

Eva Kristiansen

Strategies for search and detection of acoustic transmitters using unmanned surface vehicle

Masteroppgåve i Kybernetikk og robotikk

Veileder: Jo Arve Alfredsen

Juni 2019

Eva Kristiansen

Strategies for search and detection of acoustic transmitters using unmanned surface vehicle

Masteroppgåve i Kybernetikk og robotikk
Veileder: Jo Arve Alfredsen
Juni 2019

Noregs teknisk-naturvitskaplege universitet
Fakultet for informasjonsteknologi og elektroteknikk
Institutt for teknisk kybernetikk

Samandrag

Denne oppgåva tar for seg baneplanlegging i forbindelse med søk etter akustiske sendarar. Akustisk telemetri har gjort store framsteg dei siste åra, i samband med miniatyrisering av elektronikk. Signal frå akustiske sendarar i fisk registrerast av mottakarar som er festa i stasjonære bøyer, eller i ulike bevegelige plattformer. I denne oppgåva er målet å lage strategiar for å effektivt gjennomsøkje eit område for slike akustiske sendarar ved hjelp av eit autonomt overflatefartøy.

Søkjeproblemet vert formulert i et diskret kart, og tre ulike metodar vert implementert og testa i simulasjon. Den første metoden er ein metode som prøver å dekkje så mykje som mulig av området. Den andre metoden er ein grådige metode, og velgjer heile tida retningar som lokalt optimaliserer sannsynet for deteksjon av fisk. Den tredje metoden er ein tilfeldig strategi med inspirasjon frå biologien.

I simulasjon er begge dei systematiske stiane merkbar betre enn den tilfeldige, i den forstand at dei finn fleire akustiske sendarar på same tid. Den grådige metoden er litt betre enn den dekkjande metoden i dei tilfella der den grådige metoden har riktig informasjon og måla er i bevegelse.

Ein Matlab-applikasjon som kan lage dekkjande stiar for eit geografisk område er implementert i forbindelse med denne oppgåva. Denne applikasjonen kan sende veipunkt til Neptus, som er eit kontrollsenster for styring av autonome farkostar.

Abstract

This thesis considers path planning for search and detection of acoustic transmitters. In fish telemetry, signals from acoustic transmitters attached to fish are registered. Recent advances in acoustic telemetry systems combined with the growing availability of autonomous systems is predicted to increase the yield of biological data from fish telemetry and our understanding of fish ecology. In this thesis, the goal is to devise paths for an unmanned surface vessel that efficiently search an area for acoustic transmitters.

The search problem is formulated in a discrete map, and three different methods for path planning are implemented and tested in simulation. The first method is a path that aims to cover the search area. The second method is a greedy path planner, that aims to maximise the local probability of detection at each step. The third method is a random path generator inspired by biology.

Both systematic paths have better performance than the randomised one in simulations in terms of finding the most targets. The greedy method performs slightly better in the case where it has the correct information and the targets are moving.

A MatLab-application was also made within the scope of this thesis. The application calculates paths that aim to cover an area, and broadcasts the paths as way-points to Neptus, which is a control centre for autonomous vessels.

Preface

This thesis concludes my masters degree in Cybernetics and Robotics at the Norwegian University of Science and Technology. The topic is path planning for search of fish with acoustic tags, and is part of a greater vision where autonomous vessels can search for, detect and track fish. It is truly an exciting future we are stepping into, where autonomous machines are becoming increasingly available and useful.

I would like to thank my supervisor, Jo Arve Alfredsen, for formulating such an exciting assignment, as well as showing genuine interest through the semester and always being helpful. Our conversations have been truly motivating. Also, thank you João Fortuna for helping with and supplying base code for the implementation of communication between MatLab and Neptus on a very short notice.

I would also like to thank my fellows at the office for enjoyable company throughout the semester, who have made the work on this thesis a lot more manageable, and especially Andreas T. Eggesvik, who is a great friend and sparring partner. Lastly, thank you Fredrik Chrislock, for always supporting me, and my mother, Eli Kristiansen for the help with proofreading this thesis.

Eva Kristiansen
Trondheim, June 2019

Contents

Samandrag	1
Abstract	i
Preface	ii
Table of Contents	iv
List of Tables	v
List of Figures	viii
1 Introduction	1
1.1 Background	1
1.2 Previous work	3
1.3 Motivation	5
1.4 Problem description	5
1.5 Outline	6
2 Theory	7
2.1 Acoustic fish telemetry	7
2.1.1 Acoustic telemetry systems	7
2.1.2 Detection range	8
2.2 Digital maps	11
2.3 Coordinate systems and reference frames	12
2.3.1 Geodetic datums	12
2.3.2 Map projections	12
2.3.3 Earth centered reference frames	12
2.3.4 Tangent plane reference frames	13
2.3.5 Body-frame	13
2.4 Guidance, navigation and control	14

2.4.1	Vessel model	14
2.4.2	Line-Of-Sight Guidance	14
3	Search problem	17
3.1	Search space	17
3.2	Searcher characteristics	19
3.3	Target characteristics	19
3.4	Sensor model	19
3.5	Objective	23
4	Path generation	25
4.1	Map representation	26
4.1.1	Occupancy grid sampling	26
4.1.2	Search map sampling	28
4.2	Complete coverage path	29
4.3	Greedy search	31
4.4	Lévy flight path	32
4.5	Way-point generation from search paths	33
4.6	Speed assignment	34
4.7	Multiple searchers	34
5	Simulation	37
5.1	Target position data	40
5.2	USV model and control	41
5.3	Sensor model	42
5.4	Search paths	42
5.5	Results	48
5.6	Paths for multiple searchers	53
6	Implementation	55
6.1	LSTS toolchain	55
6.1.1	MatLab IMC communication	55
6.2	Path generator application	55
7	Discussion	59
7.1	Future work	60
	Bibliography	62
A	Master thesis assignment	67
B	Matlab IMC communication	69
B.1	Plan specification IMC message	69
B.2	Sending IMC messages in Matlab	70

List of Tables

2.1	System parameters for TBR700 and Thelma Biothel 13mm tag	10
5.1	Simulation parameters	48
5.2	Path metrics for the three search paths	48
5.3	Target detection metrics for greedy path	49
5.4	Target detection metrics for coverage path	49
5.5	Target detection metrics for Lévy flight path	49

List of Figures

1.1	USV Otter with mounted hydrophone	3
2.1	Absorption by frequency, from [1] (adapted from [2])	9
2.2	Noise in the sea, from [1](adapted from [3])	9
2.3	Transmission loss with distance, from [1]	10
2.4	ECEF and ENU coordinate systems	13
3.1	Lateral range reduction due to target swimming depth	20
3.2	Range test in Gaulosen	21
3.3	Histogram of detected signal ratio by range	22
4.1	Polygonal map of Ulvenvatnet	25
4.2	Occupancy grid and search map. White cells are open (0).	26
4.3	Occupancy grid sampling	27
4.4	Square in circle with radius R	28
4.5	Distance transforms computed from centre cell with different metrics	29
4.6	Distance transform in search map	30
4.7	Paths calculated with and without path length restriction. Numbers indicate order of visitation. Start and goal point in the green dot.	31
4.8	Greedy search in interest value map where dark blue represents low values and yellow represent high values	31
4.9	Lévy flight search path	32
4.10	Search space divided in two by k means clustering	35
5.1	Map of Hornindalsvatnet	37
5.2	Discrete potential field for fish movement and resulting migration path for two fish	39
5.3	Interest value map with high values within 200m of shore, and then decreasing with distance to shore	40
5.4	Target position data	41
5.5	Search map with start and goal point	43

5.6	Coverage search path	44
5.7	Interest value map for Hornindalsvatnet. Yellow = high value, blue = low value	45
5.8	Greedy search path	46
5.9	Examples of Lévy flight paths	47
5.10	Heatmap of coverage	51
5.11	Successfully navigating between known obstacles. Number represent way-points	52
5.12	Coverage-like path for 3 searchers. Start and goal points marked in red	53
6.1	Generate coverage way-points app.	56
6.2	Send way-points	57
6.3	Way-points in Neptus	57

Introduction

1.1 Background

Environmental sustainability is an important topic on the current agenda, politically, scientifically, commercially and culturally. Laws and regulations for fishery and aquaculture industry are meant to protect the ocean and marine ecosystems, and to preserve the resources we have today for future generations. To make well informed decisions about laws and regulations, knowledge of the ocean and its inhabitants is needed. A key part of understanding life in the sea is the movement and behaviour of aquatic animals. A conventional method for marine animal research consists of marking and recapturing. While this gives information on coarse movement (marking and recapture site), as well a means of estimating population sizes, a more fine-scale movement tracking method is of interest, and telemetry methods have been developed since the 1950s.

Aquatic animal telemetry, or fish telemetry, has emerged as a way to track and monitor the movements of marine animals. The word telemetry means remote measuring, and is derived from Greek "tele", meaning remote, and "metron", meaning to measure [4]. There are two main methods of aquatic animal telemetry; acoustic telemetry and satellite telemetry [5]. For both approaches electronic transmitters called tags are attached to an aquatic animal externally, inserted in the stomach or implanted surgically. Fish are the most commonly tagged animals [5]. This tag may record and transmit various data. In the simplest case, this can involve transmitting a unique identification code for the specimen. In satellite telemetry the data is transmitted to land-based receivers via orbiting satellites each time the carrier specimen rises to the surface, or by detaching and floating to the surface for transmission. This transmission requires tags of a certain size, that are only suitable for studies of larger animals, such as sea turtles, sharks, birds and large fish. Whilst satellite telemetry is limited by tag size, it enables tracking in wide areas by the satellite network. Tags for acoustic telemetry are noticeably smaller, and allow for tracking of both younger specimen and smaller species [5]. However, as data is transmitted through the use of acoustic signals, the receivers, hydrophones, are required to be within a relatively close

range of the tags, typically less than 1000 m. The receivers are placed in moored buoys or moving vessels. The range limitation has focused the use of acoustic telemetry to partially enclosed bodies of water, such as rivers and lakes, and coastal environments such as fjords [5]. An example of this usage can be found in a study of migration patterns of wild and hatchery-reared Atlantic salmon smolts [6]. In this study, moored receivers were placed in the river Lærdalselva, and the fjord it connects to, Lærdalsfjorden.

Receivers have traditionally been placed on manually steered surface vessels to track tagged specimen. In this approach, a researcher continuously steers towards the strongest signal direction, using a directional receiver (hydrophone) [7]. Traditional tracking and mapping from a research vessel can be found in for instance a study of giant trevally (*caranx ignobilis*) in Hawaii [8], and a study of striped bass in New Jersey [9]. Although this approach is clearly useful for mapping and tracking in combination with the use of moored receivers, it is labour intensive. In other studies, moving receivers have been attached to large marine animals. One study applying this strategy aimed to research the impact of seal predation on fish stocks [10]. The authors of the study conclude that the results provide a proof of concept that seals fitted with GPS and acoustic receivers may provide information on fish outside the reach of fixed receiver arrays.

As autonomous vessels have become more available in the recent years, their capabilities for carrying receivers have also been researched. For instance, active autonomous tracking and following of a tagged leopard shark has been shown using an autonomous underwater vessel (AUV) in SeaPlane Lagoon, Port of Los Angeles, CA in the fall of 2011 [11]. AUVs have also been used for mapping purposes, where the goal is to detect presence of tagged specimen, as opposed to following them. This can be seen in a study of atlantic sturgeon, where an AUV glider was deployed as an extension of a moored receiver array [12]. The glider followed a large zig-zag pattern along a 120 km stretch of coastline over the course of 79 days. An AUV following zig-zag patterns has also been used to map occurrences of atlantic sturgeon and their habitat in the Hudson river [13]. In this study, the data from detections of acoustic tags is combined with readings from echogram side scans. To provide overlapping for these side scans, the AUV was also set to move in lawn-mower or parallel track patterns.

A natural next step when receiving a signal from an acoustic tag in a mapping scenario, is to move in a pattern that maximises the abilities to more accurately locate the source tag. Alternatively, an estimate can be calculated a posteriori by the detections received on the original search path. The disadvantage of this, is that the geometry of the localisation problem will be limited by the search path. This has previously been addressed by the design of a payload to override a native control system of an AUV [14].

Autonomous active tracking and localisation of acoustic fish tags is also a topic of research at NTNU. The research is mostly focused on the use of omnidirectional receivers. A concept where unmanned surface vessels carrying omnidirectional receivers are used to track acoustic tags has been presented [15]. Further, an eXogenous Kalman filter is has been developed for the tag position estimation task, and compared with a benchmark

extended Kalman filter [16]. Master students have also been working with autonomous moving platforms in acoustic telemetry. One master thesis [17] considers an approach towards tracking acoustic tags in open ocean environments using four unmanned surface vessels, similar to the concept in [15, 16]. In the master thesis [18], a single omnidirectional receiver attached to an AUV is used to estimate the position of a stationary acoustic tag, by moving the receiver in various strategic patterns. The Department of Engineering Cybernetics at NTNU has also recently acquired 4 unmanned surface vessels of the kind USV Otter from Maritime Robotics [19]. Additional hardware has in the spring 2019 been added to the Otter USV at the department, and a hydrophone has been mounted as shown in fig. 1.1



Figure 1.1: USV Otter with mounted hydrophone

1.2 Previous work

Search theory was first developed during the second world war. In the first half of 1942, u-boats sank 397 allied merchant ships in the Atlantic [20]. ASWORG (Antisubmarine Warfare Operations Research Group), was created as a part of the United States navy to help counter the u-boat threat. This is where search theory, a field within operations research, has its beginning. The resulting work in search theory was presented in the report *Search and Screening* [21], which defines central terms within the field and lays a theoretical foundation. Movement patterns commonly used in search such as creeping-line-patterns, parallel sweeps and barrier search is described here. Parallel and creeping line patterns resemble what has previously been referred to as lawnmower patterns.

In 1975 the Lanchester Prize, a prize given yearly to the best contribution within operations Research, was awarded to Lawrence D. Stone for *Theory of Optimal Search* [22]. One of the main goals of this publication was to make a unified and comprehensive presentation of previous results within search theory. This book dealt mainly with stationary target search problems, and a method of Lagrange multipliers and maximising Lagrangians is used to solve most of the optimisation problems. In the following decades, research on search theory expanded to further include problems with moving targets and path-constrained problems, where the movements of the searcher are limited and a path is to be calculated. The book *Optimal Search For Moving Targets* [23] collects and presents the results of the research conducted since 1975's *Theory of Optimal Search*.

A common application of optimised search today is search and rescue. An example of this is the U.S. Coast Guards system, Search and Rescue Optimal Planning System (SAROPS) [24], which has been in use since January 2007. More recently, interest in using autonomous vehicles in search and rescue has increased, due to recent technological advances. In [25], a mixed integer programming problem is proposed to solve optimal path planning for multiple agents searching for a stationary target. The authors point out that one of the main advantages of their approach is the ability to calculate an optimality gap. In [26], three different approaches to path planning in search and rescue are taken - a complete coverage approach, a genetic algorithm and a local hill climbing method. None of these methods involve optimal solutions, and as such do not provide metrics on optimality gaps. However, the methods seek to solve the search and rescue problem in an intelligent way to minimise search time. The methods are all applied to a discretized map where each cell has a probability of hosting the search target. The complete coverage approach aims for complete coverage of all cells of nonzero detection probability. The local hill climbing approach is a greedy algorithm, that always moves in the direction of highest detection probability. The genetic algorithm uses cumulative detection of probability for a path as a fitness measure.

The complete coverage path problem (CPP) mentioned above is a known problem from robotics. CPP, in short, is the problem of calculating a path ensuring total coverage of a given area, whilst avoiding known obstacles. A full definition of the problem can be found in one of the earliest work on complete coverage paths [27], and includes additional concerns such as avoiding overlapping paths and the path consisting of simple motion trajectories. The complete coverage problem, and variations of it, is found in for example vacuum cleaner robots, lawn movers and de-mining robots. Intuitively, if there are resources available to cover an entire search area, and the search target is equally likely to be anywhere in the search area, the search problem will become the problem of complete coverage. Complete coverage is also useful when optimal search strategies are used in a discrete way to cover a continuous space, as to cover a selected "cell" on the continuous space. This approach has for instance been used for search in a built environment, such as an office floor [28]. A 2013 survey [29], provides an overview of coverage path planning methods. This includes both one and multiple agents, and environments known a priori and environments that are only known through the sensory input of the agent. Coverage path planning known as a problem of high complexity, the lawnmower problem, coverage

path planning without obstacles is NP-hard [30].

While randomised methods do not guarantee any form of coverage, they may still be a viable approach towards search in the case of for instance limited search resources. The Lévy foraging hypotheses suggests that predators in the wild forage in a specific type of random patterns known as Lévy walks or Lévy flights. A Lévy walk consists of clusters of small steps and longer steps in between. The step length is calculated from the probability distribution function in eq. (1.1), where $1 < \mu \leq 3$ and l_j represent the flight length of each consecutive step j in the random walk [31].

$$P(l_j) = l_j^{-\mu} \tag{1.1}$$

There has been a lot of dispute concerning the presence of Lévy flights in predator movement. The 1996 *nature* article, *Lévy flight search patterns of wandering albatrosses* [32], was the first study testing the Lévy foraging hypotheses on data from animals in their natural environment [33]. The optimality of the Lévy flight for foraging when the target sites are sparse was later shown [31]. There has been pointed out flaws in the data used in the 1996 article, these critiques are summarised briefly in the *Science* article *Do Wandering Albatrosses Care About Math?*[34]. However, later studies suggest that predators may after all follow Lévy flight pattern. There has for instance been shown strong evidence for Lévy motion in species like sharks, tuna, billfish and ocean sunfish [35]. A recent review [33] summarises the history of the dispute about the Lévy foraging hypothesis, and concludes: "the key question now is not whether some organisms have Lévy walk movement patterns, but when and why they do".

1.3 Motivation

Technological advances in acoustic telemetry systems as well as in the fields on unmanned and autonomous vessels are enabling new techniques in the studies of fish movement ecology. The use of unmanned vessels will yield new and interesting data for these studies. In a long-term perspective, it could be possible to use unmanned vessels to search, detect and track acoustic tags. The main goal of this thesis is to devise search plans for an unmanned surface vessel (USV), carrying an acoustic receiver, in order to efficiently detect acoustic transmitters.

1.4 Problem description

The full problem statement given by supervisor Jo Arve Alfredsen is given in appendix A, and summarised below:

The core task of this assignment is to develop methods for devising USV search plans in a given search area and with given limitations in sensor sensitivity, vehicle performance, environmental conditions, and knowledge of target distributions. The project consists of the following tasks:

- Literature survey and study of the theory of optimal search, including bio inspired strategies such as Lévy flight/walk.
- Review of the application of autonomous vehicles for search purposes, and particularly cases similar to the target problem.
- Based on the acquired theoretical background, develop an explicit formulation of problem of calculating efficient search plans for the USV with respect to the:
 - Sensor characteristics (receiver/transmitter, signal propagation)
 - Vehicle performance and limitations
 - Target behaviour and distribution probability
 - Characteristics of the search area and environmental conditions
 - Single or multiple cooperating vehicles
- Develop a simulation model of the system and candidate planning algorithms, and evaluate the performance of different search strategies.
- Develop a search plan generator that takes in relevant inputs and provides output that can be interpreted by the unified navigation environment (Dune/Neptus).
- Plan and perform a field test using USV Otter and discuss practical aspects and use of the search algorithm.

1.5 Outline

This thesis is structured as follows.

Chapter 2 presents underlying concepts needed for the development of the search paths. This includes a short description of acoustic fish telemetry and detection range for acoustic telemetry systems.

Chapter 3 further defines the search problem in terms of the search space, searcher and target characteristics, a sensor model and objective.

Chapter 4 presents a discrete map representation used for the search as well as 3 different ways to generate search paths.

Chapter 5 describes simulations performed to evaluate the search paths from the previous chapter as well as detection results from the simulations.

Chapter 6 describes a search path generator in MatLab that is capable of communication with Neptus through the IMC message format.

Finally, the thesis is concluded in chapter 7, which discusses the results of the work and recommends topics for future research.

Theory

2.1 Acoustic fish telemetry

This section aims to give a short introduction to acoustic telemetry systems, and a few considerations on the detection range of the tag-receiver system. Note that effects such as false positives may also occur, but are not treated in this section as they will not have direct impact on the search path generation developed in this thesis. More about acoustic telemetry can be found in the source material in the article *Acoustic telemetry overview* [7], and the technical report *Fish telemetry manual* [1]. More thorough descriptions of underwater sound are given in the book *Principles of Underwater Sound* [36].

2.1.1 Acoustic telemetry systems

An acoustic telemetry system consists of transmitters (tags) and receivers (hydrophones). In fish telemetry the tag is attached to, or implanted in fish. The receiver can be placed on stationary platforms such as a moored buoy, or a mobile platform such as a boat or underwater vessel or even large marine animals.

Acoustic tags consist of an acoustic transducer, a battery and electronics [7]. The transmission range of the acoustic tag is a function of its size. This is because of the relationship between the size of acoustic transducers and their most efficient frequency, and the relationship between transmission frequency and range. In general, smaller resonant elements will be more efficient at higher frequencies and higher frequencies have shorter transmission range. This means that studies on smaller species also involve tags with shorter transmission ranges [7].

There are two main types of receivers. Hydrophones can either be directional, or omnidirectional. A typical use case for directional hydrophones is tracking studies from research vessels, where the researchers steer towards the signal. Omnidirectional receivers are often attached to moored buoys.

2.1.2 Detection range

The detection range of the sensor in use is crucial information when designing a search algorithm, as it will factor into the spacing of adjacent strafes in the search path. Underwater sound is complicated and propagation depends on a number of factors. The general recommendation when using systems depending on underwater acoustics is to perform a range test to determine the range of a system in the relevant conditions. However, it is possible to determine approximate values of range performance by calculations based on the passive sonar equation. The passive sonar equation [36]¹ is given in eq. (2.1)

$$DT = SL - 20\log_{10}(R) - \alpha R - NL \quad (2.1)$$

,where:

R	=	Range [m]
DT	=	Required signal to noise ratio for detection [dB] at range R
SL	=	Tag signal level [dB re 1 μ Pa @ 1m]
α	=	Absorption [dB/m]
NL	=	Noise level [dB re 1 μ Pa]

The units of the variables in the equation are shown in square brackets. The range R , is the size of interest. Signal to noise ratio, DT , is a property of the receiver unit, and signal to noise ratio above a certain threshold is required for detection. The unit of the tag signal level, SL , is an expression of the acoustic pressure at a 1 meter distance from the tag. The absorption, α , is dependent on the properties of the surrounding body of water like temperature, salinity and conductivity, as well as frequency of transmission [7]. α is noticeably smaller for fresh water than salt water. Absorption α at a temperature of 5°, by frequency is shown in fig. 2.1. The noise level NL depends on thermal and sonic noise produced by weather in addition to noise from nearby vessels and machinery[1]. Figure 2.2, provides nominal values for noise in the sea as a function of frequency for different wind/sea conditions.

¹The presented form of the passive sonar equation is found in [7], and is a composition of multiple equations in [36]

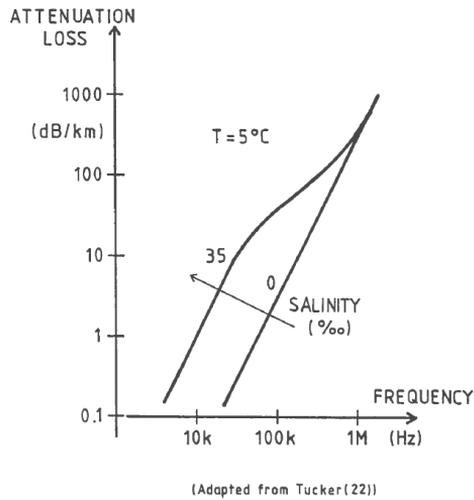


Figure 2.1: Absorption by frequency, from [1] (adapted from [2])

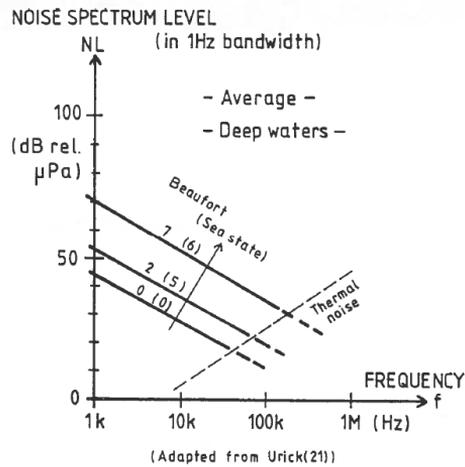


Figure 2.2: Noise in the sea, from [1](adapted from [3])

The noise level NL , for calm weather (Beaufort number and sea state 0) from fig. 2.2 is approximately

$$NL = 20dB$$

, and the bandwidth dependant noise

$$10\log_{10}B = 30dB$$

This gives a total noise contribution of $50dB$. Inserting for the values in eq. (2.2) then gives the transmission loss

$$TL = 85dB$$

Since the attenuation increases with frequency, and $f \approx 70kHz$, using the curves for $f = 100kHz$ in fig. 2.3 will give a conservative estimate. For spherical spreading, the ranges for salt water and fresh water can be estimated at respectively

$$R_{salt} \approx 500m \quad \text{and} \quad R_{fresh} \approx 5000m.$$

Alternatively, this can be calculated from eq. (2.1). α can be found from fig. 2.1 for the given frequency for salt and fresh water as $\alpha_{salt} \approx 50dB/km$ and $\alpha_{fresh} \approx 1dB/km$. Solving eq. (2.1) iteratively then gives

$$R_{salt} \approx 590m \quad \text{and} \quad R_{fresh} \approx 7500m.$$

These examples illustrate two points. First, the need for range tests as the parameters in eq. (2.1) are difficult to estimate accurately. Second, there is a notable difference between ideal ranges in salt and fresh water for otherwise equal conditions.

2.2 Digital maps

There are two main methods of representing digital maps. One of these methods is using a discrete raster where the pixels in the raster contains information about the containing area. This representation needs the most storage space. The second and more common method represents the maps as a collection of polygonal shapes. A common way of representing polygons is by a collection of vertices, where the vertices of the outer polygon edges are listed in clockwise direction, while vertices representing holes are listed in counterclockwise direction.

Digital maps are often distributed as *shapefiles*, which supports storage of point, line and polygon features. The shapefile format consists of three files to represent a map: a main file (.shp), an index file (.shx) and a table file (.dbf). The format is developed and distributed by ESRI (Environmental Systems Research Institute). More information about the format can be found in the technical description [39]. Programming languages like Python and Matlab have packages that offer support for reading and writing shapefiles.²

²MatLab mapping toolbox: <https://se.mathworks.com/products/mapping.html>
Python PyShp: <https://pypi.org/project/pyshp/>

An electronic system for geographical or map data is often referred to as GIS (Geographic Information System). QGIS is an open source GIS program that has been useful in the work with this thesis.

2.3 Coordinate systems and reference frames

Positions and motions are always described with respect to a coordinate frame. As this thesis involves navigation, some clarification of terms related to coordinate frames and reference systems is given in this section.

2.3.1 Geodetic datums

A geodetic datum is a reference system for measurements. Geodetic datums are typically defined by an ellipsoid used to approximate the shape of the Earth. A position can then be given by a longitude and latitude on the ellipsoid and a height above it. As different ellipsoids fit the shape of the Earth with different accuracy in given areas, there are a number of local geodetic datums, as well as different global approximations. The difference in positions defined in local and global datums can be significant. Examples of global datums are ITRF (International Terrestrial Reference Frame) and WGS84 (World Geodetic System), which is used for GPS. The datum used for the main map series of Norway, the N50-series, is EUREF89, which is a regional datum for Eurasia [40].

2.3.2 Map projections

Map projections are used to project the spherical shape of the earth onto a flat area. UTM(Universal Transverse Mercator) is a Gauss-Krüger-projection with a zone width of 6° [41]. Given a UTM zone, a position is described by Easting and Northing within the zone, given in meters. The coordinates within an UTM zone are cartesian, which simplifies calculations of distance and area in comparison to using latitude/longitude-based coordinates where spherical effects have to be considered. This property makes UTM suitable for local navigation and mapping purposes.

2.3.3 Earth centered reference frames

ECI - Earth Centered Inertial

Earth Centered Inertial(ECI) frame, $\{i\} = \{x_i, y_i, z_i\}$ has origin in the centre of the earth, x-axis in the equatorial plane pointing towards vernal equinox, z-axis along the earth rotational axis and the y-axis is completing the right-hand frame [42]. The frame is a non accelerating frame where Newtons laws of motion apply [43], often referred to as an inertial frame

ECEF - Earth Centered Earth Fixed

Earth Centered Earth Fixed(ECEF) frame, $\{e\} = \{x_e, y_e, z_e\}$ is similar to ECI frame, except that it is earth fixed and is thus not inertial. ECEF is used for global navigation [43].

2.3.4 Tangent plane reference frames

Tangent planes can be used for local navigation, and is usually referred to as flat Earth navigation. While these coordinate systems are not inertial, it is common to assume they are inertial for local navigational purposes [43]. The two common conventions for tangent plane coordinate systems are East-North-Up (ENU) and North-East-Down (NED), $\{n\} = \{x_n, y_n, z_n\} = [N, E, D]^T$. An illustration of ENU with respect to ECEF is shown in fig. 2.4.

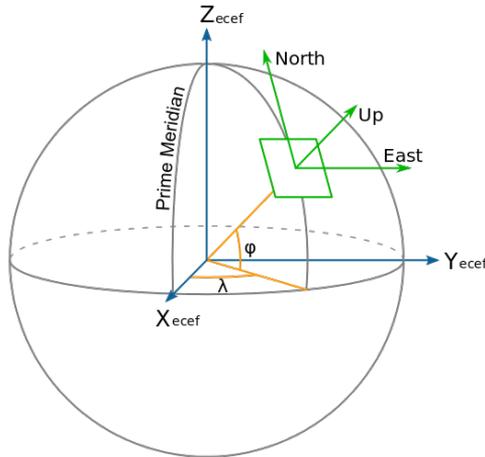


Figure 2.4: ECEF and ENU coordinate systems

2.3.5 Body-frame

The body-frame, $\{b\} = \{x_b, y_b, z_b\}$ is fixed to a point in the vessel of interest (marine-craft, plane, robot, etc.). The position and orientation of the vessel can then be described by the position and orientation of the body-frame with respect to the inertial coordinate frame. Body-frames are typically useful for describing linear and angular velocities of a vessel.

2.4 Guidance, navigation and control

Marine control systems are often divided into three parts: guidance, navigation and control. The following definitions are provided by [43].

Guidance: the guidance system is the part of the system that computes the reference position, velocity and acceleration of the marine craft. The guidance system uses the marine craft states and optional external inputs such as weather data to compute references that follow an external input, such as for instance waypoints provided by an operator.

Navigation: the navigation system determines the position, attitude, course and distance traveled of the marine craft.

Control: the action of determining necessary control forces and moments to follow a reference.

2.4.1 Vessel model

Surface vessels are commonly represented by a 3DOF model by assuming the roll and pitch angles, ϕ and θ , are small. This gives the following 3DOF kinematics [43]:

$$\dot{\boldsymbol{\eta}} = \mathbf{R}_z(\psi)\boldsymbol{\nu} \quad (2.3)$$

,where $\mathbf{R}_z(\psi)$ is the rotation matrix about the z-axis:

$$\mathbf{R}_z = \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (2.4)$$

, $\boldsymbol{\eta} = [N, E, \psi]^T$ is and $\boldsymbol{\nu} = [u, v, r]^T$ is . The dynamics of the vessel can be represented by the rigid-body kinetics as [43]:

$$\mathbf{M}_{RB}\dot{\boldsymbol{\nu}} + \mathbf{C}_{RB}(\boldsymbol{\nu}) = \boldsymbol{\tau}_{RB} \quad (2.5)$$

where \mathbf{M}_{RB} is the rigid-body inertia matrix, \mathbf{C}_{RB} is Coriolis and centripetal forces and $\boldsymbol{\tau}_{RB} = [X, Y, T]^T$ is forces in surge(x_b) and sway(y_b) direction and momentum in yaw(ψ).

2.4.2 Line-Of-Sight Guidance

Line-of-Sight guidance is a method used to follow straight line paths described in [43]. LOS guidance for navigation between waypoints in the 2D(2 dimensional)-plane $[x, y]$ is implemented by assigning values to the steering angle $\chi(t)$:

$$\chi(t) := \text{atan2}(\dot{y}(t), \dot{x}(t)) \quad (2.6)$$

While maintaining a positive speed $U > 0$:

$$U(t) := \|\mathbf{v}(t)\| = \sqrt{\dot{x}(t)^2 + \dot{y}(t)^2} \quad (2.7)$$

$\text{atan2}(y, x) \in [-\pi, \pi]$ is a two argument, four-quadrant version of the inverse tangent function $\text{atan}(y/x) \in [-\frac{\pi}{2}, \frac{\pi}{2}]$. LOS-guidance considers the path defined by the way-points $\mathbf{p}_k = [x_k, y_k]^T$ and $\mathbf{p}_{k+1} = [x_{k+1}, y_{k+1}]^T$. Defining the angle α_k as

$$\alpha_k := \text{atan2}(y_{k+1} - y_k, x_{k+1} - x_k) \quad (2.8)$$

allows computation of the along-track distance (path tangential distance to vessel), $s(t)$, and cross-track error (path normal distance to vessel), $e(t)$, for a vessel at position $p = [x(t), y(t)]^T$:

$$s(t) = [x(t) - x_k] \cos(\alpha_k) + [y(t) - y_k] \sin(\alpha_k) \quad (2.9)$$

$$e(t) = -[x(t) - x_k] \sin(\alpha_k) + [y(t) - y_k] \cos(\alpha_k) \quad (2.10)$$

The cross-track error is relevant for path following since $e(t) = 0$ means the vessel is following the path. This gives the control objective for straight-line following as:

$$\lim_{t \rightarrow \infty} e(t) = 0 \quad (2.11)$$

Two steering laws based on LOS guidance are described in [43]; *enclosure-based steering* and *look-ahead-based steering*. In look-ahead-based steering, the desired course angle, χ_d , is set as:

$$\chi_d(e) = \alpha_k + \chi_r(e) \quad (2.12)$$

where

$$\chi_r(e) := \arctan\left(\frac{-e}{\Delta}\right) = \arctan(-K_p e) \quad (2.13)$$

$\chi_r(e)$ can be interpreted as either a "velocity-path relative angle", ensuring that the velocity is directed towards a point on the path a distance $\Delta > 0$ ahead, or as a proportional control law.

Search problem

This chapter aims to give a more explicit formulation of the problem of calculating a search path. Each section discusses an aspect of the problem, and the objective is given in section 3.5.

3.1 Search space

In search problems, a prior probability distribution on the target location is assumed known. This distribution can be based on previous known target locations, knowledge about terrain or other types of expert domain knowledge.

Prior distributions can be defined in discrete or continuous search spaces. The following definitions of discrete and continuous prior distributions is given by [23].

Discrete prior distribution. Assume a discrete search space consists of N cells. The discrete prior distribution then describes the probability $p(n)$ of the target being in cell n for each cell $n \in [1, 2, \dots, N]$. It is assumed that:

$$\sum_{n=1}^N p(n) = 1 \quad (3.1)$$

Continuous prior distribution. A continuous prior distribution is given by a probability density function. A uniform distribution can be used to describe the case when there is no prior information about target position. For a region \mathcal{R} in the plane with area A , the uniform distribution is defined as:

$$p_u(x) = \begin{cases} \frac{1}{A} & x \in \mathcal{R} \\ 0 & \text{otherwise} \end{cases} \quad (3.2)$$

Normal prior distributions can be used when we have information of the targets last known location with an estimate of the location uncertainty. The bivariate normal probability density function in the x_1, x_2 -plane is written as [44]:

$$p_G(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}} e^{-\frac{1}{2(1-\rho^2)}\left(\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2} - \frac{2\rho(x_1-\mu_1)(x_2-\mu_2)}{\sigma_1\sigma_2}\right)} \quad (3.3)$$

The parameters μ_1, μ_2 denote the mean of the distribution on the x_1, x_2 -axis, whilst σ_1, σ_2 denote the standard deviations. ρ is the correlation between x_1 and x_2 . In the case where x_1 and x_2 are uncorrelated, the expression simplifies to:

$$p_G(x_1, x_2) = \frac{1}{2\pi\sigma_1\sigma_2} e^{-\frac{1}{2(1-\rho^2)}\left(\frac{(x_1-\mu_1)^2}{\sigma_1^2} + \frac{(x_2-\mu_2)^2}{\sigma_2^2}\right)} \quad (3.4)$$

A discrete prior distribution can in general describe both discrete and continuous real-life spaces. Consider a search area in the plane, the discrete prior distribution can then be defined by imposing a grid of cells on the search area. Algorithms exist for search problems in both discrete and continuous priors. The description of the prior distribution and the search space is a design choice for each specific application. In the search-and-rescue path planning articles [25] and [26], mentioned in chapter 1, the search area is represented as a grid of cells and a discrete prior distribution. In general, a discrete representation of the search space will occupy more storage space than a continuous one, as the discrete space will typically be represented as a matrix of values, whilst a continuous space can be represented by a probability distribution function.

Continuous and discrete formulations are both somewhat capable of incorporating vessel dynamics in the path planning. In a discrete path constrained search, the consecutive steps in a path are restricted to neighbouring cells. Discrete path constrained search formulations are especially relevant when transit between cells take significant time, or when the ratio of searcher to target speed is low [23]. Optimal paths for discrete searcher path problems can be found using branch and bound type algorithms.

In continuous constrained search, the dynamical model of the searcher with states $y(t)$ is directly considered in the calculation of the path. This means that instead of calculating a path directly, optimal control inputs can be calculated. The optimisation problem, where $E(q(T, w))$ denotes the probability of failing to detect the target, U and Y , are closed convex constraint sets and $h(y(t), u(t))$ is the dynamical model of the searcher, becomes [23]:

$$\text{find control } u(t) \in U \text{ and initial condition } y_0 \in Y \quad (3.5)$$

$$\text{that minimise } E(q(T, w)) \quad (3.6)$$

$$\text{subject to } \dot{y}(t) = h(y(t), u(t)), t \in [0, T], y(0) = y_0 \quad (3.7)$$

This formulation is given for one target and one searcher, but is extended upon in [23] to involve multiple targets and multiple searchers.

While the representation of a continuous search space is more efficient, and allows for a higher level of integration of searcher dynamics, a discrete search space formulation will be used in this thesis. This is due to the simplicity of the formulation, as well as generally lower path computation times. A natural consequence of the discrete space is that the search path will consist way-points for the searcher vessel to follow. This means that the search path is independent of the guidance, navigation and control tasks described in section 2.4, and rather acts as an input to a guidance system.

3.2 Searcher characteristics

The main restriction of the searcher vessel is the total search effort. The USV Otter proposed as a searcher vessel is an electrical vehicle, limiting effort by battery consumption and capacity. For a speed of $U = 2kts \approx 1m/s$, the Otter USV has capacity to run for $T = 20h$ [19].

The Otter USV is claimed to have high manoeuvrability. The neighbouring cells of a current cell can be therefore be defined as any of the 8 adjacent cells in the discrete map. A more realistic behaviour could be obtained by associating a cost to changing the velocity direction of the searcher vessel. The cost for each cell would be an increasing function according to the required change of velocity direction to reach the cell.

3.3 Target characteristics

For the scenario of search for acoustically tagged fish, it is natural to assume multiple targets. The targets can be modelled as stationary or moving in the calculation of a search path.

In the discrete searcher path problem, moving targets can be simulated using Markov Chains to at each time step update the discrete probability distribution. This implies that the search path will depend on the original distribution and how it develops over time, whereas the search path for a stationary target only would depend on the original distribution.

As mentioned in chapter 1, a long term goal of an USV search for acoustic tags is detection and tracking of fish and thereby obtaining information about fish movement patterns. The target movement will not be simulated in the generation of the search paths in this thesis, but the search paths will be evaluated for instances of both stationary and moving targets in chapter 5.

3.4 Sensor model

An ideal approximation of the acoustic tag-receiver system is given by a sphere about the receiver within which every signal transmitted is received. Upper estimates of the radius

of this sphere can be found as demonstrated in section 2.1.2. By applying a conservative estimate for the radius R , it is possible to consider the tag-receiver system as a definite range sensor in 2D-space as in eq. (3.8):

$$P(\text{detection}|\text{transmission}) = \begin{cases} 1 & d \leq R \\ 0 & d > R \end{cases} \quad (3.8)$$

, where d is the Euclidean distance between tag and receiver:

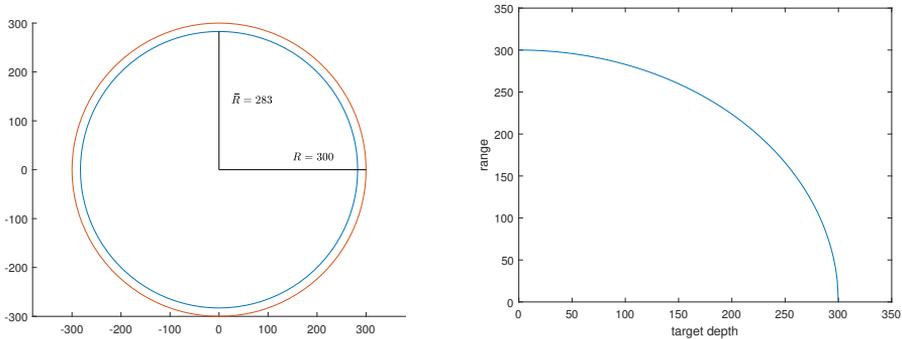
$$d = \sqrt{(x_{t,1} - x_{r,1})^2 + (x_{t,2} - x_{r,2})^2} \quad (3.9)$$

A detrimental effect to the effective range, R , in the 2D definite range formulation is the geometry of the problem. The targets, tagged fish, move in 3-dimensions. The lateral range of the sensor is at its largest at the depth of the receiver, and decreases in both directions. The swimming depth of the target therefore has to be accounted for.

Consider a target swimming at a depth d_t below the surface and let the origin of a coordinate system x, y, z be placed in the receiver. The intersection of the plane $z = -d_t$ and the sphere $x^2 + y^2 + z^2 = R^2$ is a circle representing the 2D lateral range of the sensor at depth d_t . This circle is given by:

$$x^2 + y^2 = \bar{R} = \sqrt{R^2 - d_t^2} \quad (3.10)$$

$R = 300m$ and \bar{R} for a target at depth $d_t = 100m$ is shown in fig. 3.1a. The reduction of lateral range at $d_t = 100m$ is only $R - \bar{R} = 17.2m$. The lateral range as a function of target swimming depth is shown in fig. 3.1b. Because of the circular slope, the reduction is small for a large range of target swimming depths. This implies that this effect does not have to be accounted for explicitly given a conservative range estimate. Exceptions are if the targets are deep water fish or the estimated range is small.



(a) $R = 300m$ and \bar{R} for $d_t = 100m$

(b) Lateral range as function swimming depth, $R = 300m$

Figure 3.1: Lateral range reduction due to target swimming depth

As mentioned in section 2.1.2, it is common to recommend range tests to find the range of a tag-receiver system for a given use-case. Data from a range test performed in Gaulosen, a part of the Trondheim-fjord, was supplied by supervisor Jo Arve Alfredsen. Illustrations of the signal receptions at different ranges is shown in fig. 3.2. The data set consists of $n = 116$ signal transmissions. As all signals emitted from a distance of $R = 230m$ or less are received it seems reasonable to assume that a definite range model with $R \approx 230m$ can describe the receiver-tag system. This strengthens the assumption of using a definite range sensor model.

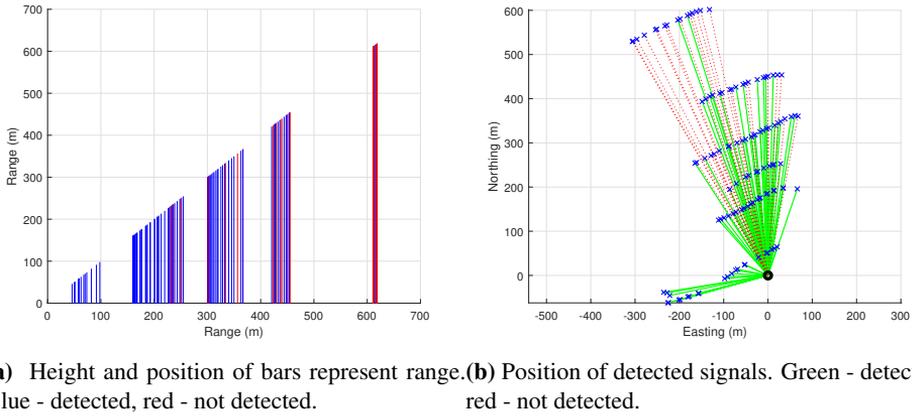


Figure 3.2: Range test in Gaulosen

As an alternative to the definitive range model, the detection probability could be expressed as a decreasing function of distance between receiver and tag as in eq. (3.11). A histogram showing ratio of detected signals by distance width bin size of $100m$ is shown in fig. 3.3. The number of samples per bin is indicated. Data like this could possibly be utilised to fit a sensor model.

$$P(\text{detection}|\text{transmission}) = f(d), \quad d = \sqrt{(x_{t,1} - x_{r,1})^2 + (x_{t,2} - x_{r,2})^2} \quad (3.11)$$

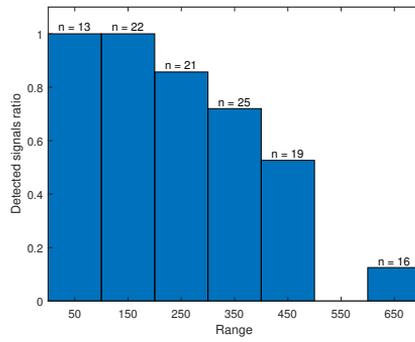


Figure 3.3: Histogram of detected signal ratio by range

3.5 Objective

In the previous sections in this chapter, it has been established that the search path problem is modelled as a discrete searcher path problem, searching for multiple targets, with a definite range sensor model and an upper limit on search effort given by the battery time of the USV(s). Assuming a grid of size $n \times n$ with a cell size corresponding to sensor range, and an upper time limit of T , a path can be described by a sequence of cells: $p = (\{i, j\}_1, \{i, j\}_2, \dots, \{i, j\}_T)$. Each cell has a probability of containing target k , $P_{i,j}(k)$, which sums to 1 over all the cells for each target:

$$\sum_{i=0}^n \sum_{j=0}^n P_{i,j}(k) = 1 \quad (3.12)$$

Using a binary variable $x_{i,j}$ to represent whether a cell $\{i, j\}$ is part of a path p , such as in [26],

$$x_{i,j} = \begin{cases} 1 & \{i, j\} \in \mathcal{P} \\ 0 & \text{otherwise} \end{cases} \quad (3.13)$$

the cumulative probability of detection over a path p can then be described as:

$$PC(p) = \sum_{i=0}^n \sum_{j=0}^n \sum_{k=0}^n x_{i,j} P_{i,j}(k) \quad (3.14)$$

The optimal search path problem then becomes that of finding a path p^* with cumulative probability, $PC(p^*)$ over the path, such that:

$$PC(p^*) \geq PC(p) \quad \forall \quad p \in \mathcal{P} \quad (3.15)$$

, where \mathcal{P} is the set of all possible paths that hold the time limit restriction.

In addition to this, the path should be designed such that collisions with known mapped obstacles are avoided. It is desirable to add constraints to the start and end point of the searcher, for easy deployment and recovery during research missions. This will decrease the set \mathcal{P} to only include paths with start and end point at predetermined cell locations.

To expand this formulation for multiple searchers, the path p^* of interest can be replaced by a set of paths $\mathcal{P}^* \in \mathcal{P}$.

A natural expansion of this for the future would be to consider additional constraints than time limit on the USV performance, such as adding costs to difficult to manoeuvre path segments and movement.

Path generation

The goal of this chapter is to generate a search path represented by way-points and a desired forward speed for movement between the way-points. The path should be made in such a way that it gives good performance with respect to the objective presented in chapter 3 - it should be efficient. However, the path planning methods presented do not guarantee optimality, nor do they supply optimality bounds. This chapter begins by explaining the map representations used to generate search paths in section 4.1. Further, three methods of search path generation are presented in sections 4.2 to 4.4. Section 4.5 explains the conversion from search paths in discrete space to way-points and section 4.6 discusses the forward speed of the searcher. Finally, a method of dividing the search space for multiple searchers is presented in section 4.7. Throughout the chapter, the map of a small lake called Ulvenvatnet, shown in fig. 4.1, will be used to illustrate the concepts.



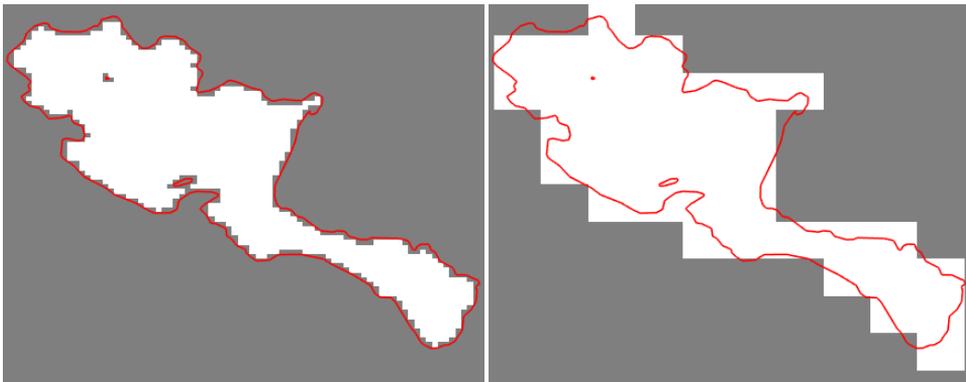
Figure 4.1: Polygonal map of Ulvenvatnet

4.1 Map representation

Since the sensor of the searcher in this case has a much larger range than the size of the searcher, two discrete grid maps are created to define the search space. One map has resolution chosen with respect to the sensor range, whilst the other has resolution chosen according to the size of the searcher.

The map with resolution chosen according to searcher size is an occupancy grid. In an occupancy grid, each cell has a value according to the probability of the cell being occupied. For the task of representing a polygonal map, this means that the cells that contains obstacles are marked 1, and the open cells are 0.

The map with resolution chosen with respect to the sensor range, will be referred to as the search map, and is an up-sampled version of the occupancy grid. The 'occupancy' of the cells in this map is determined by a threshold on the number of occupied cells in the underlying occupation grid. For instance, a cell in the search map can be open (0), if more than half of the containing cells in the high resolution grid are open (0). An open cell in the search map is viable to be a part of the search paths, which will all be calculated in the search map. The resulting path is then transformed using the occupancy grid to return way-points that avoid known obstacles in the environment, as explained in section 4.5. An occupancy grid and a search map generated in Ulvenvatnet is shown in fig. 4.2.



(a) Occupancy grid

(b) Search map

Figure 4.2: Occupancy grid and search map. White cells are open (0).

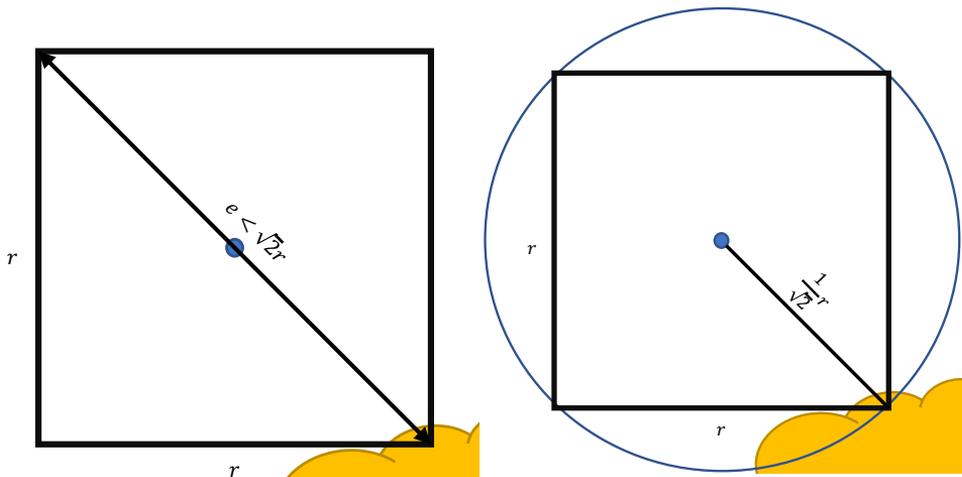
4.1.1 Occupancy grid sampling

To create an occupancy grid, the resolution must first be chosen. This is a consideration mainly between storage and accuracy, as a large resolution leads to large maps. This section explains how occupancy grids have been sampled from polygonal maps in this thesis, and aims to give some insights in the choice of occupancy grid resolution. The first thing to consider is that each cell should be large enough to room the searcher. Given a searcher vessel with dimensions $w \times b$, where $w \geq b$, each cell should at least be of size $w \times w$. This will ensure that the path from the centre of one open cell to the centre of another

open cell does not lead to a collision. Secondly, the resolution of the map should be high enough to avoid large losses of information from the original polygonal representation of the map, like for instance removing passages that are safe to travel. It is also convenient for up-sampling and down-sampling purposes that the occupancy grid is of size $2^n \times 2^n$ (square and a power of 2) cells, so this should also be considered when choosing the resolution.

In an occupancy grid all cells that are partially covered by an obstacle are marked as occupied. Consider an occupancy grid with cells of size $r \times r$. The longest possible distance from an occupied area of the cell to the furthest cell edge is the diagonal of the square cell: $e < \sqrt{2}r$, see fig. 4.3a.

A simple way to determine the occupancy status of each cell, is to determine whether the centre point of this cell is occupied the polygonal map or not. To ensure that each cell that contains any occupied space is marked as occupied, the occupied areas of the polygonal map have to be expanded by a buffer. Consider a grid with cells of size $r \times r$, then the furthest distance from the centre point of a cell to a point in the occupied space within the cell is $\frac{1}{\sqrt{2}}r$, see fig. 4.3b. Expanding the occupied areas of the polygonal map by $\frac{1}{\sqrt{2}}r$ will then ensure that no partially occupied cells are marked open. This method does however also lead to some unoccupied cells being marked as occupied. The circle in fig. 4.3b illustrates this. This error is also of magnitude $e < \sqrt{2}r$.



(a) Discretization error illustrated by a single cell. (b) Discretization error due to expanding polygon sampling method. Yellow region represents an obstacle.

Figure 4.3: Occupancy grid sampling

Additionally to expanding the polygon occupied areas by $\frac{1}{\sqrt{2}}r$, a safety buffer b can be

added as an additional buffer to the occupied areas of the polygonal map. For instance, b can be a consideration of the depth profile of the a body of water and the draught of the searcher vessel and the accuracy of the original map.

In summary, the occupancy grid map is created by first expanding the occupied space in the polygonal map by a buffer of $a + \frac{1}{\sqrt{2}}r + b$, where a is the inaccuracy of the original polygonal map, $r \times r$ is the size of the cells in the occupancy grid and b is an additional safety margin decided by the user. The occupancy grid is then calculated by sampling the centre of each $r \times r$ cell and checking whether the point is in the occupied space of the expanded polygonal map.

4.1.2 Search map sampling

The resolution of the search map is chosen such that a visitation of the sensor to the middle of a cell ensures that the sensor will cover the entire cell. The omnidirectional hydrophone sensor with range R will have a circular reach in 2D-space. The search map cell size can then be decided by the size of the largest square that fits inside the circle defined by the sensor range, see fig. 4.4. This implies that the cell size should be at most $\sqrt{2}R \times \sqrt{2}R$. In addition to this, as it is convenient to create the search map from up sampling of the occupancy grid, it should be size $2^m \times 2^m$. This means that the size of the grid cells will be shrunk to accommodate this.

As mentioned earlier, the open/occupied state of a search map cell is determined by a threshold, t , on the number of underlying open/occupied cells in the occupancy grid. This threshold will determine if a cell should be searched or not. A high threshold will mean that large areas will not be searched. A low threshold could possibly lead to complicated paths that could favour areas with a small searchable space to areas with more searchable space, resulting in less area covered given limited time.

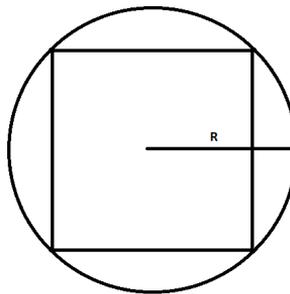


Figure 4.4: Square in circle with radius R

4.2 Complete coverage path

No prior information about the target(s) is equivalent to assuming a uniform prior distribution over the search space. When this is the case, it can be useful to move in a path that covers the search space - a complete coverage path. The problem of finding a complete coverage path is well known from robotics. A simple way of generating an approximate path of complete coverage is by dividing the space into a robot/sensor-sized occupation grid, computing the *distance transform* from a goal cell in the occupation grid, and moving in the direction of steepest ascent according to the distance transform [45]. This is referred to as wavefront coverage planning [29], and can be interpreted as a discrete potential field approach. Distance transform/wavefront navigation can also be used to find short paths from any position in a discrete map to a goal by moving in the direction of steepest descent. Distance transforms can be computed using different distance metrics and different types of connectivity based on application. City-block-distance (4-connectivity), Euclidean-distance (8-connectivity) and chessboard-distance (8-connectivity) are shown in fig. 4.5. In general, 8-connectivity will give shorter paths than 4-connectivity as 4-connectivity restricts the path to turns that are a multiple of 90° . The distance transform with euclidean distance measure computed in the search map from the previous section is shown in fig. 4.6. The process of assigning a distance to each cell in the map amounts to a breadth first search, which has worst case complexity $O(|V|+|E|)$, where $|V|$ is the number of vertices (cells) and $|E|$ is the number of edges between cells (depends on connectivity). This is the most computationally heavy part of the process, as the path-finding reduces to a simple look-up for each consecutive step in the path from any starting point in the map.

A clear advantage of this path planning method is the inherent ability to choose starting and end points freely. Since the distance transform from the goal has been computed, and the distance transform is proportional to the true distance, it is also simple to modify the method to allow for restrictions on path lengths. When the maximum path length is equal to the distance of the path this far, and the distance to the goal, start moving in the direction of steepest descent instead of steepest ascent. Paths with and without path length restrictions are shown in fig. 4.7, and the procedure for calculating these paths in a discrete grid is summarised in algorithm 1.

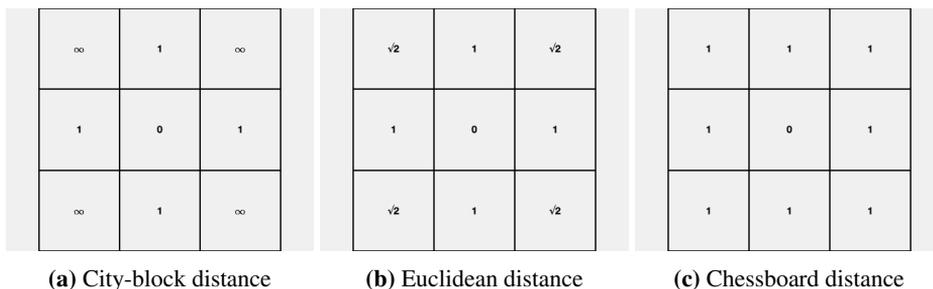


Figure 4.5: Distance transforms computed from centre cell with different metrics

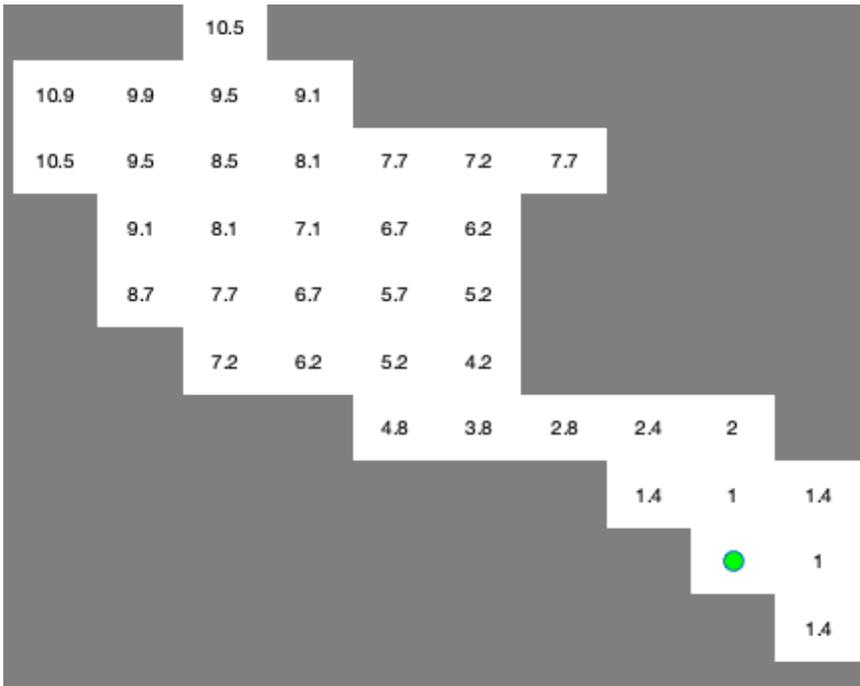


Figure 4.6: Distance transform in search map

Algorithm 1: Coverage path with path length restriction

```

Compute distance transform from goal cell
Set start cell to current cell
Set path length to zero
while path length + distance to goal < maximum path length do
    Find neighbouring cell with highest value;
    Set current as visited and add to path;
    Update path length;
    Set current to neighbouring cell with highest value;
    if current == goal then
        | return path
    end
end
while current != goal do
    Find neighbouring cell with lowest value;
    Set current as visited and add to path;
    Set neighbouring cell to current;
end
Add goal cell to path;
return path;

```

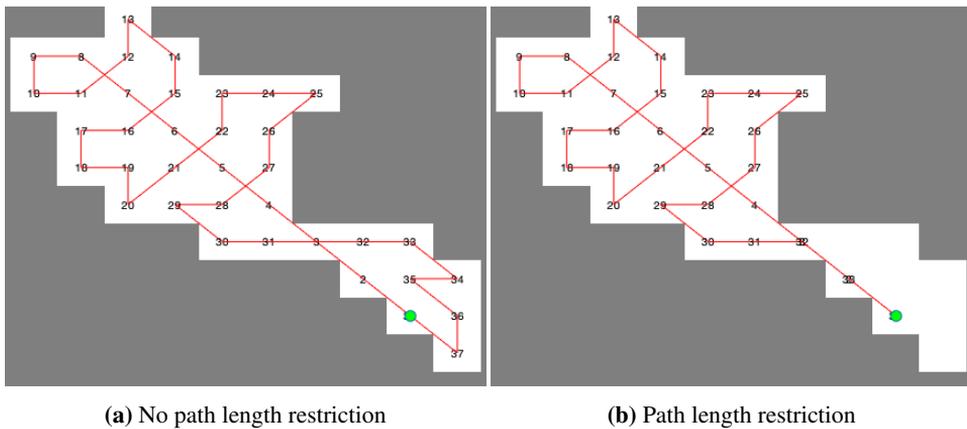


Figure 4.7: Paths calculated with and without path length restriction. Numbers indicate order of visitation. Start and goal point in the green dot.

4.3 Greedy search

The greedy search algorithm assumes there is given a prior distribution on target location, and locally moves in the direction of higher probability values. It is assumed that a start point and goal point is given, and the distance transform from this goal is calculated in order to be able to keep path length restrictions in the same way as in section 4.2. This means that two maps are defined. One map where the value of each cell is an interest value proportional to the probability of a target being in the cell, and the other map where each cell is assigned a distance transform value from the goal cell. The interest value of a cell is set to zero when the cell has been added to the path. An example field and the corresponding greedy path without path length restriction is shown in fig. 4.8

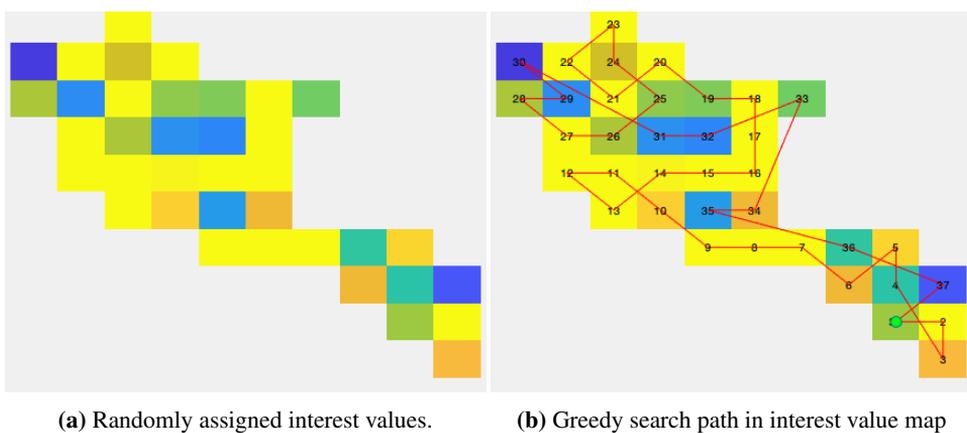


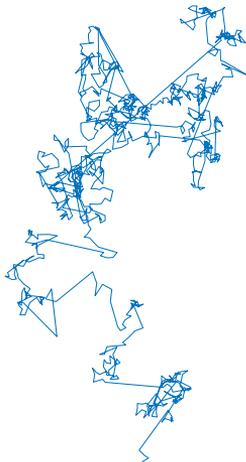
Figure 4.8: Greedy search in interest value map where dark blue represents low values and yellow represent high values

4.4 Lévy flight path

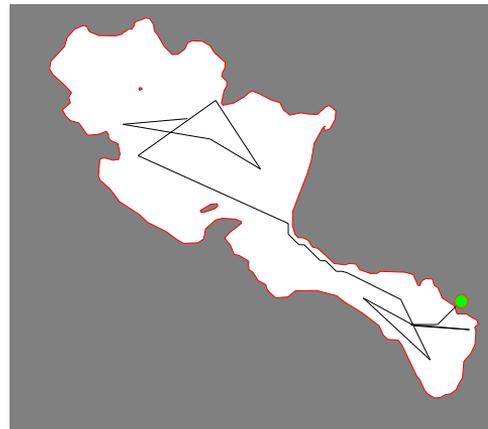
This search path will be termed a Lévy flight path, although it is not truly a Lévy flight path as it will be restricted by the geometry of a map. The Lévy flight path algorithm picks a new way-point in direction α_j with distance l_j from the previous way-point. α_j is picked from a uniform distribution in the interval $[0, 2\pi)$. The distance l_j is picked from the distribution in eq. (1.1), with a minimum value of l_{min} .

$$P(l_j) = l_j^{-\mu} \quad (1.1)$$

The distance l_j and angle α_j is re-elected until the new way-points falls within unoccupied space. When it does, the occupancy grid is used to calculate a path between the previous and new way-point. For this type of search, the starting point can be specified, however, it is more difficult than earlier to accommodate for choosing goal points. A Lévy flight for $\mu = 2$ without area restrictions is shown in fig. 4.9a and a Lévy flight path is shown in fig. 4.9b.



(a) Lévy flight with $\mu = 2$



(b) Lévy search path with $\mu = 2$ in Ulvenvatnet

Figure 4.9: Lévy flight search path

4.5 Way-point generation from search paths

The procedures explained in section 4.2 and section 4.3 output a path that gives a visitation order for cells in the search map. These cells can be a varying degree of unoccupied space depending on the chosen openness threshold t . The way-points are generated by first adding the centre of the start cell, then the centre of the next cell in the path is calculated, and it is checked whether the straight line between them intersects occupied space. If this is the case, the distance transform from the goal to the start is calculated, and a path is generated by moving in direction of steepest descent. If the start or goal is in occupied space, the occupancy grid is used to find a close-by cell that is open, and the centre of this cell is added as start/goal in stead. In addition to this, the way-points are filtered, and the redundant way-points are removed. Redundant way-points are those who do not change the direction of the searcher on the path. This is summarised in pseudo-code in algorithm 2

Algorithm 2: Way-point generation from paths in search map

Add centre of first cell as way-point

for *cell in path* **do**

if *centre point of cell is in occupied space* **then**
 | find open cell close to point in occupancy grid
 | set next point as centre of open cell close

else
 | set next point as centre of cell

end

if *direct line between previous and next way-point passes through occupied space*
 then

 | calculate distance transform and steepest descent path between points in
 | occupancy grid
 | add way-points as centre points in occupancy grid path

end

end

for *way-point in way-points* **do**

if *direction between way-point and next is equal to direction between way-point and
 previous* **then**
 | remove way-point

end

4.6 Speed assignment

The procedures in sections 4.2 to 4.4 are used to calculate way-points for the searcher to visit. Another important aspect of the searcher path is the forward speed moving between way-points. Consider the case where the searcher passes directly over a stationary target with a tag with transmission interval T . To ensure detection, the tag then has to be within range for time T . When the searcher passes directly over the target at speed V_d , the target is within reach for time $T = \frac{2R}{V_d}$, where R is the range of the sensor. The speed that guarantees detection when moving directly above a stationary target is given by eq. (4.1).

$$V_d \leq \frac{2R}{T} \quad (4.1)$$

This will be used as an upper limit for the speed assignment for the search plans. Further considerations could be made on efficient speed assignment given battery consumption characteristics of the searcher vessel. The speed assignment would then also take into account which speed is the most efficient in order to maximise search effort.

4.7 Multiple searchers

A way to accommodate for using multiple searchers is to divide the search map in one part for each searcher. These parts should be of approximately even size such that each searcher can cover the map assigned efficiently. A way of doing this is by K-means clustering in the search map, where K corresponds to the number of searchers. By letting each cell in the search map have a feature vector consisting of their index, K-means clustering can be applied using any of the distance metrics mentioned in section 4.2. This will minimise the geometric spread with respect to the distance metric within each cluster. K-means clustering works by first assigning a random centre point for each cluster. Then the data points are assigned to the closest cluster with respect to the chosen distance metric. The centre point for each cluster is then recalculated. When the centre points for each cluster no longer moves between iterations, the algorithm is complete.

After the search map has been divided, the search path generation procedures in sections 4.2 to 4.4 can be applied as before for each searcher in their respective maps. A map divided by K-means clustering is shown in fig. 4.10.

Simulation

A simulation environment was created in MatLab to evaluate the search paths from the previous chapter. The main code for the simulations can be found in the attached zip-archive, and a demo capable of all search paths and target types can be found in `demo.m`. The simulation case considers a study of migrating salmon smolts in a lake called Hornindalsvatnet, which is the deepest lake in Europe with a maximum depth of 514m. A map of Hornindalsvatnet from the N50 map series from Kartverket [46] is shown in fig. 5.1. This is also the map used for the path generation in the simulations. The N50 and N250 map series are available for students and employees at NTNU for use in teaching and research. The map series are given in UTM projection in zone 32, in datum WGS84, and distributed as ESRI shapefiles.

Salmon migrating in Hornindalsvatnet move between the points marked 'x' on the map, between Horndøla in the northeast and Eidselva river in the southwest.

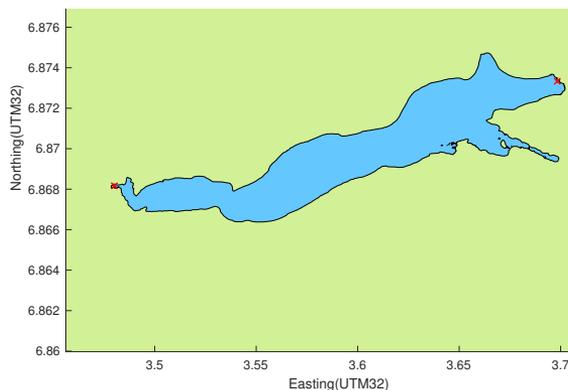


Figure 5.1: Map of Hornindalsvatnet

Biologists believe that salmon tend to stay close to shores during migration. In a fjord migration study of atlantic salmon smolts [47] in the Alta fjord, 54% of detections were within 200m of the shore. To mimic this behaviour of salmon of following shores to the exit point of the lake, a potential field movement is combined with random movement. Each salmon, i , is assigned a forward speed u_i and a heading θ_i . This gives the following dynamical equations for position $[x_{1,i}, x_{2,i}] = [E, N]$:

$$\dot{x}_{1,i} = u_i \cdot \cos(\theta_i) \tag{5.1}$$

$$\dot{x}_{2,i} = u_i \cdot \sin(\theta_i) \tag{5.2}$$

The forward speed of the salmon were set to a constant value of $u = 1.5m/s$. As in the greedy search in section 4.3, the simulated salmon will seek towards areas of high potential. This is done by letting the heading, θ_i , of each salmon be chosen as the direction of steepest ascent in the field. This direction is picked with probability $a = 0.4$ for each time the direction is changed, and otherwise a random direction from a uniform distribution in the interval $[0, 2\pi)$ is chosen. The salmon changes directions at time intervals of $T = 120s$. The discrete potential field used to generate the movement, together with a resulting movement of two salmon starting in the upper right of the lake (mouth of Horndøla) and travelling for $\approx 15h$ is shown in fig. 5.2.



(a) Yellow=high value, blue=low value



(b) Simulated migrating salmon

Figure 5.2: Discrete potential field for fish movement and resulting migration path for two fish

5.1 Target position data

To compare the search paths, 4 different data sets of 100 instances each were generated. Each of the instances contains position data for 20 targets/fish. The acoustic tags send signals at an interval of $T = 60s$.

Data set 1 - migrating targets

The position for each of the targets were initialised according to the values in the interest value map in fig. 5.3, with a high uniform probability of initialisation within 200m of the shores, and a lower probability that decreases linearly with the distance to the shores otherwise. This was done as an alternative to setting the initial position of the target at the mouth of Eidselva river as in fig. 5.2 to avoid the target positions being similar for all the simulated instances. The movement of the targets were simulated with same procedure and parameters as explained above. An example is shown in fig. 5.4a

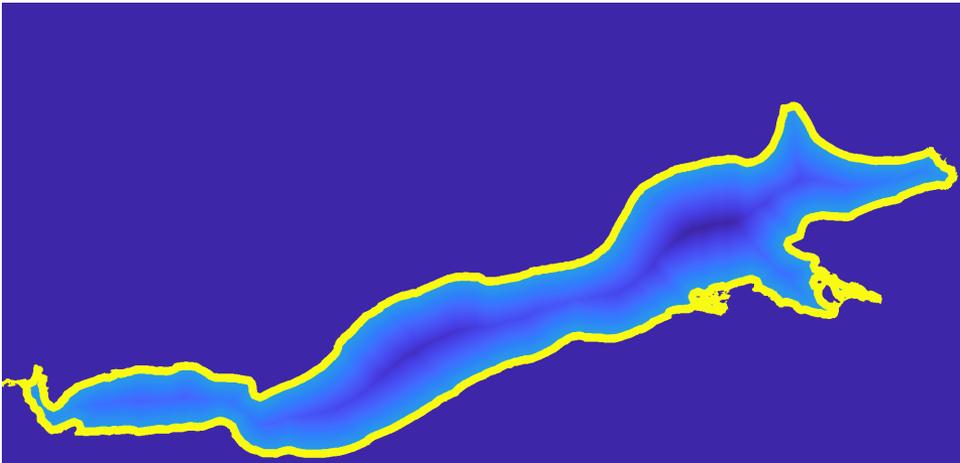


Figure 5.3: Interest value map with high values within 200m of shore, and then decreasing with distance to shore

Data set 2 - random moving targets

Data set 2 consists of random moving targets with random initial position. Simulated with the same procedure as in data set 1, except for using the randomly picked direction at each time instance. Example is shown in fig. 5.4b.

Data set 3 - stationary targets

Data set 3 has stationary targets with positions set in the same way as the initial positions in data set 1. An example is shown in fig. 5.4c.

Data set 4 - random stationary targets

Data set 4 has stationary targets with random positions. Example is shown in fig. 5.4d.

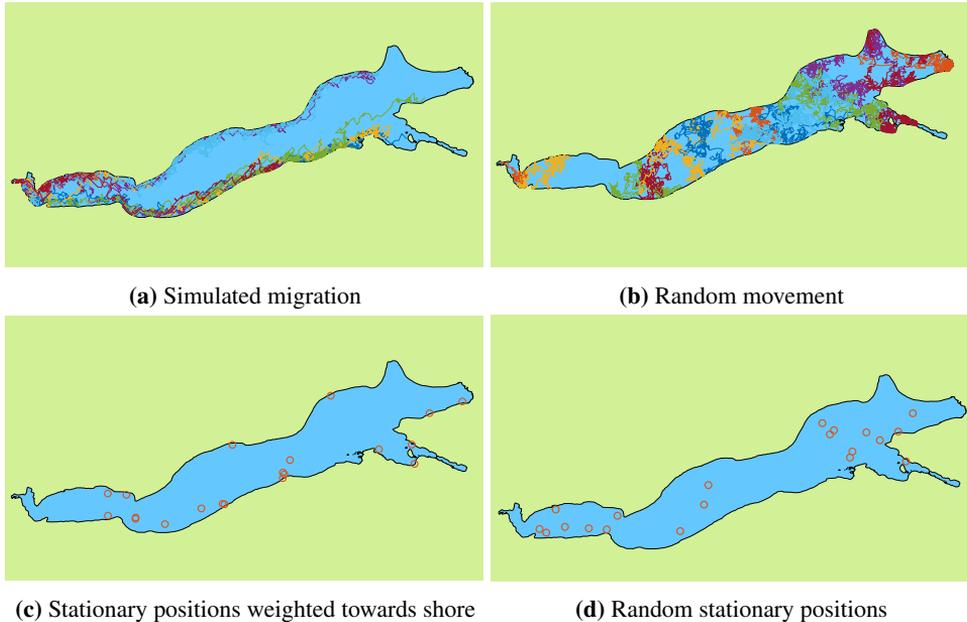


Figure 5.4: Target position data

5.2 USV model and control

A simple vessel model was used, where only the vessel kinematics are considered. Expanding eq. (2.3) gives the position, $[x_1, x_2] = [E, N]$, of the USV as :

$$\dot{x}_1 = u \cdot \cos(\psi) \quad (5.3)$$

$$\dot{y}_2 = u \cdot \sin(\psi) \quad (5.4)$$

Look-ahead based LOS-guidance was used to calculate inputs for the heading between way-points. The surge speed was held at a constant $u = V_d$. The maximum speed of the Otter with a sonar mounted is $4.5kts \approx 2.3m/s$ [19], this is assumed to be the maximum speed also with a hydrophone mounted. With a range of $R = 550m$ and a tag transmission interval of $T = 60s$, the speed limitation from section 4.6 becomes $V \leq 18.33m/s$, so the speed should be set to the maximum value:

$$V_d = 2.3m/s$$

The advantage of incorporating a vessel model in stead of directly finding positions from the path using a consideration of velocity and time passed, is the possibility to include environmental disturbances and more complex vessel models to evaluate the path planning under more realistic conditions.

5.3 Sensor model

The sensor model in the simulations is a definitive range sensor model, as in eq. (3.8):

$$P(\text{detection}|\text{transmission}) = \begin{cases} 1 & d \leq R \\ 0 & d > R \end{cases} \quad (3.8)$$

, where d is the Euclidean distance between the receiver and the tag, and R is the sensor range. The range was set to $R = 550m$. This is considerably lower than the calculated ideal ranges for fresh water calculated in section 2.1.2. This is done purposefully, as the ranges calculated tend to be very optimistic. Additionally, the range had to be set in such a way that the combination of search time and range gave interesting search paths, that did not cover the entire search area by very simple motions.

5.4 Search paths

Search paths for the simulation are generated with a time limit of $t_{search} = 10h$. This was chosen purposefully to avoid the complete coverage and greedy paths completely covering the search space which would give little discrepancy in performance in terms of the number of targets found. Since the complete coverage path does no longer give complete coverage, it is referred to as the coverage path from here on.

Since the USV otter has a size along its largest side of $2m$, the occupancy grid resolution was set to $r_{occupancy} \approx 5.6m$. The buffer for the occupancy grid was set to $b = 50m$, since the accuracy of the maps from the N50-series used is 10-15m[46], this will lead to the paths being at least 35m from the actual shore. The paths were generated based on 8-connectivity with Euclidean distance. This could possibly lead to collisions when traversing between cells that are diagonally placed with respect to each other. However, this is not the case due to the additional safety buffer of 35m. The resolution of the search map is set to $r_{searchmap} = 725m$ as a result of the searcher range and sampling (wanted resolution according to section 4.1.2 is $778m$). The threshold for open cells in the search map was set to $t = 0.25$. The grid with assigned start and goal point for the coverage and greedy search path is shown in fig. 5.5.

The coverage path is shown in fig. 5.6. The greedy search path was made in the up-sampled version of fig. 5.3 shown in fig. 5.7. Note that the cells closest to the shores are not necessarily the ones with highest interest value. This is because they contain areas with no probability at all of detecting fish (land). The corresponding greedy search path for one searcher is shown in fig. 5.8

For the Lévy flight search paths, the start point was randomly initialised. Only one path was generated for the coverage and greedy strategies. For the Lévy flight path, one instance was generated for each of the target position instances (100 paths), due to the random nature of the search. The parameters of the Lévy flight path were set to $\mu = 2$ and $l_{min} = R = 550m$. Three different instances are shown in fig. 5.9.

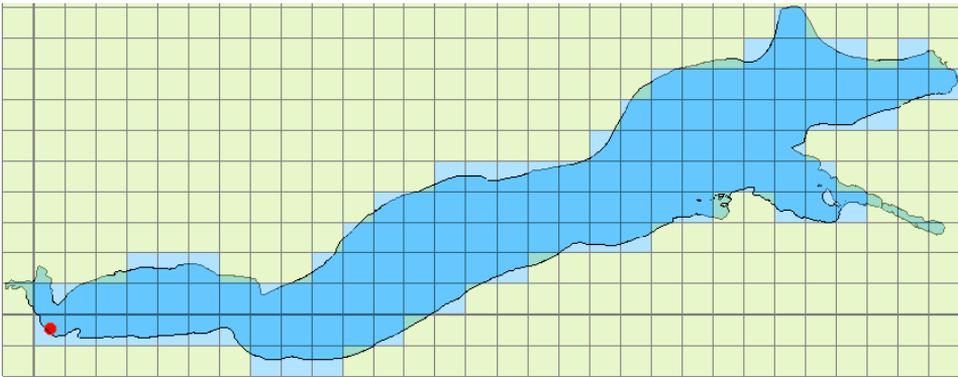
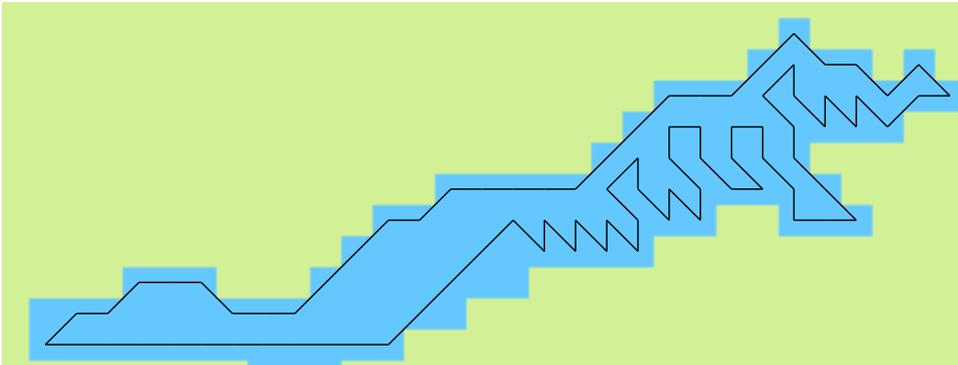
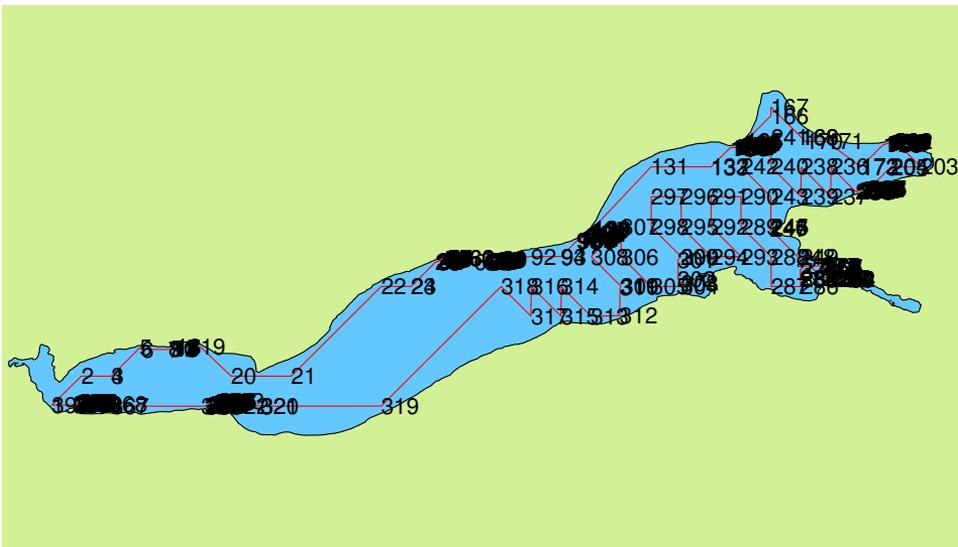


Figure 5.5: Search map with start and goal point



(a) Path in search map



(b) Path in polygonal map. Numbers mark way-points

Figure 5.6: Coverage search path

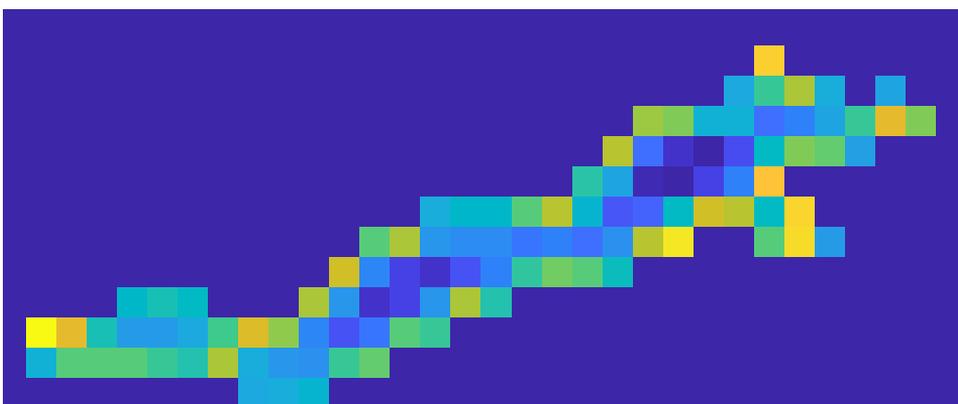
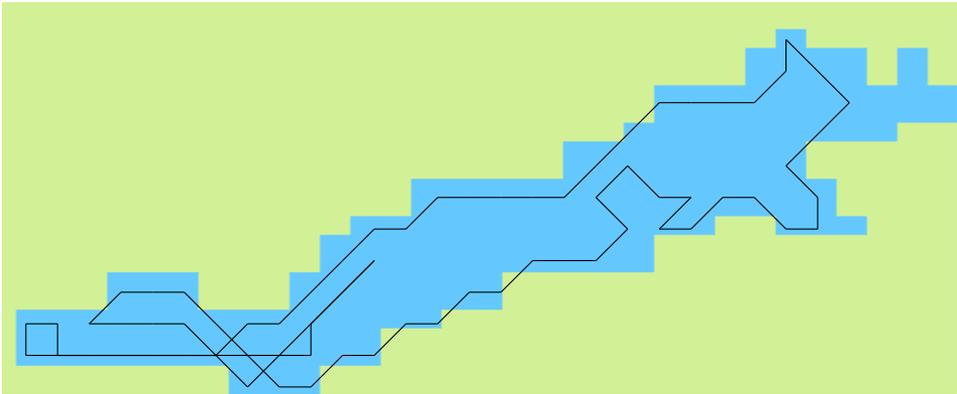
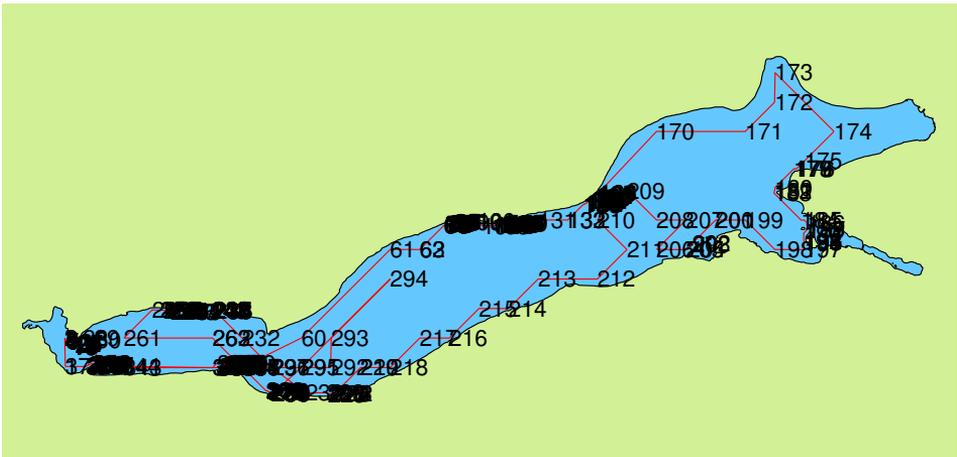


Figure 5.7: Interest value map for Hornindalsvatnet. Yellow = high value, blue = low value



(a) Path in search map



(b) Path in polygonal map. Numbers mark way-points

Figure 5.8: Greedy search path



Figure 5.9: Examples of Lévy flight paths

5.5 Results

The parameters used in the simulations are summarised in table 5.1. The remaining of this chapter will present results from the simulations and make a few remarks. The results are discussed further in chapter 7.

Parameter	Value	Explanation
t_{search}	10 h	search time
V_d	2.3 m/s	searcher speed
T	60s	tag transmission interval
r	5.66 m	occupancy grid resolution
a	15m	largest map inaccuracy
b	35 m	safety buffer (minimum distance from path to shore)
R	550 m	sensor range
rs	725 m	search map resolution
t	0.25	open cell threshold for search map

Table 5.1: Simulation parameters

Percentage of area covered, percentage of area covered for more than $T = 60s$ and percentage of interest value gathered for each of the paths is shown in table 5.2. This was calculated by checking whether the center point of all the cells in the occupancy grid with a resolution of $r = 5.66m$ were covered. This means that the areas restricted from navigation by the safety buffer are not considered. A heatmap for coverage for the coverage type and greedy path is shown in fig. 5.10.

Path	Area covered	Area covered for $T_{cov} \geq T = 60s$	Interest value
Coverage	90.9%	90.6%	83.4%
Greedy	78.2%	77.5%	80.1%
Lévy	45.1%(mean)	44.5%(mean)	36.9%(mean)

Table 5.2: Path metrics for the three search paths

Not surprisingly, the coverage path covers the most area seconded by the greedy path. The Lévy flight path covers the least area. This is due to the random nature of the path

generation, where previously visited areas are not excluded from being chosen. Although the greedy path covers notably less area than the coverage path, it gathers similar amounts of interest value. This can be explained by the greedy nature of the path and is also visible from comparing the interest value distribution in fig. 5.3 to the coverage for the greedy path in fig. 5.10. Further, the values for area covered and area covered for more than $T = 60s$ are very similar. This is likely due to the low speed of the vessel compared to the upper speed limit calculated.

Target detection metrics for each of the search path types are shown in tables 5.3 to 5.5. The metrics included are mean number of detections, median number of detections for each target and mean time to first detection. The mean number of detections is related to the detection probability in eq. (3.14). The number of detections per target is relevant for position estimation of the targets. The median is used for detections per target since there tends to be outliers of targets that have been detected a lot of times. The Lévy flight

dataset	Detections				Detections per target			Time to first detection(h:mm:ss)			
	mean	SE	min	max	median	min	max	mean	SE	min	max
1	16.3	1.80	12	20	7	0	94	0:11:08	0:13:57	0:00:10	1:05:26
2	15.4	1.97	10	19	7	0	92	0:25:05	0:22:02	0:00:23	1:36:58
3	18.2	1.20	15	20	8	0	107	0:13:29	0:19:33	0:00:07	1:39:13
4	15.5	1.81	11	20	8	0	106	0:21:32	0:22:27	0:00:07	1:55:10
sum	16.3	2.04	10	20	8	0	107	0:17:48	0:20:31	0:00:07	1:55:10

Table 5.3: Target detection metrics for greedy path

dataset	Detections				Detections per target			Time to first detection(h:mm:ss)			
	mean	SE	min	max	median	min	max	mean	SE	min	max
1	14.4	2.18	10	19	6	0	76	0:12:02	0:15:19	0:00:08	1:07:15
2	16.9	1.59	13	20	8	0	71	0:20:08	0:21:01	0:00:21	1:39:56
3	18.4	1.01	16	20	8	0	81	0:10:35	0:15:02	0:00:08	1:26:58
4	18.1	1.35	14	20	8	0	81	0:18:27	0:22:02	0:00:12	2:20:15
sum	16.9	2.23	10	20	8	0	81	0:15:18	0:19:00	0:00:08	2:20:15

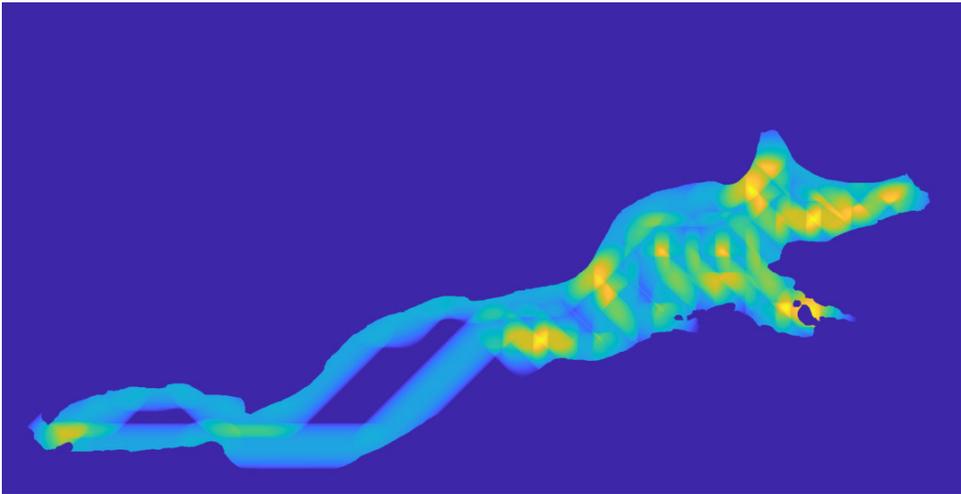
Table 5.4: Target detection metrics for coverage path

dataset	Detections				Detections per target			Time to first detection (h:mm:ss)			
	mean	SE	min	max	median	min	max	mean	SE	min	max
1	7.7	3.75	0	18	0	0	90	∞	-	0:00:01	∞
2	9.3	2.93	2	17	0	0	96	0:24:40	0:32:52	0:00:21	3:08:21
3	7.6	2.75	1	15	0	0	112	0:28:54	0:55:45	0:00:01	5:39:45
4	8.7	2.86	2	16	0	0	121	0:17:56	0:24:22	0:00:05	2:24:28
sum	8.3	3.16	0	18	0	0	121	∞	-	0:00:01	∞

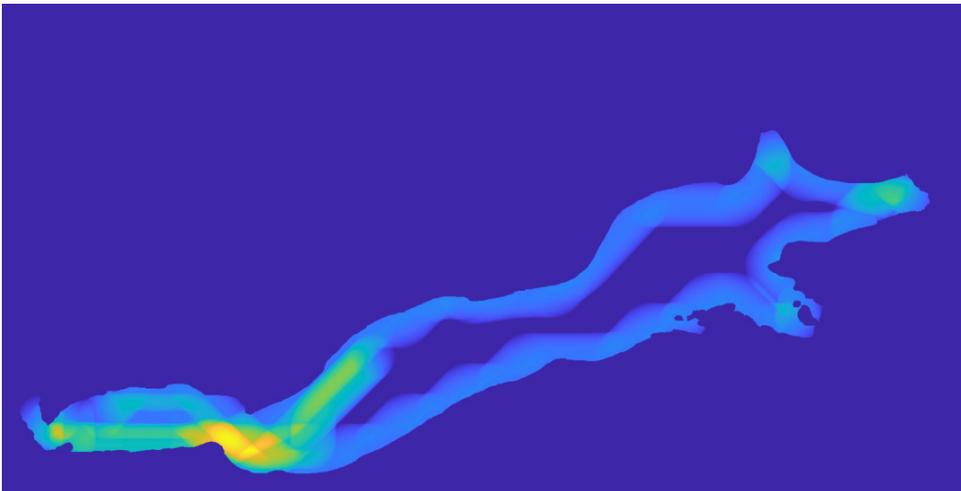
Table 5.5: Target detection metrics for Lévy flight path

path has a considerably lower value of mean detections over all data sets. It is also the only path that has instances that do not detect any targets. The median number of targets detected is also the lowest for this path. The coverage path has the highest mean number of detections when considering all four data sets. Considering it covers the most area, this is not surprising. Further, it should be noted that the greedy path has the best performance for the data set of migrating targets - data set 1. This is likely due to the amount of time the path spends along the shores.

Additionally to good detection capabilities, it was stated in section 3.5 that the search path should avoid known obstacles. An example of a search path successfully manoeuvring a known obstacle in the polygonal map is shown in fig. 5.11.



(a) Coverage path



(b) Greedy path

Figure 5.10: Heatmap of coverage

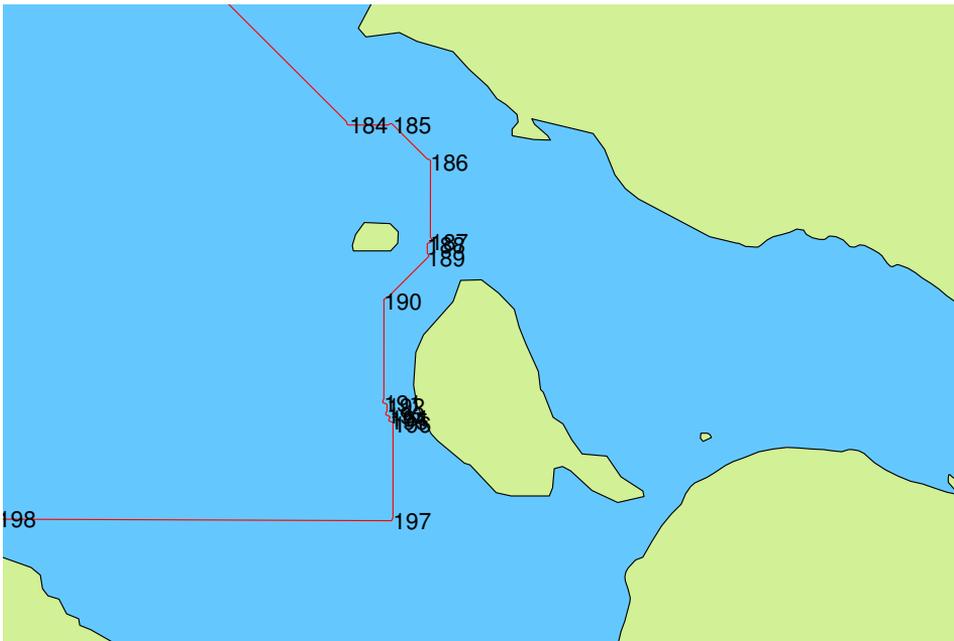


Figure 5.11: Successfully navigating between known obstacles. Number represent way-points

5.6 Paths for multiple searchers

Coverage type paths for 3 searchers in Hornindalsvatnet is shown in fig. 5.12. Since this is equivalent to comparing performance for the search strategies in a smaller area/over a longer period of time, no metrics were calculated for this type of multiple-searcher approach.

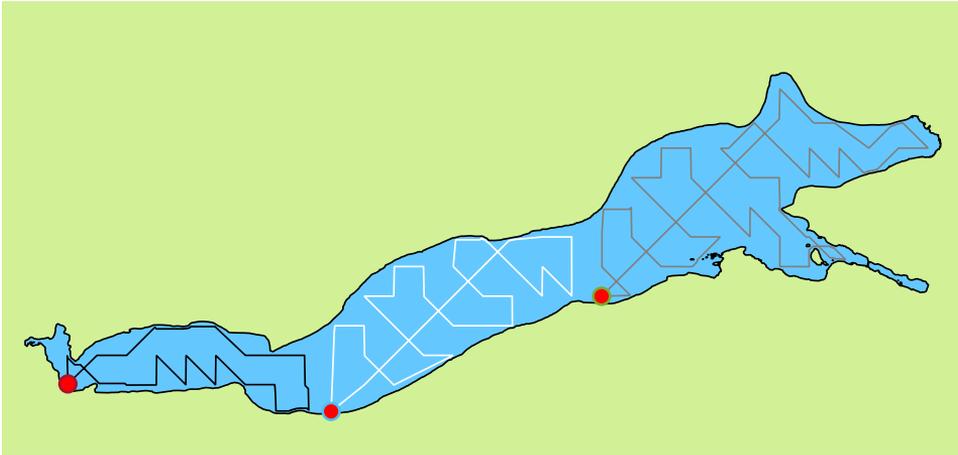


Figure 5.12: Coverage-like path for 3 searchers. Start and goal points marked in red

Implementation

6.1 LSTS toolchain

The LSTS toolchain developed at the University of Porto by the Underwater Systems and Technology Laboratory, is an open source software toolchain developed for vehicle control. The toolchain consists of the components. Three components of the toolchain are relevant for this implementation: *Neptus*, *Dune* and *IMC*. *Neptus* is a control and command centre, where missions can be planned, monitored and reviewed. *Dune* (Unified Navigation Environment) is the system that runs on the vehicle and provides modules for control, navigation, simulation, networking, sensing and actuation. The systems in the LSTS toolchain communicate using the IMC (Inter Module Communication) messaging protocol. The LSTS toolchain is frequently used at NTNU and the control systems for the USV Otter are currently being developed using the toolchain.

6.1.1 MatLab IMC communication

MatLab can communicate on the IMC message format through the use of Java bindings for IMC.¹ The IMC message type used to send a sequence of way-points in the implementation is shown in appendix B.1. This message, a *PlanSpecification*, can be digested by *Neptus*, and the sequence will appear in the console view.

6.2 Path generator application

A path generator application was implemented using MatLab App Designer (fig. 6.1). The application uses the N50-maps to generate coverage paths, and broadcasts the way-points as IMC-messages that can be interpreted by *Neptus*. The MatLab-code used to create and

¹See <https://github.com/LSTS/imcjava/wiki/Controlling-vehicles-with-Matlab>. Some of the information on this page is outdated, and functions should be crosschecked in the imcjava repository <https://github.com/LSTS/imcjava>

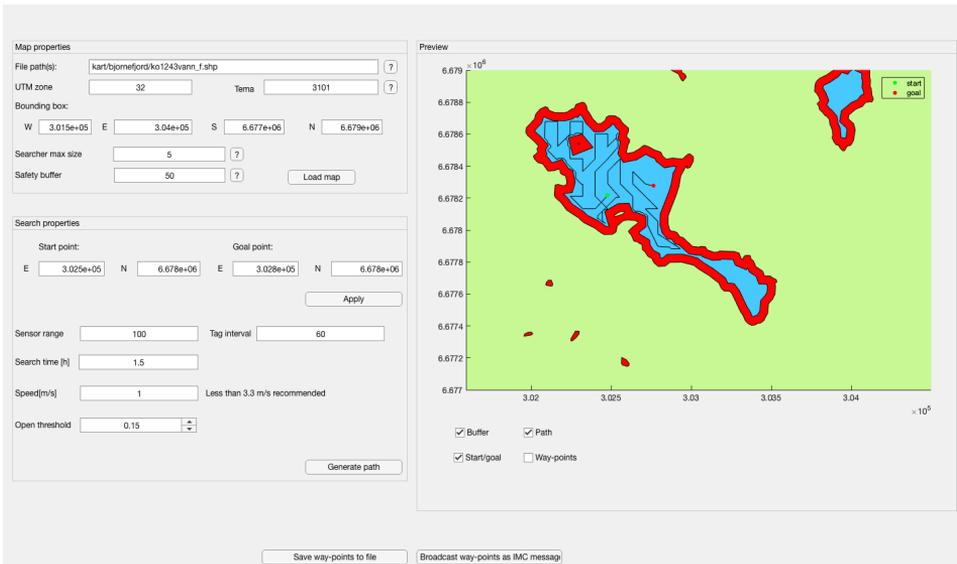


Figure 6.1: Generate coverage way-points app.

broadcast the IMC-message in the application is shown in appendix B.2.

The application can be found in the app-folder in the attached zip-archive. Note that the application depends on the *util* and *navigation* - folders and the relative path between the application and these folders must therefore not be changed. The work flow for generating a path and sending the way-points to Neptus in the application is as follows:

1. Enter map properties
2. Click *Load map*. This may take some time depending on the size of the bounding box area and the resolution of the occupancy grid. The map will appear in the preview panel when the calculation is complete.
3. Enter start and goal point and click *Apply*. The start and goal points will appear in the preview.
4. Enter remaining search properties
5. Click *Generate path*. The path will appear in the preview.
6. Click *Broadcast way-points as IMC message* to open the pop up shown in fig. 6.2.
7. Add the path to a 'libimc.jar'-file which defines the IMC Java-bindings.² Click *Connect to IMC* to initiate an IMC connection.
8. Select which node to broadcast to from the dropdown-menu. Refresh if the node is not, there but should be. Click *Send way-points*.

²More about this can be found here: <https://github.com/LSTS/imcjava>

A path for a 2 hour search in the fjord outside Børsa created in the MatLab application and imported in Neptus is shown in fig. 6.3.

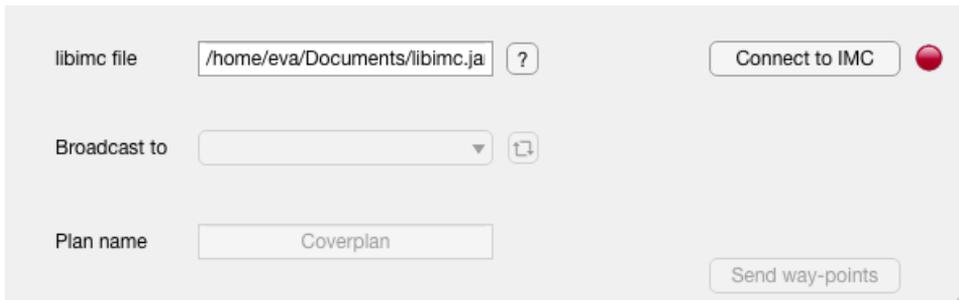


Figure 6.2: Send way-points



Figure 6.3: Way-points in Neptus

Discussion

The search paths created by the coverage and greedy method both outperform the Lévy paths in the simulations in chapter 5. The results for both paths of finding over 16/20 targets on average without capacity to cover the entire search area are promising. The coverage path has the best overall performance, but when considering the migrating targets from the first data set, the greedy path has better mean detection values. This indicates that when the information utilised for the greedy path is accurate, the approach is better than plain motions created to cover the search space. A logical next step would be to implement a method that finds an optimal search plan, and compare to the approaches developed in this thesis. This would create more ground for comparing performance of search paths. The results presented in chapter 5 points to the coverage path as a good candidate for search for acoustic tags. It requires no information about the search space except for the map geometry and gives good performance in terms of mean detected signals as well as a reasonable amount of signals detected per target. However, the high amount of signals per target may be an effect of the relatively low speed while traversing the path, compared to the upper limit on the speed given by eq. (4.1).

The Lévy paths in the simulations are heavily restricted by the geometry of the map. Inspecting the differences between fig. 4.9a, fig. 4.9b and fig. 5.9, shows that the paths used in the search are quite far from the original type of path. This may have had a detrimental effect on the results for the Lévy flight paths. Additionally to this, the Lévy flight strategy is claimed to be efficient when target locations are sparse, which they are not in the simulations.

A restricting factor of both the coverage and the greedy path is the maps they operate in. The sampling method for the search map heavily restricts the size of the cells by demanding the map be of a size $2^m \times 2^m$. This leads to the paths either having large areas of overlap if the search map resolution is chosen considerably higher than what suggested in section 4.1.2, or large areas being missed if the resolution is lower. Furthermore, the paths created in the occupancy grid become excessively jagged, as shown in fig. 5.11. This

effect is worse for 4-connectivity formulations than for the 8-connectivity Euclidean distance metric used to generate fig. 5.11. Realising that any type of path finding could be used for this task, it should be simple to improve this step of the way-point generation by implementing another path finding algorithm that generate smoother paths such as for instance A*. Another possible solution to this is to store the search map and occupancy grid as one common data structure known as a quad-tree. In a quad-tree, the resolution of an area is dependent on the structure of the original map. By doing the path-planning in such a structure, the jagged paths could possibly be avoided. This could also possibly open new possibilities and more flexibility with respect to the limited resolution of the search map.

Environmental factors have been not been incorporated past the restrictive shape of the maps. For instance, a strong current could lead to heavy deviations from the search path, or alternatively limit the search time by excessive battery usage. For the future, this could be considered in the path planning to for instance make paths that are efficient also with respect to environmental factors such as currents.

In the simulations in chapter 5, the vessel model is instantaneous and only considers vessel kinematics. This leads to perfect path-following, which is not realistic. Large discrepancies from the path will be detrimental to the results, and should be expected in real life experiments.

The paths generated in this thesis are generated based on maps of body of water. In reality, there are other permanent structures that are obstacles that should be avoided, such as floating docks and fish farming facilities. These permanent obstacles could be added by adding more layers from the N50 maps. However, the paths can be modified by adding safety buffers to the shore, where many obstacles are, and also editing the way-points manually in Neptus when imported. A larger restriction of the usage is that the USV Otter is currently 'blind', meaning it has no awareness of moving or unmapped obstacles.

No field tests were performed within the scope of the spring 2019 because of unforeseen events with the independent development of the control system on the USV Otter.

7.1 Future work

Recommendations for future work are listed below. Additionally to the recommendations listed below, local collision avoidance is needed in order to have a fully operational system that can perform autonomous search.

- Implement weighting of turns to avoid sharp turns in the coverage and greedy paths.
- Implement the search in a quad-tree structure in stead of the double map solution.
- Implement an optimal path finder using branch and bound methods and compare to the methods developed in this thesis.
- Perform field tests.

-
- Expand the path generator application to include other path types.
 - Expand the path generator application to allow for multiple searchers.

Bibliography

- [1] B. H. I. Mohus, "Fish telemetry manual," December 1983.
- [2] D. G. Tucker, *Applied underwater acoustics*. The Commonwealth and international library of science, technology, engineering and liberal studies. Physics division, Oxford: Pergamon Press, 1966.
- [3] R. J. Urick, *Principles of Underwater Sound*. New York: McGraw Hill, 1975.
- [4] N. S. Adams, J. W. Beeman, and J. H. Eiler, *Telemetry Techniques: A User Guide For Fisheries Research*. Bethesda, Maryland: American Fisheries Society, 2012.
- [5] N. E. Hussey, S. T. Kessel, K. Aarestrup, S. J. Cooke, P. D. Cowley, A. T. Fisk, R. G. Harcourt, K. N. Holland, S. J. Iverson, J. F. Kocik, J. E. M. Flemming, and F. G. Whoriskey, "Aquatic animal telemetry: A panoramic window into the underwater world," *Science*, vol. 348, no. 6240, 2015.
- [6] H. A. Urke, T. Kristensen, J. B. Ulvund, and J. A. Alfredsen, "Riverine and fjord migration of wild and hatchery-reared atlantic salmon smolts," *Fisheries Management and Ecology*, vol. 20, no. 6, pp. 544–552, 2013.
- [7] D. G. Pincock and S. V. Johnston, "Acoustic telemetry overview," in *Telemetry Techniques: A User Guide for Fisheries Research* (J. W. B. Noah S. Adams and J. H. Eiler, eds.), ch. 7.1, pp. 305–338, Bethesda, Maryland: American Fisheries Society, 2012.
- [8] B. M. Wetherbee, K. N. Holland, C. G. Meyer, and C. G. Lowe, "Use of a marine reserve in kaneohe bay, hawaii by the giant trevally, *caranx ignobilis*," *Fisheries Research*, vol. 67, no. 3, pp. 253 – 263, 2004.
- [9] C. L. Ng, K. W. Able, and T. M. Grothues, "Habitat use, site fidelity, and movement of adult striped bass in a southern new jersey estuary based on mobile acoustic telemetry," *Transactions of the American Fisheries Society*, vol. 136, no. 5, pp. 1344–1355, 2007.

-
- [10] D. Lidgard, W. Bowen, I. Jonsen, and S. Iverson, "Predator-borne acoustic transceivers and gps tracking reveal spatiotemporal patterns of encounters with acoustically tagged fish in the open ocean," *Marine Ecology Progress Series*, vol. 501, pp. 157–168, 03 2014.
- [11] C. M. Clark, C. Forney, E. Manii, D. Shinzaki, C. Gage, M. Farris, C. G. Lowe, and M. Moline, "Tracking and following a tagged leopard shark with an autonomous underwater vehicle," *Journal of Field Robotics*, vol. 30, no. 3, pp. 309–322, 2013.
- [12] M. W. Breece, D. A. Fox, K. J. Dunton, M. G. Frisk, A. Jordaan, and M. J. Oliver, "Dynamic seascapes predict the marine occurrence of an endangered species: Atlantic sturgeon *acipenser oxyrinchus oxyrinchus*," *Methods in Ecology and Evolution*, vol. 7, no. 6, pp. 725–733, 2016.
- [13] T. M. Grothues, J. Dobarro, J. Ladd, A. Higgs, G. Niezgod, and D. Miller, "Use of a multi-sensored auv to telemeter tagged atlantic sturgeon and map their spawning habitat in the hudson river, usa," in *2008 IEEE/OES Autonomous Underwater Vehicles*, pp. 1–7, Oct 2008.
- [14] T. Dodson, T. M. Grothues, J. H. Eiler, J. A. Dobarro, and R. Shome, "Acoustic-telemetry payload control of an autonomous underwater vehicle for mapping tagged fish," *Limnology and Oceanography: Methods*, vol. 16, no. 11, pp. 760–772, 2018.
- [15] A. Zolich, T. A. Johansen, J. A. Alfredsen, J. Kutteneuler, and E. Erstorp, "A formation of unmanned vehicles for tracking of an acoustic fish-tag," in *OCEANS 2017 - Anchorage*, pp. 1–6, 2017.
- [16] R. P. Jain, A. P. Aguiar, J. Borges de Sousa, A. Zolich, T. Johansen, J. A. Alfredsen, E. Erstorp, and J. Kutteneuler, "Localization of an acoustic fish-tag using the time-of-arrival measurements: Preliminary results using exogenous kalman filter," 2018.
- [17] A. Ekanger, "Developing an autonomous tracking system for the atlantic salmon," Master's thesis, NTNU, 2018.
- [18] S. S. Løvskar, "Positioning of periodic acoustic emitters using an omnidirectional hydrophone on an auv platform," Master's thesis, NTNU, 2017.
- [19] "Maritime robotics webpage, usv otter." <https://maritimerobotics.com/mariner-usv/otter/>. online: accessed 21.02.2019.
- [20] T. Wisting, "Slaget om atlanterhavet," *Store Norske Leksikon*, October 2018. online; accessed 20.02.2019.
- [21] B. O. Koopman, "Search and screening," tech. rep., Operations Evaluation Group, Office of the Chief of Naval Operations, Navy Department, Washington, D.C., 1946. OEG report no.56.
- [22] L. D. Stone, *Theory of optimal search*, vol. v. 118 of *Mathematics in science and engineering* ;. New York: Academic Press, 1975.
-

-
- [23] L. D. Stone, *Optimal search for moving targets*, vol. Volume 237 of *International series in operations research & management science* ; Cham, Switzerland: Springer International Publishing, 2016.
- [24] T. M. Kratzke, L. D. Stone, and J. R. Frost, “Search and rescue optimal planning system,” in *2010 13th International Conference on Information Fusion*, pp. 1–8, IEEE, 2010.
- [25] J. Berger and N. Lo, “An innovative multi-agent search-and-rescue path planning approach,” *Computers & Operations Research*, vol. 53, pp. 24 – 31, 2015.
- [26] L. Lin and M. A. Goodrich, “Uav intelligent path planning for wilderness search and rescue,” in *2009 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 709–714, Oct 2009.
- [27] Z. Llang Cao, Y. Huang, and E. Hall, “Region filling operations with random obstacle avoidance for mobile robots,” *Journal of Robotic Systems*, vol. 5, pp. 87 – 102, 04 1988.
- [28] H. Lau, S. Huang, and G. Dissanayake, “Optimal search for multiple targets in a built environment,” in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3740–3745, Aug 2005.
- [29] E. Galceran and M. Carreras, “A survey on coverage path planning for robotics,” *Robotics and Autonomous Systems*, vol. 61, no. 12, pp. 1258 – 1276, 2013.
- [30] E. M. Arkin, S. P. Fekete, and J. S. Mitchell, “Approximation algorithms for lawn mowing and milling,” *Computational Geometry*, vol. 17, no. 1, pp. 25 – 50, 2000.
- [31] G. M. Viswanathan, S. V. Buldyrev, S. Havlin, M. G. E. D. Luz, E. P. Raposo, and H. E. Stanley, “Optimizing the success of random searches,” *Nature*, vol. 401, no. 6756, 1999.
- [32] G. Viswanathan, V. Afanasyev, S. Buldyrev, and E. Murphy, “Levy flight search patterns of wandering albatrosses,” *Nature*, vol. 381, no. 6581, pp. 413–413, 1996.
- [33] A. M. Reynolds, “Current status and future directions of lévy walk research,” *Biology Open*, vol. 7, no. 1, 2018.
- [34] J. Travis, “Do wandering albatrosses care about math?,” *Science*, vol. 318, no. 5851, pp. 742–743, 2007.
- [35] N. E. Humphries, N. Queiroz, J. R. M. Dyer, N. G. Pade, M. K. Musyl, K. M. Schaefer, D. W. Fuller, J. M. Brunnschweiler, T. K. Doyle, J. D. R. Houghton, G. C. Hays, C. S. Jones, L. R. Noble, V. J. Wearmouth, E. J. Southall, and D. W. Sims, “Environmental context explains lévy and brownian movement patterns of marine predators,” *Nature*, vol. 465, no. 7301, 2010.
- [36] R. J. Urick, *Principles of Underwater Sound*. New York: McGraw Hill, 1983.
-

-
- [37] “Thelma biothel tbr700.” <https://www.thelmabiotel.com/receivers/tbr-700/>. Accessed: 20.05.2019.
- [38] “Thelma biothel 13mm transmitter.” <https://www.thelmabiotel.com/transmitters/13mm/>. Accessed: 20.05.2019.
- [39] *ESRI Shapefile Technical Description*, 1998. Accessed:04.06.19.
- [40] “Referanserammer for noreg.” <https://kartverket.no/Posisjonstjenester/bruke-referanserammer/Referanserammer-for-Noreg/>, September 2018. Accessed: 26.03.2019.
- [41] “Kartprosjeksjoner.” <https://kartverket.no/Posisjonstjenester/Kartprosjeksjoner/>, September 2018. Accessed: 26.03.2019.
- [42] T. H. Bryne, “Lecture notes in ttk5 - kalman filtering and navigation,” September 2018.
- [43] T. Fossen, *Handbook of Marine Craft Hydrodynamics and Motion Control*. John Wiley & Sons, 2011.
- [44] Y. L. Tong, *The Bivariate Normal Distribution*, pp. 6–22. New York, NY: Springer New York, 1990.
- [45] A. Zelinsky, R. Jarvis, J. C. Byrne, and S. Yuta, “Planning paths of complete coverage of an unstructured environment by a mobile robot,” in *In Proceedings of International Conference on Advanced Robotics*, pp. 533–538, 1993.
- [46] “Norge 1:50000.” <https://www.kartverket.no/Kart/Norge-150-000/>, January 2019. Accessed: 15.05.2019.
- [47] E. Halttunen, A. H. Rikardsen, J. G. Davidsen, E. B. Thorstad, and J. B. Dempson, *Survival, Migration Speed and Swimming Depth of Atlantic Salmon Kelts During Sea Entry and Fjord Migration*, pp. 35–49. Dordrecht: Springer Netherlands, 2009.

A Master thesis assignment

This page is empty. The master thesis assignment is added on the next page.



MASTER'S THESIS ASSIGNMENT (30 Stp.)

Name: Eva Kristiansen
Program: Engineering Cybernetics
Title: Strategies for search and detection of acoustic transmitters using unmanned surface vehicle
Title (Norw.): Strategier for søk og deteksjon av akustiske sendere med ubemannet overflatefartøy

Project description

Acoustic fish telemetry represents a powerful scientific tool in the study of migratory behaviour and movement ecology of fish. Advances in acoustic transmitter/receiver technology in combination with the growing availability of new enabling technologies such as unmanned vehicles and autonomous systems, will make transformative shifts in the state of the art within fish telemetry, which will boost the yield of relevant biological data and ultimately our understanding of fish ecology.

In this project we intend to explore the potential of using of networked unmanned surface vehicles (USVs) as robotic carriers of acoustic telemetry receivers for search, detection and stealthy tracking of tagged fish. More specifically, the core task of this assignment is to develop methods for devising efficient USV search plans in a given sea area and with given limitations in sensor sensitivity, vehicle performance, environmental conditions, and knowledge of target distributions. The project consists of the following tasks:

- Literature survey and study of the theory of optimal search, including bioinspired strategies such as Lévy flight/walk.
- Review of the application of autonomous vehicles for search purposes, and particularly cases similar to the target problem.
- Based on the acquired theoretical background, develop an explicit formulation of problem of calculating efficient search plans for the USV with respect to the:
 - Sensor characteristics (receiver/transmitter, signal propagation)
 - Vehicle performance and limitations
 - Target behaviour and distribution probability
 - Characteristics of the search area and environmental conditions
 - Single or multiple cooperating vehicles
- Develop a simulation model of the system and candidate planning algorithms, and evaluate the performance of different search strategies.
- Develop a search plan generator that takes in relevant inputs and provides output that can be interpreted by the unified navigation environment (Dune/Neptus).
- Plan and perform a field test using USV Otter and discuss practical aspects and use of the search algorithm.

Project start: 21. January 2019
Project due: 17. June 2019
Host institution: Department of Engineering Cybernetics, NTNU
Supervisors: Jo Arve Alfredsen, NTNU/DEC

B Matlab IMC communication

B.1 Plan specification IMC message

Header	8 fields
sync	65108 [0xFE54]
mgid	551
size	307 byte
timestamp	2019-Jun-12 10:06:12.412 UTC
src	24286 [0x5EDE]
src_ent	0
dst	23077 [0x5A25]
dst_ent	255

PlanSpecification	9 fields														
plan_id	Coverplan														
description															
vnamespace															
variables															
start_man_id	goto1														
maneuvers	PlanManeuver	4 fields				PlanManeuver	4 fields				PlanManeuver	4 fields			
	maneuver_id	goto1				maneuver_id	goto2				maneuver_id	goto3			
	data	Goto	11 fields			Goto	11 fields			Goto	11 fields				
			timeout	0 s			timeout	0 s			timeout	0 s			
			lat	1.10525224			lat	1.10528278			lat	1.10525224			
			lon	0.17599632			lon	0.17590938			lon	0.17599632			
			z	0.0 m			z	0.0 m			z	0.0 m			
			z_units	DEPTH			z_units	DEPTH			z_units	DEPTH			
			speed	2.0			speed	2.0			speed	2.0			
			speed_units	METERS_PS			speed_units	METERS_PS			speed_units	METERS_PS			
			roll	0.00000000			roll	0.00000000			roll	0.00000000			
			pitch	0.00000000			pitch	0.00000000			pitch	0.00000000			
			yaw	0.00000000			yaw	0.00000000			yaw	0.00000000			
	custom	tuplelist		custom	tuplelist		custom	tuplelist							
start_actions					start_actions					start_actions					
end_actions					end_actions					end_actions					
transitions	PlanTransition	4 fields				PlanTransition	4 fields				PlanTransition	4 fields			
	source_man	goto1				source_man	goto2				source_man	goto3			
	dest_man	goto2				dest_man	goto3				dest_man				
	conditions	ManeuverIsDone				conditions	ManeuverIsDone				conditions				
	actions					actions					actions				
start_actions															
end_actions															

B.2 Sending IMC messages in Matlab

The functions given here are part of a MatLab application, which is a popup from the main application. The application has private variables defining a waypoint list (`app.wps`), the current UTM zone (`app.UTM`) and a java imc object (`app.imc`). The function in listing B.1 initiates an IMC connection. The function in listing B.2 builds and sends an IMC message. The IMC connection is stopped upon closing the app as shown in listing B.3.

Listing B.1: ConnecttoIMCButtonPushed

```
1 function ConnecttoIMCButtonPushed(app, event)
2     p = app.libimcfileEditField.Value;
3
4     app.ConnecttoIMCButton.Enable = 'off'; % Disable button - ...
        working on stuff
5
6     javaaddpath(p);
7     import pt.lsts.imc.*
8     import pt.lsts.imc.net.*
9
10    try
11        app.imc = IMCProtocol(); % This will fail if the path ...
            is wrong
12    catch
13        javarmpath(p); % Delete wrongly appended item to java path
14
15        % Display error message
16        msg = sprintf('Path %s is invalid',p);
17        uialert(app.SendIMCmessageUIFigure,msg,'Invalid path');
18        app.ConnecttoIMCButton.Enable = 'on'; % Enable ...
            button, operation failed so used should be able to ...
            try again
19        return
20    end
21
22    % Wait for systems
23    sys = app.imc.systems();
24    while length(sys) < 1
25        pause(5) % pause to allow system to get values
26        sys = app.imc.systems();
27    end
28
29    % Update dropdown
30    app.updateBroadcasttoDropDown(cell(app.imc.systems()))
31
32    % Save path to jar-file
33    app.pathtolibimc = p;
34
35    % Update view with new options
36    app.Lamp.Color = [34 139 34]./255;
37    app.BroadcasttoDropDown.Enable = 'on';
38    app.PlannameEditField.Enable = 'on';
39    app.SendwaypointsButton.Enable = 'on';
40    app.refreshButton.Enable = 'on';
41
```

Listing B.2: SendwaypointsButtonPushed

```

1 function SendwaypointsButtonPushed(app, event)
2
3     % Need to re-import
4     import pt.lsts.imc.*
5     import pt.lsts.imc.net.*
6
7     % Set plan name
8     planid = app.PlannameEditField.Value;
9     if isempty(planid)
10        planid = 'plan';
11    end
12
13    % Create planspec with maneuvers and transitions
14    planspec = PlanSpecification();
15    planspec.setPlanId(planid);
16    maneuvers = planspec.getManeuvers();
17    transitions = planspec.getTransitions();
18
19
20    for i=1:size(app.wps,2)
21
22        wp = app.wps{i};
23
24        % Find lat-lon in radians
25        [lat,lon] = utm2ll(wp(1),wp(2),app.UTM);
26        lat = deg2rad(lat);
27        lon = deg2rad(lon);
28
29        % Build maneuvers
30        goto = Goto();
31        goto.setZ(0);
32        goto.setZUnitsStr('DEPTH');
33        goto.setSpeed(app.speed);
34        goto.setSpeedUnitsStr('METERS_PS');
35        goto.setLat(lat);
36        goto.setLon(lon);
37
38        maneuver = PlanManeuver();
39        maneuver.setManeuverId(sprintf('goto%d',i))
40        maneuver.setData(goto);
41
42        maneuvers.add(maneuver);
43
44        % Build transitions
45        if i ≠ size(app.wps,2)
46            transition = PlanTransition();
47            transition.setConditions('ManeuverIsDone');
48            transition.setSourceMan(sprintf('goto%d',i))
49            transition.setDestMan(sprintf('goto%d',i+1))
50
51            transitions.add(transition);
52        end

```

```
53
54     end
55     % Finish planspec
56     planspec.setManeuvers (maneuvers);
57     planspec.setTransitions (transitions);
58     planspec.setStartManId ('gotol');
59
60     % Broadcast planspec
61     sendto = app.BroadcasttoDropDown.Value;
62
63     app.imc.sendMessage (sendto, planspec);
64 end
```

Listing B.3: SendIMCmessageUIFigureCloseRequest

```
1 function SendIMCmessageUIFigureCloseRequest (app, event)
2     % Enable button in calling app
3     app.callingapp.BroadcastwaypointsasIMCmessageButton.Enable ...
4         = 'on';
5
6     % Stop IMC connection
7     if ~isempty (app.pathtolibimc)
8         % Stop IMC connection
9         app.imc.stop();
10        app.imc = [];
11        javarmpath (app.pathtolibimc);
12    end
13
14    % Delete app
15    delete (app)
16 end
```

