

THESIS

DESIGN OF AN AUTONOMOUS DREDGE BOT CONTROLLER



A L^AT_EX class, typeset according to the Royal IHC brandguidelines. This report has full support for glossaries, biblatex and various environments. Such as equations, figures and listings. Glossaries descriptions are combined with tooltips, such that a digital reader, has imidiate acces to the description.



THE TECHNOLOGY
INNOVATOR.

ROYALIHC.COM

THESES
AUTONOMOUS CRAWLER DESIGN
IHC MTI B.V.

Jelle Spijker

IHC MEDUSA B.V.

May 18, 2020

Client Royal IHC
External reference HAN-666
Internal reference JS01
Version 0
Status final
Classification none

IHC MTI B.V.
P.O. Box 2, 2600 MB Delft
Delftsepoort 13, 2628 XJ Delft
S: info@ihcmti.com
T: +31 88 015 2535
M: info@ihcmti.com



**THE TECHNOLOGY
INNOVATOR.**

ROYALIHC.COM

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	May 18, 2020	
REVIEWD BY	ir. Frits Hofstra	May 18, 2020	
REVIEWD BY	J. B. van Elburg MSc. BEng.	May 18, 2020	
APPROVED BY	A. J. P. M. Koevoets MBA BSc.	May 18, 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 May 18, 2020

CHAPTER SUMMARY

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetur.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetur at, consectetur sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae,



dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetur a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maece-
nas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend
consectetur. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium
ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu
urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam,
pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum,
eros sem dictum tortor, vel consectetur odio sem sed wisi.



CHAPTER CONTENTS

SUMMARY	II
Contents	IV
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 DREDGER	2
2.3 HYDRAULIC DREDGING PRINCIPALS	4
2.3.1 DREDGE PUMP	4
2.3.2 AUGER DREDGE HEAD	4
3 RESEARCH	6
3.1 UNDERWATER COMMUNICATION	6
3.1.1 WIRED COMMUNICATION	6
3.1.2 WIRELESS COMMUNICATION	8
3.2 SENSORS	12
3.2.1 STATE SENSING	13
3.3 LOCATION UNDER UNCERTAINTY	18
3.3.1 LOCALIZATION REFINEMENT USING KALMAN FILTERS	19
3.3.2 BASIC KALMAN FILTERING	19
3.4 COVERAGE PATH PLANNING	23
3.4.1 MORSE-BASED CELLULAR DECOMPOSITION	24
3.4.2 LANDMARK-BASED TOPOLOGICAL COVERAGE	28
3.4.3 GRID-BASED METHODS	33
4 CONCEPT DESIGN	38
4.1 STRATEGY DECISION	38
5 DESIGN VALIDATION	39
5.1 SIMULATION	39
6 NOMENCLATURE	41
7 GLOSSARY	45
8 ACRONYMS	48
9 BIBLIOGRAPHY	50
APPENDICES	55
A CRAWLER PARTLIST	56
B APPLIED RESEARCH METHODS	58
C KALMAN EXAMPLE FALLING BALL	60

CHAPTER 1 INTRODUCTION

This chapter will first specify three use-cases, specified in the project assignment, in which an AOD 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 AOD and stand at the basis for the controller design.

1.1.1 ARBITRARY SHAPED SPACE

An AOD is placed in a predefined arbitrary shaped space, not too complex, with an area of 3500m^2 . The shape of this space is set, but the movement pattern is unrestricted. The AOD 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 AOD 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 AOD 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 AOD 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 AOD 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 AOD 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 [39] 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 an 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. Schrieck [64] states that the issue in maintain existing waterways and harbours, preserve the depth of the bed by regular 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 characteristics of this kind of work is the weak cohesion of the soil to be removed, since it consist of recently deposited sediment and no significant consolidation has taken place yet.

A special kind of maintenance dredging is sanitation dredging which is a process specially 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 it needs to be removed with care and precision.

2.2 COMMONLY USED VESSELS AND EQUIPMENT

Common dredge tools used during maintenance work are listed below, of this list backhoes and suction dredgers are mostly used during port maintenance. Vlasblom [72] states that dredgers can be divided in mechanical dredgers and hydraulic dredgers. Where the difference lies in the way the soil is excavated; either mechanical or hydraulic.

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 which, is a storage container or compartment..

The various types of mechanical dredgers won't be described in this section, since the Autonomous Operating Dredgebot (AOD) used in our uses-cases will be of a hydraulic type.

2.2.2 HYDRAULIC DREDGER

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

PLAIN SUCTION DREDGER

Vlasblom [72] describes a plain suction dredger as an stationary dredger, consisting of a pontoon anchored by one or more wires an 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 DREDGER

The Trailing Suction Hopper Dredger (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 attached to each suction tube and is trailed across the sea bed to loosen the soil before it is pumped up [64]. This soil is stored in a hopper which is periodically discharged, at an designated location, through dumping or pumping out.

AUGER SUCTION DREDGER

According to VBKO Vereniging van waterbouwers in bagger-, kust- en oeverwerken [11] an Auger Suction Dredger (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, where it cuts the material and actively transports in to the centre, where it is 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 types of dredgers are well suited for sanitation maintenance.

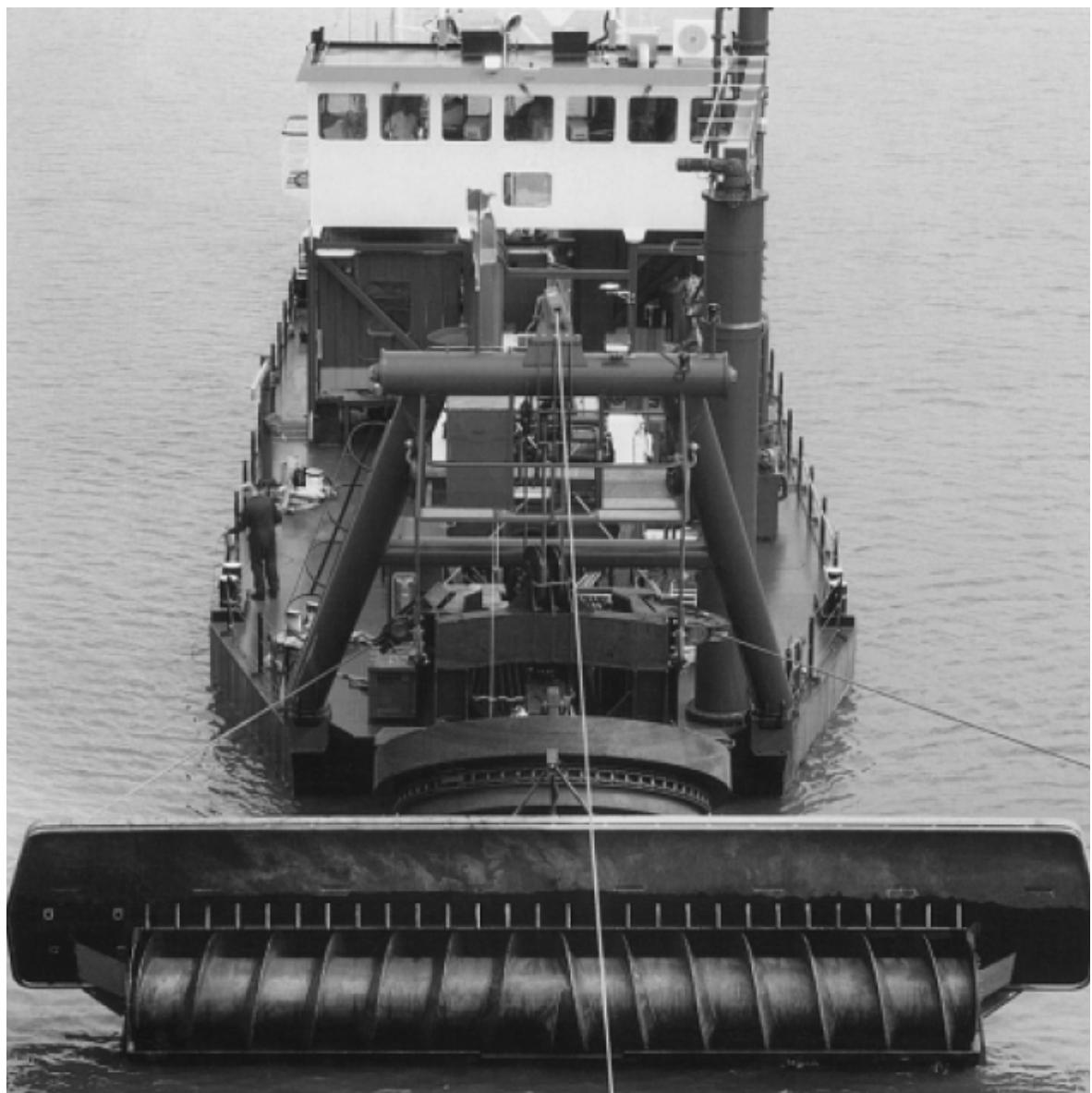


FIGURE 2.1: AUGER SUCTION DREDGER [11]

CUTTER SUCTION DREDGER

According to Vlasblom [72] a Cutter Suction Dredger (CSD) is a stationary dredger equipped with a cutter device (cutter head) which excavate the soil before it is 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 [59] hydraulics systems are the de-facto industry of transportation for dredged sedimented, or slurry; Hydraulic systems consists of pipes, either flexible or rigid, combined with centrifugal pumps, a suction mouth and a discharge unit. These components are usually placed in series. A slurry moving through a hydraulic system experiences friction, both from shearing of a fluid along a wall and internal shearing of the fluid itself. This friction results in a pressure drop along these components. Coupled with a pressure drop needed to overcome a height difference, result in a needed pressure, which the pump has to deliver for a certain flow-rate.

The section below shortly describe 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 constrains such as a flexible dredge line to shore can be added to the equation. Since the dredge bot does not have a holding space to store collected sediment this is part of the normal operation. It was however opted, to not applied 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 [59]. Assuming a steady flow, the pump basically increases the Bernoulli head of the flow between point 1, the eye and point 2, the exit [48].

2.3.2 AUGER DREDGE HEAD

An auger umbilical which, is a electronic cable connecting an underwater vehicle, 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 possible polluted, layers at a relatively high production rate [64].

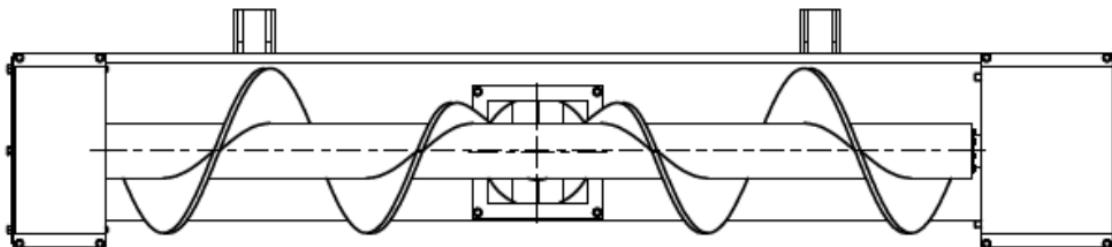


FIGURE 2.2: SCHEMATIC DRAWING OF AN AUGER DREDGE HEAD [74]

The auger is in effect a screw conveyor which guiding the material towards the suction head. Green and Perry [33] states that the screw conveyor one of the oldest and most versatile conveyor types 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 pump are the limiting factor in the mass flow, and that the auger simply delivers what is needed.

3 CHAPTER RESEARCH

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 ??, 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 Global Positioning System (GPS) deprived environment.

Section ?? describes the use of cooperative localization techniques and Kalman-filter which, is 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.

Lastly an survey is made for useful strategy at a higher abstraction level. Section ??, describes how a crawler could best perform its main task: covering and dredging a large basin, uniformly. These so called Coverage Path Planning (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 which, is a electronic cable connecting an underwater vehicle,. 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 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 is 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 ?? 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 which, is a sequential binary sequence,. 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 fibre optic cables, where a modulated pulse of light propagates through a glass tube through the principle of Total Internal Reflection (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. [70] 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 fibre 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 fibre 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 fibre optical systems facilitates much longer runs for signal transmission. The attenuation loss experienced in fibre optic cables can be attributed to microscopic and macroscopic impurities in the fibre material and structure, which cause absorption and scattering of light signal. In figure ?? 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 fibre-optics are paramount.

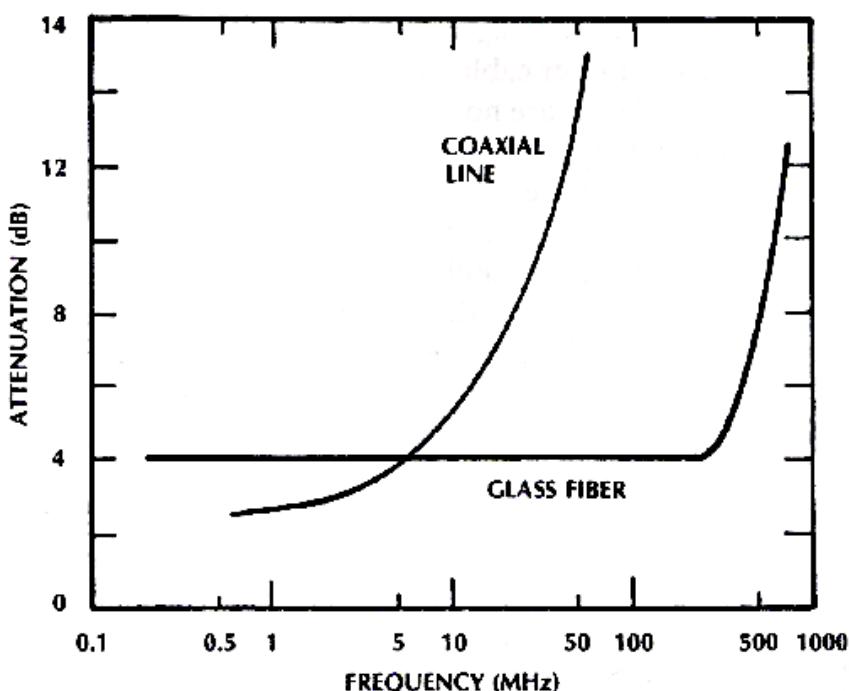


FIGURE 3.1: EFFECTIVE ATTENUATION FIBRE VS COPPER CABLE 1 km [9]

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 [16] states that most present day vehicles are Remote Operated Vehicle (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 Autonomous Underwater Vehicle (AUV) – which operate without an umbilical tether. This statement is supported by Valavanis, Gracanin, Matijasevic, et al. [7], 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 its 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 [32], whom describes a mathematical model which allow the state representation of the dredge bot, as described in section ??, 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 is assumed that these forces can be neglected. According to Feng and Allen [23] the effects of the cable can be reduced when it is deployed by a drum on the shore with negligible tension when it is 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 Institute of Electrical and Electronics Engineers (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 is more commonly known Ethernet. Which consists of a multitude of protocols. In this IEEE norms are the physical layer, data link layers and the Media Access Control (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 Logical Link Control (LLC) sub layer and the network's physical layer. Where the LLC makes it possible to let several network protocols coexist. According to Jolectra [71] the current dredge bot makes use of an *Allen Bradley ETHERNET/IP adapter* of type 1769-AENTR, which is allows the use Common Industrial Protocol (CIP), Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). Where CIP is used by EtherNet/IP, and is a familiar and widely used protocol for controllers.

3.1.2 WIRELESS COMMUNICATION

Freitas [61] 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 Electromagnetic Waves (EMW) in water.

He [61] 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.* [51] 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 [43] and physical constraints of the medium [37].

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 [68].

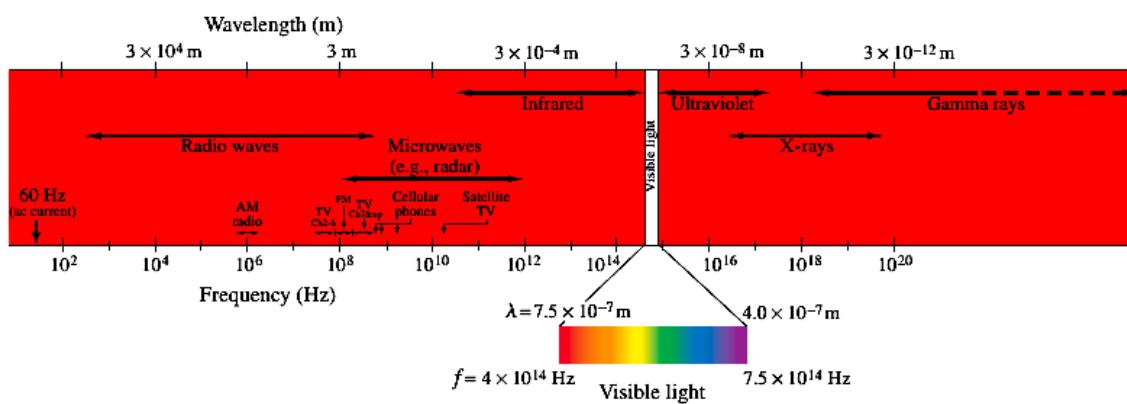


FIGURE 3.2: ELECTROMAGNETIC SPECTRUM [68]

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.

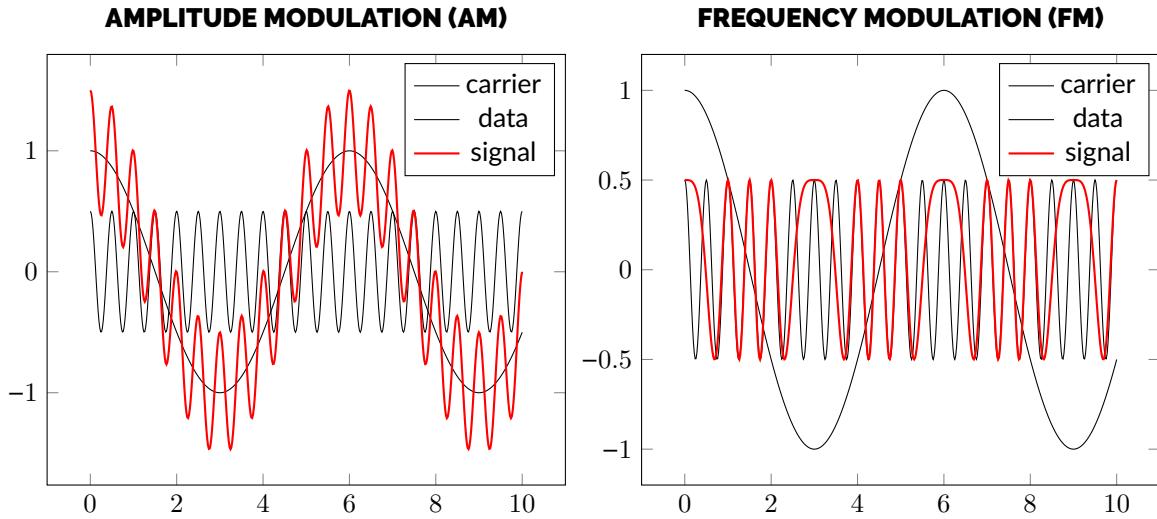


FIGURE 3.3: SIGNAL MODULATION

Hagman, Elias [40] 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 [52] 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 [40] 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 [40].

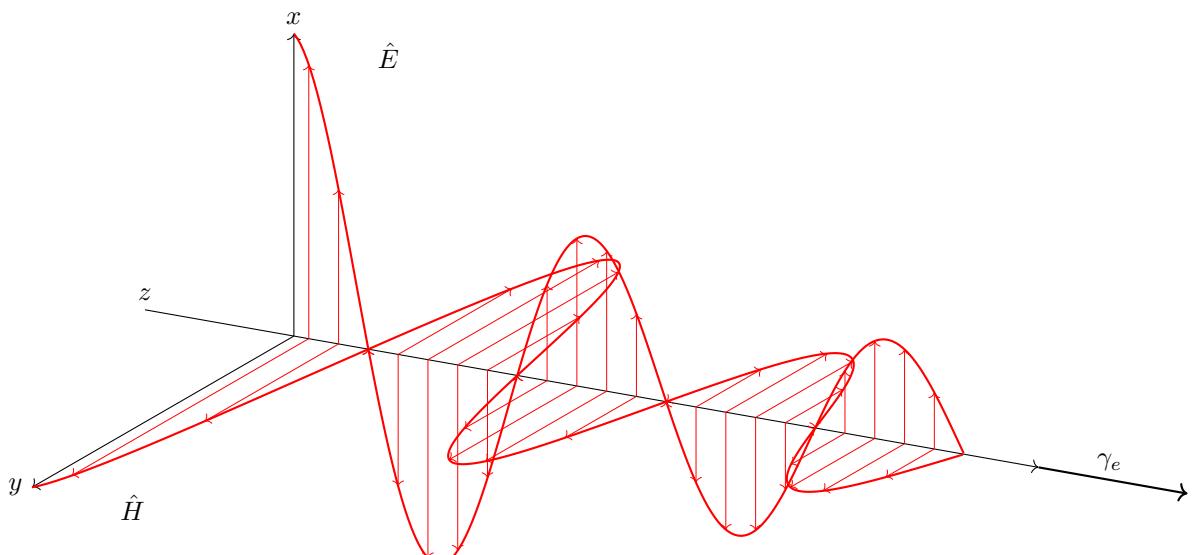


FIGURE 3.4: DAMPENING OF ELECTRIC AND MAGNETIC FIELD

Where \hat{E} and \hat{H} are the amplitudes of the electric and the magnetic field wave and γ_e which is propagation constant given in [m] expressed in ϵ_e which is permittivity given in [-], as shown in equation 3.3, where μ_e which is electromagnetism permeability given in [H/m] and σ_e which is Electrical conductivity given in [S/m] of a material. Here α_e which is attenuation given in [dB/m] and β which is phase factor of a wave given in [-].

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

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

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

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

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 [60] 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. [57] 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.

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

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 [44] 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 [44] 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 [44], Ainslie [41] 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

Sub-section 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 [61] 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.

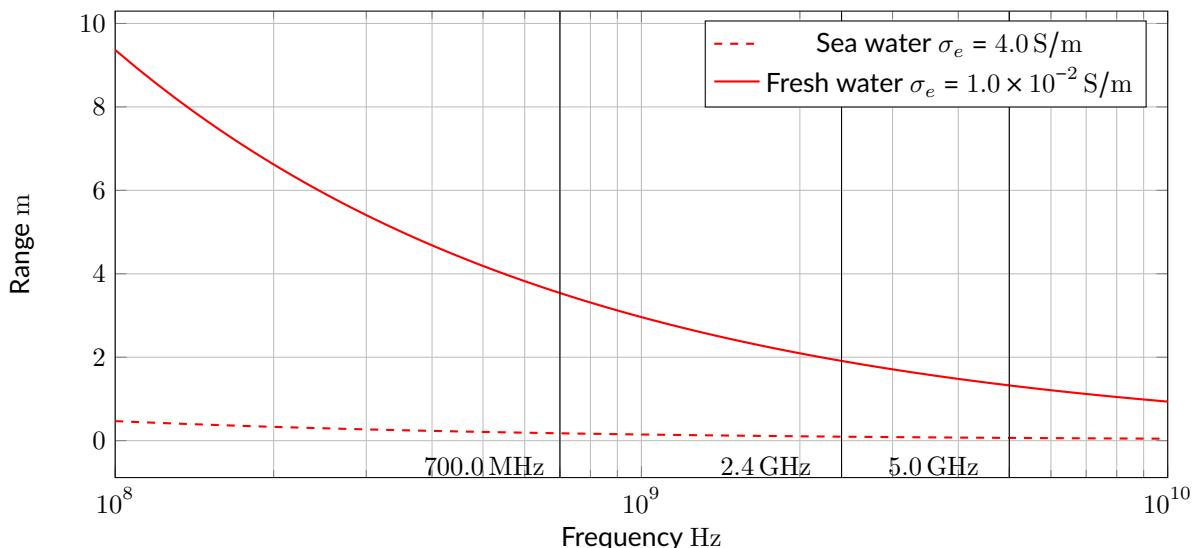


FIGURE 3.5: PROPAGATION RANGE OF WI-FI IN WATER.

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 Internet of Things (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 [26] 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.

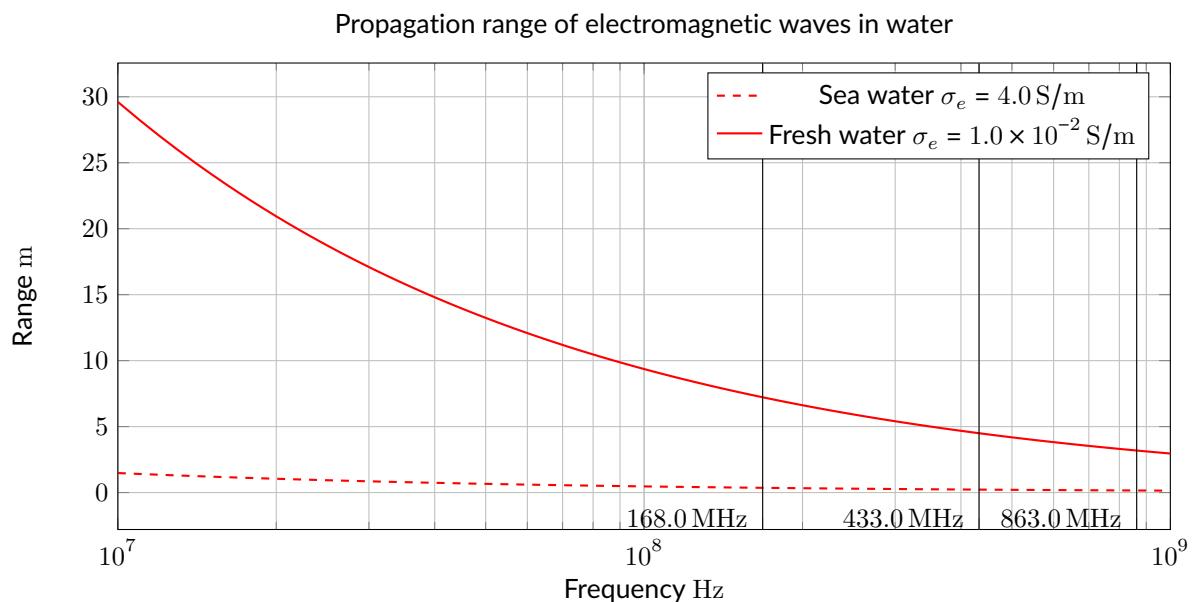


FIGURE 3.6: PROPAGATION RANGE OF LO-FI IN WATER

ELECTRIC CURRENT

Another way to communicate is through the use of electric current. Hagman, Elias [40] 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 [40]. 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 [40], Claus [60] and Domingo [49] state that acoustic communication is therefore the preferred way. This type of communication makes use of Sound Waves (SW), or Acoustic Waves (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 measure than the displacement [68]. This principle is used by hydrophones; These are in-effect microphones designed to be used underwater. Using piezo-electric transducers to convert pressure waves into electricity. Although acoustic communication is the preferred method, there are a lot of challenges to overcome. According to Tetley and Calcutt [35] transmitting and receiving acoustic energy in seawater is affected by the often unpredictable ocean environment. Lanbo, Shengli, and Jun-Hong [37] and Edward Tucholski [28] 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 and 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 which, is 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,. 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.

Whilst the sensors described in Section 3.2.1 determine the state of a dredge bot; Namely its orientation and position. The sensor described in Section ?? describe a variety of acoustic, vision and light 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 ??, the state vector \vec{x}_k describe the position in a global reference frame and the orientation of the dredge bot itself. This state vector can be obtained using a fusion of multiple sensors, which are described below.

INERTIAL MEASUREMENT UNIT

Leccadito, Bakker, Niu, et al. [58] describes Inertial Measurement Unit (IMU) as a platform of sensors which output measurements of the vehicle state, such as angular rates and accelerations. The sensors usually consist of a gyroscope, which outputs angular rates about the three vehicle axes, and accelerometer, which output acceleration also along each of the three axes. These sensors are sometimes complemented with a magnetometer, which measures the strength of a magnetic field, like the one generated by the earth, along three axes.

ACCELEROMETER

There are many different types of Micro Electro Mechanical System (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. [58] 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.

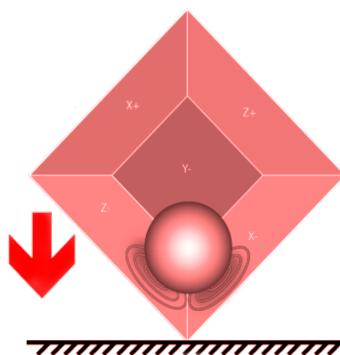


FIGURE 3.7: GRAVITATIONAL PULL ON MULTIPLE AXES [58]

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.

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

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. [66] states that the problem with accelerometers is that they 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 [47], 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. [66] observed no drift of the signal but established that it contains a lot of noise. Kownacki [45] describes that a Kalman filter is a good candidate to filter the noise, using a gyroscope. Where the Analog Digital Conversion (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. gyroscope 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 being sent in a circular path. These sensor 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.

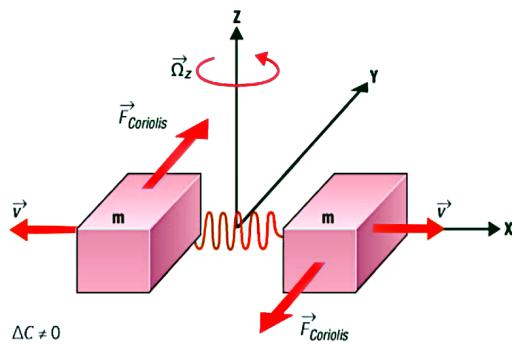


FIGURE 3.8: GYROSCOPE USING CORIOLIS EFFECT [58]

Leccadito, Bakker, Niu, et al. [58] tells that most MEMS gyroscope are based on the tuning fork structure, where the Coriolis effect is used to measure ω which is angular velocity given in $[\text{rad/s}]$. 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 ???. From where the heading at a certain axis can be calculated using the trapezoidal-rule which, is a technique for approximating the definite integral,. 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. [66] observed that the computed results drifts 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. An other source of drift is temperature related, Feng, Li, and Zhang [67] states that a gyroscope is sensitive to temperature variations, so the surrounding temperature variations leads to a the 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 which is true angular velocity given in [rad/s], but unknown and B_d which is slow chancing component of the signal; this is the gyroscope drift given in [rad/s]. Where n_s which is stochastic component of a signal given in [rad/s].

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

Abyarjoo, Barreto, Cofino, et al. [66] further states that the slow chancing 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-chancing 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 [29] and T is the measured temperature of the gyroscope in K and T' which is rate of temperature variation given in [K/s]

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

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} [bit] and 2^{16} [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. [58] explains that a magnetometer coupled with an accelerometer can effectively calculate a heading angle. This is further explained by Konvalin [36] 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 earth's 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. [58], 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} which is raw magnetometer vector given in [T] and \vec{m}_{hi} which is hard iron adjusted vector given in [T]. 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. [58] describes that soft iron effects on the sensor are determined by the orientation of the materials, and it is 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.

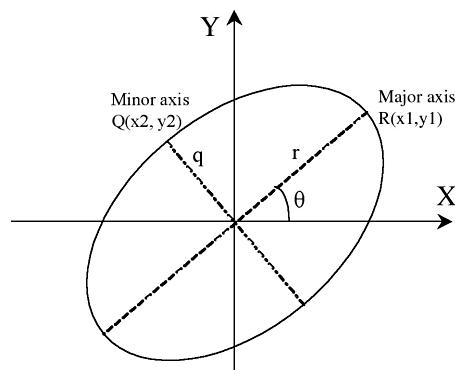


FIGURE 3.9: SOFT IRON DISTORTION [36]

It is 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 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 calculate 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.

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

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

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

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

PRESSURE SENSOR

White [48] 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 Δ_p 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 which is height error given in [m] is known. A height for the pressure sensor, with regards to the top fluid column can be obtained. It is 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.

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

Three types of pressure measurements are usually performed, according to Webster [13]:

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.

These measurements are usually made by applying one of the following methods:

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 [13].

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.

3.3 LOCATION UNDER UNCERTAINTY

Due to the absence of an ubiquitous global localization system such as GPS in underwater environments, AUV 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 AUV [62]. 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 Long Base Line (LBL) and Ultra Short Base Line (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 Terrain Relative navigation (TRN) and Simultaneous Localization And Mapping (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 will be illustrated in section ??

Other methods for navigation under uncertainty are based on probabilities taken into account *a priori* known characteristics of sensors and actuators such as Linear-Quadratic Gaussian Motion Planning (LQG-MP) and Rapidly exploring Random Trees (RRT). According to Galceran, Nagappa, Carreras, et al. [56] 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, is submerged equivalent of an above-water topographic 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 [42], which uses a team of three robots where two alternating robots act as stationary beacons. Others like Wei Gao, Yalong Liu, and Bo Xu [65] use a single surface which act as an communication and navigation aid. It is quite common to filter sensor readings and state vectors from the multiple robots using a Kalman filter. Section ?? describes in more detailed how swarms of bots could help each other in estimating their position.

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 is 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 Linear Quadratic Estimation (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 is important that the basics are explained. The section will feature a short description of the background and workings of a Kalman filter and quaternion which, is a number system that extends the complex numbers. They were first described by Irish mathematician William Rowan Hamilton in 1843 and applied to mechanics in three-dimensional space. A feature of quaternions is that multiplication of two quaternions is non-commutative. Hamilton defined a quaternion as the quotient of two directed lines in a three-dimensional space or equivalently as the quotient of two vectors.,

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 [30], d’Andréa-Novel and Lara [54], Chui and Chen [12], Grewal and Andrews [69] 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 which is position along the z-axis given in [m] and v_z which is velocity along the z-axis given in [m/s].

A Kalman filter works by estimating the state of a process based on *a priori* states. It is 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 an 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.

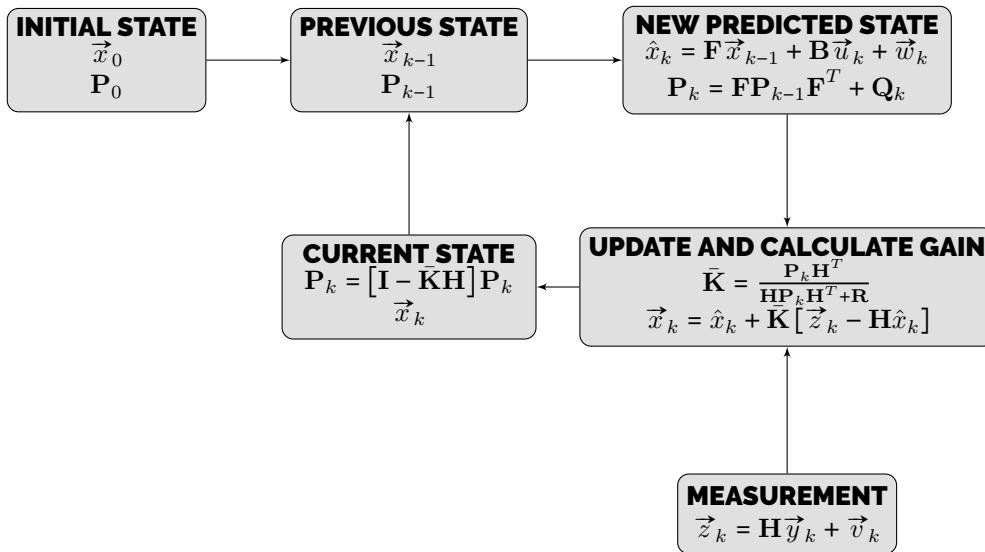


FIGURE 3.10: FLOW OF A KALMAN FILTER

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 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 variables 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 is initially assumed that errors 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 σ_{s_z} and the velocity σ_{v_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.

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

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

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 which, is 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,. 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 [69] 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.

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

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

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.

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

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

The above described example of a falling ball can be simulated with a Python script. Such a script can be found in appendix C. 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 [73] an effective way to measure the results of a simulated Kalman filters, is the Normalized Estimated Error Squared (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 .

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

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

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

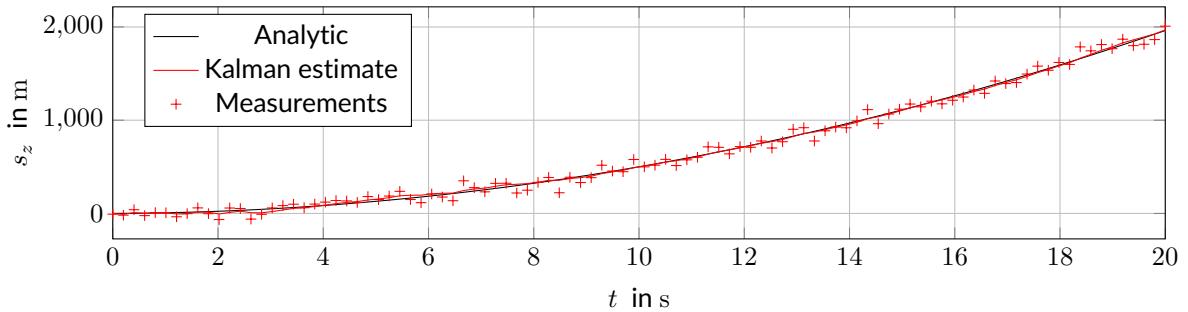


FIGURE 3.11: COMPARISON OF ESTIMATED, MEASURED AND REAL POSITION

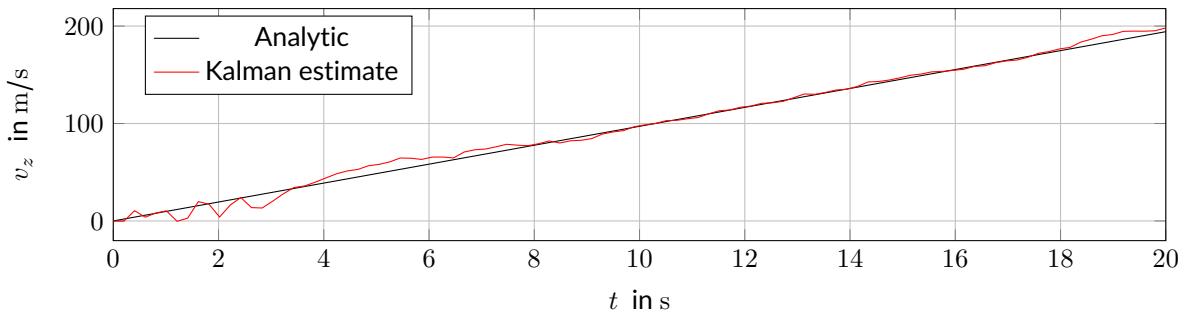


FIGURE 3.12: COMPARISON OF ESTIMATED AND REAL SPEED

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 is small relative to the estimation error then it is performing worse than if it is 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.

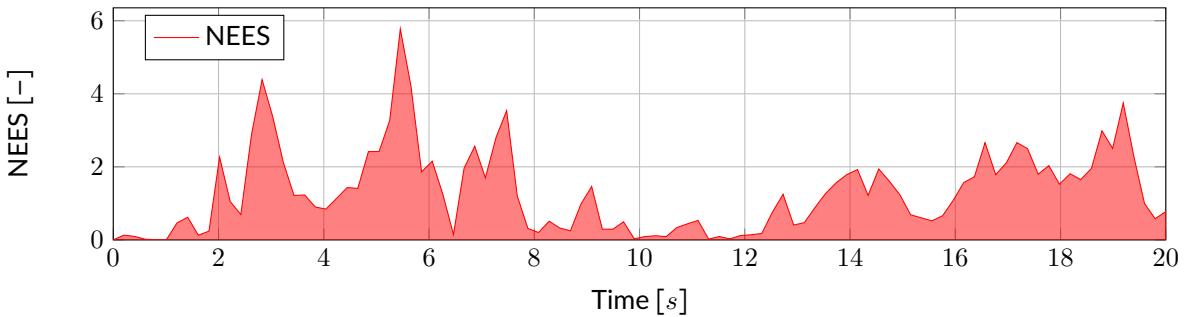


FIGURE 3.13: NEES for Kalman filter

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 [27], hence that this task is called a Coverage Path Planning (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 [55].

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 is not always possible to satisfy all these criteria for a complex environment. Sometimes a priority consideration is required.

Galceran [63] 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 [25] 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. [21] 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 [34] 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. [21] 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.

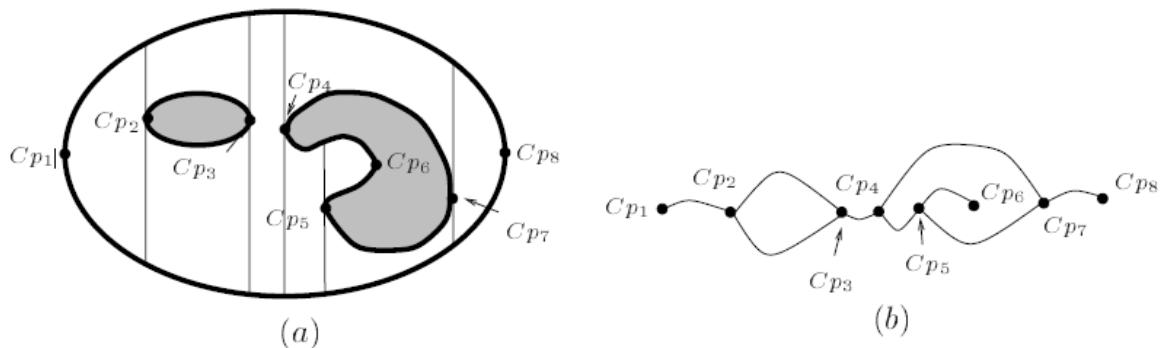


FIGURE 3.14: (a) Exact cellular decomposition, (b) Graph representation

The above describe technique has a minor short-coming, Choset, Acar, Rizzi, et al. [15] 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 Boustrophedon Cellular Decomposition (BCD) approach was introduced.

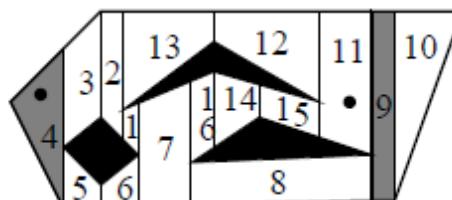


FIGURE 3.15: TRAPEZOIDAL DECOMPOSITION OF BOUND FREE SPACE[15].

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 [55][14][21]. Such as spiral, spike or squarel.

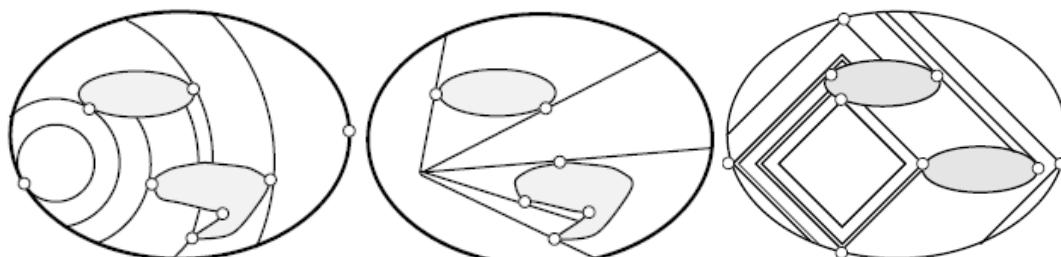


FIGURE 3.16: SPIRAL, SPIKE AND SQUAREL [21]

Once each cell is identified a strategy for the infill is executed, which is described by Huang [19] as coverage paths. Each region [or cell red.] is decomposed into sub-regions, a traveling salesman

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 [19] 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.

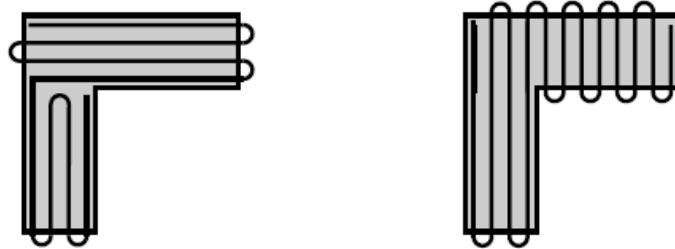


FIGURE 3.17: DIFFERENT SWEEP DIRECTIONS [19]

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 ??, dictates that the crawler encounters unknown obstacles.

ON-LINE MORSE-BASED BOUSTROPHEDON DECOMPOSITION

Acar and Choset [20] 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.

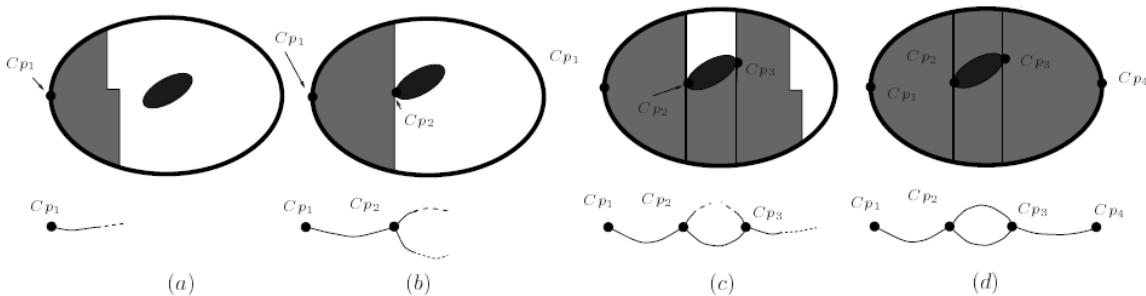


FIGURE 3.18: INCREMENTAL GRAPH CONSTRUCTION [20]

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 [20].

On-line detection of critical points is illustrated by Galceran and Carreras [55]. 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.

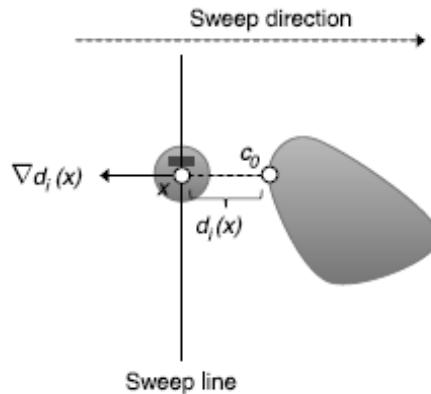


FIGURE 3.19: CRITICAL POINT DETECTING [55]

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 a 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 an 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 Boustrophedon Cellular Decomposition (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 [17] shows how a detector range of a sensor can be utilized which is $r < \delta_s < \infty$ here r which is radius given in [m] and δ_s which is range of a sensor-detector-range given in [m].

Morse-based cellular decomposition combined with Generalized Voronoi Diagram (GVD) describe how this can be exploited to find an optimum coverage pad for a space which consists of vast-cell which, is a cell located in a vast open space, and narrow-cell which, is a cell is located in a narrow space, bound between multiple walls,. 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 is 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 which, is a points where its surface normal of the boundary of the free configuration space is non-smooth,, 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 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 is in a vast environment, since no second boundary is detected and slips into its initial modus.

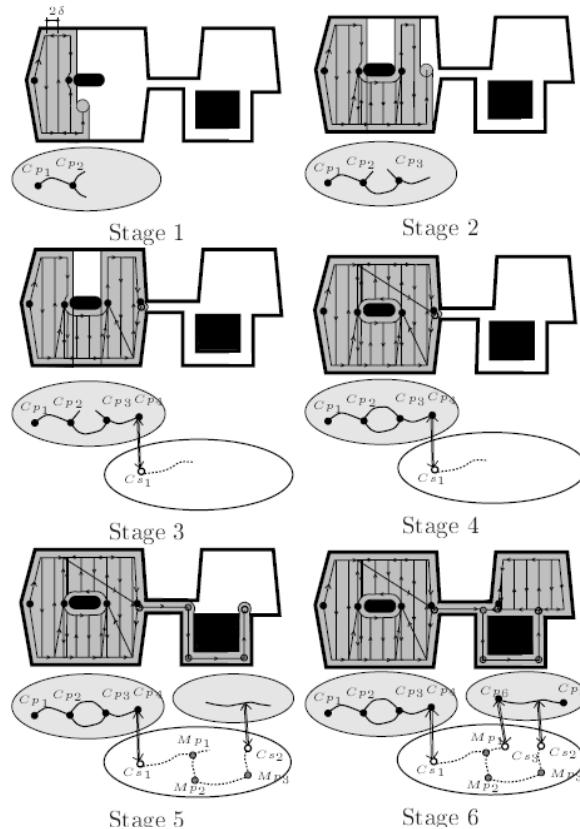


FIGURE 3.20: STAGES OF INCREMENTAL CONSTRUCTION [17]

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 to associated critical points represented as black dots. NARROW-cells is 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 de-

generate 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 [10] 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 is 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 [31].

SLICE DECOMPOSITION

Slice decomposition makes use of simpler landmarks. Galceran and Carreras [55] 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 a abrupt change in the topology between segments in consecutive slices, each slice is a sensor sweep line where the δ_{sx} 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).

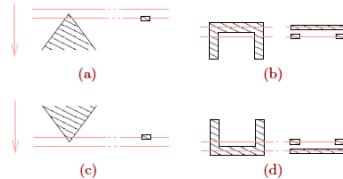


FIGURE 3.21: (a) SPLITTING OF SEGMENTS (b) MERGING OF SEGMENTS [25]

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 a formed by maintaining a list $D_{l,t}$, which consist of active obstacles and free space cells. This list is created via algorithm 3.1. This algorithm consist 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.

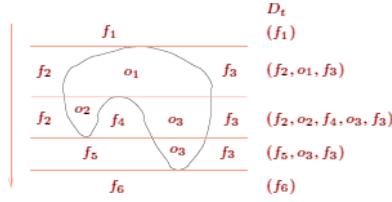


FIGURE 3.22: SLICE DECOMPOSITION GENERATED BY ALGORITHM 3.1 [25]

Wong [31] 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 [31]. 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.

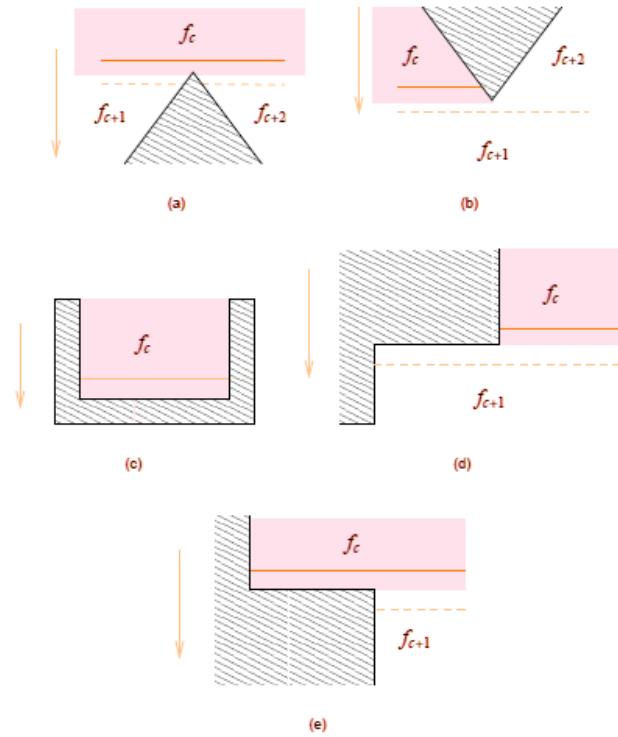


FIGURE 3.23: (a) SPLIT, (b) MERGE, (c) END, (d) LENGTHEN, (e) SHORTEN

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 on (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

```

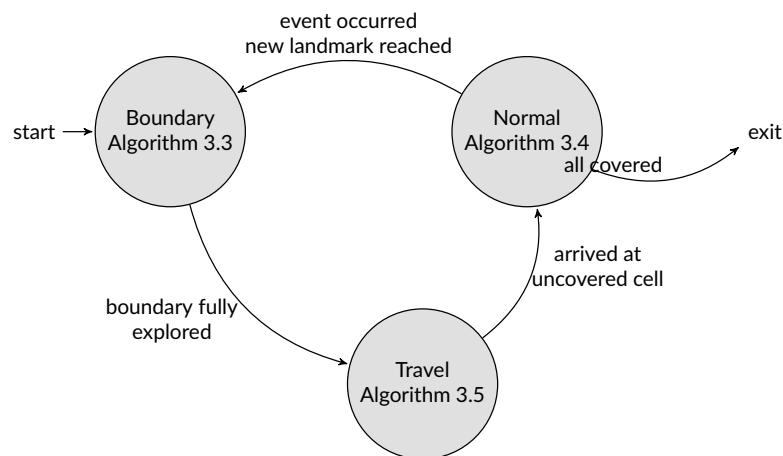
ON-LINE TOPOLOGICAL COVERAGE ALGORITHM

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 in to perform its task in a 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 un each update, based on the new partial topological map [31].

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 [50].

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 is at the end of a strip and the boundary is fully explored it gets in to the state travel, described in algorithm 3.5, otherwise it turns around and continues in the boundary state.

**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 31. 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 ← boundary
7: end procedure
  
```

LISTING 3.5: TRAVEL STATE

```

1: procedure TravelState
2:    $T(n) \leftarrow \text{search } G$ 
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 [31] 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.

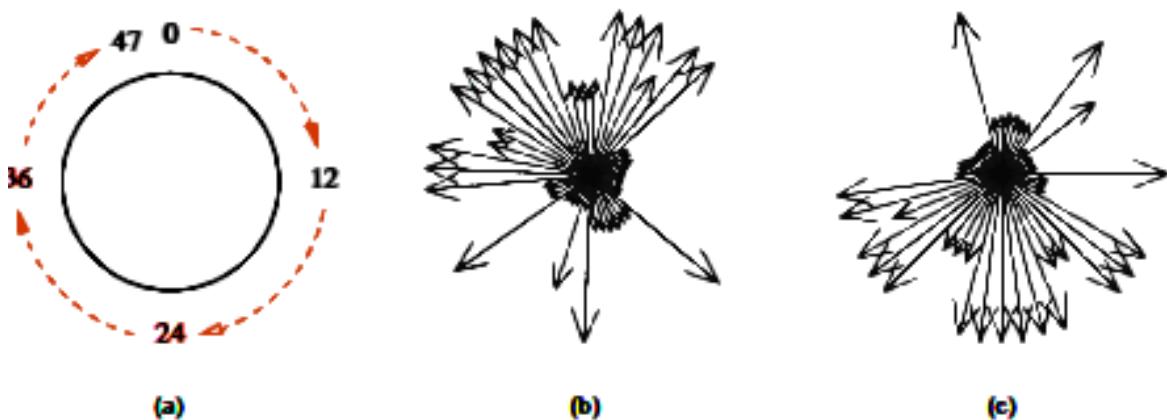


FIGURE 3.25: ROTATION OF SONAR READING TO MOST OCCUPIED DIRECTION [31]

This vector is fed into a Multi-Layer Perceptron (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 is rather free space. The value can either be binary or a probability [62]. These grid cells are typically square in shape, but it is 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 [8] proposed method, whom uses a conventional wave-front algorithm (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

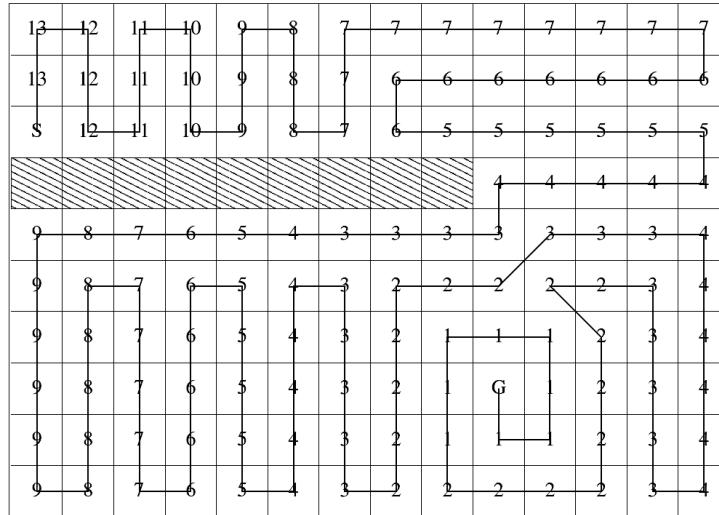


FIGURE 3.26: COVERAGE PATH GENERATED FROM A DISTANCE TRANSFORM [31]

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 [18]. 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.

GRID-BASED COVERAGE USING SPANNING TREES

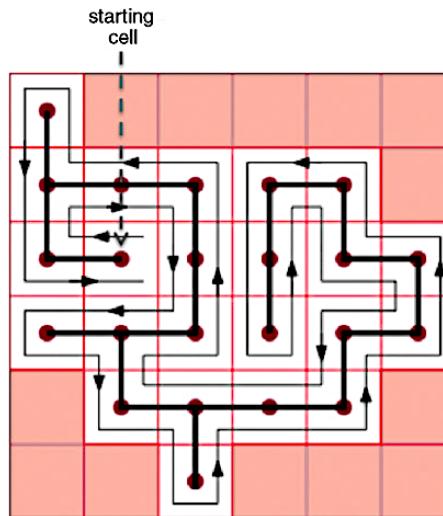
An sub-family of grid-based coverage are systematic spiral paths algorithms. These work by following a systematic spiral spanning tree of a partial grid map. This map is constructed using its on-board sensors [62] and uses two different sizes of grid cells. The smaller grid cell is the same size as the robot. Four of these grid cells then form a mega cell. [31]. The Spiral Spanning Tree Coverage (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

4	3	2	1	G
4	3	S	1	1
4	3	2	2	2
6	5	4	4	4
deadlock				

FIGURE 3.27: DEADLOCK STATE DUE TO INCORRECT STARTING POINT

the current cell has no new neighbors (a mega cell is considered old if at least one of its four smaller cells is covered, it is considered new otherwise). 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 [62].

**FIGURE 3.28: COVERAGE PATH GENERATED WITH THE SPIRAL-STC ALGORITHM [55].****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 [31] and Lee, Baek, Choi, et al. [46] 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.* [24] 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.* [22] 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 [22][38]. 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 [55]. This architecture is illustrated in figure 3.29.

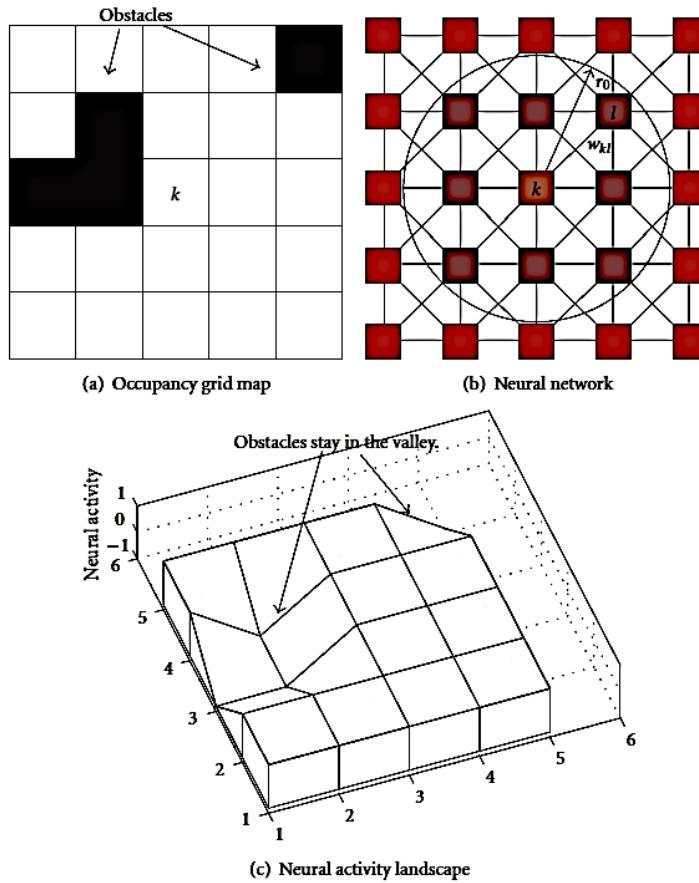


FIGURE 3.29: ARCHITECTURE OF A NEURODYNAMICS MODEL [53]

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 which is the receptive field radius of the k th neuron given in [m] is the 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 [53]. 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 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 $[I_k]^-$ 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.

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

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 [38]. An advantage of this method is that can handle non stationary environments(i.e., dynamically changing obstacles) [55].

4 CHAPTER CONCEPT DESIGN

4.1 STRATEGY DECISION

	onBCD	BCD-GVD	topBCD	onSD2	onSD2NN	STC	sSTC	NN
online	5 ± 0.2	5 ± 0.2	5 ± 0.2	5 ± 0.2	1 ± 0.2	5 ± 0.2	5 ± 0.2	5 ± 0.2
energy eff.	5 ± 0.2	5 ± 0.2	5 ± 0.2	5 ± 0.2	1 ± 0.4	5 ± 0.2	5 ± 0.2	5 ± 0.2

CHAPTER DESIGN VALIDATION 5

5.1 SIMULATION

CHAPTER NOMENCLATURE 6

SIGN	DESCRIPTION	UNIT	PAGE
β	phase factor of a wave	–	9, 10
ω	angular velocity	rad/s	10, 14, 15
θ	angle on the z-axis	rad	14
γ_e	propagation constant	m	9–12
ϕ_{IMU}	roll of the IMU on the Euclidean z-axis	rad	13, 15
γ_m	specific weight of diluted water during dredging	N/m ³	17, 26
$\bar{\epsilon}_N$	mean NEES value	–	22, 23
$\epsilon_{N,k}$	NEES at time k	–	22
Δ_p	pressureloss	Pa	17
σ_s	deviation on position	m	21, 22, 27, 29
$\sigma_{s,m}$	deviation on measured position	m	21, 22
θ_{si}	soft iron axis offset along the z-axis	rad	16, 17
γ_{sw}	specific weight of submerged soil	N/m ³	17
ω_t	true angular velocity	rad/s	15
σ_v	deviation on velocity	m/s	21, 22
$\sigma_{v,m}$	deviation measured on velocity	m/s	21, 22
γ_w	specific weight of water	N/m ³	17
a	acceleration	m/s ²	13
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, 19
$a_{z,k}$	acceleration along the z-axis at time k	m/s ²	19–21
$a_{z,k-1}$	acceleration along the z-axis at time $k - 1$	m/s ²	19
B	a control-input model which is applied to a control vector	–	19, 20
B_d	slow changing component of the signal; this is the gyroscope drift	rad/s	15

SIGN	DESCRIPTION	UNIT	PAGE
C	cost a function	–	25, 26
c	singel cell	–	29
Cp	critical point in graph	–	26
Cp_0	critical point 0 in graph	–	26, 27
Cp_1	critical point 1 in graph	–	26, 27
Cp_2	critical point 2 in graph	–	26, 27
Cp_3	critical point 3 in graph	–	26, 27
Cp_i	critical point i th in graph	–	26
$\Delta d_{1,2}$	distance between two points	m	10
$d_i(x)$	distance between x points and obstacle i	m	26, 27
$D_{l,t}$	line segment at t	–	29
$D_{l,t-1}$	line segment at $t - 1$	–	29
\hat{E}	amplitude of the electric field wave	V/m	9, 10
e	Eulers number 2.7182818284	–	10
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	9, 10
F	force	N	13, 19, 20
G	adjacency graph	–	25, 26, 32
g	standard gravity model	m^2/s	17, 19
G_1	adjacency subgraph 1	–	25, 26
G_2	adjacency subgraph 2	–	25, 26
\hat{H}	amplitude of the magnetic field wave	A	9, 10, 19–22
H_y	is a vector field that describes the magnetic influence of electric charges in relative motion and magnetized materials	A/m	9, 10
i	imaginary unit	–	10
\bar{K}	Kalman gain matrix	–	20–22
$L_{\alpha,\epsilon}$	electromagnetic signal strength	dB	10
m	mass	kg	13, 16
\vec{m}_{hi}	hard iron adjusted vector	T	16, 17

SIGN	DESCRIPTION	UNIT	PAGE
\vec{m}_{si}	soft iron adjusted vector	T	16, 17
$n_{a,l}$	neural activity of a the l th neuron	—	36, 37
n_s	stochastic component of a signal	rad/s	15
n_x	number of elements in the state vector	—	22
p	pressure	Pa	17
\mathbf{P}_0	initialization covariance matrix of state estimation uncertainty	—	19, 20
p_a	atmospheric pressure	Pa	17
\mathbf{P}_k	covariance matrix of state estimation uncertainty	—	20-22
\mathbf{P}_{k-1}	covariance matrix of a priori state estimation uncertainty	—	20
\mathbf{Q}_k	covariance matrix of process estimation uncertainty	—	19, 20
q_{si}	smallest magnitude of a point on the ellipse, and thus the vector of the B-axis	T	16, 17
\mathbf{R}	covariance matrix of state estimation uncertainty	—	19-21
r	radius	m	27
r_0	the receptive field radius of the k th neuron	m	36, 37
r_{si}	greatest magnitude of a point on the ellipse, and thus the vector of the A-axis	T	16, 17
S	a movement in a graph	—	26
s_z	position along the z-axis	m	19, 20, 22
$s_{z,k}$	position along the z-axis at time k	m	19
$s_{z,k-1}$	position along the z-axis at time $k - 1$	m	19-21
$s_{z,m}$	measured position along the z-axis	m	21
$s_{z,m,k}$	measured position along the z-axis at time k	m	20, 21
T	temperature	K	15, 17
t	time	s	10, 14, 19-23, 29
T'	rate of temperature variation	K/s	15
\vec{u}	unit vector	—	24
\vec{u}_k	control inputs	—	19, 20
\vec{v}_k	measurement noise	—	19, 20
v_z	velocity along the z-axis	m/s	19, 20, 23

SIGN	DESCRIPTION	UNIT	PAGE
$v_{z,k}$	velocity along the z-axis at time k	m/s	19
$v_{z,k-1}$	velocity along the z-axis at time $k - 1$	m/s	19-21
\vec{w}_k	process noise	—	19, 20
x	location along the x-axis	m	26, 27, 29, 31, 35
\vec{x}_0	state vector describing the initial state of a system at the k th component of \vec{x}	—	19, 20
$\vec{x}_{g,k}$	ground truth state, describing the real state of a system k th component of \vec{x}	—	22
x_h	hard iron distortion along the x-axis	T	15
\vec{x}_k	state vector describing the state of a system at the k th component of \vec{x}	—	13, 18-22
\vec{x}_{k-1}	a priori state of \vec{x}_k , conditioned on all prior measurements, except the one at time t_k	—	19, 20
x_m	raw magnetometer value along the x-axis	T	15, 16
y	location along the y-axis	m	35
y_1	y-index of greatest magnetic magnitude	T	16, 17
y_h	hard iron distortion along the y-axis	T	15
\vec{y}_k	measured values	—	20
y_m	raw magnetometer value along the y-axis	T	15, 16
z	height	m	17
z_ϵ	height error	m	17
\vec{z}_k	measured values mapped to the state space	—	19-21
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	12–15, 18
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	25, 27
bandwidth	a difference between the upper and lower frequencies in a continuous set of frequencies	11
bathymetric-map	submerged equivalent of an above-water topographic map	18
bitstream	a sequential binary sequence	6
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	23, 25, 28, 33
critical-point	a value of average degree, which separates networks	24, 26–28
cusp-point	are points where its surface normal of the boundary of the free configuration space is non-smooth	27, 28
deadlock-state	a standstill situation, from which the algorithm has no means of escape	34
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	13, 18, 27
draghead	a suction mouth which is dragged across a water body	3
dredgeline	a pipeline which transports excavated slurry	1, 6, 7
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	9
erosion	an action of surface processes (such as water flow or wind) that removes soil	1
geophysical-navigation	navigation using landmarks	18
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	12–15, 18

KEY	DESCRIPTION	PAGE
hard-iron	a constant additive disturbance in the magnetic field of the magnetometer	16
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	6, 12, 14, 18, 19, 21, 23
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	11
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 is a vector field	9
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	12, 13, 15, 18
maxwell	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	9
monotonically	a function between ordered sets that preserves or reverses the given order	37
morse-function	a function for which all critical points are non-degenerate and all critical levels are different	24, 25
narrow-cell	a cell is located in a narrow space, bound between multiple walls	27, 28
odometry	the use of data from motion sensors to estimate change in position over time	14
off-line	algorithm which plans an optimal path ahead of time, thus which needs to know the environment <i>a priori</i>	18, 23, 26
on-line	an algorithm which has the ability to adapt when needed	23, 24, 30
optimal-path	a sequence of steps which are optimized	23, 25
polarized-plane	is a confinement of the electric field vector or magnetic field vector to a given plane along the direction of propagation	9

KEY	DESCRIPTION	PAGE
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	12, 17, 18
quaternion	a number system that extends the complex numbers. They were first described by Irish mathematician William Rowan Hamilton in 1843 and applied to mechanics in three-dimensional space. A feature of quaternions is that multiplication of two quaternions is non-commutative. Hamilton defined a quaternion as the quotient of two directed lines in a three-dimensional space or equivalently as the quotient of two vectors	19
reeb-graph	a mathematical object reflecting the evolution of the level sets of a real-valued function on a manifold	27
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
soft-iron	a result of material that distorts the magnetic field of magnetometer, but does not necessarily generate its own magnetic field	16
trapezoidal-rule	a technique for approximating the definite integral	14
traveling-salesman-problem	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 is 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, 4, 6, 7, 30, 43
vast-cell	a cell located in a vast open space	27, 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

8 CHAPTER ACRONYMS

KEY	DESCRIPTION	PAGE
ADC	Analog Digital Conversion	14, 15
AOD	Autonomous Operating Dredgebot	1, 2
ASD	Auger Suction Dredger	3
AUV	Autonomous Underwater Vehicle	7, 18
AW	Acoustic Waves	12
BCD	Boustrophedon Cellular Decomposition	24, 26–28
CIP	Common Industrial Protocol	8
CPP	Coverage Path Planning	6, 23
CSD	Cutter Suction Dredger	3
EMW	Electromagnetic Waves	8–10, 12
GPS	Global Positioning System	6, 12, 14, 18
GVD	Generalized Voronoi Diagram	27
IEEE	Institute of Electrical and Electronics Engineers	8, 10, 11
IMU	Inertial Measurement Unit	13
IoT	Internet of Things	11
LBL	Long Base Line	18
LLC	Logical Link Control	8, 10
LQE	Linear Quadratic Estimation	18
LQG-MP	Linear-Quadratic Gaussian Motion Planning	18
MAC	Media Access Control	8, 10
MEMS	Micro Electro Mechanical System	13–15
MLP	Multi-Layer Perceptron	33
MTI	IHC MTI B.V.	17
NEES	Normalized Estimated Error Squared	22, 23
RESL	Robotic Embedded Systems Laboratory	11

KEY	DESCRIPTION	PAGE
ROV	Remote Operated Vehicle	7
RRT	Rapidly exploring Random Trees	18
SLAM	Simultaneous Localization And Mapping	18
Spiral-STC	Spiral Spanning Tree Coverage	34, 36
SW	Sound Waves	12
TCP	Transmission Control Protocol	8
TIR	Total Internal Reflection	6
TRN	Terrain Relative navigation	18
TSHD	Trailing Suction Hopper Dredger	3
UDP	User Datagram Protocol	8
USBL	Ultra Short Base Line	18

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] 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.
- [8] H. Choset and P. Pignon, "Coverage Path Planning: The Boustrophedon 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).
- [9] Joseph C. Palais, *Fibre Optic Communication*. Prentice Hall, 1998.
- [10] 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).
- [11] 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.
- [12] 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).
- [13] J. G. Webster, *The Measurement, Instrumentation, and Sensors Handbook*, en. Springer Science & Business Media, 1999, Google-Books-ID: b7UuZzf9ivlC, isbn: 978-3-540-64830-7.
- [14] H. Choset, "Coverage of Known Spaces: The Boustrophedon 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).
- [15] 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).

- [16] 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.
- [17] 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).
- [18] 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).
- [19] 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).
- [20] 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).
- [21] 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).
- [22] 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).
- [23] 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>.
- [24] 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).
- [25] 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).
- [26] 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).
- [27] H. M. Choset, *Principles of Robot Motion: Theory, Algorithms, and Implementation*, en. MIT Press, 2005, isbn: 978-0-262-03327-5.
- [28] Edward Tucholski, *Underwater Acoustics and Sonar - SP411*. 2006.
- [29] 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).
- [30] g. Welch and G. Bishop, "An introduction to the Kalman Filter", en, Jul. 2006.
- [31] 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).

- [32] 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>.
- [33] 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>.
- [34] 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).
- [35] L. Tetley and D. Calcutt, *Electronic Navigation Systems*, en. Routledge, Jun. 2007, Google-Books-ID: mybjgp2gGRsC, isbn: 978-1-136-40724-6.
- [36] 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).
- [37] 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).
- [38] 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).
- [39] Training Institute for Dredging, *Ingewijden Training*, en. MTI Holland, 2008.
- [40] Hagman, Elias, "Design of a High Speed, Short Range Underwater Communication System", PhD thesis, Eidgenossische Technische Hochschule Zurich, 2009.
- [41] M. Ainslie, *Principles of Sonar Performance Modelling*, en. Springer Science & Business Media, Sep. 2010, isbn: 978-3-540-87662-5.
- [42] 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).
- [43] 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).
- [44] s. Jiang and S. Georgakopoulos, "Electromagnetic Wave Propagation into Fresh Water", en, Apr. 2011.
- [45] 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).
- [46] 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).
- [47] J. R. Nistler and M. F. Selekwa, "Gravity compensation in accelerometer measurements for robot navigation on inclined surfaces", *Procedia Computer Science*, Complex adaptive systems, 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).
- [48] F. M. White, *Fluid Mechanics*, en. McGraw Hill, 2011, isbn: 978-0-07-352934-9.
- [49] 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).

- [50] 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).
- [51] 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/>.
- [52] 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).
- [53] 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).
- [54] 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.
- [55] 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>.
- [56] 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).
- [57] 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).
- [58] 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).
- [59] C. Van Den Berg, *IHC Merwede Handbook for Centrifugal Pumps and Slurry Transportation*. MTI Holland, 2013.
- [60] J. Claus, "Design and Development of an Inexpensive Acoustic Underwater Communications and Control System", en, PhD thesis, Florida institute of technology, Florida, 2014.
- [61] P. Freitas, "Evaluation of Wi-Fi Underwater Networks in Freshwater", PhD thesis, UNIVERSIDADE DO PORTO, Porto, 2014.
- [62] E. Galceran, "Coverage path planning for autonomous underwater vehicles", en, PhD thesis, Universitat de Girona, 2014.
- [63] —, "Towards Coverage path planning for autonomous underwater vehicles", en, PhD thesis, Universitat de Girona, 2014.
- [64] G. van der Schrieck, *Dredging Technology - Guest lecture notes CIE5300 Issue 2014*, en. GLM van der SCHRIECK BV, 2014.
- [65] 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).

- [66] 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).
- [67] 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).
- [68] D. C. Giancoli, *Physics Principles with Applications*, en. Pearson Education, Limited, Jan. 2015, Google-Books-ID: jYlyngEACAAJ, isbn: 978-1-292-05712-5.
- [69] 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.
- [70] 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).
- [71] Jolectra, *PLC diagrams Besturingspaneel container - 216.156.P1*. Jun. 2016.
- [72] W. Vlasblom, "Designing Dredging Equipment", TU Delft, Delft University of Technology, Lecture notes, 2016.
- [73] Roger R Labbe jr, *Kalman and Bayesian Filters in Python*. Jan. 2017.
- [74] R. v. d. Wetering, "Ihc mti crawler - final report", IHC MTI B.V., report, 2018.

APPENDICES

A APPENDIX CRAWLER PARTLIST

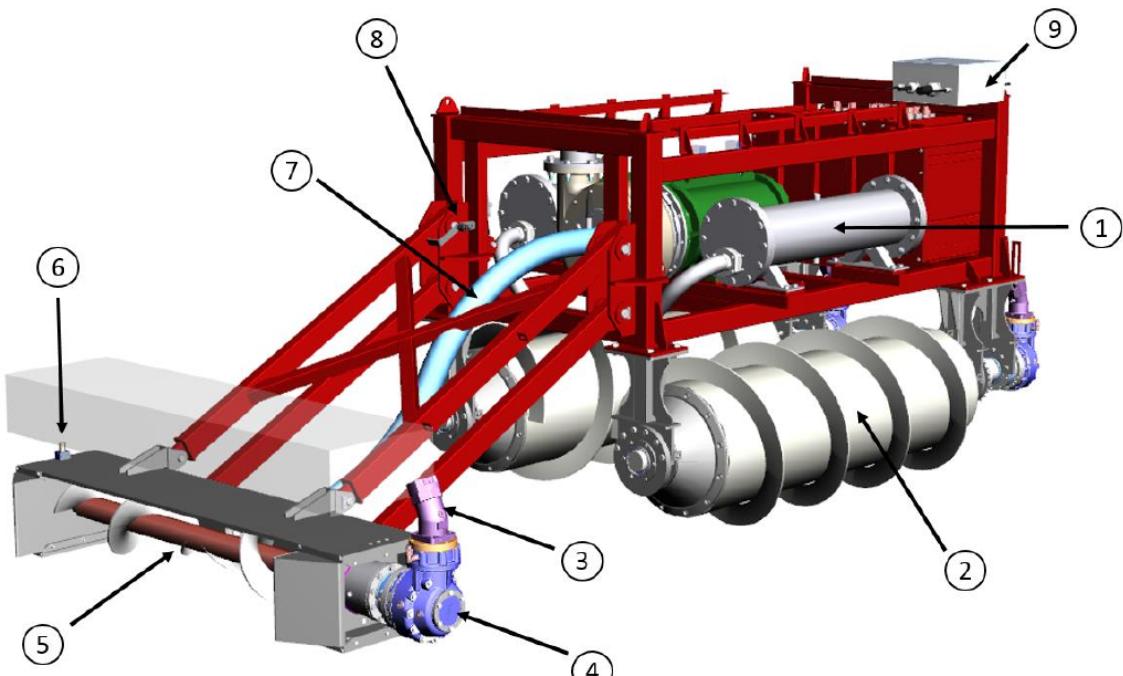


FIGURE A.1: MAJOR PARTLIST SIDEVIEW [74]

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

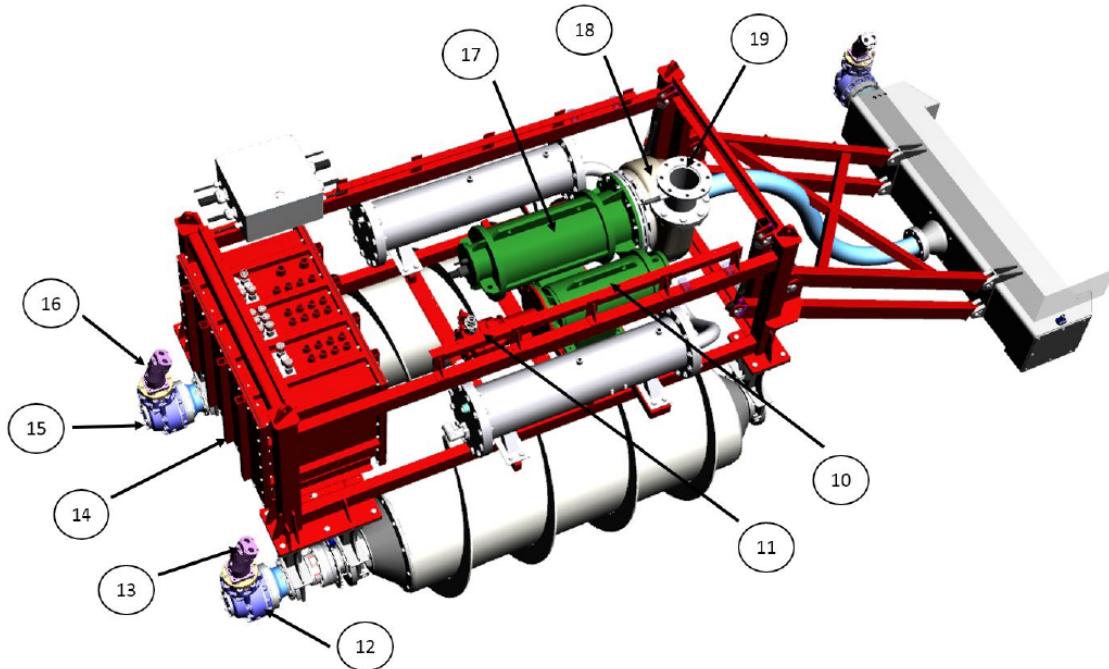


FIGURE A.2: MAJOR PARTLIST TOPVIEW [74]

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

B APPENDIX 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 KALMAN EXAMPLE FALLING BALL

LISTING C.1: FALLING BALL EXAMPLE

```
import matplotlib.pyplot as pl

pl.rcParams['legend.loc'] = 'best'
import numpy as np
from numpy import dot
from scipy.linalg import inv

def build_real_values():
    num_of_time_steps = 100
    dt = 0.2
    t = np.linspace(start=0., stop=num_of_time_steps * dt, num=num_of_time_steps)
    s = np.zeros((num_of_time_steps, 1))
    v = np.zeros((num_of_time_steps, 1))
    a = np.zeros((num_of_time_steps, 1))

    a = np.ones((num_of_time_steps, 1)) * 9.81 # Build acceleration profile
    for i in range(1, num_of_time_steps):
        v[i] = v[i - 1] + a[i] * dt # Build speed profile
        s[i] = s[i - 1] + v[i] * dt + 0.5 * a[i] * dt ** 2 # Build position profile

    return [t, dt, s, v, a]

def build_measurement_values(t, S):
    y = np.zeros((t.size, 2, 1))
    S_m = np.random.normal(0., 50, (len(S), 1)) + S
    y[:, 0] = S_m
    return y

def build_control_values(t, a):
    u = np.ones((t.size, 2, 1)) * a[0]
    return u

def init_kalman(t, dt):
    phi_s = 5

    F = np.array([
        [1., dt],
        [0., 1.]
    ])

    B = np.array([
        [0.5 * dt ** 2, 0.],
        [0., dt]
    ])

    H = np.array([
        [1., 0.],
        [0., 0.0]
    ])
```

```

Q = np.array([
    [(1 / 3) * dt ** 3, 0.5 * dt ** 2],
    [0.5 * dt ** 2, dt]
]) * phi_s

R = np.array([
    [40**2, 0.0],
    [0.0, 15**2]
])
v = np.random.normal(0, 25e-2, (t.size, 2, 1))
w = np.random.normal(0, 25e-2, (t.size, 2, 1))
return [F, B, H, Q, R, v, w]

def kalman(t, kalman_values, u, z, error):
    x = np.zeros((t.size, 2, 1))
    P = np.zeros((t.size, 2, 2))
    P[0, :, :] = np.array([
        [error[0] ** 2, 0.],
        [0., error[0] ** 2]
    ])

    xhat = np.zeros((t.size, 2, 1))
    y = np.zeros((t.size, 2, 1))

    F = kalman_values[0]
    B = kalman_values[1]
    H = kalman_values[2]
    Q = kalman_values[3]
    R = kalman_values[4]
    v = kalman_values[5]
    w = kalman_values[6]

    K = np.zeros((t.size, 2, 2))

    for k in range(1, t.size):
        xhat[k] = dot(F, x[k - 1]) + dot(B, u[k]) + w[k]
        Phat = dot(F, dot(P[k - 1], F.T)) + Q

        y[k] = z[k] - dot(H.T, xhat[k])

        S = dot(H.T, dot(Phat, H)) + R
        S = np.linalg.inv(S)
        K[k] = dot(Phat, dot(H, S))
        x[k] = xhat[k] + dot(K[k], y[k])

        P[k] = dot(np.eye(2) - dot(K[k], H.T), Phat)
    return [x, K, P, xhat, z]

def NEES(xs, est_xs, ps):
    est_err = xs - est_xs
    err = np.zeros(xs[:, 0].size)
    i = 0
    for x, p in zip(est_err, ps):
        err[i] = (np.dot(x.T, inv(p)).dot(x))
        i += 1
    return err

def plot_results(t, x, xground, a, u, y, K, P, xhat, z, nees):
    pl.figure()
    pl.subplot(311)
    pl.plot(xground[:, 0])

```

```

pl.plot(x[:, 0], '+')
pl.plot(z[:, 0], '.')
pl.subplot(312)
pl.plot(xground[:, 1])
pl.plot(xhat[:, 1], 'o')
pl.plot(x[:, 1])
pl.subplot(313)
pl.plot(nees)
pl.tight_layout()
pl.savefig('fallingBall.png')
pl.show()

def save_results(t, xground, x, z, nees):
    # save csv
    datPos = np.zeros((t.size, 4, 1))
    datPos[:, 0] = t.reshape((t.size, 1))
    datPos[:, 1] = xground[:, 0]
    datPos[:, 2] = x[:, 0]
    datPos[:, 3] = z[:, 0]
    np.savetxt('fallingBallPos.dat', datPos, delimiter=',')

    datSpeed = np.zeros((t.size, 3, 1))
    datSpeed[:, 0] = t.reshape((t.size, 1))
    datSpeed[:, 1] = xground[:, 1]
    datSpeed[:, 2] = x[:, 1]
    np.savetxt('fallingBallSpeed.dat', datSpeed, delimiter=',')

    datNEES = np.zeros((t.size + 1, 2, 1))
    datNEES[0:t.size, 0] = t.reshape((t.size, 1))
    datNEES[0:t.size, 1] = nees.reshape(t.size, 1)
    datNEES[-1, 0] = t[-1]
    datNEES[-1, 1] = 0.0
    np.savetxt('fallingBall_NEES.dat', datNEES, delimiter=',')

    np.savetxt('fallingBall_meanNEES.dat', [np.mean(nees)], delimiter=',')

def main():
    [t, dt, s, v, a] = build_real_values()
    z = build_measurement_values(t, s)
    u = build_control_values(t, a)
    [F, B, H, Q, R, vv, w] = init_kalman(t, dt)
    error = [40, 10]
    kalman_values = [F, B, H, Q, R, vv, w]
    x, K, P, xhat, y = kalman(t, kalman_values, u, z, error)
    xground = np.zeros(x.shape)
    xground[:, 0] = s
    xground[:, 1] = v
    nees = NEES(xground, x, P)
    print(np.mean(nees))
    save_results(t, xground, x, z, nees)
    plot_results(t, x, xground, a, u, y, K, P, xhat, y, nees)

if __name__ == '__main__':
    main()

```