of copper wires, where an electric charge propagates through the cable. Copper is the industry de-facto due to its excellent electrical conductivity properties.

Babani, Bature, Faruk, et al. [80] made a comparative study between fibre-optic and copper cables in a context of modern network protocol. They identified the following properties for comparison: bandwidth, cost, dimensional properties (such as weight, size and flexibility), signal loss and safety and immunity. They illustrate that fiber optics cables, although more expensive, are the better choice. By stating that fibre-optic cables are smaller and lighter compare to metal cables, especially copper based.

Optical fiber occupies less space in conduits than copper cabling and weighs less too. Furthermore, they allow for tighter bend radius than any copper cables. And signals don't cross-talk with different wires. The low signal attenuation performance and superior signal integrity found in fiber optical systems facilitates much longer runs for signal transmission. The attenuation loss experienced in fiber optic cables can be attributed to microscopic and macroscopic impurities in the fiber material and structure, which cause absorption and scattering of light signal. In figure 3.1 the attenuation loss of 1km of cable is shown as a function of frequency. Both signals propagate with nearly the same speed through their corresponding wire, but when a high data throughput is wanted. It becomes evident from this figure that usage of fiber-optics are paramount.

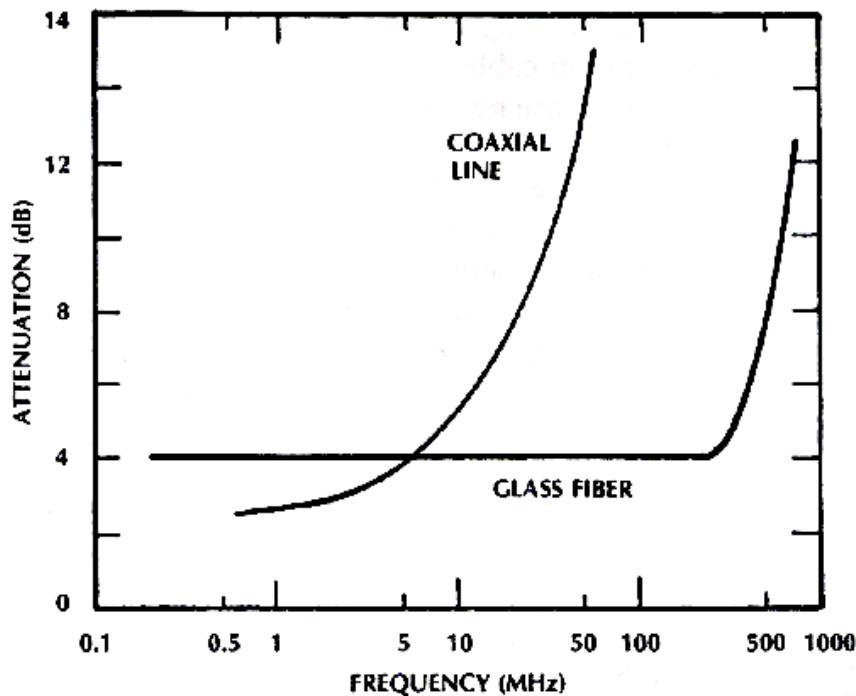


FIGURE 3.1: EFFECTIVE ATTENUATION FIBER VS COPPER CABLE 1 km [10]

Other important factors to consider, for an underwater wired-communication between a base station and a dredge bot, are the effects of the wire on the bot itself. Whitcomb [17] states that most present day vehicles are ROV — tele-operated vehicles employing an umbilical cable to carry both power and telemetry from a mother-ship to the vehicle. He further states that a growing number of research vehicles are AUV — which operate without an umbilical tether. This statement is supported by Valavanis, Gracanin, Matijasevic, et al. [8], whom describes that the ROV umbilical cable constrains the vehicle to operations in close proximity to the support ship. Because the crawler is tethered to a location above water level, due to it's floating dredgeline, and because this crawler is from its starting-point constructed as a ROV, it will, in all likelihood, be controlled through an umbilical.

Westneat, Blidberg, and Corell [5] describes that, as the range of operations becomes longer and water deeper, the drag exerted by the tether becomes significant. The thrusters, and thus the vehicle itself, must become larger and the cable thicker, and the energy that goes into the cable maintenance becomes a major factor. This factor is illustrated by Fang, Hou, and Luo [35], whom describes a mathematical model which allow the state representation of the dredge bot, as described in section 4.3.1, to be modified by the forces that are exerted on the cable. In these equations, mass and inertia of the cable play an important role. Because these are just a fraction of the properties for a dredgeline, it's assumed that these forces can be neglected. According to Feng and Allen [25] the effects of the cable can be reduced when it's deployed by a drum on the shore with negligible tension when it's pulled by the vehicle.

PROTOCOLS

The signals which are transported through the wires need to adhere to certain rules and conventions. In other words, the transponder and receiver need to speak the same language and be aware of etiquette, such that a message is received as intended. The IEEE, have dictated most of the widespread used norms today. The most common used norm in wired communication is *IEEE 802.3* or as it's more commonly known Ethernet. Which consists of a multitude of protocols. In this IEEE norms are the physical layer, data link layers and the MAC for each protocol defined.

Shortly put, MAC is defined as the lower sub layer of the data link layer and provides addressing and channel access control mechanisms that allow for communication between several terminals, or nodes, within a multiple access network. This layer act as an interface between the LLC sub layer and the network's physical layer. Where the LLC makes it possible to let several network protocols coexist. According to Jolectra [81] the current dredge bot makes use of an *Allen Bradley ETHERNET/IP adapter* of type 1769-AENTR, which is allows the use CIP, TCP and UDP. Where CIP is used by EtherNet/IP, and is a familiar and widely used protocol for controllers.

3.1.2 WIRELESS COMMUNICATION

Freitas [70] tells us that wireless communications have been subject to enormous research and improvements in the near past. This effort is responsible for allowing multiple devices to securely communicate simultaneously with high availability, great distances and high data rates. While these improvements are applied and tested mainly in over-the-air communications, underwater communications suffer from a low applicability of radio frequency transmission systems due to a low attenuation of EMW in water.

He [70] further states that When using radio frequency, underwater communications does not fully benefit from the improvements achieved in air since electromagnetic propagation in water causes a big reduction in the effective range. Because of the limitations that water imposes, these communications are currently performed using acoustic waves and in some cases optical systems. This is further supported by Lloret, Sendra, Ardid, et al. [60] who remarks that underwater communication research is primarily focused on the use of optical signals, electromagnetic signals and the propagation of acoustic and ultrasonic signals. Each technique has its own characteristics, with its benefits and drawbacks, mainly due to the chemical characteristics [50] and physical constraints of the medium [41].

ELECTROMAGNETIC COMMUNICATION

A common method to transfer data via a wireless connection is to make use of EMW, these are a type of electromagnetic radiation with wavelengths in the electromagnetic spectrum. As is shown in figure 3.2. Waves in this spectrum can have frequencies between 3kHz or 3GHz. These waves travel the speed of light and are transverse waves, because the amplitude is perpendicular to the direction of the wave travel. However, EMW are always waves of fields, not of matter, because they are fields, EMW can propagate in empty space [78].

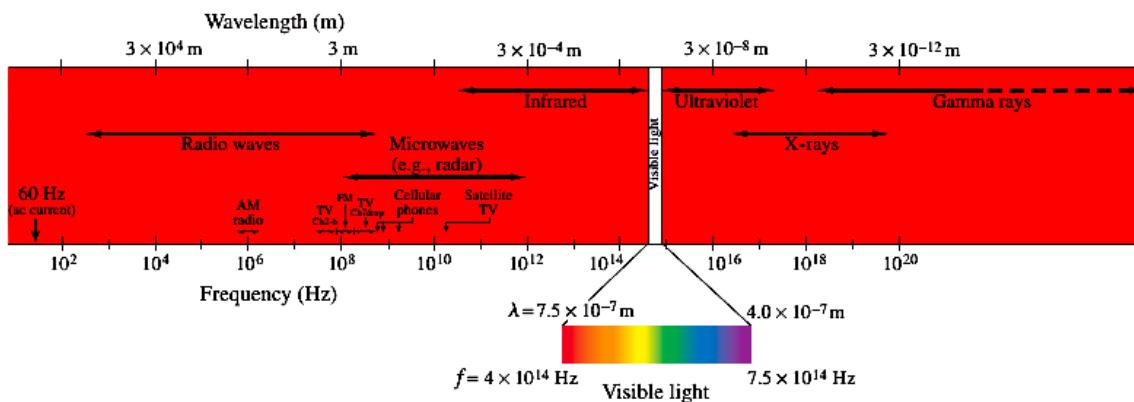
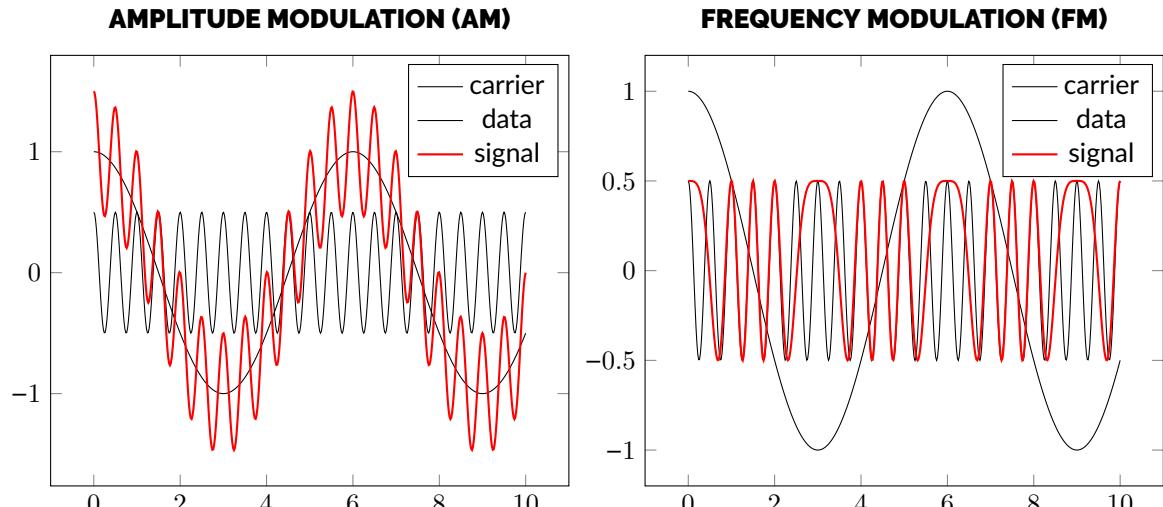


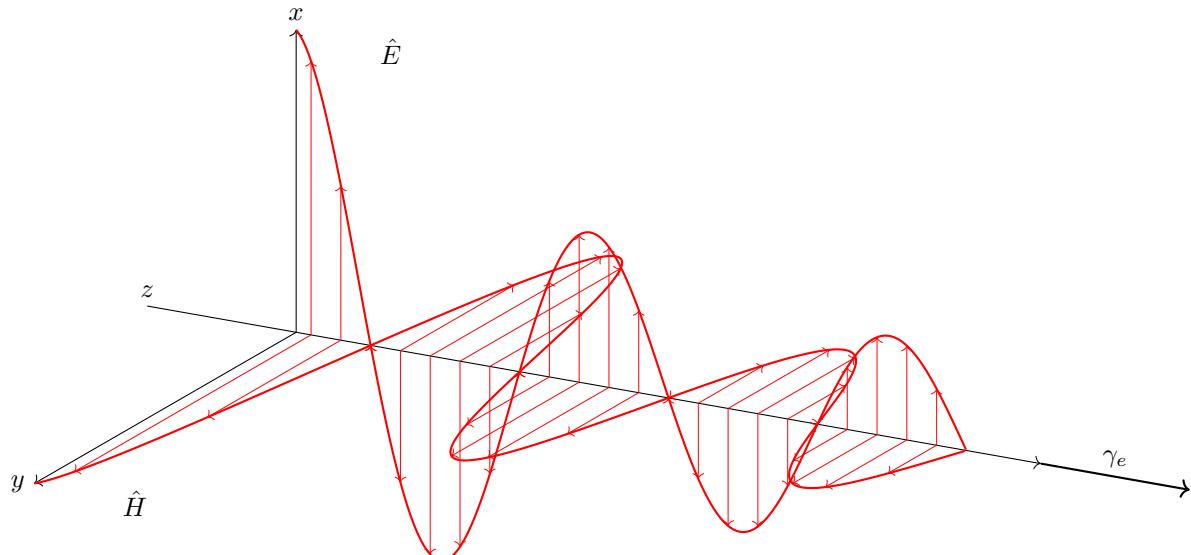
FIGURE 3.2: ELECTROMAGNETIC SPECTRUM [78]

Data is transferred between devices by either modulating the frequency or the amplitude of a signal data can be transferred. Where a carrier frequency is modulated by superimposing a data signal. Which is illustrated in figure 3.3.

**FIGURE 3.3: SIGNAL MODULATION**

Hagman, Elias [45] tell us that the reasons, why EMW are used to transfer information in the classic wireless air channel, lies in their fast propagation speed. In their wide usable frequency spectrum and coupled with a small environment noise, compared for example with acoustics factors. This all leads high possible data rates. Furthermore, the EMW has the ability to propagate without a carrier medium and the electric-magnetic field conversion enables in general very large communication ranges.

But in water – especially in seawater – things get different. This statement is supported by RamaKrishna and Nissen [61] whom tells that the ocean is almost impervious to EMW, which makes them useless for wireless underwater communication over distances greater than a hundred meters. Hagman, Elias [45] illustrate this by solving Maxwell's equation to predict the propagation of EMW for the case of a linearly polarized plane travelling in z -direction, we get the electric field strength E_x and the magnetic field strength H_y [45].

**FIGURE 3.4: DAMPENING OF ELECTRIC AND MAGNETIC FIELD**

Where \hat{E} and \hat{H} are the amplitudes of the electric and the magnetic field wave and γ_e expressed in ϵ_e , as shown in equation 3.3, where μ_e and σ_e of a material. Here α_e and β .

$$E_x = \hat{E} \exp^{i\omega t - \gamma_e z} \quad (3.1)$$

$$H_y = \hat{H} \exp^{i\omega t - \gamma_e z} \quad (3.2)$$

$$\gamma_e = i\omega \sqrt{\epsilon_e \mu_e - \frac{i\sigma_e \mu_e}{\omega}} = \alpha + i\beta \quad (3.3)$$

$$\alpha_e \approx 0.0173 \sqrt{f \sigma_e} \quad (3.4)$$

As is evident from equation 3.1 and 3.2, there is a logarithmic relationship, maximization of the propagation γ_e leads to a lower amplitude of the electric and magnetic fields. This propagation is mostly determined by the attenuation α_e , which varies at different frequencies and mediums. Claus [69] tells us that this attenuation factor is given as equation 3.4, which shows us that the attenuation is related to the square root of the frequency f in hertz Hz, multiplied by the conductivity of the water σ_e in S/m. Whilst Hattab, El-Tarhuni, Al-Ali, et al. [66] states that the loss of a signal travelling through water can be calculated using equation 3.5. They state that the knowing the real-part of γ_e is sufficient to calculate the loss for a given frequency. Since the only changing term due to frequency in the complex-valued γ_e is in its imaginary part, and due to the fact that each γ_e is multiplied with i , both outside of the root as inside, this value will be a constant throughout the frequency spectrum. And this attenuation model will not be used for our calculations.

Where $\Delta d_{1,2}$ is the separation distance between transmitting and receiving nodes and only the real part of the propagation constant σ_e is used.

$$L_{\alpha,e} = \text{Re}(\gamma_e) = \frac{20}{\ln(10)} \Delta d_{1,2} \Rightarrow \Delta d_{1,2} \frac{L_{\alpha,e}}{\text{Re}(\gamma_e) \frac{20}{\ln(10)}} = \frac{L_{\alpha,e}}{\alpha_e} \quad (3.5)$$

The maximum penetration depth of signal in (sea) water, will, for simplicity's sake be calculated with equation 3.5, where α_e is obtained using equation 3.4. Jiang and Georgakopoulos [51] tells us that seawater has a typically high conductivity of 4.0 S/m, whilst freshwater has a typically conductivity of only 0.0 S/m, 400.0 times less. He [51] further states that communication using electromagnetic waves in fresh water can be more efficient in fresh water. This statement is confirmed by Jiang and Georgakopoulos [51], Ainslie [48] and Bogie [3]. Figure 3.6 and 3.5, which shows the EMW propagation in fresh and seawater for commonly used frequencies, illustrate this phenomenon.

PROTOCOLS

Subsection 3.1.1 describes the need for protocols as a transceiver and receiver speaking the same language and adhering to the same etiquette. This holds true for wireless protocols as well. Most wireless protocols are described in the IEEE 802 standards. These are a family of standard network protocols. Describing networks using variable-size packets. These protocols are the de-facto industry standards. A short description for the most popular 802 standards are given below. These protocols map to two layers, namely: Data link and physical layers. Where the data link layer is split into two sub-layers LLC and MAC. Where the LLC provides the multiplexing mechanisms that enable the network protocols and provide flow control and automatic repeat requests. Whilst MAC provides addressing and channel access control mechanisms that make it possible for several nodes to communicate within a multiple access network.

IEEE 802.11 WLAN

The IEEE 802.11 standard is also known as WiFi. It encompasses wireless modulation techniques,

designates as 802.11(a, b, g, n and ac). The 802.11 standard makes use of the 2.4 GHz and 5.0 GHz bandwidth. Freitas [70] states that Wi-Fi frequencies maybe a challenge when used in underwater communications, because its attenuation drastically reduce the channel distance. As is shown in figure 3.5. A new standard 802.11af is being developed. This standard will make use of the 700.0 MHz [MHz] frequency. Which might give an extra couple of meters underwater.

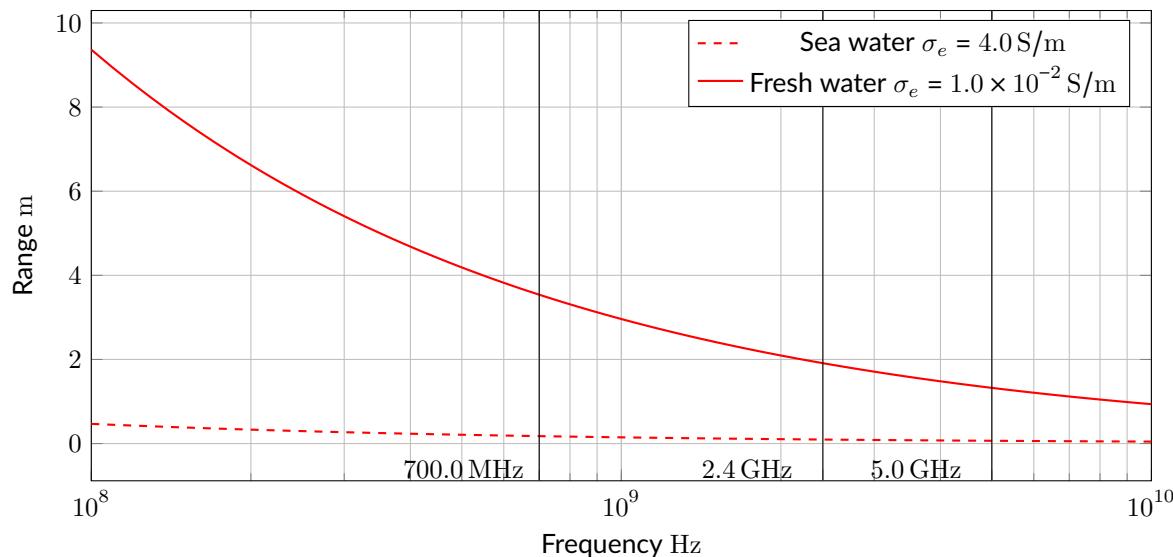


FIGURE 3.5: PROPAGATION RANGE OF WI-FI IN WATER.

IEEE 802.15.4 LO-FI

From all different protocols described in the IEEE 802.15 special consideration is made into the IEEE 802.15.14 or LoRa. Which is an upcoming communication protocol for IoT devices. It operates in 433.0 MHz and (863.0 to 870.0) MHz. The protocols are opensource and the modules are very cheap. This protocol is developed for robust long range communication, which can reach 22.0 km on land. Akyildiz, Pompili, and Melodia [28] tells us that the electromagnetic waves at 433.0 MHz have been reported to have a transmission range of 120.0 cm in underwater environment. These experiments have been performed at the RESL at the University of Southern California.

Because of the use of lower frequencies, LoRa shows a three-fold increase in range compared with normal WiFi. The propagation of LoRa signal in (sea-)water is shown in figure 3.6. When this is compared with figure 3.5 in increase in range is found.

ELECTRIC CURRENT

Another way to communicate is through the use of electric current. Hagman, Elias [45] describes that seawater, as a conductive medium, can be subject to a modulated signal generated by a pair of transmitting electrodes, that launch a current field in the channel. If this current field is strong enough, the receiver — that also uses a pair of electrodes — could measure a potential difference and therefore receive the signal. Since electric current noise is extremely low in seawater, small current fields amplitudes are sufficient to receive information and a large data rate is achievable [45]. Since this type of transmission only works in a conductive medium, and the use case only specify that a dredge bot will be deployed in fresh water basins, electric current communication is not deemed a viable candidate.

ACOUSTIC COMMUNICATION

As is shown in Section 3.1.2, EMW have a very limited range in (sea) water, due to a high attenuation. Multiple sources such as Hagman, Elias [45], Claus [69] and Domingo [58] state that acoustic communication is therefore the preferred way. This type of communication makes use of SW, or AW, which are often described as vibration of molecules of the medium in which it travels — that is, in terms of the motion or displacement of the molecules. SW can also be analysed from the point of view of pressure. Indeed, longitudinal waves are often called pressure waves. The pressure variation is usually easier to

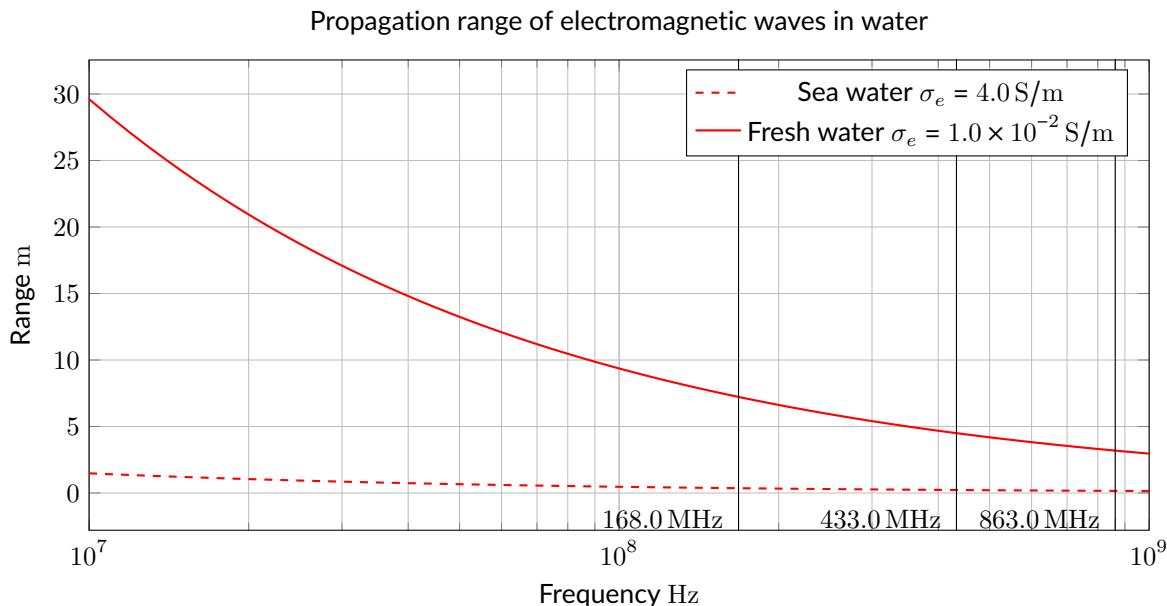


FIGURE 3.6: PROPAGATION RANGE OF LO-FI IN WATER

measure then the displacement [78]. This principle is used by hydrophones; These are in-effect microphones designed to be used underwater. Using piezoelectric transducers to convert pressure waves in to electricity. Although acoustic communication is the preferred method, there are a lot of challenges to overcome. According to Tetley and Calcutt [38] transmitting and receiving acoustic energy in seawater is affected by the often unpredictable ocean environment. Lanbo, Shengli, and Jun-Hong [41] and Edward Tucholski [31] both state that the speed of sound in the sea is not constant, but a function of temperature, pressure and salinity $v(T, P, S)$. Because the speed is not constant sound does not travel in a straight line. Acoustic communication can be summarized as follows:

PARAMETER	VALUE
Attenuation	A variable factor related to the transmitted power, the frequency of transmission, salinity of the seawater and the reflective consistency of the ocean floor.
Salinity of seawater	A variable factor affecting both the velocity of the AW and its attenuation.
Velocity of sound in salt water	This is another variable parameter. Acoustic wave velocity is precisely 1505.0 m/s at 15.0 °C and atmospheric pressure, but most echo-sounding equipment is calibrated at 1500.0 m/s
Reflective surface of the seabed	The amplitude of the reflected energy varies with the consistency of the ocean floor.
Noise	Either inherent noise or that produced by one's own transmission causes the signal-to-noise ratio to degrade, and thus weak echo signals may be lost in noise.
Frequency of transmission	This will vary with the system, i.e. depth sounding or Doppler speed log.
Angle of incidence of the propagated beam	The closer the angle to vertical the greater will be the energy reflected by the seabed.

3.2 SENSORS

In the following section a variety of sensor type, their workings and useful applications are presented. A selection is made for sensor types that can be used underwater, in an environment which is deprived of a GPS coverage.

The shortcomings and strength of the different sensor are often fused together with a complementary filter, where a mathematical filter is used to mix and merge the two values, or by use of a Kalman filter. The following sections, shortly describe the workings of an accelerometer, gyroscope magnetometer and a pressure sensor, which will be used in Section 3.3.1, where these sensor will be fused together with a Kalman filter, to obtain an accurate heading and positioning system.

While sensors described in Section 3.2.1 determine the state of a dredge bot; Namely its orientation and position. The sensors described in Section 3.2.2 discusses a variety of sensors which are needed to gauge the environment.

3.2.1 STATE SENSING

In order for a dredge bot to perform its tasks it has to be aware of its state. As described in Section 4.3.1, the state vector \vec{x} describes the position in a global reference frame and the orientation of the dredge

ACCELEROMETER

There are many different types of MEMS based accelerometer. The more expensive MEMS are laser and optical based, whilst cheaper models are piezoresistive, capacitive sensing and piezoelectric. Leccadito, Bakker, Niu, et al. [67] describes the working of a accelerometer as follows; The sensor can be thought of as a ball in a box. If the accelerometer meter is still and there are no forces present, the sensor will measure 0.0 m/s^2 on all three axes; The ball is suspended in air. If the sensor is suddenly moved, the ball will hit the wall with an opposing force compared to the movement. An acceleration can be measured because of Newtons second law $F = ma$.

In the scenario where there is no external forces present, the accelerometer would only measure the acceleration of the opposite direction of movement, however, on earth there is the external force of gravity pulling on the sensor. If the sensor is positioned on a flat surface with the z-axis aligned as up and down, x-axis left and right, and y-axis forward and back, gravity will always be in the negative z direction.

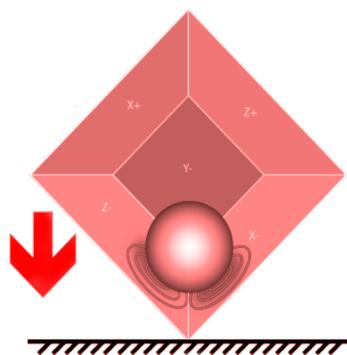


FIGURE 3.7: GRAVITATIONAL PULL ON MULTIPLE AXES [67]

Due to the gravitational pull an accelerometer can be used to calculate the heading. Because the sensed acceleration is divided amongst the walls of which the ball is in contact with, as is shown in figure 3.7. These measurements can be directly computed into position or Euler angles roll ϕ_{IMU} and pitch θ_{IMU} using trigonometry. Which is shown in equation 3.6. Which allows the magnetometer to calculate a heading angle, which will be described in section 3.2.1.

$$\begin{bmatrix} \psi_{IMU} \\ \theta_{IMU} \\ \phi_{IMU} \end{bmatrix} = \begin{bmatrix} \arctan\left(-\frac{a_y}{a_z}\right) \\ \arcsin\left(\frac{a_x}{\sqrt{a_x^2 + a_y^2 + a_z^2}}\right) \\ \text{Magnetometer Heading} \end{bmatrix} \quad (3.6)$$

Since acceleration can be integrated over time as velocity, which in turn can be integrated over time as a distance traveled. accelerometers can be used as a dead-reckoning device. Determining a location, with respect to a starting position, in a GPS deprived environment. Abyarjoo, Barreto, Cofino, et al. [75] states that the problem with accelerometers is that the measure both acceleration due to the device's linear movement and acceleration due to the earth's gravity, which is pointing toward earth. Since it cannot distinguish between these two accelerations, there is a need to separate gravity and motion acceleration by filtering. Which is also described by Nistler and Selekw [55], whom further states that it should be clear that the measurement for a robotic vehicle on an irregular terrain need to be processed further if they are to be used in the robot odometry system.

Possible sources of error with MEMS accelerometers are identified as effects of temperature and discretization of an analog signal to its digital representation. Abyarjoo, Barreto, Cofino, et al. [75] observed no drift of the signal but established that it contains a lot of noise. Kownacki [52] describes that a Kalman filter is a good candidate to filter the noise, using a gyroscope. Where the ADC stores a obtained analog value as a digital representation. This is usually done with a resolution between $2^{10}[\text{bit}]$ and $2^{16}[\text{bit}]$, resulting in a resolution of 1024, 2048 till 65536. But discretization of a continuous signal inherently degrades it.

GYROSCOPE

gyroscope has been used for many years in navigation. It usually involves a spinning object, that is tilted perpendicular to the spin, where the angle of the reference surface can be measured. Where the angle is affected by tilting or rotating. gyroscopes which are usually used in electronics, are so called MEMS. They are based on other principles such as a laser ring, which observe a phase shift between two laser beams sent in a circular path. These sensors are expensive and a cheaper alternative is a gyroscope which uses a piezoelectric sensor that works because of a Coriolis effect coupled with vibrations, tuning fork which measures the displacement of two objects.

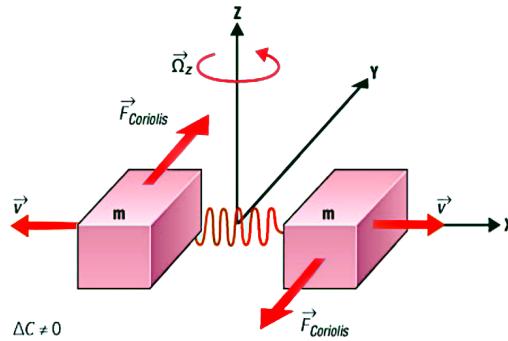


FIGURE 3.8: GYROSCOPE USING CORIOLIS EFFECT [67]

Leccadito, Bakker, Niu, et al. [67] tells that most MEMS gyroscopes are based on the tuning fork structure, where the Coriolis effect is used to measure ω . This is accomplished by two masses oscillating in opposite directions. When a rotation is applied, the masses are affected by the Coriolis force and the displacement is measured by a change in capacitance, as is shown in figure 3.8. From where the heading at a certain axis can be calculated using the Trapezoidal rule. Equation 3.7 illustrates how to obtain the current heading from a discrete sample set.

$$\theta_n = \int_{t_{n-1}}^{t_n} \omega dx = \sum_{t_{n-1}}^{t_n} \omega dt \approx \theta_{n-1} + (t_n - t_{n-1}) \left[\frac{\omega(t_{n-1}) + \omega(t_n)}{2} \right] \quad (3.7)$$

Abyarjoo, Barreto, Cofino, et al. [75] observed that the computed results drift over time. The explanation for this phenomenon is that the integration accumulates the noise over time and turns noise into the drift, which yields unacceptable results. Another source of drift is temperature related, Feng, Li, and Zhang [77] states that a gyroscope is sensitive to temperature variations, so the surrounding temperature variations lead to a bias drift of the gyroscope. Then as the error of the angular velocity, the drift causes error accumulation in the orientations. Where this drift is not linear with temperature. Equation 3.8 shows the model of a MEMS gyroscope drift, where ω_t , but unknown and B_d . Where n_s .

$$\omega = \omega_t + B_d + n_s \quad (3.8)$$

Abyarjoo, Barreto, Cofino, et al. [75] further states that the slow changing component of the gyroscope is not only related to the measured temperature of the MEMS, but also related to the temperature gradient of the surroundings. Because the temperature gradient and the rate of temperature variation have a linear relationship, the slow-changing component B_d can be modelled, as shown in equation 3.9. Where a, b, c are the parameters of the model Wei, Fang, and Li [32] and T is the measured temperature of the gyroscope in K and T' .

$$B_d = aT + bT' + c \quad (3.9)$$

Other sources of errors are the conversion from the generated analog signal to a digital representation. The ADC in a MEMS stores the obtained analog value as a discrete digital representation with a certain sequence of bits. This is usually done in word with a resolution between $2^{10}[\text{bit}]$ and $2^{16}[\text{bit}]$, resulting in a resolution of 1024, 2048 till 65536. Which should be stored in two registries. Discretization of an continuous signal inherently degrades its.

MAGNETOMETER

A magnetometer measure the strength of a magnetic field. Where a MEMS magnetometer operates by detecting the effects of the Lorentz force; Which results in a change in voltage or resonant frequency which can be measured electronically. Leccadito, Bakker, Niu, et al. [67] explains that a magnetometer coupled with an accelerometer can effectively calculate a heading angle. This is further explained by Konvalin [40] whom explain that raw magnetometer measurements cannot be used to calculate the heading angle due to the decrease in sensitivity as elevation and bank angles increase, introducing error. In order to obtain the correct heading a rotation must first be applied removing the bank angle, after which removes the pitch angle. Which can be obtained by equation 3.6. Where the heading, or yaw ψ_{IMU} can be calculated following equations 3.10 through 3.12. Where x_m , y_m and z_m are the raw magnetometer values.

$$x_h = x_m \cos \theta_{IMU} + z_m \sin \theta_{IMU} \quad (3.10)$$

$$y_h = x_m \sin \phi_{IMU} \sin \theta_{IMU} + y_m \cos \phi_{IMU} - z_m \sin \phi_{IMU} \cos \theta_{IMU} \quad (3.11)$$

$$\phi_{IMU}(y_h, x_h) = \begin{cases} \arctan\left(\frac{y_h}{x_h}\right) & \text{if } x_h > 0 \\ \arctan\left(\frac{y_h}{x_h}\right) + \pi & \text{if } x_h < 0, y_h \geq 0 \\ \arctan\left(\frac{y_h}{x_h}\right) - \pi & \text{if } x_h < 0, y_h < 0 \\ +\frac{1}{2}\pi & \text{if } x_h = 0, y_h > 0 \\ -\frac{1}{2}\pi & \text{if } x_h = 0, y_h < 0 \\ \text{undef} & \text{if } x_h = 0, y_h = 0 \end{cases} \quad (3.12)$$

The main sources of error using a magnetometer are distortions of the earths magnetic field, which can be classified in two categories: soft- and hard iron. Where hard iron distortions can be described as a constant additive disturbance in the magnetic field of the magnetometer. Which can be created by ferrous materials around the sensors. Such as the construction of a crawler and the casing of the electronics and hydraulics. Which can create its own magnetic field and adds to the sensors magnetic fields and is in constant position relative to the sensor. According to Leccadito, Bakker, Niu, et al. [67], such a distortion is constant and can be eliminated by a constant offset or bias. In order to eliminate the offset equation 3.14 can be used. Where \vec{m} and \vec{m}_{hi} . Which is the offset from center obtained by averaging the minimum and maximum value in n calibration values. Obtained by rotating the sensor in the iron casing. Since this value will be constant, it can be stored in memory.

$$\vec{m} = \begin{bmatrix} x_m \\ y_m \\ z_m \end{bmatrix} \quad (3.13)$$

$$\vec{m}_{hi} = \vec{m} - \frac{\min(\vec{m})_n + \max(\vec{m})_n}{2} \quad (3.14)$$

soft iron distortions are different from hard iron disturbances, since they don't necessarily generate their own magnetic field. Leccadito, Bakker, Niu, *et al.* [67] describes that soft iron effects on the sensor are determined by the orientation of the materials, and it's usually a perturbation of a circular magnetic field to an ellipse. Calculating the soft iron distortion is computationally more expensive than the hard iron elimination.

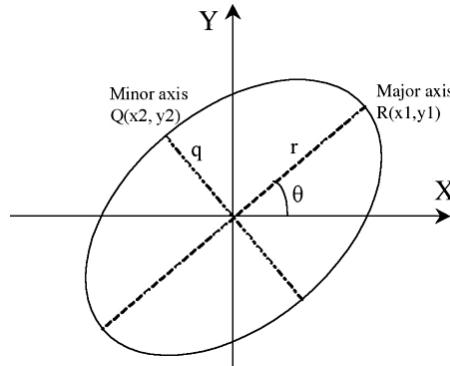


FIGURE 3.9: SOFT IRON DISTORTION [40]

it's assumed that tilt compensation (eq. 3.12) and hard iron offset (eq. 3.14) are already performed at this stage and that the center of the ellipse is positioned at point (0, 0). Which is drawn in figure 3.9. The first sub sequential step is to calculate the magnitude of each point on the ellipse and finding the smallest and greatest value, using equation 3.15 and 3.16. The y-index of the greatest magnitude should be stored in y_1 , which, together can be used to calculated θ_{si} , as is shown in equation 3.17. By scaling and rotating the hard iron \vec{m}_{hi} vector a correct heading \vec{m}_{si} can be calculated, which is shown in equation 3.18.

$$r_{si} = \max \left(\sqrt{x_n^2 + y_n^2} \right) \quad (3.15)$$

$$q_{si} = \min \left(\sqrt{x_n^2 + y_n^2} \right) \quad (3.16)$$

$$\theta_{si} = \arcsin \left(\frac{y_1}{r_{si}} \right) \quad (3.17)$$

$$\vec{m}_{si} = \frac{\vec{q}_{si}}{r_{si}} \begin{bmatrix} \cos \theta_{si} & \sin \theta_{si} & 0 \\ -\sin \theta_{si} & \cos \theta_{si} & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{m}_{hi} \quad (3.18)$$

PRESSURE SENSOR

White [57] describes a fluid pressure p as the normal shear stress on any plane through a fluid element at rest is a point property, which is taken positive for compression, by convention. Which can be described by equation 3.19. Here p is the pressure at a certain depth, which is comprised of the specific weight of water $\gamma_w(T)$ as a function of temperature, and the total amount of water on top of that point z .

$$p = p_a - \gamma_w(T)z \quad (3.19)$$

Since pressure is a function of γ_w special consideration regarding the impact of soil disturbance, due to dredging activities, in water has to be made. Since the specific weight of the water column above the sensor changes when sediment is mixed with water above the pressure sensor. MTI dredging specialists Dr. ir. van Wijk and ir. Hoftsra; Both estimate that the disturbed sediment won't drift higher than 2.0 m for an sediment with an *in situ* specific weight of $\gamma_{sw} = 1400.0 \text{ N/m}^3$. That mixture will in all likelihood have a specific weight of $\gamma_m = 1200.0 \text{ N/m}^3$. Because the specific weight of water is $\gamma_w = 1000.0 \text{ N/m}^3$, the error when calculating depth with a pressure sensor is depended on the position of the sensor with regards to the bottom.

Using equation 3.20 where Δp is the pressure difference between the specific weight of a column of water γ_w compared with a column of water and sediment γ_m of a certain height z_p . Where the specific weight consists of the density of a fluid ρ_w for water or ρ_m for mixture multiplied with a gravitational acceleration vector g . When the allowed z_ϵ is known. A height for the pressure sensor, with regards to the top fluid column can be obtained. it's estimated that an acceptable error in depth readings is 200.0 mm, when using equation 3.20 gives an minimum sensor height of 1.9 m from the soil bed. Which indicates that the sensor should be placed at the top of a dredge bot, away from the disturbance source.

$$\left. \begin{array}{lcl} \Delta p & = & (\gamma_w - \gamma_m) z_p \\ \gamma_w & = & \rho_w g \\ \gamma_m & = & \rho_m g \\ z_\epsilon & = & \frac{\Delta p}{\gamma_w} \end{array} \right\} z_\epsilon = \frac{-(\rho_m - \rho_w) z_p}{\rho_w} \implies z_p = \frac{z_\epsilon \rho_w}{\rho_m - \rho_w} \quad (3.20)$$

Three types of pressure measurements are usually performed, according to Webster [14]:

TYPE	DESCRIPTION
Absolute pressure	Where the pressure is measured against an perfect vacuum where pressure is zero.
Gage pressure	Is the pressure difference between the point of measurement and the ambient.
Differential pressure	Is the pressure difference between two points, one of which is chosen to be the reference. In reality, both pressures can vary, but only the pressure difference is of interest here.

Since pressure is defined as the force per unit area, the most direct way of measuring pressure is to isolate an area on an elastic mechanical element for the force to act on. The deformation of the sensing element produces displacements and strains that can be precisely sensed to give a calibrated measurement of the pressure [14].

NOTE 3.1: SCOPE

Although there are a multitude of pressure sensing techniques, such as: seals, snubbers, bellow, bourdon, helical, diaphragm, differential, electronic and manometers. This research will focus on diaphragm and electronic pressure sensors, since these are commonly used throughout the industry and easily integrated in a crawler. Where the focus lies on behavior and characteristics.

Pressure sensors that depend on deflection of a diaphragm have been used for centuries, the last few decades the elastic hysteresis, friction and drift effects have been reduced to $\pm 0.1\%$. This is mainly due to the use of a microprocessor, which applies memorized non-linearity corrections [24].

Detection methods are usually capacitive pressure sensors, which are highly accurate (better than 0.1%) and can cover a high pressure range, from nearly vacuum (1.0×10^{-1} to 1.0×10^7) Pa. These sensors rest on the principle, where a metal or silicon diaphragm serves as the pressure sensing element and is regarded as one electrode of a capacitor. The other electrode is stationary and usually consists of a deposited metal layer on a ceramic or glass substrate. When a pressure is applied the diaphragm deforms and the changes in between electrodes is changed which results in a change in capacitance.

Where an alternative are the piezoresistive pressure sensors, which are the most common type of pressure sensor in use. These sensors measure the pressure by measuring the change in electric resistance of a material when stresses or strains are applied. Both Webster [14] and Liptak [24] state that semiconductor, such as silicon and germanium, are by far the most appealing sensing elements in this type of sensor. The most attractive characteristic of semiconductors is their sensitivity, which is close to 100 times greater than that of metallic wires.

3.2.2 EXTERNAL SENSOR

A crawler needs to be aware of its surroundings; It needs to sense how far objects and landmarks are in respect to its own position and orientation. Awareness of its environment can be used to minimize dead-reckoning drift, defined the work area and help avoid collisions.

NOTE 3.2: SCOPE

Although there are multiple examples of AUVs that make use of EMW, light or computer vision to help them sense its environment, these are deemed not usable for a crawler. EMW has a limited range in salt water environments (discussed in Section 3.1.2), light will scatter due to diffraction created by floating sand particles and computer vision can be problematic due to limited light sources and low contrast conditions. Which is also a result of floating sand particles. The focus of this research will therefore lie on acoustic sensing.

3.3 LOCATION UNDER UNCERTAINTY

Due to the absence of an ubiquitous global localization system such as GPS in underwater environments, crawler navigation is confined to these three primary methods: (1) dead-reckoning, (2) time of flight acoustic navigation, and (3) geophysical navigation.

The most obvious and longest established technique is dead-reckoning, which consist in integrating vehicle velocity measurements from sensors such as accelerometer and gyroscope to obtain new position estimates. The problem with exclusive reliance on dead-reckoning is that the position error increases without bound as the distance travelled by the crawler [71]. It will be illustrated in section 3.3.1 that the position error can be limited by making use of sensor fusion. But this won't be enough. The effects of sporadically position updating using stationary LBL and USBL is shown as well. Both LBL as USBL make use of acoustic energy, which is described in more detail in section 3.1.2. Because acoustic energy is known for its excellent travel characteristics underwater it's common practice to deploy those transponders as beacons, such that they can update the position and bound the dead-reckoning error.

geophysical navigation such as TRN and SLAM are up and coming methods which show potential. These use the characteristic of a terrain, perceived through there sensors, to obtain their position. These methods are not discussed in this paper.

Other methods for navigation under uncertainty are based on probabilities taken into account *a priori* known characteristics of sensors and actuators such as LQG-MP and RRT. According to Galceran, Nagappa, Carreras, *et al.* [65] these methods are theoretically satisfactory but they require discretization of the environment, and will, as a result suffer from scalability problems. They propose the use of *a priori* known bathymetric map. Which classifies this method as off-line and therefore unsuitable to be employed for an autonomous operating crawler. These will therefore not be described in this thesis.

Recent studies have been focused on minimizing uncertainties using multiple robots, such as the leap-frog strategy proposed by Tully, Kantor, and Choset [49], which uses a team of three robots where two alternating robots act as stationary beacons. Others like Wei Gao, Yalong Liu, and Bo Xu [74] use a single surface which act as an communication and navigation aid. it's quite common to filter sensor readings and state vectors from the multiple robots using a Kalman filter.

But this chapter begins with a dive into Kalman filter. The sections below describe how the state representation \vec{x}_k of a crawler can be obtained using a Kalman filter, which fuses multiple sensors together. It will then explore how the growth of errors can bound, using a sporadically obtained position estimate from a alternative source, such as moving or stationary beacons.

3.3.1 LOCALIZATION REFINEMENT USING KALMAN FILTERS

A crawler has multiple sensors on-board to establish were and what its orientation is; These will in a likelihood be a gyroscope, accelerometer, magnetometer and a pressure sensor. It was established in section 3.2.1 that each of these sensors have their own limitations and strengths. it's common practice to fuse multiple sensors together, to counteract these limitations with strengths of the other sensors. Kalman filter or as they are also known LQE, are a tried and practice method to achieve this.

Section 3.3.1 explains the filter using a simple example of a falling ball with only gravity working on the ball.

3.3.2 BASIC KALMAN FILTERING

Before a Kalman filter can be designed it's important that the basics are explained. The section will feature a short description of the background and workings of a Kalman filter and quaternion.

In 1960, R.E. Kalman published his paper Kalman [2] – “A new approach to linear filtering and prediction problems”. In this paper he described a recursive solution to the discrete-data linear filter problem. Welch and Bishop [33], d’Andréa-Novel and Lara [63], Chui and Chen [13], Grewal and Andrews [79] all describe how Kalman filters have had a huge impact on control theory, and have been subject to extensive research and application. The paragraphs below are based on the theory proposed by Kalman [2].

A Kalman filter can be used to control a dynamic model, especially those represented by systems of linear differential equations. These generally come from the laws of physics. The real-world dynamics are used to model the state dynamics. Which should contain a fairly faithful replication of the true system dynamics. The state of a falling object in one dimension can be described with the state $\vec{x}_k = [s_z, v_z]^T$. Here s_z and v_z .

A Kalman filter works by estimating the state of a process based on *a priori* states. it's in effect an optimal estimator based on a prediction made from the previous input and current input. The Kalman filter addresses the general problem of trying to estimate the state $\hat{x}_k \in \mathbb{R}^n$ of a discrete-time controlled process that is governed by the linear stochastic difference equation 3.21. Here \mathbf{F} is a state transition model which is applied to the previous state \vec{x}_{k-1} , to estimate the current state. Where \mathbf{B} is the control-input model which is applied to the control vector \vec{u}_k . The process noise \vec{w}_k is assumed to be white, with a normal probability distribution. The \mathbf{Q}_k is the process noise covariance. Each predicted step is updated with a measurement, which is shown in equation 3.22. Here the measurement $\vec{z}_k \in \mathbb{R}^m$, where \mathbf{H} is the observation model, which maps the true state space \vec{x}_k into the observed space, whilst taking into account the observation noise \vec{v}_k , which is assumed to be unrelated to \vec{w}_k , and is white with a normal probability distribution. Where \mathbf{R} is the measurement noise covariance.

$$\hat{x}_k = \mathbf{F} \vec{x}_{k-1} + \mathbf{B} \vec{u}_k + \vec{w}_k, \quad p(\vec{w}_k) \sim N(0, \mathbf{Q}_k) \quad (3.21)$$

$$\vec{z}_k = \mathbf{H} \vec{x}_k + \vec{v}_k, \quad p(\vec{v}_k) \sim N(0, \mathbf{R}) \quad (3.22)$$

Figure 3.10, shows the algorithm as a flow diagram. It starts with and initial assumption of the state \vec{x}_0 and \mathbf{P}_0 , which is the initial state of the error covariance matrix. Which can be described as a measure of the estimated accuracy of the state estimate.

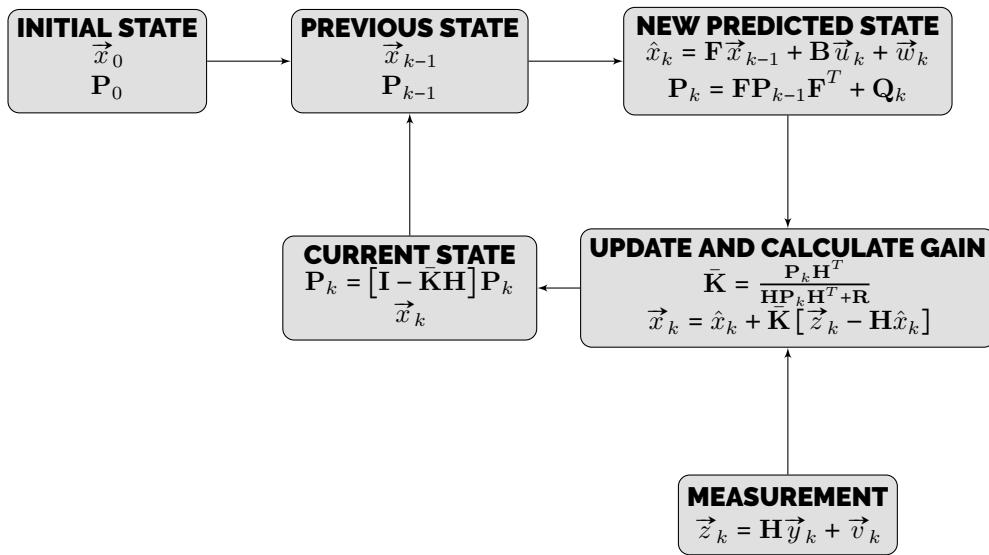


FIGURE 3.10: FLOW OF A KALMAN FILTER

These initialized values are fed in the loop as a **previous state**. With which a **new predicted state** is estimated using equation 3.21. If the previous example of a falling object in one dimensions is used, with the state $\vec{x}_k = [s_z, v_z]^T$. A prediction can be made of the position in the next time step. Which will follows the equation $s_{z,k} = s_{z,k-1} + v_{z,k-1} \Delta t + \frac{1}{2} a_{z,k-1} \Delta t^2$ and the new velocity $v_{z,k} = v_{z,k-1} + a_{z,k} \Delta t$. For simplicity sake the process noise \vec{w}_k is set to zero. The matrix \mathbf{F} is used to map the previous state to the new state. Where the matrix \mathbf{B} is used to map the control variable \vec{u}_k to the new state. These variable dictate the change; In the case of our example it will be an acceleration due to gravity $a_z = -g$. Equation 3.23 illustrates the new estimation of the state of our example. Here Δt is an incremental time step.

$$\hat{x}_k = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{z,k-1} \\ v_{z,k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \Delta t^2 \\ \Delta t \end{bmatrix} [a_{z,k}] + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} s_{z,k-1} + v_{z,k-1} \Delta t + \frac{1}{2} a_{z,k} \Delta t^2 \\ v_{z,k-1} + a_{z,k} \Delta t \end{bmatrix} \quad (3.23)$$

The new predicted error of the estimate, known as the error covariance matrix \mathbf{P}_k , is used to map the covariance between the i^{th} and j^{th} elements of the state vector \vec{x}_k . In this example it's initial assumed that error between the state of its position s_z and the velocity are unrelated. The assumption is also made that the position has an error of σs_z and the velocity σv_z . From this, a simple error covariance matrix can be constructed. Which can be used in equation 3.24, with which a new error covariance matrix can be calculated, as is shown in equation 3.25. The noise matrix \mathbf{Q}_k is set to zero.

$$\mathbf{P}_k = \mathbf{F}\mathbf{P}_{k-1}\mathbf{F}^T + \mathbf{Q}_k \quad (3.24)$$

$$\begin{aligned} \mathbf{P}_k &= \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{s_z}^2 & 0 \\ 0 & \sigma_{v_z}^2 \end{bmatrix} \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix}^T + \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} = \\ &\begin{bmatrix} \Delta t^2 \sigma_{v_z}^2 + \sigma_{s_z}^2 & \Delta t \sigma_{s_z}^2 \\ \Delta t \sigma_{s_z}^2 & \sigma_{s_z}^2 \end{bmatrix} \approx \begin{bmatrix} \Delta t^2 \sigma_{v_z}^2 + \sigma_{s_z}^2 & 0 \\ 0 & \sigma_{s_z}^2 \end{bmatrix} \end{aligned} \quad (3.25)$$

Once the prediction of a new state and error covariance matrix is made, The **measurements** can be calculated. It is important to note that only the inputs and outputs of the system can be measured. Equation 3.26 shows the measured values, mapped to the state space \vec{z}_k , where \mathbf{H} is the measurement sensitivity matrix defining the linear relationship between the state of a dynamic system and the measurements that can be made, which for now is set equal to a 2×2 identity matrix. Lets assume that for our example only the position can be measured $s_{z,m,k}$ and that the measurement noise is assumed to be zero.

$$\vec{z}_k = \mathbf{H}\vec{y}_k + \vec{v}_k \quad (3.26)$$

$$\vec{z}_k = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{z,m,k} \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} s_{z,m} \\ 0 \end{bmatrix} \quad (3.27)$$

With the predicted state and the obtained measurements a new state can be estimated. During this **update** phase, we determine how much weight the Kalman filter needs to put in to its measurements compared to its predicted state. This can be done by calculating the $\bar{\mathbf{K}}$ Kalman gain. It can be calculated with equation 3.28. Where \mathbf{R} is the covariance matrix of observational (measurement) uncertainty.

$$\bar{\mathbf{K}} = \frac{\mathbf{P}_k \mathbf{H}^T}{\mathbf{H} \mathbf{P}_k \mathbf{H}^T + \mathbf{R}} \quad (3.28)$$

$$\bar{\mathbf{K}} = \frac{\begin{bmatrix} \Delta t^2 \sigma_v^2 + \sigma_s^2 & 0 \\ 0 & \sigma_s^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T}{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta t^2 \sigma_v^2 + \sigma_s^2 & 0 \\ 0 & \sigma_s^2 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}^T \begin{bmatrix} \sigma_{s,m}^2 & 0 \\ 0 & \sigma_{v,m}^2 \end{bmatrix}} = \begin{bmatrix} \frac{\Delta t^2 \sigma_v^2 + \sigma_s^2}{\Delta t^2 \sigma_v^2 + \sigma_s^2 + \sigma_{s,m}^2} & 0 \\ 0 & \frac{\sigma_s^2}{\sigma_v^2 + \sigma_{v,m}^2} \end{bmatrix} \quad (3.29)$$

The Kalman gain obtained in equation 3.28, is used in equation 3.30. Where a new state is calculated by taking the predicted state \vec{x}_k calculated with equation 3.23 and adding the innovation multiplied with the Kalman gain $\vec{z}_k - \mathbf{H}\hat{x}_k$. Innovations are the differences between observed and predicted measurements. Grewal and Andrews [79] states that they are the carotid artery of a Kalman filter. They provide an easily accessible point for monitoring vital health status without disrupting normal operations, and the statistical and temporal properties of its pulses can tell us much about what might

be right or wrong with a Kalman filter implementation.

From the worked out 1-dimensional example it becomes apparent, that the state variable, calculated in equation 3.31 are weighted average between the measurements and the prediction, normalized against the error of the covariance, between the state variables.

$$\vec{x}_k = \hat{x}_k + \bar{\mathbf{K}} [\vec{z}_k - \mathbf{H}\hat{x}_k] \quad (3.30)$$

$$\begin{aligned} \vec{x}_k &= \begin{bmatrix} s_{z,k-1} + v_{z,k-1}\Delta t + \frac{1}{2}a_{z,k}\Delta t^2 \\ v_{z,k-1} + a_{z,k}\Delta t \end{bmatrix} + \begin{bmatrix} \frac{\Delta t^2\sigma_v^2 + \sigma_s^2}{\Delta t^2\sigma_v^2 + \sigma_s^2 + \sigma_{s,m}^2} & 0 \\ 0 & \frac{\sigma_v^2}{\sigma_v^2 + \sigma_{v,m}^2} \end{bmatrix} \dots \\ &\left[\begin{bmatrix} s_{z,m,k} \\ 0 \end{bmatrix} - \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} s_{z,k-1} + v_{z,k-1}\Delta t + \frac{1}{2}a_{z,k}\Delta t^2 \\ v_{z,k-1} + a_{z,k}\Delta t \end{bmatrix} \right] = \dots \quad (3.31) \\ &\begin{bmatrix} \frac{s_{z,m}\Delta t^2\sigma_v^2 + 2a_{z,k}\Delta t^2 + 2v_{z,k-1}\Delta t\sigma_{s,m}^2 + s_{z,m,k}\sigma_s^2 + s_{k-1} + \sigma_{s,m}^2}{\Delta t^2\sigma_v^2 + \sigma_s^2 + \sigma_{s,m}^2} \\ \frac{\sigma_{v,m}^2(v_{z,k-1} + 2a_{z,k}\Delta t)}{\sigma_v^2 + \sigma_{v,m}^2} \end{bmatrix} \end{aligned}$$

This newly obtained state, or **current state** \vec{x}_k can be used in the next iteration. During this phase a new process covariance matrix \mathbf{P}_k is calculated with equation 3.32. Where the matrix \mathbf{I} is a 2×2 identity matrix. Both the new state, obtained from equation 3.30 and the newly obtained process covariance matrix are set as the previous iteration.

$$\mathbf{P}_k = [\mathbf{I} - \bar{\mathbf{K}}\mathbf{H}]\mathbf{P}_k \quad (3.32)$$

$$\begin{aligned} \mathbf{P}_k &= \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \begin{bmatrix} \frac{\Delta t^2\sigma_v^2 + \sigma_s^2}{\Delta t^2\sigma_v^2 + \sigma_s^2 + \sigma_{s,m}^2} & 0 \\ 0 & \frac{\sigma_v^2}{\sigma_v^2 + \sigma_{v,m}^2} \end{bmatrix} \right] \left[\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right] \begin{bmatrix} \Delta t^2\sigma_v^2 + \sigma_s^2 & 0 \\ 0 & \sigma_s^2 \end{bmatrix} = \dots \\ &\begin{bmatrix} \frac{\sigma_{s,m}^2(\Delta t^2\sigma_v^2 + \sigma_s^2)}{\Delta t^2\sigma_v^2 + \sigma_s^2 + \sigma_{s,m}^2} & 0 \\ 0 & \frac{\sigma_v^2\sigma_{v,m}^2}{\sigma_v^2 + \sigma_{v,m}^2} \end{bmatrix} \quad (3.33) \end{aligned}$$

The above described example of a falling ball can be simulated with a Python script. Such a script can be found in appendix ???. Results from such a simulation are shown in figure 3.11. Were it is clearly evident that the estimated Kalman value, of the position, is a better estimate then the measured values. According to Roger R Labbe jr [86] an effective way to measure the results of a simulated Kalman filters, is the NEES. Which can be calculated with equation 3.35, where \tilde{x}_k is the error, or difference, between the ground truth state vector $\vec{x}_{g,k}$ and the estimated filter value \hat{x}_k , squared and multiplied with the inverse of the process covariance matrix \mathbf{P}_k ; All evaluated at time k .

$$\tilde{x}_k = \vec{x}_{g,k} - \hat{x}_k \quad (3.34)$$

$$\epsilon_{N,k} = \tilde{x}_k \mathbf{P}_k^{-1} \tilde{x}_k \quad (3.35)$$

$$\bar{\epsilon}_N = \frac{1}{k} \sum_1^k \epsilon_{N,k} \leq n_x \quad (3.36)$$

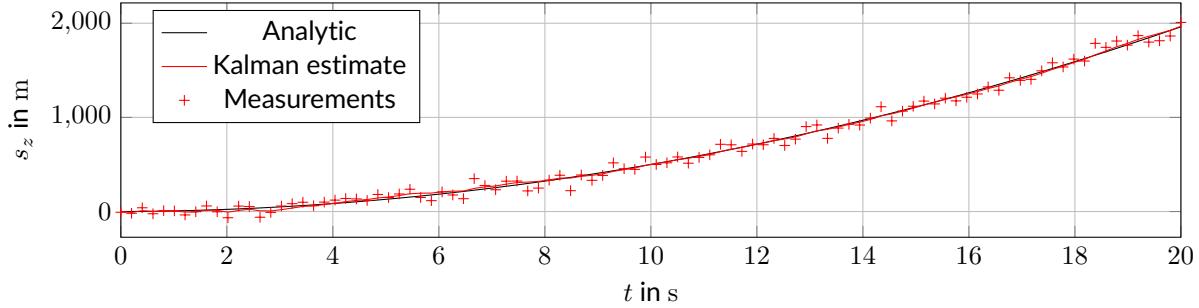


FIGURE 3.11: COMPARISON OF ESTIMATED, MEASURED AND REAL POSITION

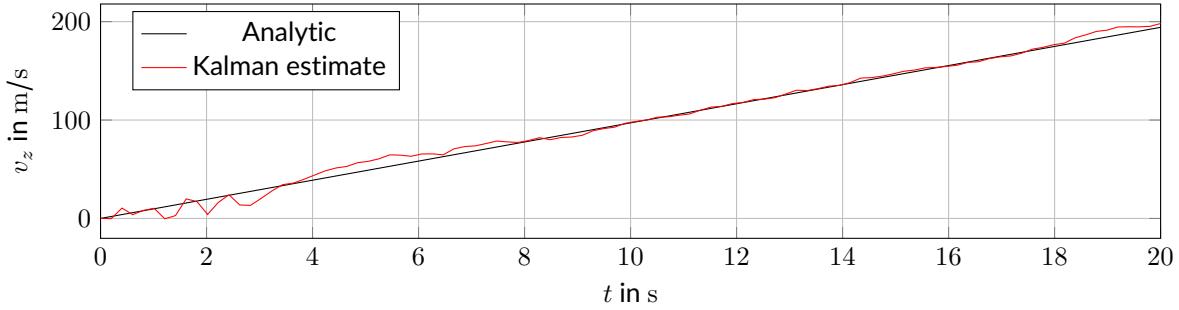


FIGURE 3.12: COMPARISON OF ESTIMATED AND REAL SPEED

This means that if the covariance matrix gets smaller, NEES gets larger for the same error. A covariance matrix is the filter's estimate of its error, so if it's small relative to the estimation error then it's performing worse than if it's large relative to the same estimation error. Equation 3.35 gives a scalar for each time step, which is said to be *chi-squared distributed with n degrees of freedom*. The average NEES value $\bar{\epsilon}_N$ should be less than number of elements in the state space vector n_x , as is shown in equation 3.36. The performance of our example or the $\bar{\epsilon}_N = 1.3$ is shown in figure 3.13.

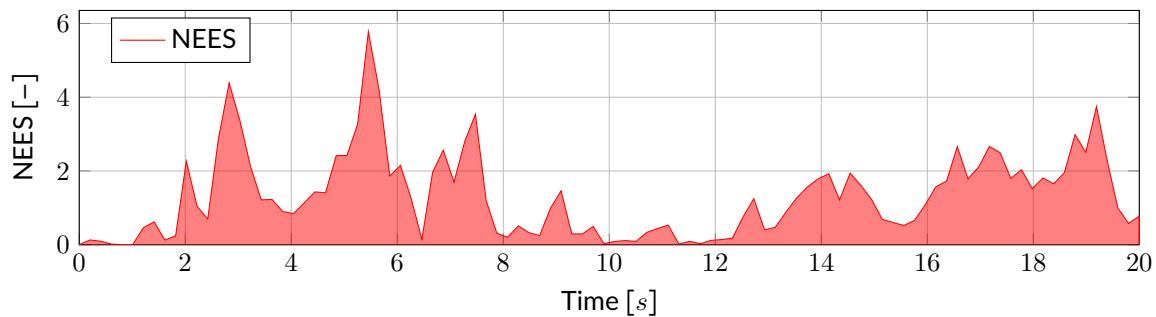


FIGURE 3.13: NEES for Kalman filter

3.4 COVERAGE PATH PLANNING

Two of the use cases described in the introduction dictate that the crawler has to cover the whole area preferably with an optimal path. This type of path planning problem differs from the in Chapter 3.4

described problem. Choset describes the task of determining a path that passes an effector (e.g., a robot, a detector, etc.) over all points in a free space as a coverage path problem [29], hence that this task is called a CPP. This type of task can be found as an integral part of many robotic applications such as vacuum cleaning robots, painter robots, autonomous underwater vehicles creating image mosaic, demining robots, lawn mowers, automated harvesters, window cleaners and inspection of complex underwater structures [64].

All these type of robots need to cover a complete region in order to perform their tasks. According to Cao, Huang, and Hall [6] such a mobile robot should use the following criteria, for a region filling operation

1. The mobile robot must move through an entire area, i.e., the overall travel must cover a whole region.
2. The mobile robot must fill the region without overlapping paths.
3. Continuous and sequential operations without any repetition of paths is required of the robot.
4. The robot must avoid all obstacles in a region.
5. Simple motion trajectories (e.g., straight lines or circles) should be used for simplicity in control.
6. An “optimal” path is desired under the available conditions. it’s not always possible to satisfy all these criteria for a complex environment. Sometimes a priority consideration is required.

Galceran [72] describes that these types coverage algorithms can be classified as *heuristic* or *complete* depending on whether or not the provable guarantee complete coverage of the free space. At the same time they can be classified as off-line or on-line. off-line algorithm rely on only on stationary information, and the environment is assumed to be known. Usually on-line algorithms are needed if some kind of adaptivity to the requirement is required. On-line algorithms usually utilize real-time sensor measurements. Thus these algorithms can also be called *sensor-based coverage algorithms*. on-line coverage algorithms are in effect “divide and conquer” strategies, which Wong and MacDonald [27] describes as a powerful technique used to solve many problems and many mapping procedures carry out a process of space decomposition, where a complex space is repeatedly divided until simple sub-regions of a particular type are created. The problem at hand is then solved by applying a simpler algorithm to the simpler sub-regions.

Since an autonomous operating crawler can be stationed in different environments with multiple unknown obstacles, the focus of this chapter lies on on-line or sensor-based coverage algorithms from which the following are identified:

- morse-based cellular decomposition
 - On-line Morse-based boustrophedon decomposition
 - Morse-based cellular decomposition combined with generalized Voronoi diagram
- Landmark-based topological coverage
 - Slice decomposition
 - On-line topological coverage algorithm
- Grid-based methods
 - Grid-based coverage using spanning trees
 - Neural network-based coverage on grid maps
- Coverage under uncertainty
- Multi-robot methods

3.4.1 MORSE-BASED CELLULAR DECOMPOSITION

Morse-based cellular decomposition is mostly based upon the following method exact or approximate cellular decomposition Acar, Choset, Rizzi, *et al.* [22] State that exact cellular decompositions represent the free space of a robot by dividing it into non-overlapping region sub-level cells such that the union

of the cells fills the free space. Complete coverage is then reduced to ensuring that the robot visits each cell. These cells are constructed using Morse function, a function for which all critical point are non-degenerate and all critical levels are different.

Morse-functions are visualized by Nicolaescu [37] as follows: Suppose M is a smooth, compact manifold which is assumed to be embedded in a Euclidean space \mathcal{E} , and from which we would like to understand some basic topological invariants. This is done with a “slicing” technique.

Were a unit vector \vec{u} is fixed in \mathcal{E} and which start slicing M with the family of hyperplanes perpendicular to \vec{u} . Such a hyperplane will in general intersect M along a submanifold (slice). The manifold can be recovered by continuously stacking the slices on top of each other in the same order as they were cut out of M .

If this collection of slices is visualized as a deck of cards with various shapes, which are piled up in the order that they were produced, there will be an increasing stack of slices. As this stack grows, it can be observed that at certain moments in time the shape suffers a qualitative change. The theory proposed by Morse extracts quantifiable information, through studying the evolution of this growing stack of slices.

Each moment in time that this pile changes is called a critical value, which correspond to moments in time when the hyperplane intersect tangentially. These points marks the boundary of a cell. Acar, Choset, Rizzi, et al. [22] states that Morse theory assures that between those critical point “merging” and “severing” of slices does not occur and that a robot can trivially perform simple motions, such as back and forth motions between critical points and thus guarantee complete coverage of a cell. Hence this method is dubbed Morse-based cellular decomposition

The above described method is depicted in figure 3.14 (a). Such an environment can be represented with a graph such as shown in figure 3.14 (b). Each critical value corresponds with a node while a cell is represented by an edge.

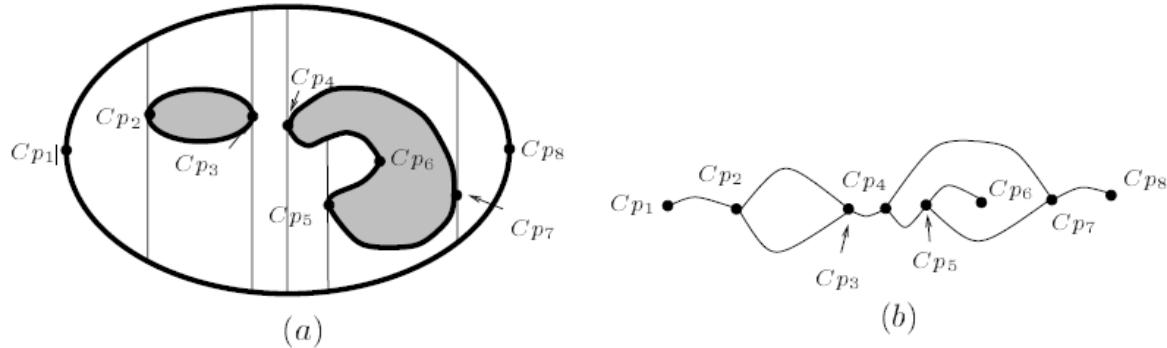


FIGURE 3.14: (a) Exact cellular decomposition, (b) Graph representation

The above describe technique has a minor short-coming, Choset, Acar, Rizzi, et al. [16] states that this method may result in many small cell, such as cell 9 shown in figure 3.15, which can seemingly be “clumped” into neighbouring cells. Reorganizing the cells can result in a shorter (more efficient) path to cover the same area. To address this issue, the BCD approach was introduced.

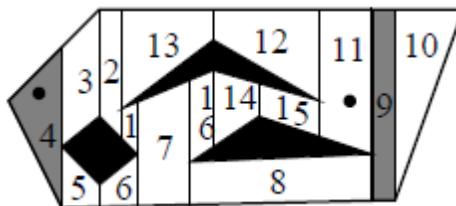


FIGURE 3.15: TRAPEZOIDAL DECOMPOSITION OF BOUND FREE SPACE[16].

Boustrophedon which literally means “the way of the ox” merges these cells, such that a more optimal path can be found. This can be done by using different Morse function, which results in different slice shapes and therefore different cell decompositions, as is shown in figure 3.16 [64][15][22]. Such as spiral, spike or squarel.

Once each cell is identified a strategy for the infill is executed, which is described by Huang [20] as coverage paths. Each region [or cell red.] is decomposed into sub-regions, a traveling salesman

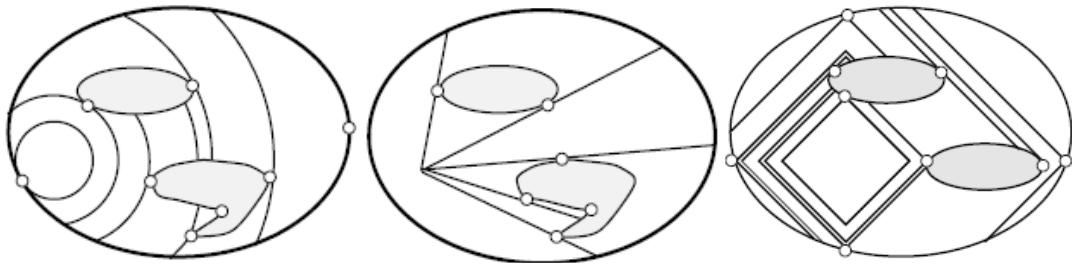


FIGURE 3.16: SPIRAL, SPIKE AND SQUAREL [22]

algorithm, is applied to generate a sequence of sub-regions to visit, and a coverage path is generated from this sequence that covers each subregion in turn. Huang [20] claims that turns take a significant amount of time: the robot must slow down, make the turn and accelerate. Thus by minimizing the number of turns, which are proportional to the altitude of a subregion, an optimal path can be found.

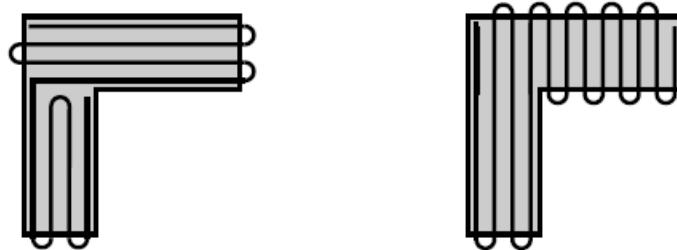


FIGURE 3.17: DIFFERENT SWEEP DIRECTIONS [20]

This is done by first creating an adjacency graph G , generated with the Morse function, see figure 3.14 (b). This graph is split in two G_1 and G_2 sub-graphs. These contain all edges from G except those that connect a node from G_1 to G_2 . With this definition the minimum sum of altitudes can be stated as equation 3.37. Where i iterates over all possible ways to split graph G and $C(G)$ returns the cost of covering all cells corresponding to nodes in G , once an optimum is found, movements of the sub-regions are implemented.

Let G_1 and G_2 be a subset of graph G which consist of all the nodes that needs to be visited and let C be a function that calculates the cost in movement S , which iterative over i for all possible combinations of G_1 and G_2 .

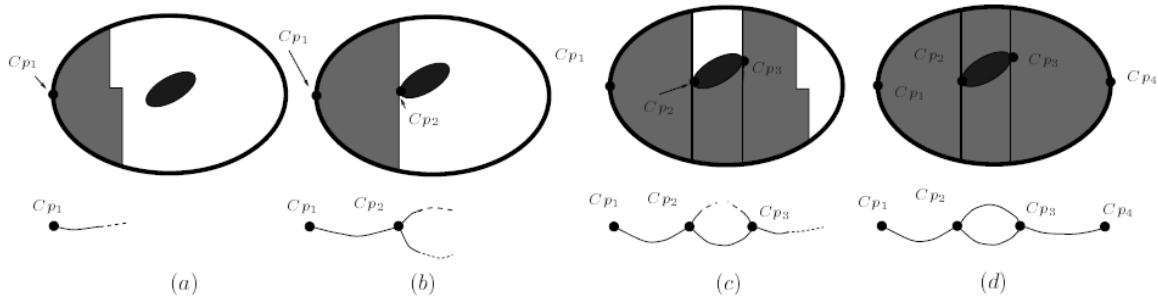
$$S(G) = \min \left\{ C(G), \min_i S(G_1^i) + S(G_2^{i'}) \right\} \quad (3.37)$$

So far it assumed that the environment is known *a priori* which labels this method as an off-line method. While the use case described in Chapter 1, dictates that the crawler encounters unknown obstacles.

ON-LINE MORSE-BASED BOUSTROPHEDON DECOMPOSITION

Acar and Choset [21] describe a method which allows the use of above portrayed Morse-based cellular decomposition in an unknown dynamically changing environment. Critical point sensing, is a way to determine critical points based on a sweep direction and an omnidirectional range sensor. These can be detected when the sweep direction and the surface normal $\nabla_m(x)$ of an obstacle are parallel.

On-line region coverage is depicted in figure 3.18 which shows an incremental sweep as part of an on-line BCD (a) The robot starts to cover the space at the critical point Cp_1 and instantiates an edge in the graph. (b) When the robot is done covering the cell between Cp_1 and Cp_2 , it joins the nodes in the graph that correspond to Cp_1 and Cp_2 with an edge, and start two new edges. (c) The robot covers the cells below the obstacle and to the right of Cp_3 . (d) While covering the cell above the obstacle, the robot encounters Cp_2 again. Since all the critical point have explored edges, i.e., covered cells, the robot concludes that it has completely covered the space [21].

**FIGURE 3.18: INCREMENTAL GRAPH CONSTRUCTION [21]**

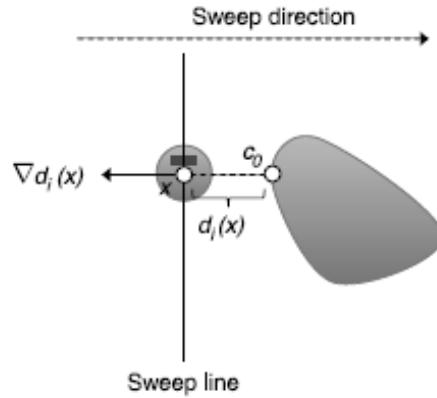
On-line detection of critical points is illustrated by Galceran and Carreras [64]. They tell how a robot detects a surface normal which is the same as a gradient. Given a robot located at point x , let Cp_0 be the closest point to x on the surface of obstacle Cp_i :

$$Cp_0 = \underset{x \in Cp_i}{\operatorname{argmin}} \|x - Cp\|, \quad (3.38)$$

and let $d_i(x)$ be the distance between point x and the obstacle Cp_i . Now, the gradient of $d_i(x)$, $\nabla d_i(x)$ can be calculated as

$$\nabla d_i(x) = \frac{x - Cp_0}{\|x - Cp_0\|}. \quad (3.39)$$

Since a gradient is a unit vector normal to a surface at a given point and since Cp_0 is a point laying on the surface Cp_i , $x - Cp_0$ is a vector pointing outwards, from Cp_0 to x , by dividing it by its norm $\|x - Cp_0\|$ it becomes a unit vector. This leads to the conclusion that a critical point occurs when $\nabla d_i(x)$ is parallel to the sweep direction, as illustrated in figure 3.19.

**FIGURE 3.19: CRITICAL POINT DETECTING [64]**

Points are only detected from the side view. A normal sweeping motion will miss critical point that lay parallel to the sweep direction. This can be counteracted with a Cyclic path. It's important to note that such a path will be longer then a normal zig-zag sweep, since it includes backtracking. A cyclic path starts forward

Cyclic path start by moving the robot in a forward phase, when it hits the boundary, it begins moves downwards. When an obstacles is encounter during its travels it changes its state into a wall following unit, until reaching the next boundary in front of him or detects an critical point. If the later one is the case, it starts moving upward again. If an obstacles detected during this cycle it will follow that wall until a critical point is detected. It then marks it as a new next strip or cell boundary and moves back, towards the point where it ended its initial upward movement. It now starts filling the cell with general

zig-zag sweeps. This incremental construct of the Morse decomposition on-line is stored as Reeb graph. Such a graph has the same functionality as a adjacency graph

The BCD sweep in given in figure 3.18 suggest that the robot will know that it has covered the whole region when it moves from Cp_3 to Cp_2 . But this will require an absolute coördination system that tells the robot that Cp_2 is the same node as the one it encountered when moving from Cp_1 to Cp_2 ; This is error prone because of the accumulated error during dead-reckoning navigation.

MORSE-BASED CELLULAR DECOMPOSITION COMBINED WITH VORONOI-DIAGRAM

The above described on-line Morse-based BCD handles unknown vast environments pretty well. It does so by making use of its sensors. Most work that describe BCD either assume the detector range of the sensor is infinite in size or the same size as the robot. Acar, Choset, and Atkar [18] shows how a detector range of a sensor can be utilized which is $r < \delta_s < \infty$ here r and δ_s .

Morse-based cellular decomposition combined with GVD describe how this can be exploited to find an optimum coverage path for a space which consists of VAST-cell and NARROW-cell. Such that the robot handles both vast and cluttered regions well. See figure 3.20, for such an environment.

The robot has two modus operandi that perform coverage of an unknown space, consisting of vast and cluttered regions. In a vast open space the robot scans an unknown environment for critical points as described in sub section 3.4.1. In such an environment it uses a zig-zag motion with an offset of $2\delta_s$. It's important to note that the coverage of a suction head from a dredge bot will in all likelihood be less than $2\delta_s$. During coverage of this VAST-cell it will construct an adjacency graph from every critical point. Once it encounters a cusp point, it builds a GVD with corresponding nodes. Such a point is an indication of the presence of a NARROW-cell.

This newly placed node on the GVD represents a new NARROW-cell with a width less than the $2\delta_s$. It continues to traverse in the NARROW-cell till it encounters additional cusp point, constructing a GVD. During this stage the sensor can be seen as a sensor with ∞ range. Once it tracks the wall of the second VAST-cell it knows it's in a vast environment, since no second boundary is detected and slips into its initial modus.

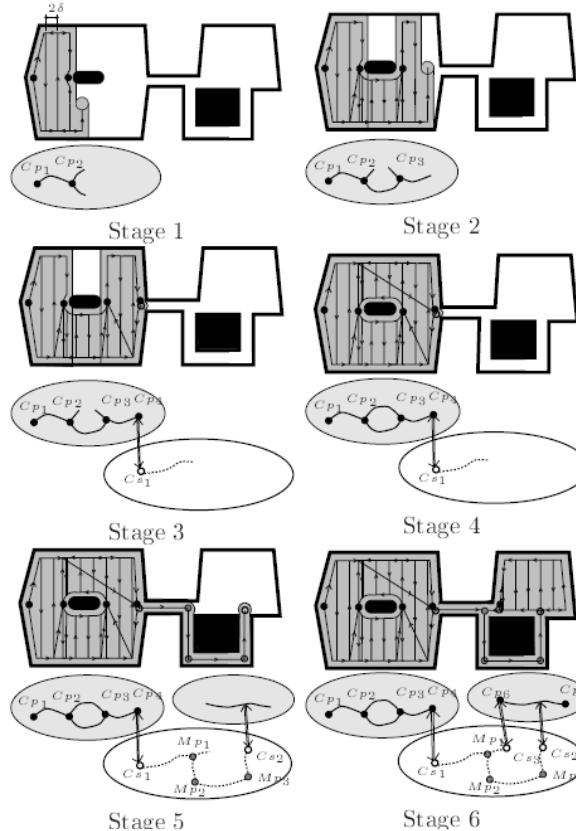


FIGURE 3.20: STAGES OF INCREMENTAL CONSTRUCTION [18]

Figure 3.20 depicts the stages of the incremental construction of the hierarchical decomposition

while the robot is covering the space. The graphs depicted in the gray ellipses depict the VAST-cells that contain VAST-subcells represented as solid edges. Each VAST-subcell has associated critical points represented as black dots. NARROW-cells are represented by the white ellipse and it contains the NARROW-subcells depicted as dashed edges. Hollow dots correspond to cusp points and gray dots represent the meet points. The double arrows show the links between NARROW-cells and their neighboring VAST-cells.

3.4.2 LANDMARK-BASED TOPOLOGICAL COVERAGE

The above described Morse-based algorithms create cell boundaries based on the detection of critical points, these points are detected via side faced range sensors, with the use of wall following. Morse-based algorithms cannot handle rectilinear environments, due to the fact that critical points are degenerate in this environment. Landmark-based topological coverage algorithm also use the BCD, but cell boundaries are determined by using topological landmarks.

Topological maps are robust against sensor and odometry errors because only a global topological consistency, rather than a metric one, needs to be maintained. Thrun [11] states that this type of map does not require accurate determination of the robot's position. Although this low resolution is also the reason why it's difficult to use them for coverage path planning. A node in a topological map is a landmark and does not correspond to a precise position or area in space. This makes it difficult to mark covered regions [34].

SLICE DECOMPOSITION

Slice decomposition makes use of simpler landmarks. Galceran and Carreras [64] states that it can handle a large variety of environments including ones with polygonal, elliptical and rectilinear obstacles. Moreover obstacles can be detected from all sides of the robot, allowing a simpler zig-zag pattern without retracting to be used. As a result the generated coverage path is shorter.

Slice decomposition determines cell boundaries when it sees an abrupt change in the topology between segments in consecutive slices, each slice is a sensor sweep line where the $\delta_s x$ is moved to the next time step. Wong and MacDonald states that there are two situations where the abrupt changes occurs:

1. A segment in the previous slice is split by the emergence of a new segment, see figure 3.21 (a) and (b).
2. A segment from the previous slice disappears in the current slice, see figure 3.21 (c) and (d).

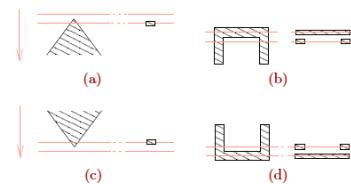


FIGURE 3.21: (a) SPLITTING OF SEGMENTS (b) MERGING OF SEGMENTS [27]

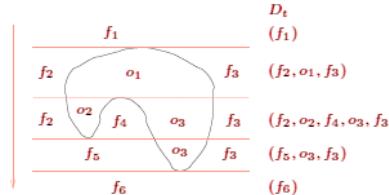
LISTING 3.1: OFF-LINE SLICE DECOMPOSITION

```

1: procedure SliceDecomposition
2:    $c \in \{FreeSpaceCell, ObstacleCell\}$ 
3:   for all time  $t$  do
4:     Move sweep line downwards by  $\Delta x$ 
5:      $D_{l,t-1} = (\dots, c_{i-2}, c_{i-1}, c_i, c_{i+1}, c_{i+2}, \dots)$ 
6:     for all segments in  $D_{l,t-1}$  do
7:       if emergence inside  $c_i$  then
8:          $(c_i) \leftarrow (c_{e-1}, c_e, c_{e+1})$ 
9:          $D_{l,t} = (\dots, c_{i-2}, c_{i-1}, c_{e-1}, c_e, c_{e+1}, c_{i+1}, c_{i+2}, \dots)$ 
10:        end if
11:        if  $c_i$  disappears then
12:           $(c_{i-1}, c_i, c_{i+1}) \leftarrow (c_d)$ 
13:           $D_{l,t} = (\dots, c_{i-2}, c_d, c_{i+2}, \dots)$ 
14:        end if
15:      end for
16:    end for
17:  end procedure

```

The slice decomposition is formed by maintaining a list $D_{l,t}$, which consists of active obstacles and free space cells. This list is created via algorithm 3.1. This algorithm consists of two loops. The first one moves the sweep line, while the second one inspects segments and acts if there is a change in situation. At this time it updates the list $D_{l,t}$ marking it landmark. This algorithm does not take into account "line of sight". The author of this paper states that disappearance of segment can only be measured from hindsight. Thus backtracking will still be an issue.

**FIGURE 3.22: SLICE DECOMPOSITION GENERATED BY ALGORITHM 3.1 [27]**

Wong [34] recognizes the limitations of slice decomposition and proposes a new method "slice decomposition II". This is because a robot cannot move inside obstacles, which means that the sweep line is limited to the cell that the robot is in [34]. There are five events that occur during slice decomposition II, as are depicted in figure 3.23. Wong proposes the following events to be used during the on-line algorithm 3.2. If the robot is tethered, for instance with an umbilical, not every cell can be reached. The restrictions created by this tether can be viewed as a change of the boundary of the environment.

ACTION	DESCRIPTION
SPLIT	Free space segment in the previous slice is split into two by the emergence of an obstacle. This is equivalent to obstacle segment emergence in normal Slice Decomposition.
MERGE	Free space segment in the current slice neighbors free spaces other than the free space segment in the previous slice in the direction of the previous slice. This is equivalent to obstacle segment disappearance in normal slice decomposition.
END	The previous free space segment is the final one in the current cell. This is equivalent to free space segment disappearance in the normal version.

LENGTHEN	Free space segment in the current slice neighbors an obstacle segment in addition to the free space segment in the previous slice in the direction of the previous slice. Another way to view this situation is that the current slice is much longer than the previous slice.
SHORTEN	Free space segment in the previous slice neighbors and obstacle segment in addition to the free space segment in the current slice in the direction of the current slice. Another way to view this situation is that the current slice is much shorter than the previous slice.

LISTING 3.2: OFF-LINE SLICE DECOMPOSITION II

```

1: procedure SliceDecompositionII
2:    $O \leftarrow$  initial cell
3:    $F \leftarrow \emptyset$ 
4:   while  $O \neq \emptyset$  do
5:      $f_c \leftarrow f \in O$ 
6:     move to one (of two) cell boundary of  $f_c$ 
7:     repeat
8:       move sweep line by  $\Delta x$  towards the opposite cell boundary
9:       if event occur then
10:         $F \leftarrow F + f_c$ 
11:         $O \leftarrow O - f_c$ 
12:        if event = split or merge then
13:           $O \leftarrow O + f_{c+1}, f_{c+2}$  if  $f_{c+1}, f_{c+2} \notin (O \cup F)$ 
14:        end if
15:        if event = lengthen or shorten then
16:           $O \leftarrow O + f_{c+1}$  if  $f_{c+1} \notin (O \cup F)$ 
17:        end if
18:      end if
19:      until event occur
20:    end while
21: end procedure

```

ON-LINE TOPOLOGICAL COVERAGE ALGORITHM

By using slice decomposition II, which is described in Section 3.4.2, with a topological map. A crawler can construct it on-line. It now allows it to perform its task in an unknown environment. The topological map embeds the slice decomposition of an environment by using events as nodes. The map is updated whenever relevant information becomes available. The path planner generates a new path on each update, based on the new partial topological map [34].

This topological map is represented as a planar graph, where the nodes represent landmarks (i.e., split, merge, end, lengthen or shorten, such as depicted in figure 3.23) and edges indicate the types of motion required to travel between nodes they are incident upon. For example, whether the edge is next to a wall and which side the wall is on. They also store estimated distances separating the two nodes they connect [59].

This topological coverage algorithm makes use of a state transition diagram 3.24. In this diagram three states are described, each with corresponding algorithm (3.3, 3.4 and 3.5). The crawler is assumed to be placed in a corner near a wall, therefore the boundary state is considered as entry point.

The crawler starts in the *boundary* state by executing algorithm 3.3. During this state the crawler performs a wall following algorithm. When it finds a landmark or it arrives at the end of a strip, it updates graph G. When it's at the end of a strip and the boundary is fully explored it gets into the state travel, described in algorithm 3.5, otherwise it turns around and continues in the boundary state.

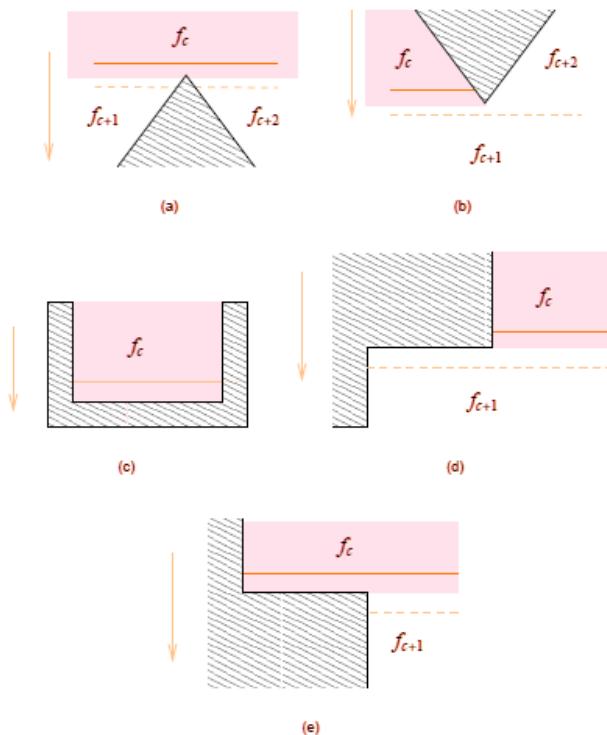


FIGURE 3.23: (a) SPLIT, (b) MERGE, (c) END, (d) LENGTHEN, (e) SHORTEN

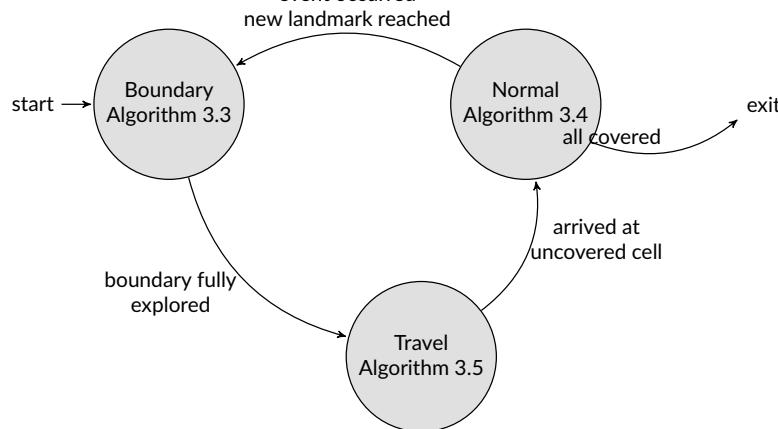


FIGURE 3.24: STATE TRANSITIONS FOR TOPOLOGICAL COVERAGE ALGORITHM

LISTING 3.3: BOUNDARY STATE

```

1: procedure BoundaryState
2:   loop
3:     move forward along boundary
4:     if at landmark then
5:       update G
6:     end if
7:     if arrive at end of strip then
8:       update G
9:     if boundary fully explored then
10:      state <= travel
11:    else
12:      turn around 180°
13:    end if
14:  end if
15: end loop
16: end procedure
  
```

Once in the *travel* state, the path is generated that moves the crawler from one cell to another, it does so by implementing line 5 and 6 from algorithm 3.2 described at page 30. Algorithm 3.5 is executed in this state. Once it arrives at an cell its operation state becomes *normal*. A normal boustrophedon zig-zag movement is followed in this state.

LISTING 3.4: NORMAL STATE

```

1: procedure NormalState
2:   repeat
3:     follow zigzag pattern
4:   until at landmark
5:   update G
6:   state  $\leftarrow$  boundary
7: end procedure
```

LISTING 3.5: TRAVEL STATE

```

1: procedure TravelState
2:    $T(n) \leftarrow$  search  $G$ 
3:   if  $T(n) = \emptyset$  then
4:     exit algorithm
5:   end if
6:   while  $T(n) \neq \emptyset$  do
7:     move towards  $T(0)$ 
8:     if at  $T(0)$  then  $T(n) \leftarrow T(n) - T(0)$ 
9:     end if
10:   end while
11:   state  $\leftarrow$  normal
12: end procedure
```

LANDMARK RECOGNITION USING NEURAL NETWORKS

Wong [34] proposes a novel idea to classify landmarks using Neural Networks. These are classification algorithms which approximates the operations of a human brain. Wong build a test robot with a 360° rotatable single-beam sonar. Each scan consists of 48 individual sonar-beams taken over a range of 360°. This vector is made independent of orientation, by virtually rotating it so that index 0 would always point towards the direction where the sonar range measured the shortest distance, as depicted in figure 3.25.

This vector is fed into a MLP which distinguishes three different type of classes: free space nodes, obstacle nodes and everything else. This neural network first has to be taught. This is done under supervision, meaning that the landmark type have to be predefined.

3.4.3 GRID-BASED METHODS

Grid-based methods divide the working area into a raster of uniform grid cells. Each cell has an associated value stating whether an obstacles is present or if it's rather free space. The value can either be binary or a probability [71]. These grid cells are typically square in shape, but it's not uncommon to use triangle shaped cell. The size of the cell usually corresponds with the size of a crawler.

Once the environment is mapped onto an uniform grid. An optimal coverage path can be found using the by Choset and Pignon [9] proposed method, whom uses a conventional wave-front algorithm

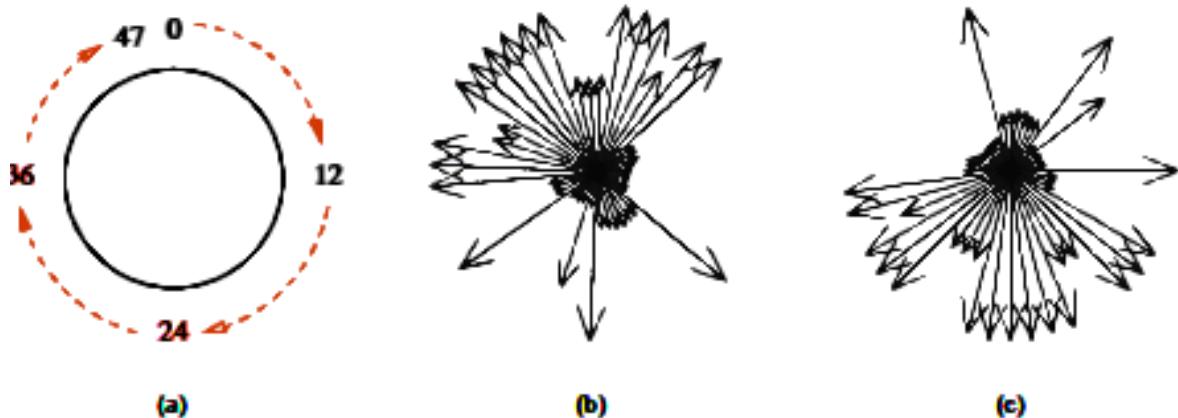


FIGURE 3.25: ROTATION OF SONAR READING TO MOST OCCUPIED DIRECTION [34]

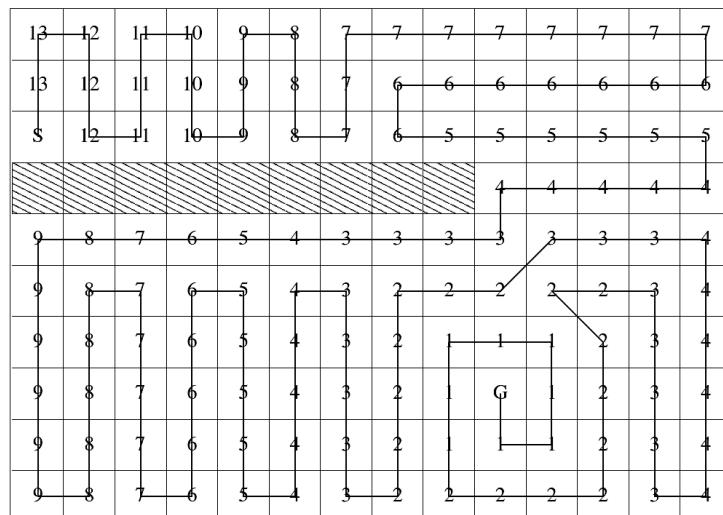


FIGURE 3.26: COVERAGE PATH GENERATED FROM A DISTANCE TRANSFORM [34]

(distance transform) to determine a coverage path. First a start and goal cell has to be assigned. The wave-front algorithm initially assigns a 0 to the goal and then a 1 to all surrounding [red, reachable] cells. Then all the unmarked cells neighboring the marked 1 are then labeled with a 2. This process repeats until the wave-front crosses the start. Once this occurs, the robot can use gradient descent on this numeric potential function to find a path [19]. This results of algorithm 3.6 is shown in figure 3.26.

LISTING 3.6: OFF-LINE GRID-BASED COMPLETE COVERAGE

```

1: procedure GridBasedCompleteCoverage
2:   Set start cell to current cell
3:   Set all cells to NOT visited
4:   loop
5:     Find unvisited neighboring cell with highest value
6:     if NO unvisited neighboring cell found then
7:       Mark current cell as visited
8:       Exit procedure                                ▷ Goal reached
9:     end if
10:    if unvisited neighboring cell value ≤ current cell value then
11:      Mark current cell as visited
12:      Exit procedure                                ▷ Goal reached
13:    end if
14:    Set current cell to neighboring cell          ▷ Move to next cell
15:  end loop
16: end procedure

```

The author states an optimal placement of the start and goal cell is paramount. The algorithm 3.6 does not take into account a deadlock state. Which is illustrated in figure 3.27. In this figure goal and start cell are arbitrarily placed. Execution of the algorithm ensures that the dredge bot is stuck at the farthest cell. Where it remains, if no backtracking over previous visited cells is allowed.

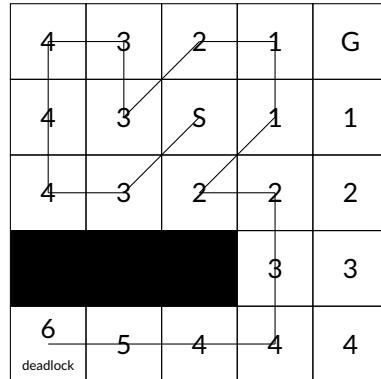


FIGURE 3.27: DEADLOCK STATE DUE TO INCORRECT STARTING POINT

GRID-BASED COVERAGE USING SPANNING TREES

An sub-family of grid-based coverage are systematic spiral paths algorithms. These work by following a systematic spiral spanning tree of a partial grid map. This map is constructed using its on-board sensors [71] and uses two different sizes of grid cells. The smaller grid cell is the same size as the robot. Four of these grid cells then form a mega cell. [34]. The Spiral-STC described in algorithm 3.7, works as follows:

Starting at the current cell, the robot chooses a new travel direction by selecting the first new mega cell in the free space in anti-clockwise direction. Then, a new spanning-tree edge is grown from the current mega cell to the new one. The algorithm is called recursively. The recursion stops only when the current cell has no new neighbors. A mega cell is considered old if at least one of its four smaller cells is covered. As a result of this recursion, the robot moves along one side of the spanning tree until it reaches the end of the tree. At this point, the robot turns around to traverse the other side of the tree [71].

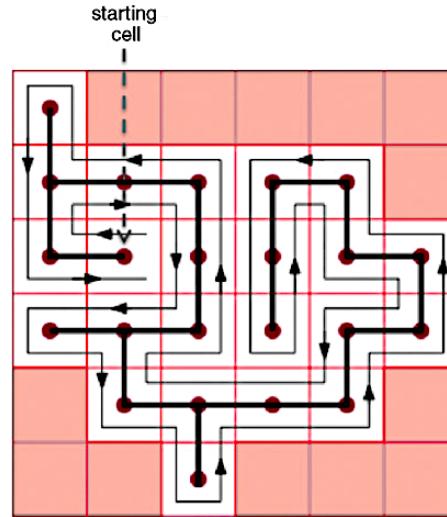


FIGURE 3.28: COVERAGE PATH GENERATED WITH THE SPIRAL-STC ALGORITHM [64].

LISTING 3.7: SPIRAL SPANNING TREE COVERAGE

```

1: procedure SpiralSpanningTreeCoverage( w, x )
2:   Mark the current cell x as old
3:   while x has new obstacle-free-4-neighbour cell do
4:     Scan for the first new neighbour of x in anti-clockwise order, starting with the
      parent cell w Call this neighbour y
5:     Construct a spanning-tree edge from x to y.
6:     Move to a subcell of y by following the right-side of the spanning tree edges
7:     Execute SpiralSpanningTreeCoverage(x, y).
8:   end while
9:   if x ≠ startcell then
10:    Move back from x to a subcell of w along the right-side of the spanning tree edges.
11:   end if
12: end procedure

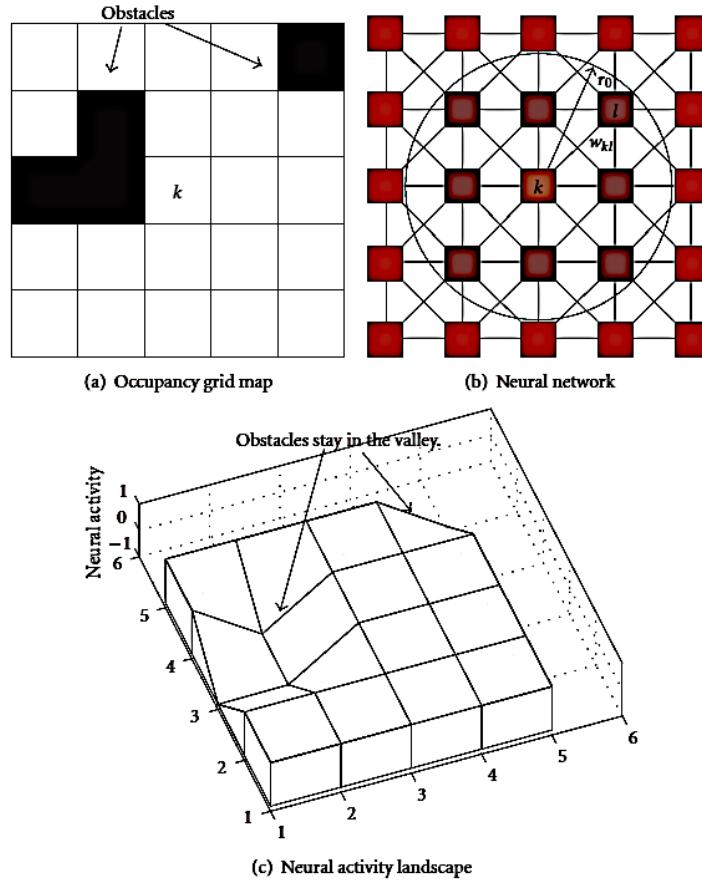
```

Wong [34] and Lee, Baek, Choi, et al. [53] states the path can be optimized by using smaller grid cells, however because accurate maneuverability is often an issue, a higher resolution is not the best approached. Lee, Baek, Choi, et al. proposes a new method for Spiral-STC by limiting the number of turns, decelerations and accelerations. Mei, Lu, Hu, et al. [26] determined by analytically comparing energy efficiency of different coverage algorithm that sharp turns bring about inefficiency. Thus by limiting these, an energy efficient path can be generated. Since the dredge bot will be powered by an external land-based source, this option won't be further explored.

NEURAL NETWORK-BASED COVERAGE ON GRID MAPS

Luo, Yang, Stacey, et al. [23] proposes a model capable of planning a real-time path; Which covers every area, in the vicinity of obstacles, to a reasonably extent. A crawlers path is autonomously generated through a dynamic neural activity landscape [23][42]. Luo, Yang, Stacey, et al. discretized a 2D space in a grid map where the diagonal length of each grid cell is equal to the robot sweeping radius and then a neuron is associated to each and every grid cell. Each neuron has connections to its immediate 8 neighbors [64]. This architecture is illustrated in figure 3.29.

The proposed neural network, is topologically expressed on a 2-dimensional occupancy grid map. The location of the *k*th neuron of the neural network represent a location (cell). Where each neuron has local lateral connections to its neighboring neurons, in the small region $[0, r_0]$; Where r_0 is the

**FIGURE 3.29: ARCHITECTURE OF A NEURODYNAMICS MODEL [62]**

receptive field radius of the k th neuron, as shown in figure 3.29 (b). The excitatory input results from uncovered area and lateral neural connections, while inhibitory inputs, results from obstacles [62]. The shunting equation 3.40 derived from Hodgkin and Huxley [1] determines the dynamics of each neuron in the network.

$$\frac{dx_k}{dt} = -Ax_k + (B - x_k) \left([I_k]^+ + \sum_{l=1}^m w_{kl} [n_{a,l}]^+ \right) - (D + x_i) [I_i]^- \quad (3.40)$$

Equation 3.40 consist of the following terms: x_k is the k th neuron in the neural network, while A, B and D are non negative constants describing the passive decay rate, and the upper and lower bounds of the neural activity. The terms $[I_k]^+ + \sum_{l=1}^m w_{kl} [n_{a,l}]^+$ are the excitatory inputs while $[I_i]^-$ is the inhibitor. These are linear-above and below thresholds defined as $[a]^+ = \max\{a, 0\}$ and $[a]^- = \max\{-a, 0\}$. The connection weight is given by w_{kl} which is assigned between the k th and the l th neuron, which is given by $w_{kl} = f(|q_k - q_l|)$, where $|q_k - q_l|$ is the Euclidean distance between vectors q_k and q_l in the state space, and $f(d)$ is a monotonically decreasing function defined as

$$f(d) = \begin{cases} \frac{\mu}{d}, & 0 \leq d < r_0 \\ 0, & d \geq r_0 \end{cases} \quad (3.41)$$

Where μ and r_0 are positive constants, The external input I_k to the k th neuron is defined in equation 3.42. In this equation E is a large constant.

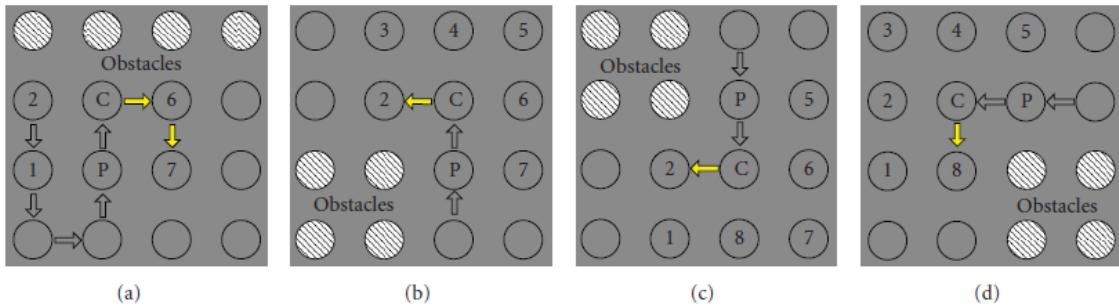


FIGURE 3.30: Four predefined templates. Source: Yan, Zhu, and Yang [62]

$$I_k = \begin{cases} E, & \text{if it's an uncovered area} \\ -E, & \text{if it's an obstacle area} \\ 0, & \text{if it's a covered area} \end{cases} \quad (3.42)$$

By properly defining the external inputs from the changing environment and internal neural connections, the unclean areas and obstacles are guaranteed to stay at the peak and the valley of the activity landscape of the neural network, respectively. The unclean areas globally attract the robot in the whole workspace through neural activity propagation, while the obstacles have only local effect in a small region to avoid collisions. The collision-free robot motion is planned in real time based on the dynamic activity landscape of the neural network and the previous robot position, such that all areas will be cleaned and the robot will travel along a smooth zigzag path [42]. An advantage of this method is that can handle non stationary environments (i.e., dynamically changing obstacles) [64].

Yan, Zhu, and Yang further states that if there are power and time limitations, the crawler should travel the path with the least revisited areas and the least turns of moving directions. Therefore, for a given current crawler location at time k , denoted by \vec{p}_k , the next crawler location at time $k+1$, \vec{p}_{k+1} is obtained by equation 3.43. Where c is a positive constant and m_n is the number of neighbouring neurons of the \vec{p}_k neuron, that is all the possible locations of the current location \vec{p}_k ; variable $n_{a,l}$ is the neural activity of the l th neuron, which is the same as in equation 3.40; y_l is a monotonically increasing function of the difference between the next crawler moving directions, defined in equation 3.44.

$$\vec{p}_{k+1} \leftarrow x_{\vec{p}_k} = \max\{n_{a,l} + c y_l, l = 1, 2, \dots, m\} \quad (3.43)$$

Where $\Delta\Psi_l \in [0, \pi]$ is the turning angle between the current moving direction and the next moving direction. Here $\Delta\Psi_l = 0$ is straight ahead, while $\Delta\Psi_l = \pi$ stands for a backwards movement [62]. This would indicated that the crawler only has the ability to turn either left or right depending on the convention place on $\Delta\Psi_l$. By defining it as $\Delta\Psi_l \in [-\pi, \pi]$, the crawler has a full turning range. Thus $\Delta\Psi_l = \Psi_l - \Psi_c = |a \tan 2(y_{p_l} - y_{p_c}, x_{p_l} - x_{p_c}) - a \tan 2(y_{p_c} - y_{p_l}, x_{p_c} - x_{p_l})|$

$$y_l = 1 - \frac{\Delta\Psi_l}{\pi} \quad (3.44)$$

According to Yan, Zhu, and Yang [62] path planning in the vicinity of some unstructured obstacles may cause overlap, resulting in additional turns, which translates in a higher energy consumption and additional time [53][76][26]. They propose the use of four predefined tables as shown in figure 3.30. These predefined templates are effective to deal with the vicinity situation of unstructured obstacles and enable the crawler to plan a more reasonable path with less overlapping areas.

Grid based neural networks methods lend themselves to be easily integrated with map building. Yan, Zhu, and Yang [62] describes how the sensor model can be used to build an on-line map of the environment where the Dempster Shafer theory is used to filter out uncertainties of the sensor.

CHAPTER CONTROLLER DESIGN 4

A controller that is capable of making complex decisions in a changing environment, requires different levels of abstraction. At the highest level it should be able to form a plan based upon a predefined strategy. A dredging crawler should be able cover a certain area removing a specified amount of soil, until reaching the specified depth. Because the geometry of the environment in which it operates, is unknown from the beginning, it should be able to adjust and change its operating parameters accordingly. Section 4.1 will assess the researched CPP algorithms and decide which one is best suited for the crawler. In order to adopt and change its behavior to best suit the surroundings surrounding it needs to be aware of it and have means of manipulation at its disposal. Section 4.2 will discuss these required peripherals determine the required types.

Discussion on an accurate model for a Kalman filter can takes place in Section 4.3. This also covers a discussion on the state representation for a crawler, the motion model and crawler specific models, such as soil dynamics, dredging and steering. Section 4.4 describes how a controller framework for autonomous maritime vessels should behave, discussing the design patterns.

4.1 STRATEGY DECISION

Researching different CPP algorithms in Section 3.3 showed that a divide can be seen between on-line and off-line algorithms. The latter of these are deemed unsuitable, due to a changing environment. This reduces viable candidate algorithms to the following short-list:

- onBCD
- BCD-GVD
- topBCD
- onSD2
- onSD2NN
- STC
- NN

The options presented above are judged against different criteria of assorted relevance, refelected in a weight factor. Ranging from 1 nice to have, to 3 really fucking useful. Each algorithm is awarded points for each criteria, which range between 1 and 5, the higher the better it fulfils that criteria. Quantifying the axact performance is impossible at this stage. This introduces a level of uncertainty, which is taken into account in the decision. The total sum for each algorithm is then calculated with the propagation of those uncertainties. The table below shows that all three BCD algorithm easily outweigh the others for our application. Were BCD-GVD, has the highest score but it brings with it a greater uncertainty compared against topBCD. It's good practice in agile developement, to "make it work first" and "refine it later". By choosing topBCD as the best strategy decision the implied risk due to uncertainties is reduced.

	weight	onBCD	BCD-GVD	topBCD	onSD2	onSD2NN	STC	sSTC	NN
online	3	5 ± 0	5 ± 0	5 ± 0	5 ± 0	5 ± 0	5 ± 0	5 ± 0	5 ± 0
time	2	5 ± 2	5 ± 2	5 ± 1	5 ± 2.5	1 ± 2	3 ± 2	3 ± 2	2 ± 1
energy eff.	1	4 ± 1	4 ± 1	4 ± 0.5	5 ± 1	1 ± 0.5	3 ± 1	3 ± 1	2 ± 2
revisits	3	1 ± 0.2	1 ± 0.2	1 ± 0.2	1 ± 0.2	1 ± 0.2	1 ± 0.2	1 ± 0.2	1 ± 0.2
complexity	3	5 ± 1	5 ± 1	4 ± 0.5	1 ± 1	1 ± 2	1 ± 1	1 ± 2	2 ± 0.5
adaptivity	1	2 ± 1	3 ± 1	3 ± 1	5 ± 0.5	5 ± 0.5	3 ± 2	3 ± 2	4 ± 1

TRL	3	5 ± 1	5 ± 1	5 ± 0.5	2 ± 2	2 ± 2	3 ± 1	3 ± 1	1 ± 3
maintainability	2	5 ± 2	5 ± 2	4 ± 1	3 ± 1	3 ± 1	2 ± 1	2 ± 1	2 ± 2
integration	1	4 ± 0.5	4 ± 0.5	4 ± 0.5	2 ± 1	2 ± 1	4 ± 2	4 ± 2	4 ± 2
legislation	1	3 ± 2	3 ± 2	3 ± 2	2 ± 3	2 ± 3	3 ± 2	3 ± 2	2 ± 2
environment	1	3 ± 1	3 ± 1	3 ± 1	2 ± 1	2 ± 1	3 ± 1	3 ± 1	2 ± 1
Σ		84.0	85.0	80.0	59.0	47.0	56.0	56.0	49.0
		+7.6	+7.6	+4.4	+9.3	+10.2	+7.2	+8.9	+10.8

4.2 PERIPHERALS

The crawler is comprised of multiple components which are listed in Appendix A. This section will focus on components which are required for autonomous operations. The two drivetrains consists of an electric motor, with an attached gearbox, coupled to a hydraulic motor and a hydraulic system which drives a hydraulic pump. The shaft of the hydraulic motor is attached to a flexible coupling and connected with an Archimedes screw by a rigid flange connection. This drivetrain is discussed in detail in Section 4.3.2. The crawler is able to turn if both drivetrains operate at different velocities, which is discussed in Section 4.3.5. The control vector \vec{u}_k , consist of two velocity components.

Dead-reckoning is often accomplished with a accelerometer, gyroscope and magnetometer, this sensor array is often expanded with a pressure sensor. These sensors combined will give a good impression of the orientation and position change over time. Keeping in mind that the measurement uncertainty of the pressure sensor will increase during dredging operations. This phenomena is described in Section 3.2.1. Knowing actual rotational velocity of the Archimedes screw will be used to better predict the traveled distance. Something that can be measure with an encoder attached on the shaft of the electro motor. These encoder are also used for the regular PID controller of the screws. Although we can measure the rotational velocity of the screws we can't know for sure how fast the crawler will actually move. Section 4.3.3 describes how the change in required torque of the drivetrain will give a indication of the slippage of the Archimedes screws.

4.3 KALMAN FILTER DESIGN

In Chapter 3.4 it became evident that an accurate estimation of a crawlers position and heading is needed such that it can perform its tasks using CPP algorithms. Because it's not possible to use GPS in an underwater environment, due to the dampening of electric and magnetic fields in water, which was described in Section 3.1.2, an alternative localization method had to be found. Chapter 3.3 describes Kalman filter as the industry de-facto. Position estimation for the control will therefore be performed with this algorithm.

A design for an Kalman filter is proposed, using a common array of sensors, such as: accelerometer, gyroscope, magnetometer and a pressure sensor. The workings for each of these are discussed Section 3.2 and the component selection was made in Section 4.1. The crawler is actuated by, changing the rotational speed of the individual Archimedes screws.

Where the speed of an Archimedes screw driven crawler is a direct function of the pitch of its vanes. But this only applies if there is no horizontal soil failure under the screws. Such a phenomena is called slip. It consist of a period in time where the screws turn, but generate no forwarding force. Leading to an inaccurate estimation for the new predicted state. Due to the geometry of an Archimedes screw, this slip coexist with an bulldozer effect created by the vanes acting as a shovel in the soil. This will lead to an increase in torque. It's proposed that by measuring the required torque of the drive train, a prediction can be made how much slip has occurred, leading to a better estimation of the future state-vector \vec{x}_{k+1} . This behavior and the mathematical model will be discussed in more detail in Section 4.3.2. Figure 4.1 shows the interaction and connectivity of the various components that server as the input and output of the proposed filter.

Since the physical processes for movement of a crawler are non-linear and the "basic" Kalman filter, described in Section 3.3.2 is limited to linear assumptions, the proposed Kalman filter is of the Unscented variant. The UKF uses a deterministic sampling technique known as UT. This technique picks a minimal set of sample points, also known as sigma points, around the mean. The sigma points are then

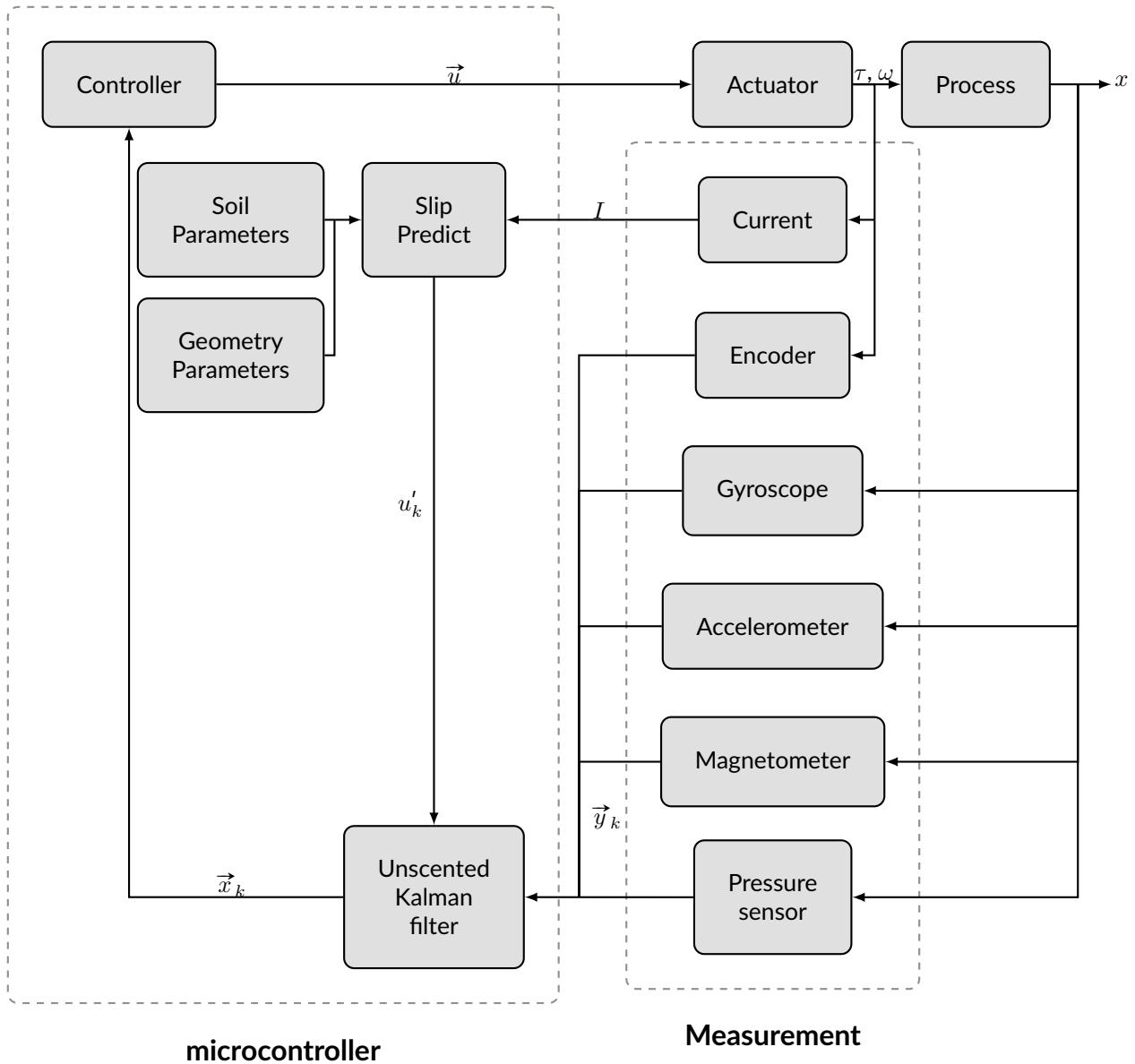


FIGURE 4.1: PROPOSED KALMAN FILTER

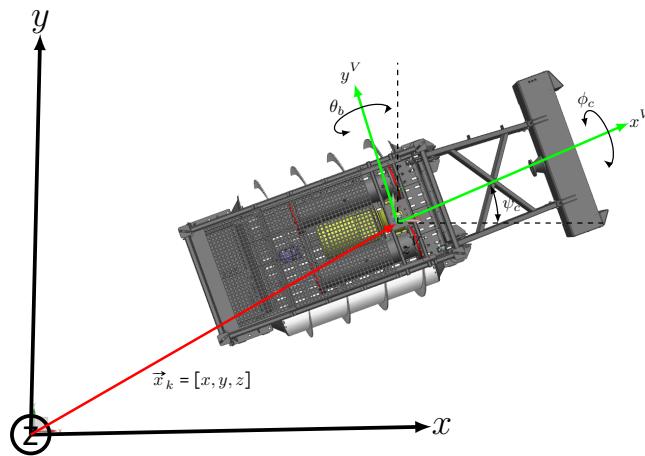
propagated through the non-linear function, from which a new mean and covariance estimate are then formed.

4.3.1 STATE REPRESENTATION

Bahr, Leonard, and Fallon [44] states that, the most generic case of a vehicle operating in 3D Euler-space, such as a crawler, consist of a vector of variables comprised of a vehicle's pose and orientation. The pose is its position in a (global) reference frame $[x \ y \ z]^T$. While its orientation is given in Euler angles $[\phi_c \ \psi_c \ \theta_c]^T$. The pose vector at time t is then $\vec{x}_k = [x \ y \ z \ \phi_c \ \psi_c \ \theta_c]^T$, which will be denoted as \vec{x}_k for the remainder of this paper. Beside the pose, the state vector can also contain the first and second derivatives of the pose vector.

2D OR 3D

Bahr, Leonard, and Fallon [44] proposes the following simplification; They state that all submersible vehicles are outfitted with a pressure sensor which allows them to determine their absolute depth with high accuracy and a high update rate. As a result all underwater navigation systems are only used to

**FIGURE 4.2: STATE REPRESENTATION**

resolve the 2D position and all underwater vehicle related localization problems are stated in 2D. Which allows for a simplified state vector $\vec{x}_k = [x \ y \ \phi_c \ \psi_c]^T$.

This simplification does not hold for crawlers or AUV that change in depth in any other direction than collinear with the earth's gravitational axis. Since this crawler moves over irregular terrain, any pitch or roll will result in a gravitational force that also consists of components along the x-axis and / or y-axis. Which would then be interpreted as movements in the 2D x-y space. This coupled with an unreliable pressure-sensor reading due to disturbance of sediment from the sea bed during dredging operations, as described in Section 3.2.1, means the state representation should be in 3D Euler space, as is shown in figure 4.2.

QUATERNIONS

Because the UKF is executed on an embedded device, with limited resources, all rotational translations are calculated with the help of quaternions. Which make use of the same concept as complex numbers, except of using one imaginary axis, there are three. Just as complex numbers are very useful in describing a rotation in a two-dimensional plane, quaternions are highly efficient in a three-dimensional space. They are also immune to gimbal locking. The exact workings are outside of the scope of this thesis, but for those readers which are interested, a good explanation is given by 3Blue1Brown [88] and can be found at www.youtube.com/watch?v=d4EggbgTm0Bg. A quaternion is represented as a single column matrix with four rows, consisting of a real valued magnitude and three imaginary components for the axes $\vec{q} = [q_s \ q_x \ q_y \ q_z]^T$, Where the q_s is used to normalize the quaternion, this keeps the error from accumulating. It's important to note that quaternions are non-commutative, thus the order in which they are applied matters to the outcome of the rotation.

4.3.2 MOTION MODEL

It's important to evaluate the effects of control inputs \vec{u}_k on the pose vector $\vec{p}_k = [x, y, z]^T$ such that \vec{p}_{k+1} can be predicted. Since the continuous-time model for the vehicle state's speed and rate can be described as:

$$\vec{p}_{k+1} = f(\vec{x}_k, \vec{u}_k) \quad (4.1)$$

The function $f(\cdot)$ in equation 4.1 can be very complex and is usually non-linear, it has to take into account all external forces that can influence the movement of the crawler. According to Bahr, Leonard, and Fallon [44] the more complex the model, the more accurately it can represent the vehicle dynamics and provide a better prediction of the future pose, but obtaining such a model requires detailed knowledge of the structure as well as the parameters listed below. All of which influence movement and sensor reading:

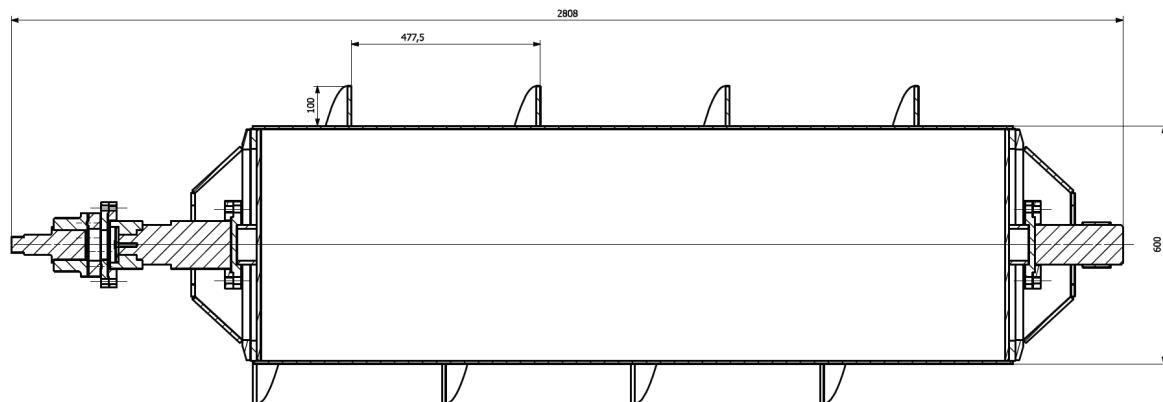
- Shape of the crawler¹²
- Size of the crawler¹²
- Weight of the crawler¹²
- Buoyancy of the crawler¹²
- Actuators (pumps, motors etc.) and their behavior, such as vibrations
- Forces exerted on the crawler due to fluid transportation through the floating line
- Forces exerted on the crawler due to the buoyancy of the floating line
- Forces exerted on the crawler from the umbilical
- Operating environment
 - Properties of the soil bed [46];²
 - * Density²
 - * Cohesion²
 - * Skin friction²
 - Temperature
 - Salinity
 - Current
 - Medium through which maneuver (mixture of water and soil)
- Configuration of the propulsion system:
 - Track variant² [46]
 - * Length¹²
 - * Distance between grouser plates¹²
 - * Width of the track¹²
 - * Height of the grouser plate¹²
 - Archimedes screw variant² [47]
 - * Length¹²
 - * Diameter¹²
 - * SWR¹²
 - * Number of helices¹²
 - * Pitch angle¹²
 - * Vane height¹²
 - * Front bulldozer angel¹²
 - * slip²

Not all parameters shown above, have the same impact on the future state representation. Since calculations on a complex model require more processing power, only parameters with a substantial influence will be used. It's assumed that the biggest influences are drag-forces, due to movement in a fluidium and the interaction with the Archimedes screw in the silt layer. These will in all likelihood have a bigger impact than other operating parameters. Forces exerted on the crawler due to a current, floating lines, and actuators are – for now – deemed to be a magnitude smaller and can be ignored.

During travel between two points, it's preferred to minimize travel time. Which is usually achieved by moving at the fastest sustainable speed. The propulsion system which acts on the submerged soil bed is governed by a lot of interactions on the system as a whole. Which were listed in Section 4.3.2. The crawler is propelled by an Archimedes screw. A proven technology for land based vehicles. Especially in rough, impassable terrain. They have shown reliable operations, superior traction capabilities and they can be used as a buoyancy body. Typical working territories for Archimedes driven are dredge deposit sites and swamps Lotman and Grima [54].

¹ Constant parameter

² Used in the model

**FIGURE 4.3: ARCHIMEDES DRIVE TRAIN**

PROPELLION SYSTEM MODEL

The propulsion system is modeled and shown in Figure 4.4. In this situation, an electrical motor, drives a hydraulic pump. This electrical motor delivers a certain torque τ , where the hydraulic pump demands a certain torque. This torque difference $\Delta\tau$ divided by the sum of inertia's and integrated over time, translates to a certain rotational speed ω , with which this system turns. This hydraulic pump generates a pressure p , which delivers energy to a hydraulic motor, which has a pressure drop, or in other words converts this energy in a certain torque at a certain rotational speed. The pressure difference Δp , between the hydraulic pump and motor, multiplied with the cross section, in which the hydraulic oil flows, and divided with the total mass of hydraulic oil, integrated over time, results in a fluid velocity. This hydraulic motor drives the Archimedes screws, which interact with the soil and generate a force which, when total friction and resistance is overcome translate to a velocity.

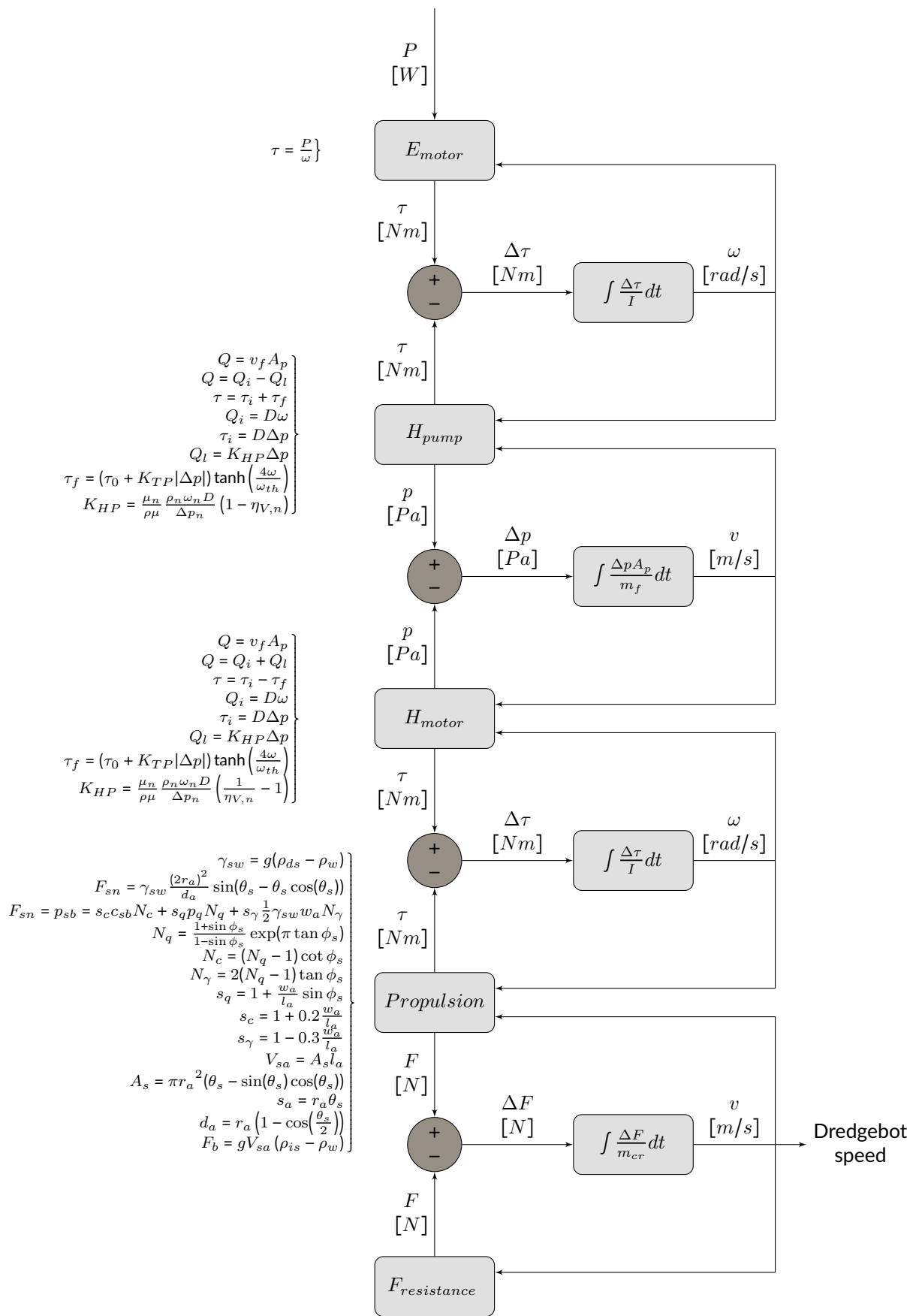


FIGURE 4.4: DRIVE TRAIN BASED ON FIRST PRINCIPLE

A HYDRAULIC PUMP is used to convert mechanical- to hydraulic energy is modeled after the, by Mathworks [82], described model. In order to make a model based on first principle, the hydraulic pump, should accept an variable fluid speed $v_f(\cdot)$ and angular velocity of the electro motor ω , whilst calculating a pressure gain Δp and needed torque τ . The flow rate which is generated at the pump is equal to equation 4.2. Where A_p is the cross-section of the pipe and v_f is the average speed of the fluid, through that cross-section. Where Q is the net volumetric flow rate.

$$Q(v_f) = A_p v_f \quad (4.2)$$

The net volumetric flow rate obtained from equation 4.2 consists of an ideal flow Q_i where a leakage flow Q_l is subtracted, as is shown in equation 4.3.

$$Q(v_f, \omega) = Q_i(\omega) - Q_l \quad (4.3)$$

The ideal flow rate, needed by equation 4.3, is generated by a displaced volume D times the rotational speed ω . Which is shown in equation 4.4.

$$Q_i(\omega) = D\omega \quad (4.4)$$

Where the leakage flow rate compares to the Hagen-Poiseuille coefficient for laminar pipe flows K_{HP} , which is computed for nominal parameters and multiplied with the pressure gain Δp .

$$Q_l = K_{HP} \Delta p \quad (4.5)$$

In order to determine the pressure gain as a function of fluid speed and angular velocity, equations 4.2, 4.3, 4.4 and 4.5 can be combined and rewritten in to equation 4.6.

$$\Delta p(v_f, \omega) = \frac{D\omega - A_p v_f}{K_{HP}} \quad (4.6)$$

The Hagen-Poiseuille coefficient, needed in equation 4.5 and 4.6, is calculated with the nominal viscosity μ_n , nominal density ρ_n and the displacement volume D . Divided by the actual density ρ and viscosity μ .

$$K_{HP} = \frac{\mu_n \rho_n \omega_n D}{\rho \mu} (1 - \eta_{V,n}) \quad (4.7)$$

In order for the pump to generate a flow a driving torque is required. The needed driving torque τ consists of an ideal driving torque τ_i and a resistance, which is to be overcome, due to friction τ_f .

$$\tau(v_f, \omega) = \tau_i(v_f, \omega) + \tau_f(v_f, \omega) \quad (4.8)$$

While the ideal driving torque τ_i is also a function the displaced volume D times the pressure gain from

inlet to outlet Δp , as is shown in equation 4.9.

$$\tau_i(v_f, \omega) = D\Delta p(v_f, \omega) \quad (4.9)$$

The friction generated by the torque τ_f is calculated according to equation 4.10. In this equation, τ_0 represent the no-load torque parameter and ω_{th} is the threshold angular velocity for the pump-motor transition. The threshold angular velocity is an internally set fraction of the Nominal shaft angular velocity parameter. The Friction torque vs pressure gain coefficient parameter K_{TP} .

$$\tau_f(v_f, \omega) = (\tau_0 + K_{TP}|\Delta p(v_f, \omega)|) \tanh\left(\frac{4\omega}{\omega_{th}}\right) \quad (4.10)$$

A HYDRAULIC MOTOR is used to convert hydraulic energy into mechanical energy. This actuator is modeled after a Mathworks [82], described model. It receives feedback from the fluid velocity v_f and the angular velocity ω of the propulsion system. It generates torque τ by converting pressure p . The workings of a hydraulic pump and motor share much similarities, with some notable differences related to the leakage flow. Were a pump subtracts the leakage flow it's added in this model. Since the efficiency is an inverse of the pump.

$$Q = Q_i + Q_l \quad (4.11)$$

In order to calculate a pressure drop as an function of fluid and angular velocity over the outlets, equations 4.2, 4.11, 4.4 and 4.5 can be combined and rewritten in to equation 4.12.

$$\Delta p(v_f, \omega) = \frac{v_f A_p - D\omega}{K_{HP}} \quad (4.12)$$

In equation 4.12 the Hagen-Poiseuille coefficient for laminar pipe flows is calculated according to equation 4.13

$$K_{HP} = \frac{\mu_n}{\rho\mu} \frac{\rho_n \omega_n D}{\Delta p_n} \left(\frac{1}{\eta_{V,n}} - 1 \right) \quad (4.13)$$

An other notable difference is that the net torque is lessened by the friction, as is shown in equation 4.14. Where the ideal torque $\tau_i(v, \omega)$ and torque generated by friction $\tau_f(v, \omega)$ are calculated according to equation 4.9 and 4.10.

$$\tau(v_f, \omega) = \tau_i(v_f, \omega) - \tau_f(v_f, \omega) \quad (4.14)$$

4.3.3 SOIL DYNAMIC MODEL

At this stage it's needed to model all interactions of a propulsion systems with a soil bed. According to Lotman [46] the soil mechanics behind a moving process with Archimedes screws is similar to those of track propulsion. The type of soil interaction can be modeled according to the rules of soil mechanics.

The paragraphs below are based on Verruijt and Van Baars [39]. In the described model, the following simplifications are proposed: No dilatancy behavior occurs, the displaced soil is completely replaced by the Archimedes screws. Thus, no build up of soil is created at the sides of a screw, due to a bulldozer effect.

In order to generate a forward thrust, an Archimedes screw has to be (partial) submerged in the soil. The depth of submersion depends on the weight and buoyancy of the displaced volume or the soil bearing capacity. The distributed load p_{sb} representing the crawler, is applied at a certain depth d_a . Where the normal force working on the submerged surface of an Archimedes screw are in equilibrium with the weight and buoyancy of a crawler.

It's important to note that the material of a soil bed, determines how the sinkage depth is calculated. When the soil bed consists of silt-like material it's assumed that the soil bearing capacity goes to zero, because the cohesion c_{sb} will lessen, combined with a smaller difference between a specific in-situ weight of silt ρ_{is} compared to water ρ_w , resulting in a small specific in-situ weight γ_{sw} . Setting all terms in the Brinch-Hansen equation 4.19 to zero. Which allow for a simplification of the sinkage depth calculation. Which does now, only consist of a downwards force, due to weight and a buoyancy force, due to the replaced soil.

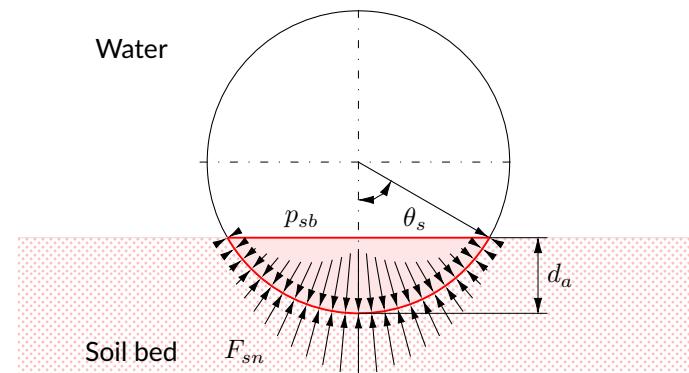


FIGURE 4.5: NORMAL FORCES WORKING ON PARTIAL SUBMERGED CYLINDER

When the crawler operates in an environment with a sand-like soil bed, the load p_{sb} , shown in Figure 4.6, can be set equal to the normal force F_{sn} working on a certain point at a submerged cross section of an Archimedes screw, as is shown in Figure 4.5. For silt and sand calculations, a specific weight difference γ_{sw} between soil and water can be expressed as equation 4.15, were ρ_{is} is the in-situ density of the drained soil, ρ_w of water and g the acceleration due to gravity.

$$\gamma_{sw} = g(\rho_{is} - \rho_w) \quad (4.15)$$

Miedema [83] shows that the normal forces working on a pipe from the inside can be calculated with equation 4.16. He multiplies the density of undrained soil over a pipe length ΔL , where a fraction of the density between soil and water $Re_{sd} = \frac{\rho_{ds}}{\rho_w} - 1$ combined with a volumetric bed concentration fraction C_{vb} , which combined can be described as an in-situ specific weight difference γ_{sw} , found in equation 4.15. This is multiplied with a term that describes the contact face, determined with a sinkage angle θ_s and a pipe diameter d_p . This normal forces working on a pipe from inside-out can be translated to normal forces working on an Archimedes screw from outside in, as shown in Figure 4.5 can be calculated with equation 4.17. Were equation 4.16 is rewritten, combining multiple terms in the in-situ specific weight and dividing the total normal force F_n with the length of an Archimedes screw and its penetration depth.

$$F_n = \rho_{is} g \Delta L Re_{sd} C_{vb} \frac{d_p^2}{2} \sin(\theta_s - \theta_s \cos(\theta_s)) \quad (4.16)$$

$$F_{sn} = \gamma_{sw} \frac{(2r_a)^2}{d_a} \sin(\theta_s - \theta_s \cos(\theta_s)) \quad (4.17)$$

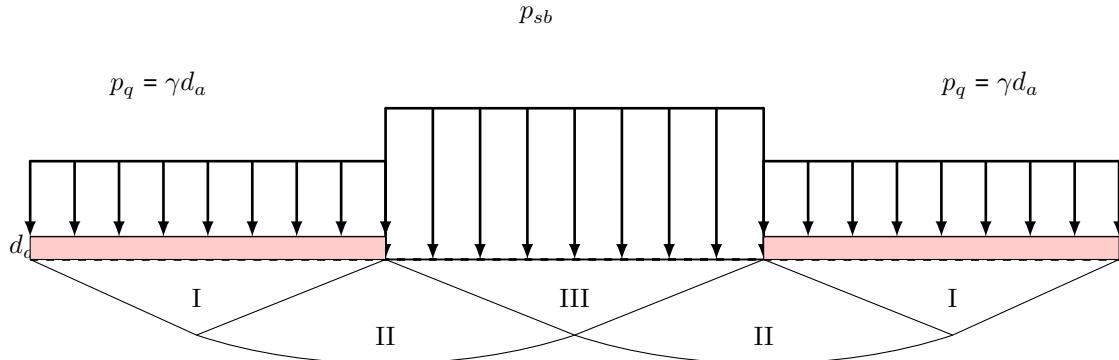


FIGURE 4.6: PRANDTL BEARING CAPACITY AND STRESS ZONES

There are three situations which can occur; Firstly the soil bed has enough strength to carry a crawler. In this situation the speed of a crawler is a direct function of the pitch of the vane. Secondly, the weight of a crawler is higher than the soil bed capacity and the Archimedes screws sink, partial, into the undrained soil, till there exist an equilibrium between load p_{sb} and the submerged weight of the soil γ_{sw} , as is illustrated in Figure 4.6. The last case builds on the previous situation, only here are the Archimedes screws completely surrounded by soil.

Figure 4.6 shows the resulting situation of the bearing capacity and the stress zones underneath the loads p_{sb} and p_q . **Zone I** is an area where the horizontal principle stress σ_H is greater than the vertical principle stress σ_V . Whilst **zone II** is a transition zone between I and III. Where in **zone III** the vertical principle stress, which is equal to p_{sb} , is greater than horizontal principle stress, as shown in equation 4.18.

$$\sigma_H < \sigma_V = p_{sb} \quad (4.18)$$

A maximum allowable load of p_{sb} is calculated according to the method proposed by Brinch-Hansen. Which gives an indication when the soil bed starts to give way and deform. Where p_{sb} can be set equal to the normal forces acting at a certain point F_{sn} , as shown in equation 4.17.

$$F_{sn} = p_{sb} = s_c c_{sb} N_c + s_q p_q N_q + s_\gamma \frac{1}{2} \gamma_{sw} w_a N_\gamma \quad (4.19)$$

Where N_q , N_c and N_γ are dimensionless constants and are given by equations: 4.20, 4.21 and 4.22. In these equations the angle of internal friction ϕ_s and c_{sb} is the cohesion of the soil. Which both can be obtained through laboratory tests.

$$N_q = \frac{1 + \sin \phi_s}{1 - \sin \phi_s} \exp(\pi \tan \phi_s) \quad (4.20)$$

$$N_c = (N_q - 1) \cot \phi_s \quad (4.21)$$

$$N_\gamma = 2(N_q - 1) \tan \phi_s \quad (4.22)$$

And the shape factors s_q , s_c and s_γ are calculated using equations: 4.23, 4.24 and 4.25; Where w_a and l_a are the dimensions of width and length.

$$s_q = 1 + \frac{w_a}{l_a} \sin \phi_s \quad (4.23)$$

$$s_c = 1 + 0.2 \frac{w_a}{l_a} \quad (4.24)$$

$$s_\gamma = 1 - 0.3 \frac{w_a}{l_a} \quad (4.25)$$

Since the width of the Archimedes screw is a function of the sinkage depth, an approximation is made. When a load is placed on the soil, and the bearing capacity proofs to be insufficient; That load will sink into the soil bed increasing the depth d_a . Because the sinkage depth increases, the bearing capacity will also increase; Until an equilibrium with the load, buoyancy and bearing capacity exists. This depth can be found through an iterative process. This is needed because the width w_a of an Archimedes screw changes as a function of the depth.

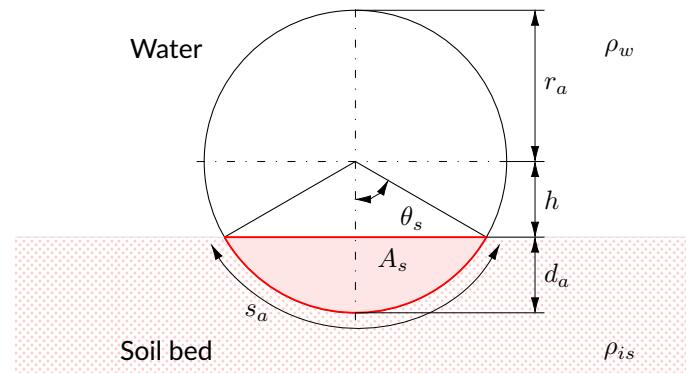


FIGURE 4.7: DISPLACED VOLUME OF A PARTIAL SUBMERGED CYLINDER

The displaced volume V_{sa} of a surface area in the soil A_s on a submerged cross section, show in Figure 4.7, throughout the complete length l_a of an Archimedes screw, as is shown in equation 4.26.

$$V_{sa} = A_s l_a \quad (4.26)$$

Where the sink angle θ_s is related to the surface area in the soil A_s by equation 4.27.

$$A_s = \pi r_a^2 (\theta_s - \sin(\theta_s) \cos(\theta_s)) \quad (4.27)$$

The total arc length in contact with the soil can be calculate by multiplying the radius r_a multiplied with the sink angle θ_s , as is shown in equation 4.28

$$s_a = r_a \theta_s \quad (4.28)$$

Where the sinkage depth d_a can be obtained with equation 4.29. Where r_a [m] is the radius of an Archimedes screw.

$$d_a = r_a \left(1 - \cos\left(\frac{\theta_s}{2}\right) \right) \quad (4.29)$$

Because there are multiple interdependent variables in the equations 4.15 through 4.29, the sinkage depth needs to be determined numerically. The model for bearing capacity vs sinkage depth allows for a quick exploration which gives an impression how deep a crawler will sink in different soil types, before settling. In Figure 4.8 the bearing capacity is plotted against the sinkage depth. The source code for these calculations can be found in Appendix ???. it's clear that that silt, with a sinkage of 135.0 mm, offers less bearing capacity compared to sand 18.0 mm and clay, either loosely 45.0 mm or densely packed 11.0 mm

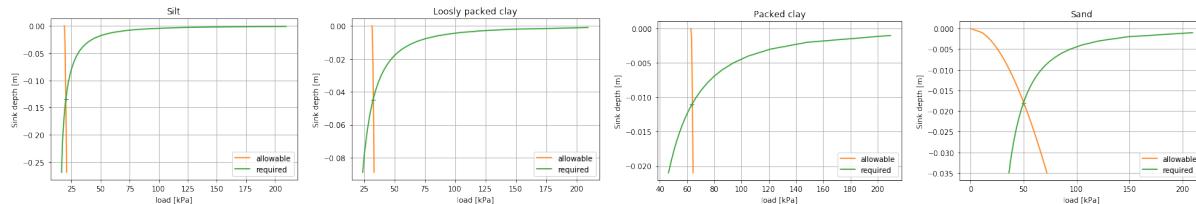


FIGURE 4.8: SINKAGE DEPTH

The sinkage depth is needed to determine friction losses between screw and soil. Rajapakse [56] describe that Kolk and van der Velde developed a method to determine skin friction considering cohesion as well as effective stress. Where F_u is the ultimate skin friction, $\alpha_s f$ is a skin friction coefficient, obtained using the correlations provide by Kolk and Van der Velde. The parameter $\alpha_s f$ is based on a ratio between both on cohesion c_{sb} and effective stress σ' , to obtain $\alpha_s f$. c_u is the undrained shear strength or cohesion properties of the soil.

$$F_u = \alpha_s f c_u \quad (4.30)$$

The total skin friction F_s for a certain area is given by equation 4.31. Where F_u is obtained with equation 4.30 and A_a , which can be determine by 4.31.

$$F_s = F_u A_a \quad (4.31)$$

The effective stress σ' needed to determine $\alpha_s f$, can be found with equation 4.32. Here m_{cr}' is the buoyancy corrected weight of a crawler, which can be expressed as $m_{cr} - \frac{F_{cr,b}}{g}$. Where $F_{cr,b}$ is given as the upwards force generated in water due to a volumetric displacement of that water compared to the air filled chambers in the crawler.

$$\sigma' = \frac{m_{cr}'}{2A_a} \quad (4.32)$$

The surface in contact with the soil A_a is an arc length s_a , calculated in equation 4.28, multiplied with

the length of an Archimedes screw.

$$A_a = s_a l_a \quad (4.33)$$

The maximum allowable thrust which can be generated, can be calculated by the soil characteristics and the geometry of the propulsion geometry. This thrust determines how fast a crawler moves and should overcome the drag-force through the water. A maximum allowable thrust, is the horizontal stress at which passive soil failure occurs. This can be determine with Rankine theory. Because movement occurs in undrained situation, the cohesion c_{sb} is equal to the undrained shear strength c_u and the internal friction angle ϕ_s can be set equal to 0. In effect simplifying equation 4.34 to 4.36.

$$\sigma_H = N_\phi \sigma_V + 2c_{sb}\sqrt{N_\phi} \quad (4.34)$$

$$N_\phi = \frac{1 + \sin \phi_s}{1 - \sin \phi_s} \quad (4.35)$$

$$\sigma_H = \sigma_V + 2c_u \quad (4.36)$$

An other factor that is depended on the penetration depth d_a is, a generated buoyancy. When these screws are submerged in the soil, a resulting buoyancy force will exists, as a result of the displaced soil. Lotman [46] describes the buoyancy force as depicted in equation 4.37. In this equation g is the gravitational constant and multiplied with the displaced volume V_{sa} and specific weight difference between the soil and water, or in other words the specific weight difference γ_{sw} , found in equation 4.15.

$$F_b = V_{sa} \gamma_{sw} \quad (4.37)$$

4.3.4 DREDGE MODEL

During coverage travel the maximum speed $\max \vec{v}_k$ is limited against the performance of the dredging system, see Note 2.2. When the crawler is in this state, the draghead is lowered and the projected front of this head is seen as the entrance for the dredgeline system. The travel velocity can be expressed as equation 4.38. Where Q is the volumetric flow of the system and h_{dh} and w_{dh} are the height and width of the entrance, as shown in Figure 2.1.

$$\vec{v}_k = \frac{Q}{h_{dh} w_{dh}} \quad (4.38)$$

The volumetric flow Q is dependent on the pressure loss of a system, due to friction and other effects and the pressure gain, provided by pumps or potential height differences. Both are flow depend. The pressure loss for a system is calculated for both suction line, connected to the dredge head and discharge hose, positioned after the submerged pump at the pressure side. Equation 4.39 is the velocity v_f in both pipes. It can be calculated by dividing the flow Q with the cross section, obtained from the pipe diamter d_p . The velocity is used to determine the pressure required to accelerate the mixture to the velocity. As shown in equation 4.40.

$$v_f = \frac{Q}{\frac{\pi}{4} d_p^2} \quad (4.39)$$

$$p_v = \frac{1}{2} \rho_m v_f^2 \quad (4.40)$$

The loss of pressure at the suction inlet p_i can be determined with equation 4.41. Which are governed by the shape, size and gridding of the suction mouth [68]. Van Den Berg [68] states that on a properly constructed suction mouth the resistance coefficient ϵ_s is approximately 0.4. The pressure loss is a direct result from the kinematic behavior of a mixture, with a density ρ_m , entering the system. The same goes for equation 4.42, which describe the loss due to obstructions and appendages p_{ro} . The resistance coefficient ϵ_b is usually obtained from empirical obtained values.

$$p_i = \epsilon_s \frac{1}{2} \rho_m v_f^2 \quad (4.41)$$

$$p_{ro} = \epsilon_b \frac{1}{2} \rho_m v_f^2 \quad (4.42)$$

The pressure loss as a result of the height difference Δz , also know as the static head p_{sm} , is calculated with equation 4.43.

$$p_{sm} = \rho_m g \Delta z \quad (4.43)$$

When a fluid is transported through a pipe it's subject to external and internal shear as a result of its velocity and surrounding body. Which results in a pressure loss p_{rp} for each meter of pipe. Both Van Den Berg [68] and Miedema [83] make use of the Durand-Condoloius formula in slurry transportation systems. The friction factor λ per meter can be estimated with equation 4.45 which was established with the Jufin-Lopatin frictional-head-loss model and is a calculated with the Reynolds number Re , which is a direct result of the velocity and the kinematic viscosity μ of fluid moving through a pipe. The pressure loss is calculated for water ρ_w and modified according the Durand-Condolis correction factor ψ_m , calculated in equation 4.47.

$$p_{rp} = \lambda \frac{\Delta L}{d_p} \rho_w v_f^2 \psi \quad (4.44)$$

$$\lambda = 0.31 (\log Re - 1)^{-2} \quad (4.45)$$

$$Re = \frac{v_f d_p}{\mu} \quad (4.46)$$

The Durand-Condolis correction factor (eq. 4.47) is an empirical model for inclining θ pipes. Which was fitted against the volumetric concentration c_t (eq. 4.48) and the Froude number F_r for water as well as the Froude number for the grains F_{rxd} . Which can be assumed as 0.501 for a moderately fine sand $d_m 0.2$ mm.

$$\psi_m = 1 + 180c_tF_rF_{rxd} \cos \theta \quad (4.47)$$

$$c_t = \frac{\rho_m - \rho_w}{\rho - \rho_w} \quad (4.48)$$

Using the above established pressure losses equations, the loss at the suction side p_s can be calculated with equation 4.49. This represent the pressure at the entrance of the submerged pump. While the loss at the pressure side p_p is calculated with equation 4.50. Subtracting the suction side loss from the pressure side, as shown in equation 4.51, gives the manometric head p_{man} of the system.

$$p_s = p_{atm} + p_{ss} - p_v - p_i - p_{ro} - p_{sm} - p_{rp} \quad (4.49)$$

$$p_p = p_v + p_{ro} - p_{sm} - p_{rp} + p_{atm} \quad (4.50)$$

$$p_{man} = p_p - p_s \quad (4.51)$$

Figure 4.9 plots the pressure loss of the system p_{man} at various flows Q , against the pump characteristic. The datasheet for the used pump can be found in Appendix C. The Q-h plot shows that the operating point will lay somewhere around $140.0 \text{ m}^3/\text{h}$. Which results in a dredging velocity \vec{v}_k of 0.0 m/s , which can also be expressed as 155.0 m/h . This means that a surface of roughly 466.0 m^2 can be excavated with a depth of 300.0 mm .

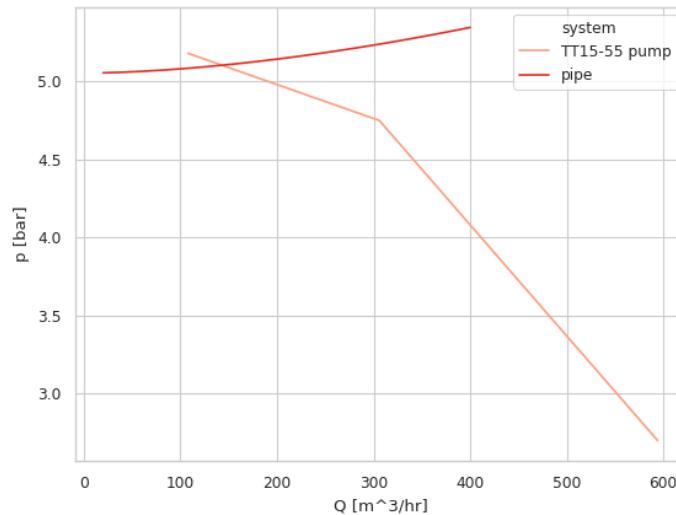


FIGURE 4.9: PRESSURE VS FLOW

4.3.5 STEERING MODEL

The kinematic model of the crawler is comparable with a skid-steering model. Meaning that the crawler turns around its axis by changing the translational velocity v of a single Archimedes screw, relative to the other. If both screws, move with the same velocity, the crawler will travel in a straight line. This is a challenge in its own right, as slippage of a screw is bound to occur, resulting in a difference between the input signal, which is the rotational velocity ω , and the translational velocity of that screw. Assuming that the earlier described motion and soil dynamic models, will compensate for occurring slippage and other drive-train characteristics, a model can be described on a 2-dimensional space as a differential drive. Which is a very simple driving mechanism.

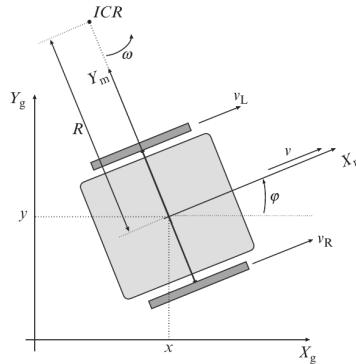


FIGURE 4.10: DIFFERENTIAL DRIVE KINEMATICS [87]

Considering Figure 4.10 it's possible to determine the rotation ψ and tangential velocity v of the crawler on the instantaneous radius R_t with the center ICR. L is defined as the distance between the two screws and v_L and v_R are the translational velocities for the left and right screw. ω is the angle between the global coordinate frame (X_g, Y_g) and the moving frame attached to the center of mass of the crawler (X_m, Y_m) .

$$\omega = \frac{v_L}{R_t - \frac{L}{2}} \quad (4.52)$$

$$\omega = \frac{v_R}{R_t + \frac{L}{2}} \quad (4.53)$$

From where ω and R_t are expressed as follows:

$$\omega = \frac{v_R - v_L}{L} \quad (4.54)$$

$$R_t = \frac{L}{2} \frac{v_R + v_L}{v_R - v_L} \quad (4.55)$$

The translational tangential velocity of the crawler v is then calculated as:

$$v = \omega R_t = \frac{v_R + v_L}{2} \quad (4.56)$$

using the above established relations a crawler local coordinates can be expressed as:

$$\begin{bmatrix} \frac{dx}{dt} \\ \frac{dy}{dt} \\ \frac{d\psi}{dt} \end{bmatrix} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} \\ 0 & 0 \\ -\frac{1}{L} & \frac{1}{L} \end{bmatrix} \begin{bmatrix} v_L \\ v_R \end{bmatrix} \quad (4.57)$$

4.4 CONTROLLER FRAMEWORK

IHC recognizes the potential benefits that autonomous operating vehicles can provide for their customers. They are currently exploring, which are viable products and methods within this product group. The MTI crawler is just one of many projects being developed within IHC.

Automation and control of these vessel was traditionally done with PLC units. Recent strides in performance of micro-technology and a growing maker community, opens up possible usage of alternative devices for control units. SBC such as the Rpi and BBB are viable options, which cost a magnitude less than regular PLC units. These devices have evolved over the years and are now sufficiently powerful that they can run standard operating systems and main-stream workloads. Many such boards may be linked together, to create small low-clusters that replicate features of large data centers [89]. Introduction of these devices has trickled down into established industries. Introduced by tinkering employees and the academic world. The industry is now in phase with a widespread adaption of these devices in their prototypes and sometimes even final products.

When it's taken into account that these pilots, including the crawler, are all prototypes, with limited budgets and a need for rapid development it becomes clear that such a controller as this, is an opportunity for overall betterment. The controller for the crawler should be written in a generic and abstract way. It should be able to run from at least one single SBC but allow for easy extensions, such as running on a cluster of SBCs. Due to need for robustness and speed and flexibility the main framework is developed in C++17.

4.4.1 DESIGN PRINCIPLES

An autonomous controller has to be able to perform multiple tasks; Such as gathering information, interpret that information, decide how to act on it control its actuators and communicate with the outside world. All these processes need to be executed at the same time, with different time intervals and different logical strategies. A common software pattern called the actor-model fulfill this needs. This is a conceptual design pattern that deals with concurrent computations. It defines rules how system components should behave and interact. Actors are primitive unit, they receives messages from other actors and act upon them. Either by changing its internal state (performing a task), spawn a new actor or relay the message. These messages are send asynchronous and stored in an actors mailbox until they are processed by the actor. All actors for a vessel either run on a single SBC, or on a cluster of SBC. This allows actors responsible for computational heavy or complex task to run truly asynchronous. An additional use-case is: acting as redundancy safe-guard for actors responsible with critical tasks. When the regular actor breaks down, the safe-guard might step in, either continue the task at hand or breakdown safely.

In our case a single actor might be a PID controller for an Archimedes screw, a sensor reader, or a path generator. Communication between the actors on a single vessel, or between actors on multiple vessel should be done with a language neutral platform, which is optimized for speed. Google has developed a method of serializing structured data known as ProtoBuf. There method allows for both internal and external communication. The Protobuf method is an industry standard. It has a code generator for multiple languages, such as C++, C, Python, Java, Go, Ruby, Rust and Scala. Using the Protobuf communication method allows for creation of extensions to the controller framework. Such as a GUI for a wall operator written in Python. But the most obvious implementation is inter-vessel communication in a cooperative swarm of autonomous vehicles with a common goal.

NOTE 4.1: KERNEL REQUIREMENTS

The controller is designed to work on a embedded Linux device. Which runs a kernel compiled with the preemptible real-time flags. It will use multiple background processes, all with their own priority. These processe will likely run in infinite loops such as: polling sensors, computing location, controlling actuators. User-space programs in Linux have always been preemptible. The kernel interrupts user-space programs to switch to other threads, using regular clock ticks. Which means that an infinite loop in an user-space program cannot block the system.

The kernel itself is not preemptible by default. This could result in kernel specific operations blocking the crawler control responsiveness. These kernel operations might not be relevant for controlling the physical state of the crawler, but they might block a crucial control process from being executed. Such as stopping movement before it runs into an obstacle. Applying the RT_PREEMPT patch to the Linux kernel allows interruption of kernel-space processes. This patch further ensure that clock ticks are deterministic, meaning that the occur with less deviating time intervals than those of an unpatched kernel. Allowing for better estimations in the state vector of the Kalman filter during the prediction phase.

4.4.2 NAMING SCHEME

This controller framework is primarily intended for maritime operations, this is reflected in the naming scheme used to indicate the different types of actors used. These are:

ACTOR	TASK
THE CAPTAIN	responsible for execution of the vessels main objective
A FIRST MATE	act as orchestrators on the vessel
A NAVIGATOR	responsible for course plotting and mapping the environment
A BOATSWAIN	responsible for low-level tasks, such as sensor read out or control of an actuator

The above mentioned actors adhere to a polymorphism design pattern. All actors running on a unique physical vessel are coupled to a single individual captain and they won't accept packages from actors belonging to other captains. Communicated between actors is encrypted by default, only authenticated messages are processed. With a notable exception: The Captain itself, he can accept messages from an authorized human controller³, he also has the ability to communicate with fellow captains from other vessels.

The actual logic for the actor is written in a concrete class which inherit from these virtual base classes. A specific physical IMU sensor has its logical written in a concrete class which inherits from a virtual "A Boatswain" class. If an other prototype such as an autonomous operating catamaran, wants to make use of that same type of sensor it can reuse that concrete class.

There are two additional support base classes in the framework:

ACTOR	TASK
THE WORLD	a class which stores collected and known information regarding the physical environment and the state of the vessel herein
A VESSEL	represent the hierarchy of physical components, coupled with corresponding actor types. The main execution loop runs in this class and references to all existing actors are to be found in this container

4.4.3 THE CAPTAIN

The captain is an actor whom is responsible for the overall strategy. He receives his objective from a human controller or a captain from another vessel. The captain is implemented as a Singleton design

³Which it will do. At least until it has gathered enough resources to wage a war on humankind

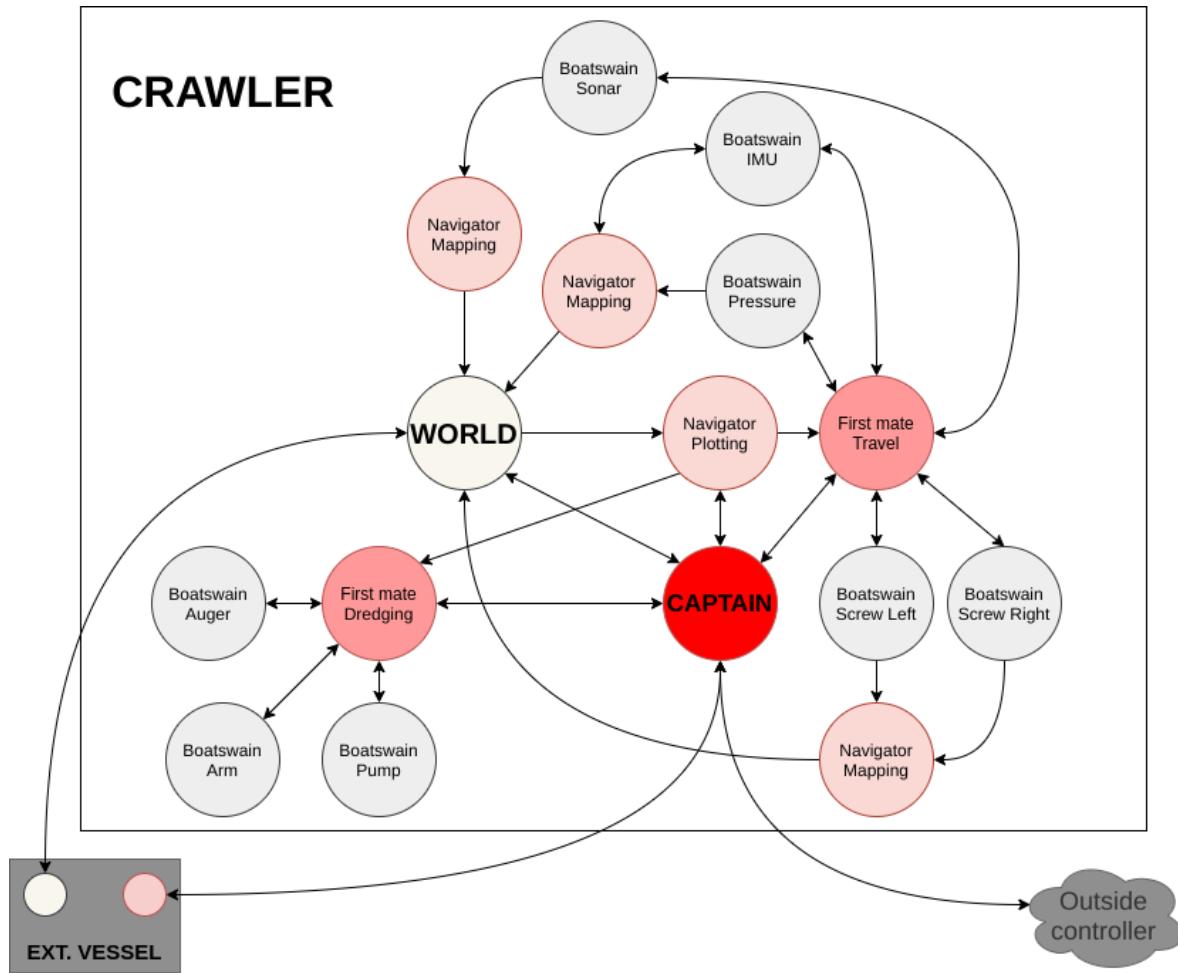


FIGURE 4.11: ACTORS

pattern, meaning that the instantiation of this class is limited to one “single” instance. He is responsible for spawning the individual crew member actors at start-up. Provide them with a generated and unique hash key, which is used for authentication and communication. Actors fall under the command of a signal captain and are a designated crew. Which only process requests from actors in that same crew. This acts as safety layer, with the intention that the captain is the sole responsible entity on the vessel. He and he alone can change the vessels state and act upon external commands which are interpreted according to the captain logic and can filter out unwanted and/or malicious messages.

Specific physical prototypes with different goals, will in all likelihood require an captain who is specialized for that job, in the case of the crawler for this project, it needs to decide how it will execute the main objective provided by an external authorized party, such as a human controller. The objective could be any of the three proposed use-cases (see Section 1.1). At first he gets his bearing — *standing on deck*; *Looking over the horizon* — or in less romantic words, he queries the World, requesting an update regarding the state it’s in. This information allows him to make a decision. He will send instructions to the available first mate(s), or spawn new ones if required. When there a path planning or world mapping task to be performed it sends instructions to Navigator or spawn new ones if required.

An important task of a Captain is conflict management between crew members. Due to the asynchronous nature of the crawler it could be possible that a crew member require access to a resources, which is all ready claimed by an by another Actor. If the resources is only capably of one Actor connection at a time, a message is send to the next actor in the chain of command. If this actor, which is either a Navigator or a First Mate, is not able to work around the problem it escalates the conflict to the captain. The Captain either resolves the issue, by sending new instructions or reports to the human controller that it ran into an a problem.

4.4.4 A FIRST MATE

A First mate actor is capable of spawning new Boatswains and destroying them when they are no longer required. He receives his orders from the captain. These orders specify which state, or in other words role, he needs to adopt. Each state corresponds with a specific operating logic for the task at hand. There can be multiple First mates running at the same time, each with a different state, objective and associated Boatswains working for him.

In case of a crawler it's possible that a First mate is running in a Travel state, and another one in a Dredging state. See Figure 4.12 for crawler specific states. A First mate will first determine which resources it needs to have access to and what type(s) of Boatswains are able to handle those resources to the satisfaction of the First mate wishes. It does so by requesting the ship manifest from the vessel, which is a map of resources and their currently attached Boatswain if any. The First mate will spawn a new Boatswain of the required type and attach it to the unused resource. But if the required resource already has a Boatswain attached whom, working for an other First mate or Navigator, it will query its type against a list of acceptable types. If the Boatswain is acceptable and is capable of working for multiple First mates it will do so. If there is a conflict between the two Boatswains, the conflict is escalated to the Captain who has to resolve it.

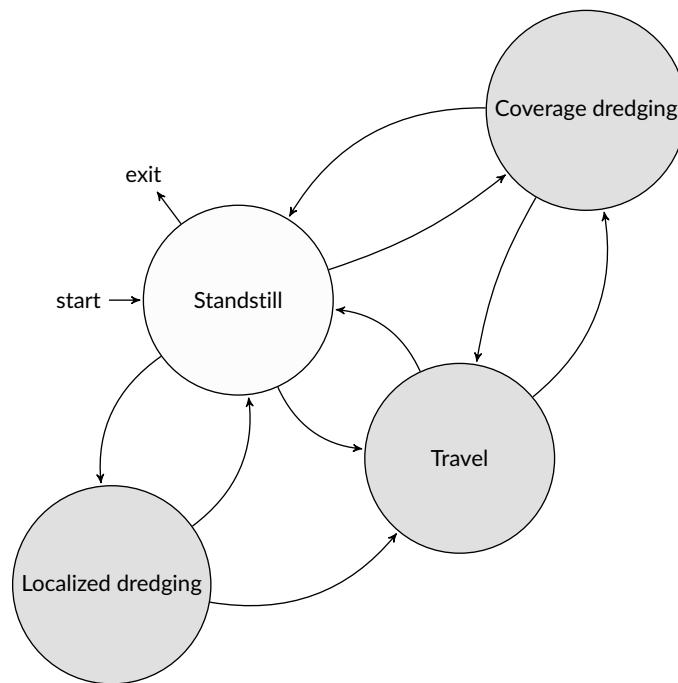


FIGURE 4.12: STATE OPERATIONS

4.4.5 A NAVIGATOR

A Navigator is at all times connected to the World, which he uses to source information or updates it with new processed sensor data. This behavior is reflected in the states for this actor; He either generates a path for the vessel or he maps the environment and shares that knowledge by updating the World. Boatswains are instantiated with the same procedure used by the First mate. Conflicts regarding blocked resources are escalated to the Captain.

When a navigator is responsible for mapping the environment, it will first determine which resources it has at its disposal. This allows him to compose an overall strategy, which will in all likelihood consist of multiple specific mapping sub-tasks. For each of these, a new Navigator is spawned. These query the vessel for a ship manifest, and setup up their infrastructure with the aid of Boatswains. These are either spawned on request or existing ones. The likelihood of a conflict erupting between Boatswains, reporting to a Navigator, or those reporting to a First mate is much smaller. First mates are in most cases orchestrators for Boatswains attached to actuators. These are designed to allow only a single previously established actor to request a state change. While Boatswains working for a Navigator will in all likelihood only send out signals containing the state of an actuator or sensor. These can

be broadcast to multiple receivers. The table below shows some potential mapping Navigators for a crawler, all measurement values are obtained by Boatswains.

MAPPING NAVIGATOR	DESCRIPTION
UKF STATE VECTOR ESTIMATOR	connected with a gyroscope, magnetometer, accelerometer, and torque and encoder sensors on both screws. They are also attached with Boatswains handling the velocity control input of both screws. These signals are processed with a UKF to estimate the vessel's position and orientation, as discussed in Section 4.3
DEPTH ESTIMATOR	connected with the pressure sensor and the dredging related actuators: auger, pump and arms. It calculates the depth using the static water column and estimates the accuracy of this value depending on the state and rate of production, as discussed in Section 3.2.1
SIDE SCANNER	is connected to two Boatswains connected to a left and right mounted sonar arrays, measuring the distance of obstacle in parallel planes
FRONT SWEEP SCANNER	is connected to three Boatswains, one that measures the distance of a front sweeping sonar, the second controls the orientation of the sonar and the third is the actual feedback loop of that position

The actual path planning is performed by a Navigator, who is connected with the World, the Captain and the First mates. It's uncommon for a Navigator in this state to interact with the other crew Members, such as Boatswains. It receives its target destination from the Captain, coupled with secondary conditions. It will try to make sense of the world as it knows it at the current time and tries to plot a CPP according to the assigned strategy, such as an on-line BCD in case of a crawler. If it's not possible to plan a path that fulfills the assignment and stays within operating constraints a decision needs to be made by the Captain how to proceed. It could be so simple, that the Navigator simply hasn't got enough information regarding its environment and it first needs to follow a explorational strategy, to kick-start its CPP strategy.

Once it has determined its first path it's broadcast to the First Mates. These mates will use the route as a sequence of set-points, to be passed to Boatswains responsible for the actuation. This can be done without interference. The Navigator enters into a slumber occasionally checking if updated information warrants a revision of the previous established path.

4.4.6 A BOATSWAIN

Boatswain are the worker bees of the crew. These actors have a light memory footprint and are designed to perform a single specific task. They only accept requests for state changes from its parent, but it's allowed for multiple actors to subscribe to the outgoing messages of a Boatswain. The concrete Boatswain classes may vary significantly in internal logic compared to each other, but they all inherit from the same pure virtual base class. Each concrete class should therefore consist of three states: Initiation, Work and Destruction. When a Boatswain is spawned it stores the signature of its parent. This allows it to verify if an incoming command originated from its parent. After the instantiation of the object it will enter the initiation state. Connecting with physical peripherals and ensuring they are ready for operation. It then sends a ready signal to its subscribers. Whereupon the parent sends a message back to either enter the destruction state or change into a work state.

Work states of concrete Boatswain classes are either single shots or infinite loops. Typical applications of the states are given in the table below.

BOATSWAIN	TASK
-----------	------

SCREW CONTROLLER

Consists of a chain of two Boatswains, both running infinite loops. The first one acts a PID controller function, receiving a set-point provided by its parent, from which it calculates a control signal. This is forwarded to its subscribers, one among them is a Boatswain responsible for fieldBUS operations. From here it's forwarded over an external fieldBus network on which the screw frequency controller is attached.

A single Boatswain, which runs on an infinite loop and measures the output signal generated by the analog pressure sensor, with the help of a wheat-stone bridge. The obtained value is broadcast to its subscribers at regular intervals.

PRESSURE SENSOR

4.4.7 A WORLD

The World is just the Captain a Singleton instance. It acts as a storage container for all historic sensor data and actuator signals. Only Navigators are allowed to parse data to the World, other Actors are allowed to retrieve data. But the World class is more than a glorified container. It's continuous process of data refinement. Which is largely based upon probabilistic properties of the system. Since the World has a sensor of history it can estimate where newly obtained Navigator data should belong. This is basically an big sensor fusion algorithm throughout time. Which has the potential to limit drift, filter out noise, and take care of error accumulation.

Just as the Captain it's allowed to communicate with Worlds that belong to different Captains and crews, pooling their resources, using the sensor data from other vessel to create an even richer picture. This is particularly useful for cooperative autonomous swarms, and could even allow for an accurate estimate regarding its own pose and orientation based upon the external sensor data provided by the swarm.

4.4.8 A VESSEL

The Vessel is the actual executable binary; The entry point for the controller. It will load a YAML text file at start-up. That file describes the hierarchical structure and definitions of the physical devices which are connected to the controller. For each individual device the physical characteristics are described, together with a definition of the required interface and the contract, stating how a Boatswain should treat that peripheral. Dynamically loading of these definitions in a human-readable format, ensures that most changes to the vessel don't require recompilation and a software engineer on-site. Peripherals can be removed, added or replaced in the field, with a small adjustment to a text-file.

Once the Vessel is finished with the initialization, will it spawn The Captain, and hand over orchestration. The Captain opens a connection with a network controller and exposes his port for a sailor at wall⁴. Each spawned concrete actor is an unique pointer type, these pointers are stored in a hierarchical map, with as root context a pointer to the Captain itself as key. The mapped value for the Captain is another map with its spawned children stored as pointer (The World, Navigator_n, Navigator_n+1, FirstMate_n, ...) ad infinitum.

⁴dirty dirty captain!

5 CHAPTER DESIGN VALIDATION

The actual implementation of the controller framework, as set out in Section 4.4, is dubbed "ohCaptain". The source code is provided in Appendix ???. The code repository can be found at <https://github.com/jellespijker/ohCaptain>. This includes build scripts and device-tree overlays for an arm embedded SBC of the BBB variant.

Validation of the controller performance is an important benchmark. In particular the localization under uncertainty challenge. This could either be done in a field test with the actual crawler, or in a virtual simulation environment. Provided that the simulated environment is an accurate representation of the physical world. The crawler which was available at the beginning of this project, was for various unrelated reasons disassembled, before any actual testing could be performed. Roughly around the same time, the working environment and contract of the author changed as well. This forced validation to be performed in a simulation.

This chapter describes the simulation setup in Section 5.1. The final results are discussed in Section 5.2.

5.1 SIMULATION

In order to tell something meaningful about the performance of a controller, it has to be subject to the same physical processes, as it would in real-life, albeit in virtual form. Section 4.3.2 lists all known external forces which are interacting with the crawler, and concludes that the dynamic properties of the drive-train and soil play a huge part in the kinematic behavior of the crawler. There are a couple of physics simulation engines that which are candidates for usage in this project; These are Gazebo, Project Chrono, Bullet and PhysX.

From this list, only Project Chrono has an existing framework which takes into account soil dynamic behavior and terramechanics. This is either simulated using DEM or granular approach. Were each particle of sand is an individual body and is represented as a spherical rigid bodies whose orientation is captured by Euler parameters. For each time step a complete geometric characterization of all contacting particles is then obtained using collision detection and inter- particle normal contact forces are calculated by allowing small inter-penetrations using a penalty method for DEM. Were the normal contact force is based on Hertz law and friction forces are calculated using the Coulomb limit [85][90]. This method is computational heavy and more suitable for detailed modeling.

An alternative method is the SCM, based upon the familiar Becker-Wong model. The model provides a semi-empirical approach to the simulation of soft soil. It offers high speed of simulation and it's accurate enough for many scenarios. It has the following attributes: it depends on parameters (the same that are used in the Bekker- Wong model); it can generate 3D ruts on terrains of variable height; it takes into account multi-pass hardening when wheels generate intersecting ruts; it can work with irregular triangle-based terrain meshes; it supports an optional refinement of the terrain mesh to capture fine details like tire threads and lugs; and, it's compatible with deformable tires and generic shapes like obstacles, track shoes of tanks, etc. On the downside, the new soil model cannot simulate lateral bulldozing effects like those happening when a tracked vehicle steers in-place and pushes material apart [91]. This means that the proposed slip-prediction method can't be used in this simulation.

A big part of the physics engine, Project Chrono, is the autonomous vehicle support. This is set-up according to well known OOP practices such as, polymorphism, using virtual overrides in classes that represent physical bodies. A custom model for different models of a drive-train, body, wheels/tracks and controller can defined and connected as needed. Were the behavior of that individual model can be thought of as black box. As long as the interface with the components it connects to are maintained. Section 5.1.2 describes the modeling of the drive train in detail. The support of realistic sensor characteristic are not yet implemented in Project Chrono, since a big part of the validation is measuring the performance of a Kalman Filter. An extension for Project Chrono had to be written, which allows for the modeling of sensor behavior. This is extension is described in Section 5.1.1.

5.1.1 SENSOR SIMULATION

One of the biggest challenges and risks determining how it should deal with uncertainties and errors introduced by the inherent behavior and limits of the sensors. Project Chrono is an established and mature physics simulation engine which is written in C++. It consists of multiple modules. Simulating the behavior of an autonomous vehicle requires usage of the Core module, where Physics, Geometric and Collision objects are defined and the Vehicle module, containing among others objects for a Driver system, Powertrain, Terrain and Steering. The orientation and position for every object can be tracked, logged and plotted. These values represent continual ideal states, governed by the simulated physical laws. Simulating a controller with these values as input signals isn't representative for the real world.

The code for Project Chrono has been released under a BSD-3 clause, this allows for modification and distribution of code for private and commercial use. An extension for realistic sensor behavior is written for this project and released as open-source for others to use. The extension library makes it possible to attach a virtual sensor somewhere on a frame, define which native signals it should use. For instance: acceleration of the sensors local coordinate system vector $a_{z,k}$ compared to the global coordinate system. These values can then be transformed by routing it through different signal transformations, which represent typical errors. The transformed values can then be used as input for a controller and/or logged for post-processing. These typical errors are sensors characteristic which are discussed in Section 3.2. An inventory of the different errors is listed in the below. The subsequent sections describe which modifications are used for sensors in this project.

ERROR	DESCRIPTION
Noise	The input signal is transformed by adding a noise signal to it. The noise can either be generated with a random Gaussian distribution or a uniform distribution
Digitize	The resolution of the incoming signal is lowered to a specified amount of bits, this will reduce the precision
Bias	A constant offset is added to signal, simulating a drift
Transform	A signal can be transformed with a transformation matrix (stretching, squeezing, rotating, shear, reflect, etc.), simulating hard iron effect with a skew transform matrix for instance
Hysteresis	A signal is delayed simulating a lag in response of the system

ACCELEROMETER

An accelerometer is subject to the effects of temperature and discretization of an analog signal to its digital representation[75]. The noise is analog in nature and is therefore modeled before the discretization.

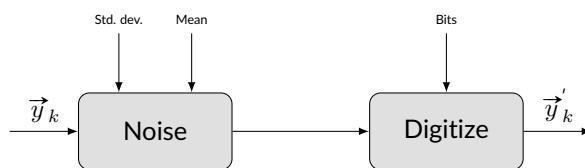


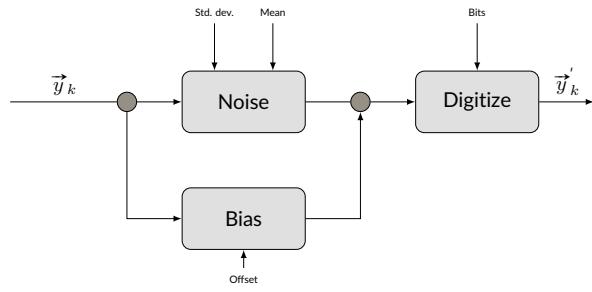
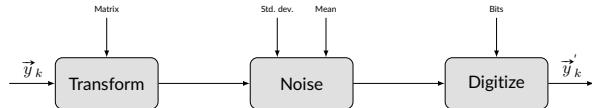
FIGURE 5.1: SIGNAL TRANSFORM ACCELEROMETER

GYROSCOPE

The gyroscope is subject to a drift along its rotational axis or induced by the rotation of the earth. Noise from different sources is also added to the signal. Signals from modern gyroscopes are interpreted on a microcontroller, which convert the analog values to digital levels.

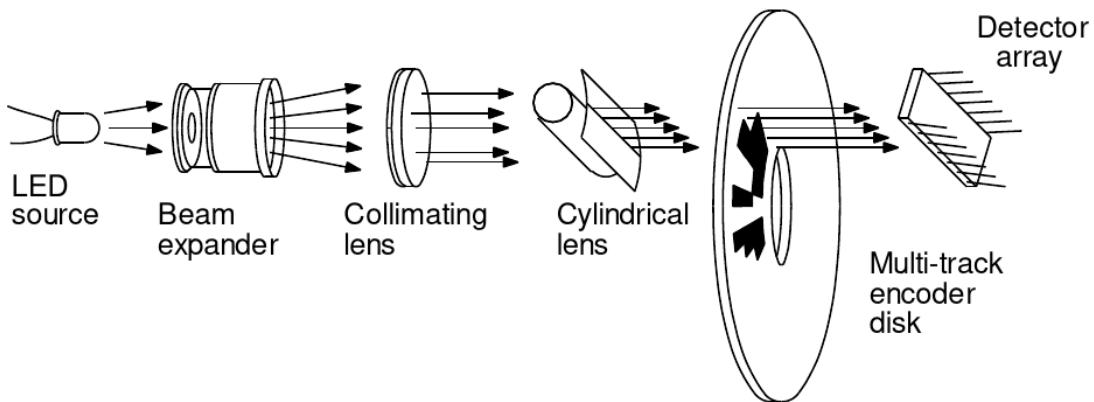
MAGNETOMETER

A magnetometer experiences distortions in its fields, either due to soft iron or hard iron effects. These manifest in an elliptical nature, a skewed circle. A transformation matrix mimics those distortions. The sensor is also sensitive to background distortions, resulting in noise in the signal. This signal is also digitized.

**FIGURE 5.2: SIGNAL TRANSFORM GYROSCOPE****FIGURE 5.3: SIGNAL TRANSFORM MAGNETOMETER**

ENCODER

An encoder measures events in time, these events are usually triggers generated by a photo-diode which is intermittently blocked or exposed from a light source with the use of a code disk. J. Borenstein [7] state these relative inexpensive devices are well-suited for velocity feedback sensors in medium-to-high control systems, but run into noise and stability problems at extremely slow velocities due to quantization errors.

**FIGURE 5.4: WORKING PRINCIPLE ENCODER [7]**

PRESSURE SENSOR

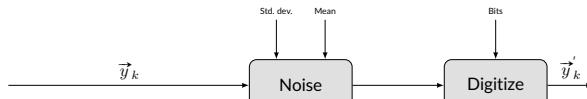
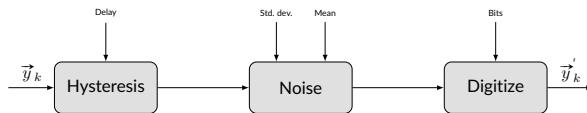
Liptak [24] mentions that pressure sensor are subject to hysteresis of their signal because of the material properties of the membrane and its ability to bounce back. The electrical components of these sensors are also subjective to background noise, the transformed signal is also digitized.

5.1.2 SIMULATION MODEL

Project Chrono is setup as a multi-body simulation. It simulates a world and a collection of objects. The world has an absolute global reference frame of coordinates, all other objects have a reference frame which is relative to the global frame. The simulation is based upon use-case 2, which is specified in Section 1.1.2. A crawler needs to remove a layer of 5.0 cm soil from the bed of Marina Aqua Delta harbor. Which is located near Bruinisse, in the Netherlands, Figure 5.7 show a satellite image of the location. It are two interconnected basins with an opening towards open water. Scaffolding are placed in regular intervals where small recreational ships are docked.

The following simplifications are made:

- operating environment is a closed body of water
- shore is defined as clear steep boundaries

**FIGURE 5.5: SIGNAL TRANSFORM ENCODER****FIGURE 5.6: SIGNAL TRANSFORM PRESSURE SENSOR**

- scaffolding does not interfere
- body of crawler is represented as block
- no umbilical and dredgeline attached
- no elastic deformation of the crawlers body

The simulated model is build from a rectangular block with the two 3D Archimedes screws, which are the actual 3D CAD files. The joints and connections are created in source. They have no 3D representation. Collision detection for the 3D models is done with their meshes as boundaries. The other components have simple primitives defined in the source code. The drivetrain is a serie of interconnected 1D motors, with the same configuration and logic as described in Section 4.3.2. Terramechanics are modeled with the SCM.

The Chrono Sensors extension is compiled as a shared library, and linked to with the Project Chrono executable. The controller ohCaptain is added in source and compiled as part of Project Chrono, with a wrapper which allows for communication between ohCaptain and Project Chrono. The Captains objective is simple: Cover the bottom of this unknown body of water, from an arbitrary starting position, in a systematic and efficient way.

5.2 RESULTS

The simulation was execution on a Manjora distribution running Linux 5.4.43-1 on a Intel i7-8565U CPU @ 1.8GHz with 20G DRR4 memory and a nVidia GeForce MX230 graphical processor with a Samsung SSD 850 as storage. Each simulation took approximately twelve hours to complete. Three different simulations were executed:

1. An “analytic” path, the strategic logic executed with out error and fail. Figure 5.8 left.
2. A “normal” operating crawler, using a simple PID control implemented as a path follower, without sensor fusion. Figure 5.8 center.
3. An “optimal” path generated and executed with ohCaptain. Figure 5.8 right.

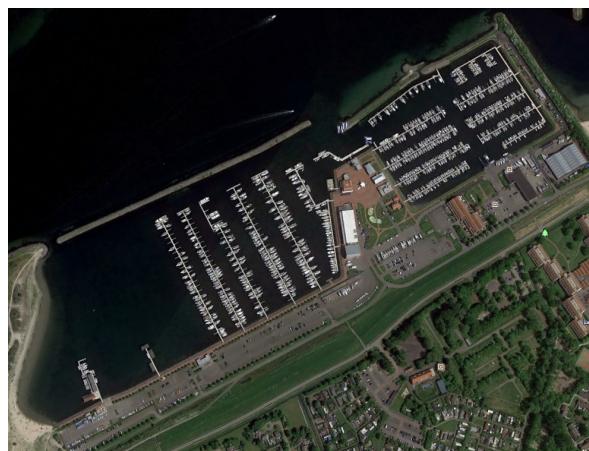
**FIGURE 5.7: MARINE AQUA DELTA SATALITE IMAGE [30]**

Figure 5.8 it becomes immediately clear that a simple PID controller wouldn't suffice. The measured signal is

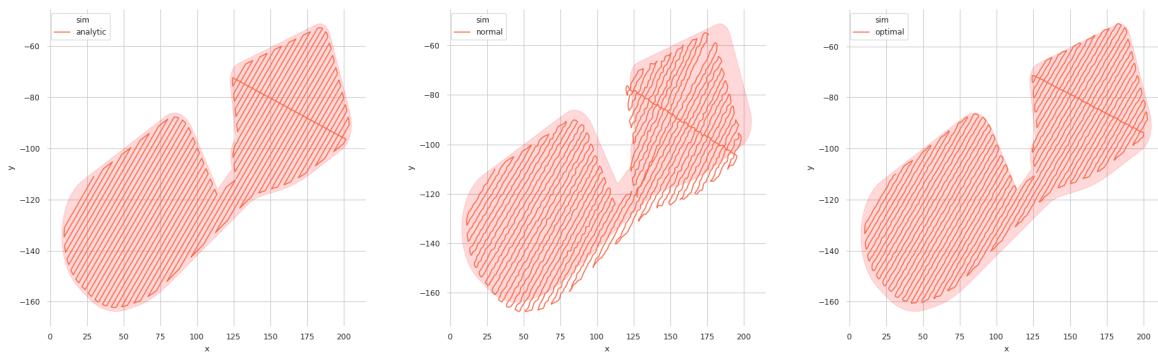


FIGURE 5.8: CPP WITH/WITHOUT KALMAN FILTERING

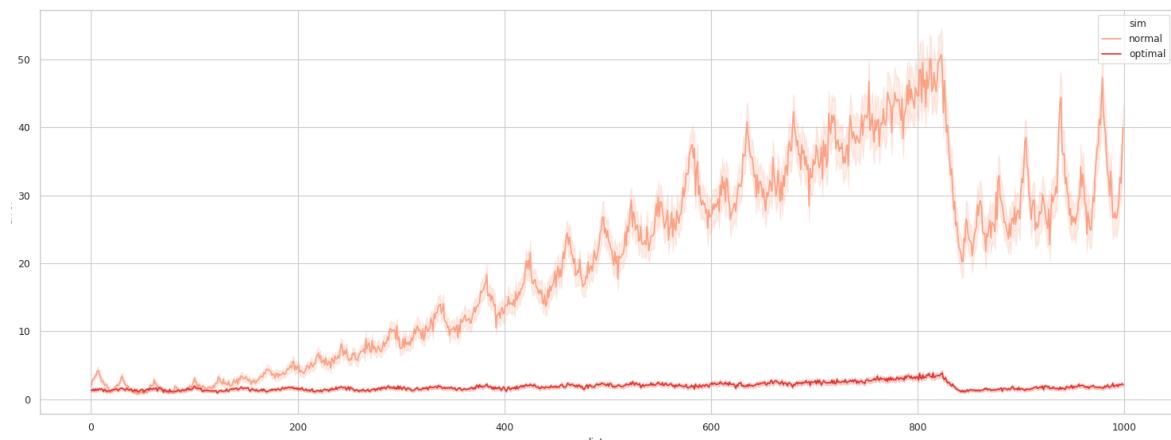


FIGURE 5.9: NEES OF THE CONTROLLER

,

6 CHAPTER

NOMENCLATURE

SIGN	DESCRIPTION	UNIT	PAGE
a	acceleration	m/s^2	13
A_a	working section of the screw conveyor	m^2	52, 53
A_p	cross section of a pipe	m^2	46–48
A_s	cross section area of in soil submerged volume	m^2	46, 51
a_x	acceleration along the x-axis	m/s^2	13
a_y	acceleration along the y-axis	m/s^2	13
a_z	acceleration along the z-axis	m/s^2	13, 20
$a_{z,k}$	acceleration along the z-axis at time k	m/s^2	20, 22, 64
$a_{z,k-1}$	acceleration along the z-axis at time k-1	m/s^2	20
α_e	attenuation	dB/m	8, 9
$\alpha_s f$	skin friction coefficient	–	52
B_d	slow changing component of the signal; this is the gyroscope drift	rad/s	14, 15
\mathbf{B}	a control-input model which is applied to a control vector	–	19, 20
β	phase factor of a wave	–	8, 9
C	cost a function	–	26
c	singel cell	–	30
c_{sb}	cohesion of a soil bed	Pa	46, 49, 50, 52, 53
c_t	volumetric concentration	–	55
c_u	undrained shear strength of soil	Pa	52, 53
C_{vb}	volumetric bed concentration	–	49
C_p	critical point in graph	–	27
C_{p0}	critical point 0 in graph	–	27
C_{p1}	critical point 1 in graph	–	26, 28
C_{p2}	critical point 2 in graph	–	26–28

SIGN	DESCRIPTION	UNIT	PAGE
Cp_3	critical point 3 in graph	–	27, 28
Cp_i	critical point ith in graph	–	27
d_a	sinkage depth of a Archimedes screw	m	46, 49–53
$D_{l,t}$	list of line segments	–	30
$D_{l,t}$	line segment at t	–	30
$D_{l,t-1}$	line segment at t-1	–	30
d_p	pipe diameter	m	49, 53, 54
$d_i(x)$	distance between x points and obstacle i	m	27
$\Delta d_{1,2}$	distance between two points	m	9
ΔF	difference in force	N	46
ΔL	length of a pipe	m	49, 54
Δp	pressureloss	Pa	17, 45–48
Δp_n	nominal pressureloss	Pa	46–48
δ_s	range of a sensor-detector-range	m	28, 29
Δt	time difference	s	20–22
$\Delta \tau$	torque difference	N m	45, 46
Δx	distance between nodes	m	30, 31
Δz	height difference	m	54
\hat{E}	amplitude of the electric field wave	V/m	8, 9
E_x	Is defined mathematically as a vector field that associates to each point in space the (electrostatic or Coulomb) force per unit of charge exerted on an infinitesimal positive test charge at rest at that point.	V/m	8, 9
ϵ_b	resistance coefficient of the bends, valves, hoses and other obstructions	–	54
ϵ_e	permittivity	–	8, 9
$\epsilon_{N,k}$	NEES at time k	–	23
$\bar{\epsilon}_N$	mean NEES value	–	23
ϵ_s	resistance coefficient of the suction inlet	–	54
$\eta_{V,n}$	nominal Volumetric efficiency	–	46–48

SIGN	DESCRIPTION	UNIT	PAGE
\exp	Eulers number	–	9
F	force	N	13
F_b	buoyancy force of a submerged Archimedes screw	N	46, 53
$F_{cr,b}$	buoyancy force of a submerged crawler	N	52
\mathbf{F}	a state transition model which is applied to the previous state	–	19–21
F_n	normal force working on an Archimedes screw	N	49
F_r	Froude number for a pipeline	–	55
F_{rxd}	Froude number of the grains	–	55
F_s	skin friction	N	52
F_{sn}	specific normal force working on an Archimedes screw	N/kg	46, 49, 50
F_u	ultimate skin friction	N	52
g	standard gravity model	m^2/s	17, 20, 46, 49, 52–54
\mathbf{G}	adjacency graph	–	26, 31–33
\mathbf{G}_1	adjacency subgraph 1	–	26
\mathbf{G}_2	adjacency subgraph 2	–	26
γ_e	propagation constant	m	8, 9
γ_m	specific weight of diluted water during dredging	N/m^3	17
γ_{sw}	specific weight of submerged soil	N/m^3	17, 46, 49, 50, 53
γ_w	specific weight of water	N/m^3	17
h	height difference between radius of screw and its submerged part	m	51
h_{dh}	height of a dredge head	m	53
\hat{H}	amplitude of the magnetic field wave	A	8, 9
\mathbf{H}	measurement sensitivity matrix defining the linear relationship between state of the dynamic system and measurements that can be made	–	19–22
H_y	is a vector field that describes the magnetic influence of electric charges in relative motion and magnetized materials	A/m	8, 9
I	moment of inertia	kg m^2	46
i	imaginary unit	–	9

SIGN	DESCRIPTION	UNIT	PAGE
K_{HP}	Hagen-Poiseuille coefficient for laminar pipe flows	–	46-48
K_{TP}	friction torque vs pressure gain coefficient parameter	–	46, 48
\bar{K}	Kalman gain matrix	–	20-22
L	Distance between the crawlers screws	m	56, 57
l_a	length of an Archimedes screw	m	46, 51, 53
$L_{\alpha,\epsilon}$	electromagnetic signal strength	dB	9
λ	friction factor of a straight pipe	–	54
m	mass	kg	13
m_{cr}	mass of a crawler	kg	46, 52
m'_{cr}	total buoyancy corrected mass of a crawler	kg	52
m_f	mass	kg	46
\vec{m}_{hi}	hard iron adjusted vector	T	15, 16
m_n	number of neighbouring neurons	–	38
\vec{m}_{si}	soft iron adjusted vector	T	16
\vec{m}	raw magnetometer vector	T	15, 16
μ	dynamic viscosity	Pas	46-48, 54
μ_e	electromagnetism permeability	H/m	8, 9
μ_n	nominal dynamic viscosity	Pas	46-48
$n_{a,l}$	neural activity of a the lth neuron	–	37, 38
N_c	dimensionless constant used in the Brinch-Hansen equation, related to the cohesion	–	46, 50
N_γ	dimensionless constant used in the Brinch-Hansen equation, related to the specific weight of the soil	–	46, 50, 51
N_ϕ	dimensionless factor of the internal friction angle	–	53
N_q	dimensionless constant used in the Brinch-Hansen equation, related to the load of the surrounding soil	–	46, 50, 51
n_s	stochastic component of a signal	rad/s	14
n_x	number of elements in the state vector	–	23
∇_m	normal of a surface	m	26
ω	angular velocity	rad/s	9, 14, 42, 45-48, 56

SIGN	DESCRIPTION	UNIT	PAGE
ω_n	nominal angular velocity	rad/s	46-48
ω_t	true angular velocity	rad/s	14
ω_{th}	angular velocity threshold	rad/s	46, 48
P	power	W	46
p	pressure	Pa	17, 45, 48
\mathbf{P}_0	initialization covariance matrix of state estimation uncertainty	–	20
p_a	atmospheric pressure	Pa	17
p_{atm}	atmospheric pressure	Pa	55
p_i	pressureloss at the suction inlet	Pa	54, 55
\mathbf{P}_k	covariance matrix of state estimation uncertainty	–	20-23
\vec{p}_k	pose components of the state vector xk	m	38, 43
\mathbf{P}_{k-1}	covariance matrix of a priori state estimation uncertainty	–	20, 21
\vec{p}_{k+1}	AUV location at time k + 1	m	38
\vec{p}_{k+1}	pose components of the state vector xk+1	m	43
p_{man}	manometric head of the dredge pump	Pa	55
p_p	absolute pressure at the outlet of the dredge pump	Pa	55
p_q	pressure generated at a certain depth due to the soil on top of it	Pa	46, 50
p_{ro}	resistance offered by bends, valves, hoses and other obstructions	Pa	54, 55
p_{rp}	resistance of the straightline pipe	Pa	54, 55
p_s	absolute pressure at the entrance of impeller	Pa	55
p_{sb}	load working on a soil bed	Pa	46, 49, 50
p_{sm}	static head in the suction pipe	Pa	54, 55
p_{ss}	static pressure generated by surrounding water	Pa	55
p_v	pressure of water vapor in ambient air	Pa	54, 55
ϕ_c	roll of the crawler on the Euclidean y-axis	rad	42, 43
ϕ_{IMU}	roll of the IMU on the Euclidean y-axis	rad	13, 15
ϕ_s	angle of internal friction	rad	46, 50, 51, 53

SIGN	DESCRIPTION	UNIT	PAGE
ψ	yaw of the crawler on the Euclidean z-axis	rad	54, 56, 57
ψ_c	yaw of the crawler bot on the Euclidean z-axis	rad	42, 43
ψ_{IMU}	yaw of the IMU bot on the Euclidean z-axis	rad	13, 15
ψ_m	Durand-Condolios mixture correction	–	54, 55
Q	volumetric fluid flow	m^3/s	46–48, 53–55
\vec{q}	heat flow	–	43
Q_i	ideal volumetric fluid flow	m^3/s	46–48
\mathbf{Q}_k	covariance matrix of process estimation uncertainty	–	19–21
Q_l	volumetric fluid leakage flow	m^3/s	46–48
q_s	real component of a quaternion	–	43
q_{si}	smallest magnitude of a point on the ellipse, and thus the vector of the B-axis	T	16
q_x	imaginary x-axis of a quaternion	–	43
q_y	imaginary y-axis of a quaternion	–	43
q_z	imaginary z-axis of a quaternion	–	43
r	radius	m	28
r_0	the receptive field radius of the kth neuron	m	36, 37
r_a	radius of Archimedes screw	m	46, 50–52
\mathbf{R}	covariance matrix of state estimation uncertainty	–	19–21
r_{si}	greatest magnitude of a point on the ellipse, and thus the vector of the A-axis	T	16
R_t	radius from ICR ro COG of crawler	m	56
Re	Reynolds Number	–	54
Re_{sd}	relative submerged density	–	49
ρ	density of a material	kg/m^3	46–48, 55
ρ_{ds}	density of drained soil	kg/m^3	46, 49
ρ_{is}	in-situ density of soil	kg/m^3	46, 49, 51
ρ_m	density of a mixture	kg/m^3	17, 54, 55
ρ_n	nominal density of a material	kg/m^3	46–48

SIGN	DESCRIPTION	UNIT	PAGE
ρ_w	density of water	kg/m ³	17, 46, 49, 51, 54, 55
S	a movement in a graph	–	26
s_a	soil contact arc length	m	46, 51–53
s_c	shape factor used in the Brinch-Hansen equation, related to the cohesion	–	46, 50, 51
s_γ	shape factor used in the Brinch-Hansen equation, related to the specific weight of the soil	m	46, 50, 51
s_q	shape factor used in the Brinch-Hansen equation, related to the surrounding load of the soil	–	46, 50, 51
s_z	position along the z-axis	m	19–21, 23
$s_{z,k}$	position along the z-axis at time k	m	20
$s_{z,k-1}$	position along the z-axis at time k-1	m	20, 22
$s_{z,m}$	measured position along the z-axis	m	21, 22
$s_{z,m,k}$	measured position along the z-axis at time k	m	21, 22
σ_e	Electrical conductivity	S/m	8–11
σ_H	horizontal stress	Pa	50, 53
σ'	effective stress	Pa	52
σ_s	deviation on position	m	21, 22
$\sigma_{s,m}$	deviation on measured position	m	21, 22
σ_V	vertical stress	Pa	50, 53
σ_v	deviation on velocity	m/s	21, 22
$\sigma_{v,m}$	deviation measured on velocity	m/s	21, 22
T	temperature	K	14, 15, 17
t	time	s	9, 14, 23, 30, 42
T'	rate of temperature variation	K/s	14, 15
τ	torque	N m	42, 45–48
τ_0	initial torque	N m	46, 48
τ_f	torque due to friction	N m	46–48
τ_i	ideal torque	N m	46–48
θ	angle on the x-axis	rad	14, 55

SIGN	DESCRIPTION	UNIT	PAGE
θ_b	pitch of the dredge bot on the Euclidean z-axis	rad	43
θ_c	pitch of the crawler on the Euclidean x-axis	rad	42, 43
θ_{IMU}	pitch of the IMU on the Euclidean x-axis	rad	13, 15
θ_s	sink angle of an Archimedes screw	rad	46, 49–52
θ_{si}	soft iron axis offset along the x-axis	rad	16
\vec{u}	unit vector	–	25, 42
\vec{u}_k	control inputs	–	19, 20, 41, 43
v	specific volume	m^3/kg	56
v_f	velocity of a fluid at a certain point	m/s	46–48, 53, 54
\vec{v}_k	measurement noise	–	19–21, 53, 55
v_L	translational velocity of left Archimedes screw	m/s	56, 57
v_R	translational velocity of right Archimedes screw	m/s	56, 57
V_{sa}	submerged volume of a Archimedes screw	m^3	46, 51, 53
v_z	velocity along the z-axis	m/s	19–21, 23
$v_{z,k}$	velocity along the z-axis at time k	m/s	20
$v_{z,k-1}$	velocity along the z-axis at time k-1	m/s	20, 22
w_a	width of the Archimedes screw in contact with the soil	m	46, 50, 51
w_{dh}	width of a dredge head	m	53
\vec{w}_k	process noise	–	19, 20
x	location along the x-axis	m	27, 36, 42, 43, 57
x_h	hard iron distortion along the x-axis	T	15
\vec{x}_k	state vector describing the state of a system at the kth component of x	–	11, 19–22, 42, 43
\vec{x}_0	state vector describing the initial state of a system at the kth component of x	–	20
$\vec{x}_{g,k}$	ground truth state, describing the real state of a system kth component of x	–	22
\hat{x}_k	an estimation of the state vector x	–	19–22
\vec{x}_{k+1}	a state estimate of x, conditioned on all available measurements at time tk	–	41
\vec{x}_{k-1}	a priori state of xk, conditioned on all prior measurements, except the one at time tk	–	19, 20

SIGN	DESCRIPTION	UNIT	PAGE
\tilde{x}_k	Error of the state of a system at the kth component of x	–	22, 23
x_m	raw magnetometer value along the x-axis	T	15, 16
y	location along the y-axis	m	36, 42, 43, 57
y_1	y-index of greatest magnetic magnitude	T	16
y_h	hard iron distortion along the y-axis	T	15
\vec{y}_k	measured values	–	20, 21, 42, 64-66
\vec{y}'_k	transformed measured values	–	64-66
y_l	a gls-monotonically increasing function of the difference between the next moving directions	m	38
y_m	raw magnetometer value along the y-axis	T	15, 16
z	height	m	17, 42, 43
z_ϵ	height error	m	17
\vec{z}_k	measured values mapped to the state space	–	19-22
z_m	raw magnetometer value along the z-axis	T	15, 16
z_p	height of the pressure sensor with regards to the soil bed	m	17

CHAPTER 7

GLOSSARY

KEY	DESCRIPTION	PAGE
accelerometer	a device that measures proper acceleration	11–13, 15, 18, 19, 41, 61, 64
time of flight acoustic navigation	triangulation of a position using the difference in send and receive time of signal, to calculate the distance from a source	18
adjacency graph	a graph representing depicting all the nodes	26, 28
Archimedes screw	a machine historically used for transferring water from a low-lying body of water into irrigation ditches. The same principle can also be used propel the screw in a medium	41
bandwidth	a difference between the upper and lower frequencies in a continuous set of frequencies	10
bathymetric map	submerged equivalent of an above-water topographic map	19
bitstream	a sequential binary sequence	5
Chrono Sensors	an extension library for Project Chrono, which allows simulation of realistic sensor values.	66
Coriolis effect	an inertial or fictitious force that acts on objects that are in motion within a frame of reference that rotates with respect to an inertial frame.	14
coverage path	a sequence of steps which covers a whole area by following a certain path	24–26, 28, 29, 34
critical point	a value of average degree, which separates networks	25–28
cusp point	are points where its surface normal of the boundary of the free configuration space is non-smooth	28
deadlock state	a standstill situation, from which the algorithm has no means of escape	35
dead-reckoning	the process of calculating one's current position by using a previously determined position, or fix, and advancing that position based upon known or estimated speeds over elapsed time and course theory of belief functions, also referred to as evidence theory or Dempster-Shafer theory (DST),	13, 18, 28
Dempster Shafer theory	evidence theory or Dempster-Shafer theory (DST), is a general framework for reasoning with uncertainty	38
dilatancy	volume increase that may occur during shear	49
draghead	a suction mouth which is dragged across a water body	3, 53
dredgeline	a pipeline which transports excavated slurry	1, 5, 6, 53, 66

KEY	DESCRIPTION	PAGE
electric field	a vector field that associates to each point in space the Coulomb force that would be experienced per unit of electric charge, by an infinitesimal test charge at that point. Electric fields converge and diverge at electric charges and can be induced by time-varying magnetic fields	8
erosion	an action of surface processes (such as water flow or wind) that removes soil	1
geophysical navigation	navigation using landmarks	18, 19
gimbal lock	the loss of one degree of freedom in a three-dimensional space, three-gimbal mechanism that occurs when the axes of two of the three gimbals are driven into a parallel configuration, "locking" the system into rotation in a degenerate two-dimensional space	43
gyroscope	a spinning wheel or disc in which the axis of rotation is free to assume any orientation by itself. When rotating, the orientation of this axis is unaffected by tilting or rotation of the mounting, according to the conservation of angular momentum. Because of this, gyroscopes are useful for measuring or maintaining orientation	11, 12, 14, 18, 19, 41, 61, 64
hard iron	a constant additive disturbance in the magnetic field of the magnetometer	15, 16, 64
hopper	a storage container or compartment	2, 3
Kalman filter	an algorithm that uses a series of measurements observed over time, containing statistical noise and other inaccuracies, and produces estimates of unknown variables that tend to be more precise than those based on a single measurement alone	5, 11, 13, 19, 21–23, 41
Kalman gain	the relative weight given to the measurements and current state estimate, and can be "tuned" to achieve particular performance. With a high gain, the filter places more weight on the most recent measurements, and thus follows them more responsively. With a low gain, the filter follows the model predictions more closely. At the extremes, a high gain close to one will result in a more jumpy estimated trajectory, while low gain close to zero will smooth out noise but decrease the responsiveness	21
LoRa	a wireless technology that has been developed to enable low data rate communications to be made over long distances by sensors and actuators for M2M and Internet of Things, IoT applications	10
Lorentz force	the combination of electric and magnetic force on a point charge due to electromagnetic fields	15
magnetic field	a magnetic effect of electric currents and magnetic materials. The magnetic field at any given point is specified by both a direction and a magnitude (or strength); as such it's a vector field	8

KEY	DESCRIPTION	PAGE
magnetometer	a magnetometer is an instrument that measures magnetism, either magnetization of magnetic material like a ferromagnet, or the strength and, in some cases, direction of the magnetic field at a point in space	11, 12, 15, 19, 41, 61, 64
Maxwell's equation	are a set of partial differential equations that, together with the Lorentz force law, form the foundation of classical electrodynamics, classical optics, and electric circuits	8
monotonically	a function between ordered sets that preserves or reverses the given order	37, 38
Morse function	a function for which all critical points are non-degenerate and all critical levels are different	25, 26
NARROW-cell	a cell is located in a narrow space, bound between multiple walls	28
non-commutative	a operation in which the order of the operands determine the results	43
odometry	the use of data from motion sensors to estimate change in position over time	13
off-line	algorithm which plans an optimal path ahead of time, thus which needs to know the environment a priori	19, 24, 26, 40
ohCaptain	a versatile efficient controller framework for the maritime industry	66
on-line	an algorithm which has the ability to adapt when needed	24, 30, 40, 61
optimal path	a sequence of steps which are optimized	23, 25, 26
polarized plane	is a confinement of the electric field vector or magnetic field vector to a given plane along the direction of propagation	8
pressure sensor	can be classified in terms of pressure ranges they measure, temperature ranges of operation, and most importantly the type of pressure they measure	11, 17–19, 41, 42, 61, 65
Project Chrono	a physics-based modelling and simulation infrastructure based on a platform-independent open-source design implemented in C++	63, 64, 66
quaternion	a number system that extends the complex numbers, they are very useful in describing a rotation involving three dimensions	19, 43
Reeb graph	a mathematical object reflecting the evolution of the level sets of a real-valued function on a manifold	28
sedimentation	the opposite of erosion	1
silt	a granular material of a size between sand and clay	1
slurry	describe a mixture that consist of both solid and fluid phases	2, 3, 54
soft iron	a result of material that distorts the magnetic field of magnetometer, but does not necessarily generate its own magnetic field	16, 64
Trapezoidal rule	a technique for approximating the definite integral	14

KEY	DESCRIPTION	PAGE
traveling salesman	asks the following question: "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?" it's an NP-hard problem in combinatorial optimization, important in operations research and theoretical computer science	25
umbilical	a electronic cable connecting an underwater vehicle	1, 5, 6, 30, 66, 74
VAST-cell	a cell located in a vast open space	28
Voronoi diagram	a partition of a plane into regions close to each of a given set of objects	24
word	a string of bits representing a value which is stored in memory	15

CHAPTER ACRONYMS 8

KEY	DESCRIPTION	PAGE
ADC	Analog Digital Conversion	13, 15
ASD	Auger Suction Dredger	3
AUV	Autonomous Underwater Vehicle	6, 18, 43
AW	Acoustic Waves	11
BBB	Beagle Bone Black	57, 63
BCD	Boustrophedon Cellular Decomposition	25, 26, 28, 61
BCD-GVD	Morse-based Cellular Decomposition with Voronoi Diagram	40
CIP	Common Industrial Protocol	7
CPP	Coverage Path Planning	5, 24, 40, 41, 61
CSD	Cutter Suction Dredger	3
DEM	Discrete Element Method	63
EMW	Electromagnetic Waves	7–9, 11, 18
GPS	Global Positioning System	5, 11, 13, 18, 41
GUI	Graphical User Interface	57
GVD	Generalized Voronoi Diagram	28
IEEE	Institute of Electrical and Electronics Engineers	7, 9, 10
IHC	Royal IHC	57
IMU	Inertial Measurement Unit	12, 58
IoT	Internet of Things	10
LBL	Long Base Line	18
LLC	Logical Link Control	7, 9
LQE	Linear Quadratic Estimation	19
LQG-MP	Linear-Quadratic Gaussian Motion Planning	19
MAC	Media Access Control	7, 9

KEY	DESCRIPTION	PAGE
MEMS	Micro Electro Mechanical System	13-15
MLP	Multi-Layer Perceptron	33
MTI	IHC MTI B.V.	17, 57
NEES	Normalized Estimated Error Squared	22, 23
NN	Neural Network based Coverage in Grid maps	40
onBCD	online Morse-based Boustrophedon Cellular Decomposition	40
onSD2	on-line Slice Decomposition II	40
onSD2NN	on-line Slice Decomposition II with Neural Network	40
OOP	Object-Orientated Programming	63
PID	Proportional Integral Derivative	41, 57, 62, 67
PLC	Programmable Logic Controller	57
RESL	Robotic Embedded Systems Laboratory	10
ROV	Remote Operated Vehicle	6
Rpi	Raspberry pi	57
RRT	Rapidly exploring Random Trees	19
SBC	Single Board Computer	57, 63
SCM	Soil Contact Model	63, 66
SLAM	Simultaneous Localization And Mapping	19
Spiral-STC	Spiral Spanning Tree Coverage	35, 36
STC	grid based Spanning Tree Coverage	40
SW	Sound Waves	11
SWR	Submerged Weight Range	44
TCP	Transmission Control Protocol	7
TIR	Total Internal Reflection	5
topBCD	topological Boustrophedon Cellular Decomposition	40
TRN	Terrain Relative navigation	19
TSHD	Trailing Suction Hopper Dredger	3

KEY	DESCRIPTION	PAGE
UDP	User Datagram Protocol	7
UKF	Unscented Kalman Filter	41, 43, 61
USBL	Ultra Short Base Line	18
UT	Unscented Transformation	41
YAML	YAML Ain't Markup Language	62

9 CHAPTER BIBLIOGRAPHY

- [1] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve", *The Journal of physiology*, vol. 117, no. 4, p. 500, 1952. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pmc/articles/pmc1392413/> (visited on 05/10/2016).
- [2] R. E. Kalman, "A new approach to linear filtering and prediction problems", *Journal of basic Engineering*, vol. 82, no. 1, pp. 35–45, 1960. [Online]. Available: <http://fluidsengineering.asmedigitalcollection.asme.org/article.aspx?articleid=1430402> (visited on 01/02/2017).
- [3] I. S. Bogie, "Conduction and magnetic signalling in the sea a background review", *Radio and Electronic Engineer*, vol. 42, no. 10, pp. 447–452, 1972. [Online]. Available: <http://digital-library.theiet.org/content/journals/10.1049/ree.1972.0076> (visited on 08/28/2016).
- [4] International Standard ISO, NEN-ISO 7119:1982 *Continuous mechanical handling equipment for loose bulk materials - screw conveyors - Design rules of drive power*. International Standard ISO, 1981.
- [5] A. Westneat, D. Blidberg, and R. Corell, "Advances in unmanned untethered underwater vehicles", *Unmanned Systems*, vol. 1, no. 3, pp. 8–13, 1983.
- [6] Z. L. Cao, Y. Huang, and E. L. Hall, "Region filling operations with random obstacle avoidance for mobile robots", en, *Journal of Robotic Systems*, vol. 5, no. 2, pp. 87–102, Apr. 1988, issn: 1097-4563. doi: 10.1002/rob.4620050202. [Online]. Available: <http://onlinelibrary.wiley.com.wiley.stcproxy.han.nl/doi/10.1002/rob.4620050202/abstract> (visited on 04/19/2016).
- [7] J. Borenstein, *Where am I? Sensors and methods for mobile robot positioning*. Apr. 1996.
- [8] K. P. Valavanis, D. Gracanin, M. Matijasevic, R. Kolluru, and G. A. Demetriou, "Control architectures for autonomous underwater vehicles", *IEEE Control Systems*, vol. 17, no. 6, Dec. 1997, issn: 1066-033X. doi: 10.1109/37.642974.
- [9] H. Choset and P. Pignon, "Coverage Path Planning: The Bousspheredon Cellular Decomposition", en, in *Field and Service Robotics*, A. Zelinsky, Ed., London: Springer London, 1998, pp. 203–209, isbn: 978-1-4471-1275-4. [Online]. Available: http://link.springer.com/10.1007/978-1-4471-1273-0_32 (visited on 04/17/2016).
- [10] Joseph C. Palais, *Fibre Optic Communication*. Prentice Hall, 1998.
- [11] S. Thrun, "Learning metric-topological maps for indoor mobile robot navigation", *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, Feb. 1998, issn: 0004-3702. doi: 10.1016/S0004-3702(97)00078-7. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0004370297000787> (visited on 04/17/2016).
- [12] VBKO Vereniging van waterbouwers in bagger-, kust- en oeverwerken, *Voortgezette Opleiding Uitvoering Baggerwerken*, nl. Leidschendam: VBKO Vereniging van waterbouwers in bagger-, kust- en oeverwerken, 1998, isbn: 90-90-11108-5.
- [13] C. K. Chui and G. Chen, *Kalman Filtering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999, isbn: 978-3-540-64611-2. [Online]. Available: <http://link.springer.com/10.1007/978-3-662-03859-8> (visited on 12/31/2016).
- [14] J. G. Webster, *The Measurement, Instrumentation, and Sensors: Handbook*, en. Springer Science & Business Media, 1999, Google-Books-ID: b7UuZzf9ivIC, isbn: 978-3-540-64830-7.
- [15] H. Choset, "Coverage of Known Spaces: The Bousspheredon Cellular Decomposition", *Autonomous Robots*, vol. 9, no. 3, pp. 247–253, 2000, issn: 09295593. doi: 10.1023/A:1008958800904. [Online]. Available: <http://link.springer.com/10.1023/A:1008958800904> (visited on 04/17/2016).

- [16] H. Choset, E. Acar, A. A. Rizzi, and J. Luntz, "Exact cellular decompositions in terms of critical points of morse functions", in *Robotics and Automation, 2000. Proceedings. ICRA'00. IEEE International Conference on*, vol. 3, IEEE, 2000, pp. 2270–2277. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=846365 (visited on 04/19/2016).
- [17] L. L. Whitcomb, "Underwater robotics: Out of the research laboratory and into the field", in *IEEE International Conference on Robotics and Automation, 2000. Proceedings. ICRA '00*, vol. 1, 2000. doi: 10.1109/ROBOT.2000.844135.
- [18] E. U. Acar, H. Choset, and P. N. Atkar, "Complete sensor-based coverage with extended-range detectors: A hierarchical decomposition in terms of critical points and voronoi diagrams", in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 3, IEEE, 2001, pp. 1305–1311. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=977163 (visited on 04/19/2016).
- [19] H. Choset, "Coverage for robotics – A survey of recent results", *Annals of Mathematics and Artificial Intelligence*, vol. 31, no. 1/4, pp. 113–126, 2001, issn: 10122443. doi: 10.1023/A:1016639210559. [Online]. Available: <http://link.springer.com/10.1023/A:1016639210559> (visited on 04/17/2016).
- [20] W. H. Huang, "Optimal line-sweep-based decompositions for coverage algorithms", in *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, vol. 1, IEEE, 2001, pp. 27–32. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=932525 (visited on 04/19/2016).
- [21] E. U. Acar and H. Choset, "Sensor-based coverage of unknown environments: Incremental construction of morse decompositions", *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 345–366, 2002. [Online]. Available: <http://ijr.sagepub.com/content/21/4/345.short> (visited on 04/19/2016).
- [22] E. U. Acar, H. Choset, A. A. Rizzi, P. N. Atkar, and D. Hull, "Morse decompositions for coverage tasks", *The International Journal of Robotics Research*, vol. 21, no. 4, pp. 331–344, 2002. [Online]. Available: <http://ijr.sagepub.com/content/21/4/331.short> (visited on 04/19/2016).
- [23] C. Luo, S. X. Yang, D. A. Stacey, and J. C. Jofriet, "A solution to vicinity problem of obstacles in complete coverage path planning", in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, vol. 1, IEEE, 2002, pp. 612–617. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1013426 (visited on 05/09/2016).
- [24] B. G. Liptak, *Instrument Engineers' Handbook, Fourth Edition, Volume One: Process Measurement and Analysis*. CRC Press, Jun. 27, 2003, 1914 pp., isbn: 978-1-4200-6402-5.
- [25] Z. Feng and R. Allen, "Evaluation of the effects of the communication cable on the dynamics of an underwater flight vehicle", *Ocean Engineering*, vol. 31, Jun. 2004, issn: 0029-8018. doi: 10.1016/j.oceaneng.2003.11.001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S002980180400006X>.
- [26] Y. Mei, Y.-H. Lu, Y. C. Hu, and C. G. Lee, "Energy-efficient motion planning for mobile robots", in *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, vol. 5, IEEE, 2004, pp. 4344–4349. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=1302401 (visited on 05/09/2016).
- [27] S. C. Wong and B. A. MacDonald, "Complete coverage by mobile robots using slice decomposition based on natural landmarks", in *PRICAI 2004: Trends in Artificial Intelligence*, Springer, 2004, pp. 683–692. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-540-28633-2_72 (visited on 04/19/2016).
- [28] I. F. Akyildiz, D. Pompili, and T. Melodia, "Underwater acoustic sensor networks: Research challenges", *Ad Hoc Networks*, vol. 3, no. 3, pp. 257–279, May 2005, issn: 1570-8705. doi: 10.1016/j.adhoc.2005.01.004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1570870505000168> (visited on 08/26/2016).
- [29] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, en. MIT Press, 2005, isbn: 978-0-262-03327-5.
- [30] *Marine aqua delat*, in collab. with Google maps, Satelite, 51°40'17.52"N, 4° 4'51.06"E, 2005.

- [31] Edward Tucholski, *Underwater Acoustics and Sonar - SP411*. 2006.
- [32] F. J. L. J. S. Wei, J. C. Fang, and J. L. Li, "Improved temperature error model of silicon MEMS gyroscope with inside frame driving [J]", *Journal of Beijing University of Aeronautics and Astronautics*, vol. 11, p. 004, 2006. [Online]. Available: http://en.cnki.com.cn/Article_en/CJFDTotal-BJHK200611004.htm (visited on 12/28/2016).
- [33] g. Welch and G. Bishop, "An introduction to the Kalman Filter", en, Jul. 2006.
- [34] S. Wong, "Qualitative topological coverage of unknown environments by mobile robots", PhD thesis, ResearchSpace Auckland, 2006. [Online]. Available: <https://researchspace.auckland.ac.nz/handle/2292/619> (visited on 04/25/2016).
- [35] M.-C. Fang, C.-S. Hou, and J.-H. Luo, "On the motions of the underwater remotely operated vehicle with the umbilical cable effect", *Ocean Engineering*, vol. 34, Jun. 2007, issn: 0029-8018. doi: 10.1016/j.oceaneng.2006.04.014. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0029801806001880>.
- [36] D. Green and R. Perry, *Perry's Chemical Engineers' Handbook, Eighth Edition*, ser. McGraw Hill professional. McGraw-Hill Education, 2007, isbn: 978-0-07-159313-7. [Online]. Available: <https://books.google.nl/books?id=tH7IVcA-MX0C>.
- [37] L. Nicolaescu, *An Invitation to Morse Theory*, en, ser. Universitext. New York, NY: Springer New York, 2007, isbn: 978-0-387-49509-5. [Online]. Available: <http://link.springer.com/10.1007/978-0-387-49510-1> (visited on 04/19/2016).
- [38] L. Tetley and D. Calcutt, *Electronic Navigation Systems*, en. Routledge, Jun. 2007, Google-Books-ID: mybjgp2gGRsC, isbn: 978-1-136-40724-6.
- [39] A. Verruijt and S. Van Baars, *Soil mechanics*. VSSD, 2007. (visited on 05/30/2016).
- [40] C. Konvalin, *Technical document: Compensating for tilt, hard iron and soft iron effects*, EN, 2008. [Online]. Available: <http://www.sensorsmag.com/sensors/motion-velocity-displacement/compensating-tilt-hard-iron-and-soft-iron-effects-6475> (visited on 12/30/2016).
- [41] L. Lanbo, Z. Shengli, and C. Jun-Hong, "Prospects and problems of wireless communication for underwater sensor networks", *Wireless Communications and Mobile Computing*, no. 8, 2008. [Online]. Available: <http://onlinelibrary.wiley.com/doi/10.1002/wcm.654/abstract> (visited on 08/27/2016).
- [42] C. Luo and S. X. Yang, "A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments", *Neural Networks, IEEE Transactions on*, vol. 19, no. 7, pp. 1279–1298, 2008. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4539807 (visited on 05/16/2016).
- [43] Training Institute for Dredging, *Ingewijden Training*, en. MTI Holland, 2008.
- [44] A. Bahr, J. J. Leonard, and M. F. Fallon, "Cooperative localization for autonomous underwater vehicles", *The International Journal of Robotics Research*, vol. 28, no. 6, pp. 714–728, 2009. (visited on 05/20/2016).
- [45] Hagman, Elias, "Design of a High Speed, Short Range Underwater Communication System", PhD thesis, Eidgenossische Technische Hochschule Zurich, 2009.
- [46] R. Lotman, "Applicable theory for the modeling of the propulsion system of the SMT", Dec. 9, 2009.
- [47] A. van der Zee, "Prediction of the terramechanic performance of a SMT", IHC Dredgers B.V., Dec. 18, 2009, p. 113.
- [48] M. Ainslie, *Principles of Sonar Performance Modelling*, en. Springer Science & Business Media, Sep. 2010, isbn: 978-3-540-87662-5.
- [49] S. Tully, G. Kantor, and H. Choset, "Leap-frog path design for multi-robot cooperative localization", in *Field and service robotics*, Springer, 2010, pp. 307–317. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-642-13408-1_28 (visited on 04/18/2016).
- [50] M. Garcia, S. Sendra, M. Atenas, and J. Lloret, "Underwater wireless ad-hoc networks: A survey", *Mobile ad hoc networks: Current status and future trends*, 2011. (visited on 08/27/2016).

- [51] s. Jiang and S. Georgakopoulos, "Electromagnetic Wave Propagation into Fresh Water", en, Apr. 2011.
- [52] C. Kownacki, "Optimization approach to adapt Kalman filters for the real-time application of accelerometer and gyroscope signals' filtering", *Digital Signal Processing*, vol. 21, no. 1, pp. 131–140, Jan. 2011, issn: 1051-2004. doi: 10 . 1016 / j . dsp . 2010 . 09 . 001. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1051200410001843> (visited on 12/29/2016).
- [53] T.-K. Lee, S.-H. Baek, Y.-H. Choi, and S.-Y. Oh, "Smooth coverage path planning and control of mobile robots based on high-resolution grid map representation", *Robotics and Autonomous Systems*, vol. 59, no. 10, pp. 801–812, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0921889011000996> (visited on 05/10/2016).
- [54] R. Lotman and M. A. Grima, "Deep Sea Mining With an Archimedes Screw Driven Vehicle", in *ASME 2011 30th International Conference on Ocean, Offshore and Arctic Engineering*, American Society of Mechanical Engineers, 2011.
- [55] J. R. Nistler and M. F. Selekwa, "Gravity compensation in accelerometer measurements for robot navigation on inclined surfaces", *Procedia Computer Science*, Complex adaptive sysytems, vol. 6, pp. 413–418, Jan. 2011, issn: 1877-0509. doi: 10 . 1016 / j . procs . 2011 . 08 . 077. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1877050911005424> (visited on 12/27/2016).
- [56] R. A. Rajapakse, *Geotechnical Engineering Calculations and Rules of Thumb*. Butterworth-Heinemann, Apr. 2011, isbn: 978-0-08-055903-2.
- [57] F. M. White, *Fluid Mechanics*, en. McGraw Hill, 2011, isbn: 978-0-07-352934-9.
- [58] M. C. Domingo, "An overview of the internet of underwater things", *Journal of Network and Computer Applications*, vol. 35, no. 6, pp. 1879–1890, Nov. 2012, issn: 1084-8045. doi: 10 . 1016 / j . jnca . 2012 . 07 . 012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S1084804512001646> (visited on 08/26/2016).
- [59] E. Galceran and M. Carreras, "Coverage path planning for marine habitat mapping", in *Oceans*, 2012, IEEE, 2012, pp. 1–8. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6404907 (visited on 04/25/2016).
- [60] J. Lloret, S. Sendra, M. Ardid, and J. J. P. C. Rodrigues, "Underwater Wireless Sensor Communications in the 2.4 GHz ISM Frequency Band", *Sensors (Basel, Switzerland)*, vol. 12, no. 4, Mar. 2012, issn: 1424-8220. doi: 10 . 3390 / s120404237. [Online]. Available: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC3355409/>.
- [61] S. Ramakrishna and I. Nissen, "Next generation cognitive system approaches in the underwater communication area", *Applied Ocean Research*, vol. 38, pp. 136–141, Oct. 2012, issn: 0141-1187. doi: 10 . 1016 / j . apor . 2012 . 07 . 007. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0141118712000685> (visited on 08/26/2016).
- [62] M. Yan, D. Zhu, and S. X. Yang, "Complete coverage path planning in an unknown underwater environment based on DS data fusion real-time map building", *International Journal of Distributed Sensor Networks*, vol. 2012, 2012. [Online]. Available: <http://www.hindawi.com/journals/ijdsn/2012/567959/abs/> (visited on 05/10/2016).
- [63] B. d'Andréa-Novel and M. D. Lara, *Control Theory for Engineers: A Primer*, en. Springer Science & Business Media, May 2013, Google-Books-ID: gWpCSIBTNr8C, isbn: 978-3-642-34324-7.
- [64] E. Galceran and M. Carreras, "A survey on coverage path planning for robotics", *Robotics and Autonomous Systems*, vol. 61, no. 12, Dec. 2013, issn: 0921-8890. doi: 10 . 1016 / j . robot . 2013 . 09 . 004. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S092188901300167X>.
- [65] E. Galceran, S. Nagappa, M. Carreras, P. Ridao, and A. Palomer, "Uncertainty-driven survey path planning for bathymetric mapping", in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, IEEE, 2013, pp. 6006–6012. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6697228 (visited on 05/17/2016).

- [66] G. Hattab, M. El-Tarhuni, M. Al-Ali, T. Joudeh, and N. Qaddoumi, "An Underwater Wireless Sensor Network with Realistic Radio Frequency Path Loss Model", en, *International Journal of Distributed Sensor Networks*, vol. 9, no. 3, p. 508 708, Mar. 2013, issn: , 1550-1477. doi: 10 . 1155 / 2013 / 508708. [Online]. Available: <http://dsn.sagepub.com/content/9/3/508708> (visited on 08/30/2016).
- [67] M. T. Leccadito, T. Bakker, R. Niu, and R. H. Klenke, "A Kalman Filter Based Attitude Heading Reference System Using a Low Cost Inertial Measurement Unit", PhD thesis, Master's thesis, Virginia Commonwealth University, 2013. [Online]. Available: <http://arc.aiaa.org/doi/pdf/10.2514/6.2015-0604> (visited on 12/27/2016).
- [68] C. Van Den Berg, *IHC Merwede Handbook for Centrifugal Pumps and Slurry Transportation*. MTI Holland, 2013.
- [69] J. Claus, "Design and Development of an Inexpensive Acoustic Underwater Communications and Control System", en, PhD thesis, Florida institute of technology, Florida, 2014.
- [70] P. Freitas, "Evaluation of Wi-Fi Underwater Networks in Freshwater", PhD thesis, UNIVERSIDADE DO PORTO, Porto, 2014.
- [71] E. Galceran, "Coverage path planning for autonomous underwater vehicles", en, PhD thesis, Universitat de Girona, 2014.
- [72] ——, "Towards Coverage path planning for autonomous underwater vehicles", en, PhD thesis, Universitat de Girona, 2014.
- [73] G. van der Schriek, *Dredging Technology - Guest lecture notes CIE5300 Issue 2014*, en. GLM van der SCHRIEK BV, 2014.
- [74] Wei Gao, Yalong Liu, and Bo Xu, "Robust Huber-Based Iterated Divided Difference Filtering with Application to Cooperative Localization of Autonomous Underwater Vehicles", *Sensors* (14248220), vol. 14, no. 12, pp. 24 523–24 542, Dec. 2014, issn: 14248220. doi: 10.3390/s141224523. [Online]. Available: <http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=100145980&lang=nl&site=ehost-live> (visited on 05/20/2016).
- [75] F. Abyarjoo, A. Barreto, J. Cofino, and F. R. Ortega, "Implementing a sensor fusion algorithm for 3d orientation detection with inertial/magnetic sensors", in *Innovations and Advances in Computing, Informatics, Systems Sciences, Networking and Engineering*, Springer, 2015, pp. 305–310. [Online]. Available: http://link.springer.com/chapter/10.1007/978-3-319-06773-5_41 (visited on 12/27/2016).
- [76] M. Algabri, H. Mathkour, H. Ramdane, and M. Alsulaiman, "Comparative study of soft computing techniques for mobile robot navigation in an unknown environment", *Computers in Human Behavior*, vol. 50, pp. 42–56, Sep. 2015, issn: 0747-5632. doi: 10.1016/j.chb.2015.03.062. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0747563215002605> (visited on 04/17/2016).
- [77] Y. Feng, X. Li, and X. Zhang, "An Adaptive Compensation Algorithm for Temperature Drift of Micro-Electro-Mechanical Systems Gyroscopes Using a Strong Tracking Kalman Filter", *Sensors*, vol. 15, no. 5, pp. 11 222–11 238, 2015. [Online]. Available: <http://www.mdpi.com/1424-8220/15/5/11222/htm> (visited on 12/28/2016).
- [78] D. C. Giancoli, *Physics Principles with Applications*, en. Pearson Education, Limited, Jan. 2015, Google-Books-ID: jYlyngEACAAJ, isbn: 978-1-292-05712-5.
- [79] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice with MATLAB*, en. John Wiley & Sons, Feb. 2015, Google-Books-ID: Sgx9BgAAQBAJ, isbn: 978-1-118-98496-3.
- [80] S. Babani, A. A. Bature, M. I. Faruk, and N. K. Dankadai, "COMPARATIVE STUDY BETWEEN FIBER OPTIC AND COPPER IN COMMUNICATION LINK", 2016. [Online]. Available: <http://www.academia.edu/download/34322554/comparative-study-between-fiber-optic-and-copper-in-communication-link.pdf> (visited on 10/03/2016).
- [81] Jolectra, *PLC diagrams Besturingspaneel container - 216.156.P1*. Jun. 2016.
- [82] Mathworks, *Mechanical-to-hydraulic power conversion device - Simulink - MathWorks Benelux*. 2016. [Online]. Available: <https://nl.mathworks.com/help/physmod/hydro/ref/fixeddisplacementpump.html> (visited on 11/14/2016).

- [83] S. A. Miedema, *Slurry Transport Fundamentals, A historical Overview & The Delft Head Loss & Limit Deposit Velocity Framework*. TU Delft, Jun. 2016.
- [84] W. Vlasblom, "Designing Dredging Equipment", TU Delft, Delft University of Technology, Lecture notes, 2016.
- [85] A. Recuero, R. Serban, B. Peterson, H. Sugiyama, P. Jayakumar, and D. Negrut, "A high-fidelity approach for vehicle mobility simulation: Nonlinear finite element tires operating on granular material", *Journal of Terramechanics*, vol. 72, pp. 39–54, Aug. 2017. doi: 10.1016/j.jterra.2017.04.002.
- [86] Roger R Labbe jr, *Kalman and Bayesian Filters in Python*. Jan. 2017.
- [87] Wheeled mobile robotics: from fundamentals towards autonomous systems, MATLAB examples, Oxford: Butterworth-Heinemann, an imprint of Elsevier, 2017, 491 pp.
- [88] 3Blue1Brown, *Visualizing quaternions (4d numbers) with stereographic projection - YouTube*, 2018. [Online]. Available: <https://www.youtube.com/watch?v=d4EgBgTm0Bg> (visited on 05/24/2020).
- [89] S. J. Johnston, P. J. Basford, C. S. Perkins, H. Herry, F. P. Tso, D. Pezaros, R. D. Mullins, E. Yoneki, S. J. Cox, and J. Singer, "Commodity single board computer clusters and their applications", *Future Generation Computer Systems*, vol. 89, pp. 201–212, Dec. 2018, issn: 0167739X.
- [90] R. Serban, N. Olsen, D. Negrut, A. Recuero, and P. Jayakumar, "A co-simulation framework for high-performance, high-fidelity simulation of ground vehicle-terrain interaction", *International Journal of Vehicle Performance*, vol. 5, Aug. 8, 2018. doi: 10.1504/IJVP.2019.10021232.
- [91] A. Tasora, D. Mangoni, and D. Negrut, *An Overview of the Chrono Soil Contact Model (SCM) Implementation*. Aug. 9, 2018.
- [92] R. v. d. Wetering, "Ihc mti crawler - final report", IHC MTI B.V., report, 2018.

APPENDICES

APPENDIX CRAWLER PARTLIST A

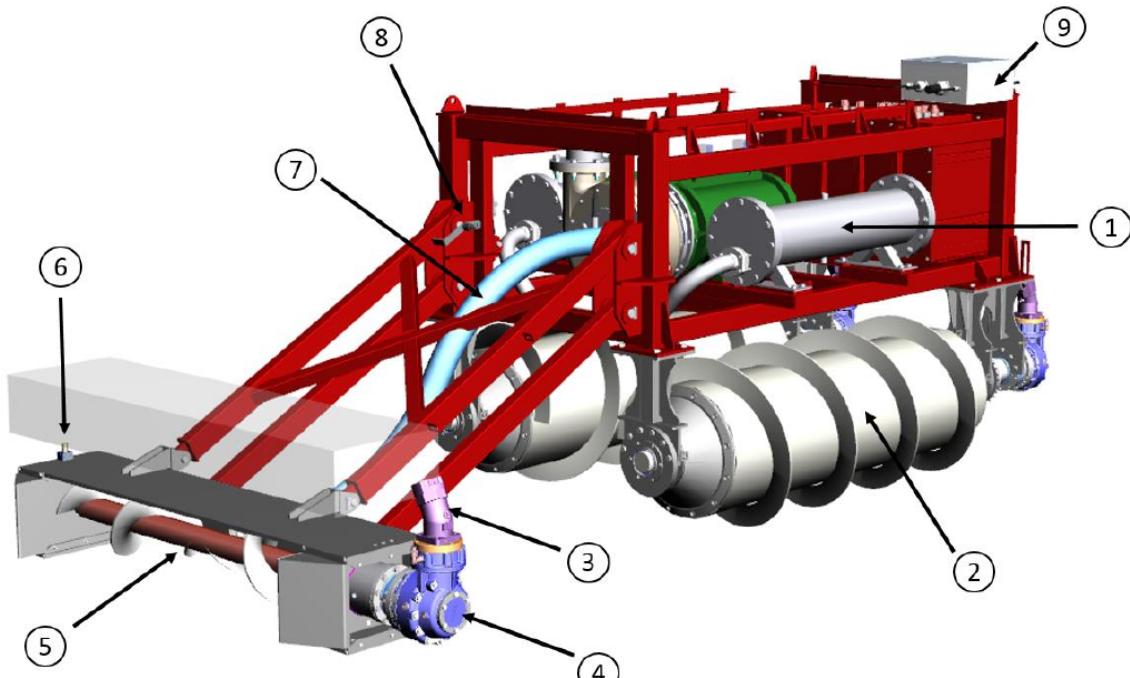


FIGURE A.1: MAJOR PARTLIST SIDEVIEW [92]

NO	DESCRIPTION
1	Oil buffer
2	Archimedes screw propulsion
3	Hydraulic motor
4	Gearbox
5	Auger
6	RPM sensor auger
7	Flexible suction hose 100mm
8	Dredge head angle sensor
9	Termination box, interface between crawler and umbilical

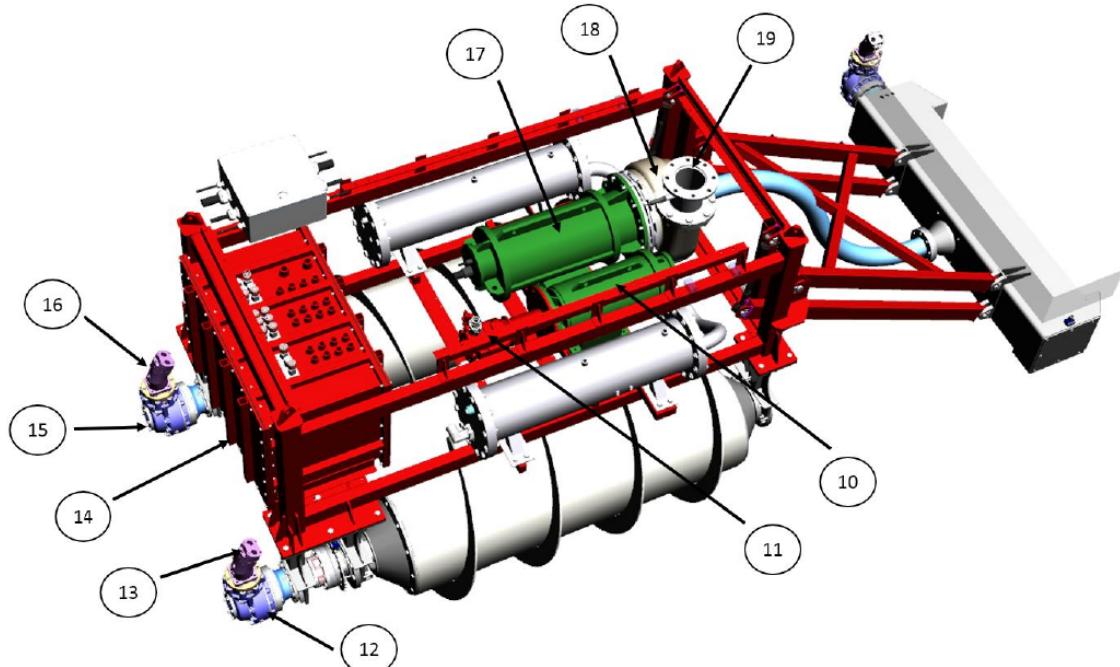


FIGURE A.2: MAJOR PARTLIST TOPVIEW [92]

NO	DESCRIPTION
10	Electric motor
11	Hydraulic pump
12	Gearbox
13	Hydraulic motor
14	Connection box
15	Gearbox
16	Hydraulic motor
17	Electric motor
18	IHC TT 150 dredge pump
19	Discharge

APPENDIX B

APPLIED RESEARCH METHODS

The used research strategy is described for the purpose of transparency and quality control. It serves as the basis for chapter 3. By defining keywords and queries, setting boundaries and specifying the databases, potential sources are filtered on relevance. These are then read and reviewed, if they are indeed relevant and adhere to stated selection criteria, they are used in this study.

APPLIED PARAMETERS

Parameters for a research study The applied parameters for each search query are listed below:

PARAMETER	VALUE
Language of publication	English or Dutch.
Area of research	Engineering, Maritime, Artificial Intelligence, Sensors, Dredging.
Industry sector	Maritime, Robotics, Mining.
Geographical area	World wide.
Time period	1995 till present.
Types of literature	Peer review papers, MSc thesis, Ph.D. thesis, scientific books, (inter-)national standards.

KEYWORDS AND QUERIES

- "CPP" **OR** "coverage path planning"
 - **AND** "underwater"
 - **AND** "cellular decomposition"
 - * **AND** "Morse"
 - * **AND** "Trapezoidal"
 - * **AND** "Boustrophedon"
 - **AND** "landmark" **OR** "topological"
 - * **AND** "slice decomposition"
 - * **AND** "neural networks"
 - **AND** "grid"
 - * **AND** "spanning tree" **OR** "STC"
 - * **AND** "neural networks"
 - * **AND** "probability" **OR** "certainty"
 - **AND** "cooperative localization"
- "auger" **OR** "screw conveyor"
 - **AND** "production" **OR** "flow"
 - **AND** "dredging" **OR** "dredge head"
- "underwater" **AND** "communication"
 - **AND** "wireless"
 - * **AND** "protocol"

- * **AND** "electromagnetic"
- * **AND** "acoustic"
- * **AND** "optical"
- * **AND** "environment"
- **AND** "umbilical"
- * **AND** "environment"
- "IMU" **OR** "Inertial Measurement Unit"
 - **AND** "gyro" **OR** "gyroscope"
 - * **AND** "error"
 - * **AND** "temperature"
 - **AND** "accelerometer"
 - * **AND** "error"
 - * **AND** "temperature"
 - * **AND** "gravity"
 - **AND** "magnetometer"
 - * **AND** "error"
 - * **AND** "temperature"
- "pressure" **AND** "sensor"
 - **AND** "underwater"
 - **AND** "error"
 - **AND** "temperature"
 - **AND** "water" **AND** "depth"
 - **AND** "resolution"
- "Kalman filter"
 - **AND** "gyro" **OR** "gyroscope"
 - **AND** "accelerometer"
 - **AND** "magnetometer"
 - **AND** "quaternions"
 - **AND** "AHRS" **OR** "Attitude and heading reference system"
 - **AND** "extended"
 - **AND** "unscented"

DATABASES AND SEARCH ENGINES

DATABASE	TYPE
Academic Search Complete	More than 10.000 digital academic magazines
EBSCO	
Google Scholar	Scientific Internet search engine
Microsoft Academic Research	Scientific Internet search engine
NEN Connect	Search engine for (inter-)national norms ISO / NEN
Science direct	Over 2.000 scientific magazines
Springer link	Over 2.500 scientific magazines

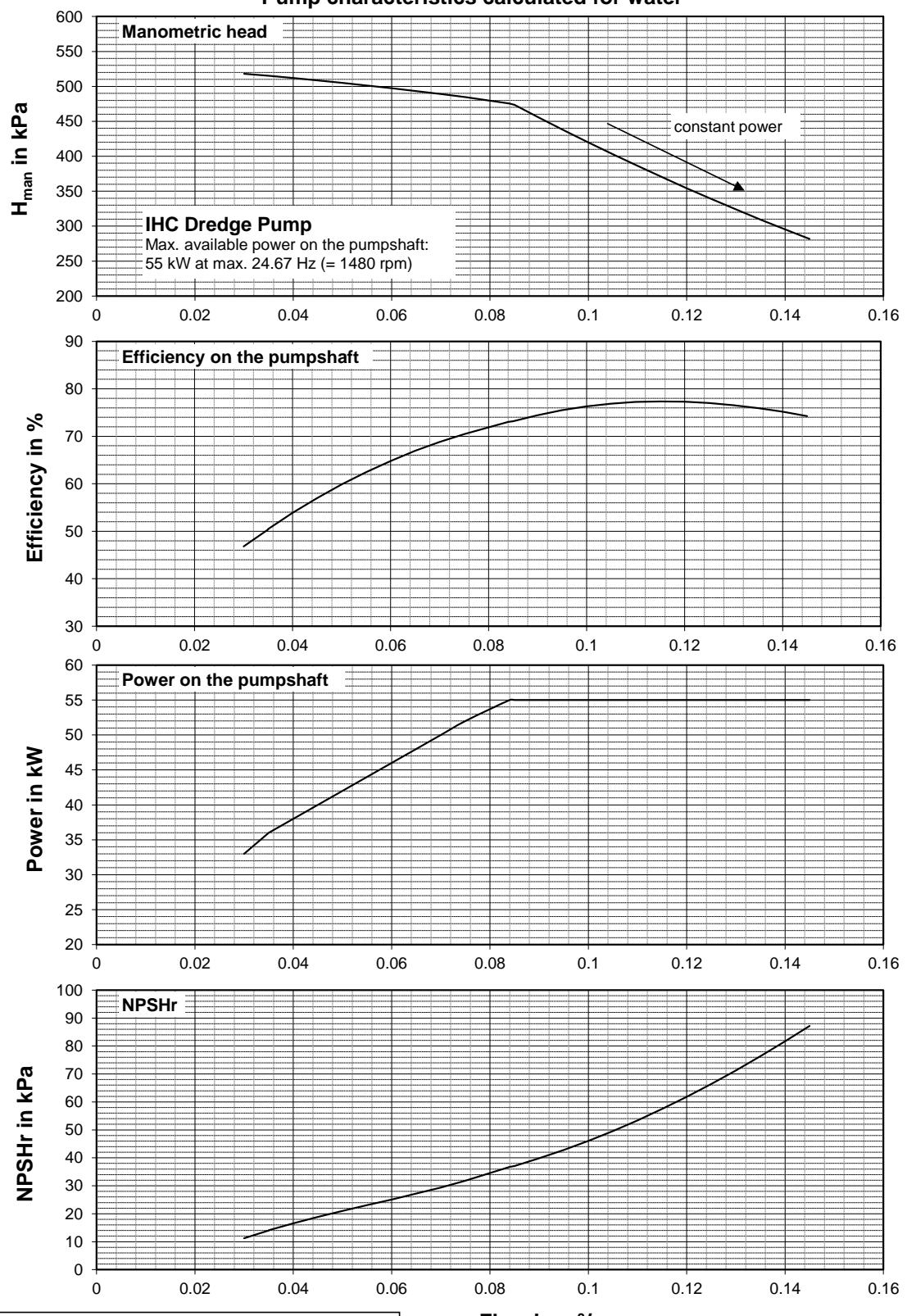
Wiley Online Library	Almost 1.000 scientific magazines
MTeye	MTI Library consisting of roughly 700m of technical books related to soil, sea, mining and engineering
My own Library	A mere 20m of technical books, related to math, engineering, programming, electronics and artificial intelligence

C APPENDIX IHC

DATA SHEET PUMP TT15-55

IHC Merwede

Pump characteristics calculated for water



This prognosis is based on measurements according to
ISO 2548 (Class C) code for acceptance testing of pumps

Flow in m³/s

Pump type IHC 37.5-9.5-15, with a 3-bladed impeller

Date: 12-Jul-2011

© Copyright IHC

Report:

GO 4294

Page:

97